Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Υλοποίηση Ολοκληρωμένου Συστήματος Μηχανικής Μάθησης για την Πρόβλεψη Τιμών Ακινήτων

Διπλωματικη Εργασια

ΣΤΥΛΙΑΝΙΔΗΣ ΧΡΗΣΤΟΣ

Επιβλέπων: Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εργαστηριο Συστηματων Τεχνητης Νοημοσυνης και Μαθησης
Αθήνα, Ιούνιος 2021

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Υλοποίηση Ολοκληρωμένου Συστήματος Μηχανικής Μάθησης για την Πρόβλεψη Τιμών Ακινήτων

Διπλωματικη Εργασια

ΣΤΥΛΙΑΝΙΔΗΣ ΧΡΗΣΤΟΣ

Επιβλέπων: Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 15/6/2021.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Ανδρέας-Γεώργιος Σταφυλοπάτης	Κόλλιας Στέφανος	Στάμου Γεώργιος
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2021

(Υπογραφή)

.....

Χρηστος Στυλιανιδης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η αγοραπωλησία ακινήτων αποτελεί ένα αναπόσπαστο στοιχείο της κοινωνίας. Αποτελεί κομβικό σημείο στη ζωή σχεδόν κάθε ανθρώπου και προσφέρει τη δυνατότητα δημιουργίας σημαντικού πλούτου για όποιον μπορεί να κρίνει εύστοχα τις τιμές. Συνεπώς, μοντέλα Μηχανικής Μάθησης που μπορούν να δώσουν εύστοχες προβλέψεις τιμών αποτελούν αντικείμενα μεγάλου ενδιαφέροντος.

Σε επίπεδο δημοσίως διαθέσιμης έρευνας έχουν γίνει σημαντικά βήματα, τα οποία επικεντρώνονται κυρίως στην αξιολόγηση της απόδοσης διαφορετικών αλγορίθμων σε ήδη διαθέσιμα σύνολα δεδομένων. Τα σύνολα δεδομένων αυτά έχουν περάσει προσεκτική επεξεργασία προκειμένου να περιέχουν ορθά και όσο γίνεται πιο πλήρη, αν και λιγοστά σε ποσότητα, στοιχεία.

Εναντίως, στον ιδιωτικό τομέα πλήθος εταιριών που ειδικεύονται στην αγοραπωλησία ακινήτων έχουν πρόσβαση σε τεράστιες ποσότητες δεδομένων με τα οποία μπορούν να τροφοδοτήσουν τα μοντέλα τους. Το αποτέλεσμα είναι τα μοντέλα αυτά συχνά να έχουν καλύτερες δυνατότητες γενίκευσης σε νέα δεδομένα του πραγματικού κόσμου.

Σκοπός της παρούσας διπλωματικής εργασίας είναι να προσομοιώσει την προσέγγιση μιας πιθανής νεοφυούς επιχείρησης η οποία, χωρίς να διαθέτει ένα έτοιμο σύνολο δεδομένων, επιθυμεί να δραστηριοποιηθεί στον τομέα της πρόβλεψης τιμών ακινήτων. Πραγματεύεται την συλλογή δεδομένων από διαφορετικές πηγές, την αξιολόγηση της ποιότητάς τους, την επεξεργασία τους και τέλος την εκπαίδευση μοντέλων μηχανικής μάθησης με βάση τα δεδομένα αυτά και την αξιολόγηση της απόδοσής τους.

Λέξεις Κλειδιά

Πρόβλεψη Τιμών Ακινήτων, Ακίνητα, Μηχανική Μάθηση, Εξόρυξη Δεδομένων, Ανάλυση Δεδομένων, Επεξεργασία Δεδομένων, Σωλήνωση Δεδομένων, Web Scraping, Νευρωνικά Δίκτυα, Τυχαία Δάση, Στοχαστική Κάθοδος Κλίσης, Ενίσχυση Κλίσης.

Abstract

Buying and selling real estate is an integral part of society. It is a key point in the life of almost every person and offers the possibility of creating significant wealth for anyone who can accurately judge prices. Therefore, Machine Learning models that can provide accurate price predictions are subjects of great interest.

In terms of research that is publicly available, important steps have been taken which focus mainly on evaluating the performance of different algorithms in already available datasets. These datasets have been carefully processed in order to contain correct and complete, albeit small in quantity, data.

In the private sector, on the other hand, a number of real estate companies have access to vast amounts of data with which they can power their models. The result is that these models often have better generalization capabilities on new real-world data.

The purpose of this thesis is to simulate the approach of a potential start-up which, without having a ready-made dataset, wishes to operate in the field of real estate price forecasting. It deals with the collection of data from different sources, the evaluation of their quality, their processing and finally the training of machine learning models based on this data and the evaluation of their performance.

Keywords

Real Estate Price Prediction, Real Estate, Machine Learning, Data Extraction, Data Analysis, Data Processing, Data Pipeline, Web Scraping, Neural Networks, Random Forests, Stochastic Gradient Descent, Gradient Boosting.

Ευχαριστίες

Με την ευκαιρία της ολοκλήρωσης της διπλωματικής μου εργασίας θα ήθελα να ευχαριστήσω τον κ. Ανδρέα Σταφυλοπάτη για την ευκαιρία που μου έδωσε να εκπονήσω τη διπλωματική μου εργασία στο Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης.

Θα ήθελα να ευχαριστήσω επίσης τον κ. Γεώργιο Σιόλα ο οποίος μου προσέφερε πολύτιμη βοήθεια κατά τη διάρκεια της εκπόνησης της διπλωματικής μου εργασίας και έδειξε ενδιαφέρον από την πρώτη στιγμή για το αντικείμενο που αυτή πραγματεύεται.

Ευχαριστώ ακόμη όλους τους φίλους που έκαναν τα φοιτητικά μου χρόνια μια περίοδο που θα θυμάμαι για πάντα και τους καθηγητές της σχολής που μοιράστηκαν μαζί μου το πάθος τους για την επιστήμη των υπολογιστών.

Τέλος, θέλω να ευχαριστήσω την οικογένεια μου που είναι πάντα δίπλα μου, δείχνοντας κάθε μέρα αγάπη και ενδιαφέρον. Ό,τι καταφέρνω το οφείλω πρώτα σε αυτούς.

Χρήστος Στυλιανίδης

Περιεχόμενα

Κατάλογος Σχημάτων

Κεφάλαιο 1

Εισαγωγή

Το πεδίο της Μηχανικής Μάθησης έχει γνωρίσει μεγάλη άνθηση την τελευταία δεκαετία χάρη στην αύξηση των διαθέσιμων δεδομένων και υπολογιστικών πόρων καθώς και τη βελτίωση των αλγορίθμων στους οποίους βασίζεται. Το αποτέλεσμα είναι να παρατηρείται η χρήση της σε διαρκώς περισσότερους τομείς και με διαρκώς πιο εντυπωσιακά αποτελέσματα. Ένας τομέας στον οποίο η Μηχανική Μάθηση βρίσκει εφαρμογή είναι η πρόβλεψη των τιμών ακινήτων.

Επί πολλά χρόνια ο υπολογισμός της αξίας ενός ακινήτου αποτελούσε προϊόν της εργασίας εμπειρογνωμόνων, οι οποίοι χρησιμοποιώντας μαθηματικά μοντέλα αλλά και προσωπική εμπειρία, εκτιμούσαν την ιδανική τιμή ενός ακινήτου. Καθώς όμως διαρκώς περισσότερα δεδομένα που αφορούσαν την πώληση ακινήτων γίνονταν διαθέσιμα σε ηλεκτρονική μορφή, το έδαφος άρχισε να γίνεται ολοένα και πιο πρόσφορο για την εφαρμογή Μηχανικής Μάθησης για να συμπληρώσει, ακόμα και να αντικαταστήσει, την ανθρώπινη εκτίμηση. Αποτέλεσμα των παραπάνω είναι η συνεχής ανάπτυξη και βελτίωση συστημάτων Μηχανικής γνώσης για την εκτίμηση τιμών ακινήτων μέχρι και σήμερα.

Βασικός παράγοντας που καθορίζει την ευστοχία ενός τέτοιου συστήματος είναι η ποιότητα και ποσότητα των δεδομένων. Παρόλο που βελτιώσεις συνεχίζουν να λαμβάνουν χώρα σε επίπεδο αλγορίθμων, το τελικό σύστημα είναι τόσο καλό όσο τα δεδομένα του επιτρέπουν. Συχνά μάλιστα υπάρχει μια αντιστρόφως ανάλογη σχέση μεταξύ ποιότητας και ποσότητας δεδομένων.

Συνεπώς, για την ανάπτυξη ενός αποδοτικού συστήματος πρόβλεψης τιμών ακινήτων απαιτείται ιδιαίτερη επιμέλεια στην δημιουργία του σετ δεδομένων που θα το τροφοδοτήσει, ώστε να έχει τόσο το απαραίτητο μέγεθος όσο και την απαραίτητη ποιότητα. Η σημαντικότερη δυσκολία σε ένα τέτοιο εγχείρημα είναι η έλλειψη ποιοτικών δημοσίως διαθέσιμων δεδομένων. Τα δεδομένα αποτελούν συχνά πνευματική ιδιοκτησία ή μπορεί να περιέχουν ελλειπείς ή λανθασμένες πληροφορίες. Έτσι, καθίσταται δύσκολη η δημιουργία ενός συστήματος υψηλών επιδόσεων.

1.1 Αντικείμενο της διπλωματικής

Αντικείμενο της διπλωματικής αυτής αποτελεί η δημιουργία ενός ολοκληρωμένου Pipeline Μηχανικής Μάθησης με στόχο την πρόβλεψη τιμών ακινήτων. Το Pipeline θα αποτελείται από τη συλλογή και την επεξεργασία των σχετικών δεδομένων, καθώς και την σύγκριση των επιδόσεων διαφορετικών αλγορίθμων πάνω στα δεδομένα αυτά.

Σκοπός είναι η προσωμοίωση της δραστηριότητας μιας νεοφυούς επιχείρησης η οποία επιθυμεί να δραστηριοποιηθεί στον τομέα της αγοραπωλησίας ακινήτων αλλά δεν διαθέτει ένα έτοιμο dataset.

Η τοποθεσία που αναλύεται είναι η πόλη της Νέας Υόρκης. Τα δεδομένα για τα ίδια τα ακίνητα προέρχονται από τον ιστότοπο Zillow στον οποίο αναρτώνται αγγελίες πώλησης ακινήτων. Δεδομένα που αφορούν χαρακτηριστικά μιας τοποθεσίας όπως η ποιότητα των σχολείων ή τα αξιοθέατα προέρχονται από διαρετικές πηγές που αναλύονται με λεπτομέρεια στα ακόλουθα κεφάλαια. Για την συλλογή των δεδομένων χρησιμοποιείται Web Scraping και δημοσίως διαθέσιμα APIs.

Στο τέλος, οι προβλέψεις του Pipeline που δημιουργήθηκε συγκρίνονται με το Zillow Estimate (Zestimate), προβλέψεις από το ίδιο το Zillow δηλαδή, προκειμένου να κριθεί η απόδοση του συστήματος και να βγουν τα σχετικά συμπεράσματα.

1.2 Οργάνωση του τόμου

Το υπόλοιπο της διπλωματικής οργανώνεται ως εξής:

Το **Κεφάλαιο 2** ξεκινά με την παρουσίαση της αγοράς ακινήτων. Ακολουθεί η ανάλυση της έννοιας του Web Scraping και των σχετικών με αυτό εργαλείων που χρησιμοποιήθηκαν. Ακολούθως, παρουσιάζεται η έννοια του public facing API. Έπειτα, στο **Κεφάλαιο 3**, παρουσιάζεται το θεωρητικό υπόβαθρο της εργασίας, δηλαδή τα frameworks και οι αλγόριθμοι οι οποίοι δοκιμάζονται στα πλαίσια της δημιουργίας του Pipeline.

Το **Κεφάλαιο 4** παρουσιάζει με λεπτομέρεια τη διαδικασία εξαγωγής δεδομένων από τις διαφορετικές πηγές που χρησιμοποιήθηκαν. Γίνεται αναφορά σε κάθε πηγή ξεχωριστά και αιτιολογείται ο τρόπος με τον οποίο εξήχθησαν τα δεδομένα από την εκάστοτε πηγή. Τέλος, παρουσιάζεται η δομή και το περιεχόμενο των συλλεχθέντων δεδομένων.

Στο **Κεφάλαιο 5** περιγράφεται η διαδικασία επεξεργασίας των δεδομένων. Τα δεδομένα παιρνούν από διάφορα στάδια καθαρισμού και δεδομένα από διαφορετικές πηγές ενώνονται σε ένα ενιαίο dataset.

Ακολουθεί η διαδικασία σύγκρισης των διαφορετικών αλγορίθμων πάνω στα δεδομένα στο **Κεφάλαιο 6**. Η απόδοση των εκπαιδευμένων μοντέλων δοκιμάζεται σε ένα σύνολο δεδομένων που φυλάχτηκε εξαρχής για τον σκοπό αυτό (test data)- σαν προσομοίωση του deployment του μοντέλου σε ένα πραγματικό παραγωγικό περιβάλλον. Τέλος, αξιολογείται η απόδοση των μοντέλων και συγκρίνεται με αυτήν του Zestimate - του μοντέλου που χρησιμοποιεί το ίδιο το Zillow.

Στο **Κεφάλαιο 7** συγκεντρώνονται περιληπτικά τα συμπεράσματα της εργασίας και προτείνονται μελλοντικές επεκτάσεις της.

1.3 Σχετική έρευνα

Το πεδίο της αγοράς ακινήτων έχει αποτελέσει το κέντρο έρευνας η οποία εστιάζει επί το πλείστον στην εύρεση αποδοτικών και εύστοχων αλγορίθμων.

Οι Byeonghwa Park και Jae Kwon Bae παρουσίασαν τα αποτελέσματα μιας έρευνας για το αν ένα ακίνητο θα πωληθεί για περισσότερα ή λιγότερα χρήματα της τιμής αγγελίας [1] το 2015. Χρησιμοποίησαν τους αλγόριθμους C4.5, RIPPER, Naïve Bayesian και AdaBoost. Οι Mohammad Hossein Rafiei και Hojjat Adeli έχουν παρουσιάσει τις προβλέψεις τους για την τιμή πώλησης πριν κατασκευαστεί καν ένα ακίνητο [2] το 2016. Χρησιμοποίησαν μια μηχανή Boltzmann και γενετικούς αλγόριθμους. Οι Guangli Liu και Xiaohui Zong το 2017 εξερεύνησαν την δυνατότητα πρόβλεψης της τιμής μεταπώλησης ακινήτων [3]. Η πρότασή τους βασίζεται στη χρήση Twin Support Vector Regression. Επίσης το 2018, ο Alejandro Baldominos και οι συνεργάτες του χρησιμοποίησαν support vector regression, k-nearest neighbors, regression trees και multi-layer perceptrons για την πρόβλεψη τιμών ακινήτων στη Μαδρίτη της Ισπανίας [4].

Πέρα από τους παραπάνω αλγόριθμους οι οποίοι εστιάζουν σε καταγεγραμμένα χαρακτηριστικά ακινήτων, ορισμένες έρευνες έχουν εστιάσει στην εναλλακτική της ανάλυσης εικόνας για την πρόβλεψη τιμών ακινήτων. Για παράδειγμα, το 2017, ο Quanzeng You και οι συνεργάτες του παρατήρησαν πως η χρήση ενός Recurrent Neural Network για αναγνώριση εικόνων απέδιδε καλύτερα από βασικούς αλγόριθμους για την πρόβλεψη τιμών ακινήτων [5]. Την ίδια χρονιά, οι Koziarski Michal και Cyganek Boguslaw παρουσίασαν τα αποτελέσματα της έρευνάς τους για ανάλυση εικόνων με την παρουσία θορύβου. Τα αποτελέσματά τους μπορούν να χρησιμοποιηθούν και στην περίπτωση των ακινήτων όπου ο θόρυβος αποτελεί συχνό φαινόμενο [6]. Το 2018, οι Omid Poursaeed, Tomas Matera και Serge Belongie χρησιμοποίησαν Recurrent Neural Netwroks για την πρόβλεψη του επιπέδου πολυτέλειας ενός ακινήτου [7]. Επιπλέον, οι Edward L. Glaeser, Michael Scott Kincaid και Nikhil Naik παρατήρησαν το 2018 βελτίωση της πρόβλεψης τιμής ακινήτων χρησιμοποιώντας ανάλυση εικόνας και πως συμβάντα όπως κατάσχεση από μία τράπεζα επηρέαζαν αρνητικά την εμφάνιση ενός ακινήτου με το χρόνο [8]. Τέλος, οι Lotfi A. Zadeh, Saied Tadayon και Bijan Tadayon διατύπωσαν το 2018 ένα σύνολο μεθόδων με τις οποίες μπορεί να γίνει αποδοτικά και εύστοχα αναγνώριση εικόνας, κάτι που μπορεί να χρησιμοποιηθεί και στην περίπτωση της αγοράς ακινήτων [9].

Μία ακόμα προσέγγιση είναι αυτή της επεξεργασίας φυσικής γλώσσας για εξαγωγή χαρακτηριστικών από δεδομένα αγγελιών. Το 2014, ο D. Stevens καταγράφει τα αποτελέσματα μιας ενθαρρυντικής έρευνας για την πρόβλεψη τιμών ακινήτων (τόσο κατηγοριοποίηση όσο και παλινδρόμηση) χρησιμοποιώντας εξόρυξη κειμένου [10]. Το 2016, ο Moloud Shahbazi και οι συνεργάτες του παρουσίασαν μεθόδους κατάταξης ακινήτων βάσει επενδυσιμότητας αντλώντας πληροφορίες από γραπτό κείμενο [11]. Χρησιμοποίησαν τόσο επιβλεπόμενη όσο και μη επιβλεπόμενη μάθηση. Το ίδιο έτος οι Sherief Abdallah και Deena Abu Khashan παρουσίασαν

μια παρόμοια προσέγγιση η οποία προσέθετε στα αρχικά δεδομένα και την περιγραφή [12]. Τα δεδομένα τους συλλέχθηκαν από ιστότοπους αγγελιών. Το 2017, ο Giannis Bekoulis και οι συνεργάτες του παρουσίασαν τα αποτελέσματα μιας εργασίας στην οποία επανυπολόγιζαν τα χαρακτηριστικά ενός ακινήτου από κείμενα αγγελιών [13]. Τέλος, οι Lily Shen και Στεπηεν Ροσς, παρουσίασαν το 2019 μια προσέγγιση ανάλυσης της απαλής (soft) πληροφορίας σε δεδομένα κειμένου ακινήτων και πώς αυτά επηρεάζουν τις προοπτικές πώλησης [14].

Τέλος, ένα σημαντικό κομμάτι της εργασίας αποτελεί το Web Scraping το οποίο περιγράφεται πιο λεπτομερώς στο Κεφάλαιο 2. Ο Daniel Glez-Pena και οι συνεργάτες του παρουσίασαν το 2014 εναλλακτικές web scraping για περιπτώσεις στις οποίες δεν υπάρχει πρόσβαση σε κάποιο API [15]. Το 2015, ο De. S. Sirisuriya παρουσίασε μία συγκριτική μελέτη των διαφορετικών μεθόδων web scraping. Επιπλέον, το 2018 οι Vlad Krotov και Leiser Silva παρουσίασαν μία μελέτη σχετικά με τις νομικές και ηθικές διαστάσεις του web scraping [16]. Η προσέγγιση αυτή είναι ιδιαίτερα σημαντική στον τομέα του web scraping καθώς δεν αρκεί να συλλέξει κανείς τις πληροφορίες που τον ενδιαφέρουν αλλά και να φροντίσει ώστε να το κάνει χωρίς να επιβαρύνει ή να βλάψει την πηγή από την οποία τις συλλέγει [17].

Κεφάλαιο 2

Πρόσβαση σε δεδομένα αγοραπωλησίας ακινήτων

2.1 Αγορά ακινήτων

2.1.1 Σημασία της αγοράς ακινήτων

Η ακίνητη περιουσία είναι ένας από τους πιο σημαντικούς τομείς της οικονομίας με σημαντική συμβολή στο ΑΕΠ. Οι στεγαστικές αγορές είναι ιδιαίτερα σημαντικές για την συνολική οικονομία. Η προσιτή τιμή της στέγασης και οι τιμές ενοικίασης επηρεάζουν άμεσα τον πλούτο των ιδιοκτητών και των ενοικιαστών, επηρεάζοντας τις καταναλωτικές δαπάνες. Ως εκ τούτου, η ανάπτυξη των αγορών κατοικιών δεν παρακολουθείται στενά μόνο από ιδιωτικά νοικοκυριά, εμπορικές τράπεζες και θεσμικούς επενδυτές, αλλά και από τις κεντρικές τράπεζες και τις κυβερνήσεις για αποφάσεις νομισματικής και δημοσιονομικής πολιτικής.

Η κατάρρευση των Lehman Brothers και Bear Sterns το 2008 έδειξε πόσο σοβαρή συνέπειες που μπορεί να έχει η επιδείνωση των αγορών κατοικιών στα χρηματοπιστωτικά ιδρύματα. Πέρα από δανεισμό σημαντικών ποσών σε επιχειρήσεις ακινήτων οι τράπεζες βασίζονται σε μεγάλο βαθμό στην ακίνητη περιουσία ως ασφάλεια και ένα υποκείμενο περιουσιακό στοιχείο. Αν και η διεθνής οικονομία και οι αγορές ακινήτων είναι στενά διασυνδεδεμένες με πολλούς τρόπους, οι αγορές κατοικιών εξακολουθούν να διαθέτουν ιδιοσυγκρατικές ιδιότητες.

Η εξέλιξη των τιμών στην αγορά κατοικίας παρουσιάζει μεγάλο ενδιαφέρον για τους ιδιοκτήτες κατοικιών και τους επενδυτές ακινήτων. Ο πλούτος τους επηρεάζεται άμεσα από τις αλλαγές στις τιμές των κατοικιών. Ως εκ τούτου, η διερεύνηση των θεμελιωδών παραγόντων που έχουν την ικανότητα να εξηγούν τις τιμές των κατοικιών είναι ζωτικής σημασίας για όλους.

Η τιμή ενός ακινήτου επηρεάζεται από πλήθος παραγόντων οι οποίοι αφορούν τόσο χαρακτηριστικά του ίδιου του ακινήτου όπως μέγεθος, αριθμός και είδος δωματίων και υλικά κατασκευής όσο και χαρακτηριστικά της τοποθεσίας του - εύκολη πρόσβαση σε συγκοινωνίες, θόρυβος, ποιότητα κοντινών σχολείων, κλπ. Συνεπώς, μια μελέτη του τρόπου με τον οποίο κάθε χαρακτηριστικό επηρεάζει την τιμή ενός ακινήτου μπορεί να επιφέρει μεγάλα οφέλη.

2.1.2 Η κατάσταση στον χώρο των δεδομένων ακινήτων

Τα δεδομένα αγοραπωλησίας ακινήτων αποτελούν παγκοσμίως ένα πολύτιμο περιουσιακό στοιχείο το οποίο μπορεί να αξιοποιηθεί για την παραγωγή σημαντικού πλούτου. Η αγορά ενός ακινήτου σε τιμή χαμηλότερη της πραγματικής ή η πώληση σε τιμή υψηλότερη της πραγματικής μπορούν να αποφέρουν σημαντικά κέρδη, ακόμα και αν η διαφορά είναι μικρή, εφόσον επενδυθεί επαρκές κεφάλαιο.

Αυτό έχει ως αποτέλεσμα οι εταιρίες πίσω από ιστότοπους όπως το Zillow στην Νέα Υόρκη ή ο Spitogatos στην Ελλάδα να θεωρούν τα δεδομένα αυτά σημαντικό κομμάτι των επιχειρήσεών τους και συνεπώς, να μην επιτρέπουν την ελεύθερη πρόσβαση στα δεδομένα αυτά. Προκειμένου να αποκτηθεί επομένως πρόσβαση στα δεδομένα αυτά, απαιτούνται σύνθετες ενέργειες, οι οποίες παρουσιάζονται σταδιακά παρακάτω.

2.2 Web Scraping

2.2.1 Εισαγωγή στο Web Scraping

Το web scraping είναι η διαδικασία συλλογής δομημένων δεδομένων ιστού με αυτοματοποιημένο τρόπο. Ονομάζεται επίσης εξαγωγή δεδομένων ιστού. Μερικές από τις κύριες περιπτώσεις χρήσης του web scraping περιλαμβάνουν παρακολούθηση τιμών, παρακολούθηση ειδήσεων και έρευνα αγοράς μεταξύ πολλών άλλων.

Σε γενικές γραμμές, η εξαγωγή δεδομένων ιστού χρησιμοποιείται από άτομα και επιχειρήσεις που θέλουν να κάνουν χρήση του τεράστιου όγκου διαθέσιμων στο κοινό δεδομένων ιστού για τη λήψη πιο έξυπνων αποφάσεων.

Σε αντίθεση με τη συνηθισμένη, χειροκίνητη διαδικασία εξαγωγής δεδομένων με μη αυτόματο τρόπο, το web scraping χρησιμοποιεί έξυπνο αυτοματισμό για να ανακτήσει εκατοντάδες, εκατομμύρια ή ακόμα και δισεκατομμύρια στοιχεία δεδομένων από το Διαδίκτυο.

Οι web scrapers λειτουργούν με έναν φαινομενικά απλό τρόπο. Καταρχάς, στον web scraper θα δοθούν μία ή περισσότερες διευθύνσεις URL για φόρτωση πριν από τη δημιουργία. Στη συνέχεια, αυτός φορτώνει ολόκληρο τον κώδικα HTML για την εν λόγω σελίδα. Οι πιο προηγμένοι web scrapers θα φορτώσουν ολόκληρο τον ιστότοπο, συμπεριλαμβανομένων στοιχείων CSS και Javascript.

Στη συνέχεια, ο scraper θα εξάγει όλα τα δεδομένα στη σελίδα ή συγκεκριμένα δεδομένα που θα επιλέξει ο χρήστης πριν από την εκτέλεση της εργασίας. Στην ιδανική περίπτωση, ο χρήστης θα περάσει από τη διαδικασία επιλογής των συγκεκριμένων δεδομένων που θέλει από τη σελίδα.

Τέλος, ο web scraper θα εξάγει όλα τα δεδομένα που έχουν συλλεχθεί σε μορφή που είναι πιο χρήσιμη για τον χρήστη.

Ο λόγος που χρησιμοποείται web scraping στα πλαίσια της εργασίας είναι ότι τα δεδομένα που αφορούν τα χαρακτηριστικά και τις τιμές ακινήτων δεν βρίσκονται έτοιμα σε δομημένη μορφή στην οποία να έχει οποιοσδήποτε πρόσβαση.

Επειδή το web scraping μπορεί να είναι επιβαρυντικό για τον ιστότοπο στον οποίο δρα (λόγω του πλήθους των αιτημάτων που μπορεί να στείλει σε μικρό χρονικό διάστημα), είναι σημαντικό να ρυθμιστεί έτσι ώστε να μην παρεμποδίσει την ομαλή λειτουργία του ιστότοπου όσο λειτουργεί.

2.2.2 Μέθοδοι Web Scraping

Θα παρουσιαστούν 4 άξονες που διαφοροποιούν τους web scrapers. Φυσικά, υπάρχουν περισσότερες ακόμα περισσότερες δυνατότητες διαχωρισμού αλλά εδώ παρουσιάζονται οι 4 αυτές βασικές.

- **Αυτο-κατασκευασμένο ή Προ-κατασκευασμένο:** Η δημιουργία από το μηδέν ενός web scraper απαιτεί κάποιες προηγμένες γνώσεις προγραμματισμού. Το εύρος αυτής της γνώσης αυξάνεται επίσης με τον αριθμό των δυνατοτήτων που είναι επιθυμητές να έχει ο web scraper. Για τον λόγο αυτό υπάρχει πλήθος από προκατασκευασμένες λύσεις αλλά και πλατφόρμες που προσφέρουν έτοιμα δομικά στοιχεία για την δημιουργία ενός web scraper. Οι περισσότερες από τις πλατφόρμες αυτές προσφέρουν και επιπλέον υπηρεσίες όπως χρήση proxies για την δημιουργία αιτημάτων από τους scrapers.

- **Επέκταση προγράμματος περιήγησης ή λογισμικό:** Σε γενικές γραμμές, οι web scrapers διατίθενται σε δύο μορφές: Επεκτάσεις προγράμματος περιήγησης ή λογισμικό υπολογιστή. Οι επεκτάσεις προγράμματος περιήγησης είναι προγράμματα που μοιάζουν με εφαρμογές και μπορούν να προστεθούν σε ένα πρόγραμμα περιήγησης. Οι επεκτάσεις προγράμματος περιήγησης έχουν το πλεονέκτημα της απλούστερης εκτέλεσης και της ενσωμάτωσης απευθείας στο πρόγραμμα περιήγησης. Ωστόσο, αυτές οι επεκτάσεις περιορίζονται συνήθως από τη διαμονή τους σε αυτό. Αυτό σημαίνει ότι κάθε προηγμένη δυνατότητα που θα έπρεπε να εμφανιστεί εκτός αυτού θα ήταν αδύνατο να εφαρμοστεί. Για παράδειγμα, οι περιστροφές IP δεν θα ήταν δυνατές. Από την άλλη, το λογισμικό υπολογιστή είναι λιγότερο βολικό από τις επεκτάσεις του προγράμματος περιήγησης αλλά προσφέρει συνήθως περισσότερο προηγμένες λειτουργίες που δεν περιορίζονται από αυτό που μπορεί και δεν μπορεί να κάνει ένα πρόγραμμα περιήγησης.

- **Διεπαφή χρήστη:** Η διεπαφή χρήστη μεταξύ των web scrapers μπορεί να ποικίλει πολύ. Για παράδειγμα, ορισμένα εργαλεία web scraping εκτελούνται με ένα ελάχιστο περιβάλλον εργασίας χρήστη και μια γραμμή εντολών. Από την άλλη πλευρά, ορισμένοι web scrapers έχουν ένα πλήρες περιβάλλον εργασίας χρήστη όπου ο ιστότοπος είναι ορατός ώστε ο χρήστης να επιλέγει εύκολα ποια δεδομένα θέλει να διαγράψει και ποια να κρατήσει. Αυτοί οι web scrapers είναι συνήθως πιο εύκολο να χρησιμοποιηθούν για τα περισσότερα άτομα με περιορισμένες τεχνικές γνώσεις.

- **Στο Cloud ή τοπικό:** Σε περίπτωση που οι πόροι ενός προσωπικού υπολογιστή ή ενός απλού εταιρικού συστήματος δεν αρκούν για την ικανοποιητική απόδοση ενός web scraper, μία συνήθης λύση είναι η μεταφορά του στο Cloud. Η μεταφορά στο Cloud δίνει τη δυνατότητα δυναμικής κατανομής πόρων ανάλογα με τις ανάγκες του scraper.

Λύνει επίσης προβλήματα όπως η δυναμική απόδοση IP καθώς συνήθως οι υπηρεσίες που προσφέρουν web scraping υπηρεσίες βασισμένες στο Cloud προσφέρουν και τη δυνατότητα εναλλαγής IP σε κάθε κλήση του web scraper. Με τον τρόπο αυτό αποφεύγεται το μπλοκάρισμά του από την σελίδα την οποία επισκέπτεται. Συνήθως οι Cloud πλατφόρμες οι οποίες προσφέρουν δυνατότητες φιλοξενίας web scrapers παρέχουν και δικές τους αρχιτεκτονικές για την κατασκευή τους και τον έλεγχο της ομαλής λειτουργίας τους.

2.2.3 Apify

Το Apify είναι μια ενιαία πλατφόρμα για έργα Web Scraping, εξαγωγής δεδομένων και αυτοματισμού ρομποτικής διαδικασίας (RPA), βασισμένη στο Cloud.

Κατά τη διάρκεια συγγραφής της παρούσας εργασίας, διαθέτει τόσο μία ελεύθερη βαθμίδα που προσφέρει ένα συγκεκριμένο αριθμό πόρων κάθε μήνα όσο και διάφορες επί πληρωμή βαθμίδες οι οποίες προσφέρουν μεγαλύτερο αριθμό πόρων και δυνατοτήτων.

Βασικό συστατικό της πλατφόρμας στο οποίο βασίζεται και το Web Scraping είναι οι Actors. Οι Actors είναι προγράμματα Cloud χωρίς διακομιστές που εκτελούνται στην πλατφόρμα Apify και μπορούν να εκτελέσουν αυθαίρετες εργασίες υπολογιστών, όπως αποστολή email ή ανίχνευση ιστότοπων με εκατομμύρια σελίδες. Μπορούν να ξεκινήσουν χειροκίνητα, χρησιμοποιώντας το API της πλατφόρμας ή ένα χρονοδιάγραμμα και μπορούν να ενσωματωθούν σε άλλες εφαρμογές.

Η πλατφόρμα προσφέρει έτοιμες λύσεις αλλά και τη δυνατότητα δημιουργίας custom λύσεων βασισμένων στο API της, με χρήση Javascript.

Με βάση τους 4 άξονες διαφοροποίησης των web scrapers που παρουσιάστηκαν παραπάνω το Apify κατατάσσεται ως εξής:

- Το Apify υποστηρίζει τόσο Αυτο-κατασκευασμένους όσο και Προ-κατασκευασμένους web scrapers. Ένας χρήστης μπορεί είτε να φτιάξει έναν δικό του είτε να χρησιμοποιήσει έναν από τους ήδη έτοιμους που προσφέρονται στην πλατφόρμα.

- Οι scrapers αποτελούν ξεχωριστό κομμάτι λογισμικού και δεν λειτουργούν μέσω προγράμματος περιήγησης. Έχουν έτσι μεγαλύτερη δυνατότητα εξατομίκευσης και εξειδίκευσης.

- Το User Interface προσφέρει ζωντανή παρακολούθηση της εργασίας του scraper, των πόρων που καταναλώνει και των δεδομένων που συλλέγει.

- Η πλατφόρμα βρίσκεται εξ΄ ολοκλήρου στο Cloud με δυνατότητα τοπικής αποθήκευσης των δεδομένων. Επιπλέον, σε περίπτωση που το επιθυμεί, ο χρήστης μπορεί να κατεβάσει τοπικά και να τρέξει ορισμένα είδη scrapers.

Ο κύριος λόγος που επιλέχθηκε η χρήση της είναι η δυνατότητα αξιοποίησης της υποδομής της ώστε να μην χρειαστεί ο σχεδιασμός ενός web scraper από το μηδέν. Επιπλέον

πλεονέκτημα που προσέρει είναι η δυνατότητα χρήσης περιστρεφόμενων Proxies με σκοπό την αποφυγή μπλοκαρίσματος του scraper από κάποιον ιστότοπο.

Εναλλακτικές επιλογές που εξετάστηκαν είναι τα:

- Scrapy Framework: Ένα ανοιχτού κώδικα framework για την εξαγωγή των δεδομένων από ιστότοπους. Είναι γραμμένο στη γλώσσα Python και χαρακτηρίζεται από ταχύτητα, την απλότητα αλλά και την επεκτασιμότητά του.

- Σελενιυμ: Πρόκειται για ένα ανοιχτού κώδικα, αυτοματοποιημένο testing framework που χρησιμοποιείται για την επικύρωση εφαρμογών ιστού σε διαφορετικά προγράμματα περιήγησης και πλατφόρμες. Χρησιμοποιείται με πολλές γλώσσες προγραμματισμού όπως Java, C, Python, κ.λπ. Οι δοκιμές που γίνονται χρησιμοποιώντας το εργαλείο δοκιμής Selenium αναφέρονται συνήθως ως Selenium Testing και καθώς αυτοματοποιούν ενέργειες ενός περιηγητή ιστού, μπορούν να χρησιμοποιηθούν και για σκοπούς web scraping.

Εν τέλει, επιλέχθηκε η πλατφόρμα Apify για τους λόγους που αναφέρθηκαν παραπάνω.

2.3 XPath

Το XPath (XML Path Language) είναι μια γλώσσα ερωτημάτων για την επιλογή στοιχείων από ένα έγγραφο XML. Επιπλέον, το XPath μπορεί να χρησιμοποιηθεί για τον υπολογισμό τιμών (π.χ. συμβολοσειρές, αριθμούς ή τιμές Boolean) από το περιεχόμενο ενός εγγράφου XML. Το XPath καθορίστηκε από το World Wide Web Consortium (W3C).

Το XPath διαθέτει επίσης ένα μεγάλο αριθμό από συναρτήσεις οι οποίες επιτρέπουν την πιο εύκολη πρόσβαση και επεξεργασία δεδομένων από δομές XML.

Ένα παράδειγμα XPath αποτελεί η εξής έκφραση η οποία ξεκινάει από τη ρίζα ενός XML αρχείου και βρίσκει το όνομα κάθε πρότζεκτ μέσα σε μια λίστα με όλα τα πρότζεκτς:

```
//projects/project/@name
```

2.4 Public Facing API

2.4.1 Ορισμός API

Ένα API (Application Programming Interface) είναι ένα σύνολο λειτουργιών που επιτρέπει στις εφαρμογές να έχουν πρόσβαση σε δεδομένα και να αλληλεπιδρούν με εξωτερικά στοιχεία λογισμικού, λειτουργικά συστήματα ή μικροσυσκευές.

Για απλοποίηση, ένα API παρέχει το αίτημα ενός χρήστη σε ένα σύστημα και στέλνει την απόκριση του συστήματος πίσω στον χρήστη. Σκοπός του είναι η παροχή ενός καλά τεκμηριωμένου και δομημένου τρόπου ώστε να κάνει διαθέσιμα τα δεδομένα της μια εφαρμογή στον έξω κόσμο. Συνεπώς, ένα API παρέχει ένα σύνολο τελικών σημείων (endpoints) τα

οποία μπορεί ένας χρήστης ή μια άλλη εφαρμογή να χρησιμοποιήσει ώστε να πάρει πληροφορίες από το σύστημα το οποίο βρίσκεται πίσω από το API. Οι κλήσεις και αποκρίσεις προς και από ένα API γίνονται συνήθως μέσω JSON ή XML φορτίου ώστε να μπορούν να επεξεργαστούν με ευκολία.

2.4.2 Ορισμός REST

Το REST, ή Representational State Transfer, είναι ένα αρχιτεκτονικό στυλ για την παροχή προτύπων μεταξύ συστημάτων υπολογιστών στο διαδίκτυο, το οποίο διευκολύνει την επικοινωνία μεταξύ συστημάτων. Συστήματα συμβατά με REST, συχνά αποκαλούμενα συστήματα RESTful, χαρακτηρίζονται από το πώς είναι αγνωστικά όσον αφορά την κατάσταση (stateless) και διαχωρίζουν την κατάσταση του πελάτη και του διακομιστή.

