KADİR HAS UNIVERSITY
SCHOOL OF GRADUATE STUDIES
PROGRAM OF COMPUTER ENGINEERING

APPLYING MACHINE LEARNING ALGORITHMS IN SALES PREDICTION

JUDI SEKBAN

MASTER'S THESIS

ISTANBUL, AUGUST, 2019

APPLYING MACHINE LEARNING ALGORITHMS IN SALES PREDICTION

JUDI SEKBAN

MASTER'S THESIS

Submitted to the School of Graduate Studies of Kadir Has University in partial fulfillment of the requirements for the degree of Master's in the Program of Computer Engineering

ISTANBUL, AUGUST, 2019

DECLARATION OF RESEARCH ETHICS

I, Judi Sekban, hereby declare that:

- this Master's Thesis/Project/PhD Thesis is my own original work and that due references have been appropriately provided on all supporting literature and resources;
- this Master's Thesis/Project/PhD Thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- I have followed "Kadir Has University Academic Ethics Principles" prepared in accordance with the "The Council of Higher Education's Ethical Conduct Principles".

In addition, I understand that any false claim in respect of this work will result in disciplinary action in accordance with University regulations.

Furthermore, both printed and electronic copies of my work will be kept in Kadir Has Information Center under the following condition as indicated below:
The full content of my thesis/project will be accessible only within the campus of Kadir Has University.

Judi Sekban

19/09/2019

KADIR HAS UNIVERSITY

SCHOOL OF GRADUATE STUDIES

ACCEPTANCE AND APPROVAL

This work entitled **APPLYİNG MACHİNE LEARNİNG ALGORİTHMS İN SALES PREDİCTİON** prepared by **JUDİ SEKBAN** has been judged to be successful at the defense exam held on **19.08.2019** and accepted by our jury as **MASTER'S THESIS**.

APPROVED BY:

Prof. Dr. Hasan Dağ (Advisor)	Kadir Has University	Hasandağ
Asst. Prof. Dr. Arif Selçuk Öğrenci	Kadir Has University	
Prof. Dr. Mustafa Bağrıyanık	ITÜ	

I certify that the above signatures belong to the faculty members named above.

Prof. Dr. Sinem Açıkmeşe
Dean of School of Graduate Studies:
DATE OF APPROVAL:

TABLE OF CONTENTS

APPLYING MACHINE LEARNING ALGORITHMS IN SALES PREDICTION

ABSTRACT

Machine learning has been a subject undergoing intense study across many different industries and fortunately, companies are becoming gradually more aware of the various machine learning approaches to solve their problems. However, in or- der to to fully harvest the potential of different machine learning models and to achieve efficient results, one needs to have a good understanding of the application of the models and of the nature of data. This thesis aims to investigate different approaches to obtain good results of the machine learning algorithms applied for a given prediction task. To this end the thesis proposes and implements a four different algorithms, a stacking ensemble technique, and a specific approach to feature selection to develop models. Using different configurations, the results are compared one against another. All of these are done after applying the necessary data prepossessing and feature engineering steps.

Keywords: Machine Learning, Prediction, Sales, Feature Selection, Feature Engineering

SATIŞ TAHMİNİ KONUSUNDA MAKİNE ÖĞRENİMİ UGULAMA

ÖZET

Makine öğrenimi bir çok endüstride üzerinde yoğun çalışmalar yapılan bir konu olmuştur, ve neyse ki şirketler kendi problemlerini çözebilecek çeşitli machine learning yaklaşımları hakkında günden güne daha fazla bilgi sahibi oluyorlar. Fakat, farklı makine öğreniminin modellerinden en iyi şekilde sonuç almak ve verimli sonuçlara ulaşabilmek için, modellerin uygulanış biçimlerini ve verinin doğasını iyi anlamak gerekir. Bu tez, belli bir tahmin görevi için, uygulanan farklı makine öğreniminin algoritmalarını ne kadar iyi sonuç verdiklerini araştırır. Bu amaçla tez, 4 faklı algoritma, bir istifleme topluluğu tekniği ve modeli geliştirmek için belirli bir özelllik seçme yaklaşımı sunar ve uygular. Farklı konfigürasyonlar uygulayarak sonuçlar birbiriyle test edilir. Bütün bu işlemler, gerekli veri önislemeleri ve özellik mühendisliği adımları tamamlandıktan sonra yapılır.

Anahtar Sözcükler: Makine Öğrenimi, Tahmin, Satışlar, Özellik Seçimi, Özellik Mühendisliği.

ACKNOWLEDGEMENTS

I would like express my sincere gratitude to Prof. Dr. Hasan Da˘g for his support, guidelines, very precious insights, and always expecting more from me throughout my thesis development process. I also want to thank the rest of my committee members: Dr. O¨g˘r. U¨yesi Arif Sel¸cuk O¨g˘renci and Prof. Dr. Mustafa Ba˘grıyanık.

DEDICATION

First and foremost, I thank Allah for letting me live to see this thesis through. Next, I dedicate this thesis to my parents (Dr. Faeyk and Eng. Nahda) who have given up their dream for me to pursue mine. To my very special and lovely husband, Ahmet, who has been supportive along way the journey from the very beginning of taking the first course in this master program. Not least of all, I owe so much to my whole family (my sisters Rama and Marwa and my youngest brother Mohammed) for their undying support, their unwavering belief that I can achieve so much.

LIST OF TABLES

Chapter 1: LIST OF FIGURES

Chapter 1: INTRODUCTION

1.1 Business Background

Sales play a significant role in any business and the tasks undertaken by the sales department have a lot to do with the business growth. That is because when cus- tomers buy a product or more that a particular business offers it basically indicates how much the customers trust these products and the business itself and how likely they are to recommend those products to people in their network. In addition to the fact that sales are the source of revenue of the business and thus they are what keeps that business alive. According to this, it is crucial to keep a track on the products information and how they relate to the sales numbers. Actually, there has been many organizational uses of sales data in corporate environments and that is why it has been a topic of research for decades now (Kenton and Lawrence, 1991). In fact, researches showed that companies perform better when the decision they make are based on data rather than intuition.

