Spark-based loyalty reward system for an energy company

National Technical University of Athens
School of Electrical and Computer Engineering
M.Sc.: Data Science and Machine Learning
Author: Dimitris Karpontinis

July 23, 2024

MSc Thesis

Spark-based loyalty reward system for energy company

Dimitris Karpontinis

AM: 03400135

email: dimitrioskarpodinis@ntua.gr

Supervisor: Prof. Stefanos Kollias

Thesis Examination Committee

Dr. Athanasios Voulodimos
Assistant Professor, School of Electrical and Computer Engineering (NTUA)

Dr. Giorgos Stamou
Professor, School of Electrical and Computer Engineering (NTUA)

Dr. Stefanos Kollias
Professor Emeritus, School of Electrical and Computer Engineering (NTUA)

July 23, 2024

Copyright

This thesis is an original work, copyright and intellectual property rights have been respected throughout the research and development process. I affirm that I do not infringe upon any copyrights or intellectual property rights of others.

Περίληψη: Η παροχή υπηρεσιών ανταμοιβής σε πιστούς πελατές αποτελεί μια δοκιμασμένη στρατηγική που εταιρείες αξιοποιούν με στόχο την ανάπτυξη και διατήρηση του πελατολογίου της. Στην εργασία αυτή το σύστημα που υλοποιήθηκε έχει ως στόχο την παροχή πιστοποιημένης πράσινης ενέργειας σε επιλεγμένους πελάτες με βάση τις ανάγκες τους, την εμπιστοσύνη τους στην εταιρεία και την οικολογική συμπεριφορά τους.

Το σύστημα αυτό υλοποιήθηκε χρησιμοποιώντας δομές μεγάλων δεδομένων και υπολογιστική νέφους. Τα δεδομένα που χρησιμοποιήθηκαν βασίζονται στην κατανάλωση πελατών, την διαθέσιμη παραγωγή καθώς και περαιτέρω χαρακτηριστικά των πελάτων που προέκυψαν από τεχνικές εξόρυξης δεδομένων. Τέλος μέρη του αλγορίθμου αναπτύχθηκαν και τοπικά με στόχο την εκτέλεση τους σε συσκεύες άκρων (edge devices).

Λέξεις Κλειδία: Εξόρυξη Δεδομένων, Μεγάλα Δεδομένα, Υπολογιστική Νέφους, Σύστημα Επιβράβευσης Πελατών.

Abstract: Providing rewards to loyal customers is a common technique that companies employ in order to maintain and boost their clientele. In this work a system for providing certified green energy to a selection of clients has been implemented. This selection process relies on the client's needs, their loyalty as well as their attitude towards ecology, with an emphasis on transparency and adaptability to include as many clients as possible.

This system was implemented using cloud computing services and big data frameworks. The data utilized are based on the customer consumption, the available produced energy and other client features with were identified using data extraction techniques. Finally, components of the algorithm were also implemented locally, in order to be later deployed on edge devices.

Keywords: Data Mining, Big Data, Cloud Computing, Loyalty Reward System

Contents

1 Introduction

This dissertation thesis aims to present an implementation blueprint of consumption matching algorithms. The primary task of this project, is to create a transaction table in which a production site's green energy will be matched to a portion of the energy consumed in a client site, for a given date and time. This process will take into account a plethora of factors with the most important being the customer's ecological mindset and loyalty.

Below we provide a theoretical example of such a table:

Year	Month	Production Site	Consumption Site	Energy Provided
2023	3	Production Site 1	Leof. Katechaki, Zografou 157 72	2 MWh
2023	2	Production Site 2	Stournari & 28is Oktovriou, Athina 106 82	0.2 MWh

Table 1: Consumption/Production Matching Table Example

At this point it must be clarified that this matching procedure is virtual. In other words the energy produced at production site 1 will not actually be transferred to the corresponding consumption site. Such an endeavour would require to install cables between these sites for each different combination of production/consumption sites, which is not feasible.

Alternatively the matching procedure will be accomplished using Guarantees of Origin (GOs). A guarantee of origin is a document verifying the amount of energy produced, the way it was produced such as solar, wind or nuclear as well as the date, time and place it was produced. This energy will then be released on the grid where it could no longer be distinguished from any non renewable forms of energy. Therefore the only way to track the produced energy is though the Guarantee of Origin document.

The transactions included in the table must follow the A.C.I.D. properties. These properties are briefly explained in the following table:

Property Name	Property Description
Atomicity	Either all changes to the data are performed or none is.
Consistency	Data is in a consistent state when the transaction starts and when it ends.
Isolation	Each transaction can only see the end result of all other transactions.
Durability	After a transaction completes, changes to data persist, even in a system failure.

Table 2: A.C.I.D. principles

Before executing any transactions, their ability to satisfy these properties will be tested. Testing will be used considerably in this study, mainly to check

data quality and formatting standards.
The tools used for this analysis will be presented in section Required Tools. The tables utilized during the implementation will be thoroughly analysed in section Table Documentation The cleaning tasks that were implemented, such as methods used to format and filter out data, will be presented in section Data Cleaning Process. The curation tasks required, before the actual energy matching can initiate, will be presented in section Data Curation Process. Finally the energy matching algorithm as well as some concluding remarks are presented in sections Energy Matching and Conclusion respectively.

2 Required Tools

Firstly the code base will be written entirely in python. This choice was made based on the need to use clustering methodologies to categorise the company's customers while also being able to manipulate large volumes of data.

As already stated the matching algorithm will be used to create a transaction table. This table will eventually include energy transactions for every minute from January 2022 until January 2023. In total the table will include over 10^5 data points creating the need for a distributed computing framework to be utilized. For this reason the python library pyspark is selected. Pyspark is an API for the spark analytics engine. This engine was developed using scala and is designed for big data workloads, making it ideal for this project. Specifically all ETL related steps discussed in sections 4, 5 will be completed using pyspark.

After preprocessing is completed the remaining data will be used to cluster customer behavior based on the remaining features. In this process it is essential to understand what features will be used, how will they be utilised and what features should be dropped. For feature selection relevant publications must be studied. For feature utilization and combination dimensionality reduction tools such as Linear Discrimination Analysis, Independent Component Analysis and Principal Component Analysis will be useful. This ranking procedure is essential in order to actually select between different customers with similar energy needs during the matching procedure.

It is important to understand that this project's ultimate goal is to provide 24/7 green energy matching for each customer. Therefore the inclusion of the entirety of the clientele is not only desirable but essential. This process however will be accomplished gradually by identifying energy intensive loads and practises and ranking them accordingly.

Finally after the data are processed and the desirable features are selected and properly combined, the energy matching algorithm needs to be implemented. The algorithm will need to combine records from the production and consumption tables to appear like table 1. For this reason some types of joins must be executed, meaning that the use of SQL and specifically Spark SQL will be advantageous. The selection of spark SQL in particular is based on the large volume of data used as well the fact that it is possible to create and run spark sql queries directly from pyspark.

3 Table Documentation

In order to optimally utilize all the available tables in our analysis, the contents of the table fields must be thoroughly documented. After the fields have been explained, the ETL scripts used to clean and curate these fields will be discussed. At that point we have to filter out data points whose values can not be used during the analysis phase. This task is made easier by studying and documenting the contents of the fields available. This way we can decide which field values are expected and which ones are unexpected and should therefore be dropped. Additionally, we can understand what format the data should have and whether a record should be modified rather than dropped.

Our initial data is stored in four different main tables. Additionally the consumption data are divided in fifteen customer address specific subtables. The columns of each table along with an explanation of each column are presented below.

3.1 Mapping Process

The mapping process describes the mapping of a single energy production record to multiple energy consumption records. It differs from the energy matching process in that this mapping produces potential transactions that might not be actualized. For the final energy matching table to be produced , which describes how much of the energy produced is provided to each consumer, the Ranking Process must first be performed.

3.1.1 Raw Production Table

We begin with the raw production table. This table contains energy production data from a single production site. As table availability increases, this will become a production subtable. However the fields of all production subtables is expected to be the same.

Column Name	Column Description
delDay	Deletion Day or Day of Energy Measurement
delHour	Deletion Hour or Hour of Energy Measurement
delQrt	Deletion Quarter or Quarter of Energy Measurement
unitName	Production Site Address
volume	Energy produced during this date and time as measured during a later time
fileId	The time after which the volume measurement occurs expressed as a string object
updateDate	The date at which the volume check occurs expressed as a date time object

Table 3: Production Table Column Descriptions

DelDay contains datetime objects expressing the date when the energy was produced. DelHour contains integers from 1 to 23, expressing the hour in which this energy was produced. DelQrt contains the integer values {0, 15, 30, 45} expressing the quarter in which the energy was produced. The measurement itself was taken at the day of production but the exact time of measurement is not known. UnitName contains strings expressing the name of the production site. For this analysis we only have one production site therefore this field's value will be constant throughout.

Volume contains floating point values that must be validated be non zero and with exactly two decimal points. These values express the energy produced at the specified date and time.

FileId is a string with three possible values 'D' meaning day, 'W+1' meaning one week later, 'W+7' meaning seven weeks later. These values express how long after the initial measurement took place, the verification check was performed. A measurement check is performed by a third party in order to verify the accuracy of the energy producer's initial production claim. The most accurate value check is the one performed at seven weeks after the initial measurement occurred.

Finally updateDate contains datetime objects expressing the date at which the volume check is performed. It will be later checked in 4.1 that the date expressed by the fields "delDay", "delHour", "delQrt" plus the days passed after the initial measurement, as expressed by "fileId" is equal to the date shown in "updateDate".

The quality of a record will be later checked based on the percentage difference between the three checks. The best quality metric will be given to records for which the difference between the day after check and the week after check are below 2%. In case this check fails, meaning that the difference between the day after check and the week after check is greater than 2%, it is examined whether the difference between the week after check and the the seven weeks after check is below 2%.

Any records that don't meet any of these two conditions is dropped as the measurement accuracy is not sufficient for further analysis. Additionally all the records that don't follow any of these two conditions are stored on a separate table as insights to be investigated in the future.

As far as the data format is concerned, it must be validated that the production table's energy is not zero, as there is no use of such records during the matching process. In practical terms an energy production of zero can not be defined for 15 minute periods. In other words, given that each record holds a 15 minute period's aggregated production, no energy value can possibly be equal to zero. This means that any zero values found where most likely substituted from NaNs or are the result of faulty equipment In any case these values can not be utilized during the analysis process.

3.1.2 Raw Consumption Table

Having documented the production table columns, we move on to the fifteen consumption tables. Given that the fields for all these tables are the same, only one tabular is provided.

Column Name	Column Description
BuildConsumption	The consumption of that specific site at this date and time.
Timestamp	The date and time at which the energy was consumed.

Table 4: Consumption Table Column Descriptions

The timestamp column contains datetime objects of the format "month/day/year". The different timestamps must be one minute apart. This is checked later in 4.2 and records that don't conform to this requirement are filtered out.
Records that do not have an one minute difference from the immediately previous and next records are stored as an insight and will be further investigated.

Specifically two tables are produced. One table for records that have a greater than one minute difference from their adjacent records and one table for records that have a smaller than one minute difference from their adjacent records. It is expected however that most records captured will be of greater difference than one minute from their adjacent records.

This could among other reasons be the result of infrastructure related issues such as latency between sending of energy information from the responsible party and posting that information to the database itself. These database related infrastructure limitations should be resolved as soon as possible.

Other possible reasons for this expected surplus of over one minute records are consumption site related infrastructure limitations. Specifically while all households in our dataset are equipped with smart meters not all are rigorously maintained and regularly upgraded, leading to obsolete and often buggy equipment. The resolution of these limitations is not the companies responsibility and therefore the time required for it can not be estimated. Until these limitations are resolved, the requirement on the time difference between adjacent records should be relatively lenient.

The BuildConsumption column contains the energy consumed at the specified date and time as floating numbers. These values must be non zero and have exactly two decimal points. Therefore relative checks must be implemented in order to filter out any invalid records. All energy values found that do not match those criteria are considered insights and will be investigated later on.