Στο αρχιτεκτονικό στυλ REST, η υλοποίηση του πελάτη και η υλοποίηση του διακομιστή μπορούν να γίνουν ανεξάρτητα χωρίς να γνωρίζουν ο ένας για τον άλλο. Αυτό σημαίνει ότι ο κωδικός από την πλευρά του πελάτη μπορεί να αλλάξει ανά πάσα στιγμή χωρίς να επηρεάζεται η λειτουργία του διακομιστή και ο κώδικας από την πλευρά του διακομιστή μπορεί να αλλάξει χωρίς να επηρεάζεται η λειτουργία του πελάτη.

Εφόσον κάθε πλευρά γνωρίζει ποια μορφή μηνυμάτων θα στείλει στην άλλη, μπορεί η λειτουργία καθεμίας να διατηρηθεί ξεχωριστή. Διαχωρίζοντας την διεπαφή χρήστη από τα θέματα αποθήκευσης δεδομένων, βελτιώνεται η ευελιξία της διεπαφής σε όλες τις πλατφόρμες και βελτιώνεται η επεκτασιμότητα απλοποιώντας τα στοιχεία του διακομιστή. Επιπλέον, ο διαχωρισμός επιτρέπει σε κάθε συστατικό τη δυνατότητα να εξελιχθεί ανεξάρτητα.

Χρησιμοποιώντας μια διεπαφή REST, διαφορετικοί πελάτες καλούν τα ίδια τελικά σημεία REST, εκτελούν τις ίδιες ενέργειες και λαμβάνουν τις ίδιες απαντήσεις.

Στα συστήματα που ακολουθούν το πρότυπο REST ο διακομιστής δεν χρειάζεται να γνωρίζει τίποτα σχετικά με την κατάσταση στην οποία βρίσκεται ο πελάτης και το αντίστροφο. Με αυτόν τον τρόπο, τόσο ο διακομιστής όσο και ο πελάτης μπορούν να κατανοήσουν οποιοδήποτε μήνυμα λαμβάνεται, ακόμη και χωρίς να δει προηγούμενα μηνύματα. Αυτή η αγνωστικότητα κατάστασης (statelessness επιβάλλεται μέσω της χρήσης πόρων (resources) και όχι με εντολές. Οι πόροι είναι τα ουσιαστικά του Ιστού - περιγράφουν οποιοδήποτε αντικείμενο, έγγραφο ή πράγμα που μπορεί να αποθηκευτεί ή να σταλεί σε άλλες υπηρεσίες.

Αυτοί οι περιορισμοί βοηθούν τις εφαρμογές RESTful να επιτύχουν αξιοπιστία, γρήγορη απόδοση και επεκτασιμότητα, ως στοιχεία που μπορούν να διαχειριστούν, να ενημερωθούν και να επαναχρησιμοποιηθούν χωρίς να επηρεαστεί το σύστημα στο σύνολό του, ακόμη και κατά τη λειτουργία του συστήματος.

2.4.3 Κλήσεις σε REST endpoints

Στην αρχιτεκτονική REST, οι πελάτες στέλνουν αιτήματα για ανάκτηση ή τροποποίηση πόρων και οι διακομιστές αποστέλλουν απαντήσεις σε αυτά τα αιτήματα.

Ένα αίτημα αποτελείται γενικά από:

- Ένα ρήμα HTTP, το οποίο καθορίζει τι είδους λειτουργία θα εκτελέσει.

- Μια κεφαλίδα (header), η οποία επιτρέπει στον πελάτη να μεταφέρει πληροφορίες σχετικά με το αίτημα.
- Μια διαδρομή (path) προς έναν πόρο.
- Ένα προαιρετικό σώμα μηνυμάτων που περιέχει δεδομένα.

Υπάρχουν 4 βασικά ρήματα HTTP που χρησιμοποιούνται σε αιτήματα σε ένα σύστημα REST:

- GET - ανάκτηση ενός συγκεκριμένου πόρου (ανά αναγνωριστικό - id) ή μιας συλλογής πόρων.
- POST - δημιουργία ενός νέου πόρου.
- PUT - ενημέρωση συγκεκριμένου πόρου (ανά αναγνωριστικό - id).
- DELETE - αφαίρεση ενός συγκεκριμένου πόρου με αναγνωριστικό - id.

2.4.4 Ορισμός Public Facing API

Ένα δημοσίως προσβάσιμο API είναι ένα API στο οποίο μπορεί κανείς να έχει πρόσβαση από το δημόσιο διαδίκτυο. Αυτό συμβαίνει σε αντίθεση με τα endpoints ενός API ιδιωτικού δικτύου στο οποίο έχει πρόσβαση μόνο όποιος βρίσκεται μέσα στο συγκεκριμένο δίκτυο. Αυτό δεν σημαίνει πως οποισδήποτε μπορεί να το χρησιμοποιήσει. Τα περισσότερα APIs κάνουν χρήση ηλεκτρονικών κλειδιών ή περιορισμού των κλήσεων ανά χρονική μονάδα προκειμένου να αποθαρρύνουν την επιβλαβή χρήση τους (spam).

Παράδειγμα δημοσίως προσβάσιμου API είναι το https://api.chucknorris.io/ το οποίο δέχεται REST κλήσεις τύπου GET στον σύνδεσμο https://api.chucknorris.io/jokes/random και επιστρέφει μία απάντηση που περιέχει ένα ανέκδοτο.

Κεφάλαιο 3

Θεωρητικό υπόβαθρο

3.1 Μηχανική Μάθηση

3.1.1 Εισαγωγή στη Μηχανική Μάθηση

Η Μηχανική Μάθηση είναι ένα πεδίο της Επιστήμης Υπολογιστών στο οποίο μελετώνται αλγόριθμοι, οι οποίοι επιτρέπουν σε ένα υπολογιστικό σύστημα να πραγματοποιεί προβλέψεις ή να λαμβάνει αποφάσεις χωρίς να έχει προγραμματιστεί εξαρχής η ακριβής συλλογιστική πορεία που πρέπει να ακολουθήσει για να τις λάβει. Συγκεκριμένα, ο αλγόριθμος μηχανικής μάθησης δημιουργεί ένα μαθηματικό μοντέλο που βασίζεται σε δεδομένα - δείγματα, ή αλλιώς δεδομένα εκπαίδευσης με τα οποία τροφοδοτείται. Μοντέλα μηχανικής μάθησης, χρησιμοποιούνται σε διάφορους τομείς, όπως η πρόβλεψη των τιμών μίας μετοχής, η αυτόματη αναγνώριση ηλεκτρονικών μηνυμάτων απάτης, η επεξεργασία φυσικής γλώσσας κ.α. Ένα από τα πεδία αυτά είναι και η πρόβλεψη τιμών ακινήτων.

Η μηχανική μάθηση επιτρέπει την ανάλυση τεράστιων ποσοτήτων δεδομένων. Αν και γενικά παρέχει ταχύτερα, πιο ακριβή αποτελέσματα για τον εντοπισμό κερδοφόρων ευκαιριών ή κινδύνων, μπορεί επίσης να απαιτήσει επιπλέον χρόνο και πόρους για να εκπαιδευτεί σωστά. Ο συνδυασμός της μηχανικής μάθησης με την τεχνητή νοημοσύνη και τις γνωστικές τεχνολογίες μπορεί να την κάνει ακόμη πιο αποτελεσματική στην επεξεργασία μεγάλου όγκου πληροφοριών.

3.1.2 Κατηγορίες Μηχανικής Μάθησης

Οι αλγόριθμοι μηχανικής μάθησης κατηγοριοποιούνται συχνά ως με επίβλεψη ή χωρίς επίβλεψη.

Οι με επίβλεψη (supervised) αλγόριθμοι μηχανικής μάθησης μπορούν να εφαρμόσουν ό,τι έχουν μάθει στο παρελθόν σε νέα δεδομένα χρησιμοποιώντας επισημασμένα παραδείγματα για την πρόβλεψη μελλοντικών γεγονότων. Ξεκινώντας από την ανάλυση ενός γνωστού συνόλου δεδομένων, ο αλγόριθμος εκμάθησης παράγει μια συνάρτηση για να κάνει προβλέψεις σχετικά με τις τιμές εξόδου. Το σύστημα είναι σε θέση να παρέχει προβλέψεις για οποιαδήποτε νέα είσοδο μετά από επαρκή εκπαίδευση. Ο αλγόριθμος εκμάθησης μπορεί επίσης να συγκρίνει την έξοδο του με τη σωστή έξοδο και να βρει σφάλματα προκειμένου να τροποποιήσει ανάλογα το

εσωτερικό του μοντέλο.

Αντίθετα, οι μη επιτηρούμενοι (unsupervised - χωρίς επίβλεψη) αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται όταν οι πληροφορίες που χρησιμοποιούνται για την εκπαίδευση δεν ταξινομούνται ούτε επισημαίνονται με κάποιο τρόπο από τον άνθρωπο. Η μη επιτηρούμενη μάθηση μελετά πώς τα συστήματα μπορούν να συμπεράνουν μια συνάρτηση για να περιγράψουν μια κρυφή δομή από δεδομένα χωρίς ετικέτα. Το σύστημα δεν καταλαβαίνει τη σωστή έξοδο, αλλά διερευνά τα δεδομένα και μπορεί να εξαγάγει συμπεράσματα από σύνολα δεδομένων για να περιγράψει κρυφές δομές από δεδομένα χωρίς ετικέτα.

Οι ημι-επιβλεπόμενοι (semi-supervised) αλγόριθμοι μηχανικής μάθησης εμπίπτουν κάπου μεταξύ της εποπτευόμενης και της μη εποπτευόμενης μάθησης, δεδομένου ότι χρησιμοποιούν τόσο δεδομένα με ετικέτα όσο και χωρίς για την εκπαίδευση - συνήθως μια μικρή ποσότητα δεδομένων με ετικέτες και μια μεγάλη ποσότητα δεδομένων χωρίς. Τα συστήματα που χρησιμοποιούν αυτήν τη μέθοδο είναι σε θέση να βελτιώσουν σημαντικά την ακρίβεια της μάθησης. Συνήθως, η ημι-εποπτευόμενη μάθηση επιλέγεται όταν τα αποκτηθέντα επισημασμένα δεδομένα απαιτούν εξειδικευμένους και σχετικούς πόρους για να οδηγήσουν σε ένα καλό μοντέλο.

Οι αλγόριθμοι ενισχυμένης μάθησης (reinforcement learning) είναι μια μέθοδος μάθησης που αλληλεπιδρά με το περιβάλλον της παράγοντας ενέργειες και ανακαλύπτει σφάλματα ή ανταμοιβές. Η αναζήτηση δοκιμών και σφαλμάτων και η καθυστερημένη ανταμοιβή είναι τα πιο σχετικά χαρακτηριστικά της μάθησης ενίσχυσης. Αυτή η μέθοδος επιτρέπει σε μηχανές και πράκτορες λογισμικού να προσδιορίζουν αυτόματα την ιδανική συμπεριφορά σε ένα συγκεκριμένο πλαίσιο, προκειμένου να μεγιστοποιήσουν την απόδοσή της. Απαιτείται απλή ανατροφοδότηση ανταμοιβής για να μάθει ο πράκτορας ποια ενέργεια είναι καλύτερη. Αυτό είναι γνωστό ως σήμα ενίσχυσης.

Στα πλαίσια των επιβλεπόμενων αλγορίθμων που αφορούν την παρούσα εργασία, ένα άλλο κριτήριο κατηγοριοποίησης των αλγορίθμων Μηχανικής Μάθησης είναι το είδος προβλήματος που αυτοί επιλύουν. Δύο μεγάλες ομάδες προβλημάτων είναι τα προβλήματα ταξινόμησης (classification) και τα προβλήματα παλινδρόμησης (regression).

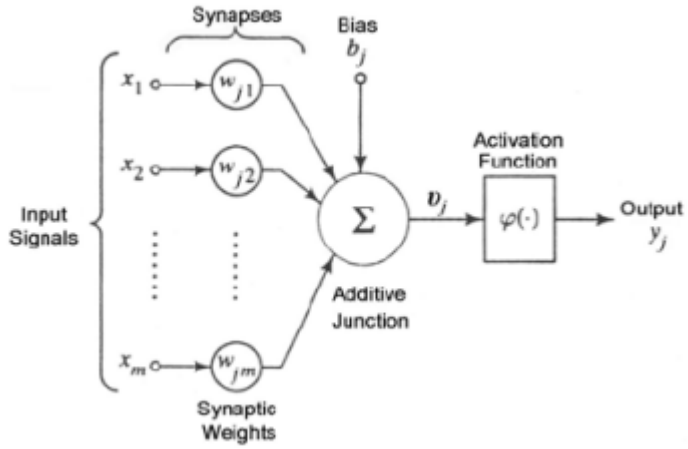
Οι αλγόριθμοι ταξινόμησης επιχειρούν να εκτιμήσουν την διακριτή ομάδα στην οποία ανήκει ένας στόχος ψ με βάση τις μεταβλητές εισόδου χ. Για παράδειγμα, όταν παρέχεται ένα σύνολο δεδομένων για ακίνητα, ένας αλγόριθμος ταξινόμησης μπορεί να προσπαθήσει να προβλέψει εάν οι τιμές για τα σπίτια πωλούν περισσότερο ή λιγότερο από τη συνιστώμενη λιανική τιμή.

Στη μηχανική μάθηση, οι αλγόριθμοι παλινδρόμησης προσπαθούν να εκτιμήσουν τη συνάρτηση χαρτογράφησης (mapping function) φ που οδηγεί από τις μεταβλητές εισόδου ξ σε αριθμητικές ή συνεχείς μεταβλητές εξόδου ψ. Σε αυτήν την περίπτωση, το ψ είναι μια πραγματική τιμή, η οποία μπορεί να είναι ακέραιος ή δεκαδική τιμή. Επομένως, τα προβλήματα πρόβλεψης παλινδρόμησης είναι συνήθως ποσότητες ή μεγέθη. Για παράδειγμα, όταν παρέχεται ένα σύνολο δεδομένων για ακίνητα και ζητείται να προβλεφθούν οι τιμές τους, αυτό είναι μια εργασία παλινδρόμησης επειδή η τιμή θα είναι μια συνεχής έξοδος. Παραδείγματα των κοινών αλγορίθμων παλινδρόμησης περιλαμβάνουν την γραμμική παλινδρόμηση και τα δέντρα παλινδρόμησης. Ορισμένοι αλγόριθμοι, όπως η λογιστική παλινδρόμηση, έχουν τη λέξη παλινδρόμηση στα ονόματά τους, αλλά δεν είναι αλγόριθμοι παλινδρόμησης.

3.1.3 Νευρωνικά Δίκτυα

Νευρώνας - Perceptron

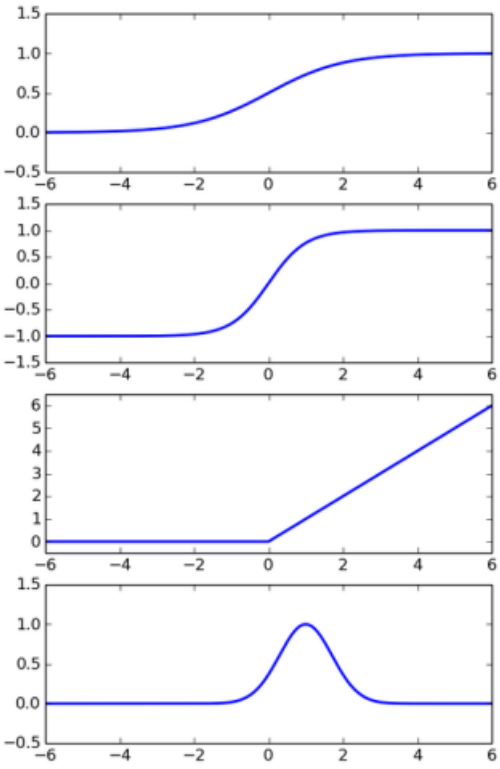
Η βασική μονάδα υπολογισμού σε ένα νευρωνικό δίκτυο είναι ο νευρώνας (Neuron), συχνά ονομαζόμενος και ως κόμβος. Το μοντέλο του Perceptron έχει περιγραφεί από τον Rosenblatt [18]. Λαμβάνει εισόδους από άλλους κόμβους ή από μια εξωτερική πηγή και υπολογίζει μια έξοδο. Κάθε είσοδος πολλαπλασιάζεται με το αντίστοιχο βάρος (Weight) και υπολογίζεται τα ολικό άθροισμα των γινομένων. Ο κόμβος εφαρμόζει μια συνάρτηση ενεργοποίησης σε αυτό το άθροισμα και έτσι προκύπτει η έξοδος του νευρώνα. Παρακάτω φαίνεται η αναπαράσταση ενός νευρώνα, καθώς και η εξίσωση της εξόδου.



Σχήμα 3.1: Δομή Perceptron

$$k = f(\sum_{i=1}^{n}(w_i * x_i + b_i)) \tag{3.1}$$

Το παραπάνω δίκτυο παίρνει ως εισόδους τα $X_1, X_2, ..., X_n$ που έχουν για βάρη τα $W_1, W_2, ..., W_n$ αντίστοιχα. Επιπλέον για κάθε είσοδο υπάρχει ακόμα μια είσοδος με τιμή 1 με βάρος b η οποία ονομάζεται πόλωση (bias). Η συνάρτηση f είναι μη γραμμική και ονομάζεται συνάρτηση ενεργοποίησης (activation function). Ο σκοπός της συνάρτησης ενεργοποίησης είναι να εισάγει μη γραμμικότητα στην έξοδο ενός νευρώνα. Αυτό είναι σημαντικό καθώς σχεδόν όλα τα πραγματικά δεδομένα είναι μη γραμμικά. Παραδείγματα τέτοιων συναρτήσεων είναι η σιγμοειδής συνάρτηση, η υπερβολική εφαπτομένη και η ReLU (Rectified Linear Unit). Η ReLu αποδίδει συχνά καλύτερα από άλλες συναρτήσεις ενεργοποίησης για πολλαπλά επίπεδα τα οποία θα παρουσιαστούν στη συνέχεια. Ο βασικός λόγος της αυξημένης απόδοσης οφείλεται στο γεγονός ότι η ReLU είναι μια γραμμική συνάρτηση μη κορεσμού. Ο κορεσμός είναι το μεγαλύτερο πρόβλημα των δυο προηγούμενων σιγμοειδών συναρτήσεων. Σε αντίθεση λοιπόν με την υπερβολική εφαπτομένη, η ReLU δεν έρχεται σε κορεσμό στο -1, 0 ή 1. Αυτό την καθιστά την συχνότερα εμφανιζόμενη συνάρτηση ενεργοποίησης.



Sigmoid

$$\phi(z) = \frac{1}{1+e^{-z}}$$

Hyperbolic Tangent

$$\phi(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$$

Rectified Linear

$$\phi(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$$

Radial Basis Function

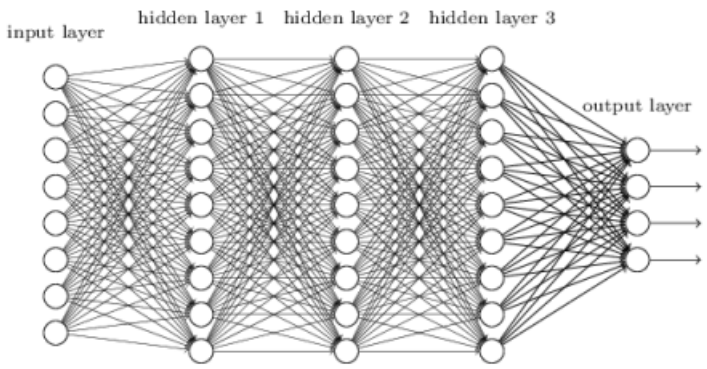
$$\phi(z, c) = e^{-(\epsilon\|z-c\|)^2}$$

Σχήμα 3.2: Συναρτήσεις Ενεργοποίησης

Δομή Νευρωνικού Δικτύου

Για την δημιουργία ενός Νευρωνικού Δικτύου, οι Τεχνητοί Νευρώνες που περιγράφηκαν παραπάνω οργανώνονται σε επίπεδα, όπου το κάθε επίπεδο επεξεργάζεται ένα σύνολο σημάτων. Το πρώτο επίπεδο ονομάζεται επίπεδο εισόδου και χρησιμοποιείται για την εισαγωγή των δεδομένων-χαρακτηριστικών στο δίκτυο. Είναι σημαντικό να αναφερθεί ότι τα στοιχεία του επιπέδου αυτού δεν είναι νευρώνες σαν αυτούς που περιγράφηκαν παραπάνω καθώς δεν εκτελούν κάποιον υπολογισμό. Στην συνέχεια, προστίθενται κρυφά επίπεδα (hidden layers), ένα ή περισσότερα, ενώ στο τέλος υπάρχει το επίπεδο εξόδου.

Οι Νευρώνες σε ένα Νευρωνικό Δίκτυο, μπορούν να χαρακτηριστούν ως μερικώς ή πλήρως συνδεδεμένοι. Έτσι, εάν όλοι οι Νευρώνες ενός επιπέδου συνδέονται με όλους τους υπόλοιπους, τότε χαρακτηρίζονται ως πλήρως συνδεδεμένοι. Σε κάθε άλλη περίπτωση χαρακτηρίζονται ως μερικώς συνδεδεμένοι. Μία τυπική περίπτωση μερικής σύνδεσης, είναι τα Δίκτυα με πρόσθια προώθηση (feedforward). Στα συγκεκριμένα, οι Νευρώνες ενός επιπέδου συνδέονται πλήρως με τους Νευρώνες στο επόμενο επίπεδο, ενώ δεν υπάρχουν συνδέσεις μεταξύ των Νευρώνων ενός επιπέδου και του προηγουμένου στο Νευρωνικό Δίκτυο. Σε αντίθετη περίπτωση το δίκτυο χαρακτηρίζεται ως δίκτυο ανατροφοδότητας (feedback network ή recurrent network), για παράδειγμα εάν υπάρχουν συνδέσεις μεταξύ νευρώνων ενός επιπέδου με το προηγούμενο επίπεδο.



Σχήμα 3.3: Νευρωνικό δίκτυο πολλών επιπέδων (Multi Layer Percepton)

Κανόνας Δέλτα

Ο κανόνας Δέλτα (Delta Rule), αναπτύχθηκε από τους Widrow και Hoff και αποτελεί γενίκευση του αλγορίθμου εκπαίδευσης Percepton. Συγκεκριμένα, είναι και αυτός καθοδηγούμενος από το σφάλμα, αφού προκύπτει από την ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος (mean square error) των διανυσμάτων εκπαίδευσης. Έτσι, δείχνει τελικά πόσο αποκλίνει ένα δίκτυο από την επιθυμητή συνάρτηση. Το μέσο τετραγωνικό σφάλμα στο στοιχειώδες Percepton, για k διανύσματα εκπαίδευσης, μπορεί να υπολογιστεί από τη σχέση:

$$MSE = \frac{1}{k}\sum_{j=0}^{k}(t_j - in_j) \tag{3.2}$$

όπου σαν σήμα εισόδου θεωρείται το $\sum_{i=0}^{k}(w_{ki} * x_i)$ και n ο αριθμός των επιμέρους σημάτων εισόδου του νευρώνα. Θεωρώντας το διάνυσμα των βαρών $(w_1, w_2, ..., w_n)$, μπορεί να οριστεί ο κανόνας Δέλτα που ονομάζεται και κανόνας επικλινούς μεθόδου (gradient decent), ως η αρνητική παράγωγος του μέσου τετραγωνικού σφάλματος, έτσι ώστε το διάνυσμα βαρών να προσεγγίσει σταδιακά το ιδανικό διάνυσμα. Θεωρώντας ότι

$$\Delta w = -\frac{\theta(MSE)}{\theta w_i} \tag{3.3}$$

και

$$\nabla(MSE) = (\frac{\theta(MSE)}{\theta w_i}, ..., \frac{\theta(MSE)}{\theta w_n}) \tag{3.4}$$

προκύπτει ότι η μεταβολή στην τιμή του βάρους w_i για ένα από τα διανύσματα εκπαίδευσης x_i ως

$$\Delta w = w_{i(new} - w_{i(old)} = \alpha * (t - input) * x_i \tag{3.5}$$

όπου input είναι το συνολικό σήμα του νευρώνα, t η επιθυμητή έξοδος, $w_{i(new)}$ και $w_{i(old)}$ η νέα και η παλιά τιμή του βάρους w_i, x_i η είσοδος της οποίας το βάρος αναπροσαρμόζεται και α ο ρυθμός μάθησης (learning rate), που ρυθμίζει το ρυθμό μεταβολής των βαρών. Η

σταθερά α επηρεάζει την ταχύτητα σύγκλισης και καθορίζει την απόδοση του κανόνα Δέλτα. Συγκεκριμένα, πολύ μεγάλες τιμές του α επιταχύνουν τη σύγκλιση στο ελάχιστο σφάλμα, όμως αυξάνουν τον κίνδυνο να προσπεραστεί το ελάχιστο MSE. Αντίθετα, οι μικρές τιμές του α μπορεί να αυξήσουν σημαντικά τον χρόνο εκπαίδευσης.

Επιπλέον, στην παραπάνω σχέση μπορεί αντί του σήματος εισόδου input να χρησιμοποιηθεί η πραγματική έξοδος του Νευρωνικού Δικτύου, λαμβάνοντας έτσι υπόψιν και τη συνάρτηση ενεργοποίησης. Η εκπαίδευση του Δικτύου σταματά όταν το μέσο τετραγωνικό σφάλμα γίνει μικρότερο από κάποια επιθυμητή τιμή. Παρόλο που ο αλγόριθμος Δέλτα, αποτελεί βελτίωση του αλγορίθμου μάθησης του στοιχειωδούς percepton, δεν μπορεί να εφαρμοστεί στα δίκτυα με κρυφά επίπεδα, επειδή δεν είναι γνωστή η επιθυμητή έξοδος t σε κάθε Νευρώνα. Για την επίλυση του συγκεκριμένου προβλήματος χρησιμοποιείται ο αλγόριθμος ανάστροφης μετάδοσης λάθους.

Αλγόριθμος ανάστροφης μετάδοσης λάθους (back propagation)

Για την παραπάνω τοπολογία είναι απαραίτητος ο υπολογισμός διορθώσεων στα βάρη του κάθε Νευρώνα ξεχωριστά. Για τον σκοπό αυτό γίνεται χρήση της ανάστροφης μετάδοσης λάθους (back propagation), η οποία βασίζεται στον γενικευμένο κανόνα Δέλτα (generalized Delta rule). Ο γενικευμένος κανόνας Δέλτα επιτρέπει τον καθορισμό του ποσοστού του συνολικού σφάλματος που αντιστοιχεί στα βάρη του κάθε Νευρώνα ακόμη και αν αυτά ανήκουν σε κρυφά επίπεδα των οποίων η επιθυμητή έξοδος δεν είναι γνωστή.

Για την εκπαίδευση του Multi Layer Percepton (MLP), γίνεται αρχικά ένα πρόσθιο πέρασμα (forward pass). Συγκεκριμένα, εισάγονται στην είσοδο δεδομένα από ένα διάνυσμα εκπαίδευσης και οι νευρώνες στο επίπεδο εισόδου παράγουν ένα αποτέλεσμα το οποίο στην συνέχεια αποτελεί είσοδο στο επόμενο επίπεδο. Η συγκεκριμένη διαδικασία επαναλαμβάνεται διαδοχικά για τα επόμενα κρυφά επίπεδα μέχρι το επίπεδο εξόδου. Έτσι, για n αριθμό νευρώνων του επιπέδου εισόδου, η είσοδος ενός κρυφού νευρώνα j, δίνεται από την σχέση:

$$in_j = \sum_{i=0}^{n} (w_{ij} * x_i) \tag{3.6}$$

όπου w_{ij} το βάρος της σύνδεσης μεταξύ των νευρώνων i, j και x_i το σήμα εισόδου του νευρώνα i. Αντίστοιχα, η έξοδος του συγκεκριμένου νευρώνα θα είναι

$$out_j = f(\sum_{i=0}^{n} (w_{ij} * x_i)) \tag{3.7}$$

όπου f η συνάρτηση ενεργοποίησης του νευρώνα. Η έξοδος αυτή θα προωθηθεί στους νευρώνες του επόμενου επιπέδου.

Είναι σημαντικό να αναφερθεί πως η συνάρτηση ενεργοποίησης για τα δίκτυα που εκπαιδεύονται με ανάστροφη μετάδοση λάθους πρέπει να είναι μη γραμμική, μονότονα αύξουσα και παραγωγίσιμη για όλες τις τιμές εισόδου [19].

Το δίκτυο ξεκινά τους υπολογισμούς όπως και ένα απλό percepton, με τυχαίες τιμές στα βάρη των νευρώνων. Αντίστοιχα, θα υπολογιστεί και το σφάλμα εξόδου για τους Νευρώνες του

επιπέδου εξόδου. Αφού υπολογιστεί το ακριβές σφάλμα στο επίπεδο εξόδου, για το οποίο είναι γνωστό το επιθυμητό αποτέλεσμα, είναι δυνατό να χρησιμοποιηθεί ο γενικευμένος κανόνας Δέλτα, για να προσαρμοστούν κατάλληλα οι τιμές των βαρών του προηγούμενου επιπέδου. Συγκεκριμένα, για k ως το επίπεδο εξόδου, και j το αμέσως προηγούμενο, μπορούμε αρχικά να ορίσουμε την ποσότητα δ_k, σύμφωνα με τον γενικευμένο κανόνα Δέλτα:

$$\delta_k = (t_k - out_k) * f'(in_k) \tag{3.8}$$

και να υπολογιστεί η μεταβολή στα βάρη

$$\Delta w_{jk} = a * \delta_k * out_j \tag{3.9}$$

όπου a ο ρυθμός μάθησης που είναι επιθυμητός. Αντίστοιχα για το κρυφό επίπεδο j:

$$\delta_j = \sum_{k=1}^{m} (w_{jk} * \delta_k) * f'(in_j) \tag{3.10}$$

$$\Delta w_{ik} = a * \delta_j * x_i \tag{3.11}$$

Με τον παραπάνω τρόπο, διαμορφώνονται τα βάρη όλων των κρυφών επιπέδων μέχρι το επίπεδο εισόδου. Αυτή η διαδικασία προσαρμογής των βαρών ονομάζεται ανάστροφο πέρασμα (backward pass) ή ανάστροφη μετάδοση (back propagation). Με τον τρόπο αυτό, ο αλγόριθμος της ανάστροφης λάθους ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα μεταξύ της εξόδου του δικτύου και της επιθυμητής εξόδου, για διανύσματα του συνόλου εκπαίδευσης.

Στην ουσία, με την παραπάνω διαδικασία αναζητείται το ολικό ελάχιστο της συνάρτησης σφάλματος. Η διόρθωση που γίνεται κάθε φορά προσπαθεί να κάνει εκείνες τις αλλαγές που θα μειώσουν το σφάλμα τοπικά. Αυτό είναι πιθανό να δημιουργήσει πρόβλημα, αφού υπάρχει ο κίνδυνος να εγκλωβιστεί το δίκτυο σε τοπικά ελάχιστα. Για να αντιμετωπιστεί το συγκεκριμένο πρόβλημα μπορεί να γίνει αρχικοποίηση των βαρών με διαφορετικούς τρόπους μέχρι να βρεθεί ένας ικανοποιητικός.

Ένα ακόμη σημαντικό ζήτημα που προκύπτει ορισμένες φορές είναι το δίκτυο να παραλύσει (network paralysis). Στην συγκεκριμένη περίπτωση ένα ή περισσότερα βάρη μπορεί να αποκτήσει πολύ υψηλή τιμή, με αποτέλεσμα να μην μεταβάλλεται σημαντικά στις επόμενες επαναλήψεις [20]. Το πρόβλημα αυτό μπορεί να επιλυθεί αυξάνοντας τον ρυθμό μάθησης a.

3.2 Εργαλεία (Frameworks) Μηχανικής Μάθησης

Στη συνέχεια παρουσιάζονται τα εργαλεία και οι δομές (frameworks) που έχουν χρησιμοποιηθεί στα πλαίσια της εργασίας, καθώς και οι αλγόριθμοι που εξετάζονται.

3.2.1 Tensorflow 2

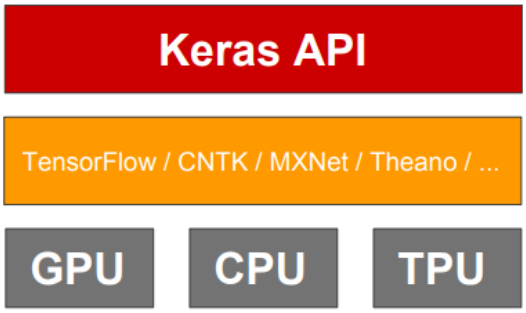
Το TensorFlow 2 είναι μια πλατφόρμα μηχανικής μάθησης ανοιχτού κώδικα. Μπορεί να θεωρηθεί ως επίπεδο υποδομής για διαφορικό προγραμματισμό. Συνδυάζει τις εξής βασικές ικανότητες:

- Εκτελεί αποτελεσματικά λειτουργίες χαμηλού επιπέδου tensor σε CPU, GPU ή TPU.
- Υπολογίζει την κλίση αυθαίρετων διαφορίσιμων εκφράσεων.
- Επιτρέπει κλιμάκωση υπολογισμού σε πολλές συσκευές (π.χ. ο υπερυπολογιστής Summit στο Oak Ridge National Lab, που εκτείνεται σε 27.000 GPU).
- Επιτρέπει εξαγωγή προγραμμάτων (¨γραφήματα¨) σε εξωτερικά συστήματα, όπως διακομιστές, προγράμματα περιήγησης, κινητές συσκευές και ενσωματωμένες συσκευές.

3.2.2 Keras

Το Keras είναι ένα API βαθιάς μάθησης γραμμένο στη γλώσσα Python, που τρέχει πάνω από την πλατφόρμα μηχανικής μάθησης TensorFlow. Αναπτύχθηκε με έμφαση στη δυνατότητα γρήγορου πειραματισμού. Στοχεύει στο να μπορεί κάποιος να πάει από την ιδέα στο αποτέλεσμα όσο το δυνατόν γρηγορότερα.

Το Keras είναι το API υψηλού επιπέδου του TensorFlow 2: Μια προσιτή, παραγωγική διεπαφή για την επίλυση προβλημάτων μηχανικής μάθησης, με έμφαση στη σύγχρονη βαθιά μάθηση. Παρέχει βασικές αφαιρέσεις και δομικά στοιχεία για την ανάπτυξη και αποστολή λύσεων μηχανικής μάθησης με υψηλή ταχύτητα επανάληψης. Δίνει τη δυνατότητα στους μηχανικούς και τους ερευνητές να εκμεταλλευτούν πλήρως τις δυνατότητες κλιμάκωσης και πολλαπλών πλατφορμών του TensorFlow 2.



Σχήμα 3.4: Αρχιτεκτονική Keras [21]

3.2.3 Scikit-learn

Το Scikit-learn (πρώην scikits.learn και επίσης γνωστό ως sklearn) είναι μια δωρεάν βιβλιοθήκη μηχανικής εκμάθησης για τη γλώσσα προγραμματισμού Python. Διαθέτει διάφορους αλγόριθμους ταξινόμησης, παλινδρόμησης και ομαδοποίησης, συμπεριλαμβανομένων μηχανών φορέα υποστήριξης, τυχαίων δασών, ενίσχυσης κλίσης, k-μέσων και DBSCAN, και έχει σχεδιαστεί για να λειτουργεί με τις αριθμητικές και επιστημονικές βιβλιοθήκες Python NumPy και SciPy. Το όνομά του προέρχεται από την ιδέα ότι είναι ένα "SciKit" (SciPy Toolkit), μια ξεχωριστή ανάπτυξη και διανεμημένη επέκταση τρίτου μέρους στο SciPy. Από τα διάφορα scikits, το scikit-learn καθώς και το scikit-image περιγράφηκαν ως ¨καλοδιατηρημένα και δημοφιλή' το Νοέμβριο του 2012. Το Scikit-learn είναι μια από τις πιο δημοφιλείς βιβλιοθήκες μηχανικής μάθησης στο GitHub [22].

3.3 Αλγόριθμοι Μηχανικής Μάθησης

Ακολουθεί μια θεωρητική εισαγωγή στους αλγόριθμους Μηχανικής Μάθησης που χρησιμοποιήθηκαν εκτός του Πολυεπίπεδου Perceptron με Back Propagation που έχει ήδη παρουσιαστεί.

3.3.1 Στοχαστική Κάθοδος Κλίσης - Stochastic Gradient Descent

Ο αλγόριθμος αυτής της κατηγορίας προσπαθεί να δημιουργήσει μία γραμμική συνάρτηση πρόβλεψης τη μορφής $f(x) = w_0+w_1x_1+...+w_nx_n$ για κάθε σύνολο χαρακτηριστικών εισόδου $x1, ..., x_n$. Στη συνέχεια η συνάρτηση $f(x)$ χρησιμοποιείται ως είσοδος σε μία σιγμοειδή συνάρτηση για να προβλεφθεί η τιμή στόχος. Η σιγμοειδής συνάρτηση δίνει πάντα εξόδους στο διάστημα (0,1), οι οποίες αντιπροσωπεύουν τη πιθανότητα να κατηγοριοποιηθεί σε κάποια κλάση η τιμή στόχος. Συμβολίζουμε με $h(f(x))$ τη σιγμοειδή συνάρτηση. Η συνάρτηση κόστους του μοντέλου έχει την εξής μορφή:

$$J = -y * logh - (1 - y) * log(1 - h) \tag{3.12}$$

όπου J η συνάρτηση κόστους για κάθε έξοδο, και y η μία τιμή στόχος. Στην περίπτωση πολλαπλών τιμών στόχων η συνάρτηση κόστους έχει την εξής μορφή:

$$J = -\sum(y_i * \log p_i) \tag{3.13}$$

όπου y διάνυσμα με μονάδα για την κλάση που αντιπροσωπεύει και μηδενικά στα υπόλοιπα στοιχεία του, y_i η τιμή στόχος του διανύσματος y, p_i η πιθανότητα να επιλεγεί η τιμή y_i. Η πιθανότητα p_i δίνεται σε αυτήν την κατηγορία από τη συνάρτηση:

$$p_i = -softmax(f(x_i)) \tag{3.14}$$

$$softmax(z_j) = \frac{e^{z_j}}{\sum_1^K (e^{z_j})}, \forall j \in 1, ..., k \tag{3.15}$$

Όσον αφορά το stochastic gradient descent, με βάση τη σχέση

$$w_{i_{new}} = w_i + \alpha * \frac{\partial J(w_i)}{\partial w_i} \tag{3.16}$$

ενημερώνονται τα βάρη κάθε m εισόδους, όπου m το μέγεθος του batch. Οι τιμές των βαρών μπορεί να ενημερωθούν είτε μετά από τον υπολογισμό των συνολικών απωλειών ενός υποσυνόλου των δεδομένων ή ακόμα και ολόκληρου του δείγματος εισόδου. Ο συντελεστής α θεωρείται υπερπαράμετρος του δικτύου και ονομάζεται ρυθμός εκμάθησης του δείγματος (learning rate). Για να αποφευχθεί η υπερεκπαίδευση του δείγματος, ιδιαίτερα σε μεγάλα δείγματα εκπαίδευσης, προστίθενται στη συνάρτηση κόστους οι όροι $L_1(\beta \sum |w_j|), L_2(\beta \sum w_j^2)$ είτε ένας από αυτούς είτε και οι δύο μαζί. Αφού εκπαιδευτεί το δείγμα με την παραπάνω διαδικασία, στη συνέχεια μπορεί να πραγματοποιήσει πρόβλεψη για κάθε δείγμα εισόδου.