Speaking of sales, supermarket sales in particular are a powerful indicator of a supermarket's good performance in terms of balanced supply and demand, optimal pricing and stocking decisions and more factors that lead to customers satisfaction. Supermarket sales data have proven important across a variety of fields, and has been used for different purposes such as nutrition mentoring as they have proved to be more practical than traditional monitoring methods such as national surveys in examining food purchasing pattern (Tin et al., 2007). In a similar area, sales data was also used to consider ways to address cost differential between healthy food options and other regular food (Siddharth, 2007). Not restricted to nutrition-related researches but also declined market share derived from supermarket sales have been used to determine whether international conflicts do actually lead consumer to boycott certain brands (Sonar and Rajkumar, 2016).

Fortunately, industries are aware of that relevance of sales and sales data and how it can drive a variety of decisions, and so it is crucial to assist in business processes and derive better results. Moreover, good sales alone are not enough and fluctuations of sales over time is a major problem faced by most of the industries, and that is why managers and decision makers seek good sales prediction models through which stability can be achieved. Sales predicting is now a non-underestimated task that has a dedicated team of statisticians, data analysts and scientists. For example, it is used for planning sales and distribution logistics (Prasun and Subhasis, 2007). And in economics, strategic planning for resource allocation can be improved using good forecasting of sales (John and

Rhonda, 2002). Despite its relevance, sales predicting is a complex process and has always been challenging because it depends on internal and external factors. Variations in customers taste and demands, financial status and the marketing budget set by a certain supermarket are among the internal ones. There are also some external influencing factors such as the overall fluctuation in economics, international trends in foods and dietary habits, and special occasions (Chi-jie, 2014).

1.2 Technical Background and Problem Framing

1.2.1 Machine Learning

The problem proposed in this thesis is sales prediction, where information about the items sold and the stores in which those items are exhibited will be used to predict the sales that items would make when sold in new stores.

The problem is defined with the three key words: machine learning, prediction, and regression, and they are overlapping that is why they are sometimes given other names and are used interchangeably such as calling machine learning as predictive modeling and vice versa.

Machine learning is the branch of science where computer algorithms are developed to perform tasks without human guidance, instead of relying on hard-coded rules. In other words, what machine learning is all about is the ability for computers to induce new knowledge. Machine learning algorithms have been used widely and successfully in many areas (Maxwell, 2015). Machine learning tasks are classified into two major categories: supervised and unsupervised machine learning. The problem presented in the thesis is a supervised task as how the majority of practical machine learning tasks are, because the algorithm has some prior knowledge to what the output should be. A supervised machine learning task is defined as follows: learning process is achieved by feeding the model with some data in order to be capable of learning a mapping function which we do not know but the algorithms will try to figure it out. Basically for machine learning problems there are input variable(s) X and output variable(s) Y, and using different machine learning algorithms a mapping function from X to Y is learned as in the following equation: $Y = f(X)$. In addition, there is the error factor that is independent from the predictors: $Y = f(X) + e$ which is called the irreducible error because it can not be reduced no matter how good the mapping function is estimated. The power of this science is that the trained model will then be able to predict values of new ”unseen” data other than the training data points that were initially fed to the model. For as we know, there are two types of supervised learning problems, classification and regression. The problem proposed here is a regression one because the values of the target variable (Sales-the Y in the equation) are of a numeric type. Thus, when an algorithm is applied to map X to Y it is a process of developing a prediction model which its ultimate objective of model development is to provide the most accurate predictions that are the closer to the real values.

1.2.2 Hyper-parameter Optimization

Hyper-parameter optimization or tuning is a process by which the best set of hyper-parameters are chosen so they result in the best performance by the model. The model basically will be making trails until the best hyper-parameters are found. Each trail is itself a full training process of the model. So, it is a process through which the training process can be controlled to some extent and its aim is to optimize the model performance depending on the possible values that these parameters may take. Hyper-parameters differ from normal parameters in that they are set before the learning process begins, and most machine learning algorithms have these hyper-parameters but they are model- or algorithm-specific. The affect of the hyper-parameters extends not only to the duration of the training process but also to the accuracy of the predictions produced by the algorithm. There are two common ap- proaches to hyper-parameter tuning: grid search and random search. In grid search approach, there will be a grid space in which different possible values of the hyper-parameters are defined and the model will test each combination of them. While in random search, no discrete set of values are provided as a grid for exploration, but rather a random combination of ranges of values are chosen and tested for the best performance.

1.3 Contibution and Brief Overview

The aim of this thesis work is to test the capabilities of four machine learning regression algorithms (Decision Tree, Random Forest, Extreme Gradient Boosting, and Support Vector Machine for Regression) for predictive modeling for sales data. These algorithms were previously chosen in the field of predicting and forecasting, yet in this thesis work they are compared with one another in a different way. The comparative analysis carried out was approached as follows: each one of the algorithms is evaluated against four performance metrics under each of the following circumstances:

1. The models are trained with the complete set of input predictor variables with their default parameters. After the models are developed, an ensemble model is developed combining all the four models.

2. Hyper-parameter optimization (model tuning) technique is applied and the models are re-trained with a new set of hyper-parameters, again with the complete set of predictor variables.
3. Specific feature selection procedure is followed and the models are re-trained according to the results of the procedure.

The implementation of the experiment will let us investigate the impact of each configuration and how these techniques would perform on the provided data. Although this thesis is focused on the supermarket retail industry, but the method- ology and the implementation process can be generalized and applied in other domains where similar cases are provided; in any industry where sales prediction is to be carried out.

