ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Σύγκριση μεθόδων πρόβλεψης μελλοντικών τοποθεσιών ατόμων σε εσωτερικούς χώρους

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Μπέλεση Χρήστου

Επιβλέπων : Ασκούνης Δημήτριος
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2019

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Σύγκριση μεθόδων πρόβλεψης μελλοντικών τοποθεσιών ατόμων σε εσωτερικούς χώρους

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Μπέλεση Χρήστου

Επιβλέπων : Ασκούνης Δημήτριος
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 3η Οκτωβρίου 2019.

.....	.....	.....
Δημήτριος Ασκούνης	Ιωάννης Ψαρράς	Χάρης Δούκας
Καθηγητής Ε.Μ.Π	Καθηγητής Ε.Μ.Π	Επ. Καθηγητής Ε.Μ.Π

Αθήνα, Οκτώβριος 2019

.....
Μπέλεσης Χρήστος

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

Περίληψη

Τα τελευταία χρόνια αναπτύσσεται ραγδαία η επιστήμη της μελέτης κίνησης ανθρώπων σε εσωτερικούς χώρους. Η μελέτη αυτή βοηθάει κατά κύριο λόγο στην εξασφάλιση της ασφάλειας των πολιτών, στην διευκόλυνσή τους αλλά και στην παροχή πληροφοριών και ψυχαγωγίας σε αυτούς που το επιθυμούν. Για να πραγματοποιηθεί αυτό πρέπει πρώτα ο χρήστης να έχει δώσει τη συγκατάθεση του και να συμμετέχει ενεργά μέσω της συσκευής κινητού του ώστε να είναι εφικτός ο εντοπισμός από ειδικά μηχανήματα και τεχνολογίες που εγκαθίστανται για αυτό το σκοπό.

Μια πρόκληση του ερευνητικού αυτού κλάδου είναι και η πρόβλεψη της τοποθεσίας του ατόμου είτε την αμέσως επόμενη χρονική στιγμή είτε κάποια στιγμή στο μέλλον , η οποία παρέχει διάφορα πλεονεκτήματα τόσο για τον ίδιο όσο και για τους ιδιοκτήτες καταστημάτων και εταιριών αφού μεταξύ άλλων επιτρέπει την στοχευμένη διαφήμιση και παροχή υπηρεσιών άμεσα στο χρήστη ανάλογα με την τοποθεσία του.

Στην παρούσα διπλωματική εργασία εφαρμόζονται κυρίως αλγόριθμοι μηχανικής εκμάθησης αλλά και αλγόριθμοι deep learning πάνω σε δοσμένα ανοιχτά δεδομένα ενός εμπορικού κέντρου ώστε να προβλεφθεί η επόμενη τοποθεσία εντός του χώρου. Ο σκοπός είναι να συγκριθούν τα αποτελέσματα που παράγουν ως προς την ακρίβεια της πρόβλεψης και τελικά να αποφανθεί η καταλληλόλητά τους υπό συγκεκριμένες συνθήκες και παραμέτρους. Για να επιτευχθεί αυτό γίνεται χρήση τόσο ιστορικού προηγούμενων τοποθεσιών, όσο και δημογραφικών χαρακτηριστικών των συμμετεχόντων στο πείραμα πολιτών.

Λέξεις Κλειδιά: Κίνηση Εσωτερικού χώρου, Πρόβλεψη Επόμενης Τοποθεσίας, Μηχανική εκμάθηση, Deep learning, Dynamic Bayesian Model, Compact Prediction Tree, Multinomial Logistic Regression.

Abstract

In recent years the science of studying the movement of people indoors has grown rapidly. This study primarily helps ensure the safety of citizens, facilitates them, and provides information and entertainment to those who desire it. To accomplish this, the user must first have their consent and actively participate through their mobile device in order to be able to be identified by specific machines and technologies installed for this purpose.

One challenge of this research field is to predict the location of the individual either at the next moment or at some point in the future, which provides various benefits for both himself and the owners of the shops and companies as it allows, among other things, targeted advertising and services supply directly to the user depending on their location.

The present thesis mainly applies machine learning algorithms but also deep learning algorithms on given open data of a shopping mall to predict the next location within the space. The purpose is to compare the results they produce, study and analyze the accuracy of the prediction and ultimately determine their suitability under specific conditions and parameters. To do this, both past visited locations and demographic characteristics of the citizens were used.

Keywords: Indoor Movement, Next Location Prediction, Machine learning, Deep Learning, Dynamic Bayesian Model, Compact Prediction Tree, Multinomial Logistic Regression.

Πρόλογος

Αρχικά, θα ήθελα να ευχαριστήσω τόσο τον κ. Δημήτριο Ασκούνη, Καθηγητή Ε.Μ.Π και επιβλέποντα της παρούσας εργασίας, που μου έδωσε την δυνατότητα να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα, όσο και τα υπόλοιπα μέλη της επιτροπής τον κ. Ιωάννη Ψαρρά και τον κ. Χρυσόστομο Δούκα. Επίσης, θέλω να ευχαριστήσω τον κ. Ιωάννη Τσαπέλα, που καθ' όλη την διάρκεια εκπόνησης της εργασίας ήταν πάντα διαθέσιμος να με βοηθήσει. Τέλος να ευχαριστήσω την οικογένεια μου, τους φίλους μου και την κοπέλα μου που δεν σταμάτησαν να με ενθαρρύνουν και να με στηρίζουν μέχρι την τελευταία στιγμή.

Μπέλεσης Χρήστος,

Αθήνα, 3η Οκτωβρίου 2019

Πίνακας Περιεχομένων

Κεφάλαιο 1. Εισαγωγή

1.1 Αντικείμενο – Σκοπός

Ζούμε σε μια εποχή στην οποία ο εντοπισμός των ατόμων μέσω ηλεκτρονικών συσκευών συμβαίνει πλέον καθημερινά και έχει εισαχθεί στις ζωές μας για τα καλά.

Χαρακτηριστικά παραδείγματα είναι τα GPS που αποτελούν πλέον αναπόσπαστο κομμάτι των αυτοκινήτων ή/και των κινητών τηλεφώνων μας αλλά και γενικά οι Εφαρμογές Βάσει Τοποθεσίας – Location Based Services.

Οι Εφαρμογές Βάσει Τοποθεσίας όπως φανερώνεται από την ονομασία τους, είναι εφαρμογές που προσφέρονται στον χρήστη μέσω κινητού τηλεφώνου και λαμβάνουν υπόψη τους την γεωγραφική τοποθεσία της συσκευής που τις χρησιμοποιεί. Η γενική τους χρήση συνεπώς είναι να παρέχουν πληροφορία, ψυχαγωγία και ασφάλεια στον χρήστη αν αυτός επιλέξει να μοιραστεί την πραγματική τοποθεσία του στο χάρτη.

Οι χρησιμότητα των LBS φαίνεται αν δούμε τις κυριότερες χρήσεις τέτοιων εφαρμογών. Μία από τις πιο βασικές είναι το λεγόμενο marketing βάσει πλησιέστερης τοποθεσίας (proximity-based marketing) κατά το οποίο οι εταιρίες έχουν την δυνατότητα να εμφανίζουν στη συσκευή του χρήστη στοχευμένες διαφημίσεις ανάλογα με την γεωγραφική του θέση. Επιπλέον, μέσω των LBS παρέχονται αληθινού χρόνου πληροφορίες (real-time information), όπως ενημέρωση της κίνησης στους δρόμους και αναφορά καιρικών συνθηκών, ώστε ο χρήστης να προγραμματίσει ανάλογα την πορεία του. Άλλες πιθανές χρήσεις είναι η παροχή δυνατότητας σε εργαζόμενους εξωτερικού χώρου να δηλώσουν την παρουσία τους σε οποιαδήποτε τοποθεσία βρίσκονται αλλά και σε απλούς πολίτες να λύσουν γρήγορα το όποιο πρόβλημα αντιμετωπίσουν χωρίς να χρειάζεται να δώσουν σαφείς οδηγίες και κατευθύνσεις (όπως για παράδειγμα χαλασμένο λάστιχο αυτοκινήτου ή απώλεια αντικειμένου από την κατοχή του).

Κατά αντιστοιχία με τους εξωτερικούς χώρους, συστήματα εντοπισμού της θέσης έχουν αναπτυχθεί και για τους εσωτερικούς χώρους, τα Συστήματα Αναγνώρισης της Εσωτερικής Τοποθεσίας (Indoor Positioning Systems – IPS). Ένα IPS είναι ένα δίκτυο συσκευών που χρησιμοποιείται για τον εντοπισμό ανθρώπων ή αντικειμένων εκεί που τα GPS και άλλες δορυφορικές τεχνολογίες είτε χάνουν ακρίβεια είτε αποτυγχάνουν να τελείως, όπως για παράδειγμα σε πολυόροφα κτήρια, αεροδρόμια, σοκάκια, τοποθεσίες πάρκινγκ και γενικά υπόγειους χώρους. Μια τεράστια πληθώρα από τεχνικές και συσκευές χρησιμοποιούνται για να μας παρέχουν την εσωτερική τοποθεσία που μπορεί να ποικίλουν από ειδικά διαμορφωμένες «έξυπνες» συσκευές, Wi-Fi και Bluetooth κεραίες, ψηφιακές κάμερες και ρολόγια μέχρι εγκαταστάσεις με Relays και beacons στρατηγικά τοποθετημένες κατά την έκταση του μελετηθέντος χώρου. Η τεχνολογία των beacons έχει να κάνει ουσιαστικά με την ύπαρξη εξαρτημάτων κατάλληλων να ανιχνεύσουν κίνηση στον χώρο που έχουν τοποθετηθεί και πρακτικά διαδραματίζουν στον εσωτερικό χώρο αντίστοιχο ρόλο με αυτόν ενός φάρου στον εξωτερικό χώρο.

Τα IPS έχουν ευρεία χρήση σε κλάδους όπως εμπόριο (commercial), στρατός (military), λιανική πώληση (retail) αλλά και σε εταιρίες παρακολούθησης εμπορεύματος (inventory tracking industries).

Γιατί όμως είναι τόσο σημαντικά τα IPS και έχουν γνωρίσει τόσο μεγάλη ανάπτυξη την σήμερον ημέρα; Ένα από τα βασικά τους πλεονεκτήματα είναι ότι καθιστούν την εύρεση τοποθεσιών πολύ ευκολότερη καθώς ο χρήστης βλέπει την πραγματική του τοποθεσία στο χάρτη (blue dot) κατά την πλοήγηση του από το σημείο Α στο Β χωρίς να χρειάζεται να πληκτρολογεί το μέρος που θέλει να βρεθεί ή να απαιτείται να μετράει στροφές, πόρτες, διαδρόμους και λοιπά πράγματα που μπορούν να τον μπερδέψουν. Επιπλέον, μέσω του IPS υπάρχει η δυνατότητα του heat mapping δηλαδή η μελέτη ουσιαστικά των πολυσύχναστων (θερμών) περιοχών στο χάρτη που στην προκειμένη περίπτωση μας δείχνει την ροή των ατόμων στο χώρο. Αυτό δίνει σημαντική πληροφορία κυρίως στους καταστηματάρχες ώστε να βελτιώσουν την διαφήμιση και την προώθηση του μαγαζιού τους με σκοπό να μεγιστοποιηθούν τα κέρδη τους, όμως παράλληλα βοηθάει και στην μείωση του χρόνου των ουρών αναμονής σε αυτόματους πωλητές αναψυκτικών και φαγητών.

Ακόμα, η γνώση του πώς κινείται κάποιος βοηθάει στο να βελτιωθεί η άνεση και η ασφάλεια του. Για παράδειγμα, παρατηρώντας την πορεία κίνησης ενός ηλικιωμένου σε ένα σπίτι μπορούμε να έχουμε το νου μας για τυχόν παράξενες/ εκτός ρουτίνας κινήσεις για να βεβαιωθούμε για την ασφάλεια του. Σε ένα παράδειγμα νοσοκομείου η παρατήρηση της κίνησης της νοσοκόμας μπορεί να μας οδηγήσει στην βελτιστοποίηση της πορείας που πρέπει να ακολουθήσει ώστε η βάρδια της να γίνει πιο αποδοτική.

Σε ένα τελευταίο αλλά πολύ σημαντικό παράδειγμα που έχει να κάνει με καταστήματα, η πρόβλεψη της κίνησης των ατόμων μπορεί να δώσει γνώση στα καταστήματα για τις αποδοτικότερες θέσεις τοποθέτησης των προϊόντων τους ή να τα οδηγήσει στην επιλογή της καλύτερης δυνατής προώθησης και διαφήμισης των προϊόντων τους (targeted retail promotion).

Έτσι γεννιέται ένα ακόμα ερευνητικό ερώτημα: Πώς μπορούμε να χρησιμοποιήσουμε αυτή την διαθέσιμη πληροφορία, να την αναλύσουμε, να την συνδυάσουμε με πρόσθετες πληροφορίες, ώστε να δημιουργήσουμε καινοτομικές υπηρεσίες που παράγουν νέα γνώση και επιπρόσθετη αξία. Ένα από τα ενδιαφέροντα αποτελέσματα από τα παραπάνω και ταυτόχρονα πρόκληση είναι η πρόβλεψη μελλοντικών τοποθεσιών βάσει ιστορικών δεδομένων.

Όπως είναι σαφές η διπλωματική κινείται στο κομμάτι του indoor movement prediction και συγκεκριμένα στην πρόβλεψη της επόμενης θέσης του ατόμου. Το αντικείμενο της παρούσας εργασίας, λοιπόν, είναι η μελέτη, η σύγκριση και η αξιολόγηση των διαφορετικών μοντέλων και μεθόδων πρόβλεψης της μελλοντικής τοποθεσίας καταναλωτών σε φυσικά καταστήματα.

Ο στόχος της εργασίας αυτής είναι η εφαρμογή, μελέτη και σύγκριση διαφόρων μοντέλων σε δεδομένα μεγάλης περίπτωσης με τελικό σκοπό να ελεγχθούν η επίδοση και τα χαρακτηριστικά τους.

Η εφαρμογή των μεθόδων που θα αναλυθούν παρακάτω πραγματοποιήθηκε σε ένα σύνολο ανοιχτών δεδομένων προερχόμενα από μια μελέτη περίπτωσης που αφορά ένα μεγάλο εμπορικό κέντρο της Αμερικής. Χρησιμοποιούνται στη συνέχεια διάφορες παράμετροι που επηρεάζουν την πρόβλεψη όπως είναι η ηλικία και το φύλο του ατόμου και εφαρμόζονται μια πληθώρα από αλγορίθμους από τους οποίους προκύπτουν εν

τέλει προφανέστατα ποικίλα αποτελέσματα με σημαντικές διαφορές μεταξύ τους. Μετά από την εφαρμογή αλγορίθμων στα δοθέντα ανοιχτά δεδομένα, συγκρίνονται τα αποτελέσματα τους και βγαίνουν συμπεράσματα σχετικά με την αποδοτικότητα και την ακρίβεια τους κάτω από συγκεκριμένες συνθήκες.

1.2 Φάσεις υλοποίησης

Η εκπόνηση της διπλωματικής εργασίας πραγματοποιήθηκε μεταξύ Απριλίου και Σεπτεμβρίου 2019, και αφορούσε τόσο την συγγραφή κώδικα όσο και την προετοιμασία του θεωρητικού υποβάθρου.

Παρακάτω αναφέρονται συνοπτικά οι φάσεις υλοποίησης της εν λόγω εργασίας:

Φάση 1:

Διεξοδική μελέτη και ανάγνωση προηγούμενων εργασιών στον τομέα του indoor movement analysis και τελική απόφαση για ασχολία με future location prediction.

Φάση 2:

Εκτενής αναζήτηση των αλγορίθμων αυτών που θα μου επέτρεπαν να αναλύσω το πρόβλημα του prediction και εν τέλει να βγάλω συμπεράσματα από τα αποτελέσματα αυτών.

Φάση 3:

Σοβαρή προσπάθεια για κατανόηση των εν λόγω αλγορίθμων συμπεριλαμβανομένων των εισόδων, των παραμέτρων αλλά και των πιθανών αποτελεσμάτων αυτών.

Φάση 4:

Σωστή εκμάθηση της γλώσσας προγραμματισμού Python και αναζήτηση των απαραίτητων modules που θα χρειαζόντουσαν για την εκτέλεση της διπλωματικής εργασίας.

Φάση 5:

Λεπτομερείς μελέτη και ανάλυση των open data που είχα στα χέρια μου και που ήταν αναγκαία για να λειτουργήσουν ομαλά οι αλγόριθμοι κατά το στάδιο συγγραφής του κώδικα.

Φάση 6:

Στάδιο προγραμματισμού κατά τη διάρκεια του οποίου έγιναν πολλές δοκιμές, αλλαγές και απόπειρες ώστε να παραχθεί αποτέλεσμα που να εκπληρώνει τους στόχους της διπλωματικής.

Φάση 7:

Τελική φάση συγγραφής του τόμου της διπλωματικής εργασίας, η οργάνωση του οποίου αναφέρεται στο τελευταίο κομμάτι της εισαγωγής.

1.3 Οργάνωση τόμου

Στην υποενότητα αυτή της εισαγωγής συνοψίζεται η δομή του παρόντος τόμου ξεκινώντας από το 2ο κεφάλαιο μέχρι το τελευταίο .

Η δομή του τόμου είναι η εξής:

- Κεφάλαιο 1: Εισαγωγή και συνοπτική παρουσίαση της εργασίας.

- Κεφάλαιο2: Αναφορά βασικών θεωρητικών μερών που αφορούν το ερευνητικό πεδίο, παρουσίαση ήδη υπαρχουσών λύσεων, σύγκριση τους με την προσέγγιση της διπλωματικής και επισήμανση της διαφοροποίησής της.

- Κεφάλαιο 3: Περιγραφή της μελέτης περίπτωσης και ανάλυση των διαθέσιμων δεδομένων.

- Κεφάλαιο 4: Ανάλυση της λύσης που προτείνεται στην εργασία, σχολιασμός παραδοχών, ορισμών, προετοιμασία δεδομένων, περιγραφή/εφαρμογή αλγορίθμων και παρουσίαση αποτελεσμάτων.

- Κεφάλαιο 5: Συζήτηση αποτελεσμάτων και προτάσεις για μελλοντική έρευνα, βιβλιογραφία.

Κεφάλαιο 2. Ανάλυση του επιστημονικού πεδίου

2.1 Σχετικές εργασίες και εφαρμογές

Στο κεφάλαιο αυτό πραγματοποιείται μια ανασκόπηση της σχετικής βιβλιογραφίας για την ανάλυση δεδομένων κίνησης σε εσωτερικούς χώρους και ειδικότερα για τις μεθόδους πρόβλεψης μελλοντικών εσωτερικών τοποθεσιών. Παρουσιάζονται σχετικές υλοποιήσεις και προηγούμενες εργασίες που έχουν πραγματοποιηθεί, αναλύονται οι ακολουθούμενες μεθοδολογίες και τέλος περιγράφεται η διαφοροποίηση και η συμβολή της παρούσας εργασίας.

Το πανεπιστήμιο του Colorado [7] ασχολήθηκε με το λεγόμενο Adaptive House Project (Δημιουργία Προσαρμοζόμενου Σπιτιού), δηλαδή με την ανάπτυξη ενός έξυπνου (smart) σπιτιού, το οποίο θα παρατηρούσε την ζωή και τις ανάγκες των κατοίκων του και εν συνεχεία θα ήταν σε θέση να προβλέπει και να καλύπτει τις ανάγκες αυτές. Η κίνηση των συμμετεχόντων κατοίκων του σπιτιού καταγράφτηκε μέσω αισθητήρων εντοπισμού κίνησης (motion detectors), ενώ ο τρόπος για να προβλεφθεί το επόμενο δωμάτιο που θα βρεθεί κάποιος και οι δραστηριότητες που θα πραγματοποιηθούν σε αυτό ήταν μέσω νευρωνικών δικτύων (neural network).