3.2 Ranking Process

After cleaning the production and consumption tables, the tables used for the ranking procedure must also be processed. These tables are the customer specifications table and the customer invoices table.

The customer specifications table contains potentially insightful client features and therefore is essential for our analysis. This table is the result of the invoices table, the consumption table as well as other tables in the data warehouse. Using the fields included in this table we can capture specific customer behaviors and prioritize providing the energy matching service, to the clients that will extract the most value from it at this early stage.

Again the point is to provide the energy matching service to all our customers, yet the need for net zero solutions differs among them which calls for the implementation of a ranking algorithm.

The invoices table is an aggregation of invoices for the fifteen clients whose data is currently available. This table will be used primarily to extract extra customer specifications and insights that will be useful in implementing the ranking algorithm.

At this point it should be specified that the number of features stored for every customer is substantial, including many duplicate, obscure or obsolete fields as well. These features are stored in a relevant company database from where they need to be extracted. Admittedly, not all these features are relevant for the ranking process.

For this reason, considerable effort was placed towards selecting the appropriate fields from this pool, otherwise known as the feature engineering procedure. This process of course requires having a thorough understanding of the contents of each field. In other words the meaning of each field must first be established, before deciding whether or not the field could be beneficial in the ranking process.

However the documentation available for each field was minimal, consisting only of the field name. Consequently, it was essential that proper column documentation was written before continuing with the feature engineering procedure. The documentation tables have the following fields:

Field Name	Field Description
Column Name	The name of this column as it was written in the database.
Column Description	A detailed description of this column.
Column Type	The Python dtype of this column.
New Column Name	The new name given to this column.
Comments	Potentially useful insights about this column.

Table 5: Field Documentation Table

As seen above, apart from giving a detailed description of each field, other steps are taken in order to accomplish a thorough understanding of the database's contents.

Firstly it is decided that providing the column dtype is crucial in understanding the field's meaning. The term dtype refers to the python type of the objects stored inside this column. This information will undoubtedly provide a better understanding of what the column values represent as well as help check for unexpected entries.

It should be mentioned at this point, that while researching the theoretical dtype of these fields, it has become apparent that the above differs from the actual dtype. This finding will be significant in any future attempts to restructure and clean the database, as it will lead to inquires regarding fields with dtype discrepancies.

Another noteworthy field in this table that should provide better understanding of the column's contents is the "column new name". The necessity for this field comes from the realization that the column name is the first observation one would use in order to comprehend the meaning of values stored in this column. Additionally, the names that were given to each field before are obscure. Therefore any attempt at making the database fields more transparent must start with the renaming process.

Finally the column ”Comments” is essential for any further analysis and investigation that will lead to potentially useful insights. Specifically this column will be used to find fields the meaning of which has not been entirely clear. Furthermore columns that only have null values will have relevant comments in order to be dropped out later on. Lastly any duplicate columns should also have relevant comments in order to be dropped out as well.

By duplicate columns we refer to columns the contents of which have already been inserted in the database by another field. During the data cleaning phase a specific field among the many with the same content will be chosen and maintained in the cleaned table, while any other will be dropped.

These three conditions will be used to find any columns that are unsuitable for the energy matching process. Specifically the relevant comments mentioned above, will be properly parsed to collect columns that match these conditions. At this stage the comments will only be used to find unsuitable columns for further analysis. However any other methods that require a thorough understanding of column characteristics will also utilize the ”Comments” column.

3.2.1 Clean Customer Specifications Table

At this point the fields selected for the cleaned customer specifications table are presented. These fields are chosen, using the comments field of the corresponding documentation table. Specifically the columns chosen are the ones remaining after dropping out all unsuitable fields.

Column Name	Column Description
Επωνυμία	Clients full name
METERSTATUS	Whether the meter is active or inactive
RealRepDate	The date at which the company started representing this meter.
RealEndDate	The date at which the company stopped representing this meter.
DateSigned	The date at which the contract was signed by the client.
Activation_date	The date when HEDNO verified this meter.
MonthlyMWh	Average monthly consumption in megawatt-hours.
MonthlyDayMWh	Average morning consumption per month in megawatt-hours.
MonthlyNightMWh	Average night consumption per month in megawatt-hours.
EuroPerYear	Total annual revenue in euro from this meter.
RealMwhPYear	Total annual consumption in megawatt-hours.
ExYearlyEst	Estimation of last year's revenue from this meter in euro.
AvgMwhPYear	Average annual consumption in megawatt-hours.
AEuroPerYear	Average annual revenue in euro.
MonthsInContract	The months since the contract was signed.
PartnerCategory	Whether this is a B2B or B2C client.
Invoiced ΗΜΕΡΕΣ ΚΑΤΑΝΑΛΩΣΗΣ	The number of consumption days in this invoice.
Invoiced MWH ΗΜΕΡΑ	The total morning consumption in this invoice in megawatt-hours.
Invoiced MWH ΝΥΧΤΑ	The total night consumption in this invoice in megawatt-hours.
ΚΟΙΝΟΧΡΗΣΤΟΣ ΧΩΡΟΣ	If the meter is located in a communal space.
TM	The squared meters of the building where the meter is.

Table 6: Cleaned Customer Specs Table Part 1

Given the fact that the number of fields in the raw table is considerable it was decided to directly provide documentation for the cleaned table. The rest of the fields have been omitted because they either have limited descriptions, are duplicates of fields presented above or have inadequate non Nan values.

The table above is derived from the invoices table. Specifically this table is updated in the group database every time the corresponding invoice table is updated. This means that for this project there will be a new customer specs table every month. The database table contains extracted or created customer specifications used in various client related inquiries.

The table inside the group database is much larger, yet for the purposes of this project the raw customer specifications table, only contains a fraction of the available fields. The selection process is based on the perceived utility of each field during the implementation of an energy matching algorithm. This selection was completed by the chief artificial intelligence officer of the company and therefore it is beyond the scope of this thesis.

Every energy meter is represented by a specific energy supplier at a given date and time. The representation of an energy meter is preceded by the the meter verification and the signing of the energy contract. In other words in order for a meter to be represented by a company it must first be verified by HEDNO. After that the meter's owner must sign a contract with a specific energy supplier. Only after these two requirements have been satisfied, it is possible for an energy supplier to represent a meter. The time difference between when a contract is signed and when the corresponding meter is represented may vary but averages to about a month.

It is important to mention at this point that each record in the invoices table or customer specifications table is about a specific meter and not a specific customer. In other words in case a client owns multiple energy meters (i.e. if a client owns or rents a number of buildings) then for each one, a different record will be present in these two tables. This means that for the process of energy matching to be implemented, the records pertaining to the different meters of a specific owner must first be aggregated.

Apart from the date that a meter starts being represented by the company there is also available information regarding the date the representation stops. In case the meter is still represented at the time the group table is updated then this field is set to an arbitrary value, in this case '01-01-2099'. From this information we can deduce that this field can only be utilized when the client has left the company or moved.

In other words, when this field's value is not equal to the above default value then the value of the churn field must be 1. Of course it is easily understood that for the field to signify that a customer has left the building or the company, the value stored, must be significantly smaller than the default one by several years. If that is not the case then we can assume that, either the default value for meters still represented has been changed or an insertion error has occurred.

As already stated the date when the client signs the contract differs from the date when the company actually starts representing the corresponding meter. This disparity is caused mainly by the need for information verification both from HEDNO and the company. While not all information provided by the client are verified a select few are, before the company can proceed to represent the corresponding meter.

One of the most important factors to check is whether the client has any debt with another energy supplier. Indebted clients can not be represented, apart from very specific circumstances. Therefore in order for the meter of the client to be represented the payment of the previously incurred debt must be completed in full.

Apart from the company's verification process additional confirmation is conducted by HEDNO. This confirmation process is centered around the state of the meter to be represented by the company. HEDNO will check whether the meter has not been illegally tampered with and that any modifications to it have been applied by authorised personnel.

The meter's integrity confirmation process may vary in time depending on the meters age, whether the meter has changed owner or if the meter has been recently upgraded. For an upgraded meter and specifically a smart meter, the confirmation process is much more swift, since the majority of checks can be completed digitally. For older meters there is a necessity for HEDNO employees to physically test the meter. This can pose extra difficulty in the verification process particularly when the building is in a remote location or the meter is out of sight.

As already stated each record in the invoices and customer specifications tables corresponds to a specific meter. Therefore, if a client has multiple meters that are represented by the company then for each meter a different record will be present in these two tables. While in most fields we expect the records relating to meters of a single client to be similar, that is generally not the case with the fields 'EuroPYear' and 'AEuroPYear'.

For example if the client in question is in the b2c category then it could be the case that the two meters represent the client's permanent residence and country house. In such a case the two houses would have significantly different energy consumption profiles throughout the year and therefore contribute different total annual and average annual revenue to the company.

Although the two different meter records should be aggregated before the matching process commences, disparities such as these as well as the energy profile of the different meters should be kept as information to the resulting client record.

This recommendation is based on the fact that country homes are used mostly on holidays making their total energy consumption small enough to be completely covered by green energy. Therefore given that it is the goal of the company to provide each customer with 100 % net zero solutions as soon as possible, country homes should be prioritized in the matching process provided that the customer is considered loyal.

While prioritizing different loyal customers in terms of the amount of green energy provided to them, the number of client meters served by the company is a key factor. Especially if one of them is a country home. Another important factor to be considered is the customer's consumption volatility over the years.

For a customer whose behavior deviates significantly between the years, the ability to provide 100 % net-zero solutions might be more intricate than for other customers. Additionally consumption volatility could indicate payment delays from the customer, another factor that should be considered in the matching algorithm.

In order to compute the desired consumption volatility the field 'ExYearlyEst'can be of use. Specifically, given that the customer is punctual the difference between this field and 'EuroPYear' should be computed. If this difference is over an arbitrarily selected threshold then the customer should not be prioritised during the matching process.

In case the customer is not punctual then the field 'EuroPYear' can not be used for this computation, since the total annual amount paid might not match the customer's consumption. In that case the volatility should be calculated using the fields 'RealMWHPYear' and 'AMWHPYear'. However regardless of the result of this calculation, the lack of punctuality by the customer should be taken into account during the matching process.

Apart from consumption volatility another important factor to take into account during the matching process is revenue volatility. Customer's who provide similar revenue to the company across the years should be prioritized since they are considered a safer investment. It should become clear that this strategy does not take into account the revenue itself, just the disparity between annual revenues from a specific client.

In that sense there is no difference between a high payer and a low payer under this strategy, as long as their computed volatilities are equal.The revenue volatility will be computed as a percentage of the fields 'EuroPYear' and 'AEuroPYear'.Specifically the percentage difference of this year's annual revenue from the average annual revenue will be calculated.Then based on selected thresholds the customer's will be categorized in different classes of revenue volatility. This way clients whose revenue is considered less volatile than others, based on this classification system, will be prioritized during the matching process.

The central field for the matching process is 'MonthsInContract'. This field will be utilized as a metric of a customer's loyalty. Additionally as we will see later on, metrics regarding the loyalty of an entire group will be computed using this field. Furthermore any other strategies used in order to achieve prioritization in the matching process, will be applied after making use of this field.

For these reasons the verification of this field's values is paramount. To that end it is important to note that this field is related with the fields 'DateSigned' and 'Churn'. Specifically, if churn is not 0 then 'MonthsInContract' is at most equal to the months between date signed and the date the current invoice was published. It is reminded at this point that each customer specifications table corresponds to a specific invoice table. In the invoice table there is a field for the last invoice published for each meter represented by the company. Therefore to find the date when the last invoice was published, one must look for the record corresponding to this specific meter in the invoice table.

It must be said however that such checks between fields are based on the assumption that all other input fields used to verify a specific field, in this case 'MonthsInContract' are valid. This is an assumption that should be further investigated to ensure the validity of the field checked.

Having discussed the essential fields that should be used during the matching process some additional project features and terminology are provided.