1.4 Related work - Literature Review

1.4.1 Ensemble Learning

Ensemble learning is an important concept that is applied in a number of fields, including pattern recognition, machine learning, statistics and neural networks. As for we know, ensemble learning process goes through three stages. First stage is generating a set of models, second and not always occurring stage is pruning ensemble in which best models are selected and the ones that do not improve the performance of the ensemble are eliminated. The last stage is ensemble integration where the best models are combined together taking into consideration the multi-collinearity problem that might arise at this stage (João et al.,2012). Although the works done on ensemble machine learning methods are described to be under-represented in the literature (William and John, 2015), researches have proved that ensemble modeling often results in more accurate predictions (Ye et al., 2016). But while a reasonable amount of research focused on classification-based ensemble, where different methods for constructing ensembles have been thoroughly investigated, (Thomas, 2000), much fewer on the regression-based one, thus this research focuses on the later.

Among different applications, ensemble methods have been used for a cancer classification task on gene expression data (Ching, 2006), multi-class classification problems such as in IRIS data classification and fraud detection (Hyun-Chul et al., 2003), and macroarray data analysis (YonghongPeng, 2006). In cheminformatics, which is a branch of informatics that relies on Machine Learning, an experiment has been held to compare the performance of lin- ear and greedy ensemble models to their single components predictors (William and John, 2015). Also, forecasting has been the main objective, with an ultimate goal of improving the model performance, of most researches out there but in different fields where precise prediction is required. A wavelet-based ensemble strategy has been introduced for a load forecasting problem (Song et al., 2016).

The use of ensemble modeling was investigated for wind power prediction (Oliver and Justin, 2016). Ensemble Learning paradigms consist of bagging and stochastic gradient boosting was also used to predict streamflow, which is a process within water resource planning (Halil and Onur, 2013).

1.4.2 Decision Trees

Decision trees are one of the most interpretable machine learning algorithms. They organize information taken from training data in a tree-based system for providing decisions. Regression trees which are one of the two type of decision trees (the other one is classification trees), has been used in forecasting tasks widely; predicting bank loan credit losses (João, 2010), short-term load forecasting in power systems (H. Mori et al., 2002), and recently utilized in the prediction of solar power generation (Caroline et al., 2017).

1.4.3 Random Forest

Also known as Random Decision Forests, and as the name suggests they also imply a tree-based structure but utilize multiple trees, somehow randomly, rather than one decision tree and at the end produce the mean prediction (in case of regression problems). Random forest is technically itself an ensemble method while at the same time being a tree-based model. The application of Random Forests in forecasting ranges from predicting real-time price in the electricity market (Jie et al., 2014) which is a key factor in winning a power bid, and predicting the next 24 hours of electricity load won which several crucial tasks regarding power operators rely (A.Lahouar and J. Ben, 2015), to a very recent application where utilized in a combination with other Machine Learning techniques that is predicting solar irradiations on hourly basis (Caroline et al., 2017).

1.4.4 Extreme Gradient Boosting (Xgboost)

Like in random forests, Extreme Gradient Boosting (XGBoosting) is also a form of ensembling method of weak predictors, typically decision tress and it can also be linear though. Obviously an application of gradient boosting but and extended one that utilizes regularization features in order to control overfitting. Xgboost is being used in research areas of growing importance such as miRNA-disease association prediction, a problem that has been addressed using either Network Analysis or Machine Learning techniques (Xing et al., 2018). Also applied in old but continuing problems such as banckruptcy prediction (Maciej et al., 2018). In addition to other prediction problems such as crude oil price (Mesut and Mustafa, 2017), grid-connected photovoltaic plant production (S.Ferlito et al., 2017), and as a part of forecasting system for birds migration (Benjamin anf Kyle, 2018).

1.4.5 Support Vector Machine

Support Vector Machines are a supervised machine learning technique, or for our case SVR as for the other type of tasks this technique can handle, which is regression. What makes support vector machines different from other regression methods is the ability to fit a curve rather than a line through kernel functions. As with previously discussed methods, SVRs are also used in a wide range of forecasting problems; such as real-time flood forecasting to predict flow rates and water levels (Pao-Shan et al., 2006), time-series financial forecasting (Chih-Chouh et al., 2009), tourism demand forecasting (Kuan-Yu and Cheng-Hua, 2007), and wind speed forecasting (G.Santamaría-Bonfil et al., 2016).

Chapter 2: METHODOLOGY AND DATA ANALYSIS

The following section is dedicated for describing of the methodology followed to conduct this thesis work, in addition to briefly highlighting data analysis procedures undertaken to prepare data for predictive modeling.

2.1 Methodology

The work done in this thesis consists of four phases, namely a literature study phase, data analyzing and understanding phase, an implementation phase, and an evalu- ation phase. These phases were mostly run in a sequential manner, only for the implementation phase where many of its sub-phases were iterative.

In the literature study part the main goal was to read up on and become acquainted with the following things: decision trees, random forest, extreme gradient boosting, support vector machine for regression, ensemble models, hyper-parameter tuning. Techniques for features selection were also investigated because sometimes data com- plexity and the big number of features is an obstacle for performance improvement. Usually ,data analysis is done as the first step of any machine learning work flow, and in this experiment data analysis is undertaking as the first stage and is explained later in this chapter. Through this step, we dig deep into the data in order to thoroughly understand it and extract some relationships between variables.

The implementation phase consisted of the following parts:

1. Applying necessary feature engineering steps in order to prepare the data for the next step of model development.
2. Developing models using the complete set of features.
3. Performing hyper-parameter optimization and re-train the models as in the second step.
4. Building ensembles of the developed models.
5. Choosing a subset of features according to a defined procedure (explained in the next chapter) and develop new models using only the subset of features.

2.2 General Information About the Data

The dataset analyzed and used for the experiment of this thesis work belongs to the international brand BigMart supermarket chain, a grocery retail company, which has branches in many countries around the world and was established back in 2007. The company involves a very wide range of brands (more than 22,000 brands) of products ranging from breakfast ingredients and frozen foods to house goods and products for health and hygiene. This data set belongs to the year 2013 and includes 1559 products distributed across ten stores in different cities.