Επιπλέον, σε μια εργασία καθαρά αλγοριθμικής φύσης, ο Donald J. Patterson [8] μαζί με μια ομάδα τριών ακόμα επιστημόνων από το τμήμα του Computer Science and Engineering του πανεπιστημίου της Washington παρουσίασαν μια μέθοδο εκμάθησης του Μπαγιεσιανού μοντέλου (Bayesian Network) ενός πολίτη που κινείται σε ένα καθημερινό αστικό περιβάλλον βασισμένο στον εκάστοτε μέσο μεταφοράς του.

Με τον αλγόριθμο Dynamic Bayesian Network ασχολήθηκε και ο Sunyoung Lee μαζί με τους συνεργάτες του από το τμήμα της επιστήμης της αλληλεπίδρασης του πανεπιστήμιο της Σεούλ. Οι προαναφερθέντες έκαναν χρήση του αλγορίθμου για την πρόβλεψη επόμενης επίσκεψης στον χώρου των ευρέως γνωστών υπολογιστικών εφαρμογών. Η πεποίθηση τους είναι πως η συμπεριφορά ενός χρήστη καθορίζει την μελλοντική του επίσκεψη στο διαδίκτυο. Ωστόσο η εργασία τους δεν εξέταζε άλλο μοντέλο, αλλά ακόμα και στο ίδιο το μοντέλο που εξέταζε, τα δεδομένα που δέχονταν ο DBN σαν είσοδο, δηλαδή οι παράμετροι του, ήταν μόνο οι προηγούμενες αλληλεπιδράσεις του χρήστη με άλλους χρήστη. Κατά αντιστοιχία, στην έρευνα της διπλωματικής μας, παρουσιάζεται η πρόβλεψη τοποθεσίας σε εσωτερικό χώρου μέσω αλγορίθμων (μεταξύ άλλων και του Bayesian) με σκοπό να εμπλουτιστούν οι παράμετροι εισόδου του μοντέλου με πλουσιότερα δεδομένα και πληροφορίες όπως τα δημογραφικά στοιχεία.

Στην εργασία Prediction by Partial Matching [2] (PPM) πραγματοποιούνται προβλέψεις βασισμένες στα τελευταία K αντικείμενα της ακολουθίας όπου το K προσδιορίζει την τάξη του μοντέλου. Σε ένα K-PPM το αποτέλεσμα μιας δοσμένης ακολουθίας προβλέπεται με το να αντιστοιχίζονται τα τελευταία K αντικείμενα του γράφου με μια κορυφή του. Ωστόσο ενώ αυτή η προσέγγιση έφερε σε ορισμένες περιπτώσεις καλά αποτελέσματα [2,3], το βασικό της μειονέκτημα είναι η έλλειψη ευελιξίας στα pattern που μπορεί να μάθει. Επιπλέον, όσο μικρότερη είναι η ακολουθία τόσο πιο μικρή ακρίβεια επιτυγχάνεται, γεγονός που εντείνεται ακόμα περισσότερο αν

τα δεδομένα διαθέτουν «θόρυβο» (noisy datasets). Την λύση σε αυτό το πρόβλημα ήρθε να δώσει ο Ted Gueniche με τον Philippe Fournier Viger [5] εφαρμόζοντας τον αλγόριθμο Compact Prediction Tree (CPT) ο οποίος πρακτικά εξασφαλίζει πως το μοντέλο που θα παραχθεί δεν έχει απώλειες και δεν θα αγνοεί σημαντικές πληροφορίες της ακολουθίας που χρησιμοποιεί για την εκμάθηση (training) όταν είναι έτοιμο να παράγει πρόβλεψη. Μειονέκτημα της εργασίας αυτής αλλά και πρόκληση για την συγκεκριμένη διπλωματική είναι και πάλι ο μικρός αριθμός αλγορίθμων που χρησιμοποιούν.

Ακόμα, ο Syed Shahan Ali [10] μαζί με τους συνεργάτες του ασχολήθηκε με το μοντέλο Logistic Regression και το χρησιμοποίησε σαν μέθοδο πρόβλεψης επόμενης τιμής. Η έρευνα των εν λόγω επιστημόνων είχε σαν στόχο την πρόβλεψη των τιμών του χρηματιστηρίου μέσω του συγκεκριμένου μοντέλου με χρήση των χρηματοοικονομικών δεικτών σαν ανεξάρτητες μεταβλητές και της απόδοσης του χρηματιστηρίου (stock performance) σαν εξαρτημένη μεταβλητή. Βέβαια παρότι το ποσοστό ακρίβειας της μεθόδου κινήθηκε σε πολύ υψηλά νούμερα, δεν ελέγχθηκαν άλλες μέθοδοι για την πρόβλεψη οι οποίες εν δυνάμει θα έδιναν καλύτερα ή πιο αξιόπιστα αποτελέσματα.

Τέλος, ο Jan Petzold με τους συνεργάτες του από το πανεπιστήμιο του Augsburg και συγκεκριμένα το τμήμα της επιστήμης των υπολογιστών ασχολήθηκε με την σύγκριση διαφόρων αλγορίθμων πρόβλεψης ως προς το ποσοστό ακρίβειας τους. Αλγόριθμοι που εξετάστηκαν ήταν ο state predictor, ο Markov model και ο Elman net οι οποίοι έδωσαν σημαντικά αποτελέσματα και έδωσαν μια γενική λύση του προβλήματος της πρόβλεψης μελλοντικής τοποθεσίας.

Όπως φαίνεται από τις προαναφερθείσες έρευνες, υπάρχει μεγάλη πληθώρα μοντέλων για να παραχθεί τελικά μια πρόβλεψη επόμενης τοποθεσίας/τιμής. Κάθε μοντέλο όπως είναι φυσικό έχει τα θετικά και τα αρνητικά του. Το βασικό όμως πρόβλημα των παραπάνω εργασιών είναι ο έλεγχος μεμονωμένων μοντέλων κάθε φορά για την πρόβλεψη.

Έτσι, στην παρούσα διπλωματική θα εφαρμόσουμε τους αλγορίθμους Dynamic Bayesian Network, Compact Prediction Tree, Logistic Regression Model αλλά και ορισμένα (άλλα μοντέλα machine και deep learning) για να προβλέψουμε επόμενη τοποθεσία ενός πελάτη στο διαθέσιμο από τα ανοιχτά δεδομένα εμπορικό κέντρο με γνωστή την πορεία του μέχρι στιγμής. Στόχος είναι η ανάλυση και σύγκριση των εν λόγω αλγορίθμων τόσο με βάση τις συνθήκες κάτω από τίς οποίες εφαρμόζονται όσο και με βάση την εκάστοτε παράμετρο που θα μπαίνει σαν είσοδος στον κάθε αλγόριθμο.

Κατόπιν, θα σχολιαστούν τα αποτελέσματα και θα αποφανθεί η καταλληλόλητα του καθενός σε κάθε περίπτωση. Αυτό θα γίνει μέσω της σύγκρισης των αλγορίθμων σε 6 διαφορετικές κατηγορίες πρόβλεψης που αφορούν τόσο προηγούμενες θέσεις όσο και δημογραφικά χαρακτηριστικά ενώ παράλληλα θα ελεγχθούν 2 ξεχωριστές θεωρήσεις στην κατασκευή των μοντέλων με αποτέλεσμα να προκύψουν συνολικά 12

αποτελέσματα για κάθε αλγόριθμο τα οποία θα μελετηθούν ως προς την ακρίβεια πρόβλεψης τους και έτσι θα φανεί ποιος υπερτερεί και γιατί.

Κεφάλαιο 3. Μελέτη Περίπτωσης

3.1 Μελέτη Περίπτωσης

Στο κεφάλαιο αυτό θα αναφερθούν και θα αναλυθούν τόσο η περιπτωσιολογική μελέτη (case study) όσο και τα δεδομένα που χρησιμοποιήθηκαν για να λειτουργήσει σωστά και ομαλά η εν λόγω εργασία.

Η μελέτη περίπτωσης ή περιπτωσιολογική μελέτη, είναι η μια μεθοδολογία έρευνας, η οποία αναπτύσσεται σε βάθος και επεξηγεί ή περιγράφει ένα στιγμιότυπο ενός προβλήματος, δηλαδή μια περίπτωση. Στο πλαίσιο αυτό το πρόβλημα που θίγεται όπως έχει περιγραφεί και στο κεφάλαιο της εισαγωγής είναι η πρόβλεψη επόμενης τοποθεσίας ενός ατόμου σε εσωτερικό χώρο. Η ειδική περίπτωση που μελετάτε είναι η πρόβλεψη της μελλοντικής θέσης ορισμένων πελατών όπως αυτοί κινούνται σε ένα μεγάλο εμπορικό κέντρο έχοντας σαν γνώση τις προηγούμενες τους θέσεις. Ο τρόπος που κάτι τέτοιο επιτυγχάνεται στην παρούσα εργασία είναι μέσω ορισμένων αλγορίθμων μηχανικής μάθησης οι οποίοι με την σειρά τους εφαρμόζονται στα ανοιχτά δεδομένα που έχουμε διαθέσιμα και τα οποία θα αναλυθούν στο αμέσως επόμενο κομμάτι του κεφαλαίου.

Η μελέτη και σύγκριση αυτών των αλγορίθμων ως προς την απόδοση τους είναι σημαντική καθώς παρότι είναι γνωστό ότι ο καθένας ξεχωριστά μπορεί να δώσει λύση στο πρόβλημα της πρόβλεψης τοποθεσίας, δεν είναι απαραίτητα ξεκάθαρο ποιος στην πραγματικότητα είναι ο καλύτερος. Πιο συγκεκριμένα, σχολιάζεται το γεγονός της καταλληλόλητας κάποιου από τους αλγορίθμους, καθώς κάτι τέτοιο έχει να κάνει με την περίπτωση εφαρμογής τους αλλά και με τις παραμέτρους που δέχεται σαν είσοδο. Επιπλέον, πρέπει να αναλογιστεί κανείς, όταν αναφέρεται η καταλληλόλητα των αλγορίθμων, πως για κάποιους είναι κατάλληλος ένας αλγόριθμος αν έχει μεγαλύτερη ακρίβεια από άλλον ενώ για κάποιους άλλους αν το αποτέλεσμα του βγαίνει σε πολύ πιο γρήγορο χρόνο. Στην συγκεκριμένη έρευνα το κριτήριο που βασικά ενδιαφέρει είναι η ακρίβεια που καταφέρνει κάθε αλγόριθμος καθώς δεν υπάρχει κάποιο χρονικό όριο ή περιορισμός που να δεσμεύει. Αυτό που έχει μεγάλη σημασία όμως είναι το πόσο αυξάνεται ή ενδεχομένως μειώνεται η ακρίβεια κάθε αλγορίθμου αν σε αυτόν δοθούν σαν είσοδο περισσότερες παράμετροι από το απλό μοντέλο της γνώσης μόνο της προηγούμενης τοποθεσίας.

Άρα, ανακεφαλαιώνοντας, στην ερευνά αυτή γίνεται προσπάθεια για επίλυση του προβλήματος της πρόβλεψης επόμενης τοποθεσίας των πελατών, που κινούνται σε ένα εμπορικό κέντρο, εφαρμόζοντας αλγορίθμους πρόβλεψης σε ανοιχτά δεδομένα και στη συνέχεια ο σχολιασμός και η σύγκριση των αποτελεσμάτων τους και συγκεκριμένα της ακρίβειας τους.

3.2 Περιγραφή Δεδομένων

Στο σημείο αυτό, είναι σημαντικό να αναφερθούν και στη συνέχεια να αναλυθούν τα ανοιχτά δεδομένα που χρησιμοποιήθηκαν για να έρθει εις πέρας η εφαρμογή των αλγορίθμων και κατά συνέπεια ολόκληρη η διπλωματική εργασία.

Τα δεδομένα λήφθηκαν από τον ιστότοπο Kaggle[1]. Εκεί, παρατίθενται ανοιχτά δεδομένα που αφορούν παρατηρήσεις που συλλέχθηκαν από το μεγάλο εμπορικό κέντρο στο Σαν Φραντσίσκο της Καλιφόρνια με όνομα Westfield. Το αρχείο είναι διαθέσιμο μέσω του χρήστη και επιστήμονα Rahul Benal και αποτελείται από τέσσερα αρχεία της μορφής csv με ονόματα category_mapping.csv, domographic.csv, ping_info.csv και store_mapping.csv. Παρακάτω αναλύεται το περιεχόμενο του κάθε αρχείο ώστε ο αναγνώστης να καταλάβει ακριβώς την δομή και το περιεχόμενο των δεδομένων αυτών και κατά συνέπεια να του είναι πιο εύκολη η κατανόηση της μοντελοποίησης των αλγορίθμων.

Το αρχείο category_mapping.csv αποτελείται ουσιαστικά από έναν πίνακα 154 γραμμών και 3 στηλών. Οι 3 στήλες είναι οι: Store_name, Broad_Category και Fine_Category ενώ οι 154 γραμμές είναι οι μοναδικές "τιμές" των μαγαζιών που στεγάζονται μέσα στο εμπορικό κέντρο. Εδώ αξίζει να σημειωθεί ότι τα μαγαζιά είναι στην πραγματικότητα περισσότερα από 154 αλλά κάποια έχουν περισσότερα από ένα καταστήματα στον τεράστιο αυτό χώρο του εμπορικού ή κάποια εκτείνονται σε πάνω από έναν όροφο του κέντρου. Η πρώτη στήλη λοιπόν όπως λέει περιγραφικά το όνομα της είναι το όνομα κάθε καταστήματος (από μία φορά). Η δεύτερη στήλη δηλαδή η Broad_Category αποτελείται από 2 όρους, Retail και Dining που ουσιαστικά προσδιορίζει αν ένα κατάστημα του κέντρου είναι χώρους εστίασης ή χώρος αγοράς προιόντων. Η τρίτη στήλη είναι αυτή που προσδιορίζει πραγματικά την φύση του κάθε μαγαζιού με πιο συγκεκριμένους όρους όπως για παράδειγμα Speciality Store, Department Stores, Apparel and Accessories, Casual Restaurants και άλλα. Ο 154 x 3 πίνακας αυτός δεν θα χρησιμέψει ιδιαίτερα στην παρούσα διπλωματική αφού κάθε μαγαζί μπορεί να αντιπροσωπευτεί από ένα νούμερο και πάλι η ακρίβεια να μετρηθεί σωστά σαν ποσοστό. Αυτό είναι λογικό αφού ο αλγόριθμος μας δεν θα αντιλαμβάνεται κάτι άλλο πέρα από ένα νούμερο όταν θα πάρει μια είσοδο και θα δώσει τελικά μια έξοδο. Παρακάτω φαίνεται μια ενδεικτική φωτογραφία που δείχνει κάποια δεδομένα του συγκεκριμένου αρχείου.

[1] https://www.kaggle.com/rahulbenal/business-case-study-shopping-centre#store_mapping.csv

category_mapping.csv (6.07 KB) 3 of 3 columns Views

	Store_Name	Broad_Category	Fine_Category
	154 unique values	Retail 76% Dining 24%	Apparel And Ac... 32% Qsr Restaurants 23% Other (12) 45%
1	Amazon Pop-Up	Retail	Misc.
2	Bespoke Events	Retail	Misc.
3	Bespoke Coworking	Retail	Specialty Store
4	Dyson	Retail	Misc.
5	Penhaligon$s	Retail	Misc.
6	SF Sports	Retail	Apparel And Accessories
7	Claire's Boutique	Retail	Apparel And Accessories
8	Go! Toys and Games	Retail	Toy Stores
9	Nordstrom	Retail	Department Stores
10	Bloomingdale's	Retail	Department Stores
11	Victoria's Secret	Retail	Apparel And Accessories
12	Adidas	Retail	Apparel And Accessories
13	Finish Line	Retail	Apparel And Accessories
14	Zara	Retail	Apparel And Accessories
15	H&M	Retail	Apparel And Accessories
16	H&M Man	Retail	Apparel And Accessories
17	Bristol Farms	Dining	Casual Restaurants
18	Superdry	Retail	Apparel And Accessories
19	Sunglass Hut	Retail	Apparel And Accessories
20	Express	Retail	Apparel And Accessories

Εικόνα 1 category_mapping.csv

Στη συνέχεια έχουμε το αρχείο domografic.csv το οποίο αποτελείται από 567 γραμμές και 7 στήλες. Αυτό σημαίνει ότι συγκριτικά με το προηγούμενο αρχείο μας, αυτός ο πίνακας είναι πολύ μεγαλύτερος και φαντάζει πολύ πιο περίπλοκος στην κατανόηση του. Στην πραγματικότητα όμως αναφέρεται στα δημογραφικά χαρακτηριστικά όλων των πελατών του καταστήματος που έλαβαν μέρος στην μέτρηση και καταγραφή των θέσεων τους. Εδώ αξίζει να σημειωθεί πως δεν γνωρίζουμε ακριβώς με ποια ακριβώς τεχνολογία μετρήθηκαν οι θέσεις των πελατών, ωστόσο, όπως θα δούμε και παρακάτω, διαθέτουμε ακριβέστατες και κατατοπιστικότατες θέσεις σε δεδομένες χρονικές στιγμές. Έτσι, όπως γίνεται αντιληπτό το αρχείο περιλαμβάνει στις γραμμές του, τους 567 πελάτες που συμμετείχαν με την πρώτη στήλη από τις 7 να περιέχει απλώς τα ψευδώνυμα αυτών (Shopper_ID) με τους όρους Shopper_1 – Shopper_567. Η δεύτερη στήλη περιέχει έναν κωδικό (ID) για κάθε έναν πελάτη ο οποίος αποτελείται τόσο από νούμερα όσο και από λατινικούς χαρακτήρες, στοιχείο που στην έρευνα μας δεν θα διαδραματίσει σημαντικό ρόλο. Αμέσως μετά ακολουθούν άλλες 5 στήλες οι οποίες περιέχουν καθαρά δημογραφικά στοιχεία για κάθε πελάτη. Η τρίτη στήλη έχει τις ηλικίες (Age) του καθενός οι οποίες κυμαίνονται από τα 19 χρόνια μέχρι τα 93 πράγμα που μας ενδιαφέρει ιδιαίτερα στην εφαρμογή των αλγορίθμων καθώς όταν ο κάθε αλγόριθμος

τρέχει με παράμετρο την ηλικία λαμβάνει υπόψη και βγάζει διαφορετικό αποτέλεσμα για κάθε ηλικιακή ομάδα ατόμων. Στην τέταρτη στήλη υπάρχει το φύλο (Gender) του κάθε ατόμου (M / F), ενώ στην πέμπτη η συζυγική τους κατάσταση (Marital_Status) που για χάρη της έρευνας θεωρήθηκε ότι παίρνει τις τιμές Married (M), Single (S), Divorced (D). Στη συνέχεια , η έκτη στήλη δίνει την πληροφορία για την κατοχή σπιτιού (Owns_Home) από τον πελάτη. Έτσι, περιλαμβάνει τις τιμές Yes (Y) ή No (N) και αποτελεί παγίδα ως προς την σημασία του στην έρευνα μας. Αυτό συμβαίνει διότι εκ πρώτης όψεως φαντάζει αμελητέα πληροφορία για τον κάθε πελάτη, όμως στην πραγματικότητα η κατοχή ή μη, οικίας, αλλάζει την κινητική συμπεριφορά και άρα την πορεία του παρατηρούμενου ατόμου. Τέλος, η έβδομη και τελευταία στήλη περιλαμβάνει τον αριθμό των παιδιών κάτω από την ηλικία των 18 ετών (Number_of_Children_under_18_years_of_age) που διαθέτει κάθε άτομο, με τις τιμές να κυμαίνονται από 0 μέχρι 5 παιδιά με το ανώτερο όριο των 5 παιδιών να καθορίζεται τελείως τυχαία με βάση το διαθέσιμο μας δείγμα. Παρακάτω φαίνεται η όψη του συγκεκριμένου πίνακα :

domografic.csv (33.76 KB)