First of all, the green energy matching project at this time is targeted towards households. However the benefit of using and publishing the use of green energy and net-zero solutions for businesses is considerably greater. Given the amount of fossil fuels produced by big businesses, especially those belonging in heavy industries, it is paramount that the algorithm will eventually be implemented for the B2B setting as well.

The characterisations b2b and b2c refer to specific products or services provided by a business, based on the type of organization receiving that offering. A b2c offering meaning business to client, is a business offering provided directly to households. On the other hand a b2b offering is provided to businesses.

However before transitioning to serving business clients instead of households, one must identify the differences in customer specifications between the two. Specifically studies must be initiated in search of the most insightful fields in the invoice and customer specifications tables, in characterising businesses. Through this characterizations, a prioritization mechanism, equivalent to the one explained above for the b2c offering,can be initialised.

Even though the majority of strategies already discussed in the b2c setting are applicable in the b2b scenario some strategies should either be altered or changed altogether. For example the strategy regarding prioritizing customer's that have their country homes served by the company is not applicable. This strategy can be modified to fit the b2b framework by prioritizing smaller businesses or subsidiaries. The reasoning behind this strategy relies on the fact that smaller businesses produce less fossil fuels and have smaller energy demands, making it possible to cover their consumption needs completely with the use of green energy solutions.

In terms of ecological solutions, the difference between morning consumption and night consumption can offer considerable savings to any interested household or organization. A side project that could be beneficial to the matching process is based on identifying specific energy loads that could alternatively be completed at night. Firstly it needs to be identified whether a clients morning consumption is greater than the night consumption.

If that is the case then a second step is to identify specific energy loads that can be transitioned at night. Finally a set of recommendations for a smooth transition will be given to the customer. Given that a client will indeed follow the provided recommendations, his/her involvement in the energy matching process should be increased. In other words clients that are actively seeking to adopt an ecological attitude by accepting company recommendations in regards to energy consumption, should be considered as loyal and be rewarded accordingly.

Given the fact that energy provided at night is cheaper, it will be recommended to customers to transition any viable energy loads at night. For example in the b2c setting, washing machines can be scheduled to start at night and therefore utilize night energy. For manufacturing plants used by business clients, it is recommended that energy demanding tasks should be run at night when possible. Additionally clients that follow said recommendations should be prioritized during the matching process.

It is also advised that for business clients that follow company recommendations, additional specific suggestions should be provided. To that end studies must initiated by the group in order to inform interested customers about the options available to diminish their carbon footprints. This study must take into account the needs and goals of each interested business partner and compile an educated list of suggestions.

Having given information on all the important fields of the first part of the customer specs table, the second part of the table is now presented.

Column Name	Column Description
Φ/Β	Whether the client has installed a photovoltaic unit in the meter's location.
ΟΦΕΙΛΗ ΜΕΤΡΗΤΗ	The amount of euros owed from this meter
aggrpwr	Agreed upon power measured in KVolt Ampere
Churn	Whether the resident is no longer a client or has moved out.
MeterPostalCode	Postal Code of the area where the meter is located.
SYSTEM	Area and type of network
ΕΥΑΛΩΤΟΣ ΠΕΛΑΤΗΣ	If the client belongs to a vulnerable minority.
Physical_Street	The street of the building where the meter is located.
Phys_Number	The street number of the building where the meter is located.
Προηγ.Πάροχος	The previous energy supplier of the client.
MeterInvCycle	Invoice publication date.
Serglocation	Street name and number of the building where the meter is located.
EmailBill	Whether the client receives the invoice via email.
MYPROT	Whether the client uses the 'my protergia' service.
Region_Longtitude	The longitude of the meter's region.
Prefecture	The prefecture of the building where the meter is located.
Region	The region of the building where the meter is located.

Table 7: Final Customer Specifications Table Part 2

As already explained the field ΟΦΕΙΛΗ ΜΕΤΡΗΤΗ, which can be translated as 'meter debt' indicates the amount of money owed by the customer who owns this meter, or is a resident to the building where the meter is located. This field is always greater or equal to zero and can only be zero when the resident/owner owes no money to the company.

It is important to note that when an invoice gets published, this field instantly becomes greater than zero. In other words, it is considered that the client owes money even if the invoice payment deadline has not yet passed. This field should be consulted during the matching process. Specifically based on the expected monthly consumption of the client, a debt threshold must be set. During the matching process, clients that have exceeded their respective thresholds should not be prioritised.

The threshold should be the result of the clients average monthly consumption times a leniency factor. This leniency factor should be greater than one and a function of the clients loyalty and ecological activities. For example a client that has been in contract with the company for a significant number of months and has participated in ecological activities such as purchasing guarantee of origin certificates, should be assigned with a greater leniency factor.

The threshold should also depend on the number of meters the client has registered with the company. Specifically the level of increase in the debt threshold should depend on each building's features. For example if the two buildings registered have similar squared meters then the final threshold should almost double. Of course other parameters such as the months each house is used should also be taken into account.

This field should be utilized in conjunction with another field named 'REPAYMENT_STATUS', present in the invoice table. This invoice field as its name suggests, indicates whether or not the current invoice has been paid on time or not. The field's dtype is string and contains the values 'ONTIME' when the client has paid the previous invoice on time or 'OVERDUE' otherwise.

This field should first be recast to store the value one when the client has paid on time and zero otherwise. It is suggested that the probability the client will pay on time is approximated. This can be achieved by averaging over all the values of 'REPAYMENT_STATUS' in the invoice table after grouping by client name.

In case the debt threshold mentioned above is surpassed then the level of deprioritization should depend on this computed probability. In other words for clients that have surpassed their corresponding debt threshold, their final loyalty class will depend on the probability of paying on time. This means that if two different clients have both surpassed their debt thresholds, but one is more likely to pay on time than the other, then their resulting loyalty class should reflect this disparity on the likelihood of paying on time.

The reason that this distinction is suggested, is based on the group's values regarding loyalty and understanding. Clients that generally pay on time, but have occasionally surpassed their corresponding debt thresholds should not be demoted. At the same time clients that have not been punctual, yet their on time payment probability is gradually increasing, should be promoted in a higher loyalty class.

To be more precise clients that have consistently surpassed their debt threshold but show signs of improvement in terms of punctuality, should be promoted regardless. The most effective method of detecting this behavior modification is by computing the on time payment probability in two different time-frames. One for the last quarter of this year and another for all invoices published before that.

The reason for this distinction lies on the difficulty in detecting a substantial change in a mean value when the computation involves a considerable number of records. In other words if the probability was compared to a previous version of itself to pinpoint a behavioral change then for the latest computation all values would be given the same weight as the latest ones. This however would lead any change in the on time payment of current invoices to be muffled by the past behavior of the client.

This scheme is inspired by positive reinforcement strategies in hopes that the perks of a higher loyalty class will decrease the occurrence of surpassing the debt threshold by these specific clients. Additionally it could even be the case that this strategy could help increase the on time payment probability of other clients, resulting in an overall elevation in regards to the loyalty class of the clientele.

The problem that arises with this scheme is that it inadvertently provides perks to customers who have no intention of improving their punctuality. For example, a client who has consecutive previous unpaid invoices and pays the latest one belongs in this category of unintentionally benefited customers. A client like the one described above will indicate a difference in the two computed mean values, thus will qualify for a promotion in the loyalty ladder.

Of course this scheme is specifically designed for customers that make an effort to improve their punctuality and therefore should not reward clients that fit the description given above. In order to differentiate between the two categories and provide perks only for the intended group, an additional field needs to be utilized as well. For the scheme to be precise in the category of clients that it promotes, the field 'PAYMENT_STATUS' should also be used.

The field 'PAYMENT_STATUS' of the invoice table indicates whether or not a client has paid the current invoice, at the time the table was created. The field takes two values 'PAID' when the client has paid the current invoice and 'UNPAID' otherwise. While it's dtype is a string, integer values can be extracted from it. Specifically the field can be recast so that it has the value zero when the client has not paid the current invoice and one when the client has paid the current invoice. This way a similar computation with the one described above for the field 'REPAYMENT_STATUS' should be considered.

The recast field should be averaged to provide the probability of payment. This probability should be used in conjunction with the probability of on time payment which was computed above in order to distinguish between the two categories of clients. Specifically in case the clients current behavior indicates deviation in terms of punctuality from historical records, the following procedure should be followed.

If the client has a high probability of payment then the client should be promoted regardless of the number of times the corresponding debt threshold was surpassed. If the client has not surpassed the debt threshold an adequate number of times in a given time frame but has a high probability of payment and on time payment, then the client should not be moved. Finally if a client has surpassed the debt threshold a considerable number of times and also has a relatively small probability of payment then the client should be either demoted or maintained in the same class, depending on the probability of on time payment.

The case in which the client has not surpassed the debt threshold a considerable number of times in a specific time frame (i.e. the last quarter) but has frequently paid on time at this time frame is considered as a transitional period. This is a period in which the client has already shown an increase in punctuality for a consistent period of time, meaning that the conditions required for a loyalty class promotion have already being met.

In other words the client has already being promoted for the increase in punctuality and it is observed whether the client maintains this positive habit. It is recommended that this client is not promoted further at this time, but that the consistency shown is continuously monitored before any other related actions are taken.

Another critical parameter in regards to classifying a customer's behavior as wasteful or ecological is the area and type of network that exists where the corresponding meter is located. Specifically it must be taken into account that in areas with colder winter months clients are expected to consume more energy during these periods. Additionally, in areas where the network infrastructure is ill maintained, excessive energy consumption is already frowned upon by local authorities and network experts.

For these reasons clients whose corresponding meters are located in the north of the country are expected to have a greater consumption during the winter months than customers in the south. For this reason during the matching process and specifically during the classification of each customer as ecological or not this feature should be considered.

As already stated ecological behavior should be rewarded in similar fashion as loyalty. Particularly consistent ecological behavior should be recognized and promoted by the use of gifted green energy. But the rules implemented to decide whether a customer exhibits ecological habits should also keep in mind the client's location. Therefore we caution towards hastily classifying a customer as being excessive in regards to the consumed energy during the winter months. Specifically in cases when the customer's meter is located to the north of the country.

Similarly special attention should be given to clients that reside in areas where the network is old and ill-maintained. If network experts have expressed concerns regarding the excessive energy consumption for reasons of network limitations then the behavior of the clients located in these areas should adhere to these concerns and warnings. For this reason rules regarding the classification of clients as ecological in these areas should be stricter. Specifically clients in these areas who exhibit energy patterns that surpass the recommended or expected patterns by network and energy experts should not be considered to be exhibiting ecological traits.

All these information can be given by the customer specifications field named 'SYSTEM'.This field as already explained describes the area and type of network that exists where a specific meter is located. Therefore the rules specified above can be implemented using this field in addition to geographic information and technical knowledge regarding the quality of networks across the country.

In accordance with the company's social responsibility and values, the company provides specific perks for disadvantaged individuals or households. People and households with a low annual income, people with disabilities and households in which multiple minors reside are some of the minorities that belong to this group.

Among other things the company provides a specific invoice for these individuals and households with a lower per kilowatt pricing model. It is recommended that this group of people are given preferential treatment when it comes to loyalty perks and green energy related gifts. Specifically it is recommended that each subgroup that is invoiced in this way is also provided with night electricity perks.

In other words it is recommended that individuals and households of this group are provided with smaller energy invoices when the electricity is consumed at night. This perk is not generally available to all company customers with the appropriately upgraded meter by default, but it is recommended that it becomes available for all individuals and households in this specific group. This means that while all customer's with the appropriate meter can have smaller energy invoices when utilizing night consumption, this specific scheme will focus on additional cost reductions only for this group.

While the customer's ability to capitalize on night consumption, does not directly depend on an energy supplier, there are actions that the company could take to expedite this process. In order for a client to be invoiced differently for morning and night consumption, a specific type of energy meter needs to be installed. This installation process is not the responsibility of the energy supplier but of HEDNO.

However the installation of the required electricity meters that detect and differentiate between morning and night consumption can be affected by the company. Specifically the company can provide financial motivations for the customers regarding the installation of the required meter. Additionally the company can suggest alternative energy consumption patterns that take advantage of the resulting distinction in pricing in order for the customer to save money.