3.3.2 Random Forests

Εισαγωγή

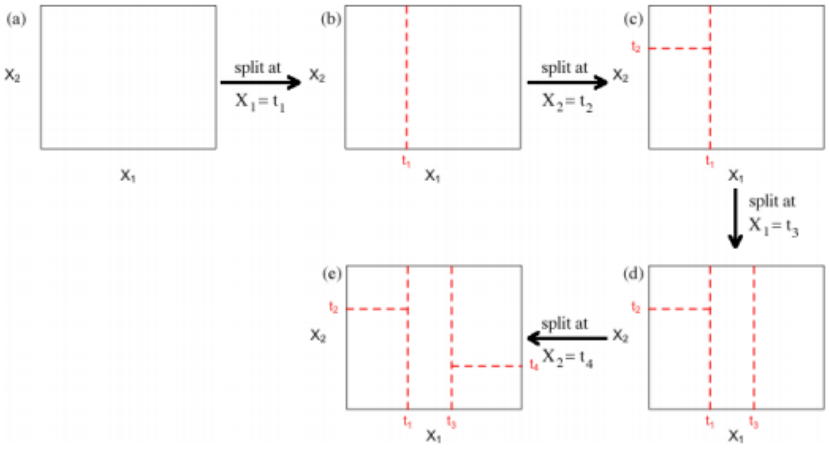
Τα random forests αποτελούν μία μέθοδο μάθησης που μπορεί να χρησιμοποιηθεί τόσο για προβλήματα ταξινόμησης όσο και για προβλήματα παλινδρόμησης. Λειτουργούν με το να κατασκευάζουν ένα πλήθος από δέντρα απόφασης σε δείγμα του συνόλου των δεδομένων κατά τη φάση εκπαίδευσης του μοντέλου και ύστερα να συνυπολογίζουν όλα τα δέντρα για να καθορίσουν την τελική έξοδο. Αποτελούν μία τροποποίηση της μεθόδου bagging, η οποία παίρνει πολλά αμερόληπτα μοντέλα με θόρυβο και βρίσκει την μέση τιμή αυτών, μειώνοντας την διακύμανση της εξόδου. Τα δέντρα απόφασης ιδανικοί υποψήφιοι για μεθόδους bagging διότι μπορούν να συλλάβουν πολύπλοκες αλληλεπιδράσεις μεταξύ των δεδομένων, ενώ αν μεγαλώσουν αρκετά βαθιά, έχουν σχετικά χαμηλή μεροληψία. Επιπρόσθετα, λόγω του θορύβου που έχουν, ο μέσος όρος τους αποτελεί έναν καλό δείκτη της πραγματικής εξόδου. Κάθε δέντρο που παράγεται λέμε πως είναι identically distributed, δηλαδή είναι ανεξάρτητο από τα άλλα δέντρα και παρέχει την ίδια κατανομή πιθανότητας ως προς την τελική έξοδο. Με αυτό τον τρόπο, ο μέσος όρος όλων των δέντρων παρέχει την ίδια μεροληψία με αυτή που παρέχει ένα δέντρο από μόνο του, οπότε πετυχαίνουμε βελτίωση μέσω μείωσης της διακύμανσης.

Αλγόριθμος CART

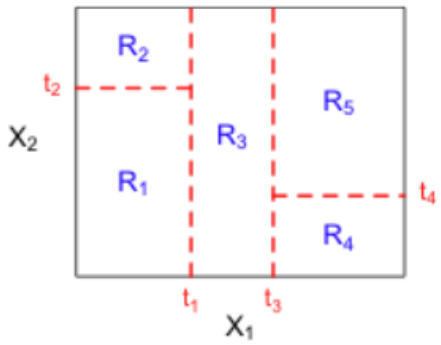
Τα μοντέλα CART (Classification Regression Trees) είναι από τις πιο δημοφιλείς μεθόδους για ταξινόμηση και παλινδρόμηση. Τα μοντέλα αυτά χρησιμοποιούν αναδρομική διχοτόμηση (binary recursive partitioning) ώστε να διαχωρίσουν τα δεδομένα σε υποσύνολα, έτσι ώστε οι καταγραφές εντός των υποσυνόλων να είναι πιο ομοιογενείς μεταξύ τους συγκρινόμενες με το πώς θα ήταν στο ίδιο υποσύνολο [23]. Σε κάθε βήμα της διαδικασίας επιλέγουμε μία συγκεκριμένη μεταβλητή και ένα σημείο διαχωρισμού και ύστερα διαχωρίζουμε τα δεδομένα μας ή ένα σύνολο αυτών σε δύο μέρη. Αυτό επιτυγχάνεται επιλέγοντας ένα σύνολο προς διαίρεση και εξετάζοντας όλες τις πιθανές μεταβλητές και όλα τα πιθανά σημεία διαχωρισμού αυτών των μεταβλητών. Ύστερα επιλέγουμε τον συνδυασμό μεταβλητής - σημείου διαχωρισμού ο οποίος βελτιστοποιεί κάποιο κριτήριο και με βάση τον συνδυασμό αυτό διαχωρίζουμε το σύνολο σε δύο μέρη και επαναλαμβάνουμε την διαδικασία αναδρομικά. Το σύνηθες κριτήριο για ένα δέντρο παλινδρόμησης είναι το υπολειπόμενο άθροισμα τετραγώνων. Έστω για παράδειγμα ένα πρόβλημα παλινδρόμησης με συνεχής απόκριση Y και εισόδους X_1 και X_2. Αρχικά χωρίζουμε το σύνολο σε δύο υποσύνολα στο σημείο $X_1 = t_1$. Ύστερα η περιοχή $X_1 t_1$ χωρίζεται στο $X_2 = t_2$ και η περιοχή $X_1 > t_1$ στο σημείο $X_1 = t_3$. Τέλος, η περιοχή $X_1 > t_3$ χωρίζεται στο σημείο $X_2 = t_4$. Το αποτέλεσμα αυτής είναι ο διαχωρισμός του συνόλου σε πέντε περιοχές $R_1, R_2, ..., R_5$. Η τιμή της απόκρισης είναι η μέση τιμή κάθε μίας εκ των πέντε περιοχών, $y_1, y_2, ..., y_5$. Επομένως σε περιοχή R_m το μοντέλο παλινδρόμησης προβλέπει την έξοδο Y μέσω σταθεράς c_m σύμφωνα με την σχέση:

$$f(x) = \sum_{m=1}^{5} c_m I\{(X_1, X_2) \in R_m\} \tag{3.17}$$

Το I είναι ένας δείκτης που δείχνει αν η παρατήρηση έγκειται σε κάποια δεδομένη ορθογώνια περιοχή. Επειδή μία παρατήρηση μπορεί να ανήκει κάθε φορά σε μόνο μία από τις πέντε περιφέρειες, τέσσερις από τους πέντε όρους του αθροίσματος θα είναι μηδενικοί. Ως αποτέλεσμα παίρνουμε το y της περιοχής που ανήκει το συγκεκριμένο σημείο.



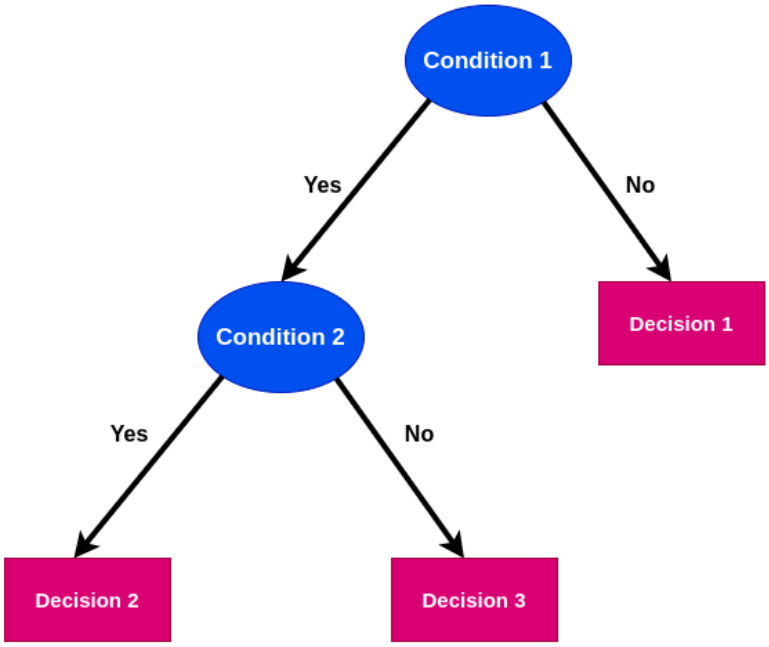
Σχήμα 3.5: Παράδειγμα recursive partitioning με δύο predictors



Σχήμα 3.6: Περιοχές στο επίπεδο X1, X2 μετά από recursive partition

Δέντρα Απόφασης - Decision Trees

Τα δέντρα απόφασης μπορούν να χρησιμοποιηθούν τόσο για την ταξινόμηση όσο και για την παλινδρόμηση. Οι μεθοδολογίες είναι λίγο διαφορετικές, αν και οι αρχές είναι ίδιες. Τα δέντρα αποφάσεων του Scikit Learn χρησιμοποιούν τον αλγόριθμο CART (Classification and Regression Trees). Και στις δύο περιπτώσεις, οι αποφάσεις βασίζονται σε συνθήκες σε οποιοδήποτε από τα χαρακτηριστικά. Οι εσωτερικοί κόμβοι αντιπροσωπεύουν τις συνθήκες και οι κόμβοι φύλλων αντιπροσωπεύουν την απόφαση βάσει των συνθηκών.



Σχήμα 3.7: Decision Tree

Δέντρα Απόφασης Παλινδρόμησης- Regression Decision Trees

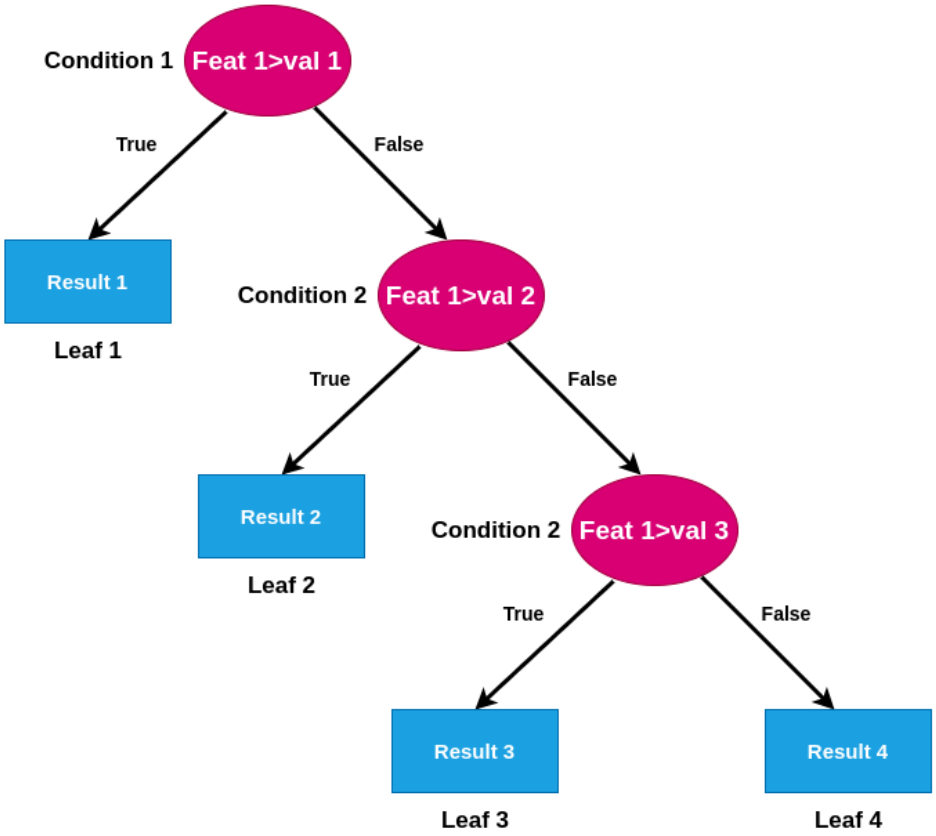
Στα δέντρα παλινδρόμησης, τα φύλλα αντιπροσωπεύουν μια συνεχή αριθμητική τιμή σε αντίθεση με τα δέντρα ταξινόμησης που συνήθως αντιπροσωπεύουν δυαδικές ή διακριτές τιμές στα φύλλα.

Το παραπάνω διάγραμμα αντιπροσωπεύει τη βασική δομή των δέντρων παλινδρόμησης. Σε κάθε επίπεδο κάποιο χαρακτηριστικό της εισόδου συγκρίνεται με μία τιμή την οποία διαμορφώνει ο αλγόριθμος (CART) και ακολουθείτε το αντίστοιχο μονοπάτι μέχρι να καταλήξει σε κάποια απάντηση. Το δέντρο γίνεται πιο περίπλοκο και δύσκολο να αναλυθεί όταν πολλαπλά χαρακτηριστικά ενσωματωθούν και αυξηθεί η διάσταση του συνόλου χαρακτηριστικών.

Τέλος, τα δέντρα αποφάσεων υποφέρουν συχνά από προβλήματα overfitting, προσαρμόζονται δηλαδή υπερβολικά καλά στα δεδομένα στα οποία εκπαιδεύονται με αποτέλεσμα να μην προσφέρουν μεγάλες δυνατότητες γενίκευσης σε νέα δεδομένα.

Random Forests

Όπως αναφέρθηκε ένα δεσισιον τρεε δεν αποτελεί καλό ταξινομητή. Για τον λόγο αυτό, ο αλγόριθμος Random Forest κάνει χρήση πολλών δέντρων μαζί. Κάθε δέντρο μπορεί να κατασκευαστεί σε διαφορετικό υπολογιστή και συνεπώς ο αλγόριθμος είναι πλήρως κατανεμημένος. Κάθε decision tree χρησιμοποιεί συχνά μόνο ένα κομμάτι από τα δεδομένα εκπαίδευσης (συγκεκριμένο μέγεθος που δίνεται από το χρήστη) για την εκπαίδευσή του. Τα δεδομένα εκπαίδευσης όμως δε μοιράζονται στα δέντρα. Κάθε δέντρο παίρνει ένα μέρος τους, αλλά αυτό επιλέγεται τυχαία. Με άλλα λόγια, μέρος των τραινινγ δατα ενός δέντρου, ενδεχομένως να



Σχήμα 3.8: Regression Decision Tree

υπάρχει και σ' άλλο δέντρο. Στη συνέχεια, για την κατασκευή του δέντρου χρησιμοποιείται μόνο ένα μέρος απ' όλα τα χαρακτηριστικά των δεδομένων. Συγκεκριμένα, κατά τη διάρκεια κατασκευής κάθε κόμβου στο δέντρο επιλέγονται τυχαία μερικά χαρακτηριστικά (συγκεκριμένος αριθμός που δίνεται από τον χρήστη) και από αυτά επιλέγεται αυτό που προσφέρει την καλύτερη διαχωρισιμότητα με βάση το υπολογιζόμενο information gain.

Μετά την εκπαίδευση των δέντρων, νέα δεδομένα διατρέχουν όλα τα δέντρα. Κάθε δέντρο δίνει μια πρόβλεψη. Ο μέσος όρος των προβλέψεων είναι και η τελική πρόβλεψη. Αν και είναι δύσκολο στην κατανόηση, η τυχαία επιλογή του training υποσυνόλου που θα χρησιμοποιηθεί για την κατασκευή κάθε δέντρου και η τυχαία επιλογή του υποσυνόλου των χαρακτηριστικών που θα ελεγχθεί σε κάθε κόμβο του δέντρου, έχει αποδειχθεί [24] ότι συμβάλουν στη βελτίωση της ακρίβειας. Οι τεχνικές αυτές ονομάζονται bootstrap aggregating και random subspace method αντίστοιχα και χρησιμοποιούνται συχνά στη μηχανική μάθηση. Παρά το γεγονός ότι χρησιμοποιείται μόνο ένα υποσύνολο των χαρακτηριστικών της εισόδου για εύρεση του καλύτερου χωρίσματος σε κάθε κόμβο, αυτό οδηγεί σε μεγάλη υπολογιστική πολυπλοκότητα, μιας και σε κάθε κόμβο υπολογίζεται το information gain για κάθε τέτοιο χαρακτηριστικό.

Τα Random Forests αποτελούν έναν από τους πιο ισχυρούς αλγόριθμους μηχανικής μάθησης ως σήμερα και χρησιμοποιούνται συχνά τόσο σε προβλήματα ταξινόμησης όσο και παλινδρόμησης.

3.3.3 Ενίσχυση Κλίσης - Gradient Boosting

Η Ενίσχυση αποτελεί μία μέθοδο Μηχανικής Μάθησης και συγκεκριμένα Μάθησης Συνόλου, η οποία χρησιμοποιείται σε προβλήματα ταξινόμησης και παλινδρόμησης. Η προσέγγιση της αφορά τον συνδυασμό μεγάλου αριθμού αδύναμων, σχετικά, μοντέλων, προκειμένου να αποκτήσει μία ισχυρότερη συνολική πρόβλεψη. Συνήθως τα μοντέλα μηχανικής μάθησης, τα οποία συνδυάζονται, είναι τα Τυχαία Δάση και τα Νευρωνικά Δίκτυα και κοινή τεχνική τους αποτελεί η απλή εύρεση του μέσου όρου τους, για να χρησιμοποιηθεί στο συνολικό μοντέλο. Όπως αναφέρουν και οι Alexey Natekin και Alois Knoll [25], κύριο χαρακτηριστικό της ενίσχυσης αποτελεί η προσθήκη νέων μοντέλων στο σύνολο, διαδοχικά. Σε κάθε επανάληψη ένα νέο αδύναμο μοντέλο εκπαιδεύεται, λαμβάνοντας υπόψιν το σφάλμα του μέχρι τώρα συνόλου.

Η Ενίσχυση Κλίσης βασίζεται, σε αντιστοιχία με τα παραπάνω, στην διαδοχική εκπαίδευση νέων ασθενών μοντέλων, που κατά κανόνα είναι πιο αδύναμα από το τελικό σύνολο των μοντέλων, με σκοπό την επίτευξη μίας πιο ακριβούς προσέγγισης της μεταβλητής απόκρισης. Βασική αρχή αυτού του αλγορίθμου αποτελεί η κατασκευή των νέων εκπαιδευόμενων μοντέλων, έτσι ώστε να έχουν μέγιστη συσχέτιση με την αρνητική κλίση της συνάρτησης κόστους του συνόλου, την οποία επιλέγει ο ερευνητής. Συνεπώς η Παλινδρόμηση Ενίσχυσης Κλίσης είναι μία γενίκευση της Ενίσχυσης Κλίσης και αποτελείται από τρία βασικά στοιχεία: την συνάρτηση κόστους προς βελτιστοποίηση, το αδύναμο προς εκπαίδευση μοντέλο και ένα μοντέλο που προσθέτει τα αδύναμα προς εκπαίδευση μοντέλα, στο σύνολο. Τα ασθενή μοντέλα που χρησιμοποιούνται συνήθως είναι τα δένδρα απόφασης, το μέγεθος των οποίων παραμένει σταθερό κατά τη διάρκεια της εκπαίδευσης [26]. Η μαθηματική μορφή που λαμβάνει τελικά το μοντέλο είναι η εξής:

$$F_m(x) = F_{m-1}(x) + \gamma_m * h_m(x) \tag{3.18}$$

όπου:

$$F_m(x) = \sum_{m=1}^{M} \gamma_m * h_m(x) \tag{3.19}$$

Όπου $h_m(x)$ είναι οι ασθενείς συναρτήσεις βάσης των δέντρων απόφασης. Σε κάθε βήμα το δέντρο απόφασης $h_m(x)$ επιλέγεται για την ελαχιστοποίηση της συνάρτησης κόστους, δοσμένου του τρέχοντος μοντέλου και του $F_{m-1}(x_i)$.

$$F_m = F_{m-1}(x) + arg_h^{min} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) - h(x)) \tag{3.20}$$

Σύμφωνα με τον Friedman [27], η επαναληπτική διαδικασία χρησιμοποιεί την συνάρτηση κόστους και τις ασθενείς συναρτήσεις και στοχεύει στον υπολογισμό – προσέγγιση της αρνητικής κλίσης της συνάρτησης σφάλματος, προκειμένου να ενημερωθεί η συνάρτηση – εκτίμηση του επόμενου βήματος. Η αρνητική κλίση της συνάρτησης σφάλματος δίνεται παρακάτω:

$$-g(x_i) = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}] \tag{3.21}$$

Με βάση την παραπάνω εξίσωση, υπολογίζεται το βέλτιστο βήμα της ενίσχυσης κλίσης γ_m, και προσαρμόζεται η συνάρτηση $h_m(x)$ σύμφωνα με αυτό και όπου γ_m δίνεται ως εξής:

$$\gamma_m = argmin_\gamma \sum_{i=1}^{n} L(y_i, \hat{F}_{m-1}(x_i) + \gamma * h(x_i)) \tag{3.22}$$

Έτσι, δημιουργείται ένα σύνολο από δέντρα απόφασης F_m, καθένα από τα οποία για την εκπαίδευσή του χρησιμοποιεί στοιχεία από τα προηγούμενα δέντρα, προκειμένου να βελτιστοποιείται η ακρίβεια του τελικού συνόλου των δέντρων απόφαση.

Κεφάλαιο 4

Διαδικασία Εξαγωγής Δεδομένων

Στο κεφάλαιο αυτό παρουσιάζονται οι διαφορετικές διαδικασίες με τις οποίες πραγματοποιήθηκε η εξαγωγή των απαραίτητων για την εργασία δεδομένων. Η εξαγωγή έγινε χρησιμοποιώντας τόσο τεχνικές web scraping μέσω της πλατφόρμας Apify όσο και κλήσεις σε public facing APIs ή απευθείας κατέβασμα δεδομένων από ιστοσελίδες της κυβέρνησης της Νέας Υόρκης.

4.1 Ανάλυση της αξίας ενός ακινήτου

Η αξία ενός ακινήτου μπορεί να αναλυθεί σε δύο επιμέρους συνιστώσες:

- **Χαρακτηριστικά του ίδιου του ακινήτου**: Αυτά περιλαμβάνουν στοιχεία όπως το εμβαδόν του ακινήτου, τον αριθμό και το είδος των δωματίων, τα υλικά κατασκευής, την ηλικία του, την επίπλωση, κλπ.

- **Χαρακτηριστικά του περιβάλλοντος του ακινήτου**: Αυτά περιλαμβάνουν στοιχεία όπως η εκτίμηση των αγοραστών για την γειτονιά στην οποία βρίσκεται το ακίνητο, η εύκολη πρόσβαση σε συγκοινωνία, η κίνηση στους κοντινούς δρόμους, ο θόρυβος, η πλησιότητα σε καλά σχολεία, κλπ.

Οι συνιστώσες αυτές έχουν έναν βαθμό ανεξαρτησίας (δεν είναι όμως πλήρως ανεξάρτητες, π.χ. σε κάποιες πλούσιες γειτονιές είναι πιο πιθανό να βρεθούν περισσότερα ακίνητα με πολλά τετραγωνικά και ακριβή επίπλωση). Είναι συνεπώς και οι δύο απαραίτητες προκειμένου να βρεθεί μια ικανοποιητική προσέγγιση της τιμής ενός ακινήτου. Στη συνέχεια παρουσιάζεται η διαδικασία συλλογής δεδομένων και για τις δύο συνιστώσες και αναλύονται τα επιμέρους χαρακτηριστικά που συλλέγονται.

4.2 Zillow Scraping

4.2.1 Η πλατφόρμα Zillow

Το Zillow Group είναι ένας δημοφιλής ιστότοπος ακινήτων στις Ηνωμένες Πολιτείες. Η Zillow και οι συνεργάτες της προσφέρουν στους πελάτες μια εμπειρία κατ' απαίτηση για πώληση, αγορά, ενοικίαση και χρηματοδότηση με διαφάνεια και σχεδόν απρόσκοπτη υπηρεσία end-to-end. Επιλέχθηκε να χρησιμοποιηθεί ως η πηγή για τα δεδομένα ακινήτων χάρη στον μεγάλο αριθμό αγγελιών που προσφέρει, τα μεταδεδομένα που έχει διαθέσιμα (π.χ. το ZHVI το οποίο θα αναλυθεί παρακάτω) αλλά και τη δυνατότητα για αναζήτηση ακινήτων τα οποία έχουν ήδη πωληθεί.

Το τελευταίο είναι σημαντικό καθώς επιτρέπει τον προσδιορισμό της πραγματικής αξίας ενός ακινήτου με όσο το δυνατόν μικρότερο σφάλμα. Σε περίπτωση που σαν μέτρο της αξίας του ακινήτου χρησιμοποιούνταν η τιμή αγγελίας θα υπήρχε πιθανό σφάλμα λόγω υπερκοστολόγησης εκ μέρους του πωλητή. Αντίθετα, η τελική τιμή πώλησης αντιπροσωπεύει την πιο αντικειμενική αξία του ακινήτου τη στιγμή που πουλήθηκε (με μικρές αποκλίσεις από την πραγματική αξία σε περίπτωση που, για παράδειγμα, ο πωλητής βιαζόταν να το πουλήσει και κατέληξε σε χαμηλότερη από την πραγματική τιμή).

Ένα επιπλέον πλεονέκτημα του Zillow είναι το Zillow Estimate (Zestimate) το οποίο αποτελεί την πρόβλεψη του Zillow για την πραγματική αξία ενός ακινήτου. Η πρόβλεψη αυτή μπορεί να αποτελέσει μέτρο σύγκρισης για το πόσο εύστοχες είναι οι προβλέψεις ενός άλλου συστήματος μηχανικής μάθησης όπως αυτό που θα αναπτυχθεί στην παρούσα διπλωματική.

4.2.2 Εξαγωγή δεδομένων από την πλατφόρμα Zillow

Για την εξαγωγή δεδομένων από το Zillow χρησιμοποιήθηκε η πλατφόρμα Apify. Συγκεκριμένα, δημιουργήθηκε ένας actor ο οποίος επέλεγε πληροφορίες μόνο για πωληθέντα ακίνητα στην πόλη της Νέας Υόρκης. Η δομή του βασίστηκε στην δομή του actor [28] ο οποίος κάνει scrape πληροφορίες μη-πωληθέντων ακινήτων. Η λογική του βασίζεται στη διαρκή τομή του χώρου αναζήτησης (Νέα Υόρκη) στα τέσσερα μέχρι τα υποσύνολά του να περιέχουν αριθμό ακινήτων μικρότερο του ορίου αναζήτησης του Zillow. Στη συνέχεια οι πληροφορίες των ακινήτων αυτών αποθηκεύονται από τον scraper. Προκειμένου να λειτουργήσει απρόσκοπτα η διαδικασία έγινε χρήση των proxies που προσφέρει το Apify.

Ο actor έτρεξε με τις εξής παραμέτρους:

- Τύπος ακινήτων: Sold.
- Μέγιστο zoom στον χάρτη τις Νέας Υόρκης: 20 φορές (20 τομές στα 4).
- Μέγιστη ηλικία αγγελίας: 01/2016 (στην πραγματικότητα το Zillow δεν διατηρεί τόσο παλαιά δεδομένα, συνεπώς η παλαιότερη αγγελία είχε ημερομηνία 01/2018).
- Τιμή: Μεταξύ 100.000 και 20.000.000 δολλάρια (το εύρος αυτό συρρικνώνεται περαιτέρω στο στάδιο της επεξεργασίας δεδομένων).

- Μία συνάρτηση επιλογής που ορίζει ποια χαρακτηριστικά θα συλλέξει ο scraper.

Λόγω περιορισμών στις δωρεάν υπηρεσίες του Apify έγιναν συνολικά 7 runs του actor για να συλλεχθεί επαρκής ποσότητα δεδομένων. Τα δεδομένα αυτά αποθηκεύτηκαν σε μορφή Json και στη συνέχεια δέχτηκαν επεξεργασία στη γλώσσα Python όπως περιγράφεται στο επόμενο κεφάλαιο.

4.2.3 Δομή δεδομένων Zillow

Ακολουθεί η παρουσίαση ενός ενδεικτικού ακινήτου που έγινε scrape από το Zillow. Κάθε είδος χαρακτηριστικού περιγράφεται στη συνέχεια.

Ενδεικτική δομή δεδομένων ακινήτου


```
{
   "zpid":29851110,
   "homeStatus":"RECENTLY_SOLD",
   "address":{
      "streetAddress":"220 Kirby St",
      "city":"Bronx",
      "state":"NY",
      "zipcode":"10464",
      "neighborhood":"None",
      "community":"None",
      "subdivision":"None"
   },
   "bedrooms":3,
   "bathrooms":2,
   "price":749000,
   "yearBuilt":1940,
   "longitude":-73.78528594970703,
   "latitude":40.8507080078125,
   "livingArea":1804,
   "currency":"USD",
   "homeType":"SINGLE_FAMILY",
   "timeZone":"America/New_York",
   "lastSoldPrice":749000,
   "zestimate":738338,
   "zestimateLowPercent":"5",
   "zestimateHighPercent":"5",
   "rentZestimate":2300,
   "restimateLowPercent":"23",
```

```
   "restimateHighPercent":"21",
   "solarPotential":{
      "sunScore":90.95,
      "buildFactor":76,
      "climateFactor":6.17,
      "electricityFactor":5.82,
      "solarFactor":2.96
   },
   "taxAssessedValue":501000,
   "taxAssessedYear":2019,
   "dateSold":1609372800000,
   "lotSize":7318,
   "mortgageRates":{
      "thirtyYearFixedRate":3.295,
      "fifteenYearFixedRate":2.943,
      "arm5Rate":3.832
   },
   "propertyTaxRate":0.95,
   "pageViewCount":76,
   "taxHistory":[
      {
         "time":1548662543619,
         "taxPaid":4998.8,
         "taxIncreaseRate":0.011855688,
         "value":30060,
         "valueIncreaseRate":0.27286586
      }
   ],
   "abbreviatedAddress":"220 Kirby St",
   "daysOnZillow":28,
   "url":"https://www.zillow.com/homedetails/220-Kirby-St-
   Bronx-NY-10464/29851110_zpid/",
   "photos":[
      "https://photos.zillowstatic.com/fp/ca18f67b23b57a9c5c2
   cd84d9b7fb26f-p_f.jpg",
   ],
   "description":"None",
   "hdpUrl":"/homedetails/220-Kirby-St-Bronx-NY-10464/2985111
   0_zpid/",
   "newConstructionType":"None",
   "photoCount":36,
```

```
"homeFacts":"None",
"resoFacts":{
   "atAGlanceFacts":[
      {
         "factValue":"SingleFamily",
         "factLabel":"Type"
      },
      {
         "factValue":"1940",
         "factLabel":"Year Built"
      },
      {
         "factValue":"Natural Gas, Forced Air",
         "factLabel":"Heating"
      },
      {
         "factValue":"Central Air",
         "factLabel":"Cooling"
      },
      {
         "factValue":"Driveway, On Street",
         "factLabel":"Parking"
      },
      {
         "factValue":"0.17 Acres",
         "factLabel":"Lot"
      }
   ],
   "bedrooms":3,
   "bathrooms":2,
   "bathroomsFull":1,
   "bathroomsThreeQuarter":"None",
   "bathroomsHalf":1,
   "bathroomsOneQuarter":"None",
   "bathroomsPartial":"None",
   "mainLevelBathrooms":"None",
   "rooms":[

   ],
   "basement":"Finished,Full,Walk-Out Access",
   "flooring":[
```

```
            "Hardwood"
        ],
        "heating":[
            "Natural Gas",
            "Forced Air"
        ],
        "hasHeating":true,
        "cooling":[
            "Central Air"
        ],
        "hasCooling":true,
        "appliances":[
            "Dishwasher",
            "Dryer",
            "Microwave",
            "Range",
            "Refrigerator",
            "Washer"
        ],
        "laundryFeatures":"None",
        "fireplaces":"None",
        "fireplaceFeatures":"None",
        "hasFireplace":"None",
        "furnished":false,
        "commonWalls":"None",
        "buildingArea":"None",
        "livingArea":"1,804 sqft",
        "aboveGradeFinishedArea":"None",
        "belowGradeFinishedArea":"None",
        "virtualTour":"None",
        "parking":0,
        "parkingFeatures":[
            "Driveway",
            "On Street"
        ],
        "garageSpaces":0,
        "coveredSpaces":"None",
        "hasAttachedGarage":false,
        "hasGarage":false,
        "openParkingSpaces":"None",
        "hasOpenParking":false,
```

```
    "carportSpaces":"None",
    "hasCarport":false,
    "otherParking":"None",
    "accessibilityFeatures":"None",
    "levels":"2.00",
    "stories":"None",
    "entryLevel":"None",
    "entryLocation":"None",
    "hasPrivatePool":"None",
    "hasSpa":false,
    "spaFeatures":"None",
    "exteriorFeatures":[
       "Basketball Court"
    ],
    "patioAndPorchFeatures":[
       "Patio"
    ],
    "fencing":"None",
    "view":[
       "Water"
    ],
    "hasView":false,
    "hasWaterfrontView":"None",
    "waterfrontFeatures":"None",
    "frontageType":"None",
    "frontageLength":"None",
    "topography":"None",
    "woodedArea":"None",
    "vegetation":"None",
    "canRaiseHorses":false,
    "lotSize":"0.17 Acres",
    "lotSizeDimensions":"None",
    "otherStructures":"None",
    "additionalParcelsDescription":"None",
    "hasAdditionalParcels":false,
    "parcelNumber":"05645-0114, 05645-0115",
    "hasAttachedProperty":false,
    "hasLandLease":false,
    "landLeaseAmount":"None",
    "zoning":"None",
    "zoningDescription":"None",
```

```
      "homeType":"SingleFamily",
      "architecturalStyle":"Ranch",
      "constructionMaterials":[
         "Frame",
         "Vinyl Siding"
      ],
      "foundationDetails":[

      ],
      "roofType":"None",
      "windowFeatures":"None",
      "propertyCondition":"None",
      "isNewConstruction":"None",
      "yearBuilt":1940,
      "developmentStatus":"None",
      "yearBuiltEffective":"None",
      "onMarketDate":1594771200000,
      "builderModel":"None",
      "builderName":"None",
      "hasHomeWarranty":false,
      "electric":"None",
      "hasElectricOnProperty":"None",
      "gas":"None",
      "sewer":[
         "Public Sewer"
      ],
      "waterSources":"None",
      "utilities":"None",
      "greenBuildingVerificationType":"None",
      "greenEnergyEfficient":"None",
      "greenIndoorAirQuality":"None",
      "greenSustainability":[
         "Frame",
         "Vinyl Siding"
      ],
      "greenWaterConservation":"None",
      "numberOfUnitsInCommunity":"None",
      "numberOfUnitsVacant":"None",
      "storiesTotal":"None",
      "hasPetsAllowed":false,
      "hasRentControl":false,
```

```
      "buildingFeatures":"None",
      "structureType":"None",
      "buildingName":"None",
      "elementarySchool":"Call Listing Agent",
      "elementarySchoolDistrict":"Call Listing Agent",
      "middleOrJuniorSchool":"Call Listing Agent",
      "middleOrJuniorSchoolDistrict":"Call Listing Agent",
      "highSchool":"Call Listing Agent",
      "highSchoolDistrict":"Call Listing Agent",
      "securityFeatures":"None",
      "communityFeatures":[
         "Near Public Transportation"
      ],
      "isSeniorCommunity":"None",
      "cityRegion":"Bronx",
      "associationFee":"None",
      "hasAssociation":"None",
      "associationAmenities":"None",
      "associationFeeIncludes":"None",
      "associationName":"None",
      "associationPhone":"None",
      "associationFee2":"None",
      "associationName2":"None",
      "associationPhone2":"None",
      "taxAssessedValue":501000,
      "taxAnnualAmount":6870,
      "listingId":"None",
      "buildingAreaSource":"None",
      "specialListingConditions":"None",
      "otherFacts":[

      ],
      "listAOR":"HGAR"
   },
   "building":"None",
   "whatILove":"None",
   "timeOnZillow":"28 days",
   "favoriteCount":"None",
   "homeValues":{
      "region":{
         "zhvi":{
```

```
            "yoy":2.539420643104903,
            "value":546894
         },
         "shortName":"10464",
         "link":"None",
         "zhviForecast":{
            "value":573411
         }
      },
      "parentRegion":{
         "zhvi":{
            "yoy":2.3395243258422584
         }
      }
   },
   "stateSearchUrl":{
      "path":"/ny/"
   },
   "buildingId":"None",
   "schools":[
      {
         "distance":0.4,
         "name":"Ps 175 City Island",
         "rating":5,
         "level":"Elementary",
         "studentsPerTeacher":13,
         "assigned":"None",
         "grades":"K-8",
         "link":"https://www.greatschools.org/school?id=02566
&state=NY",
         "type":"Public",
         "size":333,
         "totalCount":1,
         "isAssigned":true
      }
   ],
   "schoolSearchUrl":"None",
   "buildingPermits":"None",
   "highlights":"None"
}
```

Listing 4.1: Ενδεικτικό Datum Ακινήτου Zillow

Με μια πρώτη ματιά φαίνεται να υπάρχει μεγάλο πλήθος πληροφορίας για κάθε ακίνητο. Ακολουθεί η ανάλυση των πιο σημαντικών πεδίων κάθε ακινήτου:

- **zpid**: Το αναγνωριστικό της αγγελίας.
- **address**: Λεπτομέρειες διεύθυνσης. Στοιχεία όπως το zipcode θα χρησιμοποιηθούν αργότερα.
- **bedrooms, bathrooms**: Αριθμός υπνοδωματίων και μπάνιων. Στα περισσότερα ακίνητα είναι άρτιος αριθμός αλλά σε ορισμένα δεκαδικός ώστε να υποδείξει την ύπαρξη π.χ. half-bathrooms.
- **yearBuilt**: Έτος κατασκευής.
- **longitude, latitude**: Οι γεωγραφικές συντεταγμένες του ακινήτου.
- **livingArea**: Το κατοικίσιμο εμβαδόν, δηλαδή εξαιρώντας χώρους όπως κήπο.
- **homeType**: Το είδος της κατοικίας
- **lastSoldPrice**: Η τιμή πώλησης σε δολλάρια.
- **zestimate**: Η πρόβλεψη του Zillow για την πραγματική τιμή του ακινήτου. Περιγράφεται πιο αναλυτικά παρακάτω.
- **solarPotential**: Πληροφορίες για την προοπτική ηλιακής ενεργοδότησης του ακινήτου.
- **taxAssessedValue, taxAssessedYear**: Πληροφορίες για την εκτίμηση αξίας του ακινήτου από την εφορία.
- **dateSold**: Ημερομηνία πώλησης σε μορφή Unix Timestamp.
- **lotSize**: Εμβαδόν οικοπέδου.
- **url**: Το url του ακινήτου στο Zillow.
- **description**: Περιγραφή της αγγελίας.
- **resoFacts**: Μεγάλη ομάδα χαρακτηριστικών τα οποία περιλαμβάνουν ορισμένες πληροφορίες που αναφέρθηκαν ήδη και επιπλέον:
  - **atAGlanceFacts**: Σύνολο από βασικά χαρακτηριστικά όπως είδος ακινήτου και έτος χτισίματος.
  - **bathroomsFull, bathroomsThreeQuarter, bathroomsHalf, bathroomsOneQuarter, bathroomsPartial, mainLevelBathrooms**: Πιο λεπτομερής παρουσίαση των μπάνιων ανά μέγεθος.