	Shopper_Id	ID	Age	Gender	Marital_Status	Owns_Home	Number_of_Children
	567 unique values	567 unique values	19 – 93	M 53%, F 47%	M 54%, S 39%, Other (1) 7%	Y 60%, N 40%	0 – 5
1	Shopper_1	831b9564-99d7-3b57-a9c0-3ac1f31560d5	23	F	D	N	1
2	Shopper_2	1bd5093b-5a76-3ccc-b0d2-37695a858a95	42	F	M	N	1
3	Shopper_3	d08139f8-c50d-3472-adf8-6dcc4724164c	51	F	M	N	1
4	Shopper_4	29b285af-f631-3100-9a4e-5de09df7f974	35	F	D	N	2
5	Shopper_5	63fc3248-9900-3074-b887-c6ed6d73d0e3	42	M	S	N	0
6	Shopper_6	3fe4135c-06c1-395c-923d-f10f3552dd75	50	M	M	Y	1
7	Shopper_7	725954b6-5227-3d88-b6bf-8a6b822dd8d4	40	M	M	Y	0
8	Shopper_8	c0b4b558-84e1-3400-8f6f-18db12478292	58	M	M	Y	0
9	Shopper_9	eca4c7c4-7b92-3648-ba55-f84020ced575	36	F	M	N	2
10	Shopper_10	39a86f65-f08d-330b-8d37-57d082631adf	56	F	M	Y	1
11	Shopper_11	178732a0-709f-37de-8188-becd7d4f859f	23	F	S	N	0
12	Shopper_12	b78feb44-2b0b-3131-9935-f0802c08db3f	61	M	M	Y	0
13	Shopper_13	9cc20ae7-6b44-3c97-9f51-76789bbcf275	76	M	M	Y	2
14	Shopper_14	43eefca9-9f82-3adc-8efe-decc03100807	50	M	M	Y	2
15	Shopper_15	00d93a11-9304-345c-917a-c9be3ff4e999	25	F	M	Y	0
16	Shopper_16	e4b8f1dd-cd15-3477-8b38-f12b22c1e00d	71	M	M	Y	0
17	Shopper_17	1d367279-918a-3a87-9fe4-148b0b57fbaa	66	F	M	N	0

Εικόνα 2 domographic.csv

Το τρίτο αρχείο που είναι διαθέσιμο είναι το store_mapping.csv. Στο αρχείο αυτό όπως φανερώνεται από τον τίτλο του διαθέτει τα καταστήματα που υπάρχουν στο εμπορικό κέντρο. Μια σημαντική διαφορά του με το category_mapping.csv ως προς τα καταστήματα είναι ότι εδώ τα ονόματα των καταστημάτων δεν είναι μοναδικά (unique). Αυτό σημαίνει ότι αν κάποιο κατάστημα, όπως αναφέρθηκε προηγουμένως,

βρίσκεται σε πάνω από έναν όροφο, τότε θα συμπεριληφθεί αντίστοιχες φορές σαν ξεχωριστή γραμμή στον πίνακα. Έτσι, έχουμε 166 γραμμές-καταστήματα και 5 στήλες. Η πρώτη στήλη περιέχει το όνομα του κάθε καταστήματος (Store_Name) ,δηλαδή 154 διαφορετικά καταστήματα με 166 τιμές στον πίνακα ανάλογα με τον όροφο που εδρεύουν. Η δεύτερη στήλη περιλαμβάνει τον το όνομα του κάθε ορόφου του καταστήματος. Εδώ υπάρχουν 9 διαφορετικές τιμές : Concourse Level, Level 1, Level 2, Level 3, Level 4/N1, Level 5/N2, N3, N4 και N5. Στην τρίτη στήλη υπάρχει απλά ο δείκτης κάθε ορόφου δηλαδή μια τιμή σε κάθε όροφο ανάμεσα στο 0 και το 8 (9 τιμές), κάτι το οποίο προφανώς θα χρειαστεί προγραμματιστικά αφού όπως είπαμε ο αλγόριθμος καταλαβαίνει αριθμούς και όχι γράμματα. Οι τελευταίες δύο στήλες 4 και 5 του csv αρχείου έχουν την σημαντική πληροφορία του αρχείου. Εδώ περιέχονται αντίστοιχα τα γεωγραφικά πλάτη και μήκη του κέντρου του κάθε καταστήματος στην παγκόσμιο χάρτη (latitude, longitude). Οι τιμές αυτές είναι όλες πολύ κοντα στο γεωγραφικό πλάτος 37.8 και στο γεωγραφικό μήκος -122.4, κάτι που φαντάζει απόλυτα λογικό αφού το παράδειγμα περίπτωσης που ασχολείται η εργασία έχει έδρα του το Σαν Φραντσίσκο της Καλιφόρνια. Ακολουθεί ο Πίνακας του αντίστοιχου csv.

store_mapping.csv (7.98 KB)

	Store_Name	Floor_name	Floor_Index	latitude	longitude
	Bloomingdale's 3%	Concourse Level 29%	0 – 8	37.8 – 37.8	-122 – -122
	Nordstrom 3%	Level 1 22%			
	Other (152) 94%	Other (7) 49%			
1	Jamba Juice	Concourse Level	0	37.784091	-122.406813
2	Coriander Gourmet Thai	Concourse Level	0	37.784596	-122.40621
3	NYS Collection	Concourse Level	0	37.784396	-122.406587
4	Hallmark	Concourse Level	0	37.783771	-122.407377
5	Sorabol Korean BBQ & Asian Noodles	Concourse Level	0	37.784601	-122.406328
6	Pasta Moto	Concourse Level	0	37.784534	-122.406025
7	Abercrombie & Fitch	Concourse Level	0	37.783944	-122.407537
8	Fire of Brazil	Concourse Level	0	37.784447	-122.406224
9	ProActiv Solutions	Concourse Level	0	37.784161	-122.4069
10	Corner W	Concourse Level	0	37.784808	-122.406501
11	Buckhorn Grill	Concourse Level	0	37.784569	-122.406141
12	Cako Bakery	Concourse Level	0	37.78467	-122.406679
13	Claire's Boutique	Concourse Level	0	37.784126	-122.407061
14	Bristol Farms	Concourse Level	0	37.784159	-122.40635
15	PacSun	Concourse Level	0	37.783959	-122.40691
16	DAVIDsTEA	Concourse Level	0	37.784411	-122.406709
17	Andale Mexican Restaurant	Concourse Level	0	37.784511	-122.405956
18	Sunglass Hut	Concourse Level	0	37.784071	-122.406885
19	Peet's Coffee & Tea	Concourse Level	0	37.784591	-122.406782
20	Panda Express	Concourse Level	0	37.783778	-122.407161
21	GNC	Concourse Level	0	37.784053	-122.407307
22	New Era	Concourse Level	0	37.783893	-122.407389
23	LEGO	Concourse Level	0	37.784222	-122.406978
24	Mrs. Fields	Concourse Level	0	37.783953	-122.407318
25	Beard Papa's Cream Puffs	Concourse Level	0	37.784505	-122.406555

Εικόνα 3 store_mapping.csv

Τελευταίο αλλά πιο σημαντικό από όλα τα αρχεία που σχολιάστηκαν μέχρι στιγμής είναι αυτό με όνομα ping_info.csv. Το συγκεκριμένο csv είναι στην ουσία το σύνολο των παρατηρήσεων μας, δηλαδή όλες οι συνεχόμενες θέσεις των πελατών που σχηματίζουν με τη σειρά τους την λεγόμενη πορεία (trejectory) του καθενός. Όπως είναι λογικό όλοι οι αλγόριθμοι δέχονται σαν εισοδό τους στοιχεία κυρίως από το εν λόγω αρχείο. Συγκεκριμένα, η πρώτη στήλη ξανά όπως και στο domografic.csv περιέχει τα Shopper_ID με διαφορά ότι τώρα είναι ταξινομημένα κατά αύξουσα σειρά με βάση τον χρόνο που λήφθηκαν οι παρατηρήσεις. Αυτό σημαίνει ότι αν ένας πελάτης επισκέφτηκε το εμπορικό κέντρο 2 φορές με διαφορά ένα μήνα μεταξύ τους τότε οι πορείες του θα χωριστούν σε διαφορετικά σκέλη τα οποία θα τα ονομάσουμε επισκέψεις (visits). Κάθε πελάτης μπορεί να έχει τουλάχιστον ένα visit στο εμπορικό μας και άρα μπορεί να εμφανιστεί είτε μία είτε περισσότερες φορές στην $1^{η}$ στήλη με σειρά ανάλογα με τη στιγμή που διέπραξε τις πορείες του. Η $2^{η}$ στήλη αφορά τον όροφο στον οποίο παρατηρήθηκε κίνηση οποιουδήποτε πελάτη (π.χ Concourse Level κτλπ) όπως αναλύθηκε για προηγούμενο αρχείο. Οι βασικές στήλες που αποτελούν και το πιο σημαντικό κομμάτι της εργασίας είναι οι γεωγραφικές θέσεις στις οποίες καταγράφηκαν παρατηρήσεις κίνησης. Προφανώς οι τοποθεσίες αυτές αφορούν το εμπορικό μας κέντρο και άρα κινούνται στα ίδια πλαίσια που κυμαίνονταν και οι γεωγραφικές θέσεις των κέντρων των καταστημάτων στο αρχείο store_mapping.csv. Ωστόσο να σημειωθεί πως οι συντεταγμένες γεωγραφικού πλάτους και μήκους στις στήλες 3 και 4 δεν είναι αντιστοιχίζονται ακριβώς με αυτές των καταστημάτων καθώς όπως είναι λογικό οι πιθανότητες η κίνηση ενός πελάτη να καταγραφεί ακριβώς τη στιγμή που θα περάσει από το κέντρο ενός καταστήματος είναι μηδαμινή. Στην έβδομη και τελευταία στήλη του πίνακα παραθέτονται οι χρονικές στιγμές στις οποίες καταγράφηκαν οι παρατηρήσεις μας. Η μορφή των τιμών των χρονικών στιγμών είναι η εξής : 2016-12-19T01:31:08.000Z, δηλαδή έτος, μήνας, ημέρα, ώρα, λεπτά, δευτερόλεπτα. Όπως παρατηρείτε εύκολα κάποιες χρονικές παρατηρήσεις έχουν διαφορά μήνες μεταξύ τους πράγμα που δεν αποτελεί έκπληξη αφού είναι ανθρωπίνως εφικτό και λογικό κάποιος άνθρωπος να επισκέφθηκε το εμπορικό μόλις 2 φορές σε διάστημα 3 μηνών. Οι τελευταίες 2 στήλες δηλαδή η $5^{η}$ και η $6^{η}$ δεν λήφθηκαν υπόψη καθώς δίνουν απλώς την τοποθεσία του πελάτη με βάση διαφορετικές συντεταγμένες. Έτσι, καθαρά για θέμα απλότητας και ομοιομορφίας με τα αντίστοιχα χαρακτηριστικά του store_mapping.csv επιλέχθηκαν τα γεωγραφικά μήκη και πλάτη για το προσδιορισμό της τοποθεσίας των συμμετεχόντων. Στο σημείο αυτό τονίζεται πως το χρονικό διάστημα των παρατηρήσεων είναι το διάστημα από 2 Οκτωβρίου του 2016 μέχρι 13 Ιουνίου 2017 ενώ οι διαφορετικές θέσεις που καταγράφηκαν και ουσιαστικά αποτελούν και τις γραμμές του εν λόγω csv αρχείου ανέρχονται σε περίπου 108 χιλιάδες.

Παρακάτω παρατίθεται η εικόνα που δείχνει τη μορφή του ping_info.csv.

ping_info.csv (9.81 MB)

	Shopper_ID	floor	lat	lng	x	y	seenTimestamp
	568 unique values	Concourse 34% Floor 1 23% Other (4) 43%	0 37.8	-122 0	-8.26k 4.73m	-10.5m 830	2Oct16 13Jun17
1	Shopper_556	Floor 4	37.66146302	-122.2986728	-2934.558251	-16169.72229	2016-11-22T02:16:06.000Z
2	Shopper_556	Floor 1	37.78406211	-122.4069432	117.2624746	151.8238116	2016-12-23T22:01:25.000Z
3	Shopper_556	Floor 4	37.78436125	-122.4067095	156.1250868	171.3572003	2016-11-22T01:42:28.000Z
4	Shopper_360	Concourse	37.78434131	-122.4068907	134.4996733	167.6744011	2016-10-13T18:27:38.000Z
5	Shopper_360	Floor 2	37.78401733	-122.4073072	94.98169298	172.2692432	2016-10-12T21:36:18.000Z
6	Shopper_420	Floor 4	37.66162362	-122.2984593	-2908.686633	-16170.58466	2016-12-19T01:31:08.000Z
7	Shopper_420	Floor 1	37.78385587	-122.4070922	91.86057451	144.6519438	2016-12-19T01:36:10.000Z
8	Shopper_245	Floor 2	37.78397004	-122.407194	98.38964653	161.5518958	2016-10-14T03:02:55.000Z
9	Shopper_334	Floor 2	37.78382762	-122.4069863	100.3071501	137.4773985	2017-03-18T03:25:59.000Z
10	Shopper_205	Floor 1	37.78421876	-122.4068961	132.4179394	161.3388659	2017-06-06T20:33:31.000Z
11	Shopper_205	Concourse	37.78463814	-122.406411	187.6740638	161.6688386	2017-06-06T01:28:51.000Z
12	Shopper_530	Floor 1	35.08514873	-113.6637341	337611.758	-752026.1402	2017-03-08T23:50:12.000Z
13	Shopper_530	Concourse	36.54127145	-118.4354617	152099.1346	-342852.7373	2017-01-28T00:07:15.000Z
14	Shopper_359	Concourse	37.78396128	-122.407034	95.92117176	146.3723637	2017-05-20T18:34:46.000Z
15	Shopper_189	Concourse	35.23997601	-114.2209897	314753.9165	-705450.5711	2017-01-17T23:56:19.000Z
16	Shopper_189	Concourse	37.78404899	-122.4070545	101.4707622	154.5846606	2017-01-17T02:40:49.000Z
17	Shopper_189	Concourse	37.78408429	-122.4070442	104.8637975	156.7499413	2017-01-17T01:57:58.000Z

Εικόνα 4 ping_info.csv

3.3 Ορισμοί

Στο κομμάτι αυτό εξηγούνται κάποιοι από τους όρους που ακολουθούν στην συνέχεια, η γνώση των οποίων είναι σημαντική για την κατανόηση της εν λόγω διπλωματικής εργασίας.

Machine Learning: Η μηχανική μάθηση ή machine learning όπως θα αποκαλείται στην εργασία είναι ένα υποπεδίο της επιστήμης των υπολογιστών που αναπτύχθηκε από τη μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη. Σε μια σύντομη ιστορική αναδρομή, το 1959, ο Άρθουρ Σάμουελ όρισε τη μηχανική μάθηση ως «Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί». Ο επιστημονικός λοιπόν κλάδος του machine learning διερευνά την μελέτη και την κατασκευή αλγορίθμων που μπορούν να μαθαίνουν από τα δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά. Τέτοιοι αλγόριθμοι λειτουργούν κατασκευάζοντας μοντέλα από πειραματικά δεδομένα, προκειμένου να κάνουν προβλέψεις βασιζόμενες στα δεδομένα ή να εξάγουν αποφάσεις που εκφράζονται ως το αποτέλεσμα. Στο πεδίο της ανάλυσης δεδομένων, το machine learning είναι μια μέθοδος που χρησιμοποιείται για την επινόηση μοντέλων και αλγορίθμων που οδηγούν

στην πρόβλεψη. Τα αναλυτικά μοντέλα επιτρέπουν στους ερευνητές, τους επιστήμονες δεδομένων, τους μηχανικούς και τους αναλυτές να παράγουν αξιόπιστες αποφάσεις και αποτελέσματα και να αναδείξουν αλληλοσυσχετίσεις μέσω της μάθησης από ιστορικές σχέσεις και τάσεις στα δεδομένα.

Deep Learning: Το deep learning είναι ένα κομμάτι της ευρύτερης οικογένειας του machine learning και αποτελεί μια μέθοδο που βασίζεται στα τεχνητά νευρωνικά δίκτυα (neural networks). Ο όρος artificial neural networks (ANN) είναι εμπνευσμένος από τα βιολογικά συστήματα και συγκεκριμένα από την διαδικασία της πληροφορίας σε αυτά και την μετάδοση και επικοινωνία της γνώσης μεταξύ των δομών του βιολογικού συστήματος. Η διαφοροποίηση του deep learning από το machine learning έγκειται στο γεγονός ότι τα πρώτα χρησιμοποιούν πολλαπλά στρώματα (layers) ώστε σταδιακά να παράξουν πιο συγκεκριμένα και με μεγαλύτερα ακρίβεια χαρακτηριστικά για το αρχικά απλό δεδομένο που δόθηκε σαν είσοδο. Στην ουσία λοιπόν στα νευρωνικά δίκτυα άρα και στο deep learning παρατηρείται το φαινόμενο της αυτόματης εκμάθησης μέσω των διαφορετικών layers της πληροφορίας από το ίδιο το σύστημα με την πληροφορία να παρέχεται από τον χρήστη μόνο στο αρχικό, ωμό της στάδιο (raw input).

Haversine Formula: Όπως αναφέρθηκε στο κεφάλαιο που παρατίθενται τα ανοιχτά δεδομένα, οι γεωγραφικές θέσεις των πελατών δίνονται στη μορφή lat και Long δηλαδή γεωγραφικού πλάτους και μήκους αντίστοιχα. Αυτό σημαίνει πως αν κάποιος θέλει να βρει την πραγματική απόσταση 2 σημείων στην σφαιρική γη, δεν αρκεί απλά να αφαιρέσει τα τετράγωνα των lat και Long όπως ισχύει στην Ευκλείδεια γεωμετρία. Τα δεδομένα εμπίπτουν στο χώρο της σφαιρικής τριγωνομετρίας και άρα για τον υπολογισμό της λεγόμενης great-circle distance, δηλαδή της πραγματικής απόστασης πάνω στη σφαιρική γη απαιτείται η εφαρμογή της φόρμουλας Haversine η οποία εφαρμόζει συγκεκριμένο τύπο για τον υπολογισμό της, δοσμένων των lat και long. Χωρίς να αναμειχθούν σοβαρά μαθηματικά, παρατίθεται η φόρμουλα όπως αυτή αποτυπώθηκε στον κώδικα Python.

```
# haversine formula
dlon = lon2 - lon1
dlat = lat2 - lat1
a = math.sin(dlat / 2) ** 2 + math.cos(lat1) * math.cos(lat2) * (math.sin(dlon / 2) ** 2)
c = 2 * math.asin(math.sqrt(a))
r = 6371  # Radius of earth in kilometers. Use 3956 for miles
return c * r * 1000
```

Εικόνα 5 haversine απόσταση

3.4 Παραδοχές

Στο σημείο αυτό αναφέρονται και σχολιάζονται ορισμένες παραδοχές ή θεωρήσεις που έγιναν στο πλαίσιο της προετοιμασίας των δεδομένων αλλά και στο πλαίσιο της εφαρμογής αλγορίθμων και συγκομιδής αποτελεσμάτων τους. Να σημειωθεί βέβαια πως παρά το γεγονός ότι το να γίνονται παραδοχές σε ένα πρόβλημα δεν είναι αυθαίρετο και τυχαίο, πρέπει να υπάρχει μεγάλη προσοχή οι εν λόγω

παραδοχές να έχουν μια δόση κοινής λογικής και να αιτιολογούνται τεκμηριωμένα και σωστά.