For example the company could provide a loan for some or all of the installation related costs occurred by the customer. Additionally the company could provide the clients with suggestions about specific morning energy loads that could alternatively be consumed at night. Additionally, the company could indicate all potential cost reductions achieved by following the provided suggestions. This final process could yield optimal service adoption in case the specific energy needs of each different client in this group is monitored and reflected by the provided suggestions. In other words the personalizing of suggestions based on the needs and habits of each client could increase the number of customers that are wiling to install the required energy meter.

While this scheme is suggested for people that belong to a very specific group, in order to reflect the company's commitment to inclusion, the personalizing of energy suggestions could also be adopted as a policy for the entire clientele. This could not only lead to an overall increase in the use of energy meters that provide this distinction in pricing, even without the extra cost reductions, but also in the increase in adoption of all suggested ecological energy solutions in general. This in turn could help the company detect the ecologically minded individuals inside the client base and reward them accordingly via the green energy matching process.

Additionally any other rules regarding the determination of a client's loyalty class should be applied with relative leniency when that is possible, when it comes to this group of people. This process will reflect on the company's values and support of inclusion and inclusion related policies.

This will result in increasing the approval of the company and group among socially aware clients, providing them with further motivation to participate in any activities organized or supported by the group, including ecological activities and green energy matching projects. This could in turn lead to an increase in the percentage of punctually paying customers, from the socially aware portion of the clientele. Additionally, it could result in the overall adoption of green energy related solutions and suggestions made by the group.

When it comes to the use of geographic information during the matching process the field 'SYSTEM' provides almost everything that could potentially be used during the matching process. However for reasons of completeness some additional information are also given for the field 'Physical_Street'. This field contains the street names where the invoiced meters are located.

An important distinction that needs to be made at this point is between the street where the meter is located and the street where the invoice is sent. While in most cases these two addresses are the same they could potentially differ. For example in case the meter belongs to a student and the invoice is sent back to the student's parents to be paid. Moreover it could be the case that the home is provided as an accommodation from an employer. In cases such as these the invoice is sent to a third party to be paid and not to the individual or household which consumes the energy.

At this point there is no field present in the customer specification table for the location where the invoice is sent. However such information could be potentially useful, specifically in case the two fields will have different values. In case the fields have different values then it could be argued that the customer's consumption habits might be affected from the fact that a third party is paying the energy invoices. Therefore in case the customer ends up in a different payment scenario the related information gathered about consumer behavior could be unreliable in order to ascertain the ecological mindset of the client. Therefore for clients for whom the responsible group for paying their invoice changes, the green energy related gifts that are provided should be reevaluated.

The meter's specific location is not useful during the matching process but the general area of the meter could be utilized. For this reason the information given by 'SYSTEM' are preferred over the ones provided by the field 'Physical_Street'. Based on this observation other fields that could be potentially useful during the matching process are 'Prefecture' and 'Region'. These two fields provide information about the general area of the meter, specifically about the region and the prefecture respectively.

The idea behind the suggestion to use these fields in the matching process is based on the fact that the consumption behavior between residents in a specific region or prefecture is expected to be similar. These information could be valuable in the effort to ascertain a customer's ecological mindset and loyalty class.

However in order to use these fields in the matching process it is first important to understand their contents and establish their integrity. To begin with, a region is defined as an area of land that has common features. These features can be natural, such as climate or landscape. They can also be artificial, such as language or religion. [1] In Greece there are currently 13 regions 9 on the mainland and 4 island groups which are further subdivided into 74 regional units and 325 municipalities. [2]

This last administrative division of Greece has been established in 2010 with the Kalikratis reform. With this reform the prefectures became obsolete giving way to regions. [3] Given the fact that the field names 'Prefecture' and 'Region' are present in the group database, it is important to ascertain whether the group database has been correctly updated after the Kalikratis administrative reform of 2010.

In other words it needs to become clear whether the regions as defined in the group database are in agreement with the administrative reform of 2010. At the time of writing no IT professional or group database expert currently employed by the group, was able to verify the integrity of the field "Region" in its entirety. For this reason before using this field in the matching process it is essential that this verification task is carried through. Additionally in case any field values are different from the expected value, the corresponding row should be dropped or the field's contents in this record should be altered based on the available data.

Similarly to the field 'Region' there is a number of fields in the customer specifications table that could potentially be utilized in the matching process but first require integrity verification. Another field that belongs in this category is the field 'MeterInvCycle'. This field is expected to store the invoice publication dates. For this reason the presumed dtype of this column is 'date' or 'datetime'.

However upon further inspection of the column it becomes apparent that the column contains a mixture of string and integer values. For this reason before using this column it is essential that a consultation with the responsible group database expert should be scheduled in order to reveal the reason for this discrepancy as well as methods to remedy it.

Once the field has been correctly updated it could be used to verify the invoice field 'INVOICE_EXPIRATION_DATE' which itself could be utilized during the matching process to determine the loyalty class of each customer. Specifically the field 'MeterInvCycle' could be compared to 'INVOICE_EXPIRATION_DATE' in order to verify whether the values contained in it are after the date the invoice was published. The field 'INVOICE_EXPIRATION_DATE' as its name suggests contains the final dates at which the invoices can be paid on time. Therefore each of these dates must be later than the corresponding date the invoice was published.

The field 'INVOICE_EXPIRATION_DATE' in turn can be used to verify the values contained in the invoice field 'PAYMENT DELAY' which can also be used in the matching process. The field 'PAYMENT DELAY' as its name suggests includes the time before or after the payment deadline at which the invoice was paid. The field takes integer values and in case the customer has paid the corresponding invoice before the aforementioned deadline this field contains a negative value.

Specifically the field 'PAYMENT DELAY' can be verified by subtracting the values of the field 'INVOICE_EXPIRATION_DATE' by the date the invoice table was updated. The result of this operation should be equal to the number of days between the invoice payment deadline and the date the table was updated, which should be equal to the values of 'PAYMENT DELAY'.

Another field that could be useful in verifying the integrity of the data is 'SergLocation'. This field contains the customer's full address, combining the street name and the street number. For this reason it is comparable to the concatenation of the fields 'Physical_Street' and 'Phys_Number'. This means that these two fields could be used to check the integrity of the values stored in 'SergLocation'.

The difficulty in this check is based on the fact that the field 'SergLocation' has partial nulls. Therefore in order to acquire the values in these specific records, it is important to consult an IT professional with knowledge of the processes that could result in a null value as well as the correct 'SergLocation' value.

As already discussed a partially null field should be dropped in case the number of nulls exceeds eighty percent of the total records. However for this field this threshold is not reached and therefore it is suggested that the missing values are updated with the help of the individual or individuals currently employed by the group that are responsible for database maintenance and ETL processes.

Even though SergLocation might not be used in the matching process, it is important to have a specific field present in the customer specs table, that has the full address of each customer. To that end, it is recommended that after the verification of the field's integrity the postal code of the area where the customer's meter is located should be appended to 'SergLocation'.

Even though the location of the customer should be included as a field in the customer specifications table its significance is limited to providing the expected consumption patterns of the customer's general area. Any suggestion given to the client regarding demand response processes, ecological events or green energy matching updates should be provided digitally.

For this reason it is essential that the client has provided the group with his email address in order for any recommendations to be forwarded to the customer. Therefore clients that have provided their email address to the group should prioritized during the matching process in order to motivate the clientele to digitize. Using the fields available in the customer specifications table this can be inferred from the field 'EmailBill'.

The field 'EmailBill' indicates whether or not the client receives the electricity invoice via email. The field contains integer values and specifically is one when the client does receive the invoice via email and zero otherwise. This representation can be utilized to find a digitization score of the clientele by computing the average number of clients that receive the invoice via email. It is in the interest of the group to increase the digitization score of the clientele. That is because as already explained, the ability to provide suggestions and recommendations to the clientele regarding green energy initiatives is enhanced when using digital methods of communication.

In other words it is easier to reach a plethora of clients using digital mediums since the physical location of a client will not restrict receiving the message. Therefore the digitization score should be frequently monitored and actions should be taken in order to ensure it is maintained within an acceptable range.

Having said the above it is also essential to provide the correct dtype for each field in the table. For this reason it is recommended that the field's values should be replaced with TRUE or FALSE before any processing or computation based on this field is performed. After that the computation and update of the clientele's digitization score will be achieved by temporarily casting the field as integer.

Apart from providing general recommendations and suggestions to the client via email or newsletter each client could also be provided with personalized suggestions to minimize energy consumption and achieve the corresponding net zero goals. This can be more easily achieved by using the platform Myprotergia. Myprotergia is a platform for management, understanding and engaging with notions and games relating to energy consumption and demand response. Through this platform based on confidential and safely encrypted customer data, the client can be given suggestions in order to reduce emissions, energy consumption and achieve net zero goals.

Therefore the groups customers should be motivated to register to the aforementioned platform. Additionally it is recommended that the clientele's digitization score should be computed based on the field 'MYPROT' as well. This field as the name suggest is one if the client has registered in the 'my protergia' platform and zero otherwise. Similarly with the field 'EmailBill' this field should be first recast to bool and then be used in the computation of the digitization score.

The final formula for the score should include both the mean values of the fields 'MYPROT' and 'EmailBill'. Specifically the digitization score should be the weighted average between the two mean values. The exact weights used during the computation will not be defined. However it is recommended that the weight corresponding to the average value of the field 'MYPROT' should be bigger than the weight corresponding to the average value of the field 'EmailBill'.

The reason for this is that the personalizing of suggestions should lead to greater overall savings in energy consumption for the clientele. Additionally the use of the platform by the customer's allows the group to distinguish the clients based on their ecological mindset. This in turn makes it easier to provide green energy related incentives to conscientious clients. Additionally it contributes to assigning a greater loyalty class and providing corresponding energy matching perks to customers that follow personalized suggestions. This means that the decision making process during energy matching, can be greatly benefited by the data given to the group from the 'my protergia' platform.

The final non null field in the customer specifications table is 'Region_Longitude'. Even though this field is not expected to be utilized in the future during the matching process, it is advised that it is maintained in the table along with the field 'Region_Latitude'. However the field 'Region_Latitude' is filled with null values and therefore should first be restored. This can be achieved by directly using the field 'Region', after its integrity has been verified.

Before explaining the specific method used to restore and verify the fields 'Region_Longitude' and 'Region_Latitude', we remind the reader, that at this point the matching process is based solely on processes using exclusively the field 'MONTHS_IN_CONTRACT'. All other methods and procedures mentioned above, are suggestions of features that could be implemented in the future, as part of the matching process algorithm.

The values of the field 'Region_Latitude' can be restored using the field 'Region' in conjunction with any map that provides longitude and latitude values. Additionally, the values of the field 'Region_Longitude' can be verified using that same technique. These two fields are not expected to provide any additional insights in regards to the localised customer consumption behavior of each region. This holds from the fact that, as already explained, these insights will be fully provided in the future by the field 'Region'. However it is recommended that the fields 'Region_Latitude' and 'Region_Longitude' are maintained in the table for reasons of completeness.

3.2.2 Ranking Filtering Considerations

At this time the invoice table is suggested not to be utilized during the matching process apart from the fields already mentioned in 3.2.1. Therefore for the remainder of this chapter we present the initial conditions that need to be met in order for a field to be dropped from the raw customer specifications table.

Moreover, we posit some considerations that should be taken into account before deciding to drop partially null fields present in the customer specifications or invoice tables. These considerations should also be taken into account during the implementation of the invoice cleaning script in the future. Below we explain the methods used in order to select the fields that will be present in the final customer specification and invoice tables. Additionally the reasons for dropping specific fields are listed along with processes to decide whether a field should be dropped, restored or maintained. Finally a note to the importance of field documentation is presented before thoroughly describing the implemented ETL scripts, as well as the specific methods implemented for each script.