2.3 Dataset Properties

The data set consists of 8523 observations (instances or rows) and 12 attributes (variables or columns). Each of these observations represents a unique product/item that posses a unique identification number. Also, each item has a value for all the 12 variables. These variables of the data set can be classified into two groups; those related to the item itself (8 variables), and the others are related to the store/outlet in which those items are exhibited (4 variables). The complete set of variable is as follows:

1. Outlet identifier.
2. Outlet location Type.
3. Outlet Type.
4. Outlet Size.
5. Outlet establishment year.
6. Item identifier.
7. Item weight.
8. Item fat content.
9. Item visibility.
10. Item type.
11. Item MRP.
12. Item outlet sales.

First, outlets attributes. The company sells its products in ten different stores, each one with a unique id number starting with letter ”OUT” indicating an outlet. These outlets located in three different location types, they are Tier 1, Tier 2, and Tier3. Tiers are part of a classification system used in the Indian government to distinguish cities according to different aspects such as business, real estate and commercialization. Tier 1 cities are the biggest and are highly commercialized, while Tier 2 cities are smaller, they have business markets but with less population. Finally, Tier 3 cities are those with less than a million people living in it and are basically the minor cities. Outlets are further classified into 4 groups depending on the types which are Supermarket1, Supermarket2, Supermarket3, and groceries. There seem to be no differences among the 3 types of supermarkets. The outlets are either small, medium, or large in size or fall into forth unlabeled category. The year in which the outlet was established is recorded. The year takes 1 value out of 9 between 1985 and 2009.

Coming to items attributes. As mentioned previously in this section, there are 1559 items, the Ids given for the items consist of 3 letters and 2 digits. The letters refer to the type of item where ”FD”, ”DR”, and ”NC” refer to food, drink, and non- consumables respectively. Items weight feature is self-explanatory; indicates how mush the item weighs. The level of fat is also recorded as one of five provided values (Low Fat, low fat, LF, Regular, reg), but this classification can not be applicable to non-consumable items and will be treated during data preprocessing. Items visibility indicates how much space the item takes on a shelf, and thus this attribute cannot take a value of 0 because it would mean that the item is not visible which does not make sense, this issue will also be treated later. For types of Items, there are 16 different ones. Also, each item has Maximum Retail Price (MRP), that is the highest price a buyer could be charged for a product. Basically this attribute is a list of prices. Finally, the sales of an item which is the target variable.

2.4 Exploratory Data Analysis (EDA)

Exploratory data analysis is the first step towards all machine learning tasks as it enables better and deep understanding of the data and features provided. Exploring the data visually is one of the most effective ways to understand distri- butions of the variables, find missing values and to think about the best way to deal with them and investigate relationships between variables. For organizational purposes, the first part of EDA will involve uni-variate EDA and particularly start- ing with the target variable, followed by numerical variables, and finally categorical variables, each of them individually using histograms and bar plots for numerical and categorical variables respectively.

Item outlet sales variable is not normally distributed as it is clear from its distribution it is right-skewed towards the highest sales and the concentration is on low values, and thus will need to be transformed in order to be as close as possible to the normal distribution (symmetric), as will be explained later. This also applies to item visibility in terms of skewness, while no clear pattern was observed in items weight which might suggest that it has no or very little relationship with the target variable. The last numerical variable is Item MRP where 4 clear distinct groups are seen, and this is to be kept to be used in feature engineering section. The distributions of sales, item visibility, item weights, and item MRP are illustrated in **Figure 2.1**, **Figure 2.2**, **Figure 2.3**, and **Figure 2.4** respectively.

In a similar way the categorical variables are explored and visualized using barplots which are customized for non-numerical variables as they display the fi- nite set of values for given variable and their frequencies. In fat content variable, there are three values: LF, low fat, and Low Fat that indicate the same category and for this reason they are combined together. The same is applied to ”reg, Regular”.