Έτσι, ξεκινώντας από το κομμάτι της προετοιμασίας των δεδομένων, όπως αναφέραμε και στο αντίστοιχο κεφάλαιο, λήφθηκαν υπόψη για τον προσδιορισμό των τοποθεσιών μόνο οι σφαιρικές γεωγραφικές συντεταγμένες lat και long. Αυτό βέβαια δεν προκαλεί βλάβη στη γενικότητα αφού η διαφορά της απόστασης θα βγει ίδια όποια μονάδα μέτρησης και να χρησιμοποιηθεί αν πρώτα γίνουν οι απαραίτητες μετατροπές. Απλώς στην παρούσα περίπτωση χρειάστηκαν η εφαρμογή της φόρμουλας Haversine.

Όπως αναφέρθηκε πριν, διαθέσιμα από τα ανοιχτά δεδομένα είναι τα κέντρα των καταστημάτων. Για τους πελάτες είναι διαθέσιμα τα σημεία που παρατηρήθηκαν χωρίς όμως αυτά να ταιριάζουν ακριβώς με κάποιο κέντρο καταστήματος. Έτσι για να αντιστοιχηθεί κάθε πελάτης σε κάποιο μαγαζί θα πρέπει να ληφθούν πολλές παράμετροι υπόψη. Μια πρώτη λύση είναι να θεωρηθεί πως ο πελάτης ανήκει σε όποιο μαγαζί η απόσταση του από το κέντρο του είναι η ελάχιστη σύμφωνα πάντα με την φόρμουλα Haversine. Κάτι τέτοιο όμως παρουσιάζει το εξής πρόβλημα : αν ο πελάτης βρίσκεται στην άκρη κάποιο μεγάλου μαγαζιού και δίπλα από αυτό το μαγαζί στεγάζεται ένα μικρό μαγαζί τότε ο πελάτης φαίνεται να βρίσκεται πιο κοντά στο διπλανό κατάστημα από ότι στο κέντρο του καταστήματος που είναι πραγματικά και άρα να γίνει έτσι λάθος αντιστοίχιση. Κάτι τέτοιο θα μπορούσε να διορθωθεί ή να αποφευχθεί αν θεωρούσαμε συγκεκριμένη ακτίνα που καταλαμβάνουν τα καταστήματα. Ωστόσο από τη μία μεριά κάθε κατάστημα δεν έχει το ίδιο μέγεθος με το άλλο οπότε μια κοινή ακτίνα σε όλα τα μαγαζιά θα οδηγούσε είτε σε επικαλύψεις μαγαζιών είτε σε κενά σημεία που δεν θα ανήκαν σε κανένα μαγαζί ενώ θα έπρεπε. Θα μπορούσε ενδεχομένως να θεωρηθεί διαφορετική ακτίνα για κάθε κατάστημα, πράγμα όμως πολύ δύσκολα αφού από τα δεδομένα δεν έχουμε επαρκή στοιχεία για να προσδιορίσουμε κάτι τέτοιο. Ακόμα, όμως κι αν εφαρμοζόταν διαφορετικά ακτίνα για κάθε κατάστημα με κάποιο τρόπο υπάρχει το εξής απλούστατο πρόβλημα : όταν αναφέρεται κανείς σε κατάστημα εμπορικού δεν αναφέρεται απαραίτητα σε κυκλική κατασκευή που μπορεί ουσιαστικά να αντιπροσωπευθεί από μια κάποια ακτίνα αφού τα περισσότερα μάλιστα μαγαζιά από όσο γνωρίζουμε έχουν τελείως διαφορετικό και ανομοιόμορφο σχήμα από κυκλικό. Βέβαια μετά από αρκετή αναζήτηση και δοκιμές παρατηρήθηκε πως η πρώτη περίπτωση να αντιστοιχίζεται ο πελάτης στο πιο κοντινό από αυτόν μαγαζί φάνηκε να βγάζει καλύτερα και πιο σωστά αποτελέσματα.

Τελικά στην εργασία αυτή εφαρμόστηκαν οι εξής 2 παραδοχές.

Περίπτωση 1: Ο κάθε πελάτης θα αντιστοιχίζεται στο μαγαζί που είναι πιο κοντά με βάση την πραγματική του απόσταση από το κέντρο του.

Περίπτωση 2: Ο κάθε πελάτης θα αντιστοιχίζεται στο κατάστημα που είναι πιο κοντά κατά ελάχιστη απόσταση αλλά με επιπλέον κριτήριο η απόσταση αυτή να είναι μικρότερη από κάποια συγκεκριμένα μέτρα που για την εν λόγω έρευνα αποφασίστηκαν τα 50 μέτρα.

Για την 1η περίπτωση όπως αναφέρθηκε παρότι υπάρχουν κάποια μειονεκτήματα που δεν είναι αμελητέα είναι κατά πολύ καλύτερη σε απόδοση και ακρίβεια από τη λύση των κατά μια έννοια ακτινών. Ωστόσο, μετά από λογική θεώρηση και σκέψη για το πως πραγματικά είναι τα εμπορικά κέντρα, προκύπτει ο εξής

προβληματισμός : τι γίνεται αν κάποιος πελάτης δεν ανήκει πουθενά αλλά κάνει απλώς βόλτα στους διαδρόμους του κτηρίου. Να σημειωθεί εδώ πως για να μην ανήκει ο πελάτης σε κανένα μαγαζί πρέπει η ελάχιστη απόσταση του από κάποιο μαγαζί να είναι αρκετά μεγάλη, δηλαδή το κοντινότερο του μαγαζί να μην είναι «πραγματικά» κοντά. Για να λυθεί το εν λόγω πρόβλημα εξετάστηκαν 2 υποπεριπτώσεις. Η $1^{η}$ προτείνει πως αν κάποιος που πήρε μέρος στην έρευνα δεν βρίσκεται κοντά σε κάποιο μαγαζί δεν θα λαμβάνεται υπόψη καθόλου στις μετρήσεις και τους υπολογισμούς ενώ η $2^{η}$ είναι ουσιαστικά η περίπτωση 2 που αποτελεί την λύση που προτείνεται στην παρούσα διπλωματική εργασία με τα καλύτερα αποτελέσματα από όλες τις υπόλοιπες και λέει ότι αυτός ο πελάτης πρέπει να καταχωρηθεί σε κάποιο μαγαζί. Αν ο πελάτης, λοιπόν δεν ληφθεί καθόλου υπόψη στους υπολογισμούς υπάρχει το πρόβλημα της διαστρέβλωσης των αποτελεσμάτων. Αυτό συμβαίνει επειδή όπως φαίνεται από τα δεδομένα, υπάρχουν πάρα πολλοί πελάτες που έστω και 1 φορά πέρασαν/ καταγράφηκαν σε τοποθεσίες διαδρόμων του εμπορικού και άρα αν δεν ληφθεί κανείς από αυτούς υπόψη τότε στην πραγματικότητα το δείγμα μας θα μειωθεί τόσο πολύ που η πρόβλεψη θα είναι ελλαττωματική ή έστω ελλιπής. Οπότε αφού σίγουρα πρέπει οι πελάτες αυτοί να καταταχθούν κάπου πρέπει να προσδιοριστεί τόσο το που θα καταταχθούν όσο και το μετά από ποια απόσταση θα θεωρούνται ότι δεν ανήκουν σε κάποιο μαγαζί. Η απάντηση στο όριο που πρέπει να δοθεί στην ελάχιστη απόσταση (threshold) δίνεται μέσω μιας τυχαίας προσέγγισης που λαμβάνει υπόψη στο περίπου τις αποστάσεις μεταξύ διαδρόμων και καταστημάτων στα ελληνικά εμπορικά κέντρα και προσαρμόζει αυτήν την απόσταση στα αμερικανικά δεδομένα. Έτσι, σαν threshold επιλέχθηκαν τα 50 μέτρα. Για την απάντηση στο ερώτημα του που θα καταταχθούν οι πελάτες που δεν ανήκουν σε κάποιο κατάστημα προτείνεται το εξής : θεωρούμε ότι η δομή του εμπορικού είναι τέτοια στην οποία όλοι οι πελάτες που δεν ανήκουν σε κάποιο μαγαζί ανήκουν στο σημείο Χ που στην περίπτωση μας αφορά τους διαδρόμους του εμπορικού. Άρα, δώσουμε σε κάθε μαγαζί ένα νούμερο από το 0 μέχρι το 164 (τόσα είναι τα καταστήματα από τα ανοιχτά δεδομένα) τότε αν δεν ανήκει κάποιος άνθρωπος σε κάποιο κατάστημα θα θεωρηθεί ότι το νούμερο που ανήκει είναι για παράδειγμα το 200 ώστε να ξεχωρίσει με τα νούμερα των υπολοίπων καταστημάτων.

Ακόμα, κατά την εφαρμογή των αλγορίθμων όπως αναφέρθηκε χρειάστηκε να δοθεί στους αλγορίθμους σαν παράμετρος και η ηλικία η οποία όπως είναι φυσικό πρέπει με κάποιο τρόπο να ομαδοποιηθεί. Στην παρούσα εργασία θεωρήθηκαν οι ηλικιακές ομάδες [0-26) , [26,35) , [35,42) , [42,49) , [49,56) , [56,64) και 64 και άνω. Η θεώρηση και η παραδοχή για τα ηλικιακά groups έγινε με βάση τις τιμές που είχε ο πίνακας ώστε να μην υπάρχουν πάρα πολλά άτομα στην μια ομάδα και πάρα πολύ λίγα στην άλλη. Επιπλέον αναζητήθηκε και στο διαδίκτυο κατάλληλος ηλικιακός διαχωρισμός και έτσι προέκυψαν οι εν λόγω παραδοχές ώστε να υπάρχει συμβιβασμός ανάμεσα στα δυο.

Τέλος, στο κομμάτι των αποτελεσμάτων που θα αναφερθούν εκτενέστερα σε μετέπειτα κεφάλαιο της εργασίας η βασική παραδοχή που έγινε αφορά τον αλγόριθμο Logistic Regression. Ο εν λόγω machine learning αλγόριθμος είχε ένα ιδιαίτερο χαρακτηριστικό στα αποτελέσματα του συγκριτικά με τους άλλους δύο DBN και CPT. Αυτό ήταν πως ενώ το αποτέλεσμα ήταν αναμενόμενο να έρθει σε ακέραια μορφή,

αφού μιλάμε για καταστήματα με δοσμένες τιμές 0-164 (τιμή 165 για τους διαδρόμους), ο αλγόριθμος έδινε σαν έξοδο δεκαδική μορφή με ένα δεκαδικό ψηφίο μετά την υποδιαστολή. Έτσι, χρειάστηκε να γίνει η εξής παραδοχή : θα αφαιρεθεί από την τιμή πρόβλεψης η πραγματική τιμή και αν η απόλυτη τιμή της διαφοράς τους δώσει αριθμό μικρότερο του 0.5 τότε η πρόβλεψη θεωρείται σωστή αλλιώς απορρίπτεται. Αυτό για να γίνει αντιληπτό με ένα παράδειγμα σημαίνει ότι αν η σωστή τιμή είναι το μαγαζί 123 τότε αν το μοντέλο προβλέψει από 122.5 μέχρι 123.5 η πρόβλεψη θεωρείται ορθή ενώ σε αντίθετη περίπτωση λάθος.

Κεφάλαιο 4. Αλγόριθμοι πρόβλεψης μελλοντικής τοποθεσίας και αποτελέσματα

4.1 Ανάλυση Λύσης Προβλήματος

Στο σημείο αυτό, αναλύεται η προτεινόμενη λύση στα πλαίσια αυτής της εργασίας, στο πρόβλημα της πρόβλεψης επόμενης τοποθεσίας των πελατών που κινούνται στον εσωτερικό χώρο του εμπορικού κέντρου του Σαν Φραντσίσκο.

Η πρόταση που συζητιέται και αναλύεται στην εν λόγω εργασίας είναι η εξής: Εφαρμόζονται διάφοροι αλγόριθμοι στα ίδια ανοιχτά δεδομένα που είναι διαθέσιμα. Στους αλγόριθμους αυτούς δίνονται σαν είσοδο πέρα από τις προηγούμενες θέσεις των πελατών και τα δημογραφικά χαρακτηριστικά τους. Στη συνέχεια γίνονται δοκιμές των αλγορίθμων με τα προαναφερθέντα στοιχεία σαν παραμέτρους και συλλέγονται ορισμένα αποτελέσματα. Μετά όπως είναι φυσικό συγκρίνονται τα αποτελέσματα μεταξύ τους και σχολιάζονται ενδελεχώς ώστε να αποφανθεί κάτω από ορισμένες συνθήκες και παραμέτρους, ποιος έχει την καλύτερη ακρίβεια. Για να παραχθεί το αποτέλεσμα κάθε αλγορίθμου συγκρίνεται η πραγματική επόμενη θέση ενός πελάτη με την τιμή που παρήγαγε ο εκάστοτε αλγόριθμος. Επίσης όλοι οι αλγόριθμοι από τη στιγμή που αναφερόμαστε σε machine learning, απαιτούν 2 στάδια προσαρμογής. Το πρώτο στάδιο είναι αυτό της εκπαίδευσης (training) ενώ το δεύτερο στάδιο είναι αυτό της πρόβλεψης (prediction) . Αναλυτικότερα, κάθε αλγόριθμος machine learning, αλλά και deep learning που δοκιμάστηκε στην εν λόγω εργασία χωρίς όμως τα επιθυμητά αποτελέσματα, απαιτεί ένα αρχικό στάδιο προσαρμογής στα δεδομένα που επιθυμούμε να περάσουμε σαν είσοδο. Αυτό το στάδιο ονομάζεται στάδιο εκμάθησης και για το οποίο θα μιλήσουμε εκτενέστερα στο επόμενο κομμάτι του 4^{ου} κεφαλαίου.

Ακολουθεί πλήρης ανάλυση τόσο των ορισμών κάποιον όρων που χρειάζονται για την κατανόηση από τον αναγνώστη κάποιον σημαντικών εννοιών για την εργασία όσο και ανάλυση των παραδοχών που έγιναν σε επίπεδο προετοιμασίας και σε επίπεδο αποτελεσμάτων. Στη συνέχεια σχολιάζεται η απαραίτητη προετοιμασία που υπέστησαν τα ανοιχτά δεδομένα ώστε να είναι σε θέση να χρησιμοποιηθούν κατά την εφαρμογή των αλγορίθμων. Μετά, αναλύονται οι ίδιοι οι αλγόριθμοι και παρουσιάζεται εκτενέστερα η εφαρμογή τους ενώ στο τέλος του κεφαλαίου αυτού παρουσιάζονται συνοπτικά τα αποτελέσματα που προέκυψαν μετά το σωστό τρέξιμο του κώδικα.

4.2 Προετοιμασία δεδομένων

Εδώ αναφέρονται ορισμένα πράγματα που χρειάστηκαν να τροποποιηθούν στο κομμάτι των δεδομένων ώστε σε επόμενο στάδιο να χρησιμοποιηθούν για την εφαρμογή τους.

Αρχικά όπως ήδη αναφέρθηκε, κάθε κατάστημα αντιπροσωπεύεται από έναν αριθμό από το 0 μέχρι το 164. Έτσι, κάθε πελάτης αντιστοιχίζεται κάθε φορά με ένα μαγαζί ή τους διαδρόμους ανάλογα με την ελάχιστη απόσταση του από κάποιο, με συνέπεια και αυτός να αντιπροσωπεύεται από έναν αριθμό. Η συνάρτηση που δείχνει αυτή τη λειτουργία είναι η εξής:

```
'''
Calculate the nearest store for the given x y in a specific floor
We assume all x,y belong to a store
'''
def GetNearestStore(floor, x,y, storesPerFloor):
    if floor == "Concourse":
        floor = 0
    else:
        floor = int(floor.split()[1])

    stores = storesPerFloor[floor]

    minDistance = 1000000000
    min = None
    for store in stores:
        distance = Distance(x, y, store["lat"], store["lng"])
        if distance < minDistance:
            min = store["id"]
            minDistance = distance
    return min
```

Εικόνα 6 κοντινότερο μαγαζί

Οι αλγόριθμοι ωστόσο στην εργασία, δέχονται σαν είσοδο αλληλουχίες θέσεων δηλαδή συνεχόμενες τοποθεσίες πελατών. Τα στοιχεία αυτά είναι διαθέσιμα σε αρχεία csv δηλαδή πρακτικά είναι ορατά σε μορφή excel. Ωστόσο οι αλγόριθμοι όπως είναι φυσικό δεν δέχονται σαν είσοδο τους τέτοια δεδομένα οπότε έπρεπε όλα να μετατραπούν σε πίνακες ή πινακοειδείς δομές τις Python όπως τα dataframes που όμως για χάρη της εξήγησης της μεθόδου θα αποκαλεστούν πίνακες. Έτσι, δημιουργούνται τρείς πίνακες από τα 2 αρχεία pings_out.csv και store_mapping.csv μέσω των συναρτήσεων CreateStoresMapping, CreateOrGetPings και CreateOrGetVisits. Η 1η συνάρτηση δημιουργεί από το αρχείο store_mapping.csv τον πίνακα storesMapping ο οποίος περιέχει σαν γραμμές όλα τα καταστήματα ενώ οι 4 στήλες του είναι το Store_Name, lng, lat και floor. Η συνάρτηση αυτή φαίνεται στο παρακάτω σχήμα.

```
def CreateStoresMapping(stores):
    floorStoresMapping = {}
    storesMapping = {}
    id = 0
    for num, row in stores.iterrows():
        latitude = row["latitude"]
        longitude = row["longitude"]
        floor = row["Floor_Index"]

        if floor not in floorStoresMapping:
            floorStoresMapping[floor] = []

        floorStoresMapping[floor].append({"lng":  longitude, "lat": latitude, "id": id})
        storesMapping[id] = {"name": row["Store_Name"], "lng":  longitude, "lat": latitude, "floor": floor}

        id += 1
    return storesMapping, floorStoresMapping
```

Εικόνα 7 πίνακας καταστημάτων ανα όροφο

Αντίστοιχα δημιουργείται ο πίνακας pingsMapping από το αρχείο pings.out.csv με χρήση της ελάχιστης απόστασης όπως αναφέρθηκε για την σωστή αντιστοίχιση του πελάτη στο κατάλληλο μαγαζί. Αυτός ο πίνακας περιλαμβάνει σαν στήλες μόνο το Shopper_ID, τον αριθμό του καταστήματος που ανήκει ο πελάτης εκείνη τη στιγμή και το Timestamp, ένας πίνακας που θα λειτουργήσει σαν μεσολαβητής στο να φτιαχτεί ο τελικός πίνακας πετίπτωσης. Ο πίνακας περίπτωσης αλλάζει ανάλογα με την πρόβλεψη που θα γίνει κάθε φορά και στην ουσία είναι ο ίδιος πίνακας με τον μεσολαβητή χωρίς το timestamp αλλά ταξινομημένος κατά timestamp. Τέλος όπως θα αναφερθεί και στη συνέχεια οι αλγόριθμοι χρησιμοποιούν αυτόν τον πίνακα για να δημιουργήσουν το μοντέλο που ζητείται κάθε φορά και τελικά να παράξουν πρόβλεψη.