The selection process relies heavily on the corresponding documentation tables and specifically on the comments field. As already stated this field contains information about each column in the original tables. This information can be used to further understand the contents of the tables. However at this point they are only used in order to indicate which fields are unsuitable for further analysis and therefore should be filtered out during the preprocessing phase.

The reasons for fields being dropped are:

- Field was not properly explained by the responsible database expert.

  The insertion of fields in the company database has been the result of the work of multiple engineers and IT professionals. Additionally the documentation for said fields is often limited or even non existent.For this reason there is a multitude of fields which are not properly documented.

  Furthermore there is little information available regarding the individual responsible for creating each field. Even though there are certain individuals with extensive knowledge on specific sections of the database, non is able to pinpoint to each field's creator or meaning. Consequently fields that can not be explained by a database expert will be dropped.

- Field is a duplicate of another field.
  By duplicate fields we define fields that theoretically contain the same contents as another field which already existed in the database when this field was inserted. Duplicate fields is a condition that often occurs in large databases since different engineers produce new fields without carefully inspecting the existing fields in the database.

  In order to ascertain which column should be considered as the correct one and which as the duplicates, the date when the field was inserted, should be taken into account. Additionally, the integrity of the values stored in each field is also a crucial factor, to classify the fields as duplicates.

  It has been observed that many duplicate fields contain a considerable number of null or wrongly altered values compared to the original field. Furthermore, duplicate fields provide no additional information compared to the original field to which they correspond.For these reasons duplicate fields are dropped from the original table.

- Field contains null values.
  Occasionally, a non duplicate raw table field might also contain missing values leading to a need for an alternative field to be inserted or the field to be altered. Null fields are most likely the result of bugs in ETL processes or manual table insertion practises.

  A number of fields contained in the group database are the result of some initial rudimentary processing methods, especially for values that are obtained from edge devices.These methods are prone to faulty behavior and unintentional functionality and could therefore output null values.

  Additionally, a number of fields in the database are inserted manually. This means that instead of an ETL pipeline or a smart meter providing the desired values, the column is populated with values that are entered directly from the keyboard. This approach if not followed by rigorous testing procedures can lead to a number of undesirable effects such as meaningless values, unexpected column dtype and null values. For these reasons fields that contain null values are dropped from the original table.

The null fields contained in the database can either contain only null values or a mixture of real and null values. Partially null fields are only present in the group database and not in the dataset that we could acquire given this project's scope. However it is important to document the field characteristics that should be taken into account when deciding if a partially null field should be dropped or modified.

The general methodology for dropping partially null fields is to drop a field if at least eighty percent of its contents are nulls. However there are other factors that should also be taken into account before dropping partially null fields. Below we underline the most important:

- Ease of replacement.
  If a field has more than eighty percent of its values as nulls but its contents can easily be retrieved by other means, then the field should be properly modified instead of dropped. Once the field modification occurs, it is essential that we provide documentation on the methodology followed to replace the values, should nulls reappear in a later version of the field.

  A number of fields are the result of an operation between other existing columns. That is one of the many reasons for insisting on proper column documentation. If a column's dependencies on other columns is documented and its values are partially or completely missing, then a retrieval process can be initialized using the documented formula that links these columns together.

- Importance in the analysis process.
  As already indicated, for any field which is partially null, the process of filtering is not straightforward. A key consideration regarding the filtering of such a column is it's importance in the analysis process later on. A column the content of which can substantially benefit the analysis process should be kept in the data set even if only a small number of records have an actual value in this field.

In general the importance of a field in the analysis process is determined by the following criteria:

- Non correlation between other explanatory variables.

  When trying to corroborate on the importance of a field one must first check its correlation with other explanatory fields in the dataset. A Pearson correlation test is the most common way to verify any correlations between the different explanatory variables. Alternatively one could conduct a Kendall rank correlation test or a Spearman correlation test depending on the specifics of the project.

  In case the test indicates non correlation between the fields this is evidence of the importance of said field in the analysis process. A field that does not correlate with other fields might be useful in explaining the variance of the response variable in ways that other fields could not. However non correlation with other explanatory variables is not enough to positively conclude the significance of a field.

- Correlation between this column and the response variable.

  Additionally the same correlation test as the one described above should be conducted between the response variable and the desired explanatory variable. If this test concludes that there is indeed correlation between the explanatory and response variables, then there is strong evidence of the importance of this field in the analysis process.

  Correlation between these two columns in combination with non correlation between this column and all other explanatory columns suggests that this field can indeed explain the variance of the response variable in ways that other variables can not.

  We caution the reader against interpreting such a test as a testament of causation between the two variables. Even though causation can not be proven by these tests there is at this point considerable evidence that the developed model will be improved when using this variable. Therefore it is paramount that this field is utilized to the point that it is possible, given the potentially large number of null values that it could contain, during the analysis phase.

- Metric improvements in the final model.

  The most conclusive evidence of the importance of a field during the analysis phase is given after developing the model. Once the model is developed its outputs are checked using a selected metric. The most common metrics used are accuracy, precision, recall or f1 score. Given the fact that the aforementioned column will include null values two models should be developed, one for records at which this column has a null value and therefore can not be used and one where it has an actual value.

  However as already explained a partially null field could have more than eighty percent of it's values as nulls. Therefore there is a possibility that the records with actual values in this field are not enough to perform a correlation test or develop a model altogether. In this case the null free portion of the dataset should be enhanced using oversampling or resampling techniques such as SMOTE or bootstrap.
  After that all the checks and implementations mentioned above will be possible leading to a decisive conclusion regarding the significance of this field.

- Actual or expected improvement in any relevant project K.P.Is.

  At this point in this project an actual machine learning model has not been implemented. For this reason in order to evaluate the algorithm's ability to complete the desired tasks, an appropriate key performance indicator must be created. Additionally this indicator should also provide insights regarding the importance of specific fields in the analysis process. Provided that our goal is to match energy demand with green energy for specific clients, our metric has to be derived from the task relative customer specifications that are available to a pilot scope project. The available project scopes are thoroughly explained in 5.1.

  In other words we need to find which client features are of greater importance for the energy matching algorithm. The final indicator could be a function of specific customer specifications or invoice fields. This indicator can be thought of as an algorithm scoring function, in the sense that it will be utilized to indicate the quality of the implemented algorithm over time. This means that the indicator will provide information regarding how significant customer specifications are modified over time based on the implemented matching process.

  One of the desired outcomes of this project is the maintenance and expansion of the loyal energy clientele of the group. Therefore we need to define a specific metric which will clearly indicate if the loyal customers

> will stay loyal or not and if the less loyal ones will be gradually improving or not in regards to that aspect.
>
> For this reason an indicator called loyalty over time will be defined. This indicator is computed as the average loyalty of the company's clients. The loyalty of a specific client is computed based on the number of months since the client has signed a contract with the company.

It is recommended that this indicator should be computed separately for the clients considered loyal and less loyal. This way there will be constant monitoring of the average loyalty of each group. After having defined this indicator, the goal to expand the companies loyal clientele can be thought of as being equivalent to maximizing the loyalty over time value in both groups. Given that the distinction between the two groups is defined by an arbitrarily selected threshold, the loyalty over time for the less loyal customers can only be at most equal to that threshold. This can be expressed as a maximization problem thus:

Let S be the set of loyal customer and S' be the set of less loyal customers.Additionally let f be the loyalty over time of a specific set of clients. Finally let T be the threshold that separates the loyal from the less loyal customers. Then the problem of expanding the company's loyal clientele is equivalent to:

$$\max_{C \in P(\mathbb{R}^n)} f(C)$$

Where $P(\mathbb{R}^n)$ is the power set of $\mathbb{R}^n$ and $n \in \mathbb{N}$.
In case the above maximization is specifically about the less loyal customers then it holds that:

$$\max_{C \in P(\mathbb{R}^n)} f(C) = T$$

This can also be interpreted as saying that to expand the company's loyal clientele a set of loyal customers must be produced that will maximize the loyalty over time. Additionally a set of less loyal customers must be created in such a way that the corresponding loyalty over time will be equal to the arbitrarily set threshold. These constructions can be achieved primarily by using the energy matching algorithm.

The complexity arises from the fact that we need to ascertain that any alterations in the loyal and less loyal customers' loyalty over time, are a result of the matching process and not of other external or unknown parameters.

When a field has only null values however an extra step needs to be taken before substituting said values, which is an investigation on the reasons for the nullification of the column. A column that always contains nulls might be a result of incompatible or error prone code, faulty equipment such as faulty smart meters or naming mismatches between the different stages of the ingestion process (i.e. different column name in the data lake and the group database).

These contingencies should be taken care of before creating the raw dataset rather than be later remedied with the methods discussed in section 4. However given the unpredictable nature of these events the modification or filtering of these fields will most likely be conducted during cleaning. For example the 'updateDate' field that in the provided dataset had unexpected values could be the result of faulty equipment or ETL related inconsistencies. Preferably the discovery and modification of the root cause of this discrepancy would be handled before the raw dataset is created. Realistically, this scenario is less probable but the field can easily be modified using the columns 'delDay' and 'fileId' as already demonstrated, provided that the column is not completely null.

This method of modifying a field using other fields does not come without its dangers.Caution must be taken to avoid replacing values using other columns, without first validating the quality of the input columns. In other words, it is essential that the modified and newly created columns are produced using only trusted fields as inputs. This can be achieved by a number of ways such as data versioning, exception handling and frequently communicating with the responsible database maintenance professionals.

Data versioning is the process of creating time dependent versions of a specific column or an entire dataset. A versioned column which has been repeatedly checked and passed relevant quality tests can be trusted to produce the contents of other columns, should the need arise.

Data quality tests can also be set as flags inside the energy matching algorithm. This type of quality checking belongs in the category of error handling, since an error will be raised or handled if the flags are raised. This way the algorithm will not initialize and will lead to an error, if the raw data provided do not meet specific quality standards.

This process will run before data cleaning commences, producing a better 'raw' dataset to be utilized by the ETL pipeline.Therefore a column that has passed both the error handling and the cleaning data quality checks over a number of different versions can be trusted to produce other column's contents.

Finally communicating with the relevant professionals is key when trying to identify the trusted columns in the dataset, especially when the columns are vast in number and interlinked. A maintenance IT professional is responsible for updating, monitoring, versioning and modifying data in a database. For this reason such an individual has considerable knowledge regarding initial field quality and current field status at a specific date and time.

Even if a field has been validated by a data engineering or data science team multiple times in the past the input of a data maintenance IT professional should still be sought after. The reason for this is that data ingestion or extraction processes might eventually break at the company or group level. This could potentially lead to a breakdown in data quality even for fields that had been previously considered trustworthy.

An alternative process that is recommended in case something does indeed break at a higher level is to use additional data ingestion processes implemented by a third party cloud computing provider. While automated third party ingestion solutions are somewhat simplified and could not replace complex group processes they could provide a temporary solution for any data ingestion or extraction needs of a pilot data science project, thus maintaining the already established reliability of specific data columns.

Before highlighting the different stages of the cleaning stage we caution the reader on the necessity of documenting table field contents. When a group of projects tend to be interlinked via specific fields or tables the problem of misnaming or not providing fully comprehensible names to fields arises. Either by using different naming conventions, using the same words to express different concepts or using vocabulary that is incomprehensible to specific groups interested in the project, one can unintentionally create confusion among colleagues.

In such a scenario different developers, analysts or managers may interpret these fields in a plethora of ways and therefore have a vastly different understanding of the current state of a project as well as the project's potential outcomes.
It is therefore easy to see that, in order to avoid unintentionally misleading one's colleagues it is paramount to provide meticulous documentation for fields used and/or created. The specific cases in which such a need arose in this project will be thoroughly explained later on.

At this point we present the steps taken during the cleaning stage of the algorithm. Cleaning is subdivided in four categories:

- Initial preprocessing methods
- Data quality related methods
- Null handling related methods
- Findings during preprocessing that require further investigation

These steps were followed for each cleaning script implemented. These scripts are executed as AWS Glue jobs and therefore are independent of each other. Additionally all methods pertaining to the cleaning process of a specific data table are contained within the corresponding script. Finally, because of the nature of glue jobs these scripts do not import any other modules from inside the codebase, utilizing only packages available to the underlying ec2 instance where the glue job is executed.