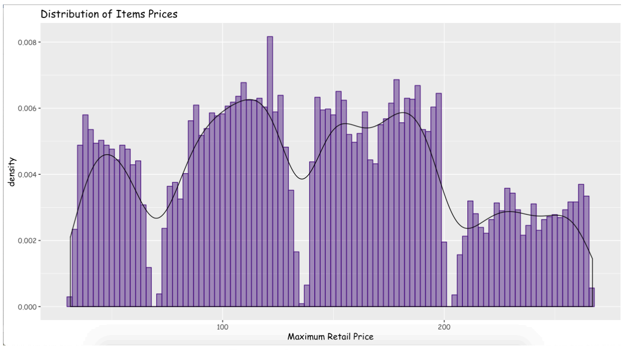


Figure 2.1 Item MRP ditribution

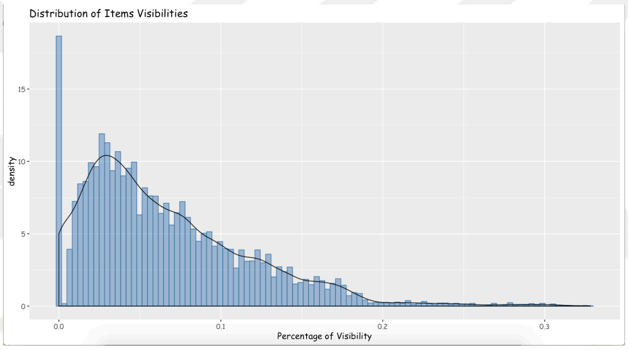


Figure 2.2 Item visibility distribution

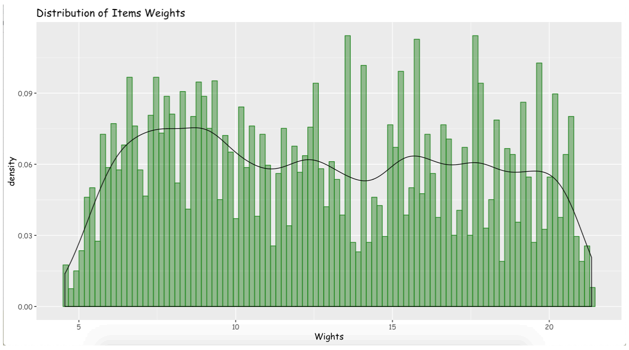


Figure 2.3 Item weight distribution

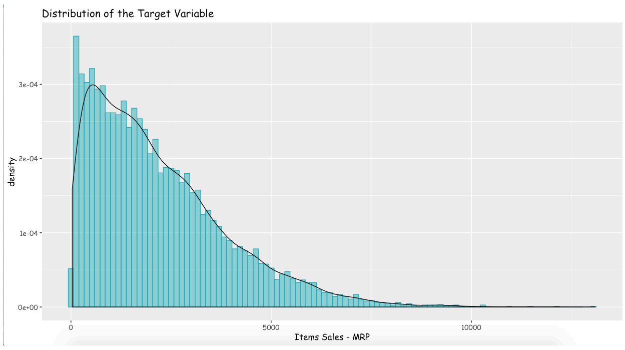


Figure 2.4 Item outlet sales distribution

As for Item type variable, it has 16 different type of items sold in the outlets. Break- fast and Seafood are the types with least data observations, where each of snacks and fruits and vegetables has approximately 2000 data points. There are 4016 ob- servations where the value of outlet size is missing or blank and that is an issue to be resolved later. Outlet establishment year, although its values are numerical in nature, but it has a finite number of values and thus it is explored with categorical variables. Comparing the nine values of outlet establishment year, it is obvious that less number of observations for the outlets established in 1998. As mentioned before, there has not been figured any clear distinction points between three types of supermarkets, but supermarket type 1 has more data points and supermarket type 2, type 3, and grocery all have low expression in the distribution. barplots of the categorical variables are numbered as **Figure 2.5** to **Figure 2.12**.

After univariate EDA bivariate analysis takes place where hidden relationships between target variable and independent variables are explored mainly using scatter plots and then used in missing data imputation, data preprocessing, and feature engineering.

2.4.1 Numerical Variables against Sales

1- Sales vs. Item Visibility: In general, it make sense that the visibility of an item will impact its sales, that is the location of the item in the store and the space it covers will actually make it less or more sold. But as clear from figures **Figure 2.13**, the more visible the item the less higher its sales. This could be because the products with highest sales are the ones with high demand and thus visibility will not affect the sales; people do not need to see the product very clearly as they are already planning to buy it.

2- Sales vs. Item Weight: **Figure 2.14** shows low correlation between the two variables as sales are well-spread across the range of weights and depending on **Figure 2.14** alone we could get rid of this variable for the model development.

3- Sales vs. Item MRP: While exploring item MRP individually the four distinct groups were observed, and they appear again when plotting item MRP against item outlet sales as shown in **Figure 2.15**. The meaning of these four groups can be also concluded as a positive relationship between the two variables is observed; when the item is low-priced it would have low sales, and as the price is getting higher, sales also grow.