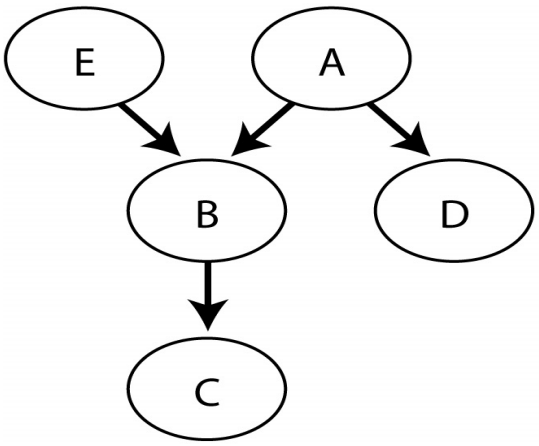
4.3 Ανάλυση Αλγορίθμων

Στην εποενότητα αυτή αναλύονται εκτενέστερα οι αλγόριθμοι που χρησιμοποιούνται στην παρούσα διπλωματική εργασία με σκοπό την παραγωγή πρόβλεψης επόμενης τοποθεσίας των πελατών. Αναφέρονται τόσο οι αλγόριθμοι που εν τέλει έδωσαν λογικό, αναμενόμενο και καλό για τα δεδομένα αποτέλεσμα όσο και οι αλγόριθμοι που δοκιμάστηκαν χωρίς να λαμβάνουν την επιτυχία που θα έπρεπε σε ποσοστό ακρίβειας. Οι αλγόριθμοι, λοιπόν που έτρεξαν σωστά και έβγαλαν αποτέλεσμα είναι ο Dynamic Bayesian Network (DBN), ο Compact Prediction Tree (CPT) και ο Logistic Regression (LR). Η σειρά παρουσίασης είναι με βάση τη σειρά εφαρμογής τους και συγγραφής τους στο κομμάτι του κώδικα. Επιπλέον να σημειωθεί ότι οι αλγόριθμοι αυτοί δοκιμάστηκαν με πολλές παραμέτρους σαν είσοδο και άρα αποτελούν αρκετά καλά δείγματα για σύγκριση μεταξύ τους. Άλλοι αλγόριθμοι που δοκιμάστηκαν ήταν στον τομέα του machine learning ο Hidden Markov Model (HMM), ο Decision Tree Regressor, ο Random Forest Regressor , ο Gradient Boosting Regressor, ο Linear Regression, ο Huber Regression και Polynomial Features χωρίς όμως να δίνουν την κατάλληλη ακρίβεια πρόβλεψης (< 5%). Τα συγκεκριμένα μοντέλα που αναφέρθηκαν και εν τέλει δεν απέδωσαν σημαντικό ποσοστό ακρίβειας κατασκευάστηκαν μέσω της βιβλιοθήκης της Python sklearn και έτρεξαν μέσω των υποβιβλιοθηκών sklearn.tree, sklearn.ensemble, sklearn.linear_model, sklearn.preprocessing. Στον ευρύ χώρο του deep learning η εργασία ασχολήθηκε με την βιβλιοθήκη της Python Keras μέσω της οποίας έτρεξε το μοντέλο Sequential. Στην πράξη το μοντέλο Sequential χρειάζεται για να είναι έτοιμο να τρέξει την εντολή model.compile η οποία δέχεται σαν παραμέτρους τα εξής: loss, optimizer και metrics. Το τελευταίο ουσιαστικά απαιτεί να προσδιοριστεί από τον χρήστη το μέγεθος που τελικά θα μετρηθεί μέσω του δικτύου που δημιουργείται. Η παράμετρος loss αφορά το πως θα χειριστούν τα λάθη, πράγμα που εμπίπτει στο πώς ο αλγόριθμος θα θεωρήσει την λάθος τιμή αν αυτή είναι πολύ κοντά στην πραγματική. Τέλος, ο optimizer που επιλέγεται κάθε φορά δίνει διαφορετικό βάρος στις παρατηρήσεις με τρόπο ώστε να εξομαλύνει κάπως τα δεδομένα. Στην εργασία όπως είναι φυσικό δοκιμάστηκαν διάφοροι optimizers, loss functions και metrics χωρίς όμως τα επιθυμητά αποτελέσματα. Στη συνέχεια ο αλγόριθμος γίνεται trained χρησιμοποιώντας σαν

δεδομένες κάποιες από τις ακολουθίες θέσεων των πελατών με σκοπό να προβλεφθούν οι υπόλοιπες. Ο αριθμός του train data ποικίλει αλλά περίπου κυμαίνεται στις πρώτες 20 χιλιάδες παρατηρήσεις. Να σημειωθεί επίσης ότι ο αλγόριθμος για να δώσει πρόβλεψη πέρα από την εντολή predict που απαιτείται, χρειάζεται ακόμα να δοθεί ο αριθμός των εποχών που θα τρέξει. Οι εποχές μπορούν συνοπτικά να χαρακτηριστούν σαν τις φορές που τρέχει ο αλγόριθμος δηλαδή τις φορές που κάνει train και τις φορές που προβλέπει κάτι. Όπως είναι φυσικό ο αλγόριθμος δεν μπορεί να δώσει τα καλύτερα δυνατά αποτελέσματα αν τρέξει μόλις 1 εποχή αφού όσο περισσότερο εκπαιδευτεί τόσο πιο αποδοτικά θα λειτουργήσει στην πρόβλεψη του. Ωστόσο ακόμα κι αν αυτό φαντάζει λογικό, υπάρχει κάποιο ανώτατο όριο στο οποίο δεν υπάρχει μεγαλύτερη βελτίωση στην ακρίβεια. Στην συνέχεια παρουσιάζονται οι αλγόριθμοι που τελικα κατάφεραν να λειτουργήσουν σωστά και να δώσουν προβλέψεις άξιες προσοχής όσον αφορά την ακρίβεια.

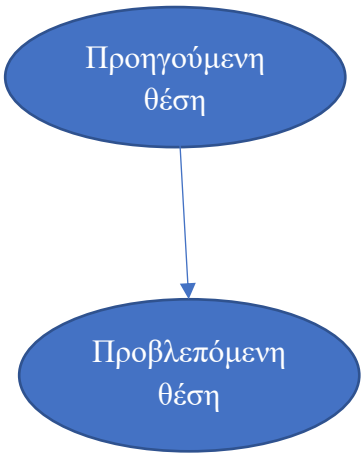
4.3.1 Dynamic Bayesian Model

Το Dynamic Bayesian Network είναι ένα δίκτυο μέσα στο οποίο παρατηρείται συσχετισμός και αλληλεπίδραση ανάμεσα στις μεταβλητές. Έτσι, αν φανταστούμε κάθε κατάσταση που αντιπροσωπεύει μια θέση ενός πελάτη, τότε σε ένα Bayesian network η μια κατάσταση προκύπτει από την προηγούμενη του με βάση την ένωση τους στο αντίστοιχο δέντρο αναπαράστασης. Στο δέντρο αυτό, κάθε κατάσταση – θέση του πελάτη αντιπροσωπεύεται από ένα οβάλ ή κυκλικό σχήμα ενώ η ακμή που ενώνει δύο ή περισσότερα διαφορετικά τέτοια σχήματα, φανερώνει την συσχέτιση μεταξύ των δύο. Παρακάτω φαίνεται ένα κλασσικό σχήμα που δείχνει αυτή τη συσχέτιση μεταξύ των εννοιών A,B,C,D,E. Το σχήμα προφέρεται ως : το B προκύπτει με γνώση του A και του E, το D προκύπτει με γνώση του A και το C προκύπτει με γνώση του B. Στην παρούσα έρευνα η συσχέτιση που ενδιαφέρει είναι αυτή ανάμεσα στις θέσεις στον γεωγραφικό χάρτη των πελατών. Η εξάρτηση που προκύπτει στο παράδειγμα μας είναι αυτή που λέει ότι μια θέση προκύπτει από κάποια συγκεκριμένα πράγματα. Τα πράγματα αυτά στον αλγόριθμο ονομάζονται παράμετροι και ανάλογα με τις παραμέτρους καθορίζεται και το δυναμικό Bayesian δίκτυο που δημιουργείται.

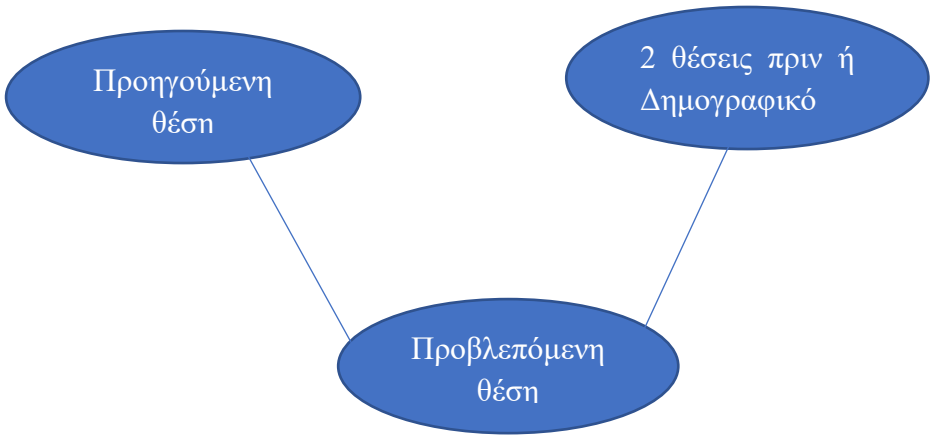


Εικόνα 8 Dynamic Bayesian Network

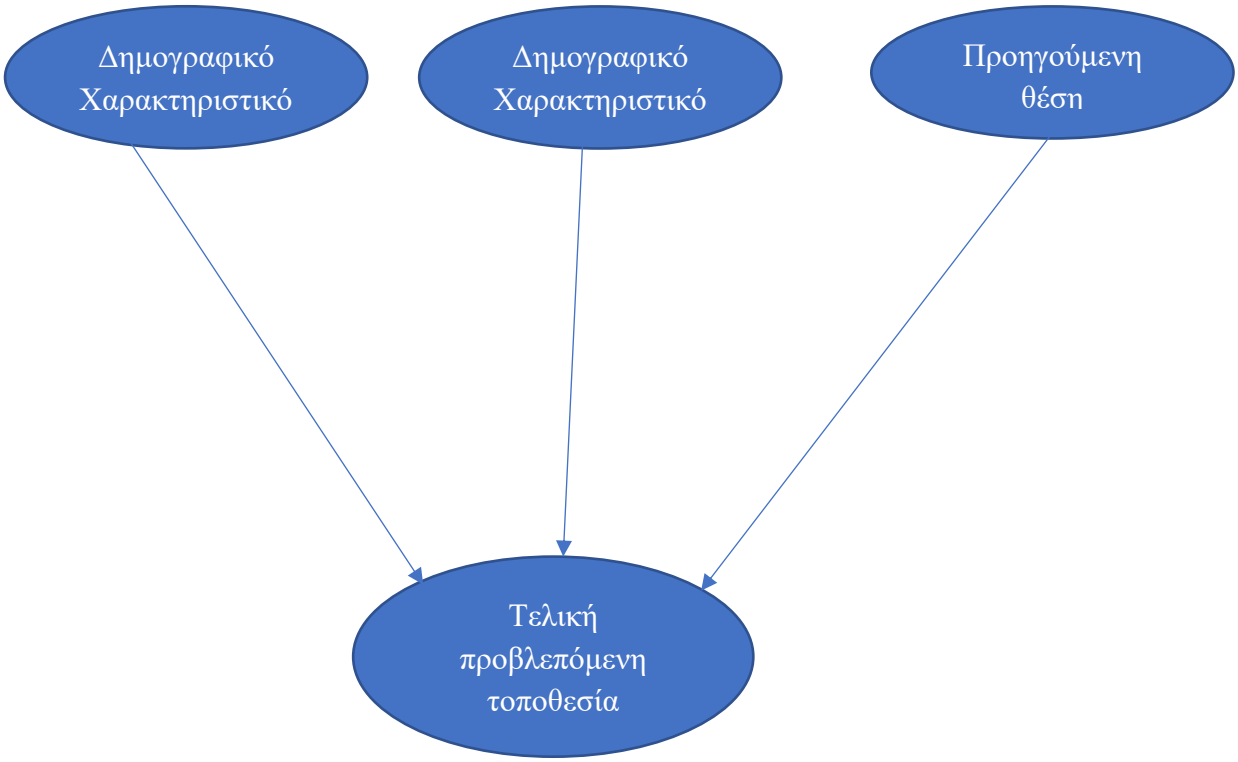
Στο προγραμματιστικό μέρος της εργασίας οι μέθοδοι που εφαρμόστηκαν στο κομμάτι του Dynamic Bayesian Network είναι η απλή πρόβλεψη (SimplePredict), η διπλή πρόβλεψη (DoublePredict), η απλή πρόβλεψη με γνώση του φύλο των πελατών (GenderPredict), η απλή πρόβλεψη γνώση της ηλικίας των πελατών, η απλή πρόβλεψη με βάση τον αριθμό παιδιών κάτω των 18 χρόνων που έχει κάθε πελάτης (ChildrenPredict) και η απλή πρόβλεψη με βάση τόσο το φύλο όσο και την ηλικία (Age_and_GenderPredict). Τα δίκτυα που σχηματίστηκαν είναι τριών μορφών και αποτελούνται από 2, 3 ή 4 μεταβλητές εκ των οποίων η μία είναι πάντα η έξοδος – τελική θέση και οι υπόλοιπες είναι η είσοδος. Το simple predict δημιουργεί το πρώτο δίκτυο, το age and gender predict δημιουργεί το δίκτυο με τις 4 μεταβλητές (ηλικία, φύλο, προηγούμενη – αρχική θέση και τελική θέση) ενώ οι υπόλοιπες κατασκευάζουν αντίστοιχο δέντρο με 3 ακμές. Παρακάτω φαίνονται αυτά τα μοντέλα με τις μεταβλητές να αποτελούν τις ακμές του γράφου.



Εικόνα 9 απλή DBS πρόβλεψη



Εικόνα 10 DBS πρόβλεψη με 2 μεταβλητές σαν είσοδο



Εικόνα 11 DBS με 3 μεταβλητές σαν είσοδο

Για να δημιουργηθούν αυτά τα δίκτυα, έγινε χρήση της βιβλιοθήκης της γλώσσας Python pgmpy και εισήχθη το μοντέλο BayesianModel. Τα στοιχεία που δέχεται σαν είσοδο του ο αλγόριθμος είναι ο πίνακας που κατασκευάστηκε κατά την επεξεργασία δεδομένων και που ανάλογα με το ποια περίπτωση εξετάζεται κάθε φορά θα έχει διαφορετικές στήλες και θα περιλαμβάνει ουσιαστικά την αλληλουχία κινήσεων των ατόμων. Στο μοντέλο την απλής πρόβλεψης που εξετάζεται πρώτα ο πίνακας που δημιουργείται αποτελείται από 2 στήλες που αντιπροσωπεύουν την αρχική και τελική θέση όπως έχει προκύψει από την μελέτη των αλληλουχιών, όλων των πελατών. Έτσι, αν ένας πελάτης έχει ακολουθήσει την πορεία καταστημάτων 5, 86, 10 τότε στον πίνακα θα προστεθούν μια γραμμή με τις τιμές 5,86 και μία με τις 86-10. Κατά την εξέταση ενός επόμενου πελάτη η αντίστοιχη αλληλουχία κινήσεων του προστίθεται στον πίνακα με την δημιουργία καινούριων γραμμών. Το βασικό πρόβλημα που προκύπτει και χρήζει επίλυσης είναι η εξέταση των διαφορετικών επισκέψεων (visits) μεταξύ του ίδιου πελάτη ώστε ο αλγόριθμος να λειτουργεί σωστά. Αυτό, όμως αντιμετωπίζεται με τον ίδιο τρόπο με τον διαφορετικό επισκέπτη αφού αν προστεθεί ξεχωριστή γραμμή για κάθε visit δεν υπάρχει σύγχυση και ο αλγόριθμος δέχεται σαν είσοδο τις σωστές αλληλουχίες. Στις υπόλοιπες περιπτώσεις ο πίνακας λειτουργεί με τον ίδιο τρόπο με την διαφορά ότι έχει όσες στήλες είναι και οι μεταβλητές του αντίστοιχου γράφου.

Παρακάτω φαίνονται οι 3 διαφορετικές περιπτώσεις πινάκων αφού όσες περιπτώσεις αφορούν 2 μεταβλητές εισόδου και 1 εξόδου χειρίζονται με τον ίδιο τρόπο είτε αφορούν δημογραφικά χαρακτηριστικά είτε πρόβλεψη με ιστορικό 2 θέσεων. Η μοναδική διαφοροποίηση στα σχήματα με τον πραγματικό πίνακα που δέχεται σαν είσοδο ο αλγόριθμος είναι η κενή γραμμή που φανερώνει την αλλαγή πελάτη ή την αλλαγή visit του πελάτη.

Πίνακας 1 απλή πρόβλεψη

Προηγούμενη Θέση	Τελική Θέση
57	123
123	8
8	2
30	20
20	150
150	111
111	99

Πίνακας 2 Ιστορία 2 θέσεων

2 Θέσεις Πίσω	Προηγούμενη Θέση	Τελική Θέση
57	123	8
123	8	2
8	2	19
1	3	120
3	120	44
120	44	158

Πίνακας 3 Πρόβλεψη με 3 μεταβλητές εισόδου

Προηγούμενη Θέση	Ηλικία	Φύλο	Τελική Θέση
57	1	1	123
123	1	1	8
8	1	1	2
1	4	2	3
3	4	2	120
120	4	2	44

Το στάδιο που ακολουθεί αμέσως μετά ονομάζεται στάδιο εκπαίδευσης ή training του αλγορίθμου. Στο στάδιο αυτό ο αλγόριθμος δέχεται σαν δεδομένα εκπαίδευσης (training data) του κάποια από τα δεδομένα που έχουν περαστεί στον πίνακα είσοδο. Στην παρούσα εργασία επιλέχτηκε ο αλγόριθμος να εκπαιδεύεται παίρνοντας σαν δεδομένα τα πρώτα 20% στοιχεία του πίνακα εισόδου.

Η τελευταία παράμετρος που δίνεται σαν είσοδος στον αλγόριθμο DBS είναι η δομή του μοντέλου δηλαδή το τελικό σχήμα του γράφου που σχηματίζεται. Αυτό συμβαίνει μέσω της δήλωσης των εξαρτήσεων μεταξύ των στηλών του πίνακα. Αν δηλαδή, στο παράδειγμα της διπλής πρόβλεψης δηλωθεί ότι A,B → C όπου A,B,C οι στήλες του πίνακα εισόδου, τότε το μοντέλο είναι αυτό του σχήματος 4.1 ενώ αν η εξάρτηση είναι η A,B,C → D τότε το μοντέλο είναι αυτό του σχήματος 4.2.

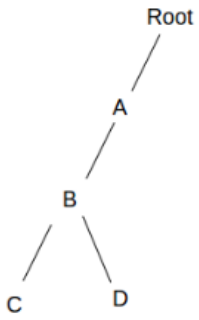
Στο τελευταίο στάδιο, αρχικά, προσδιορίζεται η μεταβλητή που θα προβλεφθεί από τον αλγόριθμο δηλαδή δηλώνεται η στήλη του πίνακα που τελικά προβλέπεται (π.χ στήλη C) και ο αλγόριθμος προβλέπει μέσω της εντολής predict μια ακέραια τιμή που αναφέρεται σε κάποιο από τα καταστήματα (0-164). Αν η προβλεπόμενη τιμή αυτή είναι ίδια με την πραγματική τιμή που βρέθηκε ο πελάτης τότε η πρόβλεψη είναι σωστή ενώ σε διαφορετική περίπτωση είναι λάθος. Τελικά το ποσοστό ακρίβειας είναι το ποσοστό των σωστά προβλεπόμενων τιμών προς τις συνολικές προβλεπόμενες τιμές. Τα αποτελέσματα του Dynamic Bayesian Network είναι (να το γράψω εδώ ή όχι?)