The reason for this implementation decision is twofold. Firstly, by creating the cleaning processes as glue jobs we diminish any form of dependency between them, making it easier to maintain and debug them in the future. Secondly, because of the vast amounts of data available to the group, it is expected that big data frameworks such as AWS Glue and AWS EMR will become increasingly more necessary in the future. This current and future need makes it essential to implement these data engineering pipelines with the aforementioned tools early on, in order to accelerate their adoption from the group.

There are four cleaning jobs, one for each available dataset. These jobs are executed in the following order:

1. Production cleaning job
2. Consumption cleaning job
3. Invoice cleaning job
4. Customer specs cleaning job

At this point the cleaning jobs for the invoice and customer specs data table are identical, in that the methods implemented are the same. However because of the future use of additional fields from both tables in the energy matching process, each cleaning script will later on contain additional different methods. For that reason two separate jobs should be present in glue, which will eventually diverge in content, with the use of more fields from each table in the matching process.

The execution order presented above is based upon the execution time of each script. Specifically it is observed that the production cleaning job takes the longest to complete, while the customer specs cleaning job takes the shortest. The reason that the invoice cleaning job takes longer to complete than the customer specs cleaning job is based on the fact that the number of records present in the invoice table is far greater than the one in the customer specs table. Therefore even if the methods in both scripts are at this time the same, the time of execution differs as a result of the disparity in the number of records in each table.

Below we document the specific cleaning methods implemented in each job.

4 Data Cleaning Process

4.1 Production Cleaning Job

The first thing that was observed while cleaning the production data is that the column names do not follow the required naming convention. The variables and columns utilized in development must follow the naming rules shown below:

- All variable or column names should use exclusively lowercase letters.
- Words in each variable or column name should be connected with the character '_'
- Every variable or column name must use at most four words.
- Every variable or column name must use at least two words.

Therefore the first step before any data cleaning occurs, is to rename the production table columns, in such a way that the above naming requirements are met. Even though the process of column renaming generally belongs in the curation stage of the ETL process, code readability should be ensured, before any other method is implemented.

One key feature of the energy sector that considerably perplexes ETL operations is the validation process. Every time energy is produced by a company, it needs to be measured and validated by the independent party HEDNO. This validation task is conducted multiple times with the later measurements being considered more reliable. As already stated, the production table holds a field called 'fileId' representing the date, at which a validation measurement took place.

The possible validation dates are one day later than the initial measurement, one week later or seven weeks later.The second method in the production cleaning job uses this field to filter out records whose validation measurements differ considerably from relating validation measurements. This method is broken down in four steps:

- The difference between adjacent validation measurements will be computed.
- The proximity between the day later measurement and the seven weeks later measurement will be checked against a threshold.
- The table will be filtered based on the proximity between the day later measurement and the week later measurement, provided that the previous check was successful.

As already stated in case the check between the day later measurement and the week later measurement fails, day ahead records can additionally be compared with the seven weeks later measurement. This extra check should only be considered if there is a lack of quality data. In case the volume of data passing the first check is acceptable no further comparisons should be performed in order to augment the cleaned dataset.

Admittedly day later records with less than 2% percent difference from seven weeks later records should not be completely discarded if that same data has failed the check with the one week later measurement. It is recommended that such records be kept as insights in order to investigate the reasons why the first check failed while the second succeeded.

The most probable cause for a measurement failing both checks is faulty energy meters. Therefore in case the day ahead measurement and the week ahead measurements for a specific meter frequently diverge beyond a predefined threshold, then the measurements from said meter should not be taken into account until the meter has been fixed.

However when the first check does not pass while the second does, the cause or the remedy to this contingency is not so clear. For this reason it is recommended that these records be kept as insights rather than be discarded. It is important to remind the reader at this point that the second validation measurement is considered more reliable than the first.

For this reason if the second check passes, it indicates the reliability of the initial validation measurement, even if the first test fails. Also it needs to be noted that for the energy transactions to be completed swiftly the validation process should be concluded as early as possible. For this reason the records which pass the first validation check are preferred than the ones only passing the second. However, before the required transaction swiftness becomes possible two conditions have to be met:

- It needs to be shown that the difference between the one week later measurement and the seven weeks later measurement is under a predefined threshold. This will allow us to act with the same level of confidence when the validation of energy records is completed using only the first check. If such a requirement is not met then investigations need to be conducted to understand and resolve the disparity between the two checks.
- After the above similarity has been established then the difference between the day ahead and week ahead validation measurements needs to be shown as almost equal. This way the validation process will be achieved at a daily basis using only the day ahead validation measurement in comparison with the initial energy measurement.

At this stage it has been empirically shown for our small dataset that the difference between the one week ahead measurement and the seven weeks ahead measurement is on average under the predefined threshold. However, such a test should be conducted at a higher level (using a greater volume of data from the group database), before concluding that this is generally the case.

Having shown the similarity between the one week and seven weeks ahead measurements, two specific data quality methods have been implemented:

- Validation measurement disparity methods
- Date disparity methods

As already discussed our energy production data is validated in different intervals. Apart from the validation of the energy itself it has been observed that the difference between the date fields also needs to be validated. In other words the relationship between the date related fields "delDay", "delHour", "delQrt", "FileId", "UpdateDate" does not always hold. The required relationship is:

$updateDate = delDay : delHour : delQrt + FileId$

The symbol ":" expresses concatenation of two datetime objects using the colon as a delimiter. The addition above between a string object and a datetime object represents the addition of a day to a datetime object. This simplified addition can be implemented in spark sql with the $date_add$ function. This relationship should hold as, the datetime represented by the expression $delDay : delHour : delQrt$ is the datetime at which the initial measurement took place in the format "Day-Month-Year:Hour:Quarter". When the days after the initial measurement, are added to the above datetime object then the final datetime object should represent the date and time at which the validation measurement took place which by definiton is *updateDate*. Additionally any records that fail to meet this requirement should be filtered out and added as insights to be examined later.

The process above is about filtering out records, based upon the deviations of the field volume in different measurement verification dates. These validation dates must correspond to the same initial measurement, indicated by the fields ('delDay', 'delHour', 'delQrt'). Another method for dropping records revolves around the specific value of the volume field. Specifically in case the field is below an arbitrarily predefined threshold the corresponding record will be dropped. This threshold is based upon the value that the energy consumption is expected to be over, provided that that meter utilized does not provide false readings.

In other words in case the value of the field volume is under a specific threshold which corresponds to the expected minimum amount of energy a meter could log, then the corresponding row will be dropped. The reason is that these records are likely produced by faulty meters. Therefore the specific value of these meters is not known and for that reason their records should be dropped during this phase of preprocessing.

After the records have been filtered out, so that only the ones with small volume deviation between the validation measurement dates are present, we are focused on the validity of the field 'UpdateDate'. This field as already explained expresses the date at which the validation took place. This field is expected to be in compliance with the fields 'delDay' and 'fileId'. Specifically, the field needs to be equal to the the value of 'delDay' plus the number of days indicated by the field 'fileId'.

For example in case the field 'fileId' has the value 'D' then the field 'updateDate' needs to be one day later than the field 'delDay. If the field 'fileId' has the value 'W+1' the field 'updateDate' has to be one week later than the field 'delDay' which is equivalent to 7 days. Finally if the field 'fileId' is equal to 'W+7' then the field 'updateDate' has to be seven weeks later than the field 'delDay' which is equal to 49 days.

However what has become apparent during preprocessing is that the field 'updateDate' is not always equal to the expected value. Based upon the presupposition that the integrity of the field 'delDay' is not in question then the only reasonable deduction is that the values of the field 'updateDate' are faulty. This can be remedied in two ways, either by modiying the field or by dropping it.

Provided that the reason for the alteration from the expected value is not known at this point it was decided that it would be best to drop the rows where the field 'updateDate' is not as expected, instead of changing them. Even though these records should not be used later on during the matching process, it is advised that the records are kept in another dataset so that an investigation on the reasons these alterations between from the expected values of 'updateDate' occurred.

The final method in the preprocessing pipeline is the modification of the 'delDay' field in case of nulls. The field 'delDay' is generaly considered trustworthy in the sense that the data contained in it are expected to be correct. However in a small percentage of records found in the production table this field contained null values.

For this reason a method was implemented in order to modify the field, when it contains nulls. The modification process is similar to the one used above in order to drop records with unexpected 'updateDate' values. The only major difference is that instead of using the fields 'delDay' and 'fileId' to detect unexpected values in 'updateDate' we use the fields 'updateDate' and 'fileId' to modify the nulls present in the field 'delDay'.

As already mentioned the field 'fielId' can take three possible values 'D' meaning that the verification process occurred one day after the initial measurement, 'W+1' meaning that the verification process occurred one week after the initial measurement and 'W+7' meaning that the verification process occurred seven weeks after the initial measurement. Therefore, for each distinct value of the field ’delDay’ three different records should be present for each different value of ’fielId’. This however is not the case for all records. It was observed that there were measurements for which the day after validation or next week validation were missing.

At this time the preprocessing phase does not include a method for dropping records that do not hold all three possible values of ’fileiD’ for a specific value of ’delDay’. The reason for this decision was based on the fact that the number of records for which all three ’fielId’ values were present was considerably small making it challenging to perform an energy matching process with the remaking data. However it is advised that an investigation should be conducted to understand the reasons why so few records have all three possible verification measurements logged.

It could become apparent through this process, that the verification measurements in specific areas are not conducted properly or punctually. Alternatively it could be the case that the reason for the lack of records for different verification measurement dates is the result of poor logging rather than the lack of measurements. In any case however, in order for the matching process to be conducted properly the reason for this disparity in data should be discovered and remedied.

Another feature that should be added in the preprocessing phase of the matching algorithm is the verification of record uniqueness for each specific date and time. In other words it must be certified that for each specific pair of values in the field ’delDay’, ’delHour’, ’delQrt’ there is only one record present in the table. In this relatively small dataset it has become apparent that the above requirement is not satisfied. This means that there have been observed multiple records for each unique triplet of values in the aforementioned fields.

This was tackled in this case by finding the mean value of these records before aggregating the consumption data for each month. However such an approach ignores the underlying problem of having multiple records where only one was expected.

Therefore it is recommended that an investigation is conducted in order to ascertain the reasons for the multitude of records for each specific pair of values from the fields ’delDay’, ’delHour’, ’delQrt’.If an average of the records that belong to the same triplet of date and time values had not been computed, these values would be wrongly added together during the aggregation stage of the ETL process, which is discussed below.

4.2 Consumption Cleaning Job

The next part of preprocessing is the cleaning process of the consumption table. This table only has two values and therefore the execution time required for this cleaning script is considerably less than the execution time for the production table. Specifically the methods implemented in the consumption cleaning script are the following:

1. Merge consumption tables.
2. Rename consumption columns.
3. Transform power to energy consumption.
4. Filter out records with adjacent frequency greater than one second.
5. Filter out records with adjacent frequency less than one second.
6. Filter out records with zero or almost zero energy consumption values.
7. Round energy consumption to two decimals.

Even though the number of methods is greater than the one for the production cleaning process, the implementation itself is more concise. This in turn leads to smaller memory requirements for the storage of this particular script as well as less time required for execution.

First of all, we merge all consumption tables provided to us by the group. Each one of these tables corresponds to a different consumer the address of which is indicated by the file name. At this stage during the implementation two possible approaches where considered. Either clean each specific table sequentially or merge them together and apply the cleaning process once.

In terms of execution time, the second option is preferable, since it does not require to iterate over each file and therefore sequentially call the python interpreter, unlike the first approach. Additionally the first approach is more IO intensive, since it requires to read all 15 consumption files. However with the second approach, only one file is read, which contains the consumption records of all customers.