  - **rooms**: Περιγραφή άλλων δωματίων.
  - **basement**: Λεπτομέρειες υπογείου.
  - **flooring**: Υλικά δαπέδου.
  - **heating, hasHeating**: Θέρμανση.
  - **cooling, hasCooling**: Ψύξη.
  - **appliances**: Εξοπλισμός του ακινήτου.
  - **fireplaces, fireplaceFeatures, hasFireplace**: Πληροφορίες για τζάκι.
  - **parking, parkingFeatures, garageSpaces, coveredSpaces, hasAttachedGarage, hasGarage, openParkingSpaces, hasOpenParking, carportSpaces, hasCarport, otherParking**: Πληροφορίες για πάρκινγκ/γκαράζ.
  - **levels, stories**: Πληροφορίες ορόφων (ή σε τι όροφο βρίσκεται το ακίνητο).
  - **hasPrivatePool, hasSpa**: Πληροφορίες σπα και πισίνας.
  - **otherStructures**: Πληροφορίες για άλλες κατασκευές εντός οικοπέδου, π.χ. εργαστήριο.
  - **architecturalStyle**: Αρχιτεκτονική του ακινήτου.
  - **constructionMaterials**: Υλικά κατασκευής.
  - **yearBuiltEffective**: Θεωρητικός χρόνος κατασκευής αν ληφθούν υπόψιν και ανακαινίσεις/αναβαθμίσεις του ακινήτου.

- **homeValues**: Περιέχει την πληροφορία του ZHVI, μιας μέσης τιμής των ακινήτων που έχουν πωληθεί στην περιοχή στην οποία ανήκει το υπό εξέταση ακίνητο. Περιγράφεται πιο λεπτομερώς παρακάτω.

- **schools**: Πληροφορίες για κοντινά στο ακίνητο σχολεία. Δυστυχώς δεν περιέχει αρκετές πληροφορίες για όλα την πόλη της Νέας Υόρκης, συνεπώς οι πληροφορίες για τα σχολεία αντλήθηκαν εν τέλει από άλλη πηγή που παρουσιάζεται πιο κάτω.

Παρόλο που εκ πρώτης όψης τα δεδομένα φαίνονται πολύ πλούσια, γίνεται γρήγορα εμφανές στην ανάλυση πως πολλά από τα χαρακτηριστικά περιέχουν ελλειπή ή λανθασμένα στοιχεία, γεγονός που μειώνει την ποιότητα του dataset. Οι ελλείψεις και οι αντίστοιχες διορθώσεις καλύπτονται στο επόμενο κεφάλαιο.

Zestimate

Το μοντέλο αποτίμησης σπιτιού Zestimate είναι η εκτίμηση της Zillow για την αγοραία αξία ενός σπιτιού. Το Zestimate ενσωματώνει δημόσια δεδομένα και δεδομένα που υποβάλλονται από χρήστες, λαμβάνοντας υπόψη τα πραγματικά χαρακτηριστικά, την τοποθεσία και τις συνθήκες της αγοράς.

Η ακρίβεια του Zestimate εξαρτάται από την τοποθεσία και τη διαθεσιμότητα δεδομένων σε μια περιοχή. Ορισμένες περιοχές διαθέτουν πιο λεπτομερείς πληροφορίες για το σπίτι - όπως

τετραγωνικά πλάνα και αριθμός υπνοδωματίων ή μπάνιων - και άλλες περιοχές δεν έχουν. Όσο περισσότερα διαθέσιμα δεδομένα, τόσο ακριβέστερη θα είναι η τιμή Zestimate [29].

Λόγω των μεγάλων ποσών και εργασίας που έχουν διατεθεί για τη βελτίωση του αλγορίθμου πίσω από το Zestimate [30], το Zestimate αποτελεί χρήσιμο benchmark για την αποδοτικότητα όποιου συστήματος κατασκευαστεί με σκοπό την πρόβλεψη τιμών.

ZHVI

Το Zillow Home Value Index (ZHVI) είναι ένα ομαλό, εποχιακά προσαρμοσμένο μέτρο της τυπικής αξίας του σπιτιού και των αλλαγών στην αγορά σε μια δεδομένη περιοχή και τύπο κατοικίας. Αντικατοπτρίζει την τυπική τιμή για τα σπίτια στο 35ο έως το 65ο εκατοστημόριο.

Ο δείκτης βελτιστοποιείται για την επίτευξη τριών κύριων στόχων σύμφωνα με το Zillow [31]:

- **Επικαιρότητα**: Τα δεδομένα για έναν συγκεκριμένο μήνα δημοσιεύονται την τρίτη Πέμπτη του επόμενου μήνα - δηλαδή, τα δεδομένα για τον Νοέμβριο του 2019 δημοσιεύονται την Πέμπτη, 19 Δεκεμβρίου. Άλλοι δείκτες στέγασης δημοσιεύουν συχνά μηνιαία δεδομένα με σημαντική καθυστέρηση ενός μήνα ή περισσότερο μετά το κλείσιμο ενός δεδομένου μήνα.

- **Περιεκτικότητα**: Το ZHVI βασίζεται σε Zestimates που υπολογίζονται σε περισσότερα από 100 εκατομμύρια σπίτια στις Η.Π.Α., συμπεριλαμβανομένων νέων κατοικιών και/ή σπιτιών που δεν έχουν διαπραγματευτεί στην ανοιχτή αγορά εδώ και πολλά χρόνια. Αυτό προσφέρει μια πληρέστερη εικόνα από τους δείκτες που βασίζονται αποκλειστικά στα δεδομένα που καταγράφονται μόνο σε εκείνα τα σπίτια που πωλούν σε μια δεδομένη περίοδο.

- **Ορατότητα**: Λόγω του τρόπου κατασκευής του, το ZHVI δίνει στους χρήστες τη δυνατότητα να παρατηρούν δυναμική σε πολύ μικρές περιοχές ή/και σε πολύ συγκεκριμένα υποσύνολα σπιτιών. Το ZHVI κυκλοφόρησε το 2006 και στην πιο πρόσφατη επανάληψή του πριν από τη δημοσίευση δεδομένων του Νοεμβρίου 2019 υπολογίστηκε ως η μέση τιμή Zestimate για ένα σταθερό (με την πάροδο του χρόνου) σύνολο σπιτιών σε μια δεδομένη περιοχή, που αντιπροσωπεύει τη μέση τιμή κατοικίας της περιοχής. Επειδή το απόθεμα των σπιτιών ήταν σταθερό με την πάροδο του χρόνου, η ανάπτυξη από μήνα σε μήνα υπό αυτές τις παραδοχές θα μπορούσε να ερμηνευθεί ως εκτίμηση του τυπικού σπιτιού.

Για τους λόγους αυτούς το ZHVI αποτελεί μία πρώτη ένδειξη της αξίας ενός ακινήτου με βάση την αξία ενός μέσου ακινήτου παρομοίων χαρακτηριστικών στην περιοχή του.

Συλλογή όλων των ZHVI

Προκειμένου να εξεταστεί η εξέλιξη των τιμών σε μια περιοχή απαιτείται η συλλογή όλων των διαθέσιμων ZHVIs για κάθε ταχυδρομικό κώδικα (zipcode) της Νέας Υόρκης. Για τον

λόγο αυτό αποθηκεύτηκαν τα ZHVIs από τον σύνδεσμο https://www.zillow.com/research/data/ σε μορφή csv. Το τελικό αρχείο περιείχε τα ZHVIs για κάθε ταχυδρομικό κώδικα από μία χρονιά και μετά.

Τα διαφορετικά ZHVIs θα χρησιμοποιηθούν προκειμένου να απεξαρτηθούν οι τιμές των ακινήτων από τον χρονικό παράγοντα. Για παράδειγμα, ένα ακίνητο που πωλήθηκε το 2018 ενδεχομένως να πωλούνταν για ελαφρώς μεγαλύτερη ή μικρότερη τιμή το 2021. Η μετατροπές τιμών θα αναλυθούν στο επόμενο επίπεδο.

4.3 Συλλογή δεδομένων από το SchoolDigger.com

4.3.1 SchoolDigger.com

Το SchoolDigger.com ιδρύθηκε το 2006 με σκοπό να προσφέρει ενημερωμένες επιλογές σχετικά με την επιλογή σχολείου από γονείς. Η βάση δεδομένων του περιέχει λεπτομερή προφίλ για πάνω από 136.000 σχολεία σε κάθε πολιτεία των ΗΠΑ.

Το SchoolDigger κατατάσσει τα σχολεία με βάση το μέσο τυπικό σκορ κάθε σχολείου. Το σχολείο με την υψηλότερη μέση τυπική βαθμολογία κατατάσσεται 1 στην πολιτεία. Το μέσο τυπικό σκορ ενός σχολείου υπολογίζεται ως εξής [32]:

Υπολογίζεται πρώτα ένα Z-Score για τη βαθμολογία (για παράδειγμα) μαθηματικών 4ης τάξης ενός σχολείου.Το Z-Score είναι το πολλαπλάσιο των τυπικών αποκλίσεων που απέχει η βαθμολογία μαθηματικών 4ης τάξης για παράδειγμα του σχολείου από τη μέση βαθμολογία 4ης τάξης της περιοχής στην οποία βρίσκεται. Ένα θετικό Z-Score σημαίνει ότι το σκορ του σχολείου είναι πάνω από τη μέση βαθμολογία των σχολείων της περιοχής και ένα αρνητικό Z-Score σημαίνει ότι το σκορ είναι κάτω από τη μέση βαθμολογίας.

$$z = \frac{\chi - \mu}{\sigma} \tag{4.1}$$

Όπου x είναι η βαθμολογία μαθηματικής επάρκειας 4ης τάξης, μ είναι η μέση βαθμολογία όλων των βαθμολογιών μαθημάτων 4ης τάξης της περιοχής και σ είναι η τυπική απόκλιση των βαθμολογιών μαθηματικών 4ης τάξης της περιοχής. Αυτό το Z-Score στη συνέχεια χαρτογραφείται χρησιμοποιώντας την τυπική κανονική κατανομή, η οποία δίνει το τυπικό σκορ.

Ο υπολογισμός αυτός γίνεται για όλα τα εξεταζόμενα μαθήματα, για κάθε τάξη και έτσι προκύπτει το τελικό Z-Score του σχολείου.

4.3.2 SchoolDigger.com API

Ο τρόπος με τον οποίο συλλέχθηκαν τα δεδομένα από το SchoolDigger.com είναι μέσω του public facing API που προσφέρει στο https://api.schooldigger.com. Για τον σκοπό αυτό χρησιμοποιήθηκε το trial του PRO version του API το οποίο επέτρεπε μέχρι 100 κλήσεις ανά λεπτό. Προκειμένου να επιτύχουν οι κλήσεις προς το συγκεκριμένο endoint απαιτείται ένα ζεύγος appIDGlobal - appKeyGlobal τα οποία παρέχονται από το PRO version και ο περιορισμός των κλήσεων σε μία ανά δευτερόλεπτο προκειμένου να μην υπερφορτωθεί το endpoint.

Τα δεδομένα χωρίζονται σε τρεις ομάδες ανάλογα με τις ηλικίες που εξυπηρετεί κάθε σχολείο: Elementary, Middle Schools και High Schools.

Τα δεδομένα που συλλέχθηκαν από το SchoolDigger.com πέρασαν από επεξεργασία σε γλώσσα Python.

4.3.3 Περιγραφή δεδομένων

Τα τελικά data που δημιουργήθηκαν είχαν την εξής μορφή:

```
{
   "schoolid":"360007805773",
   "schoolName":"...",
   "url":"https://www.schooldigger.com/go/NY/schools/0007805773/school.aspx",
   "state":"NY",
   "street":"100 W 77th St",
   "zip":"10024",
   "latitude":40.780961,
   "longitude":-73.978189,
   "lowGrade":"K",
   "highGrade":"8",
   "schoolLevel":"Elementary",
   "isPrivate":false,
   "score2016":97.57411,
   "score2017":97.8076,
   "score2018":97.06496,
   "score2019":99.03698,
   "pupilTeacherRatio2016":19.2,
   "pupilTeacherRatio2017":19.6,
   "pupilTeacherRatio2018":19.2,
   "pupilTeacherRatio2019":20.0
}
```

Listing 4.2: Ενδεικτικό Datum Σχολείου SchoolDigger.com API

Όπως φαίνεται παραπάνω σημαντικά στοιχεία κάθε σχολείου αποτελούν τα:

- **zip, latitude, longitude**: Πληροφορίες τοποθεσίας οι οποίες θα χρησιμοποιηθούν ώστε να βρεθούν τα πιο κοντινά σχολεία σε κάθε ακίνητο και με βάση αυτά να προκύψει μία "σχολική βαθμολογία" για το ακίνητο.
- **schoolLevel**: Η ηλικιακή ομάδα του σχολείου.
- **isPrivate**: Αν το σχολείο είναι δημόσιο ή ιδιωτικό. Μετά την ανάλυση των δεδομένων αποκαλύφθηκε πως το API παρείχε δεδομένα για δημόσια σχολεία μόνο. Συνεπώς,

δεν θα υπάρχει πλήρης εικόνα για όλα τα σχολεία τα οποία βρίσκονται πλησίον ενός ακινήτου. Ακόμα και τα δημόσια σχολεία μόνο ωστόσο εξακολουθούν να δίνουν μια καλή ένδειξη για το εκπαιδευτικό επίπεδο της εκάστοτε περιοχής.

- **scoreX, pupilTeacherRatioX**: Το σκορ του σχολείου και η αναλογία μαθητών προς καθηγητές κάθε σχολείου για την εκάστοτε χρονιά. Θεωρείται πως η ιδανική αναλογία μαθητών προς καθηγητές κυμαίνεται στις χαμηλές δεκάδες (π.χ. 11), ειδικά στα μαθήματα θετικών κατευθύνσεων [33].

Τα δεδομένα αυτά συσχετίστηκαν στη συνέχεια με τα δεδομένα ακινήτων με τον τρόπο που θα περιγραφεί στο επόμενο καφάλαιο.

4.4 Συλλογή δεδομένων 311 services, NYPD Complaint Data και Εγκαταστάσεων

4.4.1 New York Open Data

Το Open Data (data.cityofnewyork.us) είναι συλλογή από δωρεάν δημόσια δεδομένα που δημοσιεύονται από πρακτορεία της Νέας Υόρκης και άλλους συνεργάτες. Περιλαμβάνουν πληθώρα δεδομένων όπως δεδομένα για κλήσεις προς την αστυνομία, δημογραφικά στοιχεία και δεδομένα δημοσίων/ιδιωτικών εγκαταστάσεων. Τα δεδομένα μπορούν να φιλτραριστούν, να οργανωθούν και να αποθηκευτούν σε διάφορες μορφές για επεξεργασία.

4.4.2 311 Services και NYPD Complaint Data Historic

Τα 311 services συγκεκριμένα αφορούν κλήσεις σε μη έκτακτες δημόσιες υπηρεσίες. Σκοπός ήταν να χρησιμοποιηθούν προκειμένου να εξεταστούν τα παράπονα για φασαρία ή ενοχλήσεις στην περιοχή κάθε σπιτιού. Για την ανάκτησή τους εφαρμόστηκαν τα εξής φίλτρα:

- Ημερομηνία παραπόνου να είναι μεταγενέστερη του 2017.
- Το είδος παραπόνου να έχει να κάνει με θόρυβο ("Noise").

Ωστόσο, στην ανάλυση που ακολούθησε τη συλλογή δεδομένων φάνηκε πως τα στοιχεία αυτά δεν είχαν ικανοποιητική συσχέτιση με την τιμή ενός ακινήτου και δεν συνεισέφεραν στην βελτίωση των υποψηφίων αλγορίθμων μηχανικής μάθησης. Συνεπώς, ακολούθησε συλλογή δεδομένων από το NYPD Complaint Data Historic dataset του New York Open Data με παρόμοιο περιορισμό στην ημερομηνία.

Η διαφορά του dataset αυτού είναι πως περιέχει καταγγελίες για απλές ενοχλήσεις όπως ηχορύπανση μέχρι σοβαρά εγκλήματα και ενδεχομένως να προσφέρει καλύτερη εικόνα από τα 311 service calls. Βασικές στήλες του dataset είναι το είδος του εγκλήματος και το ζευγάρι longitude, latitude που θα χρησιμοποιηθεί για να βρεθεί ο αριθμός καταγγελιών κοντά σε κάθε κατοικία. Επιπλέον, συλλέχθηκε ο πληθυσμός ανά ταχυδρομικό κώδικα της Νέας Υόρκης ώστε να χρησιμοποιηθεί προκειμένου να βρεθούν τα παράπονα ανά κάτοικο

και όχι απλά ο καθαρός αριθμός παραπόνων, που ενδεχομένως να είναι παραπλανητικός για πυκνοκατοικημένες περιοχές.

4.4.3 Εγκαταστάσεις

Τέλος, συγκεντρώθηκε από το New York Open Data ένα dataset με τις δημόσιες και ιδιωτικές εγκαταστάσεις στην πόλη της Νέας Υόρκης. Στόχος του dataset αυτού ήταν να αξιολογηθεί η επιρροή που έχουν τα διαφορετικά ήδη κοντινών εγκαταστάσεων στην τιμή ενός ακινήτου.

Συγκεκριμένα το dataset είχε τις εξής σημαντικές στήλες:

- **FACDOMAIN**: Η ευρύτερη κατηγορία της εγκατάστασης, π.χ. PUBLIC SAFETY, EMERGENCY SERVICES, AND ADMINISTRATION OF JUSTICE.
- **FACGROUP**: Πιο συγκεκριμένη κατηγορία της εγκατάστασης, π.χ. EMERGENCY SERVICES.
- **FACSUBGRP**: Υποκατηγορία της εγκατάστασης, π.χ. FIRE SERVICES.
- **FACTYPE**: Υποκατηγορία της εγκατάστασης, π.χ. FIREHOUSE.
- **OPTYPE**: Αν η εγκατάσταση είναι δημόσια ή ιδιωτική.
- **LATITUDE, LONGITUDE**: Γεωγραφικές συντεταγμένες που θα χρησιμοποιηθούν για τον υπολογισμό αποστάσεων.

Ο διαχωρισμός των εγκαταστάσεων σε ομάδες ανάλογα με την κατηγορία τους και η συχέτισή τους με τα ακίνητα περιγράφεται στο επόμενο κεφάλαιο.

4.5 Walkscores

Τέλος, ένα σημαντικό χαρακτηριστικό ενός ακινήτου είναι η ευκολία πρόσβασης σε καταστήματα, υπηρεσίες και σημαντικές τοποθεσίες με τα πόδια. Επιπλέον, σημαντικό χαρακτηριστικό είναι η ευκολία πρόσβασης σε μέσα συγκοινωνίας. Σε ορισμένες πόλεις, όπως η Νέα Υόρκη, σημασία μπορεί να έχει επιπλέον και η ευκολία μετακίνησης με ποδήλατο.

Στις εκτιμήσεις αυτές μπορεί να βοηθήσει η υπηρεσία που προσφέρει το walkscore.com. Το walkscore.com είναι ένας ιστότοπος ο οποίος παρέχει βαθμολογίες για τα παραπάνω χαρακτηριστικά σε μια κλίμακα 0-100 για κάθε διεύθυνση που του δίνεται.

4.5.1 Μεθοδολογία βαθμολόγησης

Όπως αναφέρεται στον ίδιο τον ιστότοπο [34], τα walkscore, transitscore και bikescore υπολογίζονται όπως παρουσιάζεται παρακάτω.

Walkscore

Η Walkscore μετρά τη δυνατότητα περπατήματος οποιασδήποτε διεύθυνσης χρησιμοποιώντας ένα πατενταρισμένο σύστημα. Για κάθε διεύθυνση, το Walk Score αναλύει μεγάλο αριθμό από διαδρομές πεζοπορίας σε κοντινές παροχές. Οι πόντοι απονέμονται με βάση την απόσταση από τις παροχές σε κάθε κατηγορία. Οι παροχές σε απόσταση 5 λεπτών με τα πόδια (0,25 μίλια) έχουν μέγιστους βαθμούς. Μια συνάρτηση αποσύνθεσης (decay function) χρησιμοποιείται για να δίνει πόντους σε πιο απομακρυσμένες παροχές, χωρίς να δίνονται πόντοι σε αποστάσεις μεγαλύτερες από 30 λεπτά με τα πόδια.

Το Walk Score μετρά επίσης τη φιλικότητα των πεζών αναλύοντας την πυκνότητα του πληθυσμού και τις μετρήσεις του δρόμου, όπως το μήκος τετραγώνων και η πυκνότητα διασταύρωσης. Οι πηγές δεδομένων περιλαμβάνουν το Google, το Factual, τα Great Schools, το Open Street Map, την απογραφή των ΗΠΑ, το Localeze και δεδομένα που προστέθηκαν από την κοινότητα χρηστών του Walk Score.

Οι βαθμολογίες μπορούν να κατηγοριοποιηθούν ως εξής:

90–100	**Εξαιρετικά προσβάσιμο με τα πόδια**: Καθημερινές διαδρομές δεν απαιτούν αυτοκίνητο.
70–89	**Πολύ προσβάσιμο με τα πόδια**: Οι περισσότερες διαδρομές δεν απαιτούν αυτοκίνητο.
50–69	**Κάπως προσβάσιμο με τα πόδια**: Κάποιες διαδρομές δεν απαιτούν αυτοκίνητο.
25–49	**Εξαρτώμενο από αυτοκίνητο**: Οι περισσότερες διαδρομές απαιτούν αυτοκίνητο.
0–24	**Πολύ εξαρτώμενο από αυτοκίνητο**: Σχεδόν όλες οι διαδρομές απαιτούν αυτοκίνητο.

Transitscore

Το Transit Score είναι ένα πατενταρισμένο μέτρο για το πόσο καλά εξυπηρετείται μια τοποθεσία με τις δημόσιες συγκοινωνίες. Η βαθμολογία βασίζεται σε δεδομένα που εκδίδονται σε τυπική μορφή από οργανισμούς δημόσιας συγκοινωνίας.

Για να υπολογιστεί μια βαθμολογία, εκχωρείται μια τιμή χρησιμότητας σε κοντινές διαδρομές διέλευσης με βάση τη συχνότητα, τον τύπο της διαδρομής (σιδηρόδρομος, λεωφορείο κ.λπ.) και την απόσταση από την πλησιέστερη στάση της διαδρομής. Η χρησιμότητα όλων των κοντινών διαδρομών αθροίζεται και ομαλοποιείται σε βαθμολογία μεταξύ 0 - 100.

Οι βαθμολογίες μπορούν να κατηγοριοποιηθούν ως εξής:

90–100	**Εξαιρετική πρόσβαση σε συγκοινωνίες**: Δημόσιες συγκοινωνίες παγκόσμιας κλάσης.
70–89	**Πολύ καλή πρόσβαση σε συγκοινωνίες**: Η συγκοινωνία είναι βολική για τα περισσότερα ταξίδια.
50–69	**Μέτρια πρόσβαση σε συγκοινωνίες**: Πολλές γειτονικές επιλογές δημόσιας συγκοινωνίας.
25–49	**Φτωχή πρόσβαση σε συγκοινωνίες**: Μερικές κοντινές επιλογές δημόσιας συγκοινωνίας.
0–24	**Κακή πρόσβαση σε συγκοινωνίες**: Λιγοστές επιλογές οι οποίες περιλαμβάνουν μόνο λεωφορεία.

Bikescore

Το Bike Score μετρά κατά πόσο μια περιοχή είναι καλή για ποδηλασία. Για μια δεδομένη τοποθεσία, το Βικε Σχορε υπολογίζεται μετρώντας την υποδομή του ποδηλάτου (λωρίδες, μονοπάτια κ.λπ.), λόφους, προορισμούς και οδική συνδεσιμότητα, καθώς και τον αριθμό των μετακινούμενων ποδηλάτων.

Αυτές οι βαθμολογίες βασίζονται σε δεδομένα από το USGS, το Open Street Map και την απογραφή των ΗΠΑ.

Οι βαθμολογίες μπορούν να κατηγοριοποιηθούν ως εξής:

90–100	**Εξαιρετικά καλή για ποδηλασία**: Καθημερινές εργασίες μπορούν να πραγματοποιηθούν με ποδήλατο.
70–89	**Πολύ καλή για ποδηλασία**: Η ποδηλασία είναι βολική για τα περισσότερα ταξίδια.
50–69	**Μέτρια για ποδηλασία**: Κάποια υποδομή ποδηλάτων.
0–49	**Φτωχή για ποδηλασία**: Ελάχιστη υποδομή ποδηλάτων.

4.5.2 Μεθοδολογία συλλογής δεδομένων

Προκειμένου να συλλεχθούν οι πληροφορίες για τα παραπάνω scores έγινε χρήση του Walkscore API. Το συγκεκριμένο API δεν χρειάζεται κάποιο κλειδί ώστε να επιστρέψει απαντήσεις στις κλήσεις που δέχεται. Συνεπώς, το μόνο που χρειάστηκε ήταν να δημιουργηθούν οι κατάλληλες κλήσεις προς αυτό με τη γλώσσα Python και στη συνέχεια να διαβαστεί η παραγώμενη html.

Για την δημιουργία των κλήσεων χρησιμοποιήθηκαν τα στοιχεία διευθύνσεων/τοποθεσίας από τα δεδομένα. Συγκεκριμένα, για να δημιουργηθεί μια κλήση όπως η https://www.walkscore.com/score/220 kirby-st-the-bronx-ny-10464 λαμβάνεται το πεδίο address από το entry του ακινήτου και τα υποστοιχεία του, όπως διεύθυνση και ταχυδρομικός κώδικας, περνούν κατάλληλη επεξεργασία και ενώνονται ώστε να σχηματίσουν την κλήση.

Στη συνέχεια, διαβάζεται η απάντηση του API και με χρήση XPath Queries γίνεται εξόρυξη των τριών Scores. Παράδειγμα XPath Query αποτελεί το

`//div[@data-eventsrc="score page transit badge"]/img/@src`

το οποίο οδηγεί σε ένα string της μορφής

`pp.walk.sc/badge/transit/score/37.svg`

Το νούμερο 37 στο συγκεκριμένο string αποτελεί το Transit Score του συγκεκριμένου ακινήτου.

Με παρόμοιο τρόπο συλλέγονται και τα υπόλοιπα σκορ για κάθε ακίνητο και αποθηκεύονται όλα σε dictionaries με κλειδί το zpid (αναγνωριστικό) κάθε ακινήτου.

```
{
  "zpid":29851028,
  "walkScore":61,
  "transitScore":88,
  "bikeScore":49
},
{
  "zpid":2087430386,
  "walkScore":73,
  "transitScore":37,
  "bikeScore":"None"
},
{
  "zpid":32057100,
  "walkScore":68,
  "transitScore":54,
  "bikeScore":64
},
{
  "zpid":29851105,
  "walkScore":72,
  "transitScore":37,
  "bikeScore":70
},
{
  "zpid":2146920923,
  "walkScore":70,
  "transitScore":37,
  "bikeScore":69
}
...
```

Listing 4.3: Ενδεικτική δομή dictionary Walkscores

Επειδή ορισμένες κλήσεις μπορεί να επιστρέψουν κωδικό 404 (not found), γίνονται μέχρι και 3 επαναλήψεις για κλήσεις που απέτυχαν. Στο τέλος αυτών των κλήσεων υπάρχουν τα Walkscore και Transitscore σχεδόν για όλες τις κατοικίες ενώ το Bikescore υπάρχει για περίπου τις μισές. Στο στάδιο επεξεργασίας των δεδομένων οι τιμές που λείπουν θα συμπληρωθούν με βάση τις τιμές των πιο κοντινών γειτόνων κάθε ακινήτου.

4.6 Διαχωρισμός Στοιχείων Ακινήτου και Στοιχείων Περιοχής

Αυτά είναι όλα τα στοιχεία τα οποία συλλέχθηκαν και κατέληξαν να χρησιμοποιηθούν στα τελικά μοντέλα που εκπαιδεύτηκαν για την πρόβλεψη τιμών ακινήτων.

Από τα δεδομένα τα οποία διατηρήθηκαν κάποια χαρακτηρίζουν το ίδιο το ακίνητο και κάποια την περιοχή στην οποία βρίσκεται. Κάθε ένας από τους άξονες αυτούς αποτελεί ένα σημαντικό κομμάτι της τιμής. Δεν αρκεί όμως από μόνος του για να προσδιορίσει ικανοποιητικά την τιμή.

Αυτός είναι και ένας λόγος για τον οποίο η συλλογή ακόμα περισσότερων δεδομένων για το περιβάλλον ενός ακινήτου έχει διαρκώς μικρότερη αξία. Έχοντας ήδη μια εικόνα για τις παροχές κοντά στο ακίνητο, τα παράπονα που έχουν σημειωθεί κοντά του, την τιμή άλλων ακινήτων στην περιοχή, την ποιότητα κοντινών σχολείων, την πρόσβαση σε συγκοινωνίες, κλπ. έχουμε μια καλή προσέγγιση της αξίας περιοχής του ακινήτου. Παραπάνω εξωτερικά στοιχεία δεν θα συμβάλλουν καθοριστικά στον προσδιορισμό της τιμής του η οποία εξαρτάται πλέον κυρίως από εσωτερικά του χαρακτηριστικά όπως τα κατοικίσιμα τετραγωνικά μέτρα και τα δωμάτια.

Κεφάλαιο 5

Επεξεργασία δεδομένων

5.1 Διαμόρφωση αρχικού Dataframe

Το πρώτο βήμα στη δημιουργία του Data Pipeline μετά τη συλλογή των δεδομένων είναι η επεξεργασία τους. Αυτή περιλαμβάνει πολλά βήματα, όπως η διαγραφή διπλότυπων, η αφαίρεση στηλών και σειρών με ελειπή ή λανθασμένα δεδομένα και η συσχέτικη δεδομένων από διαφορετικά datasets όπως τα ακίνητα με τα walkscores.

5.1.1 Αφαίρεση διπλότυπων ακινήτων

Επειδή ο Apify actor που συνέλλεξε τα δεδομένα ακινήτων έτρεξε συνολικά 7 φορές, υπάρχουν στα δεδομένα διπλότυπα entries τα οποία θα χρειαστεί να αφαιρεθούν. Ο τρόπος με τον οποίο διαχωρίστηκαν τα μοναδικά δεδομένα ακινήτων είναι ο εξής:

Καταρχάς, κάθε ακινητο έχει ένα μοναδικό zpid - το αναγνωρισιτκό του στο Zillow. Συνεπώς, διαφορετικά ακίνητα θα έχουν και διαφορετικό zpid. Ωστόσο, κάποια από αυτά ενδέχεται να έχουν πωληθεί πάνω από μία φορά στη χρονική περίοδο που εξετάζουμε. Έτσι, δεν αρκεί να κρατηθούν μόνο τα μοναδικά zpids, αλλά χριάζεται να γίνει και έλεγχος πως κάποιο zpid δεν εμφανίζεται πάνω από μία φορά.

Εναλλακτικά, θα μπορούσαν να διατηρηθούν όλα τα entries από τα ακίνητα τα οποία έχουν πωληθεί πάνω από μία φορά. Επειδή τα entries αυτά ήταν λίγα (533) και ήταν προτιμότερο να διατηρηθεί το zpid ως μοναδικό αναγνωριστικό επιλέχθηκε να αφαιρεθούν όσα zpids είχαν εμφανιστεί έστω μία φορά ήδη.

Συνολικά, πριν την αφαίρεση των διπλότυπων το dataset είχε μέγεθος 276807 εγγραφές. Μετά την αφαίρεση των διπλότυπων είχε μέγεθος 77333 εγγραφές, 28% του αρχικού. Αυτό δείχνει πως υπήρχε υψηλό ποσοστό επικάλυψης μεταξύ των διαφορετικών runs του Apify Actor.

5.1.2 Αφαίρεση μη χρήσιμων στηλών

Προκειμένου να περάσουν από προεπεξεργασία τα δεδομένα ακινήτων, χρησιμοποιήθηκε το Pandas Library ώστε να διευκολύνει την ανάλυση και παρουσίαση των δεδομένων.

Αρχικά, αφαιρέθηκαν όλες οι κολώνες οι οποίες είχαν μόνο κενές τιμές. Επιπλέον, αφαιρέθηκαν οι κολώνες που περιείχαν δεδομένα με τιμές που δεν προσέφεραν χρήσιμη πληροφορία. Τέτοιες κολώνες για παράδειγμα ήταν το homeStatus (τιμές "SOLD" και "RECENTLY_SOLD" οι οποίες δεν έχουν κάποια χρησιμότητα) και το currency το οποίο είχε πάντα την τιμή "USD". Έτσι, αφαιρέθηκαν οι κολώνες:

- homeStatus
- currency
- timeZone
- mortgageRates
- propertyTaxRate
- pageViewCount
- taxHistory
- abbreviatedAddress
- daysOnZillow
- photos
- hdpUrl
- stateSearchUrl
- buildingId
- whatILove
- favoriteCount
- photoCount
- timeOnZillow
- schools

Η whatILove συγκεκριμένα θα μπορούσε να προσφέρει χρήσιμες πληροφορίες για αξιόλογα χαρακτηριστικά του ακινήτου αλλά μόλις το 6% των εγγραφών του περιείχε πληροφορίες. Επομένως, αφαιρέθηκε.

Οι σημαντικότερες κολώνες που απέμειναν μαζί με τα ποσοστά των εγγραφών που λείπουν για την καθεμία φαίνονται παρακάτω:

zpid	0.00 %
address	0.00 %
bedrooms	24.81 %
bathrooms	24.53 %
longitude	0.00 %
latitude	0.00 %
livingArea	11.33 %
homeType	0.00 %
lastSoldPrice	4.79 %
zestimate	6.95 %
solarPotential	19.06 %
taxAssessedValue	14.46 %
taxAssessedYear	13.34 %
dateSold	0.00 %
lotSize	14.52 %
url	0.00 %
description	9.71 %
resoFacts	15.86 %
homeValues	15.89 %
resoFacts	15.86 %

5.1.3 Εμπλουτισμός στηλών με στοιχεία από την στήλη Resofacts

Στη συνέχεια αναλύθηκε για κάθε ακίνητο η στήλη Resofacts η οποία περιέχει μέσα της μεγάλο αριθμό δεδομένων όπως παρουσιάστηκε στο κεφάλαιο 4. Τα δεδομένα από την κολώνα αυτή χρησιμοποιήθηκαν ώστε να δημιουργήσουν και να συμπληρώσουν τιμές του αρχικού dataframe. Για παράδειγμα, το στοιχείο bedrooms από την κολώνα Resofacts αντικατέστησε ή συμπλήρωσε την τιμή των bedrooms του αρχικού dataframe όπου η κολώνα Resofacts δεν ήταν κενή.

Για όλες τις εγγραφές που είχαν non-null resofacts στήλη ακολουθεί ένα dictionary με όλα τα στοιχεία της στήλης Resofacts και το αντίστοιχο ποσοστό non-null τιμών που περιέχονται στο σύνολο των Resofacts (παρουσίαζονται μόνο όσες έχουν ποσοστό non-null μεγαλύτερο του μηδέν.

```
{
 'parkingFeatures': 100.0,
 'furnished': 100.0,
 'hasAttachedGarage': 100.0,
 'hasPetsAllowed': 100.0,
 'hasAdditionalParcels': 100.0,
 'hasOpenParking': 100.0,
```

```
'constructionMaterials': 100.0,
'cityRegion': 100.0,
'view': 100.0,
'hasView': 100.0,
'garageSpaces': 100.0,
'parking': 100.0,
'hasAttachedProperty': 100.0,
'hasGarage': 100.0,
'homeType': 100.0,
'atAGlanceFacts': 100.0,
'hasRentControl': 100.0,
'hasLandLease': 100.0,
'hasCarport': 100.0,
'rooms': 100.0,
'foundationDetails': 100.0,
'canRaiseHorses': 100.0,
'hasHomeWarranty': 100.0,
'exteriorFeatures': 100.0,
'otherFacts': 100.0,
'hasSpa': 100.0,
'hasHeating': 99.37,
'heating': 99.37,
'hasCooling': 98.46000000000001,
'cooling': 96.09,
'appliances': 95.354,
'yearBuilt': 92.518,
'parcelNumber': 90.903,
'flooring': 89.17699999999999,
'livingArea': 87.33,
'communityFeatures': 86.811,
'taxAnnualAmount': 86.19,
'lotSize': 84.06400000000001,
'taxAssessedValue': 83.99,
'bathrooms': 78.919,
'bedrooms': 78.239,
'bathroomsFull': 72.30199999999999,
'bathroomsHalf': 72.284,
'bathroomsThreeQuarter': 60.292,
'stories': 59.282000000000004,
'yearBuiltEffective': 55.721,
'bathroomsOneQuarter': 55.595000000000006,
```

```
'middleOrJuniorSchoolDistrict': 39.056999999999995,
'elementarySchoolDistrict': 39.056999999999995,
'highSchoolDistrict': 39.056999999999995,
'basement': 32.042,
'hasFireplace': 27.16,
'hasPrivatePool': 26.5,
'structureType': 24.993000000000002,
'sewer': 20.988,
'onMarketDate': 16.918,
'architecturalStyle': 16.84,
'lotSizeDimensions': 14.616999999999999,
'isNewConstruction': 13.744,
'listAOR': 10.677,
'roofType': 10.624,
'greenSustainability': 10.578,
'propertyCondition': 9.908,
'fireplaces': 8.464,
'elementarySchool': 8.001,
'zoning': 7.978000000000001,
'electric': 7.954999999999999,
'highSchool': 7.466,
'middleOrJuniorSchool': 7.463,
'associationFee': 6.736000000000001,
'patioAndPorchFeatures': 5.856,
'hasAssociation': 5.079000000000001,
'buildingArea': 4.804,
'associationFeeIncludes': 4.311,
'fencing': 4.02,
'utilities': 3.138,
'levels': 2.7279999999999998,
'virtualTour': 2.328,
'zoningDescription': 2.158,
'laundryFeatures': 1.66,
'windowFeatures': 1.346,
'securityFeatures': 1.1360000000000001,
'otherStructures': 0.936,
'storiesTotal': 0.9079999999999999,
'associationFee2': 0.827,
'commonWalls': 0.569,
'hasWaterfrontView': 0.43499999999999994,
'associationAmenities': 0.326,
```

```
 'waterfrontFeatures': 0.29,
 'buildingAreaSource': 0.261,
 'isSeniorCommunity': 0.258,
 'additionalParcelsDescription': 0.197,
 'specialListingConditions': 0.189,
 'greenEnergyEfficient': 0.11800000000000001,
 'developmentStatus': 0.1,
 'spaFeatures': 0.095,
 'buildingFeatures': 0.066,
 'greenWaterConservation': 0.048,
 'hasElectricOnProperty': 0.041,
 'greenBuildingVerificationType': 0.031,
 'gas': 0.025,
 'frontageLength': 0.025,
 'buildingName': 0.025,
 'associationName': 0.02,
 'accessibilityFeatures': 0.015,
 'fireplaceFeatures': 0.013999999999999999,
 'builderModel': 0.006,
 'topography': 0.003,
 'coveredSpaces': 0.002,
 'openParkingSpaces': 0.002,
 'otherParking': 0.002
}
```

Listing 5.1: Υποκολώνες της κολώνας Resofacts με το ποσοστό non-null τιμών

Πέρα από τις υπάρχουσες κολώνες των οποίων τα στοιχεία συμπληρώθηκαν από την κολώνα Resofacts, οι εξής νέες κολώνες δημιουργήθηκαν:

- bathroomsFull
- bathroomsThreeQuarter
- bathroomsHalf
- bathroomsOneQuarter
- yearBuiltEffective
- hasFinishedBasement
- hasUnfinishedBasement
- heating

- cooling
- appliances
- hasFireplace
- hasParking: Παράγεται από συνδυασμό διάφορων πεδίων της Resofacts που αφορούν Parking.
- hasGarage: Παράγεται από συνδυασμό διάφορων πεδίων της Resofacts που αφορούν γκαράζ.
- stories
- hasPrivatePool
- hasSpa
- exteriorFeatures
- patioAndPorchFeatures
- fencing
- hasNaturalView: Τιμή True σε περίπτωση που το πεδίο View των Resofacts έχει τιμή από την λίστα ['Water', 'Park', 'Panoramic', 'Park/greenbelt', 'Mountain', 'River', 'Skyline', 'Bridge(s)', 'Lake'] ή το πεδίο hasWaterfrontView των Resofacts είναι True.
- otherStructures
- hasAttachedProperty
- architecturalStyle
- constructionMaterials
- roofType
- structureType
- greenSustainability
- taxAssessedValue

Από τις κολώνες αυτές αρκετές περιέχουν ελειπή δεδομένα και θα απορριφθούν κατά τη διαδικασία της εκπαίδευσης αλλά για αρχή συμπεριλαμβάνονται στα δεδομένα.