4.3.2 Compact Prediction Tree

Ο αλγόριθμος CPT αν και μοιάζει με τον Dynamic Bayesian Network στο κομμάτι της δημιουργίας γράφου/δέντρου κατά την υλοποίηση του μοντέλου πρόβλεψης, έχει ορισμένες διαφορές. Στην πραγματικότητα υλοποιεί και στην συνέχεια κάνει χρήση 3 βασικών δομών δεδομένων οι οποίες αναλύονται συνοπτικά παρακάτω.

- Δέντρο Πρόβλεψης : Το δέντρο ή prediction tree είναι ένα δέντρο με κόμβους όπου κάθε κόμβος έχει 3 στοιχεία, αντικείμενο, παιδιά , γονιός. Αντικείμενο ονομάζεται η τιμή που περιέχει μέσα ο κόμβος. Παιδιά ονομάζεται η λίστα με όλα τα μετέπειτα στοιχεία του γράφου κάτω από τον συγκεκριμένο κόμβο, δηλαδή οι απόγονοι του. Γονιός ονομάζεται η αναφορά στον προηγούμενο κόμβο του τρέχοντα κόμβου, δηλαδή στον πρόγονό του. Το δέντρο πρόβλεψης είναι μια δομή δεδομένων τύπου trie που στη ουσία συμπιέζει όλα τα δεδομένα εκμάθησης (training data) σε μια δομή δέντρου όπως φαίνεται στο σχήμα.

Sequence 1: A, B, C
Sequence 2: A, B, D



Εικόνα 12 δομή trie

- Ανεστραμμένος Δείκτης (Inverted Index) : ο Ανεστραμμένος Δείκτης είναι ένα ευρετήριο (dictionary) όπου το κλειδί είναι το αντικείμενο από το training set και το αποτέλεσμα στα δεξιά είναι το σετ από ακολουθίες όπου αυτό εμφανίζεται. Ένα τέτοιο παράδειγμα φαίνεται στο σχήμα :

Sequence 1: A,B,C,D
Sequence 2: B,C
Sequence 3: A,B

The Inverted Index for the above sequence will look like the below:

```
II = {
'A':{'Seq1','Seq3'},
'B':{'Seq1','Seq2','Seq3'},
'C':{'Seq1','Seq2'},
'D':{'Seq1'}
}
```

Εικόνα 13 Ανεστραμμένος Πίνακας

- Lookup Table (LT) : Ο LT είναι κι αυτός ένα ευρετήριο στο οποίο το κλειδί είναι το ID κάθε ακολουθίας και σαν αποτέλεσμα είναι ο τελικός κόμβος κάθε ακολουθίας. Ένα παράδειγμα φαίνεται παρακάτω :

Sequence 1: A, B, C
Sequence 2: A, B, D

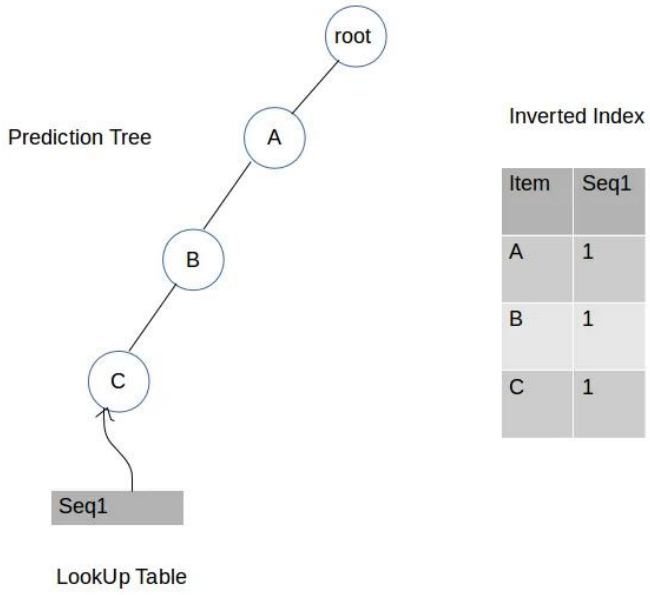
```
LT = {
"Seq1" : node(C),
"Seq2" : node(D)
}
```

Εικόνα 14 Πίνακας ελέγχου

Ο αλγόριθμος CPT όπως και ο DBS αποτελείται από 2 στάδια εκτέλεσης, το στάδιο της εκπαίδευσης και το στάδιο της πρόβλεψης. Το στάδιο εκπαίδευσης είναι το σημείο που δημιουργούνται ταυτόχρονα οι 3 δομές που περιγράφηκαν παραπάνω. Συγκεκριμένα αναλύεται ο τρόπος που λειτουργεί η φάση της εκπαίδευσης με ένα αναλυτικό παράδειγμα.

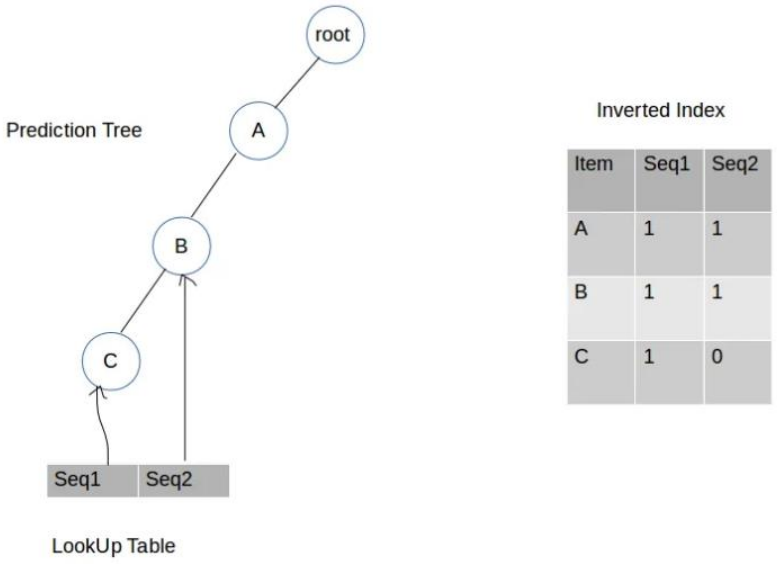
- Είσοδος του A,B,C.

Πρώτα, ξεκινάει ο αλγόριθμος με το A και ελέγχει αν υπάρχει σαν παιδί της ρίζα τους δέντρου. Αν δεν υπάρχει τότε μπαίνει στη λίστα των παιδιών της ρίζας, προστίθεται μια εισαγωγή του A στον ανεστραμμένο δείκτη με τιμή seq1 και μετά η τρέχουσα ακμή είναι η A. Στη συνέχεια ελέγχεται αν το B ανήκει στα παιδιά του A και αν όχι τότε μπαίνει στη λίστα και προστίθεται μια εισαγωγή B στον ανεστραμμένο δείκτη με τιμή seq1 και μετακίνηση του τρέχοντος κόμβου στον B. Η διαδικασία συνεχίζεται μέχρις ότου να προστεθεί το τελευταίο στοιχείο της ακολουθίας 1 δηλαδή το C στο lookup table με κλειδί seq1 και τιμή node (c) .



Εικόνα 15 είσοδος του A,B,C

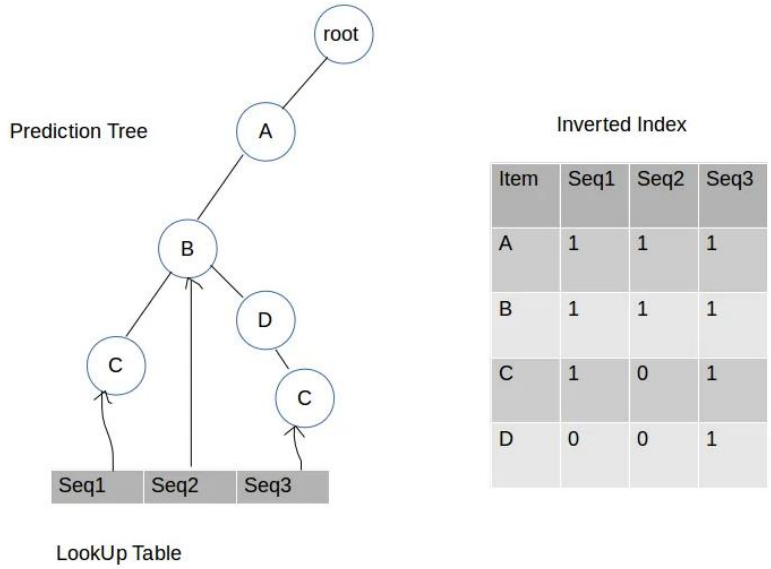
- Είσοδος του A,B.



Item	Seq1	Seq2
A	1	1
B	1	1
C	1	0

Εικόνα 16 Είσοδος του A,B

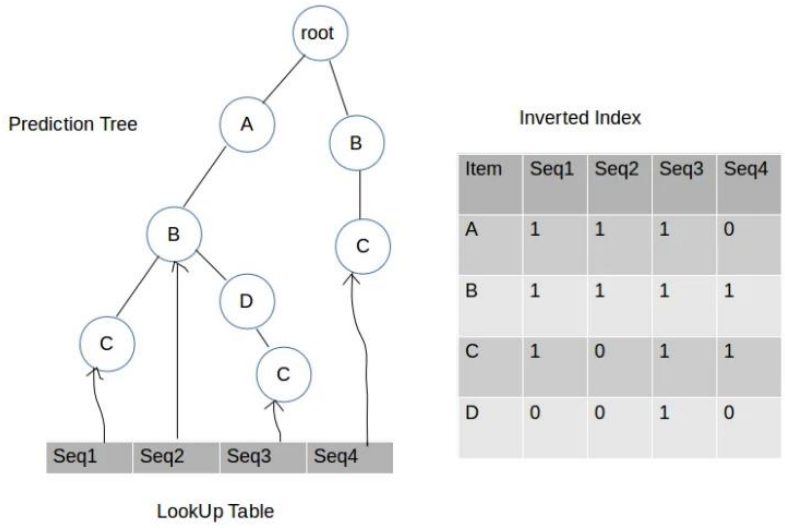
- Είσοδος του A,B,D,C.



Item	Seq1	Seq2	Seq3
A	1	1	1
B	1	1	1
C	1	0	1
D	0	0	1

Εικόνα 17 Είσοδος του A,B,C,D

- Είσοδος του B,C.



Εικόνα 18 Είσοδος του B,C

Αυτή η διαδικασία συνεχίζεται μέχρι να τελειώσουν όλες οι γραμμές του training set αφού κάθε γραμμή των δεδομένων εκπαίδευσης αντιπροσωπεύει μια ξεχωριστή αλληλουχία τοποθεσιών.

Το στάδιο πρόβλεψης περιλαμβάνει την πρόβλεψη για κάθε ακολουθία δεδομένων στο test set με την σειρά που δίνονται. Για κάποια γραμμή βρίσκονται ίδιες ακολουθίες μέσω του ανεστραμμένου γράφου και στη συνέχεια εντοπίζονται τα αμέσως επόμενα σημεία των ίδιων ακολουθιών τα οποία δημιουργούν ένα ευρετήριο που ονομάζεται πίνακας μέτρησης για χάρη ευκολίας. Τελικά η επόμενη θέση που περιέχεται περισσότερες φορές στον πίνακα μέτρησης είναι η επικρατέστερη τιμή και άρα προβλέπεται από το μοντέλο. Αυτή η διαδικασία φαίνεται καλύτερα με ένα απλό παράδειγμα στο οποίο δίνεται σαν ακολουθία η A,B στο ήδη υπάρχον παράδειγμα της φάσης εκπαίδευσης.

Παρατηρείται αρχικά ότι το A περιέχεται σε 3 ακολουθίες ενώ το B και στις 4. Ωστόσο οι όμοιες ακολουθίες που εξετάζονται είναι αυτές που περιέχουν και το A και το B αφού αυτό δόθηκε σαν ακολουθία προς εξέταση. Άρα η συναλήθευση των 2 είναι οι ακολουθίες 1,2 και 3. Έτσι αφού εξεταστούν τα επόμενα στοιχεία των ακολουθιών αυτών προκύπτει για παράδειγμα ενας πίνακας της μορφής [D,C,Y,X, D, X, D] όπου φαίνεται πως το στοιχείο D εμφανίζεται περισσότερες φορές και άρα προβλέπεται αυτό. Αν και το παράδειγμα φαίνεται απλοικό αντικατοπτρίζει ακριβώς την διαδικασία που απλώς στην περίπτωση της εργασίας χιλιάδες ακολουθίες.

Ειδικότερα στο κομμάτι της παρούσας εργασίας γίνονται οι ίδιες προβλέψεις με το αντίστοιχο Bayesian model δηλαδή η απλή πρόβλεψη (πίνακας 2 στηλών και πρόβλεψη της $2^{ης}$), διπλή πρόβλεψη (πρόβλεψη με ιστορικό 2 θέσεων), πρόβλεψη με βάση την προηγούμενη θέση και την ηλικία, φύλο, αριθμό παιδιών κάτω των 18 χρόνων (πίνακες 3 στηλών) και πρόβλεψη με βάση την προηγούμενη θέση, το φύλο ΚΑΙ την ηλικία (πίνακας 4 στηλών).

Ο αλγόριθμος Compact Prediction Tree (CPT) αποτελεί έναν πολύ λιγότερα γνωστό αλγόριθμο στο ερευνητικό πεδίο της πρόβλεψης τοποθεσίας σε εσωτερικό χώρο συγκριτικά με τους DBS, Hidden Markov Models,Directed Graphs ενώ αντίθετα τα ποσοστά ακρίβειας του σε συνδυασμό με την ταχύτητα πρόβλεψης είναι σε κάποιες περιπτώσεις καλύτερα.

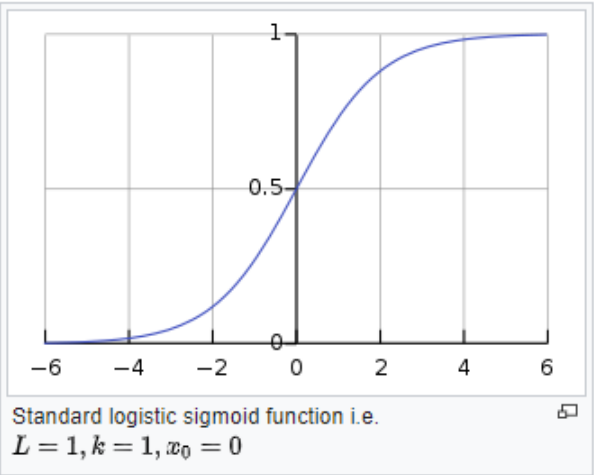
4.3.3 Logistic Regression Algorithm

Στη στατιστική ένα μοντέλο logistic χρησιμοποιείται για την μοντελοποίηση της πιθανότητας ενός γεγονότος να συμβεί ή όχι όπως για παράδειγμα αποτυχία/επιτυχία, νίκη ήττα κ.α. Αυτό σημαίνει ότι οι απαντήσεις που πρακτικά δίνει στα προβλήματα που καλείται να επιλύσει είναι ναι/όχι και άρα η λύση που παράγει είναι διττή (binary). Το logistic regression είναι ένα στατιστικό μοντέλο που στη βασική του μορφή χρησιμοποιεί την συνάρτηση logistic (ή sigmoid όπως ονομάζεται) για να μοντελοποιήσει μια διττή εξαρτημένη μεταβλητή (με 2 τιμές σαν περιεχόμενο) δοσμένων ορισμένων ανεξάρτητων μεταβλητών. Η διττή εξαρτημένη μεταβλητή, κατά την μοντελοποίηση λαμβάνει τις τιμές 0 ή 1 ενώ η απάντηση του αλγορίθμου θα είναι πάντα 2 πιθανότητες οι οποίες θα προσδιορίζουν πόσο πιθανό είναι να συμβεί το 0 και πόσο πιθανό το 1. Ο τύπος που χρησιμοποιείται στην συνάρτηση logistic και μια κατατοπιστική γραφική παράσταση φαίνονται παρακάτω :

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

Εικόνα 19 συνάρτηση logistic

- e = αριθμός Euler περίπου ίσως με 2.718
- Xo = η τετμημένη στον άξονα χ της καμπύλης
- L = μέγιστη τιμή της καμπύλης
- k = ο ρυθμός αύξησης της καμπύλης ή αλλιώς η καμπυλότητα της



Εικόνα 20 Σιγμοειδής συνάρτηση

Ωστόσο, το πρόβλημα που καλείται να αντιμετωπιστεί στην εργασία αυτή δεν έχει σαν λύση μια μεταβλητή διττής σημασίας αφού η πρόβλεψη επόμενης θέσης δεν λύνεται με ναι/όχι ή 0 και 1. Έτσι, γίνεται εισαγωγή της γενικής περίπτωσης του Logistic Regression algorithm που ονομάζεται Multinomial Regression. Το μοντέλο ML έχει την διαφορά ότι οι εξαρτημένες μεταβλητές Y που επιλύει είναι 2 ή περισσότερα διακριτά ξεχωριστά αποτελέσματα με εξάρτηση από κάποιες ανεξάρτητες μεταβλητές. Στην περίπτωση της εργασίας εξαρτημένες μεταβλητές είναι τα προβλεπόμενα μαγαζιά και άρα οι επόμενες τοποθεσίες των πελατών ενώ ανεξάρτητες είναι είτε η προηγούμενες θέσεις είτε τα δημογραφικά χαρακτηριστικά των πελατών.

Το μοντέλο στο περιβάλλον Python κατασκευάζεται μέσω της βιβλιοθήκης scikit-learn. Αρχικά δίνεται στον αλγόριθμο ένας πίνακας με στήλες όπως ακριβώς και στους προηγούμενους 2 αλγορίθμους και στη συνέχεια δίνεται η φύση κάθε μεταβλητής που αντιστοιχεί σε κάθε στήλη του πίνακα. Έτσι το μοντέλο ξέρει ότι πρέπει να προβλέψει την εξαρτημένη μεταβλητή. Ο πίνακας που δόθηκε σαν είσοδος αποτελεί το training set του συγκεκριμένου αλγορίθμου που όπως και στα άλλα μοντέλα, περιέχει το 20% των παρατηρήσεων ώστε να προβλεφθούν οι υπόλοιπες. Τέλος προβλέπεται η τιμή σε δεκαδικό αριθμό που προσεγγίζει την αντίστοιχη του καταστήματος που θα έπρεπε να προβλεφθεί.

4.4 Αποτελέσματα

Τα αποτελέσματα των αλγορίθμων που χρησιμοποιήθηκαν στην εν λόγω διπλωματική εργασία συνοψίζονται στην καταγραφή της ακρίβειας του καθενός δηλαδή του ποσοστού σωστής πρόβλεψης στο σύνολο των παρατηρήσεων. Όπως ήδη αναφέρθηκε οι αλγόριθμοι που χρησιμοποιήθηκαν και είχαν την μεγαλύτερη επιτυχία είναι ο Dynamic Bayesian Network, ο Compact Decision Tree και ο Multinomial Logistic Regression. Οι συνθήκες κάτω από τις οποίες ελέγχθηκε η απόδοση τους είναι (1) η περίπτωση 1 στην οποία κάθε πελάτης αντιστοιχίζεται χωρίς έλεγχο στο κοντινότερο κατάστημα σε αυτόν με βάση την ελάχιστη απόσταση του από όλα τα κέντρα και (2) η περίπτωση 2 κατά την οποία ο πελάτης αντιστοιχίζεται στο

κοντινότερο του μαγαζί αν η απόσταση του είναι μικρότερη από το threshold των 50 μέτρων όπως θεωρήθηκε, αλλιώς κατατάσσεται στο "μαγαζί" 165 που στην πραγματικότητα αντικατοπτρίζει τους διαδρόμους ή τις σκάλες του εμπορικού κέντρου. Κάθε αλγόριθμος παράγει 6 αποτελέσματα δηλαδή τα αποτελέσματα των διαφορετικών μοντέλων που κατασκευάζει ανάλογα με το τι θέλει να προβλέψει. Έτσι, καταγράφονται τα αποτελέσματα της απλής πρόβλεψης (simple prediction), της διπλής πρόβλεψης (double), της απλής πρόβλεψης σε συνδυασμό με την ηλικία, το φύλο, τον αριθμό παιδιών κάτω των 18 χρόνων αλλά και της απλής πρόβλεψης σε συνδυασμό τόσο με το φύλο όσο και με την ηλικία. Οι 6 αυτές προβλέψεις όπως φαίνεται και στους πίνακες που ακολουθούν είναι διαφορετικές στην περίπτωση 2 από τις αντίστοιχες τιμές της πρώτης περίπτωσης. Παρακάτω φαίνονται τα αποτελέσματα.