For the preferred approach however to minimize the calls to the python interpreter and therefore the execution time of the script, the utilization of hadoop is required. Specifically, we choose to store the raw consumption files in s3 which is build on top of apache hadoop and apache hive. Apache hadoop is a data warehouse solution for storing big data, provided by the Apache Software Foundation. It allows us the ability to partition a table by creating a directory that specifies the value of a specific field.

In other words by creating folders with a name in the format 'table_field_name=field_value' we instruct hadoop to partition the table in such a way that each subtable only contains this specific value on that specific field. This can of course work inversely as well. If we have multiple files that only differ in the value of a specific field and want to instruct hadoop to merge them together as one file, we only have to use the aforementioned format to aggregate all tables into one.

Finally it needs to be said that, this field does not need to be actually present in the tables, for the merging process to succeed. In case the field specified in the directory name does not exist, hadoop will include it in the final table regardless of its existence in the subtables.

This last feature is of particular use since it allows to avoid iterating over all files in order to append the field 'unit_name' in each table and overwriting it. Instead we simply create the directories inside s3 in the format 'unit_name=¡customer_address¿' and the merged table will include this field with the required value. In order to actually load the merged table, the direct parent s3 bucket before the partitioning of the field, needs to be provided as an argument, in the 'read_table' method implemented in the production cleaning script. The value selected for each directory is indicated by the file name, which specifies the address of the corresponding client, as already stated. At this point only 15 files are provided by the group, therefore the creation of each directory inside s3 can be done manually. However when the volume of files increases it is advised that a bash script is implemented with commands from the aws command line interface, in order to automate the process of creating directories for each customer address available.

After merging all raw consumption tables together, the two fields present in the table are further processed. First the tables are renamed in order to follow the column naming conventions described above. Specifically the two fields ”BuildConsumption” and ”Timestamp” are renamed to ”build_consumption” and ”timestamp” respectively. Therefore with the addition of the field ”unit_name” in the merged consumption table the total number of fields adds up to three, all following the required naming conventions.

As already stated the process of renaming table fields generally belongs to the curation part of the ETL process. However in order to meet the criteria presented above that the names of fields and variables must follow the renaming process had to be applied first.

After the consumption tables have been merged and it's fields have been renamed the actual cleaning of the table takes place. First of all, the values in the field 'build_consumption' need to be converted from power to energy. This is achieved by simply multiplying the power values with the number of hours in which they were consumed. In this case, this is equivalent to multiplying the power consumption with the scalar value 1/60, since power consumption is logged every minute. The realization that the values contained in the field are measured in Watts instead of Watt hours was the result of investigating undocumented code for a another project that dealt with power values.

The outputs of that project are the inputs used in the energy matching process, making the energy matching project dependent on the outputs generated by that previous project. However the process of documenting each field's contents and units of measurement when applicable, have not been completed in the previous project. Additionally the method for converting the power values which were rightly stored in the table of the previous project to the energy values that should be stored in the consumption table has not been implemented or applied.

This event is mentioned in order to highlight the importance of field documentation and task driven development. In a project that requires such attention to detail, the process of investigating the intricacies of previous projects becomes challenging. Therefore, it is of the utmost importance to document all the fields utilized by a developer regardless of their future use in this or any other project.

Moreover, the task of documenting the expected contents of the fields utilized in any new project should be completed by project experts that are or were at some point in the past, employed by the group. This way it becomes clearer what should be the expected input to the ETL process and what is the expected output.

In other words only when the field's that are utilized and generated by the ETL process, are documented before the project begins, does the development, maintenance and refinement of said process become possible. In case the fields have not been completely documented, the possibility of having to study the intricacies of project dependencies arises. This leads to an increase in development time and leads to an overall decrease in company productivity.

In each consumption dataset the data are read and posted by smart meters every one minute. However there are records present in the dataset which have a time difference from the previous record over than one minute. Similarly but less frequently there are records which have a time difference from the previous record of less than one minute. To fix this discrepancy, any records that have an unexpected time difference from the adjacent previous or future records are dropped from the dataset. This measure should be further modified by providing a confidence interval regarding the time difference between adjacent records.

Specifically it could be argued that records should only be dropped if their time difference from an adjacent record is outside of a computed confidence interval. This interval should center around one minute, which is the expected time difference and expand based on the expected time difference distribution. Given that the time difference follow a Gaussian distribution the formula for computing a 95% confidence interval for the time difference is given below:

$$(\overline{td} - \widehat{s}\ z_{0.95},\ \overline{td} + \widehat{s}\ z_{0.95})$$

Where $\overline{td}$ is the computed mean time difference between two adjacent records, $\widehat{s}$ is the computed standard deviation and $z_{0.95}$ is the z score for the 95th percentile point. Given the fact that the computed mean time difference should be one minute and that $z_{0.95} = 1.96$ the above formula is written as:

$$(1 - 1.96\ \ \widehat{s},\ 1 + 1.96\ \ \widehat{s})$$

Of course the values of the mean and standard deviation need to be computed before the confidence interval is calculated. In case the value of the mean time difference deviates considerably from the 1 minute value, then the reasons for that deviation need to be discovered. The most probable cause for the mean time difference to be substantially dissimilar to one minute is faulty equipment. Therefore the consumption tables for which the calculated mean time difference is not equal to the expected one minute value, should be excluded from the analysis until the reasons for the deviation are found and the deviation itself is fixed.

Finally by setting the mean and standard deviation to their computed values, the confidence interval for the time difference between adjacent records is calculated. Using this interval we can certify whether or not a time difference between consumption records is expected or should be filtered out of the table and investigated further.

Apart from time difference considerations, records should also be filtered out in case the consumption itself is unexpectedly low. In case a customer's energy consumption is zero or close to zero it is probably the result of a faulty measurement. Faulty measurements in turn could be the result of a number of reasons such as record falsification by the customer or faulty equipment. Given that in this dataset the consumption values were produced by smart energy meters, it is more likely for a measurement to be affected by the equipment used rather than by any falsification attempts. In any case the records that exhibit considerably low consumption values should be dropped.

Additionally if consumption from a specific meter is consistently low this meters records should be completely dropped, and the causes of these observations should be investigated. Until a conclusion is reached, on the reasons why the consumption values of a specific meter are zero or almost zero, this meter should not be utilized in the energy matching process.

Finally after all the values have been dropped or properly modified the values need to be rounded to two decimal points. This final requirement is provided by the company for reasons of presentation. Specifically, consumption values and energy matching results will be included in a customer specific dashboard. This dashboard will include results from the majority of company projects that directly affect or interest a customer.

Some of the things that could be included in the dashboard are the customer's current monthly consumption, a forecast of the total monthly consumption, the corresponding energy invoice, the percentage of energy consumed using green energy, the percentage of energy consumed that was provided by the energy matching project. As it is obvious by the number of aforementioned values, the dashboard will inevitably contain a considerable number of scalar values which is expected to grow with the number of provided services. Therefore, in order to present a comprehensible and visually appealing dashboard, it was decided by the company that all scalar values included, should contain two decimal points.

4.3 Customer Specifications Cleaning Job

The cleaning process for the invoice and customer specification tables is at this point the same. This process contains three methods that are related to the documentation table.This means that all cleaning tasks for the ranking related tables take input from the corresponding documentation table.

The methods implemented in the customer specs cleaning job are:

- Find unexplained fields
- Find null fields
- Find duplicate fields

These tasks are thoroughly explained in subsection 3.2.2. This cleaning job has been implemented twice, once using aws glue and once locally. The distinction between the two approaches lies on the use of the aws glue data catalog, which is similar to the Apache Hive Metastore and therefore can be parsed using HiveQL. This feature provides metadata support without needing to create an additional table.

Using a graphical user interface, it is possible to add additional information about the columns of a specific table, such as column dtype and description. These information called metadata can then be read inside a glue job using HiveQL to further process these columns and clean the corresponding table. HiveQL is a query language used with Apache Hive, a data warehousing system built on top of Apache Hadoop.

The use of a cloud stored easily accessed and easily modified data catalog is preferable to a data documentation table. However such a catalog is expected to grow considerably over time, making it impractical in scenarios at which the total available memory is limited. However when the customer specifications table needs to be cleaned inside an edge device, it is advisable to use a succinct data documentation table instead. Therefore it is recommended, that both of these implementations should be utilized in different stages of the ETL process.

5 Data Curation Process

5.1 Aggregation Process Considerations

The general premise of this project is to match energy production to energy consumption for a given date and time. Before this matching occurs however, a number of decisions need to be taken beginning with some aggregation related challenges.

Specifically, before any matching process takes place, there needs to be an agreed upon time period to which both energy consumption and production is aggregated at. This period will depend on a plethora of factors such as the volume of available data, the project scope, the country regulations and the guidelines given by the company.

- Data volume:

  In case the data volume is considerable the aggregation period should be maximized in order to satisfy any memory constraints presented by the hardware specifications of the production server. In other words in case the raw data used in the algorithm, is expected to be greater than the available memory on the production server, then the aggregation period needs to be properly increased in order for the clean dataset to be stored in the production environment.

  Otherwise if the data is less than expected then the aggregation time period needs to be decreased in order for the final matching table to satisfy a least amount of records threshold given by the company. Therefore, the aggregation period should be a dynamic parameter based upon the available raw data given, each time the algorithm will be expected to run.

  While in most cases the latter is considered more likely than the former, in case the production server perpetually presents an inability to store the available clean dataset, then it is suggested that a bigger production environment will be provisioned. Additionally, given the company's use of cloud technologies for the implementation and deployment of the algorithm a pay as you go approach could also be considered instead of provisioning servers of specific specifications.

  This way the transition between servers can be avoided, something that at this point is expected to occur in the near future given the increased rate of insertions in the company database.

- Project scope:

  The amount of data provided for the implementation of the matching algorithm is indicative of the amount of data that will be given to the algorithm in the production environment. This volume along with other resources provided is dependent on the project scope. By project scope we define the percentage of customer data from the database that will be used for the implementation of the algorithm and subsequently the amount of customer's for which a first deliverable shall be deployed.
  Based upon this definition the scope of a project belongs to one of the following categories:

  - Pilot: A project whose funding is relatively small and no actual deliverable has yet been produced. Projects of this type use data, mostly from the group built village in aspra spitia.
  - Company level: A project which has produced a deliverable that can be commercialized and is used by the specific company that implemented it. Projects of this type have greater access of data sources that reside in the company and group database.
  - Group Level: A project which has been at production level for a considerable amount of time and has produced significant revenue to the company which implemented it. Projects of this type have access to the majority of records in the group database after the appropriate ticket has been submitted and resolved by IT.

- Country regulations: While at this point the project is on the pilot scope, it is the company's goal that the deliverable will in the future be utilized by the company's worldwide clientele. For this reason any differences in regulation regarding the amount of data used when providing a green energy matching service should be taken into account and modulate the aggregation time period accordingly to meet said regulations.

- Company guidelines: As already stated the company envisions that this algorithm will be deployed for the worldwide clientele of the group. Currently though the project is considered to be on the pilot scope. Therefore as the algorithm becomes more refined and profitable, the aggregation time period needs to be decreased so that more transactions will be possible in a given time frame and therefore more customers can profit from the algorithm's features.

After taking into account the above parameters the aggregation period has been set at the monthly level. This means that the a key step in the curation job is to sum all production and consumption values for each month of the year. This aggregation process only takes into account the month field of the corresponding table, that is a transformed ’delDay’ field for the production table and a transformed ’Timestamp’ field for the consumption table. It does not however take into account the number of records existing for each month.

This implementation feature could potentially lead to a discrepancy in the aggregated values between different months, in case of missing values in specific months. For this reason it is recommended that a process is implemented to filter out entire months where a considerable percentage of expected monthly data is missing. This way the volume of data for the months remaining will be representative of the real monthly consumption and production volumes.

5.2 Invalid Aggregated Months

After all the data quality methods have been applied, the data are aggregated according to a specific time frequency as already stated above. In order for the aggregation step to provide representative energy data it is recommended that all time periods that contain less than expected volumes be filtered out.

The expected number of energy volumes differs for the production and consumption datasets. The reason for this difference lies on the raw data record frequency. By raw data record frequency we define the datetime difference between two adjacent records in an ordered raw dataset. By raw dataset we define a dataset to which no processing methods have yet been applied.