Επιπλέον, η κολώνα bathrooms ανανεώθηκε ώστε να λάβει υπόψιν τις έξτρα πληροφορίες από τα bathroomsFull, bathroomsThreeQuarter, bathroomsHalf, bathroomsOneQuarter. Όπου υπήρχαν πληροφορίες για τα παραπάνω, η κολώνα bathrooms πήρε την τιμή του αθροίσματος των διαφορετικών τύπων μπάνιων με τον προφανή τρόπο:

$$bathroomsFull = 1,\ bathroomsThreeQuarter = \frac{3}{4},$$
$$bathroomsHalf = \frac{1}{2},\ bathroomsOneQuarter = \frac{1}{4} \tag{5.1}$$

Όσον αφορά πληροφορίες από το rooms, παρατηρήθηκε πως πολλές κολώνες είχαν τιμή στην κολώνα την κενή λίστα ([]). Αυτές που δεν είχαν κενή λίστα είχαν πληροφορίες μικρής χρησιμότητας όπως ότι το σπίτι διαθέτει σαλόνι, τραπεζαρία, μπάνιο, κλπ. τα οποία δεν θα βοηθήσουν στην πρόβλεψη τιμών. Παρόμοιο πρόβλημα υπήρχε και σε αρκετές άλλες κολώνες όπως η flooring στην οποία κυριαρχούσαν οι κενές λίστες και το "Hardwood".

Για την μελέτη όλων των στηλών που περιείχαν λίστες χρησιμοποιήθηκε η εξής συνάρτηση για την μελέτη των τιμών τους:

```
1  def exploreListColumn(df, columnName):
2      dictionary = dict()
3      for index, row in df.iterrows():
4          if type(row[columnName]) is list and len(row[
       columnName]) > 0:
5              for x in row[columnName]:
6                  if x is not None:
7                      capX = x.capitalize()
8                      if capX in dictionary:
9                          dictionary[capX] += 1
10                     else:
11                         dictionary[capX] = 1
12     pprint.pprint(dict(sorted(dictionary.items(), key=lambda
       item: item[1], reverse=True)), sort_dicts=False)
13
14 # example use:
15 > exploreListColumn(rsdf, 'flooring')
16 > {'Hardwood': 15151,
17  'Tile': 3282,
18  'Carpet': 1417,
19  'Laminate': 574,
20  'Wall to wall carpet': 460,
21  'Other': 382,
22  'Linoleum / vinyl': 344,
23  'Concrete': 140,
24  'Slate': 80,
25  'Softwood': 51,
26  'Plywood': 1}
```

```
27 >
```

Listing 5.2: Συνάρτηση για τη μελέτη στηλών με λίστες

Παράλληλα, οι κολώνες livingArea και lotSize είχαν τιμές σε square feet (τετραγωνικά πόδια) ή σε acres (περίπου 4 στρέμματα) στα Resofacts (και παρομοίως και στις αρχικές κολώνες χωρίς να φαίνεται όμως η μονάδα μέτρησης). Συνεπώς, προκειμένου να βρίσκονται στο ίδιο σύστημα μονάδων μετατράπηκαν όλες σε τετραγωνικά μέτρα.

Επιπλέον, η κολώνα homeType εμπλοτίστηκε με τον τύπο "Cooperative" ο οποίος υπήρχε μόνο εντός των Resofacts. Προηγουμένως, όλα τα "Cooperative" ακίνητα είχαν τύπο "CONDO".

Ύστερα από τα παραπάνω βήματα επεξεργασίας, η κολώνα Resofacts αφαιρέθηκε από τα δεδομένα.

5.1.4 Τελευταίες αλλαγές και τελική εικόνα του Dataframe

Στη συνέχεια, από την κολώνα homeValues του αρχικού dataframe λαμβάνεται η πληροφορία του ZHVI. Όπως έχει ήδη αναφερθεί στο κεφάλαιο 4, το ZHVI αποτελεί ένα είδος μέσης τιμής για την τιμή πώλησης ακινήτων παρομοίων με το υπό εξέταση, στην περιοχή του υπό εξέταση ακινήτου. Ακολούθως, η κολώνα homeValues διαγράφεται.

Επιπλέον, εξετάζοντας το πεδίο state της κολώνας address αφαιρούνται όλες οι εγγραφές που δεν ανήκουν στη Νέα Υόρκη (state != "NY").

Η κολώνα schools έχει διαγραφεί καθώς οι πληροφορίες που αφορούν τα σχολεία θα λειφθούν από το schooldigger.com.

Τέλος, από την κολώνα solarPotential δημιουργείτε μία νέα κολώνα sunScore που έχει μονάχα το ηλιακό σκορ του ακινήτου και η αρχική κολώνα solarPotential διαγράφεται.

Η τελική εικόνα του Dataframe έχει ως εξής. Ορισμένες κολώνες όπως το address ή τα διαφορετικά είδη bathrooms διατηρήθηκαν στο Dataframe σε περίπτωση που χρειαστεί να αναλυθούν περαιτέρω και όχι για αν χρησιμοποιηθούν στο training.

```
1  >   df.info()
2  >   <class 'pandas.core.frame.DataFrame'>
3      Int64Index: 77320 entries, 0 to 77332
4      Data columns (total 47 columns):
5       #   Column                  Non-Null Count  Dtype
6      ---  ------                  --------------  -----
7       0   zpid                    77320 non-null  int64
8       1   address                 77320 non-null  object
9       2   bedrooms                60861 non-null  float64
10      3   bathrooms               62513 non-null  float64
11      4   bathroomsFull           47037 non-null  object
12      5   bathroomsThreeQuarter   39223 non-null  object
```

```
13      6   bathroomsHalf           47025 non-null  object
14      7   bathroomsOneQuarter     36167 non-null  object
15      8   hasFinishedBasement     20849 non-null  object
16      9   hasUnfinishedBasement   20849 non-null  object
17      10  heating                 22836 non-null  object
18      11  cooling                 62512 non-null  object
19      12  appliances              32749 non-null  object
20      13  hasFireplace            19835 non-null  object
21      14  hasParking              33959 non-null  object
22      15  hasGarage               26249 non-null  object
23      16  stories                 40211 non-null  float64
24      17  hasPrivatePool          17243 non-null  object
25      18  hasSpa                  65056 non-null  object
26      19  exteriorFeatures        10135 non-null  object
27      20  patioAndPorchFeatures   3809 non-null   object
28      21  fencing                 2616 non-null   object
29      22  hasNaturalView          1718 non-null   object
30      23  otherStructures         608 non-null    object
31      24  hasAttachedProperty     65056 non-null  object
32      25  architecturalStyle      10956 non-null  object
33      26  constructionMaterials   17043 non-null  object
34      27  structureType           16261 non-null  object
35      28  greenSustainability     6883 non-null   object
36      29  zhvi                    65033 non-null  object
37      30  region                  65033 non-null  object
38      31  sunScore                62585 non-null  object
39      32  roofType                6912 non-null   object
40      33  yearBuilt               71986 non-null  float64
41      34  yearBuiltEffective      36256 non-null  float64
42      35  longitude               77320 non-null  float64
43      36  latitude                77320 non-null  float64
44      37  livingArea              68562 non-null  float64
45      38  homeType                77320 non-null  object
46      39  lastSoldPrice           77320 non-null  int64
47      40  zestimate               73617 non-null  float64
48      41  taxAssessedValue        66180 non-null  float64
49      42  taxAssessedYear         67012 non-null  float64
50      43  dateSold                77320 non-null  int64
51      44  lotSize                 66160 non-null  float64
52      45  url                     77317 non-null  object
53      46  description             69807 non-null  object
```

```
    dtypes: float64(12), int64(3), object(32)
    memory usage: 30.3+ MB
>
```

Listing 5.3: Τελική μορφή Dataframe μετά το αρχικό processing

5.2 Συνδυασμός των δεδομένων από όλες τις πηγές

5.2.1 Ανεξαρτητοποίηση τιμών από τον χρόνο

Αρχικά, σημαντικό είναι να σημειωθεί πως οι αγγελίες ακινήτων που απαρτίζουν το προς εξέταση dataset έχουν ημερομηνία πώλησης από 01/2018 έως 03/2021. Παρόλο που οι αλλαγές στη μέση τιμή ενός ακινήτου στην περίοδο αυτή δεν είναι δραματικές μιας και πρόκειται για μόλις 3 χρόνια, κρίνεται σωστότερο να προσαρμοστούν οι τιμές πώλησης ώστε να είναι ανεξάρτητες του χρόνου.

Για τον σκοπό αυτό θα χρησιμοποιηθούν τα ZHVIs που συλλέχθηκαν από τον ιστότοπο του Zillow. Συγκεκριμένα, κατέβηκαν σε μορφή csv τα ZHVIs για κάθε ταχυδρομικό κώδικα της Νέας Υόρκης όλα τα ιστορικά ZHVIs που ήταν διαθέσιμα.

Ακολούθως, εξετάστηκε αν υπάρχουν τα ZHVIs για κάθε ακίνητο του dataset, δηλαδή για όλα τα διαφορετικά zipcodes του dataset. Για κάθε ακίνητο λήφθηκε ο ταχυδρομικός κώδικας από την στήλη address και ελέχθηκε αν υπήρχε χρονοσειρά ZHVIs για τον ταχυδρομικό κώδικα αυτό. Για την πλειοψηφία των ακινήτων των οποίων ο ταχυδρομικός κώδικας δεν είχε αντίστοιχα ZHVIs, σαν ταχυδρομικός κώδικας θεωρήθηκε αυτός της πιο κοντινής περιοχής. Κάποια ακίνητα σε απομακρυσμένες περιοχές (71 συνολικά) αφαιρέθηκαν από το dataset.

Στη συνέχεια η τιμή κάθε ακινήτου δέχθηκε επεξεργασία ώστε να έρθει στην τιμή που θα είχε αν είχε πωληθεί τον 03/2021. Με τον τρόπο αυτό οι τιμές γίνονται ανεξάρτητες του χρόνου πώλησης. Η νέα τιμή $P_{03/21}$ υπολογίζεται ως εξής:

$$P_{03/21} = P_x * \frac{\text{ZHVI}_{03/21}}{\text{ZHVI}} \tag{5.2}$$

όπου το $\text{ZHVI}_{03/21}$ είναι το ZHVI του συγκεκριμένου ταχυδρομικού κώδικα τον 03/2021 ενώ το ZHVI_x είναι το ZHVI την στιγμή που πωλήθηκε το ακίνητο. Σε περίπτωση που στη χρονοσειρά του συγκεκριμένου ταχυδρομικού κώδικα δεν υπάρχει κάποιο από τα δύο ZHVI τότε υπολογίζεται στην θέση του ένα ZHVI με βάση τη γραμμική προέκταση της υπάρχουσας χρονοσειράς.

Εκτός από τις τιμές, ανεξαρτητοποιούνται από τον χρόνο και τα ίδια τα ZHVIs θέτοντας σαν ZHVI κάθε ακινήτου το $\text{ZHVI}_{03/21}$ (είτε υπάρχει είτε έχει υπολογιστεί μέσω της γραμμικής προέκτασης των υπάρχοντων).

Η ίδια επεξεργασία θα μπορούσε να γίνει και στην στήλη taxAssessedValue όμως οι τιμές της στήλης αυτής έχουν ούτως ή άλλως μεγάλη απόκλιση από την πραγματική τιμή και συνεπώς η επεξεργασία τους δεν προσφέρει ουσιστικά πλεονεκτήματα.

5.2.2 Διόρθωση homeType

Προτού συνδυαστούν και τα υπόλοιπα δεδομένα, έγινε προσαρμογή του πεδίου homeType με τον εξής τρόπο:

- Όσα ακίνητα δεν είχαν homeType TOWNHOUSE ή COOPERATIVE και είχαν πάνω από 5 υπνοδωμάτια, άλλαξαν homeType σε MULTIFAMILY.
- Όσα ακίνητα δεν είχαν homeType TOWNHOUSE ή COOPERATIVE και είχαν πάνω από 4 μπανια, άλλαξαν homeType σε MULTIFAMILY.

Εξετάζοντας τις τιμές της κολώνας homeType προκύπτουν τα εξής:

SINGLE_FAMILY	35870
MULTI_FAMILY	24504
Cooperative	6263
CONDO	5567
TOWNHOUSE	2472
APARTMENT	1735
LOT	661
HOME_TYPE_UNKNOWN	150
MANUFACTURED	26
Other	1

Από τις παραπάνω εγγραφές και αφού εξετάστηκαν ενδεικτικά κολώνες από κάθε κατηγορία, διαγράφηκαν οι εγγραφές που είχαν homeType 'LOT', 'HOME_TYPE_UNKNOWN', 'MANUFACTURED' και 'Other'. Ο λόγος διαγραφής τους ήταν ο μικρός αριθμός εγγραφών με το συγκεκριμένα τύπο και η έλειψη επαρκών χαρακτηριστικών στην περίπτωση του 'LOT'. Επίσης η κατηγορία 'Cooperative' άλλαξε σε 'COOPERATIVE' ώστε να συμβαδίζει με τον τρόπο γραφής των υπολοίπων. Ο αριθμός των εγγραφών πλέον έγινε 76411.

5.2.3 Εισαγωγή walkScores

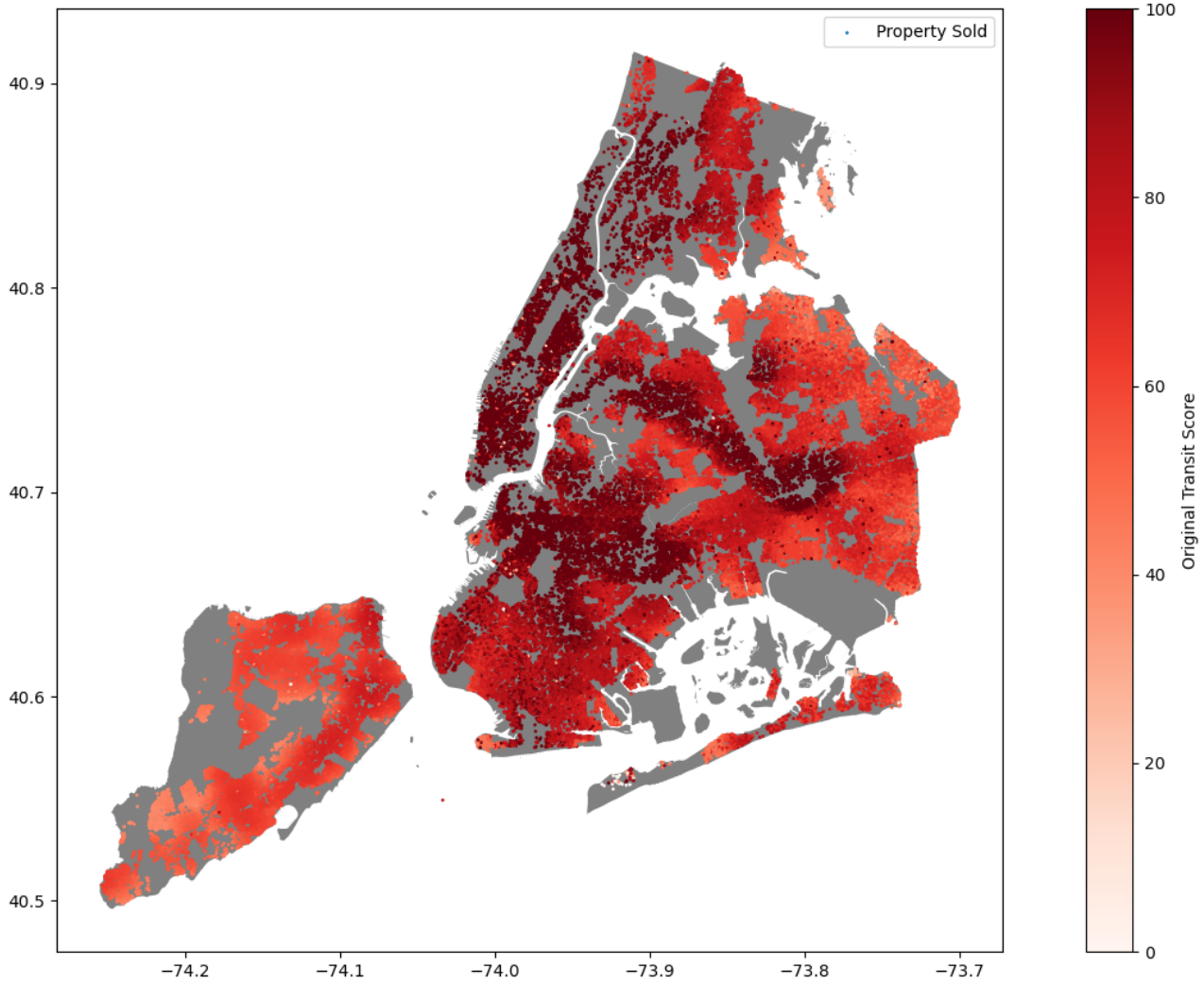
Η ένωση των δεδομένων walkScores με τα δεδομένα ακινήτων έγινε μέσω της κολώνας zpid. Κάθε walkScore entry είχε και ένα zpid από το στάδιο εξαγωγής δεδομένων. Συνεπώς με ένα loop σε όλες τις εγγραφές ακινήτων προστίθενται στην εγγραφή του ακινήτου οι αντίστοιχες εγγραφές των στηλών walkScores. Οι νέες αυτές κολώνες είναι οι:

- walkScore
- transitScore
- bikeScore

Αρχικά, οι μη κενές τιμές στις κολώνες αυτές ήταν:

- Number of walkScores: 73547
- Number of transitScores: 71197
- Number of bikeScores: 38213

Η αρχική εικόνα των δεδομένων έχει ως εξής για το transitScore:



Σχήμα 5.1: Αρχικά transitScores

Παρατηρείται μια αρκετά συνεπής εικόνα, με μικρές ασυνέπειες σε ορισμένα σημεία (π.χ. ένα πολυ χαμηλό σκορ σε μια περιοχή με κατά τα άλλα πολύ υψηλά σκορ).

Με δεδομένο ότι οι εγγραφές συνολικά ήταν 76411, υπάρχε αρκετά ικανοποιητική κάλυψη στα walkScore και transitScore και ελλιπής στα bikeScore. Η κάλυψη και των τριών στηλών μπορεί να βελτιωθεί μέσω κάποιου αλγορίθμου ο οποίος να υπολογίζει το σκορ ενός ακινήτου με βάση το σκορ των πλησιέστερων γειτόνων του.

Αυτό επιτεύχθηκε με τη χρήση της κλάσης BallTree από το πακέτο sklearn.neighbors. Η κλάση αυτή χρησιμοποιείται για αποδοτική εύρεση των πιο κοντινών γειτόνων ενός σημείου. Φορτώθηκαν, επομένως, σε αυτήν οι γεωγραφικές συντεταγμένες και χρησιμοποιώντας την μετρική Haversine για τη μέτρηση αποστάσεων με longitude και latitude, βρέθηκαν για κάθε ακίνητο όλοι του οι γείτονες σε απόσταση έως και 200 μέτρα. Τα 200 επιλέχθηκαν καθώς ακίνητα που απέχουν μέχρι τόση απόσταση μεταξύ τους θα έχουν πολύ παρόμοιες βαθμολογίες για περπάτημα, μέσα συγκοινωνίας και ποδήλατα.

Στη συνέχεια για κάθε ακίνητο υπολογίστηκε η μέση βαθμολογία όλων των κοντινών γειτόνων του για κάθε σκορ (με περιορισμό στο πόσο μπορεί η βαθμολογία του κάθε γείτονα να απέχει από τον μέσο όρο) και η τιμή που υπολογίστηκε ανατέθηκε σαν σκορ στο συγκεκριμένο ακίνητο. Η διαδικασία αυτή ονομάστηκε εξομάλυνση (smoothening) των τιμών λόγω του τρόπου λειτουργίας της και έτρεξε για όλα τα ακίνητα, ακόμα και αυτά που είχαν ήδη σκορ, προκειμένου να διορθωθούν τυχόν λανθασμένες ακραίες τιμές.

Μετά το πρώτο αυτό πέρασμα, η εικόνα των σκορ ήταν η εξής:

- Number of walkScores: 74574
- Number of transitScores: 76376
- Number of bikeScores: 74574

Η εικόνα είναι σαφώς βελτιωμένη, ειδικά για τα σκορ ποδηλάτων. Ωστόσο μπορεί να βελτιωθεί και ακόμα παραπάνω αυξάνοντας τα όριο αναζήτησης του BallTree με την παραδοχή πως οι νέες τιμές που θα καλύψουν τα κενά δεν θα είναι τόσο εύστοχες ενδεχομένως όσο οι υπόλοιπες. Συνεπώς, στη συνέχεια χρησιμοποιήθηκε ένα BallTree με ακτίνα 500 μέτρων ώστε να βρει κοντινούς γείτονες και αυτή τη φορά το σκορ προέκυπτε μόνο από το σκορ του κοντινότερου γείτονα. Η διαδικασία έτρεξε 3 φορές και την τρίτη δεν παρατηρήθηκε καμία αλλαγή, επομένως εκεί σταμάτησε.

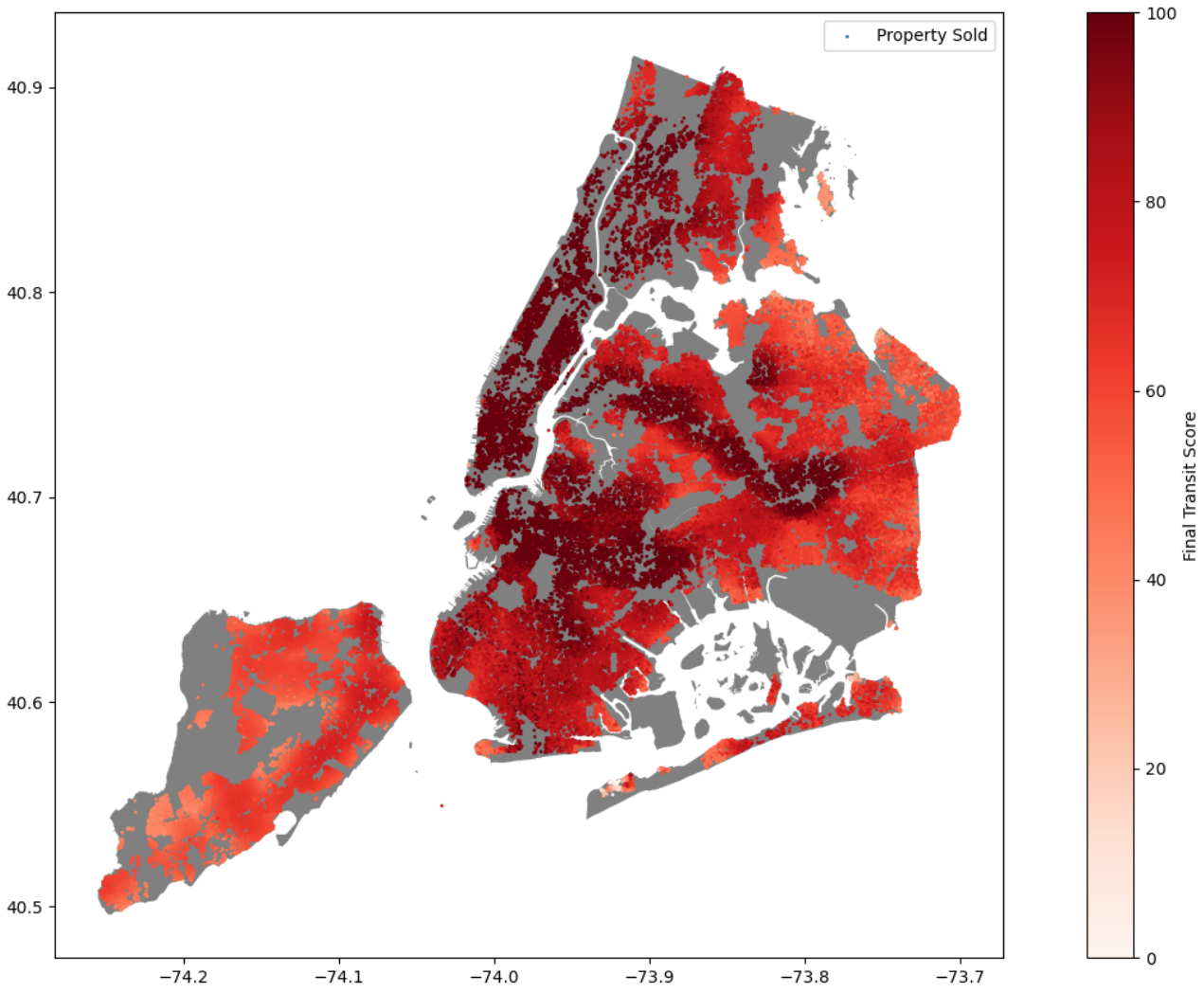
Η τελική εικόνα ήταν:
walkScores length before: 76403
transitScores length before: 76376
bikeScores length before: 74574

Number of properties changed: 1765
Iteration 1 done... 1765 properties changed.
Number of properties changed: 31
Iteration 2 done... 31 properties changed.
Number of properties changed: 0
Iteration 3 done... 0 properties changed.
No properties changed, exiting...

walkScores length after: 76411
transitScores length after: 76408

bikeScores length after: 76366

Ακολούθησε μία τελευταία επανάληψη της αρχικής διαδικασίας smoothening και η εικόνα στο τέλος ήταν η εξής:



Σχήμα 5.2: TransitScores μετά την επεξεργασία

Οι αρχικές ασυνέπειες που παρατηρήθηκαν έχουν πλέον εξαληφθεί και οι τιμές κατά τμήματα είναι ελαφρώς πυκνότερες. Το μόνο πρόβλημα φαίνεται να είναι μία εγγραφή ακινήτου του οποίου οι συντεταγμένες το τοποθετούν πάνω στον ωκεανό. Η ασυνέπεια αυτή αφαιτέθηκε από το dataset.

5.2.4 Εισαγωγή school scores

Για τον υπολογισμό σχολικών σκορ για κάθε ακίνητο χρησιμοποιήθηκε μια παραπλήσια προσσέγγιση με τον υπολογισμό των walkscores. Συγκεκριμένα, δημιουργήθηκαν τρία Ball-

Trees, ένα για κάθε βαθμίδα εκπαίδευσης: Elementary Schools, Middle Schools και High Schools. Στη συνέχεια με τη βοήθεια των BallTrees βρέθηκαν τα κοντινότερα σχολεία μαζί με τις αντίστοιχες αποστάσεις για κάθε ακίνητο. Σαν μέγιστη απόσταση σχολείου ορίστηκαν τα 5 χιλιόμετρα.

Για κάθε ένα σχολείο υπολογίστηκε ο μέσος όρος των σκορ του για τις χρονιές 2016 έως 2019, για όσες δηλαδή είχαν ληφθεί δεδομένα από τον ιστότοπο schooldigger.com. Ο μέσος όρος αυτός αποτέλεσε την βασική βαθμολογία του σχολείου. Στη συνέχεια, για κάθε ακίνητο και για όλα τα σχολεία τα οποία βρίσκονταν σε ακτίνα 5 χιλιομέτρων από αυτό υπολογίστηκε ένα τελικό σχολικό σκορ το οποίο λάμβανε υπόψιν την βαθμολογία του σχολείου αλλά και την απόσταση. Το υψηλότερο από αυτά τα σχολικά σκορ έγινε το σχολικό σκορ του συγκεκριμένου ακινήτου.

Η συνάρτηση υπολογισμού του σχολικού σκορ ήταν η εξής:

```
1  def getSchoolScore(indicesRow, distancesRow,
       schoolMeanScoreList, schoolType):
2      maxSchoolScore = 0
3      for i in range(0, len(indicesRow)):
4          meanScore = schoolMeanScoreList[indicesRow[i]]
5          distanceModifier = getSchoolDistanceModifier(
       distancesRow[i], schoolType)
6          distanceWeightedScore = meanScore * distanceModifier
7          if distanceWeightedScore > maxSchoolScore:
8              maxSchoolScore = distanceWeightedScore
9      return maxSchoolScore
10
11 def getSchoolDistanceModifier(distance, schoolType):
12     if (schoolType == 'elementary'):
13         if distance < 0.5:
14             return 1
15         else:
16             return 1 - (distance / 5) + 0.1
17     else:
18         if distance < 0.5:
19             return 1
20         else:
21             return 1 - (distance / 5) ** 2 + 0.01
```

Listing 5.4: Συνάρτηση για τον υπολογισμό του σχολικού σκορ

Όπως φαίνεται, η συνάρτηση λαμβάνει σαν είσοδο μία λίστα με τους δείκτες των πιο κοντινών σχολείων, μία λίστα με τις αντίστοιχες αποστάσεις κάθε σχολείου από το ακίνητο, μία λίστα με τα μέσα σκορ κάθε σχολείου και το είδος σχολείων που εξετάζονται (η συνάρτηση

τρέχει πρώτα για όλα τα Elementary Schools, μετά για όλα τα Primary Schools και τέλος για όλα τα High Schools).

Ο συντελεστής απόστασης που υπολογίζεται για κάθε σχολείο είναι 1 αν αυτό βρίσκεται έως και 500 μέτρα από το ακίνητο και στη συνέχεια μειώνεται όσο αυξάνει η απόσταση. Στην περίπτωση των Elementary Schools η μείωση του συντελεστή είναι ταχύτερη με την απόσταση προκειμένου να αντικατοπτρίσει την μεγαλύτερη σημασία της πρόσβασης σε κοντινό σχολείο για παιδιά μικρότερων ηλικιών.

Ένα πιθανό πρόβλημα που θα μπορούσε να προκύψει από την παραπάνω προσέγγιση είναι η αγνόηση υγρών συνόρων κατά την αναζήτηση κοντινών σχολείων (για παράδειγμα να αναγνωριστεί ως κοντινό ένα σχολείο το οποίο βρίσκεται στην απέναντι όχθη ενός ποταμού. Ωστόσο, λόγω της δομής της πόλης και της επιλογής του σχετικά μικρού (για τα δεδομένα της Νέας Υόρκης) ορίου των 5 χιλιομέτρων, είναι απίθανο να παρουσιαστεί το παραπάνω πρόβλημα και ακόμα και αν παρουσιαστεί, ο συντελεστής απόστασης κατά πάσα πιθανότητα θα οδηγήσει στην απόρριψή του. Γίνεται συνεπώς η εύλογη παραδοχή πως τέτοιες περιπτώσεις δεν θα εμφανιστούν προκειμένου να μην χρειαστεί μια αρκετά πιο σύνθετη λύση για το συγκεκριμένο πρόβλημα μικρής σημασίας.

Από όλα τα σκορ που υπολογίζονται για κάθε ακίνητο αποθηκεύεται το υψηλότερο. Οι νέες κολώνες που δημιουργήθηκαν για τα School Scores είναι οι:

- elementarySchoolScore
- middleSchoolScore
- highSchoolScore

και η τελική εικόνα των High School scores είναι:

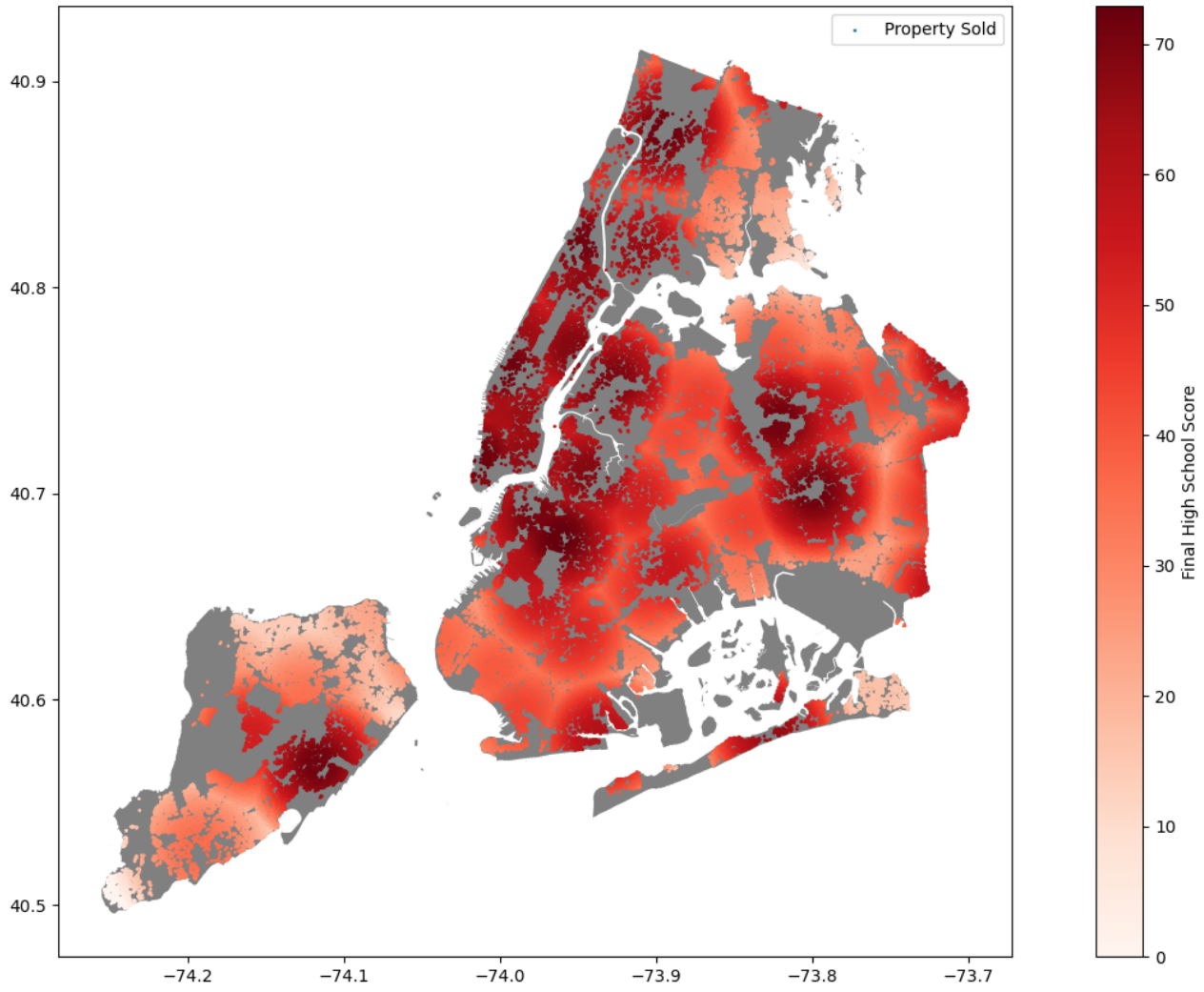


Figure 5.3: Τελικά HighSchoolScores

5.2.5 Εισαγωγή facilities (εγκαταστάσεων)

Ακολουθεί η εισαγωγή των δεδομένων εγκαταστάσεων. Τα δεδομένα αυτά συλλέχθηκαν, όπως έχει ήδη αναφερθεί, από τον ιστότοπο data.cityofnewyork.us και περιέχουν πληροφορίες για δημόσιες και ιδιωτικές εγκαταστάσεις ή αξιοσημείωτες τοποθεσίες στην πόλη της Νέας Υόρκης.