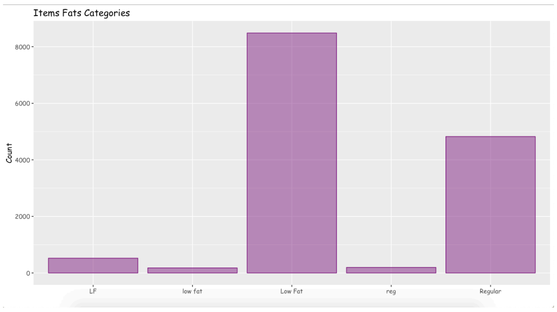


Figure 2.5 Item fat content levels

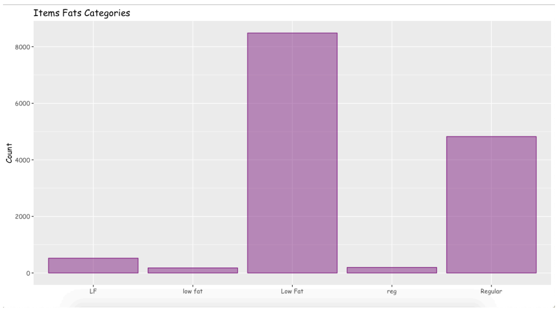


Figure 2.6 Item fat content levels cleaned

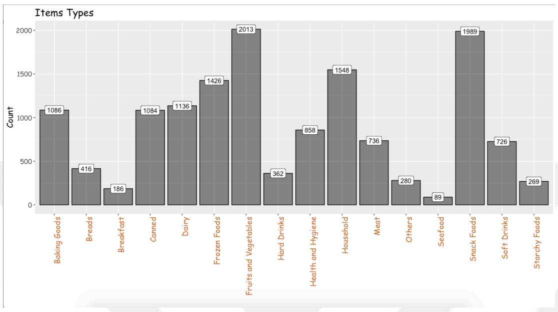


Figure 2.7 Item types

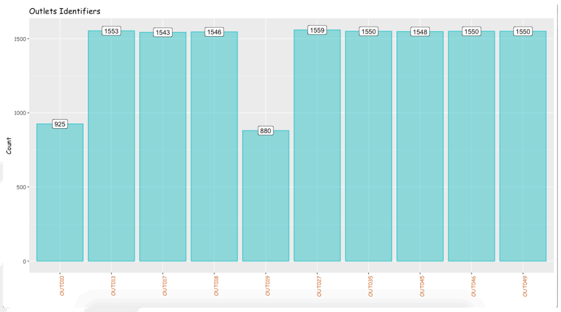


Figure 2.8 Outlet ids

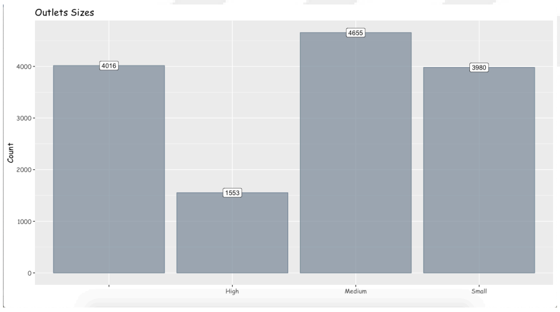


Figure 2.9 Outlet size

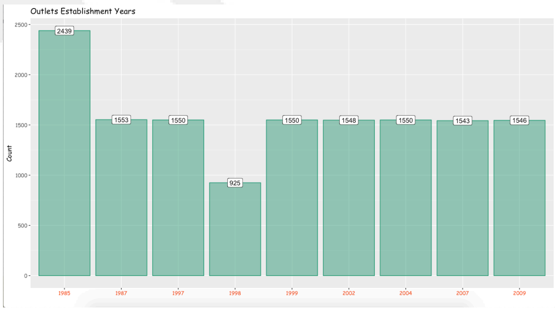


Figure 2.10 Outlet establishment year

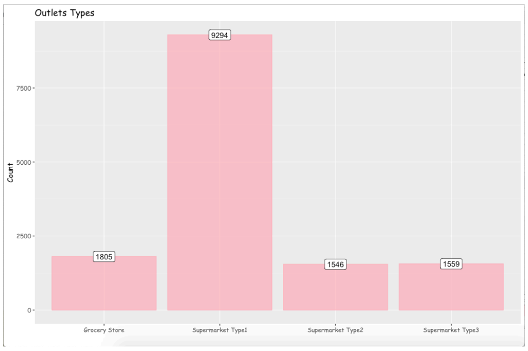


Figure 2.11 Outlet type

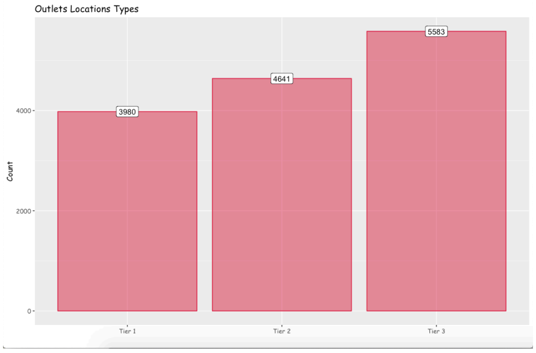


Figure 2.12 Outlet location type

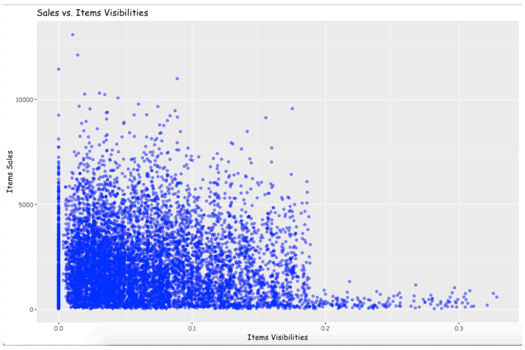


Figure 2.13 Sales vs. Visibility

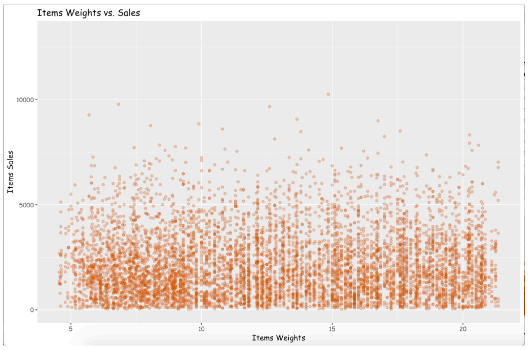


Figure 2.14 Sales vs. Weight

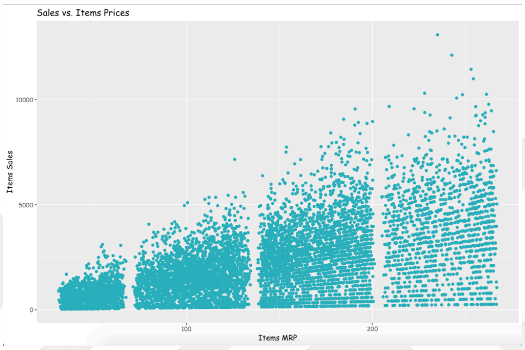


Figure 2.15 Sales vs. MRP

2.4.2 Categorical Variables against Sales

4- Sales vs. Item Type, Outlet Location Type, and Establishment Year: The distribution of the target variable across distinct categories of item type shows no important relationship. Violin plots are also utilized to discover these relationships as their width at a certain level indicates the density of data at that level, while the height refers to range of values of the target variable. As a conclusion, no variation in sales is observed due to different item type, as ashown in **Figure 2.16**. The same also applies to the distribution of item outlet sales across the three categories of outlet location type except that it is noticed, as in the box plot in **Figure 2.17**, that tier1 and tier3 look more similar which could be weird because it would be expected for tier2 and tier3 for example to be similar as those types of cities are closer to each other than how tier3 cities are close to tier1 cities in term of population and other characteristics as described before. But it is not a big issue because the difference is not that big and all three tiers are still similar. Another note about outlet location type from their bar plot in **Figure 2.12** is that more stores are presented in tier2 and tier3 cities (small and medium size cities). Also no significant meaning describes any relation between outlet establishment year and item outlet sales. While there could be different factors affected the stores established in 1998 to have lower sales than others, as shown in **Figure 2.18**.

5- Sales vs. Outlet Type: Grocery outlets seem to have most of its data observation around low values of sales, while the opposite is observed with types of supermarkets. This makes sense as grocery stores have many reasons to generate fewer sales than supermarkets' such as the smaller size of the store and thus fewer products and options presented for the customers and visitors. One approach that could have been applied to outlet type variable is to minimize the four types into only two, grocery and supermarkets. But as seen from plotting sales against outlet type, as shown in **Figure 2.19**, supermarket type 3 has the highest sales and thus it would not be a good idea to just combine the three types of supermarkets.