Πίνακας 4 Περίπτωση 1

Περίπτωση 1	**DBS**	**CPT**	**MLR**
Simple	30.11	17.154	16.942
Double	31.535	14.76	16.292
Simple + Gender	29.287	17.077	16.237
Simple + Age	29.779	16.396	16.4
Simple + Children	30.148	16.861	16.741
Simple /Age /Gender	28.417	16.108	16.703

Πίνακας 5 Περίπτωση 2

Περίπτωση 2	**DBS**	**CPT**	**MLR**
Simple	51.777	10.903	42.064
Double	53.282	8.526	44.633
Simple + Gender	50.262	10.87	43.311
Simple + Age	51.451	10.236	42.102
Simple + Children	51.408	10.659	42.428
Simple / Age / Gender	50.079	9.853	41.996

Συνοπτικά, φαίνεται πως το Dynamic Bayesian Network είναι καλύτερο στην ακρίβεια πρόβλεψης τουλάχιστον στο παράδειγμα της εργασίας αυτής, με πιο ισχυρή του πρόβλεψη αυτή της διπλής πρόβλεψης (Double) πράγμα εν μέρη λογικό αφού όσο περισσότερη γνώση έχει ένα μοντέλο τόσο περισσότερη ακρίβεια θα έχει η τελική πρόβλεψη. Μετά, το Multinomial Logistic Regression καταλαμβάνει την $2^{η}$ θέση αφού και στις 2 περιπτώσεις είναι πίσω από το DBS. Παρόλα αυτά, και στα 2 προαναφερθέντα παρατηρείται αύξηση στην περίπτωση 2 και μάλιστα κατακόρυφη, αφού το ποσοστό περίπου διπλασιάζεται. Το Compact Prediction Tree όπως φάνηκε δεν κατάφερα να αυξήσει το αρχικά μέτριο ποσοστό του και μειώθηκε αισθητά στο μισό με ποιο λογική εξήγηση αυτή που το αποδίδει σε σφάλμα κατά τη δημιουργία του μοντέλου του. Σε γενικές γραμμές βέβαια η $2^{η}$ περίπτωση είναι κατά πολύ καλύτερα δομημένη σε επίπεδο παραδοχών με συνέπεια τα αποτελέσματα της να αυξάνονται κατά πολύ συγκριτικά με την $1^{η}$ περίπτωση.

Κεφάλαιο 5. Συμπεράσματα και προοπτικές

5.1 Συμπεράσματα

Ξεκινώντας από την περίπτωση 1 φαίνεται πως όλα τα μοντέλα λειτουργούν με παρόμοιο τρόπο καθώς τα ποσοστά ακρίβειας που καταφέρνει ο καθένας είναι πολύ κοντά μεταξύ τους αν και κυμαίνονται σε χαμηλά ποσοστά. Ένας λόγος που λογικά επιδράει στο συγκεκριμένο αποτέλεσμα είναι η ύπαρξη outliers στα δεδομένα δηλαδή ορισμένων μεγάλων λανθασμένων τιμών στις τοποθεσίες των πελατών οι οποίες στην ουσία οδηγούν στην λάθος αντιστοίχιση πελατών στα καταστήματα με αποτέλεσμα να επηρεάζεται η πρόβλεψη. Επειδή τα μοντέλα λειτουργούν με παρόμοιο τρόπο στην πρόβλεψη τότε φαίνεται λογικό να κινούνται σε παρόμοιες ακρίβειες. Από την άλλη, παρατηρείται πως η ακρίβεια στην διπλή πρόβλεψη αυξάνεται αισθητά στην πρώτη περίπτωση τουλάχιστον στον Dynamic Bayesian πράγμα που τουλάχιστον σε θεωρητικό επίπεδο δικαιολογείται αφού όσο περισσότερα στοιχεία κρατάς από το παρελθόν τόσο πιο πολλά στοιχεία έχει ο αλγόριθμος για να κατασκευάσει το μοντέλο σωστά. Ωστόσο, κάτι τέτοιο δεν ισχύει στο κομμάτι της απλής πρόβλεψης με γνώση φύλου ηλικίας ή αριθμού παιδιών. Αυτό δείχνει πως η γνώση δημογραφικών στοιχείων του ατόμου δεν αλλάζει δραματικά την ακρίβεια της πρόβλεψης ή τουλάχιστον δεν την αλλάζει στον βαθμό της διπλής πρόβλεψης.

Στη δεύτερη περίπτωση παρατηρείται, τουλάχιστον από τους 2 αλγορίθμους dynamic bayesian network και multinomial logistic regression μια τεράστια αύξηση της ακρίβειας πρόβλεψης της τάξης του 100-150 % (από 20 σε 40 ή 50). Κάτι τέτοιο φαίνεται να ανταποκρίνεται στην πραγματικότητα αφού από τη στιγμή που στην αρχική θεώρηση οι πελάτες αντιστοιχίζονταν σε κάποιο μαγαζί ακόμα κι αν βρίσκονταν πολύ μακριά του. Αυτό χάλαγε το μοντέλο αφού το τροφοδοτούσε με λανθασμένα δεδομένα. Πλέον ο πελάτης αντιστοιχίζεται είτε στο κοντινό του μαγαζί είτε στους διαδρόμους του εμπορικού πράγμα που τον κατατάσσει πολύ πιο σωστά στο σημείο που βρίσκεται κανονικά . Επιπλέον η διπλή πρόβλεψη εξακολουθεί να είναι καλύτερη από την απλή ακόμα κι αν αυτή εμπλουτίζεται με δημογραφικά χαρακτηριστικά. Το βασικό μειονέκτημα της 2ης περίπτωσης είναι η μείωση της ακρίβειας της πρόβλεψης στον αλγόριθμο CPT κάτι που εκ πρώτης όψεως φαίνεται δύσκολο να εξηγηθεί. Η πιο λογική εξήγηση είναι πως επειδή το μοντέλο που κατασκευάζεται είναι ένα δέντρο που περιλαμβάνει τις τιμές των τοποθεσιών κάθε φορά που αυτές εμφανίζονται και άρα υπάρχει μεγάλη επικάλυψη και επανάληψη κλαδιών στο δέντρο τότε ο αλγόριθμος από μόνος του κλαδεύει ορισμένα παιδιά από το δέντρο με αποτέλεσμα να χάνεται η επιθυμητή πληροφορία. Αυτό ίσως εξηγεί το γεγονός ότι στον CPT στην 2η περίπτωση η double πρόβλεψη βγάζει ακόμα χειρότερα αποτελέσματα από την απλή, πράγμα που σε άλλη περίπτωση δεν θα δικαιολογούταν.

Όπως είναι λογικό ο απώτερος σκοπός που γίνονται δοκιμές και προβλέψεις τοποθεσίας σε εσωτερικούς χώρους είναι ώστε τελικά να χρησιμοποιηθούν σε πραγματικό περιβάλλον και να δουν χρήση στην καθημερινή ζωή. Τα παραπάνω αποτελέσματα, τουλάχιστον στην περίπτωση 2 που είναι και πιο ρεαλιστική αφού λαμβάνει υπόψη τον περιορισμό των 50 μέτρων και της ειδικής αυτοσχέδιας τοποθεσίας, είναι μια καλή προσομοίωση της πραγματικότητας και άρα είναι ελπιδοφόρα στο να εφαρμοστούν σε επίπεδο κανονικού και όχι ψηφιακού κόσμου.

Πράγματι, στο παράδειγμα των ελληνικών εμπορικών κέντρων όπως παρατηρεί κανείς εύκολα, φαίνεται πως υπάρχουν διάδρομοι στη μέση του εμπορικού με αποτέλεσμα να περιτριγυρίζονται κάπως από τα καταστήματα. Αυτό ερμηνεύεται σε αλγοριθμικό επίπεδο και σε επίπεδο προσομοίωσης μέσω της θεώρησης των 50 μέτρων αφού πράγματι για να θεωρηθείς πως είναι κανείς στα 50 μέτρα, σίγουρα βρίσκεται κάπου στην κεντρική περιοχή των διαδρόμων. Ωστόσο πάλι ελλοχεύει ο κίνδυνος των κυλιόμενων σκαλών του εμπορικού αφού για αυτές δεν υπάρχει ιδιαίτερος διαχωρισμός και με το προτεινόμενο μοντέλο θα θεωρηθούν κι αυτές σαν κομμάτι διαδρόμου. Βέβαια αυτό έχει να κάνει τελείως με την δομή του εμπορικού κέντρου οπότε αν υπήρχε στη γνώση από τον προγραμματιστή της μορφολογίας τότε θα ήταν εύκολο να υλοποιηθεί ανάλογα.

Αν λοιπόν όντως εφαρμόζονταν αυτά σε πραγματικές συνθήκες τότε οι καταστηματάρχες θα μπορούσαν ανάλογα να προσάρμοζαν τα καταστήματα τους εις τρόπον ώστε να μεγιστοποιήσουν τα κέρδη τους. Για παράδειγμα αν ένας ιδιοκτήτης καταστήματος έχει γνώση των θερμών περιοχών του εμπορικού δηλαδή των περιοχών που υπάρχει μεγαλύτερη κίνηση τότε μπορεί εύκολα να προωθήσει το μαγαζί του αλλά και τα προϊόντα του στο ευρύ κοινό αποδοτικότερα. Παράλληλα οι υπεύθυνοι του εμπορικού επωφελούνται από τις εν λόγω προγραμματιστικές τεχνικές αφού τα αποτελέσματα των προβλέψεων μπορούν να κάνουν γνωστό κάποιο συγκεκριμένο σημείο από το οποίο περνάνε ενδεχομένως πολλά άτομα κατά την πορεία τους και έτσι να επιτευχθεί η διευκόλυνση και η ασφάλεια των πελατών μέσω κατάλληλα διαμορφωμένων εξόδων κινδύνου ώστε σε περίπτωση ανάγκης η κίνηση να ισομοιράζεται σε διάφορες περιοχές του χώρου.

5.2 Προτάσεις για Μελλοντική Έρευνα

Ένα από τα πιο βασικά πράγματα που θα μπορούσαν να ερευνηθούν περεταίρω είναι η θεώρηση της 2ης περίπτωσης ότι κάθε πελάτης που έχει μεγαλύτερη απόσταση από 50 μέτρα από το κοντινότερο του μαγαζί, θα αντιστοιχίζεται στους διαδρόμους του εμπορικούς δηλαδή το νούμερο 165. Αν υπήρχε η γνώση της δομής του εμπορικού κέντρου τότε θα μπορούσαν να θεωρηθούν περισσότερες τοποθεσίες προσδιορισμού της θέσης των ανθρώπων μέσα στο χώρου του εμπορικού, όπως για παράδειγμα μια συγκεκριμένη θέση “διαδρόμου” για κάθε όροφο του κέντρου ή ίσως και ειδικές θέσεις για κάθε σκάλα του εμπορικού. Αυτό σε θεωρητικό επίπεδο θα βελτίωνε κατά πολύ την ακρίβεια αφού θα ήταν τελείως σωστές οι αρχικές υποθέσεις για την τοποθεσία του καθενός και άρα των εισόδων των μοντέλων.

Ακόμα, μια δυναμική προσθήκη θα μπορούσε να είναι η μελέτη και άλλων αλγορίθμων πάνω σε αυτό το κομμάτι της πρόβλεψης επόμενης τοποθεσίας όπως για παράδειγμα οι αλγόριθμοι deep learning οι οποίοι στην έρευνα αυτή αν και δοκιμάστηκαν, δεν έφεραν τα επιθυμητά αποτελέσματα.

Μετά, μια επιπλέον σημαντική συνεισφορά στην προσπάθεια επίλυσης του προβλήματος της πρόβλεψης επόμενης τοποθεσίας θα μπορούσε να είναι ο έλεγχος αλγορίθμων και ως προς άλλες τιμές πέραν της ακρίβειας τους. Για παράδειγμα, ως

προς το χρόνο εκτέλεσης ή την πολυπλοκότητα των μοντέλων και άρα της σπατάλης ή όχι υπολογιστικών πόρων. Ειδικότερα στο κομμάτι του χρόνου εκτέλεσης των μοντέλων, ορισμένες φορές είναι πιο χρήσιμο να λειτουργεί και να δίνει αποτέλεσμα γρήγορα ένας αλγόριθμος αφού γενικά οι περιπτώσεις machine και deep learning οδηγούν σε μεγάλη αναμονή αποτελέσματος. Συνεπώς μια πιο γρήγορη υλοποίηση θα μπορούσε να επιφέρει δυνατότητα υλοποίησης μοντέλων με πρόβλεψη με γνώση 3, 4 και περισσότερων θέσεων του παρελθόντος με αποτέλεσμα μεγαλύτερων ακριβειών.

Επιπλέον, συνετό θα ήταν σε μελλοντική έρευνα με κάποιο τρόπο να βελτιωθούν τα αποτελέσματα ακόμα και στο πλαίσιο των ίδιων αλγορίθμων εφαρμογής. Κάτι τέτοιο, αν και είναι απόλυτα λογικό πως όσο η τεχνολογία εξελίσσεται θα αυξάνεται και η ακρίβεια των υπολογισμών, θα μπορούσε να επιτευχθεί με την εξομάλυνση των αρχικών παρατηρήσεων. Για παράδειγμα , όπως αναφέρθηκε, στα δεδομένα της εργασίας, σε κάποια σημεία παρατηρήθηκε αδυναμία εντοπισμού των ατόμων καθώς ορισμένες τιμές (outliers) ξέφευγαν κατά πολύ από τα όρια του εμπορικού. Αυτό συμβαίνει λογικά στις λεγόμενες κρυφές περιοχές όπου το σήμα χάνεται και κατά συνέπεια ο εντοπισμός καθίσταται δύσκολος. Κάποια τέτοια δεδομένα λοιπόν σε μια άλλη ενδεχόμενη προσέγγιση θα μπορούσαν να αφαιρεθούν τελείως από το δείγμα παρατηρήσεων με σκοπό ίσως τον καθαρισμό των δεδομένων από τυχόν θορύβους. Κάτι τέτοιο ίσως έδινε καλύτερα αποτελέσματα στο κομμάτι της μοντελοποίησης και εν συνεχεία στην ακρίβεια της πρόβλεψης.

Τέλος, επειδή όπως είναι γνωστό την σύγχρονη εποχή αναπτύσσονται ευρέως τα νευρωνικά δίκτυα και γενικά ο επιστημονικός τομέας του deep learning και επειδή τα αποτελέσματα που έχει αποφέρει στον επιστημονικό κλάδο της πρόβλεψης είναι σημαντικά, ίσως θα ήταν συνετό να εξερευνηθεί λίγο περισσότερο και στον τομέα της κίνησης εσωτερικού χώρου. Κάτι τέτοιο είναι πολύ πιθανό να δώσει αποτελέσματα πολύ πιο ακριβή αφού η κατασκευή των μοντέλων θα είναι αρτιότερη.

Βιβλιογραφία

[1]Patterson, Donald & Liao, Lin & Fox, Dieter & Kautz, Henry. (2003). Inferring High-Level Behavior from Low-Level Sensors. Lecture Notes in Computer Science. 2864. 10.1007/978-3-540-39653-6_6.

[2]Mozer, Michael & Dodier, Robert & Anderson, Marc & Vidmar, Lucky & Iii, Robert & Miller, Debra. (1997). The Neural Network House: An Overview.

[3]Mozer, Michael. (2019). The Neural Network House: An Environment that Adapts to its Inhabitants.

[4]Lee, Sunyoung & Lee, Kun & Cho, Heeryon. (2010). A Dynamic Bayesian Network Approach to Location Prediction in Ubiquitous Computing Environments. [5]Communications in Computer and Information Science. 114. 73-82. 10.1007/978-3-642-16699-0_9.

[6]Byckling, Mikko. (2019). PPM: Prediction by Partial Match.

[7]Gueniche, Ted & Fournier Viger, Philippe & Tseng, Vincent. (2013). Compact Prediction Tree: A Lossless Model for Accurate Sequence Prediction. 8347. 10.1007/978-3-642-53917-6_16.

[8]Syed, Shahan & Mubeen, Muhammad & Hussain, Adnan & Lal, Irfan. (2018). Prediction of stock performance by using logistic regression model: evidence from Pakistan Stock Exchange (PSX). Asian Journal of Empirical Research. 8. 10.18488/journal.1007/2018.8.7/1007.7.247.258.

[9]Petzold, Jan. (2005). Prediction of indoor movements using Bayesian networks.

[10]Petzold, Jan & Bagci, Dr. Faruk & Trumler, Wolfgang & Ungerer, Theo. (2006). Comparison of Different Methods for Next Location Prediction. 909-918. 10.1007/11823285_96.

[11]Kaur, Manpreet & Salim, Flora & Ren, Yongli & Chan, Jeffrey & Tomko, Martin & Sanderson, Mark. (2018). Shopping Intent Recognition and Location Prediction from Cyber-Physical Activities via Wi-Fi Logs. 10.1145/3276774.3276786.

[12]Lam, Luan & Tang, Antony & Grundy, John. (2017). Predicting indoor spatial movement using data mining and movement patterns. 223-230. 10.1109/BIGCOMP.2017.7881703.

[13]Radaelli, Laura & Sabonis, Dovydas & Lu, Hua & Jensen, Christian. (2013). Identifying Typical Movements Among Indoor Objects-Concepts and Empirical Study.

Proceedings - IEEE International Conference on Mobile Data Management. 1. 197-206. 10.1109/MDM.2013.29.

[14]Nandakumar, R., Rallapalli, S., Chintalapudi, K., Padmanabhan, V.N., Qiu, L., Ganesan, A., Guha, S., Aggarwal, D., & Goenka, A. (2013). Physical Analytics: A New Frontier for (Indoor) Location Research.

[15]Wei, Wutao & Zhang, Le & Ding, Qi & Zhou, Bingrou. (2017). Dynamic Bayesian predictive model for box office forecasting. 3958-3964. 10.1109/BigData.2017.8258405.

[16]McAlinn, Kenichiro & West, Mike. (2016). Dynamic Bayesian Predictive Synthesis in Time Series Forecasting. Journal of Econometrics. 10.1016/j.jeconom.2018.11.010.

[17]Petzold, Jan & Pietzowski, Andreas & Bagci, Dr. Faruk & Trumler, Wolfgang & Ungerer, Theo. (2005). Prediction of Indoor Movements Using Bayesian Networks. 211-222. 10.1007/11426646_20.

[18]Gueniche, Ted & Fournier Viger, Philippe & Raman, Rajeev & Tseng, Vincent. (2015). CPT+: Decreasing the Time/Space Complexity of the Compact Prediction Tree. 625-636. 10.1007/978-3-319-18032-8_49.

[19]Lee, Kyewon & Ahn, Hongshik & Moon, Hojin & Kodell, Ralph & Chen, James. (2013). Multinomial Logistic Regression Ensembles. Journal of biopharmaceutical statistics. 23. 681-94. 10.1080/10543406.2012.756500.

[20]Nizetic Kosovic, Ivana. (2015). The challenges of indoor movement analysis.