Κάθε εγκατάσταση ανήκει σε ένα συγκεκριμένο είδος και τα διαφορετικά είδη με τον αριθμό εγγραφών από το καθένα είναι τα εξής:

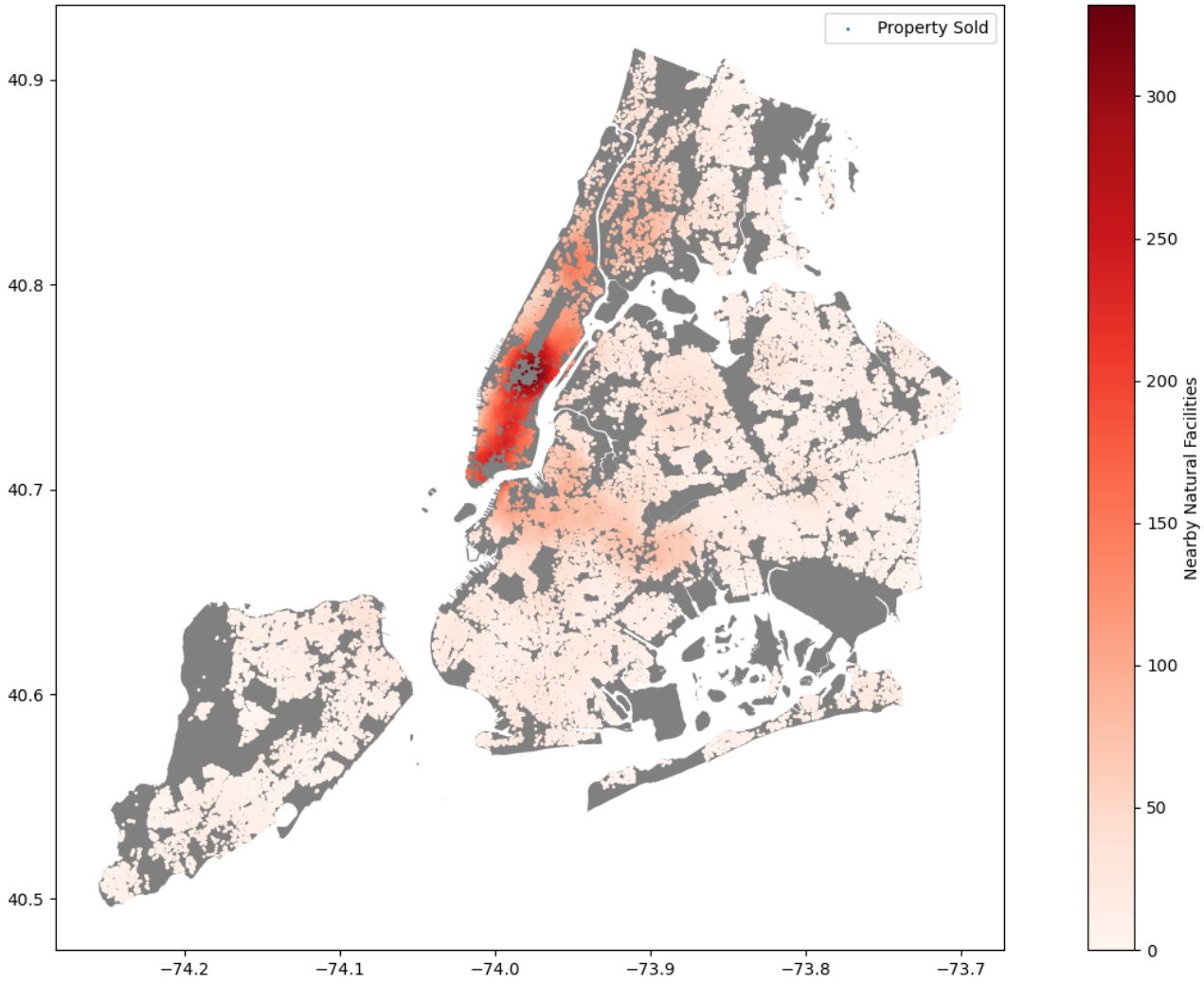
EDUCATION, CHILD WELFARE, AND YOUTH	12519
HEALTH AND HUMAN SERVICES	5914
PARKS, GARDENS, AND HISTORICAL SITES	3476
CORE INFRASTRUCTURE AND TRANSPORTATION	3276
ADMINISTRATION OF GOVERNMENT	3243
LIBRARIES AND CULTURAL PROGRAMS	2055
PUBLIC SAFETY, EMERGENCY SERVICES AND ADMINISTRATION OF JUSTICE	447

Κατά αντιστοιχία, στα προτότυπα δεδομένα ακινήτων προστέθηκαν οι εξής νέες κολώνες:

- educationFacilities
- healthFacilities
- naturalFacilities
- infrastructureFacilities
- administrationFacilities
- culturalFacilities
- publicSafetyFacilities

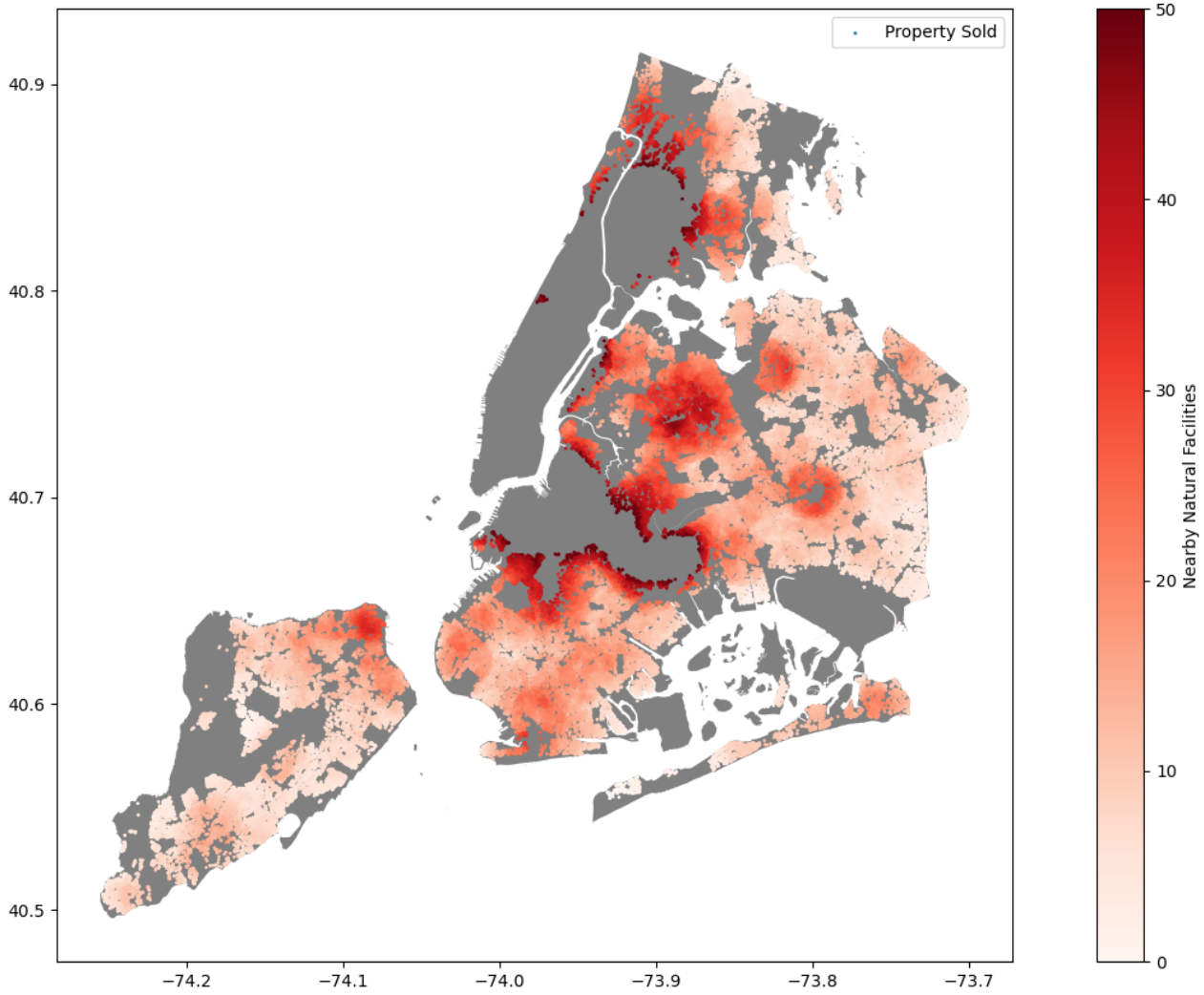
Κάθε κολώνα περιέχει τον αριθμό εγκαταστάσεων του συγκεκριμένου τύπου τα οποία βρίσκονται σε απόσταση ενός χιλιομέτρου από το συγκεκριμένο ακίνητο. Ο τρόπος με τον οποίο γίνεται το δέσιμο των δεδομένων είναι όπως και με τα σχολεία, χρησιμοποιώντας δηλαδή ένα BallTree για κάθε είδος εγκαταστάσεων και εκτελώντας ένα query στο κάθε δέντρο για κάθε ακίνητο.

Τα αποτελέσματα για τα naturalFacilities για παράδειγμα:



Σχήμα 5.4: Όλες οι naturalFacilities εγγραφές

Εκ πρώτης όψεως, τα δεδομένα αυτά φαίνεται να είναι προβληματικά καθώς συγκεντρώνονται κυρίως στην περιοχή του Manhattan. Εξετάζεται επομένως στη συνέχεια μόνο το σύνολο των εγγραφών που έχουν $naturalFacilities <= 50$:



Σχήμα 5.5: naturalFacilities εγγραφές όπου $naturalFacilities <= 50$

Φαίνεται λοιπόν πως, εν τέλει, τα δεδομένα είναι αρκετά εύστοχα, με μεγαλύτερη συγκέντρωση στην περιοχή του Manhattan.

5.2.6 Εισαγωγή complaint data

Τέλος, συνδέονται τα δεδομένα ακινήτων με τα δεδομένα παραπόνων στην πόλη της Νέας Υόρκης.

Τα δεδομένα αυτά κυμαίνονται από λιγότερο σοβαρές κατηγορίες όπως PETIT LARCENY έως πιο σοβαρές όπως MURDER NON-NEGL. MANSLAUGHTER. Προκειμένου να υπάρχει μεγαλύτερη επιρροή στην πρόβλεψη τιμών τα εγκλήματα τα οποία λαμβάνονται υπόψιν από το dataset αυτό είναι όσα ανήκουν στις κατηγορίες ['DANGEROUS DRUGS', 'BURGLARY', 'DANGEROUS WEAPONS', 'SEX CRIMES', 'RAPE', 'ARSON', 'MURDER NON-NEGL. MANSLAUGHTER', 'KIDNAPPING RELATED OFFENSES'].

Στη συνέχεια εφαρμόστηκαν 2 διαφορετικές μέθοδοι για την εισαγωγή των στοιχείων αυτών στα δεδομενα ακινήτων.

Μέθοδος ακτίνας

Με παρόμοιο τρόπο που συσχετίστηκαν τα προηγούμενα δεδομένα, δηλαδή με τη χρήση ενός BallTree, έγινε η σύνδεση κάθε ακινήτου με τον αριθμό εγκλημάτων στην περιοχή γύρω του. Η ακτίνα εντός της οποίας καταμετρήθηκαν τα εγκλήματα για κάθε ακίνητο ήταν 500 μέτρα.

Ακολούθως, ο αριθμός εγκλημάτων κοντά σε κάθε ακίνητο διαιρέθηκε με τον πληθυσμό του ταχυδρομικού κώδικα στον οποίο ανήκε το ακίνητο προκειμένου να εξομαλυνθούν τα δεδομένα και να μην επιβαρυθνούν χωρίς λόγο πολυπληθείς περιοχές.

Έτσι κάθε ακίνητο συσχετίστηκε συγκεκριμένα με τον αριθμό των εγκλημάτων στην γύρω του περιοχή και όχι απλώς με τον δήμο π.χ. στον οποίο ανήκε.

Μέθοδος zipcodes

Η προσέγγιση αυτή ήταν ελαφρώς διαφορετική. Καταρχάς, Για κάθε έγκλημα βρέθηκε το κοντινότερο σε αυτό ακίνητο και από την εγγραφή του ακινήτου βρέθηκε ο ταχυδρομικός κώδικας του εγκλήματος. Στη συνέχεια, έγινε καταμέτρηση των εγκλημάτων ανά ταχυδρομικό κώδικα και σε κάθε εγγραφή ακινήτου μπήκε ο αριθμός εγκλημάτων για τον ταχυδρομικό κώδικα στον οποίο άνηκε το ακίνητο δια τον πληθυσμό στον ταχυδρομικό κώδικα αυτό. Έτσι, κάθε ακίνητο συσχετίστηκε με την εγκληματικότητα στην ευρύτερη περιοχή του και όχι απλά σε μια κοντινή απόσταση γύρω του.

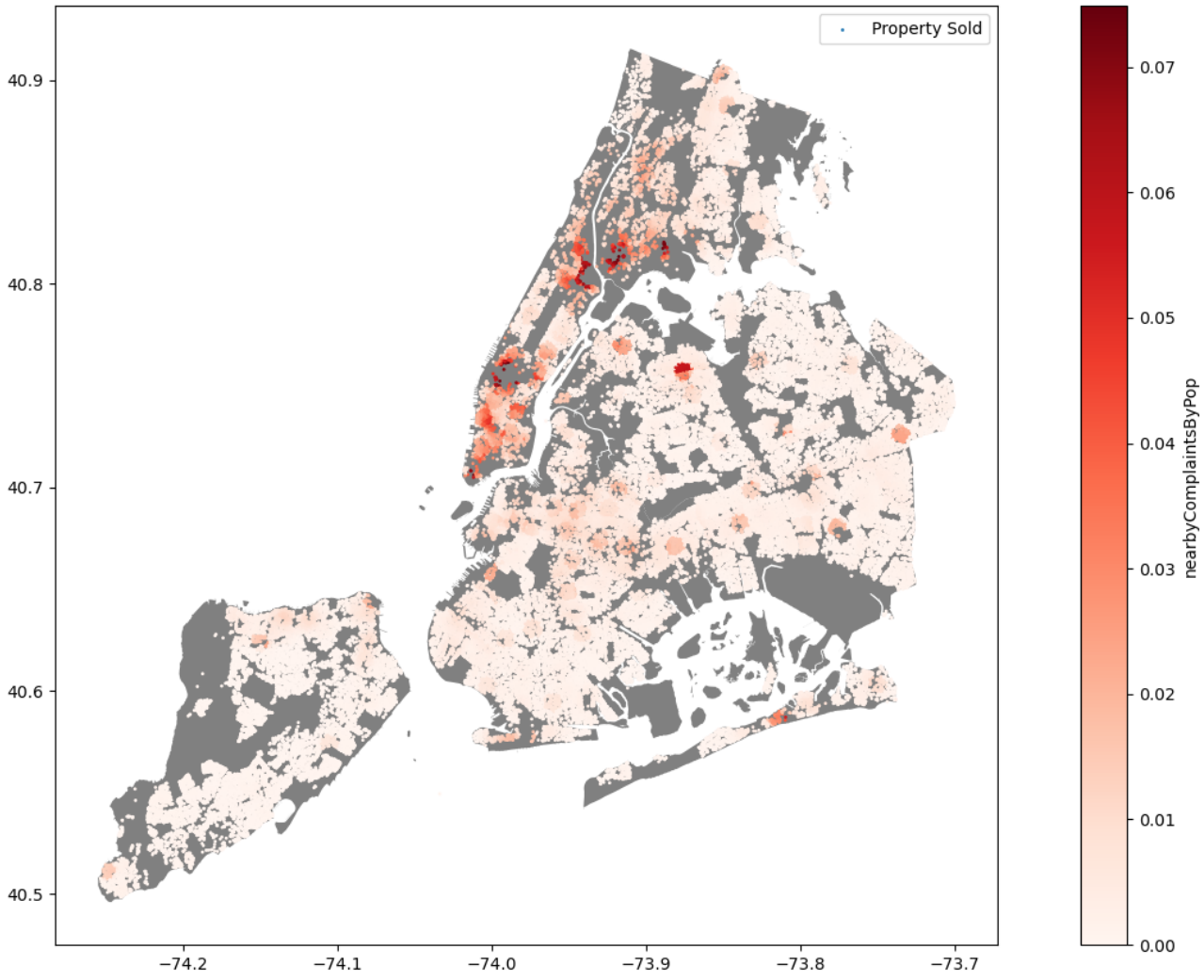
Σύγκριση

Η προσέγγιση μέσω zipcodes έχει το θετικό πως προφυλάσσει το dataset από ακραίες τιμές όπως το να υπάρχει μεγάλος αριθμός από εγκλήματα τα οποία σαν τοποθεσία να έχουν καταγεγραμένο ένα αστυνομικό τμήμα ή μια προκαθορισμένη τιμή αντί της πραγματικής τους τοποθεσίας. Κάτι τέτοιο θα οδηγούσε έναν μικρό αριθμό ακινήτων στο να χαρακτηριστούν άδικα ως σημεία μεγάλης εγκληματικότητας.

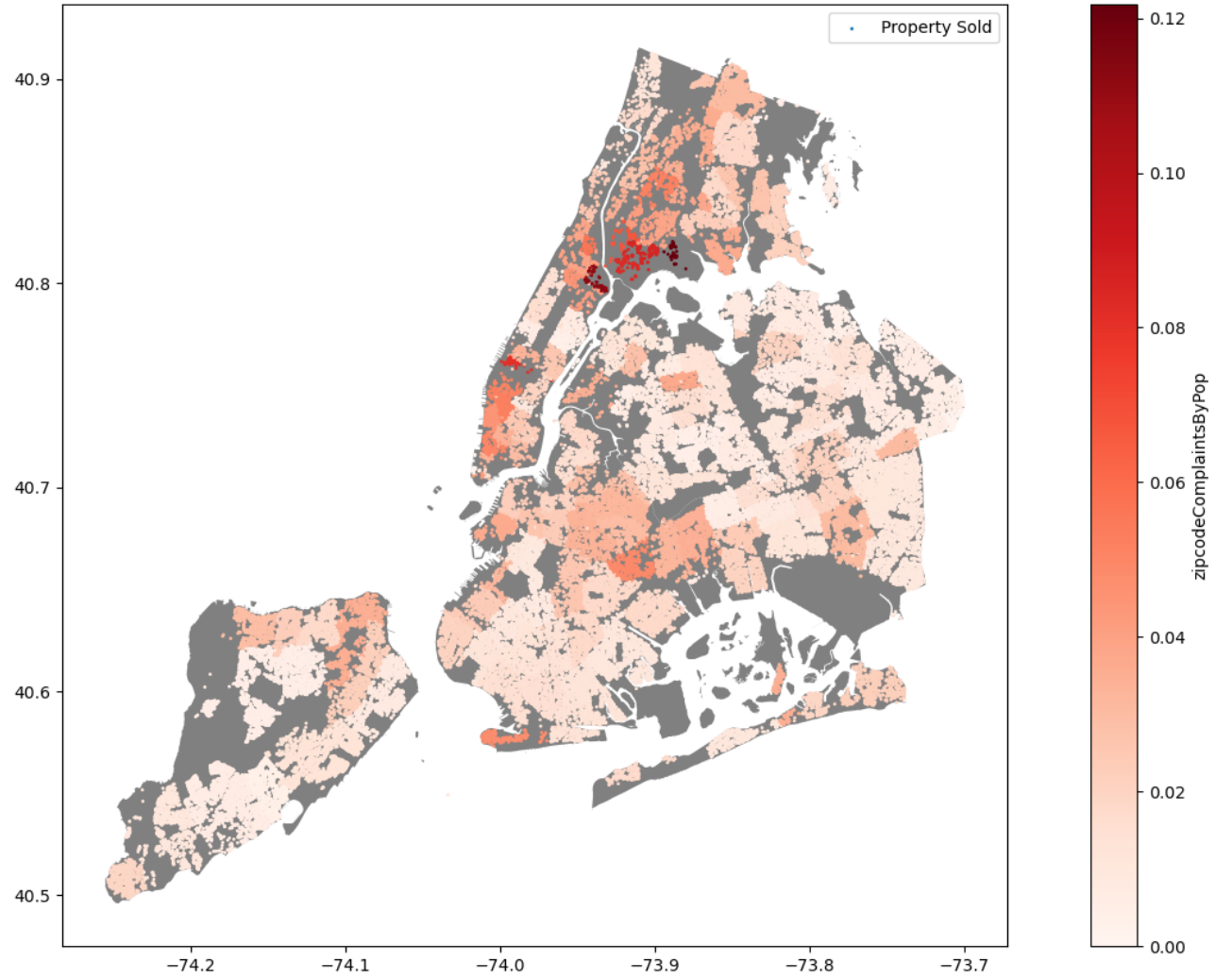
Από την άλλη, με αυτή την προσέγγιση χάνεται η ακρίβεια που παρέχει η μέθοδος της ακτίνας σε περίπτωση που τα δεδομένα είναι σωστά. Προφανώς μία ακτίνα 500 μέτρων είναι πιο ακριβής από τα δεδομένα ενός ολόκληρου δήμου π.χ.

Η εξέταση των δεδομένων εγκληματικότητας αποκάλυψε πως υπήρχαν όντως κάποια hotspots στα οποία ήταν δηλωμένα μακράν περισσότερα εγκλήματα από ότι θα ήταν φυσιολογικό. Τα περισσότερα, ωστόσο, σημεία εμφάνιζαν φυσιολογικούς αριθμούς. Για τον λόγο αυτό αποθηκεύτηκαν δεδομένα και με τις δύο μεθόδους με σκοπό να παρατηρηθεί η χρησιμότητά τους στην πράξη στο στάδιο εκπαίδευσης.

Ακολουθούν τα αποτελέσματα και των δύο μεθόδων αγνοώντας κάποιες ακραίες τιμές για μεγαλύτερη ευκρίνεια.



Σχήμα 5.6: Μέθοδος ακτίνας (500 μέτρα απόσταση)



Σχήμα 5.7: Μέθοδος zipcodes

5.2.7 Χωρισμός ακινήτων ανά κατηγορία

Εφόσον συνεννόθηκαν όλα τα δεδομένα, ακολούθησε ένας διαχωρισμός των δεδομένων με βάση το homeType. Κάθε ομάδα ακινήτων αποθηκεύτηκε σε ξεχωριστό αρχείο, προκειμένου να μπορούν ενδεχομένως να περάσουν από διαφορετική μεταχείριση προτού ξεκινήσει η διαδικασία εκπαίδευσης.

5.3 Επεξεργασία εγγραφών ακινήτων

Αφού οι εγγραφές ακινήτων έχουν εμπλουτιστεί με όλη την πληροφορία που παρουσιάστηκε παραπάνω, ακολουθεί η τελική τους επεξεργασία προτού τροφοδοτήσουν την εκπαίδευση των μοντέλων μηχανικής μάθησης. Ο αρχικός αριθμός εγγραφών είναι 76936.

5.3.1 Επεξεργασία με βάση τα δωμάτια

Καταρχάς, πρέπει να εξεταστούν τα περιεχόμενα των στηλών που αφορούν υπνοδωμάτια και μπανια. Οι κολώνες αυτές αποτελούν κύρια χαρακτηριστικά κάθε ακινήτου και είναι απαραίτητο να έχουν μη κενές και όσο γίνεται ορθές τιμές προκειμένου να εκπαιδευτούν σωστά τα μοντέλα.

Πρώτα, αφαιρούνται όσες εγγραφές έχουν τιμή NaN στην κολώνα bedrooms. Επίσης, αφαιρούνται όσες εγγραφές έχουν homeType διαφορετικό του 'MULTI_FAMILY' και αριθμό υπνοδωματίων μεγαλύτερο του 8 καθώς και όσα έχουν homeType 'MULTI_FAMILY' και αριθμό υπνοδωματίων μεγαλύτερο του 15. Οι τιμές αυτές παραπέμπουν είτε σε λάθος κατηγοριοποίηση του ακινήτου είτε σε εσφαλμένα στοιχεία. Επιπλέον, αφαιρούνται όσες εγγραφές έχουν αριθμό υπνοδωματίων ίσο με 0 και homeType διαφορετικό των ["CONDO", "COOPERATIVE"]. Αυτό γίνεται διότι ορισμένες φορές ένα "CONDO" ή "COOPERATIVE" ακίνητο μπορεί να μην διαθέτει ξεχωριστό υπνοδωμάτιο. Τα υπόλοιπα είδη ακινήτων ωστόσο θα έπρεπε να έχουν τουλάχιστον ένα.

Ακολουθεί η επεξεργασία των μπάνιων. Καταρχάς, η ήδη υπάρχουσα κολώνα bathrooms τροποποιείται με βάση τις τιμές που προκύπτουν από την εξίσωση (5.1) για τα διαθέσιμα είδη μπάνιων. Για παράδειγμα, αν μια εγγραφή διαθέτει 1 halfBathroom και 1 bathroomThreeQuarter, η τελική τιμή της κολώνας bathrooms θα είναι 1.25. Αν οι πληροφορίες ειδών μπάνιων λείπουν, η κολώνα μένει ως έχει. Δημιουργείται, επιπλέον, μία νέα κολώνα με όνομα bathroomsAnySizeCount η οποία περιέχει τον αριθμό των μπάνιων ενός ακινήτου ανεξάρτητα από το μέγεθός τους. Στις περιπτώσεις που δεν έχουν τιμές οι κολώνες για τα είδη μπάνιων, η πληροφορία εξάγεται από την αρχική bathrooms κολώνα. Στο προηγούμενο παράδειγμα η κολώνα bathroomsAnySizeCount θα είχε τιμή 2. Κατά την εκπαίδευση μοντέλων θα εξεταστεί να υπάρχει ουσιαστική διαφορά στην χρησιμότητα των 2 κολώνων.

Ακολούθως, διαγράφονται όσες εγγραφές έχουν στην κολώνα bathrooms αριθμό ο οποίος δεν διαιρείται με το 0.25. Κάτι τέτοιο είναι λάθος, εφόσον το νούμερο πρέπει να προκύπτει ως άθροισμα πολλαπλασίων των αριθμών 0.25, 0.5, 0.75 και 1. Επίσης, διαγράφονται όσες εγγραφές έχουν bathrooms 0. Ένα κατοικίσιμο ακίνητο (με την εξαίρεση όσων έχουν κοινόχρηστα μπάνια, τα οποία όμως ούτως η άλλως αποτελούν outliers) δεν μπορεί να μην έχει κανένα μπάνιο.

Στη συνέχεια, αφαιρούνται όσες εγγραφές έχουν homeType διαφορετικό του 'MULTI_FAMILY' και bathroomsAnySizeCount μεγαλύτερο του 5 καθώς και όσα έχουν homeType 'MULTI_FAMILY' και bathroomsAnySizeCount μεγαλύτερο του 10.

Τέλος, εξετάζεται η ύπαρξη μιας λογικής σχέσης μεταξύ αριθμού υπνοδωματίων και μπάνιων. Συγκεκριμένα, αφαιρούνται όσες εγγραφές πληρούν μία από τις ακόλουθες σχέσεις καθώς παρουσιάζουν δυσανάλογο αριθμό είτε μπάνιων σε σχέση με τα υπνοδωμάτια ή το ανάποδο.

$$bathrooms > 2 * bedrooms \tag{5.3}$$

$$bathroomsAnySizeCount < 3 * bedrooms \tag{5.4}$$

Ο συνολικός αριθμός εγγραφών μετά από τις παραπάνω αφαιρέσεις ήταν 56382 εγγραφές. Αφαιρέθηκε σημαντικός αριθμός εγγραφών αλλά βελτιώθηκε ήδη σημαντικά η ποιότητα των δεδομένων. Ένας λόγος που δεν χρησιμοποιήθηκε κάποια άλλη τεχνική για την αντιμετώπιση των τιμών αυτών είναι το γεγονός πως τα δεδομένα εξακολουθούν να είναι αρκετά μετά την αφαίρεση και το γεγονός πως, μετά από εξέταση ενδεικτικών προβληματικών εγγραφών, παρατηρήθηκε πως έλλειψη ορθών στοιχείων στα bedrooms και bathrooms συνεπάγεται και ελλειπή ή λανθασμένα στοιχεία στις υπόλοιπες κολώνες.

5.3.2 Year built και Year Built Effective

Όσον αφορά τη χρονιά κατασκευής ενός ακινήτου και την ¨εικονική' χρονιά κατασκευής αν ληφθουν υπόψιν ανακαινίσεις, υπάρχουν επίσης κάποιοι περιορισμοί οι οποίοι πρέπει να εφαρμοστούν.

Καταρχάς, αφαιρούνται όσες εγγραφές έχουν ως yearBuilt τιμή μεγαλύτερη του 2021 ή μικρότερη του 1900. Τέτοιες τιμές είτε είναι λανθασμένες είτε αφορούν σπίτια τόσο παλιά που η τιμή τους μπορεί να παρουσιάζει μεγάλες διακυμάνσεις λόγω ιστορικότητας ή παραμέλησης.

Ακολουθεί μια διόρθωση της κολώνας yearBuiltEffective η οποία σε ορισμένες περιπτώσεις έχει τιμή μικρότερη της yearBuilt. Στις περιπτώσεις αυτές η τιμή της yearBuiltEffective ορίζεται ως ίση της yearBuilt καθώς ο ρόλος της είναι να αντικατοπτρίζει την επιρροή αναβαθμίσεων και ανακαινίσεων στην αρχική χρονιά κατασκευής. Μια άλλη προσέγγιση θα ήταν η διαγραφή των αντίστοιχων εγγραφών αλλά εφόσον είναι λίγες σε αριθμό (1294) και η διαφορά στην τελική κολώνα μικρή, προτιμήθηκε η απλή διόρθωσή τους. Ωστόσο, όσες εγγραφές (στην πραγματικότητα μόνο μία) είχαν yearBuiltEffective μεγαλύτερο του 2021 αφαιρέθηκαν.

Τέλος, όσες εγγραφές είχαν κενή τιμή στο yearBuilt αφαιρέθηκαν και όσες είχαν κενή τιμή στο yearBuiltEffective πήραν την ίδια τιμή που είχαν και στο yearBuilt.

Ο τελικός αριθμός εγγραφών που απέμειναν ήταν 51392.

5.3.3 Living Area και Lot Size

Στη συνέχεια, εξετάζονται οι κολώνες livingArea και lotSize. Θα πρέπει να εξετασεί τόσο η αληθοφάνεια των τιμών τους όσο και η συμφωνία των τιμών αυτών με τον αριθμό υπνοδωματίων και μπάνιων.

Καταρχάς, αφαιρούνται όσες εγγραφές έχουν homeType διαφορετικό του 'MULTI_FAMILY' και Living Area μεγαλύτερη του 500 καθώς και όσα έχουν homeType 'MULTI_FAMILY' και Living Area μεγαλύτερη του 1000. Επίσης, αφαιρούνται όλες οι εγγραφές που έχουν Living Area μικρότερη του 20.

Στη συνέχεια, ορίζεται η εξής συνάρτηση η οποία ελέγχει κατά πόσο το livingArea ενός ακινήτου βγάζει νόημα σε σχέση με τα υπνοδωμάτια και τα μπάνια του:

```
1 def livingAreaIsNormal(livingArea, bedrooms, bathrooms):
2     minimum = 10 + bedrooms * 7 + bathrooms * 4
3     maximum = 300 + bedrooms * 25 + bathrooms * 15
```

```
4     return minimum < livingArea < maximum  # also returns
    False for nan values
```

Listing 5.5: Συνάρτηση για τον έλεγχο του livingArea σε σχέση με τα δωμάτια

Εφαρμόζεται, λοιπόν, σε όλες τις εγγραφές ο έλεγχος αυτός και αφαιρούνται όσες δεν τον περνούν. Στη συνέχεια, αφαιρούνται και οι εγγραφές με κενό λιινγΑρεα.

Παρόμοια αντιμετώπιση έχει και η lotSize κολώνα: Όσες εγγραφές έχουν κενό lotSize ή lotSize με τιμή μεγαλύτερη του 1000 ή μικρότερη του 20 αφαιρούνται. Έτσι, οι συνολικές εγγραφές μετά από τις παραπάνω αφαιρέσεις ήταν 41605.

5.3.4 Συμπλήρωση λογικών τιμών

Πολλές από τις κολώνες του dataset έχουν λογικές τιμές True/False. Κατά τη διαδικασία δημιουργίας τους στο κεφάλαιο 5.1.3 οι κολώνες αυτές έλαβαν NaN στις περισσότερες περιπτώσεις όπου το αντίστοιχο πεδίο του resoFacts ήταν κενό.

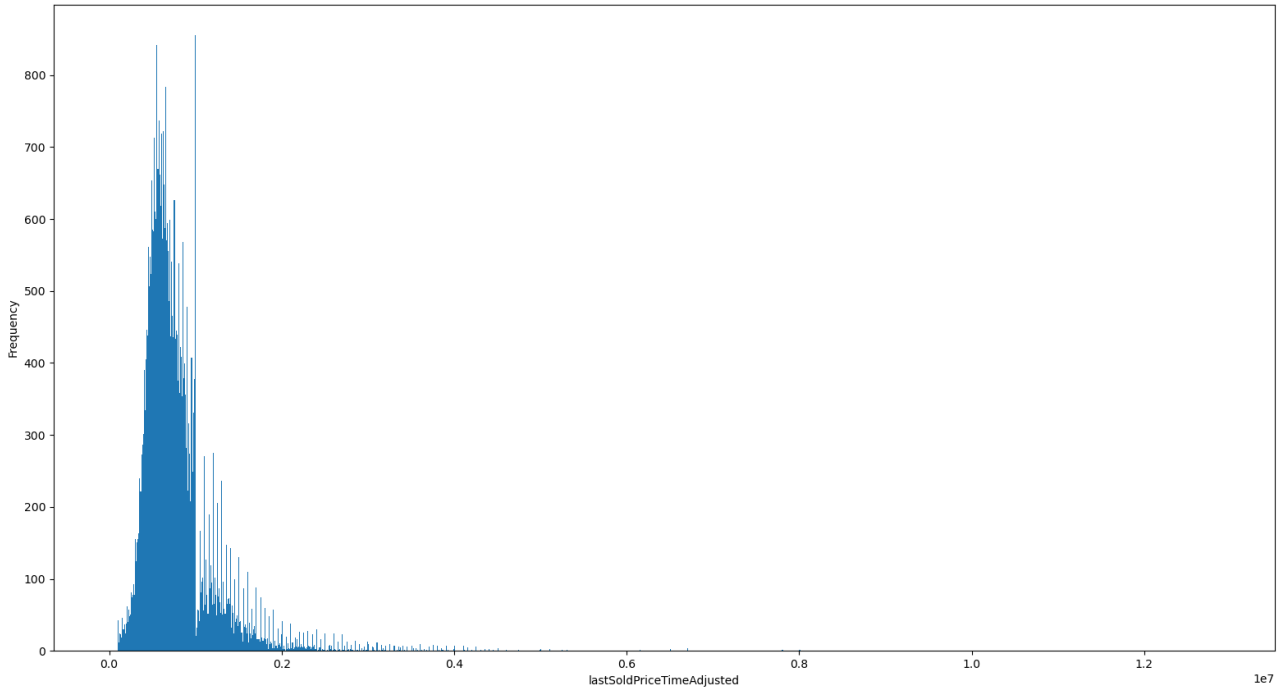
Τα στοιχεία τους ολοκληρώθηκαν ορίζοντας τις NaN τιμές τους ως False. Οι κολώνες οι οποίες συμπληρώθηκαν με αυτόν τον τρόπο είναι οι:

- hasPrivatePool
- hasSpa
- hasFireplace
- hasGarage
- hasParking
- hasFinishedBasement
- hasUnfinishedBasement

5.3.5 Έλεγχος και χειρισμός οικονομικών στηλών

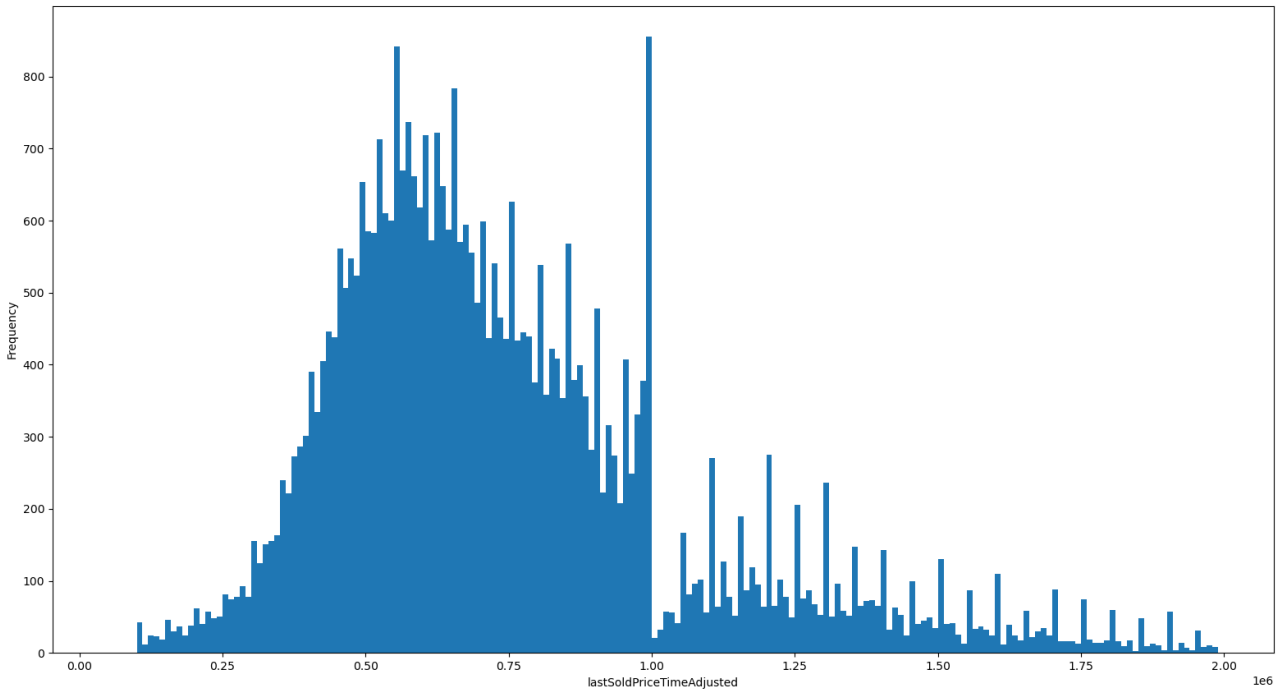
LastSoldPriceTimeAdjusted

Εκτελώντας έναν πρώτο έλεγχο στις τιμές, γίνεται φανερό πως η μεγάλη πλειοψηφία των ακινήτων φαίνεται να βρίσκεται μεταξύ των τιμών 100000 και 2000000 (2 εκατομμύρια).



Σχήμα 5.8: Όλες οι τιμές ακινήτων

Εστιάζοντας στην περιοχή αυτή, παρατηρείται καλύτερα η δομή του ιστογράμματος τιμών.



Σχήμα 5.9: Οι τιμές ακινήτων μεταξύ 50000 και 2000000

Η πλειοψηφία των τιμών είναι περαιτέρω συγκεντρωμένη στο εύρος [250000, 1000000]. Στο διάστημα αυτό προσεγγίζει την κανονική κατανομή με τη διαφορά πως υπάρχει μεγάλη συγκέντρωση τιμών στο σημείου και λίγο κάτω από το ένα εκατομμύριο. Στη συνέχεια, το πλήθος ακινήτων με τιμή πώλησης πάνω από ένα εκατομμύριο φθήνει σταθερά.

Με βάση τις παρατηρήσεις αυτές, διαγράφηκαν τα ακίνητα με τιμές άνω των 2 εκατομμυρίων καθώς η τιμή τους δεν μπορούσε να προβλεφθεί επαρκώς μόνο με τα διαθέσιμα στοιχεία αλλα θα απαιτούσε και στοιχεία όπως εσωτερική διαρρύθμιση, πολυτέλεια, κλπ. Επιπλέον διαγράφηκαν όσα είχαν τιμή κάτω από 100000 το οποίο ήταν εξαρχής το κάτω όριο.