6- Sales vs. Outlet Size: The distribution of sales across the three categories of outlet size is also very normal. And opposite to what one could have assumed initially, that is sales are higher in larger supermarkets, from the plot in **Figure 2.20** it is clear that medium-sized outlets have the highest sales and the largest outlets are combined with the lowest sales. The forth blank category will be discussed in the next subsection.

7- Sales vs. Outlet Identifier: From the boxplot of outlet identifier shown in **Figure 2.21**, it is clear that outlets OUT010 and OUT019 are different from other stores. When boxplots are filled with outlet type it is noticed that outlets OUT010 and OUT019 fall into grocery category which explains low sales compared to other outlets classified as supermarkets. Though this does not necessary mean that highest sales will go to a large sized supermarket of type 1 as it can be seen from the plot the outlet with highest sales is medium sized and a supermarket of type3. Actually, there are 6 outlets of type supermarket type 1 and of size high (large) but they do not have the highest sales. Also, there are one outlet of type supermarket type 2 (OUT018). Finally, one outlet, OUT027, of type supermarket type 3 and of size medium and it does have the best sales.

8- Sales vs. Item Fat Content Types: Low fat products recorded higher sales than regular fat products'.

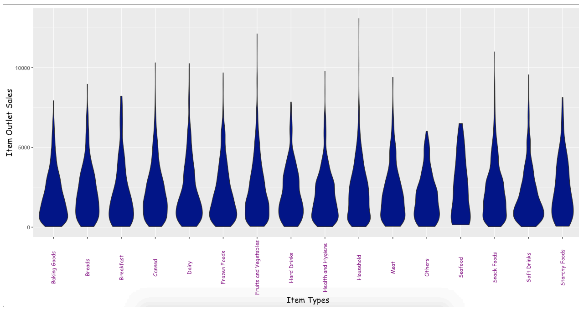


Figure 2.16 Sales vs. Item Type

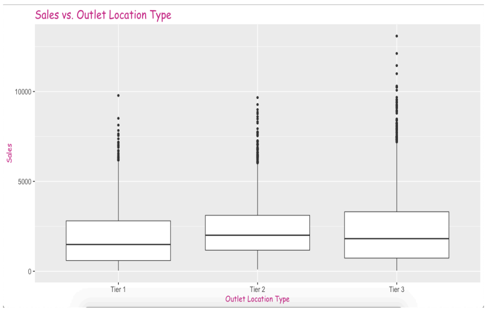


Figure 2.17 Sales vs. Outlet Location Type

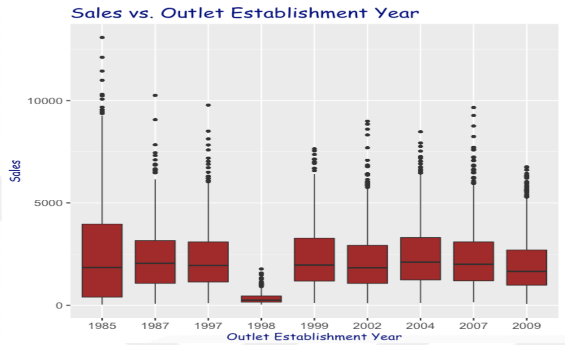


Figure 2.18 Sales vs, Outlet Establishment Year

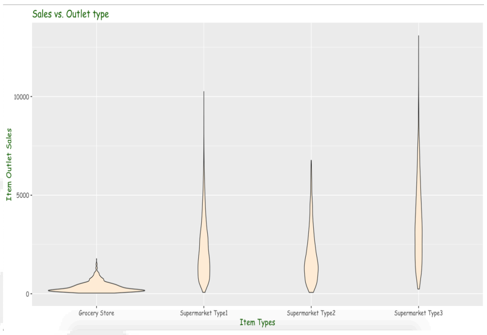


Figure 2.19 Sales vs. Outlet Type

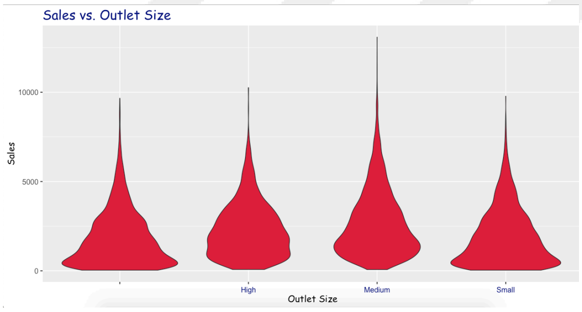


Figure 2.20 Sales vs. Outlet Size

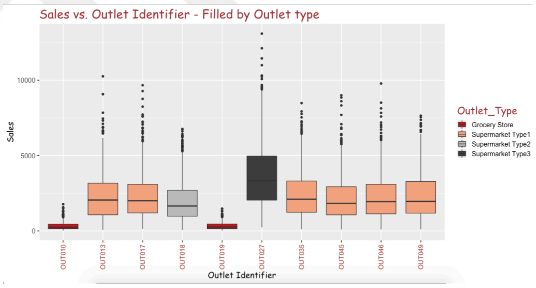


Figure 2.21 Sales vs. Outlet Id

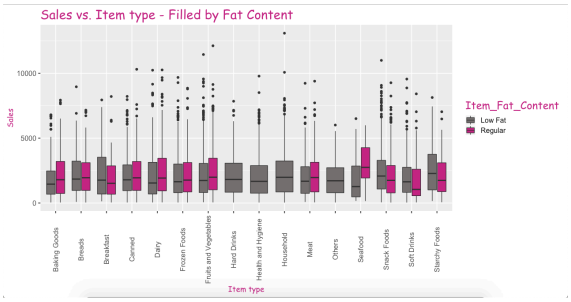


Figure 2.22 Sales vs. Item Fat Content

Chapter 3: Implementation and Experiment

This chapter describes the implementation phase starting from data cleaning and feature engineering to models development.