[21]Zhang, Ping & Jiang, Qianqian & Zhang, Ruilin & Li, Bo. (2019). Research and Implementation of Key Techniques for Indoor Movement Object Trajectory Prediction. Journal of Physics: Conference Series. 1215. 012020. 10.1088/1742-6596/1215/1/012020.

[22]Hou, jung-fu & Chou, Yu-Shin & Chang, Yau-Zen & Liu, Jing-Sin. (2010). Experimental Investigation of a Prediction Algorithm for an Indoor SLAM Platform. 6425. 154-165. 10.1007/978-3-642-16587-0_15.

Παράρτημα

Στο παράρτημα αυτό παρατίθενται κάποια βασικά κομμάτια κώδικα για την κατανόηση όπως για παράδειγμα κάποιες από τις βασικότερες συναρτήσεις.

```
def Distance(x1,y1,x2,y2):
"""
Calculate the great circle distance between two points
on the earth (specified in decimal degrees)
"""
# convert decimal degrees to radians
lon1, lat1, lon2, lat2 = map(math.radians, [y1, x1, y2, x2])

# haversine formula
dlon = lon2 - lon1
dlat = lat2 - lat1
a = math.sin(dlat / 2) ** 2 + math.cos(lat1) * math.cos(lat2) * (math.sin(dlon / 2) ** 2)
c = 2 * math.asin(math.sqrt(a))
r = 6371  # Radius of earth in kilometers. Use 3956 for miles
return c * r * 1000


Calculate the nearest store for the given x y in a specific floor
We assume a range from the stores and x,y belong to the shortest range
'''
def GetNearestStore(floor, x,y, storesPerFloor):
    if floor == "Concourse":
        floor = 0
    else:
        floor = int(floor.split()[1])

    stores = storesPerFloor[floor]

    minDistance = 1000000000
    min = None
    for store in stores:
        distance = Distance(x, y, store["lat"], store["lng"])
        if distance < minDistance:
            min = store["id"]
            minDistance = distance

                if minDistance > 50:
                    return 160
```

```
            return min


Create dictionary with all the stores per floor
'''
def CreateStoresMapping(stores):
    floorStoresMapping = {}
    storesMapping = {}
    id = 0
    for num, row in stores.iterrows():
        latitude = row["latitude"]
        longitude = row["longitude"]
        floor = row["Floor_Index"]

        if floor not in floorStoresMapping:
            floorStoresMapping[floor] = []

        floorStoresMapping[floor].append({"lng":  longitude, "lat": latitude, "id": id})
        storesMapping[id] = {"name": row["Store_Name"], "lng":  longitude, "lat": latitude, "floor": floor}

        id += 1
    return storesMapping, floorStoresMapping


Get pings of shoppers from file or create them
'''
def CreateOrGetPings(floorStoresMapping, pings):
    pingsMapping = {}
    if not os.path.isfile("ImportFilesAndModels/pings.out"):
        start = datetime.now()
        for num, row in pings.iterrows():
            nearestStore = GetNearestStore(row["floor"], row["lat"], row["lng"], floorStoresMapping)

            if row["Shopper_ID"] not in pingsMapping:
                pingsMapping[row["Shopper_ID"]] = []

            pingsMapping[row["Shopper_ID"]].append((nearestStore, row["seenTimestamp"]))

        with open("ImportFilesAndModels/pings.out", "wb") as outFile:
            pickle.dump(pingsMapping, outFile)
```

```
    else:
        with open("ImportFilesAndModels/pings.out", "rb") as inFile:
            pingsMapping = pickle.load(inFile)

    return pingsMapping


Get visits of shopper based on the pings and the timestamp of them.
A new visit is measured if 6 hours have passed from the previous ping
'''
def CreateOrGetVisits(pingsMapping):
    if not os.path.isfile("ImportFilesAndModels/visits.out"):
        visitMapping = []
        for shopperId in pingsMapping:
            visitId = 0
            visits = sorted(pingsMapping[shopperId], key=lambda v: v[1])

            currentTime = datetime.strptime(visits[0][1], "%Y-%m-
%dT%H:%M:%S.%fZ")
            for visit in visits:
                tempTime = datetime.strptime(visit[1], "%Y-%m-%dT%H:%M:%S.%fZ")
                if ((tempTime - currentTime).total_seconds() > 60 * 60 * 6):
                    visitId += 1

                visitMapping.append((visitId, shopperId, visit[0], tempTime))

                currentTime = tempTime

        with open("ImportFilesAndModels/visits.out", "wb") as outFile:
            pickle.dump(visitMapping, outFile)
    else:
        with open("ImportFilesAndModels/visits.out", "rb") as inFile:
            visitMapping = pickle.load(inFile)

    return visitMapping



Create demographices for each shopper
'''
def CreateShopperDemo(demographics):
    shopperDemo = {}
    for num, row in demographics.iterrows():
```

```
        shopperDemo[row["Shopper_Id"]] = {"Age": row["Age"], "Gender": row["Gender"], "Children": row["Number_of_Children_under_18_years_of_age"]}
    return shopperDemo



Create simple predict model or get it from file
'''
def CreateOrGetSimpleResult(values):
    if not os.path.isfile("Bayesian/Simple.out"):
        train_data = values[int(len(values) * 0.2):]
        predict_data = values[:int(len(values) * 0.2)]

        start_time = datetime.now()
        model = BayesianModel([("A", "B")])
        model.fit(train_data)
        print("Training", (datetime.now()-start_time).total_seconds())

        start_time = datetime.now()
        predict_data_new = predict_data.copy()
        predict_data_new.drop("B", axis=1, inplace=True)

        y_pred = model.predict(predict_data_new)
        predict_data = predict_data.join(y_pred, rsuffix="_pred")
        print("Predict", (datetime.now()-start_time).total_seconds())

        with open("Bayesian/Simple.out", "wb") as outFile:
            pickle.dump(predict_data, outFile)
    else:
        with open("Bayesian/Simple.out", "rb") as inFile:
            predict_data = pickle.load(inFile)

    return predict_data



Run simple predict model
'''
def RunSimplePredict(storesMapping, visitMapping):
    predictMoving = []
    keys = list(storesMapping.keys())

    # Create a list with the previous store of the shopper and the next store
    for i in range(1,len(visitMapping)):
```

```
        fromStore = visitMapping[i-1][2]
        toStore = visitMapping[i][2]

        if fromStore in keys:
            keys.remove(fromStore)

        if visitMapping[i][1] != visitMapping[i-1][1]:
            continue

        if visitMapping[i][0] != visitMapping[i-1][0]:
            continue

        predictMoving.append([fromStore,toStore])

    for key in keys:
        predictMoving.append([key,np.nan])

    values = pd.DataFrame(predictMoving, columns=["A","B"])

    predict_data = CreateOrGetSimpleResult(values)

    print("Simple", round((len(predict_data[predict_data["B"] == predict_data["B_pred"]])/len(predict_data))*100, 3),'%')


Create or get double predict model
'''
def CreateOrGetDoubleResult(values):
    if not os.path.isfile("Bayesian/Double.out"):
        train_data = values[int(len(values) * 0.2):]
        predict_data = values[:int(len(values) * 0.2)]

        start_time = datetime.now()
        model = BayesianModel([("A", "C"),("B","C")])
        maxv = list(range(0,166))
        maxv.append(np.nan)

        model.fit(train_data, state_names={"A": maxv , "B": maxv, "C": maxv})
        print("Training", (datetime.now()-start_time).total_seconds())

        start_time = datetime.now()
        predict_data_new = predict_data.copy()
        predict_data_new.drop("C", axis=1, inplace=True)
        predict_data_new["B"] = predict_data_new["B"].apply(int)
```

```
        y_pred = model.predict(predict_data_new)
        predict_data = predict_data.join(y_pred, rsuffix="_pred")
        print("Predict", (datetime.now()-start_time).total_seconds())

        with open("Bayesian/Double.out", "wb") as outFile:
            pickle.dump(predict_data, outFile)
    else:
        with open("Bayesian/Double.out", "rb") as inFile:
            predict_data = pickle.load(inFile)

    return predict_data



Run double predict model
'''
def RunDoublePredict(storesMapping, visitMapping):
    predictMoving = []
    keys = list(storesMapping.keys())

    # Create a list with the two previous positions of the shopper and the next store
    for i in range(2, len(visitMapping)):
        firstStore = visitMapping[i - 2][2]
        secondStore = visitMapping[i - 1][2]
        thirdStore = visitMapping[i][2]

        if firstStore in keys:
            keys.remove(firstStore)

        if visitMapping[i - 2][1] != visitMapping[i][1]:
            continue

        if visitMapping[i-2][0] != visitMapping[i][0]:
            continue

        predictMoving.append([firstStore, secondStore, thirdStore])

    for key in keys:
        predictMoving.append([key, np.nan, np.nan])

    values = pd.DataFrame(predictMoving, columns=["A", "B", "C"])

    predict_data = CreateOrGetDoubleResult(values)
```

```
    print("Double", round((len(predict_data[predict_data["C"] == predict_data["C_pred"]]) / len(predict_data)) * 100, 3), '%')



Create or get from a file the gender model
'''
def CreateOrGetGenderResults(values):
    if not os.path.isfile("Bayesian/Gender.out"):
        train_data = values[int(len(values) * 0.2):]
        predict_data = values[:int(len(values) * 0.2)]

        start_time = datetime.now()
        model = BayesianModel([("A", "C"), ("B", "C")])

        model.fit(train_data)
        print("Training", (datetime.now() - start_time).total_seconds())

        start_time = datetime.now()
        predict_data_new = predict_data.copy()
        predict_data_new.drop("C", axis=1, inplace=True)
        predict_data_new["B"] = predict_data_new["B"].apply(int)

        y_pred = model.predict(predict_data_new)
        predict_data = predict_data.join(y_pred, rsuffix="_pred")
        print("Predict", (datetime.now() - start_time).total_seconds())

        with open("Bayesian/Gender.out", "wb") as outFile:
            pickle.dump(predict_data, outFile)
    else:
        with open("Bayesian/Gender.out", "rb") as inFile:
            predict_data = pickle.load(inFile)

    return predict_data



Run the gender model
'''
def RunWithGenderPredict(storesMapping, visitMapping, shopperDemo):
    predictMoving = []
```

```
    keys = list(storesMapping.keys())

    # Create a list with the previous location, the gender of the shopper and the next store
    # 1 if it is female or 0 if it is male
    for i in range(1, len(visitMapping)):
        fromStore = visitMapping[i - 1][2]
        toStore = visitMapping[i][2]

        if fromStore in keys:
            keys.remove(fromStore)

        if visitMapping[i][1] != visitMapping[i - 1][1]:
            continue

        predictMoving.append([fromStore,  shopperDemo[visitMapping[i][1]]["Gender"]
== "F", toStore])

    for key in keys:
        predictMoving.append([key, 0, np.nan])

    values = pd.DataFrame(predictMoving, columns=["A", "B", "C"])
    values['B'] = values['B'].apply(int)

    predict_data = CreateOrGetGenderResults(values)

    print("Gender",                round((len(predict_data[predict_data["C"]                ==
predict_data["C_pred"]]) / len(predict_data)) * 100, 3), '%')




Create or get the age model
'''
def CreateOrGetAgeResults(values):
    if not os.path.isfile("Bayesian/Age.out"):
        train_data = values[int(len(values) * 0.2):]
        predict_data = values[:int(len(values) * 0.2)]

        start_time = datetime.now()
        model = BayesianModel([("A", "C"), ("B", "C")])

        model.fit(train_data)
        print("Training", (datetime.now() - start_time).total_seconds())
```

```
        start_time = datetime.now()
        predict_data_new = predict_data.copy()
        predict_data_new.drop("C", axis=1, inplace=True)

        y_pred = model.predict(predict_data_new)
        predict_data = predict_data.join(y_pred, rsuffix="_pred")
        print("Predict", (datetime.now() - start_time).total_seconds())

        with open("Bayesian/Age.out", "wb") as outFile:
            pickle.dump(predict_data, outFile)
    else:
        with open("Bayesian/Age.out", "rb") as inFile:
            predict_data = pickle.load(inFile)

    return predict_data




Run with age model
'''
def RunWithAgePredict(storesMapping, visitMapping, shopperDemo):
    predictMoving = []
    keys = list(storesMapping.keys())

    # Create a list with the previous store, the age group of the shopper and the next store
    for i in range(1, len(visitMapping)):
        fromStore = visitMapping[i - 1][2]
        toStore = visitMapping[i][2]

        if fromStore in keys:
            keys.remove(fromStore)

        if shopperDemo[visitMapping[i][1]]["Age"] < 26:
            ageGroup = 1
        elif shopperDemo[visitMapping[i][1]]["Age"] < 35:
            ageGroup = 2
        elif shopperDemo[visitMapping[i][1]]["Age"] < 42:
            ageGroup = 3
        elif shopperDemo[visitMapping[i][1]]["Age"] < 49:
            ageGroup = 4
        elif shopperDemo[visitMapping[i][1]]["Age"] < 56:
            ageGroup = 5
        elif shopperDemo[visitMapping[i][1]]["Age"] < 64:
```

```
            ageGroup = 6
        else:
            ageGroup = 7

        if visitMapping[i][1] != visitMapping[i - 1][1]:
            continue

        if visitMapping[i][0] != visitMapping[i-1][0]:
            continue

        predictMoving.append([fromStore, ageGroup, toStore])

    for key in keys:
        predictMoving.append([key, 0, np.nan])

    values = pd.DataFrame(predictMoving, columns=["A", "B", "C"])
    values['B'] = values['B'].apply(int)

    predict_data = CreateOrGetAgeResults(values)

    # print(predict_data)
    print("Age", round((len(predict_data[predict_data["C"] == predict_data["C_pred"]])
/ len(predict_data)) * 100, 3), '%')

Create or get the age and gender model
'''
def CreateOrGetAgeGenderResults(values):
    if not os.path.isfile("Bayesian/AgeGender.out"):
        train_data = values[int(len(values) * 0.2):]
        predict_data = values[:int(len(values) * 0.2)]

        start_time = datetime.now()
        model = BayesianModel([("A", "D"), ("B", "D"), ("C", "D")])

        model.fit(train_data)
        print("Training", (datetime.now() - start_time).total_seconds())

        start_time = datetime.now()
        predict_data_new = predict_data.copy()
        predict_data_new.drop("D", axis=1, inplace=True)

        y_pred = model.predict(predict_data_new)
        predict_data = predict_data.join(y_pred, rsuffix="_pred")
        print("Predict", (datetime.now() - start_time).total_seconds())
```

```
        with open("Bayesian/AgeGender.out", "wb") as outFile:
            pickle.dump(predict_data, outFile)
    else:
        with open("Bayesian/AgeGender.out", "rb") as inFile:
            predict_data = pickle.load(inFile)

    return predict_data



Run the age and gender model
'''
def RunWithAgeAndGenderPredict(storesMapping, visitMapping, shopperDemo):
    predictMoving = []
    keys = list(storesMapping.keys())

    # Create a list with the previous store, the age group, the gender and the next store
    # 1 if it is female and 0 if it is male
    for i in range(1, len(visitMapping)):
        fromStore = visitMapping[i - 1][2]
        toStore = visitMapping[i][2]

        if fromStore in keys:
            keys.remove(fromStore)

        if shopperDemo[visitMapping[i][1]]["Age"] < 26:
            ageGroup = 1
        elif shopperDemo[visitMapping[i][1]]["Age"] < 35:
            ageGroup = 2
        elif shopperDemo[visitMapping[i][1]]["Age"] < 42:
            ageGroup = 3
        elif shopperDemo[visitMapping[i][1]]["Age"] < 49:
            ageGroup = 4
        elif shopperDemo[visitMapping[i][1]]["Age"] < 56:
            ageGroup = 5
        elif shopperDemo[visitMapping[i][1]]["Age"] < 64:
            ageGroup = 6
        else:
            ageGroup = 7

        if visitMapping[i][1] != visitMapping[i - 1][1]:
            continue
```

```
        if visitMapping[i][0] != visitMapping[i-1][0]:
            continue

        predictMoving.append([fromStore,                                    ageGroup,
shopperDemo[visitMapping[i][1]]["Gender"] == "F", toStore])

    for key in keys:
        predictMoving.append([key, 0, 0, np.nan])

    values = pd.DataFrame(predictMoving, columns=["A", "B", "C", "D"])
    values['B'] = values['B'].apply(int)
    values['C'] = values['C'].apply(int)

    predict_data = CreateOrGetAgeGenderResults(values)

    # print(predict_data)
    print("Age and Gender", round((len(predict_data[predict_data["D"] ==
predict_data["D_pred"]]) / len(predict_data)) * 100, 3), '%')



Create or get children model
'''
def CreateOrGetChildrenResults(values):
    if not os.path.isfile("Bayesian/Children.out"):
        train_data = values[int(len(values) * 0.2):]
        predict_data = values[:int(len(values) * 0.2)]

        start_time = datetime.now()
        model = BayesianModel([("A", "C"), ("B", "C")])

        model.fit(train_data)
        print("Training", (datetime.now() - start_time).total_seconds())

        start_time = datetime.now()
        predict_data_new = predict_data.copy()
        predict_data_new.drop("C", axis=1, inplace=True)

        y_pred = model.predict(predict_data_new)
        predict_data = predict_data.join(y_pred, rsuffix="_pred")
        print("Predict", (datetime.now() - start_time).total_seconds())
```

```
        with open("Bayesian/Children.out", "wb") as outFile:
            pickle.dump(predict_data, outFile)
    else:
        with open("Bayesian/Children.out", "rb") as inFile:
            predict_data = pickle.load(inFile)

    return predict_data



Run the children model
'''
def RunWithChildrenPredict(storesMapping, visitMapping, shopperDemo):
    predictMoving = []
    keys = list(storesMapping.keys())

    # Create a list with the previous store, the number of children and the next store
    for i in range(1, len(visitMapping)):
        fromStore = visitMapping[i - 1][2]
        toStore = visitMapping[i][2]

        if fromStore in keys:
            keys.remove(fromStore)

        if visitMapping[i][1] != visitMapping[i - 1][1]:
            continue

        if visitMapping[i][0] != visitMapping[i-1][0]:
            continue

        predictMoving.append([fromStore,
shopperDemo[visitMapping[i][1]]["Children"], toStore])

    for key in keys:
        predictMoving.append([key, 0, np.nan])

    values = pd.DataFrame(predictMoving, columns=["A", "B", "C"])
    values['B'] = values['B'].apply(int)

    predict_data = CreateOrGetChildrenResults(values)

    # print(predict_data)
    print("Children",          round((len(predict_data[predict_data["C"]          ==
predict_data["C_pred"]]) / len(predict_data)) * 100, 3), '%')
```

```
Run Bayesian Models
"""
def RunBayesian(storesMapping, visitMapping, shopperDemo):
    print("~~~~~~~~~~ Bayesian ~~~~~~~~~~")

    # Model for prediction with only the previous location
    RunSimplePredict(storesMapping, visitMapping)

    # Model for prediction with the two previous locations
    RunDoublePredict(storesMapping, visitMapping)

    # Model for prediction with the previous location and the gender
    RunWithGenderPredict(storesMapping, visitMapping, shopperDemo)

    # Model for prediction with the previous location and the age
    RunWithAgePredict(storesMapping, visitMapping, shopperDemo)

    # Model for prediction with the previous location, the gender and the age
    RunWithAgeAndGenderPredict(storesMapping, visitMapping, shopperDemo)

    # Model for prediction with the previous location and the number of children
    RunWithChildrenPredict(storesMapping, visitMapping, shopperDemo)
```

Σημειώνεται πως οι συναρτήσεις αυτές που αφορούν το Bayesian είναι παρόμοιες και στα άλλα 2 μοντέλα και συνεπώς συμπεριλήφθηκαν 1 φορά.