TaxAssessedValue

Το taxAssessedValue προέρχεται από την αποτίμηση ενός ακινήτου για τον υπολογισμό των κατάλληλων φορολογικών συντελεστών. Μια αξιολόγηση λαμβάνει υπόψη τις πωλήσεις παρόμοιων ακινήτων, καθώς και τα ευρήματα επιθεώρησης κατοικίας στον υπολογισμό της. Στα δεδομένα του Zillow το taxAssessedValue συνήθως δεν είναι τόσο κοντά στην πραγματική τιμή ώστε να παίξει μονοσήμαντο ρόλο στην πρόβλεψή της. Είναι ωστόσο ένα από τα πλέον χρήσιμα χαρακτηριστικά που εξετάστηκαν.

Προκειμένου να συμβαδίζουν οι τιμές της κολώνας αυτής με τις τιμές πώλησης, αφαιρέθηκαν όσες εγγραφές είχαν taxAssessedValue μεγαλύτερο από 4000000 (3 εκατομμύρια) ή μικρότερο από 50000.

Πλέον, ο αριθμός εγγραφών ήταν 39176. Όπως ήταν φανερό και από τα ιστογράμματα, υπήρχε ένας μικρός αριθμός από ακίνητα με τιμή άνω των 2 εκατομυρίων. Τα ακίνητα αυτά

ωστόσο θα μπορούσαν να έχουν σημαντικές αρνητικές συνέπειες στην απόδοση των μοντέλων μηχανικής μάθησης.

5.3.6 Λοιπές αλλαγές και επεξεργασία

Καταρχάς, παρατηρήθηκε πως τα περισσότερα ακίνητα για τα οποία υπήρχαν πληροφορίες θέρμανσης και ψύξης ανήκαν στα χαμηλότερα τμήματα τιμών. Αντίθετα, τα πιο ακριβά ακίνητα δεν διέθεταν συχνά πληροφορίες θέρμανσης και ψύξης. Επιπλέον οι διαφορετικές τιμές στις στήλες αυτές φάνηκε να έχουν σχετικά μικρή και ασυνεπή επιρροή στην τιμή πώλησης. Συνεπώς, αφαιρέθηκαν.

Η κολώνα stories επίσης φάνηκε πως δεν είχε επιρροή στην τιμή και είχε επίσης αρκετά δεδομένα τα οποία φαίνονταν ασυνεπή (π.χ. SINGLE_FAMILY ακίνητα με 10 ορόφους). Για τον λόγο αυτό, δεν χρησιμοποιήθηκαν στην τελική εκπαίδευση. Ωστόσο, αφαιρέθηκαν τα ακίνητα που είχαν περισσότερους από 10 ορόφους. Πλέον, ο αριθμός εγγραφών ήταν 39151.

Όσον αφορά το fencing (περίφραξη), δημιουργήθηκα μία νέα κολώνα, hasFencing, η οποία είχε τιμή True εάν η αρχική κολώνα fencing είχε μη κενή τιμή.

Το ίδο και για την κολώνα otherStructures από την οποία προήλθε η κολώνα hasOtherStructures.

Ακολούθησαν οι κολώνες architecturalStyle και structureType. Επειδή οι κολώνες αυτές περιείχαν πληθώρα διαφορετικών τιμών, έγινε προσπάθεια σύμπτυξης των διαφορετικών τιμών σε μεγαλύτερες κατηγορίες οι οποίες αποθηκεύτηκαν στην κολώνα architecture. Η κολώνα εμπλουτίστηκε επίσης με βάση την κολώνα description και σχετικά keywords που αυτή περιείχε. Οι κατηγορίες αυτές με τον αντίστοιχο αριθμό εγγραφών για την καθεμία είναι:

Other	17927
Colonial	11224
Townhouse	2681
Ranch	2191
Modern	1268
Multi Story	1003
Cape	956
Contemporary	889
Tudor	598
Victorian	227
Bungalow	187

Όπως φαίνεται, η μεγάλη πλειοψηφία είναι είτε τύπου 'Other' είτε τύπου 'Colonial'. Η κατηγορία 'Other' συγκεκριμένα προέρχεται από εγγραφές οι οποίες δεν είχαν πληροφορίες αρχιτεκτονικής. Το γεγονός αυτό σημαίνει πως πιθανώς η κολώνα αυτή να μην έχει μεγάλη χρησιμότητα αλλά αυτό θα αναλυθεί στο στάδιο εκπαίδευσης.

Τέλος, αφαιρέθηκαν οι λίγες εναπομείνουσες εγγραφές οι οποίες είχαν τιμή νυλλ στις κολώνες zestimate, bikeScore, nearbyComplaintsByPop και nearbyComplaintsByPop και η

κολώνα hasNaturalView η οποία είχε υπερβολικά λίγες non-False τιμές. Επίσης αφαιρέθηκαν τα ακίνητα κατηγορίας 'COOPERATIVE' καθώς είχαν μείνει μόνο 36.

Ο τελικός αριθμός εγγραφών ήταν 38705, περίπου δηλάδη οι μισές από τις αρχικές.

5.3.7 Αφαίρεση στηλών

Από τις στήλες που απέμειναν αφαιρούνται οι εξής:

- **Οι zipcode, fencing, otherStructures, architecturalStyle, structureType, longitude, latitude, lastSoldPrice, taxAssessedYear, dateSold, description**: Καθώς δεν χρειάζονται πλέον.
- **Οι appliances, stories, exteriorFeatures, patioAndPorchFeatures, constructionMaterials, greenSustainability, roofType**: Καθώς περιέχουν λίγες και με μικρή ποικιλία ή επιρροή τιμές.

5.4 Τελική εικόνα και συσχετίσεις

5.4.1 Ανάλυση με την custom συνάρτηση rstr και heatmap γραμμικών συσχετίσεων

Η τελική εικόνα των δεδομένων μετά το καθάρισμά τους έχει ως εξής:

```
> df.count()
>   zpid                      38741
    bedrooms                  38741
    bathrooms                 38741
    bathroomsAnySizeCount     38741
    hasFinishedBasement       38741
    hasUnfinishedBasement     38741
    hasFireplace              38741
    hasParking                38741
    hasGarage                 38741
    hasPrivatePool            38741
    hasSpa                    38741
    hasFencing                38741
    hasNaturalView            38741
    hasOtherStructures        38741
    hasAttachedProperty       38741
    architecture              38741
    zhvi                      38741
    yearBuilt                 38741
    yearBuiltEffective        38741
```

```
    livingArea                    38741
    homeType                      38741
    lastSoldPriceTimeAdjusted     38741
    zestimate                     38741
    taxAssessedValue              38741
    lotSize                       38741
    url                           38739
    walkScore                     38741
    transitScore                  38741
    bikeScore                     38741
    elementarySchoolScore         38741
    middleSchoolScore             38741
    highSchoolScore               38741
    educationFacilities           38741
    healthFacilities              38741
    naturalFacilities             38741
    infrastructureFacilities      38741
    administrationFacilities      38741
    culturalFacilities            38741
    publicSafetyFacilities        38741
    nearbyComplaintsByPop         38741
    zipcodeComplaintsByPop        38741
    dtype: int64
>
```

Listing 5.6: Τελικές κολώνες

Προτού ξεκινήσει η διαδικασία εκπαίδευσης, χρειάζεται να εξεταστεί η κύρτωση και η λοξότητα των στηλών καθώς και η (γραμμική) τους συσχέτιση με την τιμή πώλησης.

Η κύρτωση είναι ένα στατιστικό μέτρο που καθορίζει πόσο έντονα διαφέρουν οι ουρές μιας κατανομής από τις ουρές μιας κανονικής κατανομής. Με άλλα λόγια, η κύρτωση προσδιορίζει εάν οι ουρές μιας δεδομένης κατανομής περιέχουν ακραίες τιμές.

Η λοξότητα αντιπροσωπεύει την ασυμμετρία σε μια στατιστική κατανομή, στην οποία η καμπύλη φαίνεται παραμορφωμένη ή στραμμένη είτε προς τα αριστερά είτε προς τα δεξιά. Η λοξότητα μπορεί να ποσοτικοποιηθεί για να καθορίσει το βαθμό στον οποίο μια κατανομή διαφέρει από μια κανονική κατανομή.

Για να μετρήσουμε τις παραπάνω ποσότητες, ορίστηκε η παρακάτω συνάρτηση, η οποία υπολογίζει τα μεγέθη αυτά μαζί με άλλες πληροφορίες, όπως κενές τιμές και τα παρουσιάζει σαν ένα Dataframe.

```
def rstr(df, pred=None):
    obs = df.shape[0]
    types = df.dtypes
    counts = df.apply(lambda x: x.count())
    nulls = df.apply(lambda x: x.isnull().sum())
    distincts = df.apply(lambda x: x.unique().shape[0])
    missing_ration = (df.isnull().sum()/ obs) * 100
    skewness = df.skew()
    kurtosis = df.kurt()
    print('Data shape:', df.shape)

    if pred is None:
        cols = ['types', 'counts', 'distincts', 'nulls', '
   missing ration', 'skewness', 'kurtosis']
        str = pd.concat([types, counts, distincts, nulls,
   missing_ration, skewness, kurtosis], axis = 1)

    else:
        corr = df.corr()[pred]
        str = pd.concat([types, counts, distincts, nulls,
   missing_ration, skewness, kurtosis, corr], axis = 1, sort=
   False)
        corr_col = 'corr '  + pred
        cols = ['types', 'counts', 'distincts', 'nulls', '
   missing_ration', 'skewness', 'kurtosis', corr_col ]

    str.columns = cols
    dtypes = str.types.value_counts()
    print('_____________________________\nData types:\n',str.
   types.value_counts())
    print('_____________________________')
    return str
```

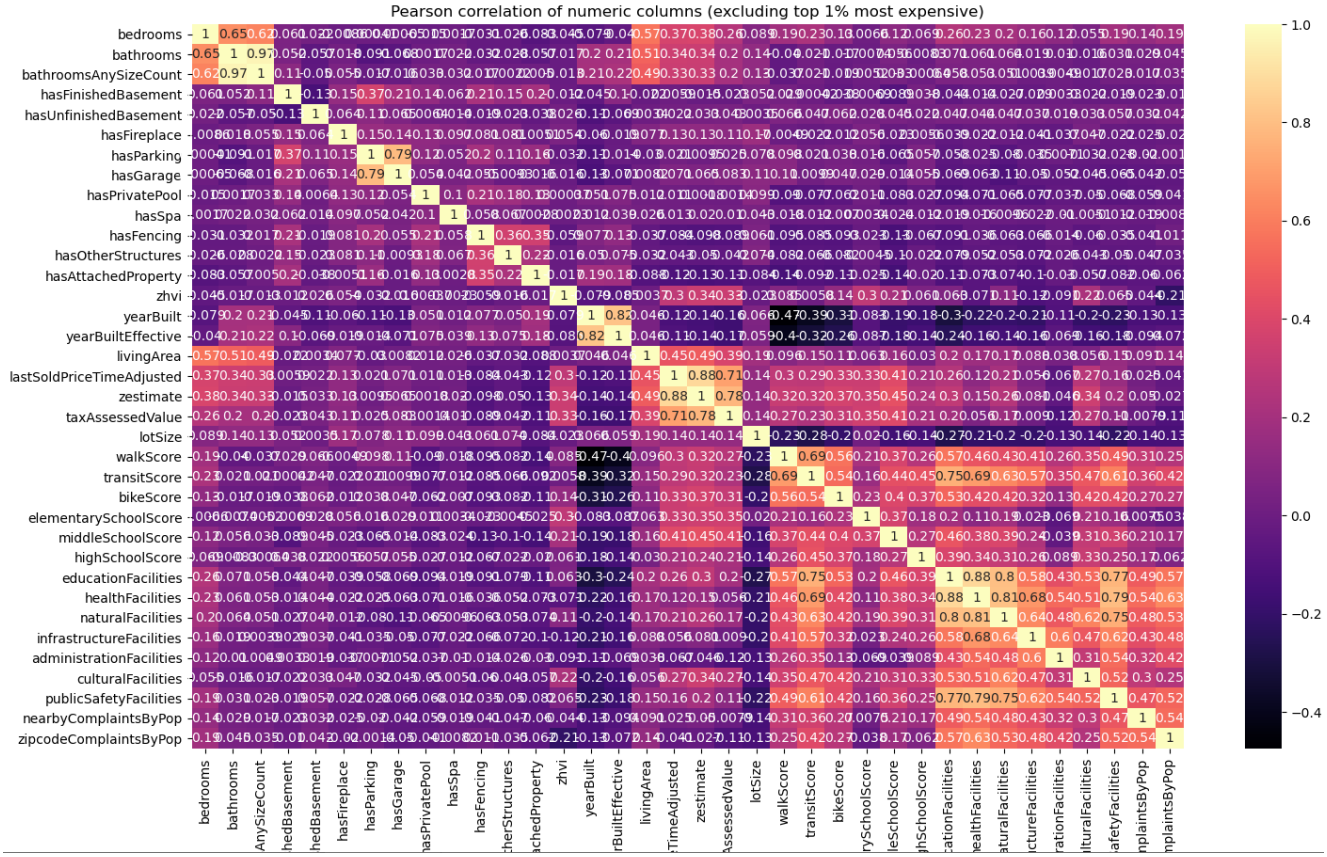
Listing 5.7: Συνάρτηση rstr

Εφαρμόζοντας την συνάρτηση αυτή στα δεδομένα ακινήτων τυπώνεται το εξής αποτέλεσμα:

	types	counts	distincts	nulls	missing_ration	skewness	kurtosis	corr lastSoldPriceTimeAdjusted
lastSoldPriceTimeAdjusted	float64	38741	36051	0	0.000000000000000000	1.095815038510292272	1.367565937905096618	1.000000000000000000
zestimate	float64	38741	37349	0	0.000000000000000000	1.571711604970835952	4.628404924867776771	0.892044773399454094
taxAssessedValue	float64	38741	2752	0	0.000000000000000000	1.819895686451896477	6.842752643936633739	0.722617356828735180
livingArea	float64	38741	3165	0	0.000000000000000000	1.055097278759751456	1.275395891523068315	0.465684942546526148
middleSchoolScore	float64	38741	35201	0	0.000000000000000000	-0.357886460399012840	-0.550936709960074822	0.416923689197171765
bedrooms	float64	38741	14	0	0.000000000000000000	1.234464270649402451	1.948185442217345908	0.370719504208726791
bikeScore	float64	38741	17416	0	0.000000000000000000	0.217851832099421716	0.619494549957131380	0.353789163048790378
elementarySchoolScore	float64	38741	30955	0	0.000000000000000000	-0.221975221561939551	-0.512329427567697060	0.344200974149858341
bathrooms	float64	38741	31	0	0.000000000000000000	1.025471349277887745	1.641600559928043548	0.343759797590718341
bathroomsAnySizeCount	float64	38741	10	0	0.000000000000000000	0.943727273721126642	1.688544239925923840	0.339362934313976572
zhvi	float64	38741	161	0	0.000000000000000000	0.916385211726923665	4.140835921812636045	0.312402321438549846
walkScore	float64	38741	3230	0	0.000000000000000000	-1.186078507169107121	0.970902311331082402	0.304474504054986050
culturalFacilities	int64	38741	123	0	0.000000000000000000	9.376699743982872093	172.256393061058560079	0.303332682878932502
transitScore	float64	38741	3314	0	0.000000000000000000	0.211672036112802581	-0.616461592373751532	0.301749421180595723
educationFacilities	int64	38741	486	0	0.000000000000000000	1.778441820821325070	3.481517737631691034	0.274108989321119434
naturalFacilities	int64	38741	148	0	0.000000000000000000	2.979771005649626492	13.685169603577438835	0.232813106447444351
highSchoolScore	float64	38741	37051	0	0.000000000000000000	0.066201125225751389	-0.448633080691315467	0.224852668604076711
publicSafetyFacilities	int64	38741	28	0	0.000000000000000000	2.225000135729506834	9.162664179099202499	0.176187924021291159
hasFireplace	bool	38741	2	0	0.000000000000000000	3.225550261340056846	8.404608372778650249	0.137524159334803775
lotSize	float64	38741	4382	0	0.000000000000000000	1.697139013104039584	4.018036213812361979	0.136540442714305627
healthFacilities	int64	38741	333	0	0.000000000000000000	2.592446249760403276	9.734771482362393868	0.135062212050681696
infrastructureFacilities	int64	38741	123	0	0.000000000000000000	2.541983101195724792	16.607926060620719966	0.077269673617131135
hasGarage	bool	38741	2	0	0.000000000000000000	0.227427092279388843	-1.948377505143921384	0.058666488332893771
nearbyComplaintsByPop	float64	38741	13440	0	0.000000000000000000	6.285969431416356556	63.433495924291420920	0.034330868377288742
hasUnfinishedBasement	bool	38741	2	0	0.000000000000000000	4.410761215683397296	17.455715648574617660	0.023551664757434299
hasSpa	bool	38741	2	0	0.000000000000000000	14.062392750318586820	195.760996002135783556	0.019610456424566498
hasPrivatePool	bool	38741	2	0	0.000000000000000000	5.701633830894719424	30.510203424943277639	0.014298487465922617
hasParking	bool	38741	2	0	0.000000000000000000	-0.212183137366948132	-1.955079249640545358	0.010161659583719356
zpid	int64	38741	38524	0	0.000000000000000000	30.945621226863703868	1268.969778311949085037	0.007989875042831862
hasFinishedBasement	bool	38741	2	0	0.000000000000000000	1.057710350752232076	-0.881294313301936416	-0.000806156884229941
zipcodeComplaintsByPop	float64	38741	163	0	0.000000000000000000	1.961458013580501536	8.722528632816555216	-0.030835297884756124
hasOtherStructures	bool	38741	2	0	0.000000000000000000	8.166871071580185770	64.701123285753027403	-0.044855899669469693
administrationFacilities	int64	38741	129	0	0.000000000000000000	1.780246074182360250	6.172071833817882158	-0.056187726413754781
hasFencing	bool	38741	2	0	0.000000000000000000	3.933401730007300312	13.472344675319515517	-0.086423794836818557
yearBuiltEffective	float64	38741	122	0	0.000000000000000000	0.469226167653963189	-1.012487055763915178	-0.114725555332811535
hasAttachedProperty	bool	38741	2	0	0.000000000000000000	5.457279745173470253	27.783336526207214945	-0.115461322920271420
yearBuilt	float64	38741	122	0	0.000000000000000000	0.580973127202873306	-0.725426354668106566	-0.131945614430609481
hasNaturalView	bool	38741	1	0	0.000000000000000000	0.000000000000000000	0.000000000000000000	NaN
architecture	object	38741	11	0	0.000000000000000000	NaN	NaN	NaN
homeType	object	38741	6	0	0.000000000000000000	NaN	NaN	NaN

Σχήμα 5.10: Αποτελέσματα rstr

Η γραμμική συσχέτιση μεταξύ όλων των δεδομένων παρουσιάζεται και στον παρακάτω πίνανα, οπού πιο ανοιχτά χρώματα ισοδυναμούν με θετική γραμμική συσχέτιση, ενώ σκούρα με αρνητική. Είναι εμφανές ότι πληροφορίες όπως τα διάφορα είδη εγκαταστάσεων έχουν υψηλή συσχέτιση μεταξύ τους.



Σχήμα 5.11: Correlation Heatmap

Προκειμένου να θεωρηθεί αποδεκτή η τιμή της λοξότητας θα πρέπει να βρίσκεται στο διάστημα [-2, 2], ενώ για την τιμή της κύρτωσης το διάστημα είναι το [-7, 7], όπως προτείνουν οι Hair et al. [35].

Οι κολώνες που παραβιάζουν τη συνθήκη της λοξότητας είναι οι culturalFacilities, naturalFacilities, publicSafetyFacilities, hasFireplace, healthFacilities, infrastructureFacilities, nearbyComplaintsByPop, hasUnfinishedBasement, hasSpa, hasPrivatePool, zpid (το οποίο δεν θα χρησιμοποιηθεί ούτως ή άλλως), hasOtherStructures, hasFencing και hasAttachedProperty. Οι κολώνες αυτές είτε περιέχουν πληροφορία από τις εγκαταστάσεις της νέας Υόρκης είτε είναι τύπου Boolean και περιγράφουν μία ούτως ή άλλως ασύμμετρη τιμή όπως την ύπαρξη σπα.

Οι κολώνες που παραβιάζουν τη συνθήκη της κύρτωσης είναι οι culturalFacilities, naturalFacilities, publicSafetyFacilities, hasFireplace, healthFacilities, infrastructureFacilities, nearbyComplaintsByPop, hasUnfinishedBasement, hasSpa, hasPrivatePool, zpid (το οποίο δεν θα χρησιμοποιηθεί ούτως ή άλλως), hasOtherStructures, hasFencing και hasAttachedProperty. Είναι ακριβώς οι ίδιες με αυτές που παραβιάζουν τη συνθήκη της λοξότητας.

5.4.2 Μετασχηματισμοί

Ορισμένα μοντέλα μηχανικής μάθησης, όπως τα τυχαία δάση, είναι πιο ανθεκτικά σε μη ισορροπημένες κατανομές. Ωστόσο, προκειμένου να δοθούν σε κάθε μοντέλο οι ίδιες είσοδοι και να έχουν όσο γίνεται πιο ιδανική κατανομή, θα εφαρμοστούν σε όσες από τις παραπάνω κολώνες δύναται (δηλαδή όχι στις Boolean) μετασχηματισμός για την βελτίωση της κύρτωσης και της λοξότητας.

Ο μετασχηματισμός εφαρμόζεται, λοιπόν, στις κολώνες culturalFacilities, naturalFacilities, publicSafetyFacilities, healthFacilities, infrastructureFacilities και nearbyComplaintsByPop. Εφαρμόζεται επιπλέον και στην κολώνα taxAssessedValue, παρόλο που βρίσκεται εντός των ορίων λόγω της σημασίας της. Μία μέθοδος μετασχηματισμού θα ήταν η χρήση του λογαριθμικού μετασχηματισμού. Ωστόσο, οι κολώνες αυτές περιέχουν ορισμένες μηδενικές τιμές και οι τιμές της nearbyComplaintsByPop περιέχει κυρίως τιμές μικρότερες του 1. Συνεπώς, ο μετασχηματισμός που θα χρησιμοποιηθεί είναι ο μετασχηματισμός τετραγωνικής ρίζας, ο οποίος προϋποθέτει μονάχα μη αρνητικές τιμές, γεγονός που ισχύει στις κολώνες αυτές.

Πράγματι, εξετάζοντας τις νέες κολώνες που προκύπτουν από τους μετασχηματισμούς παρατηρείται αισθητή βελτίωση στην κύρτωση και την λοξότητα, ακόμα και στην γραμμική συσχέτιση με την τιμή πώλησης σε κάποιες περιπτώσεις. Η culturalFacilitiesSqrt είναι η μόνη κολώνα η οποία έχει ακόμα πρόβλημα στην κύρτωση αλλά θα χρησιμοποιηθεί ως έχει μιας και βρίσκεται κοντά στα όρια και έχει βελτιωθεί αισθητά σε σχέση με την εικόνα της πρωτότυπης.

	types	counts	distincts	nulls	missing_ration	skewness	kurtosis	corr lastSoldPriceTimeAdjusted
lastSoldPriceTimeAdjusted	float64	38741	36051	0	0.0	1.095815038510292272	1.367565937905096618	1.000000000000000000
taxAssessedValueSqrt	float64	38741	2752	0	0.0	0.809176776809697684	1.807396929596396973	0.724024249318020230
taxAssessedValue	float64	38741	2752	0	0.0	1.819895686451896477	6.842752643936633739	0.722617356828735180
culturalFacilitiesSqrt	float64	38741	123	0	0.0	1.830899621320580817	9.891328477362893778	0.375715415081977921
culturalFacilities	int64	38741	123	0	0.0	9.376699743982872093	172.256393061058560079	0.303332682878932502
naturalFacilitiesSqrt	float64	38741	148	0	0.0	1.548768199632044373	3.058848682054884804	0.268574116009770325
naturalFacilities	int64	38741	148	0	0.0	2.979771005649626492	13.685169603577438835	0.232813106447444351
publicSafetyFacilities	int64	38741	28	0	0.0	2.225000135729506834	9.162664179099202499	0.176187924021291159
publicSafetyFacilitiesSqrt	float64	38741	28	0	0.0	0.185022123138483263	1.048520919756485981	0.172142748513297666
healthFacilitiesSqrt	float64	38741	333	0	0.0	0.862301658530676529	0.921495237321185545	0.161288629274861539
healthFacilities	int64	38741	333	0	0.0	2.592446249760403276	9.734771482362393868	0.135062212050681696
infrastructureFacilitiesSqrt	float64	38741	123	0	0.0	0.604548119380383886	0.444233335600132229	0.103781844050665747
infrastructureFacilities	int64	38741	123	0	0.0	2.541983101195724792	16.607926060620719966	0.077269673617131135
nearbyComplaintsByPopSqrt	float64	38741	13440	0	0.0	2.097402549460920085	7.654048474538551083	0.060315955641887867
nearbyComplaintsByPop	float64	38741	13440	0	0.0	6.285969431416356556	63.433495924291420920	0.034330868377288742

Σχήμα 5.12: Αποτελέσματα rstr για τις μετασχηματισμένες κολώνες

Άλλα είδη μετασχηματισμών όπως feature scaling και one-hot encoding τα οποία είναι απαραίτητα για τα περισσότερα μοντέλα θα παρουσιαστούν στο κεφάλαιο 6 καθώς αποτελούν τμήμα των pipelines τα οποία εκπαιδεύουν τα μοντέλα.

Μετά και από αυτούς τους μετασχηματισμούς, τα δεδομένα είναι έτοιμα να δοθούν σαν είσοδοι στα διάφορα μοντέλα μηχανικής μάθησης που θα δοκιμαστούν στη συνέχεια.

Κεφάλαιο 6

Εκπαίδευση μοντέλων και αξιολόγηση

6.1 Μεθοδολογία

Η μεθοδολογία που ακολουθήθηκε κατά την εκπαίδευση των υποψήφιων μοντέλων μηχανικής μάθησης είναι η εξής: Καταρχάς, ένα υποσύνολο των δεδομένων απομακρύνθηκε και φυλάχτηκε ώστε να λειτουργήσει σαν test set, δηλαδή τελικό έλεγχο της ευστοχίας των τελικών μοντέλων.

Στη συνέχεια, έγινε ένα δοκιμαστικό run για κάθε μοντέλο προκειμένου να κριθεί η γενική του επίδοση. Ακολούθησε hyperparameter tuning για κάθε μοντέλο προκειμένου να βρεθούν οι βέλτιστες υπερπαράμετροι με τις οποίες θα εκπαιδευτεί τελικά πάνω στα δεδομένα.

Αφού εκπαιδευτούν όλα τα μοντέλα, επιλέγονται τα πιο αποτελεσματικά και συγκρίνεται η απόδοσή τους στο τελικό test set το οποίο έχει φυλαχτεί μέχρι τη στιγμή εκείνη για τον σκοπό αυτό.

6.2 Δημιουργία test set

6.2.1 Ανάγκη δημιουργίας test set

Η δημιουργία ενός test set το οποίο θα μείνει κρυφό ως το τέλος πριν από οποιαδήποτε άλλη δράση στην εκπαίδευση μοντέλων είναι σημαντική για δύο λόγους.

Ο πρώτος είναι ο προφανής λόγος πως τα μοντέλα θα χρειαστεί να αξιολογηθούν από τον τρόπο με τον οποίο μπορούν να γενικεύσουν πάνω σε πραγματικά, άγνωστα κατά την εκπαίδευσή τους δεδομένα. Με τον τρόπο αυτό κρίνεται κατά πόσο εμφανίζεται το σύνηθες πρόβλημα του overfitting, το οποίο συμβαίνει όταν ένα μοντέλο μαθαίνει να αποδίδει πολύ καλά στο training set συγκεκριμένα και χάνει την δυνατότητα γενίκευσης σε νέα δεδομένα.

Ο δεύτερος λόγος είναι πως το ανθρώπινο μυαλό σχηματίζει και αυτό μοτίβα με μεγάλη ευκολία και η αξιολόγηση ενός μοντέλου πάνω στο test set προτού ολοκληρωθεί η διαδικασία εκπαίδευσης μπορεί να οδηγήσει στην απόφαση τροποποίησης του μοντέλου απλά και μόνο

για να αποδίδει και στο test set. Αυτό μπορεί να γίνει ακόμα και ακούσια, με τη σκέψη πως το test set αποκάλυψε κάποια σημαντική λεπτομέρεια των δεδομένων η οποία δεν ήταν σαφής από το training set. Έτσι, όμως, χάνεται το νόημα του test set, το οποίο είναι ο τελικός έλεγχος της απόδοσης των μοντέλων σε πραγματικά νέα δεδομένα. Όταν το μοντέλο βγει σε παραγωγικό περιβάλλον, η απόδοσή του θα είναι χαμηλότερη από αυτήν στο test set. Το φαινόμενο αυτό ονομάζεται data snooping bias [36].

6.2.2 Μέθοδος δημιουργίας test set

Η πιο απλή μέθοδος δημιουργίας του test set θα ήταν αυτό να δημιουργείται τυχαία κάθε φορά που τρέχει το pipeline. Ωστόσο αυτό θα σήμαινει πως το test set θα έχει κάθε φορά διαφορετικά δεδομένα και ενδεχομένως να καλύψει κάποια στιγμή ακόμα και ολόκληρο το training set. Αυτό θα καθιστούσε το test set άχρηστο στην σχεδόν βέβαιη περίπτωση που το pipeline τρέξει πολλές φορές.

Ένας τρόπος αντιμετώπισης του παραπάνω προβλήματος θα ήταν να αποθηκευτεί μία φορά το test set κατά το πρώτο τρέξιμο του pipeline ή να χρησιμοποιηθεί ένα σταθερό random seed για την κατασκευή του. Ωστόσο, οι δύο αυτές προσεγγίσεις υποφέρουν από το πρόβλημα πως η εισαγωγή νέων δεδομένων θα τις καθιστούσε άκυρες.

Η καλύτερη λύση είναι η χρήση ενός μηχανισμού που με βάση το αναγνωριστικό κάθε εγγραφής θα κρίνει κάθε φορά ντετερμινιστικά αν η εγγραφή αυτή θα ανήκει στο test set, αλλά η επιλογή θα είναι φαινομενικά τυχαία ανάλογα με κάποια ιδιότητα του αναγνωριστικού, ανεξάρτητη της τιμής πώλησης. Απαιτείται δηλαδή μία εγγραφή με συγκεκριμένο αναγνωριστικό να αντιμετωπίζεται πάντα με τον ίδιο τρόπο, άσχετα με το πόσες εγγραφές υπάρχουν, αλλά το αν μια νέα εγγραφή θα μπει στο test set να είναι τυχαίο.

Ένας τρόπος με τον οποίο μπορεί να επιτευχθεί αυτό είναι η χρήση του hash του αναγνωριστικού κάθε εγγραφής, του zpid δηλαδή. Αν το hash αυτό βρίσκεται κάτω από ένα επιλεγμένο όριο, τότε η εγγραφή εισάγεται στο test set. Η συνάρτηση splitTrainTestById που εκτελεί την εργασία αυτή είναι η εξής:

```
1  def testSetCheck(identifier, testRatio):
2      return crc32(np.int64(identifier)) & 0xffffffff <
       testRatio * 2**32
3
4  def splitTrainTestById(data, testRatio, idColumn):
5      ids = data[idColumn]
6      inTestSet = ids.apply(lambda id_: testSetCheck(id_,
       testRatio))
7      return data.loc[~inTestSet], data.loc[inTestSet]
8
9  dfTrainSingle, dfTestSingle = splitTrainTestById(df[df.
       homeType == 'SINGLE_FAMILY'], 0.1, 'zpid')
10 dfTrainMulti, dfTestMulti = splitTrainTestById(df[df.homeType
```

```
    == 'MULTI_FAMILY'], 0.1, 'zpid')
dfTrainCondo, dfTestCondo = splitTrainTestById(df[df.homeType
    == 'CONDO'], 0.1, 'zpid')
dfTrainTownhouse, dfTestTownhouse = splitTrainTestById(df[df.
    homeType == 'TOWNHOUSE'], 0.1, 'zpid')
dfTrainApartment, dfTestApartment = splitTrainTestById(df[df.
    homeType == 'APARTMENT'], 0.1, 'zpid')

dfTrain = dfTrainSingle.append([dfTrainMulti, dfTrainCondo,
    dfTrainTownhouse, dfTrainApartment]).sample(frac=1)
dfTest = dfTestSingle.append([dfTestMulti, dfTestCondo,
    dfTestTownhouse, dfTestApartment]).sample(frac=1)
```

Listing 6.1: Συνάρτηση splitTrainTestById

Επίσης, πρέπει το test set να ανταποκρίνεται επακριβώς και στα ποσοστά διαφορετικών κατηγοριών ακινήτων του dataset (και να μην έχει για παράδειγμα 5% περισσότερα 'SINGLE_FAMILY' ακίνητα από ότι θα έπρεπε). Για τον λόγο αυτό η συνάρτηση αυτή τρέχει για κάθε κατηγορία ακινήτων ξεχωριστά και τα αποτελέσματα συνεννώνονται στο τέλος.

Τα ποσοστά κάθε κατηγορίας στο αρχικό dataset ήταν τα εξής:

SINGLE_FAMILY	0.537 %
MULTI_FAMILY	0.356 %
CONDO	0.047 %
TOWNHOUSE	0.036 %
APARTMENT	0.024 %

Στο test set τα ποσοστά ήταν τα εξής:

SINGLE_FAMILY	0.550 %
MULTI_FAMILY	0.344 %
CONDO	0.047 %
TOWNHOUSE	0.035 %
APARTMENT	0.024 %

Πέρα από μία απόκλιση περίπου 1% μεταξύ των κατηγοριών 'SINGLE_FAMILY' και 'MULTI_FAMILY', παρατηρείται πως τα ποσοστά του test set αντιπροσωπεύουν εύστοχα τα ποσοστά του αρχικού dataset. Το ίδιο συμβαίνει και με τις τιμές οι οποίες για το αρχικό dataset είναι:

5th percentile	3.771e+05
25th percentile	5.574e+05
50th percentile	7.100e+05
75th percentile	9.245e+05
95th percentile	1.433e+06

ενώ για το test set είναι:

5th percentile	3.708e+05
25th percentile	5.633e+05
50th percentile	7.129e+05
75th percentile	9.222e+05
95th percentile	1.422e+06

Οι τιμές βρίσκονται πολύ κοντά και συνεπώς, φαίνεται πως το test set έχει κατασκευαστεί σωστά.

6.3 Μετασχηματισμοί για συμβατότητα με μοντέλα μηχανικής μάθησης

Η βιβλιοθήκη scikit-learn προσφέρει ορισμένες κλάσεις οι οποίες καθιστούν σημαντικά ευκολότερη την επεξεργασία στηλών οι οποίες έχουν κατηγορικά δεδομένα ή δεδομένα σε μορφή string. Επιπλέον, διαθέτει κλάσεις για το αυτόματο scaling των κολώνων ώστε μία κολώνα με πολύ μεγάλες τιμές να μην επισκιάσει τις υπόλοιπες που έχουν μικρότερες τιμές.

Οι μετασχηματισμοί αυτοί δεν απαιτείται να εφαρμοστούν σε όλα τα μοντέλα (για παράδειγμα τα τυχαία δάση λειτουργούν καλά και χωρίς feature scaling), ωστόσο, προκειμένου να υπάρχει ομοιομορφία στις εισόδους κάθε μοντέλου θα χρησιμοποιηθούν για όλα τα μοντέλα.

6.3.1 One-Hot Encoding

Το one-hot encoding είναι μία τεχνική κωδικοποίησης η οποία χρησιμοποιείται στην μηχανική μάθηση για κατηγορικές μεταβλητές. Επιτρέπει στις μεταβλητές να κωδικοποιηθούν με τρόπο ο οποίος δεν υποννοεί την ύπαρξη κάποιας διάταξης στις τιμές των μεταβλητών. Για παράδειγμα, εάν οι τιμές μιας κολώνας ήταν 'red', 'green', 'yellow' το πιθανότερο είναι πως δεν υπάρχει κάποια σχέση διάταξης μεταξύ τους ώστε να κωδικοποιηθούν ως 1, 2, 3.

Ο τρόπος με τον οποίο κωδικοποιεί το one-hot encoding μια μεταβλητή είναι μέσω της κατασκευής ενός αραιού ή πυκνού πίνακα (ανάλογα με τους περιορισμούς μνήμης) στον οποίο κάθε πιθανή τιμή της κατηγορικής μεταβλητής έχει και από μία ξεχωριστή κολώνα. Έτσι κάθε εγγραφή έχει τιμή 1 σε μία από τις παραγώμενες αυτές κολώνες και 0 στις υπόλοιπες, επιτρέποντας την εύκολη, μολονότι κοστοβόρα σε μνήμη, επεξεργασία από ένα μοντέλο μηχανικής μάθησης.

Στα παρόντα δεδομένα οι κατηγορικές μεταβλητές είναι μόνο 2 ('architecture', 'homeType') και έχουν μικρό εύρος τιμών, επομένως το one-hot encoding αποτελεί εύκολη λύση στο πρόβλημα της αναπαράστασής τους. Για την κωδικοποίηση χρησιμοποιείται ο OneHotEncoder της βιβλιοθήκης scikit-learn.

6.3.2 RobustScaler

Ένα άλλο πρόβλημα που μπορεί να προκύψει σε προβλήματα μηχανικής μάθησης είναι όταν είσοδοι με μεγάλο εύρος τιμών επισκιάζουν μεταβλητές με μικρότερο εύρος. Προκειμένου να αποφευχθεί αυτό, τα αριθμητικά δεδομένα περνούν από μια διαδικασία κλιμάκωσης ώστε να έχουν όλα παρόμοια εύρη.

Υπάρχουν διάφοροι τρόποι να γίνει η διαδικασία αυτή. Μία από τις πλέον χρήσιμες κλάσεις για αυτό τον σκοπό είναι η κλάση RobustScaler της βιβλιοθήκης scikit-learn. Κύριο χαρακτηριστικό της είναι η ανθεκτικότητα σε ακραίες τιμές.

Λειτουργεί καταργώντας τη διάμεση τιμή και κλιμακώνοντας τα δεδομένα σύμφωνα με το εύρος μεταξύ του 1ου τεταρτημορίου (25ο quantile) και του 3ου τεταρτημορίου (75ο quantile).

Η τυποποίηση ενός συνόλου δεδομένων είναι μια κοινή απαίτηση για πολλούς εκτιμητές μηχανικής μάθησης. Συνήθως αυτό γίνεται με την αφαίρεση του μέσου όρου και την κλιμάκωση σε διακύμανση μονάδας. Ωστόσο, οι ακραίες τιμές μπορούν συχνά να επηρεάσουν τον μέσο όρο/διακύμανση του δείγματος με αρνητικό τρόπο. Σε τέτοιες περιπτώσεις, η διάμεσος και το εύρος μεταξύ των παραπάνω quantiles συχνά δίνουν καλύτερα αποτελέσματα.