This datetime difference expresses the time interval between the logging of two consecutive energy measurements (production or consumption). Specifically the consumption raw data frequency is one record every minute, while the production raw data frequency is one record every day.

The formula providing an upper bound to the amount of monthly records is given below:

$$\begin{array}{l} number\_of\_monthly\_records \leq \\ raw\_data\_record\_frequency \quad frequency\_denominator\_per\_month \end{array}$$

By $frequency\_denominator\_per\_month$ we define the number of time intervals ,present in the denominator of the $raw\_data\_record\_frequency$, that exist in a month. This formula is basically the maximum amount of values that are expected based on the default record frequency and the amount of values of that frequency that exist in a month.

For example the $frequency\_denominator\_per\_month$ in our consumption dataset is the number of minutes in a month and the $raw\_data\_record\_frequency$ is once a minute. This means that we expect to have at most 43.200 consumption records in a month. This a result of the fact that there are records once a minute in the raw dataset and there are 43.200 minutes in a month.

Likewise in our production dataset $frequency\_denominator\_per\_month$ is the number of days in a month and $raw\_data\_record\_frequency$ is once a day. Therefore we expect to have at most 60 consumption records in a month. If the volume threshold is set to be at least 80% of the maximum volume then any months that have less than 34.560 consumption records or 48 production records will be filtered out.

5.3 Mapping Data Curation

After all relevant decisions have been made and all undesired records have been dropped we move to the actual curation job for the mapping process. This job was implemented exclusively on glue but there could be a need for a local implementation, in case the curation task needs to be executed in an edge device such as a battery. In general the curation process includes tasks such as column renaming, table joining, record aggregation and metadata logging.

Two of these tasks, namely column renaming and metadata logging have already being completed in order to clean the available datasets. The next step is to aggregate records using the parameters discussed in subsection 5.1. Finally the two tables that are utilized during the mapping process are joined together to create the mapping curation table.

The methods implemented in the mapping data curation job are:

1. Extract date and time features from consumption column 'timestamp'.
2. Aggregate consumption by quarter.
3. Aggregate consumption by month.
4. Aggregate production by month
5. Join production and consumption tables

First we note that all tables used in the curation job have first been cleaned. Both the cleaning jobs and the curation job belong to a specific ETL pipeline. By pipeline we define an ordered sequence of ETL scripts each of which takes as input the output of the previously run script. In the case of the mapping process pipeline, the outputs of the cleaning jobs are provided as inputs to the curation job. Therefore even though it is not mentioned above all cleaned tables first have to be read.

As shown in table 4, the raw consumption table has a column called "Timestamp". This column has been properly renamed to "timestamp" during the consumption cleaning process. However this field despite what it's name would suggest contains both the date and time at which the energy was consumed. In order for the curation process to be completed, these two values need to be extracted and stored in separate columns.

We note at this point that the reason for not renaming the column to something more appropriate in the consumption cleaning job was for reasons of compatibility with other projects using this exact cleaned consumption field. In case we decided to rename the field in a way that would indicate that it contains date and time values, implementations from other projects should also be modified in order to read the contents of the cleaned consumption table, which could prove time consuming.

After the new columns have been created in the consumption table the aggregation process begins. As stated in subsection 5.1 the aggregation period selected at this point is one month. However for the matching process to function as intended this aggregation step needs to gradually be reduced until it ideally reaches one minute. The criteria that need to be met, for the aggregation level to be reduced are thoroughly explained in subsections 4.1 and 5.1.

For this reason an additional method is implemented, to aggregate the consumption data at the quarter level. This method will not be currently utilized in a production setting. However the viability of reducing the aggregation level to one week is tested, in order to provide insights and estimations regarding the expected date at which such a reduction will be possible. Additionally a list of suggested actions are provided to the company in order to accelerate the transition period until a smaller aggregation level can be utilized in a production setting. We also remind the reader that the production data are already aggregated for every quarter of the hour and therefore no corresponding method is needed in case this aggregation period is chosen.

The method that will actually be used in the etl pipeline is the 'aggregate consumption by month' method. This method follows the extraction of date and time values from the 'timestamp' column. Specifically in the method's implementation the year and month of consumption are also extracted.

Consequently the consumption data are grouped and aggregated by the year of consumption, month of consumption and meter address. Finally after the aggregation is complete, the consumption records are summed and rounded to two decimal points so that they meet the requirements given by the company.

The aggregation process for the production table is almost identical with the aggregation process described above.The only major difference is in the use of the production plant address instead of the customer's meter address in the group by clause.

Finally after the consumption and production data have been aggregated to the agreed upon time period, the two tables are joined. This task is the first step in matching process between production and consumption. Specifically, before the use of the aforementioned metrics to determine the amount of green energy provided to each customer, an overall matching is implemented. At this stage the only requirement is for the time periods when the energy was produced and consumed to be equal. For all pairs of records between the two tables that meet this criterion, a potential energy transaction is created.

In other words we create a first energy matching draft by joining the two tables based on the month and year of consumption and production. This implementation step provides a number of insights regarding the number of customers that require energy as well as the amount of energy required at each point of date and time. The final table contains only the month and year of consumption and production, the total monthly production, the customer's meter address and the production site's address.

The reason the total monthly consumption is not defined is because it will not be of use in the next step. One thing to notice at this point is that this matching process requires knowledge of the energy consumption at the specified date and time.

This means that the energy will already be consumed by the time the matching process has completed. Therefore one could argue that the goal of providing green energy to customers in order to increase the net zero metric is not actually achieved since it could well be the case that the actual energy consumed might be produced by fossil fuels.

Although the final goal of the project is indeed to provide energy to customers before this energy is consumed this requirement also demands the implementation of production and consumption forecasting algorithms. While this algorithms are currently being developed by the company they are beyond the scope of this thesis.

6 Energy Matching

With the completion of the ETL process, we move on to the energy matching algorithm. This algorithm utilizes two tables that are produced during the ETL process, namely the cleaned customer specification table and the curated mapping table. Provided that the invoices table will not be used at this point there was no need to implement a curation algorithm for the ranking process. Below we list the steps implemented as part of the energy matching task.

1. Transform customer addresses to the appropriate format.
2. Include customer specification fields to the mapping curation table.
3. Compute the client loyalty score.
4. Compute the green energy to be provided to each customer.
5. Round green energy values to two decimal points.

Up until this point only an initial energy matching table has been produced. That table contained all potential transactions for each specific month and year in the consumption and production tables. Using the steps mentioned above that table will be further refined to pinpoint the specific amount of energy given to each customer. In general the aggregated production for each month is expected to be considerably greater than the corresponding consumption of each client. Therefore it is expected that all customers will be given a percentage of the total aggregated energy based on their loyalty score as discussed in section 3.2.

The first step in the energy matching process is to include the necessary fields from the specification table to the mapping curation table. This joining process requires a specific criterion to be met. For all pairs of records of these two tables that match this condition the specified fields are concatenated and a new record is created. In this case the criterion for joining the curated mapping table and the cleaned customer specification table is that their corresponding address fields are equal.

In other words we join records from these two tables when the customer specification field containing the customer's address and the mapping field contain the consumption address are equal. Even though these two fields express the same quantity the format used in each one is different. Specifically, the customer address in the customer specs table is written in Greek and in full.

However the consumption address in the mapping table is abbreviated, using only the first two letters of the address written in English. Additionally the address number is concatenated, creating a four character encoding of the complete address, two letters and two numbers. Therefore before the joining process can be achieved, the customer addresses in the customer specification table are modified to match the format used in curated mapping table.

After the address field in the customer specs table has been properly modified, the two tables can be joined. As explained in 3.2.1 there is a plethora of fields that could be used in order to create a loyalty score system for the company's clientele. However, at this point only the field "MonthsInContract" is used. Therefore the output table of the join process, contains all fields from the curated mapping table as well as the field "MonthsInContract" from the customer specs table.

This table contains all the fields necessary to calculate the client loyalty score and subsequently the green energy given to each client. In practice the client loyalty score is at this point the months a client has been in contract with the company. However this value needs to be modified in a way that allows for the scores to be compared with each other and at the same time. Additionally this modification should facilitate a way to attribute green energy proportionally based on the score comparisons previously made.

For this reason the "MonthsInContract" field is divided by the sum of all its values. This way all the resulting values will be at most than one and at least zero, providing an easier interpretation of its values as loyalty scores while simultaneously making it easier to compare them with each other. In terms of implementation, this task can be divided to two parts.

First a new column must be created that has the same exact value for all records. The value contained in this field is the sum of all values from the "Months InContract" field. For this to be achieved the customer specs mapping table created above, needs to be cross joined with that value. This can be though of like cross joining a one record table with the table created above.

In contrast with the other join operation described already, which was an inner join, cross join does not require a specific condition to be met for specific records of the two tables to be joined. Rather, this join operation concatenates all records of the two tables with each other, which is precisely why it is chosen at this particular point. After the customer specs mapping table has been joined with the one record table, the months in contract field is divided by the sum of its values to provide the loyalty score.

As discussed above the loyalty score is designed to be between one and zero. Apart from making the comparisons between the different scores easier, this decision provides a convenient interpretation of the scores. That is, these scores can be easily thought of as percentages of the green energy that should be supplied to each customer. This way it naturally follows that in order to get the actual energy provided to each customer, the last step is to simply multiply every loyalty score with the total energy produced at each particular month and year available. At this point we remind the reader that the initials fields of the customer specs mapping table are:

- The year the energy was produced.
- The month the energy was produced.
- The address of the production site.
- The customer's address.
- The total energy produced at that month and year.

Therefore the last step to compute the green energy that is provided to each customer is to multiply the percentage like loyalty scores, with the total energy production of a particular year and

del_year	del_month	consumption_building	production_building	client_green_energy
2022	12	om06	AISWNIAS_16443	207.04
2022	9	om06	AISWNIAS_16443	463.08
2022	12	ap55	AISWNIAS_16443	207.04
2022	12	th20	AISWNIAS_16443	207.04
2022	12	pn10	AISWNIAS_16443	207.04

Table 8: Energy Matching Final Table

month. After rounding the green energy provided to two decimal points the process matching is complete. All rounding operations conducted during the implementation of this algorithm were bround operations. This means that all energy was rounded the half even rounding technique. The reason for this decision was in order not to falsely claim excess energy production or false advertise excess green energy distribution.

Below is a fragment of the final table containing the green energy provided to each customer. The columns above contain the following values:

- del_year: The year the energy was produced.
- del_month: The month the energy was produced.
- consumption_building: The customer's address where energy was consumed.
- production_building: The address of the production site where the energy was produced.
- client_green_energy: The green energy provided to each customer based on their respective loyalty score.

We note at this stage that the date the energy was produced is equal to the date energy production was initially measured. Therefore the above descriptions are in agreement with the column descriptions provided in 3.

7 Conclusion

Given the amount of records filtered out from the raw production dataset, the fact that only one production site was available and that the total number of client records provided was only 15, the number of records in the final energy matching table was predictably limited.

However with the use of all the insights collected from the dropped records, the resulting table of the energy matching process is expected to increase substantially. Additionally, by continuously implementing the suggestions provided in section 3. The energy matching process will be gradually refined in order to take into account all the important factors necessary to deduct which customers should be considered loyal to the company and to what extent should they be rewarded for it.

The implementation associated with this pilot project is a proof of concept for the fruition of a much larger loyalty reward system that could be adopted group wise. In other words with the continuous use of the big data technologies mentioned as well as the adoption of new technologies as seen appropriate this system, could potentially be provided to the entirety of the group's energy clientele. However before any such adoption can become a reality, the need for a well maintained and frequently tested ETL pipeline needs to be established in conjunction with frequent data versioning and documentation practises.

References

[1] https://education.nationalgeographic.org/resource/region/

[2] https://en.wikipedia.org/wiki/Geographic_regions_of_Greece

[3] https://en.wikipedia.org/wiki/Prefectures_of_Greece