6.3.3 ColumnTransformer

Προκειμένου να εφαρμοστούν οι παραπάνω μετασχηματισμοί στα δεδομένα με ευκολία και ταχύτητα, γίνεται χρήση της κλάσης ColumnTransformer της βιβλιοθήκης scikit-learn. Η κλάση αυτή επιτρέπει την αλυσιδωτή εφαρμογή μετασχηματιστών σε συγκεκριμένες κολώνες ενός dataset.

Με τη χρήση της εφαρμόζονται οι παραπάνω μετασχηματισμοί στις συμβατές κολώνες του training set.

6.4 Εκπαίδευση μοντέλων μηχανικής μάθησης

Στα πλαίσια της εργασίας αυτής δοκιμάστηκαν 4 διαφορετικά μοντέλα στα δεδομένα. Αυτά είναι:

- Ένα Multi-Layer Perceptron με χρήση του Keras framework
- Ένας RandomForestRegressor με χρήση της βιβλιοθήκης scikit-learn
- Ένας GradientBoostingRegressor με χρήση της βιβλιοθήκης scikit-learn
- Ένας SGDRegressor με χρήση της βιβλιοθήκης scikit-learn

Τα μοντέλα δοκιμάστηκαν σε 3 περιπτώσεις:

- Στην πρώτη περίπτωση, τα δεδομένα με τα οποία τροφοδοτήθηκαν ήταν τα αρχικά δεδομένα τα οποία προήλθαν από όλη την παραπάνω προεπεξεργασία, χωρίς το χαρακτηριστικό zestimate.

- Στην δεύτερη περίπτωση χρησιμοποιήθηκαν μόνο δεδομένα με μέγιστη τιμή τα 1000000 (1 εκατομμύριο), πάλι χωρίς το χαρακτηριστικό zestimate. Αυτό έγινε προκειμένου να εξεταστεί η επίδραση των λίγων και μεγάλων τιμών στα αποτελέσματα των μοντέλων.
- Στην τρίτη περίπτωση, στα αρχικά δεδομένα (τιμή μέχρι 2 εκατομύρια) δεν αφαιρέθηκε από τα χαρακτηριστικά το zestimate, επομένως αξιολογήθηκε κατά πόσο ένα μοντέλο το οποίο έκανε χρήση του zestimate θα μπορούσε να φέρει καλύτερα αποτελέσματα από αυτό.

Να σημειωθεί πως αντί της δεύτερης προσέγγισης δοκιμάστηκε και η χρήση λογαριθμικού μετασχηματισμού και μετασχηματισμού τετραγωνικής ρίζας στην τιμή πώλησης. Ωστόσο, οι μετασχηματισμοί αυτοί δεν οδήγησαν σε καλύτερα αποτελέσματα από την πρωτότυπη τιμή.

Για την σύγκριση και την εύρεση βέλτιστων υπερπαραμέτρων κάθε μοντέλου χρησιμοποιήθηκε 5-Fold Cross-Validation. Δηλαδή, κάθε φορά που έτρεχε ένα μοντέλο πραγματοποιούνταν 5 επαναλήψεις. Σε κάθε επανάληψη 20% του training set χρησιμοποιούνταν σαν προσωρινό test set για τη βαθμολόγηση του μοντέλου και το υπόλοιπο 80% σαν προσωρινό training set. Σε κάθε επανάληψη το προσωρινό test set είχε ένα διαφορετικό 20% των δεδομένων, με αποτέλεσμα να καλυφθούν όλα τα δεδομένα.

6.4.1 Multi-Layer Perceptron

Με την πρώτη δοκιμαστική εκπαίδευση, το Multi-Layer Perceptron φάνηκε να αποδίδει μέτρια σε σχέση με τα υπόλοιπα μοντέλα.

Η τελική δομή του ήταν η εξής: Τρία επίπεδα συνολικά (αγνοώντας τις εισόδους). Ένα πυκνό επίπεδο με 100 μέλη και συνάρτηση ενεργοποίησης relu, ένα πυκνό επίπεδο με 200 μέλη και συνάρτηση ενεργοποίησης relu και ένα πυκνό επίπεδο με 100 μέλη και συνάρτηση ενεργοποίησης relu πάλι. Διαφορετικοί συνδυασμοί επιπέδων (και ορισμένων συναρτήσεων ενεργοποίησης) έφεραν παρόμοια αποτελέσματα και επηρέασαν κυρίως τον χρόνο εκπαίδευσης και όχι το αποτέλεσμα. Ο optimizer που κατέληξε να χρησιμοποιείται είναι ο 'nadam'.

Αξίζει να σημειθεί πως από όλα τα μοντέλα το MLP ήταν το πιο αργό στην εκπαίδευση. Η μέτρια ευστοχία του οφείλεται στην έλλειψη πολύ μεγάλου πλήθους δεδομένων το οποίο χρειάζεται προκειμένου να παράξει ισχυρά αποτελέσματα.

Λόγω της μέτριας απόδοσης του μοντέλου δεν εκτελέστηκε ανάλυση σημαντικότητας στα χαρακτηριστικά της εισόδου.

6.4.2 Random Forest

Με την πρώτη δοκιμαστική εκπαίδευση, το Random Forest φάνηκε να αποδίδει καλά σε σχέση με τα υπόλοιπα μοντέλα. Αυτό μαζί με τον Gradient Boosting Regressor ήταν τα δύο καλύτερα μοντέλα με πολύ παρόμοια ευστοχία.

Η τελική δομή του ήταν η εξής:

- max_depth=None
- max_features='sqrt'
- min_samples_leaf=1
- n_jobs=-1 ώστε να λειτουργεί γρηγορότερα το μοντέλο εκμεταλλευόμενο όλους τους πυρήνες του επεξεργαστή - αυτή η παράμετρος δεν επηρεάζει τα αποτελέσματα.
- min_samples_split=10
- ccp_alpha=0.25
- n_estimators=2000

Ο χρόνος εκπαίδευσης ήταν πολύ ικανοποιητικός, ειδικά αν ληφθεί υπόψιν και η ευστοχία των αποτελεσμάτων.

Σημαντικότητα χαρακτηριστικών στο Random Forest

Η σημαντικότητα των διαφορετικών χαρακτηριστικών στο Random Forest φαίνεται παρακάτω, όπου κολώνες με x0 προήλθαν από το one-hot encoding της κολώνας architecture και κολώνες με x1 προήλθαν από το one-hot encoding της κολώνας homeType.

0.29114676441093057	taxAssessedValueSqrt
0.09007854385275153	livingArea
0.0715989680361062	zhvi
0.056012989368186286	middleSchoolScore
0.03869574509873067	elementarySchoolScore
0.03641420658729957	bedrooms
0.03425361274350991	walkScore
0.03335376478949192	culturalFacilitiesSqrt
0.0289432370036775	bathrooms
0.026544774861582736	bikeScore
0.02569803245855101	bathroomsAnySizeCount
0.025677851994360807	lotSize
0.02469953227600176	highSchoolScore
0.019469074776962553	educationFacilities
0.01925994448045422	transitScore
0.019249114368827983	zipcodeComplaintsByPop
0.01678050613149535	nearbyComplaintsByPopSqrt
0.01650862518863192	infrastructureFacilitiesSqrt
0.016249347619067458	naturalFacilitiesSqrt
0.0157279867948339	healthFacilitiesSqrt
0.015625361623352943	administrationFacilities
0.01428296932678628	x1_MULTI_FAMILY
0.013520816212747425	yearBuilt
0.012029367900234582	yearBuiltEffective
0.0075279709076112545	publicSafetyFacilitiesSqrt
0.005957379199118539	x1_SINGLE_FAMILY
0.00295705812042639	x1_CONDO
0.002738233999435128	hasFireplace
0.0020692927662030783	hasGarage
0.0017350668542940118	x0_Townhouse
0.0016297088558675469	x1_APARTMENT
0.0016243403418622882	hasParking
0.0014380087527407013	x0_Other
0.0012684942204278858	x0_Colonial
0.0010936684992723439	hasFinishedBasement
0.0010493953141740271	x0_Modern
0.001043719590672091	x0_Contemporary
0.0010064860128126425	hasUnfinishedBasement
0.000830656687958427	x1_TOWNHOUSE
0.000742758156881989	x0_Ranch
0.0006408745248468774	hasPrivatePool
continues below...	

continued...	
0.0005853684108325024	x0_Multi Story
0.0004581115243968169	x0_Victorian
0.00035906048530693713	x0_Tudor
0.00033306671672688143	x0_Bungalow
0.00029153186449264136	hasFencing
0.0002863973359808021	hasSpa
0.0002587388742358288	x0_Cape
0.0001945658542578383	hasAttachedProperty
5.8908224589265756e-05	hasOtherStructures

Από τα παραπάνω φαίνεται πως η κολώνα taxAssessedValueSqrt παίζει μακράν τον σημαντικότερο ρόλο στην πρόβλεψη τιμής, όπως είναι αναμενόμενο άλλωστε εφόσον αποτελεί και η ίδια μία χοντρική πρόβλεψη της τιμής του ακινήτου για φορολογικούς λόγους, ενώ ορισμένα χαρακτηριστικά, συμπεριλαμβανομένων και των περισσότερων one-hot encoded κατηγορικών έχουν επιρροή μικρότερη του 1%.

Για τον λόγο αυτό αφαιρέθηκαν τα κατηγορικά χαρακτηριστικά και όσα χαρακτηριστικά είχαν σημαντικότητα μικρότερη του 1% και ξαναδοκιμάστηκε το τυχαίο δάσος. Η απόδοσή του ήταν ελάχιστη καλύτερη από την αρχική, γεγονός που δεν προκαλέι έκλπηξη. Ορισμένα χαρακτηριστικά δεν περιέχουν αρκετά ουσιαστική πληροφορία ώστε να συνεισφέρουν στην πρόβλεψη της τιμής και μπορούν ακόμα και να προσθέσουν ανεπιθύμητο θόρυβο στην πρόβλεψη.

6.4.3 Gradient Boosting Regressor

Με την πρώτη δοκιμαστική εκπαίδευση, ο Gradient Boosting Regressor φάνηκε να αποδίδει καλά σε σχέση με τα υπόλοιπα μοντέλα. Αυτός μαζί με το Random Forest ήταν τα δύο καλύτερα μοντέλα με πολύ παρόμοια ευστοχία και γρήγορους χρόνους εκπαίδευσης.

Η τελική δομή του ήταν η εξής:

- loss='huber'
- alpha=0.9
- n_estimators=500
- criterion='friedman_mse'
- max_features='sqrt'
- learning_rate=0.1
- min_samples_split=50
- min_samples_leaf=10

- max_depth=5
- ccp_alpha=0.0

Ο χρόνος εκπαίδευσης ήταν επίσης πολύ ικανοποιητικός, ειδικά αν ληφθεί υπόψιν και η ευστοχία των αποτελεσμάτων που ήταν πολύ παρόμοια με αυτήν του Random Forest. Το γεγονός αυτό είναι αναμενόμενο εφόσον τα δύο μοντέλα χρησιμοποιούν παρόμοιες αρχές για τη λειτουργία τους.

Σημαντικότητα χαρακτηριστικών στον Gradient Boosting Regressor

Η σημαντικότητα των διαφορετικών χαρακτηριστικών στον Gradient Boosting Regressor φαίνεται παρακάτω, όπου κολώνες με x0 προήλθαν, όπως αναφέρθηκε ήδη, από το one-hot encoding της κολώνας architecture και κολώνες με x1 προήλθαν από το one-hot encoding της κολώνας homeType.

0.30152443492450104	taxAssessedValueSqrt
0.11172921547241696	livingArea
0.09518525172431724	zhvi
0.08381080811333523	middleSchoolScore
0.05731549945342917	bedrooms
0.04574600887659494	elementarySchoolScore
0.045638387494575446	bathrooms
0.026382688204488386	bathroomsAnySizeCount
0.02499841028455932	culturalFacilitiesSqrt
0.0208809131751486	bikeScore
0.02045840372576275	walkScore
0.01838489466177803	highSchoolScore
0.017879821827265902	lotSize
0.0170720730482667	transitScore
0.015336089723006735	zipcodeComplaintsByPop
0.015169934577517103	x1_MULTI_FAMILY
0.012147107127783023	infrastructureFacilitiesSqrt
0.00929224185472878	nearbyComplaintsByPopSqrt
0.008940621876472665	yearBuilt
0.008450558536952169	administrationFacilities
0.007583278940778325	x1_SINGLE_FAMILY
0.00616139051541396	educationFacilities
0.005364917017672152	yearBuiltEffective
0.0043996136859152995	healthFacilitiesSqrt
0.004093439831000513	x1_CONDO
0.0038136320716779526	hasFireplace
0.003316794642329618	naturalFacilitiesSqrt
0.0022040219832267388	x1_APARTMENT
0.001461640932942171	hasGarage
0.0013604554773765773	publicSafetyFacilitiesSqrt
0.0007079433394904149	x0_Townhouse
0.0005700509278862521	x0_Bungalow
0.0005127526099977169	hasPrivatePool
0.0004791616956706611	x0_Contemporary
0.00029454367532309413	x1_TOWNHOUSE
0.00019049565030538915	x0_Other
0.00019033319976360452	x0_Colonial
0.00018925065517805126	hasParking
0.0001836729250961707	hasUnfinishedBasement
0.00015025219833514312	x0_Ranch
0.0001414111818856168	x0_Modern
0.00013284288330678488	x0_Cape
continues below...	

continued...	
6.511331609212524e-05	hasFinishedBasement
6.311973007028452e-05	x0_Victorian
1.1477825025473866e-05	x0_Multi Story
8.174748653838894e-06	hasFencing
6.85365668577791e-06	x0_Tudor
0.0	hasSpa
0.0	hasOtherStructures
0.0	hasAttachedProperty

Από τα παραπάνω φαίνεται πως η κολώνα taxAssessedValueSqrt παίζει ξανά πρωταγωνιστικό ρόλο, ενώ πλέον υπάρχουν και χαρακτηριστικά τα οποία δεν προσφέρουν απολύτως τίποτα στο μοντέλο.

Για τον λόγο αυτό αφαιρέθηκαν και από αυτό το μοντέλο τα κατηγορικά χαρακτηριστικά και όσα χαρακτηριστικά είχαν σημαντικότητα μικρότερη του 1% εκτός από το yearBuilt και ξαναδοκιμάστηκε το μοντέλο. Η απόδοσή του ήταν παρόμοια με την αρχική, όπως ήταν αναμενόμενο.

6.4.4 Stochastic Gradient Descent

Το μοντέλο Stochastic Gradient Descent είχε την χειρότερη απόδοση από τα τέσσερα μοντέλα. Παρόλο που ο χρόνος εκπαίδευσής του ήταν πολύ μικρός, αυτό δεν θα μπορούσε να δικαιολογήσει τη χρήση του στο συγκεκριμένο πρόβλημα.

Η τελική δομή του ήταν η εξής:

- max_iter=500
- loss='squared_epsilon_insensitive'
- penalty='l1'
- alpha=0.0001
- tol=1e-3
- epsilon=0.01
- learning_rate='invscaling'

Λόγω της κακής απόδοσης του μοντέλου δεν εκτελέστηκε ανάλυση σημαντικότητας στα χαρακτηριστικά της εισόδου.

6.5 Σύγκριση αποτελεσμάτων μοντέλων μηχανικής μάθησης

Παρακάτω παρουσιάζονται τα αποτελέσματα από τα cross-validation runs των 4 μοντέλων. Η μετρική που επιλέχθηκε για την αξιολόγησή τους ήταν το Root Mean Square Error (RMSE)

καθώς έχει τις ίδιες μονάδες μέτρησης με τις αρχικές τιμές αλλά τιμωρεί περισσότερο τις μεγάλες αποκλίσεις από το απλό απόλυτο σφάλμα. Η εξισωση που δίνει το RMSE είναι η εξής:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_{real} - y_{pred})^2}{n}} \tag{6.1}$$

Μαζί με τα αποτελέσματα των μοντέλων παρουσιάζεται και το zestimate το οποίο αποτελεί και το ουσιαστικό benchmark για το πόσο καλή απόδοση έχουν τα μοντέλα της εργασίας σε σχέση με ένα εξελιγμένο, industrial grade μοντέλο.

6.5.1 Σύγκριση αποτελεσμάτων για το αρχικό dataset

Σύγκριση αποτελεσμάτων cross-validation για το αρχικό dataset

Καταρχάς, παρουσιάζεται η ευστοχία των διαφορετικών μοντέλων που εκπαιδεύτηκαν και του Zestimate στα ακίνητα του training set. Η ευστοχία των μοντέλων προέρχεται από τον μέσο όρο της ευστοχίας τους στις 5 επαναλήψεις του cross-validation, ενώ η ευστοχία του Zestimate υπολογίζεται επί όλου του training set.

Μοντέλο	**RMSE**
Zestimate	152541
Multi-Layered Perceptron	182498
Random Forest	172242
Gradient Boosting Regressor	173647
Stochastic Gradient Descent	191480

Όπως έχει ήδη αναφερθεί τα δύο ισχυρότερα μοντέλα είναι τα Random Forest και Gradient Boosting Regressor. Ακόμα και αυτά ωστόσο υστερούν του Zestimate.

Σύγκριση αποτελεσμάτων test set για το αρχικό dataset

Μετά από όλες τις παραπάνω ενέργειες και αφού είχε ολοκληρωθεί το fine-tuning των υπερπαραμέτρων των μοντέλων, έφτασε η στιγμή του ελέγχου πάνω στο test set. Παρακάτω παρουσιάζονται τα αποτελέσματα των προβλέψεων των μοντέλων για τις εγγραφές του test set αφού πρώτα εκπαιδεύτηκαν από το μηδέν πάνω στο training set.

Μοντέλο	**RMSE**
Zestimate	150651
Multi-Layered Perceptron	179284
Random Forest	172166
Gradient Boosting Regressor	175937
Stochastic Gradient Descent	189995

Παρατηρείται πως το Zestimate εξακολουθεί να αποδίδει πολύ καλά και στο test set όπως ήταν αναμενόμενο. Τα μοντέλα που εκπαιδεύτηκαν επίσης παρουσιάζουν συμπεριφορά παρόμοια με την συμπεριφορά στο cross-validation. Αυτό σημαίνει πως το test set δημιουργήθηκε σωστά και αντικατοπτρίζει πράγματι την εικόνα του training set και πως τα μοντέλα απέφυγαν σε εξαιρετικά ικανοποιητικό βαθμό τον κίνδυνο του overfitting.

6.5.2 Σύγκριση αποτελεσμάτων για τιμές μικρότερες του 1 εκατομμυρίου

Σύγκριση αποτελεσμάτων cross-validation για τιμές μικρότερες του 1 εκατομμυρίου

Όπως και προηγουμένως, παρουσιάζεται πρώτα η ευστοχία των διαφορετικών μοντέλων που εκπαιδεύτηκαν και του Zestimate στα ακίνητα του training set.

Μοντέλο	**RMSE**
Zestimate	142035
Multi-Layered Perceptron	133251
Random Forest	119709
Gradient Boosting Regressor	120641
Stochastic Gradient Descent	132890

Όπως έχει ήδη αναφερθεί τα δύο ισχυρότερα μοντέλα είναι τα Random Forest και Gradient Boosting Regressor. Μεγάλο ενδιαφέρον παρουσιάζει το γεγονός πως για το μικρότερο αυτό εύρος τα μοντέλα που εκπαιδεύτηκαν ξεπερνούν την ακρίβεια του Zestimate. Αυτό κατά πάσα πιθανότητα συμβαίνει διότι το Zestimate χρειάζεται να παραμένει εύστοχο σε πολύ μεγαλύτερο εύρος τιμών και σε πολύ μεγαλύτερη γεωγραφική κλίμακα από ότι τα μοντέλα της εργασίας. Συνεπώς, πιθανώς να θυσιάζει ευστοχία σε πολύ μικρά εύρη όπως το [100000, 1000000] προκειμένου να παρέχει πολύ ισχυρότερη δυνατότητα γενίκευσης.

Ένας άλλος λόγος της φτωχής απόδοσης του Zestimate στο εύρος αυτό ενδέχεται να είναι τα ίδια τα δεδομένα τα οποία για κάποιο λόγο δεν ανταποκρίνονται καλά στον τρόπο με τον οποίο λειτουργεί το Zestimate. Στις περισσότερες εγγραφές η πρόβλεψή του είναι πολύ εύστοχη, υπάρχουν όμως κάποιες στις οποίες παρουσιάζεται πολύ μεγάλη απόκλιση. Τέτοιες εγγραφές θα επηρεάζουν σημαντικά και το RMSE.

Σύγκριση αποτελεσμάτων test set για τιμές μικρότερες του 1 εκατομμυρίου

Ακολούθως, παρουσιάζεται η ευστοχία των μοντέλων στο test set.

Μοντέλο	RMSE
Zestimate	149279
Multi-Layered Perceptron	132228
Random Forest	120641
Gradient Boosting Regressor	121626
Stochastic Gradient Descent	131638

Οι παρατηρήσεις που έγιναν πάνω στο training set εξακολουθούν να ισχύουν και εδώ. Μια επιπλέον παρατήρηση είναι πως το Multi-Layered-Perceptron φαίνεται να αποδίδει σταθερά ελαφρώς καλύτερα στο test set γεγονός που υποννοεί πως αποφεύγει εντελώς το πρόβλημα του overfitting και γενικεύει καλά, παρόλο που η καθαρή του απόδοση δεν είναι η καλύτερη. Εάν υπήρχε μεγαλύτερο πλήθος δεδομένων, ενδεχομένως να ήταν και το πιο ισχυρό μοντέλο από τα 4.

6.5.3 Σύγκριση αποτελεσμάτων για το αρχικό dataset με το Zestimate ως χαρακτηριστικό

Σύγκριση αποτελεσμάτων cross-validation για το αρχικό dataset με το Zestimate ως χαρακτηριστικό

Όπως και προηγουμένως, παρουσιάζεται πρώτα η ευστοχία των διαφορετικών μοντέλων που εκπαιδεύτηκαν και του Zestimate στα ακίνητα του training set. Η σημαντική διαφορά είναι πως το Zestimate αποτελεί πλέον χαρακτηριστικό που δίνεται στα μοντέλα ως είσοδος.

Μοντέλο	RMSE
Zestimate	152541
Multi-Layered Perceptron	142318
Random Forest	135068
Gradient Boosting Regressor	133317
Stochastic Gradient Descent	142477

Παρατηρείται πως όταν προστίθεται και σαν είσοδος Zestimate, τα μοντέλα πλέον αποδίδουν καλύτερα από το μεμονωμένο Zestimate στο εύρος τιμών των αρχικών εγγραφών. Αυτό είναι αναμενόμενο καθώς έχουν πλέον την πληροφορία του ίδιου του Zestimate και μπορούν να χρησιμοποιήσουν και μικρές λεπτομέρειες από τα υπόλοιπα δεδομένα οι οποίες κάνουν την πρόβλεψη ακόμα πιο εύστοχη στο συγκεκριμένο εύρος και τοποθεσία.

Παρατηρείται επιπλέον μια μικρή μείωση της διαφοράς στην απόδοση των διαφορετικών μοντέλων καθώς πλέον το μεγαλύτερο μέρος της πληροφορίας είναι συγκεντρωμένο στα χαρακτηριστικά zestimate και taxAssessedValueSqrt.

Σύγκριση αποτελεσμάτων test set για το αρχικό dataset με το Zestimate ως χαρακτηριστικό

Ακολούθως, παρουσιάζεται η ευστοχία των μοντέλων στο test set.

Μοντέλο	**RMSE**
Zestimate	150651
Multi-Layered Perceptron	142080
Random Forest	138243
Gradient Boosting Regressor	139059
Stochastic Gradient Descent	142625

Η εικόνα που παρατηρήθηκε στο training set παραμένει και εδώ.

Με τη σύγκριση αυτή ολοκληρώθηκε η παρουσίαση και αξιολόγηση του Data Pipeline που δημιουργήθηκε για τους σκοπούς της εργασίας.

Κεφάλαιο 7

Συμπεράσματα και Επίλογος

7.1 Συμπεράσματα

Από την εκπώνηση της παρούσας διπλωματικής προκύπτουν πολλά ενδιαφέροντα συμπεράσματα.

Καταρχάς, επιβεβαιώθηκε το γεγονός πως η πρόσβαση σε επαρκή όγκο δεδομένων για την εκπαίδευση μοντέλων μηχανικής μάθησης στην πρόβλεψη τιμών ακινήτων αποτελεί μία σύνθετη διαδικασία, στην περίπτωση που η πρόσβαση σε αυτά δεν χορηγηθεί από κάποια εταιρία που δραστηριοποιείται στον χώρο αυτό. Απαιτείται η χρήση διαφορετικών μέσων για την εξόρυξη όλων των απαραίτητων πληροφοριών, τα οποία κυμαίνονται από απλά public APIs έως σύνθετους web scrapers και χρήση IP proxies.

Επίσης, η ποιότητα των δεδομένων αποτελεί ένα βασικό εμπόδιο στην δημιουργία ενός ισχυρού μοντέλου πρόβλεψης. Θα μπορούσε μάλιστα η ποιότητα των δεδομένων να είναι ακόμα πιο σημαντική από την επιλογή μοντέλου. Λανθασμένες ή ακραίες τιμές και χαρακτηριστικά με πολλά κενά μπορούν να δημιουργήσουν θόρυβο τον οποίο ακόμα και τα πιο ισχυρά μοντέλα θα δυσκολευτούν να ξεχωρίσουν.

Η συλλογή, επομένως, επαρκών και ποιοτικών δεδομένων μπορεί να επιφέρει ικανοποιητικά αποτελέσματα στην δημιουργία ενός μοντέλου πρόβλεψης το οποίο δεν απέχει πολύ απο την απόδοση ενός πολύ σύνθετου μοντέλου, όπως το Zestimate. Εάν μάλιστα οι προβλέψεις ενός τόσο γενικού μοντέλου όπως το Zestimate ενσωματωθούν στο υπό εκπαίδευση μοντέλο, οι προβλέψεις του τελευταίου μπορεί να ξεπεράσουν το Zestimate σε ένα υποσύνολο των ακινήτων όπως τα ακίνητα αξίας μέχρι 2 εκατομυρίων στην πόλη της Νέας Υόρκης.

Επιπλέον, καθίσταται σαφές ότι υπάρχει ένα όριο στο πόσο εύστοχο μπορεί να είναι ενα τέτοιο μοντέλο χωρίς να έχει πρόσβαση σε πληροφορίες όπως η κατάσταση της επίπλωσης, η διαρρύθμιση του χώρου, ο προσανατολισμός, η αισθητική, κλπ., ενός ακινήτου. Εάν υπήρχε σχετικά πλήρης πρόσβαση και σε τέτοιου είδους δεδομένα θα μπορούσε να γίνει ακόμα πιο εύστοχο το μοντέλο.

Τέλος, μέσα από τα αποτελέσματα των μοντέλων φάνηκε η σημασία της αποφυγής του overfitting και της δημιουργίας ενός ποιοτικού test set. Το test set αυτό χρησιμοποιείται μόνο αφού έχει αποφασιστεί ποιο μοντέλα θα χρησιμοποιηθούν και με τι παραμέτρους. Έ-

τσι αποφεύγεται η δημιουργία μοντέλων τα οποία αποδίδουν στην πράξη χειρότερα από ότι περιμένει ο δημιουργός τους.

7.2 Οι προοπτικές μιας νεοφυούς επιχείρησης τεχνολογίας στον χώρο αγοραπωλησίας ακινήτων

Θα μπορούσε, συνεπώς, μια νεοφυής επιχείρηση να δραστηριοποιηθεί εύκολα στον τομέα της αγοραπωλησίας ακινήτων χωρίς άμεση πρόσβαση σε πλήθος δεδομένων όπως αυτά του Zillow; Η εργασία αυτή αποδεικνύει πως κάτι τέτοιο είναι δυνατό σε περιπτώσεις όπως η Νέα Υόρκη όπου πέρα από τα δεδομένα ακινήτων υπάρχει πρόσβαση και σε πληθώρα άλλων πληροφοριών για τις τοποθεσίες τους και το περιβάλλον τους.

Σε τοποθεσίες με λιγότερο προσβάσιμα δεδομένα, η δραστηριοποίηση στον τομέα αυτό γίνεται δυσκολότερη και απαιτεί ακόμα πιο λεπτομερείς και χρονοβόρες διαδικασίες και ενδεχομένως συνεργασία με άλλους φορείς για την συλλογή δεδομένων. Σίγουρα, το παράδειγμα της Νέας Υόρκης, στην οποία η πρόσβαση σε δεδομένα υποδομής και χαρακτηριστικών της πόλης είναι εύκολη, θα μπορούσε να αποτελέσει οδηγό για παρόμοιες πρωτοβουλίες και στην Ελλάδα.

Προκειμένου βέβαια να μπορέσει να είναι πραγματικά ανταγωνιστική μια νεοφυής επιχείρηση στον τομέα αυτό, θα μπορούσε να ακολουθήσει τις προτάσεις που παρουσιάζονται παρακάτω. Πρόκειται για προτάσεις επέκτασης της διπλωματικής και ενδεχομένως απαραίτητες προϋποθέσεις για να δημιουργηθεί ένα state of the art σύστημα πρόβλεψης τιμών ακινήτων, ικανό να παράγει πλούτο σε μια διαρκώς πιο ανταγωνιστική αγορά.

7.3 Μελλοντικές επεκτάσεις

Η παρούσα διπλωματική θα μπορούσε να επεκταθεί με διάφορους τρόπους στο μέλλον.

Ένας προφανής τρόπος είναι η συλλογή μεγαλύτερης ποσότητας δεδομένων τα οποία καλύπτουν μεγαλύτερη έκταση των Ηνωμένων Πολιτειών της Αμερικής ή μια τελείως διαφορετική περιοχή. Η προφανής δυσκολία στο εγχείρημα αυτό είναι η επιπλέον προσπάθεια που θα χρειαστεί στον σχεδιασμό της συλλογής δεδομένων. Θα χρειαστούν διαφορετικοί web scrapers και χρήση διαφορετικών APIs τα οποιά θα παρέχουν πληροφορία σε διαφορετική μορφή.

Εναλλακτικά, θα μπορούσε να δοθεί επιπλέον έμφαση στο λογαριθμικό κομμάτι της εργασίας και να αναλυθούν περισσότεροι αλγόριθμοι καθώς και περισσότερες μέθοδοι προεπεξεργασίας, κωδικοποίησης και κανονικοποίησης των δεδομένων. Ο σκοπός της παρούσας εργασίας ήταν να αποτελέσει Proof of Concept για ένα ολοκληρωμένο Data Pipeline και όχι να εστιάσει στην εύρεση του βέλτιστου μοντέλου.

Επιπλέον, προκειμένου να βελτιωθούν οι προβλέψεις που παράγονται στο τέλος του Pipeline θα μπορούσε να ενσωματωθεί στα δεδομένα οπτικό υλικό, φωτογραφίες δηλαδή από κάθε ακίνητο. Για τον λόγο αυτό θα χρειαστούν αλγόριθμοι αναγνώρισης/επεξεργασίας εικόνας και ένας αποδοτικός τρόπος αποθήκευσης ώστε να παραμείνει εύκολα διαχειρίσμο το dataset.

Τέλος, μία πιθανή επέκταση με μεγάλο ενδιαφέρον αποτελεί η μετατροπή του Pipeline σε on-line μορφή. Μια μορφή δηλαδή με δυνατότητα real-time συλλογής και επεξεργασίας δεδομένων τα οποία θα τροφοδοτούν μοντέλα μηχανικής μάθησης με στόχο την συνεχή τους εξέλιξη. Τελικός στόχος θα μπορούσε να είναι η δημιουργία ενός μοντέλου το οποίο θα προβλέπει όχι μόνο την παροντική τιμή ενός ακινήτου αλλά και την μελλοντική. Ένα τέτοιο εγχείρημα θα απαιτούσε τον μεγαλύτερο αριθμό δεδομένων, συμπεριλαμβανομένων ιστορικών στοιχείων αλλά θα οδηγούσε και στο μεγαλύτερο όφελος.

Βιβλιογραφία

[1] Byeonghwa Parka, Jae Kwon Bae, Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. *Expert Systems with Applications vol.* 42, Issue 6, pp. 2928-2934, 15 April 2015.

[2] Mohammad Hossein Rafiei, Hojjat Adeli, A Novel Machine Learning Model for Estimation of Sale Prices of Real Estate Units. *Journal of Construction Engineering and Management* vol. 142, Issue 2, February 2016.

[3] Guangli Liu, Xiaohui Zong, Research of second-hand real estate price forecasting based on data mining. *IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference*, 2017.

[4] Alejandro Baldominos, Ivan Blanco, Antonio Jose Moren, Ruben Iturrarte, Óscar Bernardez, Carlos Afonso, Identifying Real Estate Opportunities Using Machine Learning. *Applied Sciences* , 8(11), pp. 2321, 2018.

[5] Quanzeng You, Ran Pang, Liangliang Cao, Jiebo Luo, Image-Based Appraisal of Real Estate Properties. *IEEE Transactions on Multimedia*, vol. 19, Issue 12, pp. 2751 - 2759, Dec. 2017).

[6] Koziarski Michal, Cyganek Boguslaw, Image recognition with deep neural networks in presence of noise – Dealing with and taking advantage of distortions. *Integrated Computer-Aided Engineering*, , vol. 24, no. 4, pp. 337-349, 2017.

[7] Omid Poursaeed, Tomas Matera, Serge Belongie, Vision-based real estate price estimation. *Machine Vision and Applications*, vol. 29, pp. 667–676, 2018.

[8] Edward L. Glaeser, Michael Scott Kincaid, Nikhil Naik, Computer Vision and Real Estate: Do Looks Matter and Do Incentives Determine Looks. *NBER WORKING PAPER SERIES, Working Paper 25174*, [Online] Available: https://www.nber.org/papers/w25174.

[9] Lotfi A. Zadeh, Saied Tadayon, Bijan Tadayon, System and Method for Extremely Efficient Image and Pattern Recognition and Artificial Intelligence Platform. *Google Patents*, [Online] Available: https://patents.google.com/patent/US20180204111A1/en.

[10] D. Stevens, Predicting Real Estate Price Using Text Mining Automated Real Estate Description Analysis. [Online] Available: https://www.semanticscholar.org/paper/Predicting-Real-Estate-Price-Using-Text-Mining-Real-Stevens/c5d235d4a4e27b7512fab65fd0528d5abb7bbf9cciting-papers, 2014.

[11] M. Shahbazi, J. R. Barr, V. Hristidis and N. N. Srinivasan, Estimation of the Investability of Real Estate Properties through Text Analysis. *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pp. 301-306, doi: 10.1109/ICSC.2016.85, 2016.

[12] Abdallah S., Khashan D.A., Using Text Mining to Analyze Real Estate Classifieds. *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016. AISI 2016*, Advances in Intelligent Systems and Computing, vol 533. Springer, Cham. https://doi.org/10.1007/978-3-319-48308-5_19, 2016.

[13] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, Chris Develder, Reconstructing the house from the ad: Structured prediction on real estate classifieds. [Online] Available: https://biblio.ugent.be/publication/8521270/file/8521272.pdf, 2017.

[14] Lily Shen, Stephen Ross, Housing Prices and Property Descriptions: Using Soft Information to Value Real Assets. [Online] Available: https://media.economics.uconn.edu/working/2019-20.pdf, 2019.

[15] Daniel Glez-Pena, Analia Lourenco, Hugo Lopez-Fernandez, Miguel Reboiro-Jato, Florentino Fdez-Riverola, Web scraping technologies in an API world. *Briefings in Bioinformatics*, vol. 15, Issue 5, pp. 788–797, September 2014.

[16] De S Sirisuriya, A Comparative Study on Web Scraping. [Online] Available: http://ir.kdu.ac.lk/handle/345/1051, 2015.

[17] Krotov Vlad, Leiser Silva, Legality and ethics of web scraping. [Online] Available: https://www.researchgate.net/profile/Vlad-Krotov/publication/324907302_Legality_and_Ethics_of_Web_Scraping/links/5aea622345851588dd8287dc/Legality-and-Ethics-of-Web-Scraping.pdf, 2018.

[18] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review vol.* 65, pp. 6, 1958.

[19] Ν. Βασιλειάδης Φ. Κόκκορας Η. Σακελλαρίου Ι. Βλαχάβας, Π. Κεφαλάς, *Τεχνητή Νοημοσύνη. Εκδόσεις Πανεπιστημίου Μακεδονίας*, 2006.

[20] Simon O. Haykin, *Neural Networks and Learning Machines, Third Edition, Pearson Education, McMaster University, Canada*, 2009.

[21] Stanford. Introduction to Keras. `https://web.stanford.edu/class/cs20si/lectures/march9guestlecture.pdf`, 2018.

[22] Wikipedia. Scikit-learn. https://en.wikipedia.org/wiki/Scikit-learn, 2021.

[23] SPSS Inc, *SPSS Clementine 12.0 Algorithms Guide*, 2007.

[24] P. Latinne, O. Debeir and C. Decaestecker, Mixing bagging and multiple feature subsets to improve classification accuracy of decision tree combination. *Proceedings of the Tenth Belgian-Dutch Conference on Machine Learning* pp. 15-22, 2000.

[25] A. Natekin and A. Knoll, Gradient boosting machines, a tutorial, *Front. Neurorobot.,* vol. 7, no. DEC, 2013.

[26] A. Keprate and R. M. C. Ratnayake, Using gradient boosting regressor to predict stress intensity factor of a crack propagating in small bore piping, *IEEE Int. Conf. Ind. Eng. Eng. Manag.,* pp. 1331–1336, vol. 2017-Decem.

[27] J. H. Friedman, Greedy function approximation: A gradient boosting machine, *Ann. Stat.,* vol. 29, no. 5, pp. 1189–1232, 2001.

[28] Github. Zillow Scraper. https://github.com/huangyingw/cermak-petr_actor-zillow-api-scraper, 2021.

[29] Zillow. Zestimate. https://www.zillow.com/z/zestimate/, 2021.

[30] Zillow. Zillow Awards $1 Million to Team that Built a Better Zestimate. http://zillow.mediaroom.com/2019-01-30-Zillow-Awards-1-Million-to-Team-that-Built-a-Better-Zestimate, 2021.

[31] Zillow. Zillow Home Value Index Methodology, 2019 Revision: What's Changed Why. https://www.zillow.com/research/zhvi-methodology-2019-highlights-26221/, 2019.

[32] SchoolDigger. SchoolDigger.com Ranking Methodology. https://www.schooldigger.com/aboutrankingmethodology.aspx/, 2021.

[33] Wikipedia. Student–teacher ratio. https://en.wikipedia.org/wiki/Student\%E2\%80\%93teacher_ratio, 2021.

[34] WalkScore. Walk Score Methodology. https://www.walkscore.com/methodology.shtml, 2021.

[35] Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., Multivariate Data Analysis: A Global Perspective: Pearson Education International, *New Jersey,* 2010.

[36] Aurelien Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, *O'Reilly Media,* 2019.