3.1 Missing Values Treatment

A very basic and initial step is handling missing values existing in the data set be- cause some machine learning algorithms simply drop the rows containing missing values which lead to reduced training data size and eventually to a decreased per- formance of the model.

1-Items Weight: during exploratory data analysis item weight feature appeared to have 1463 missing values. These values can be imputed with the mean value of the variable. Another possible imputation value could have been the median value but since the two values are very close to each other, the choice will not be of a great impact on the model accuracy.

2-Outlet Size: There are a blank category within the outlet size variable which its main values are small, medium, and large. These blank values are related to outlets OUT010, OUT017, and OUT045 as illustrated in the **Figure 3.1** and **Figure 3.2**. One approach to deal with this blank category is to impute the blank values with the mode value of Outlet Size variable; the most occurring value, whichis medium. However, the followed approach for filling the blank category is to impute with the mode of outlet size according to outlet type. When we check the type of outlets it appears that OUT010 is a grocery store, while OUT017 and OUT045 are of type supermarket type 1. From **Figure3.1**, there are two grocery stores and they are of size small, thus, OUT010 will be assigned the value small. Calculating the outlet size for supermarket type 1 it is medium, so the size of OUT017 and OUT045 is medium.

3- Item Visibility: This variable contains some zeros among its values, which has already been mentioned not to be normal because this would mean that the product is not visible to the customers and visitors. This could be a type of error that happened during data entry for instance. One suggested approach for imputing the zeros is imputing by the median or mean value of item visibility. The followed approach is called the regression approach in which a simple linear model is built with item visibility being the target variable and the other variables as the predictors.

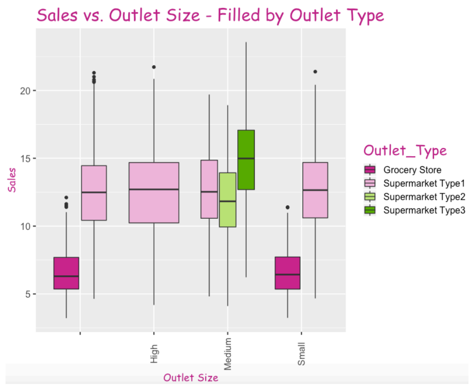
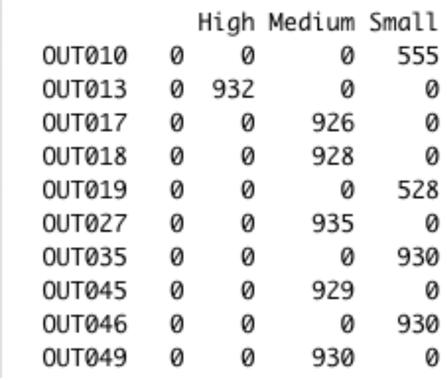


Figure 3.1 Outlet types and sizes



```
          High Medium Small
OUT010  0    0      0   555
OUT013  0  932      0     0
OUT017  0    0    926     0
OUT018  0    0    928     0
OUT019  0    0      0   528
OUT027  0    0    935     0
OUT035  0    0      0   930
OUT045  0    0    929     0
OUT046  0    0      0   930
OUT049  0    0    930     0
```

Figure 3.2 Outlet identifiers and sizes

3.2 Outliers Detection Treatment

One of the best ways to detect outliers is to demonstrate data visually. The dataset contains three continuous variables that need to be detected against outliers, item MRP, item visibility, and item weight. It appears that item MRP and Item weights do not contain any outliers, where six outliers are found in item visibility variable. These values are bigger than 1.5 * Interquartile Range of the variable. Because these are only six observations out of 8523 no treatment is needed because they are not considered as significant. Box plots of the three numerical variables are shown in **Figure 3.3**, **Figure 3.4**, and **Figure 3.5**.

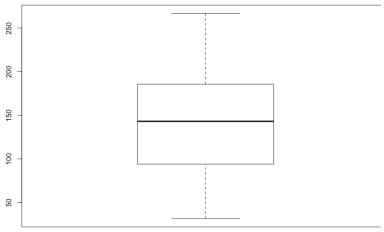


Figure 3.3   Item MRP box plot

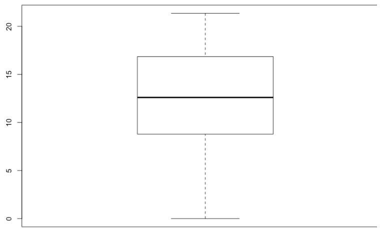


Figure 3.4 Item weight box plot

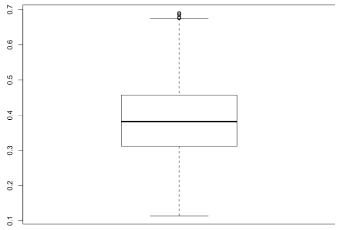


Figure 3.5 Item visibility box plot

3.3 Feature Engineering General Description

Feature engineering is the art of manipulating feature of data, whether by extracting new features or, so they better represent the underlying problem to the models. Although the lack for scientific approached to handle this process, feature engineering is considered to be an important part to the success of a machine learning model.

This importance of feature engineering partially due to the importance of the fea- tures themselves, as they are a representation of the input data used by machine learning algorithms to produce some outputs.

3.3.1 Feature Extraction

1-Item Type: In the previous chapter where bivariate exploratory data analysis was discussed, no variations in the Item outlet sales were observed according to the different types of items. For this reason, one approach to deal with these unnecessary values was to create a new variable to represent a broader category of the item rather than a very specific type. For example, the new feature could classify items into perishable (for breakfast, breads, dairy, fruits and vegetables, meat, seafood, and starchy foods) and non-perishable (for baking goods, canned, frozen foods, hard drinks, soft drinks, and snack foods), but another level of classification can be applied by taking ad- vantage of codes recorded in Item Identifier feature, where the first three letters indicate the category to which the item belongs (FD for food, DR for drink, NC for non-consumable). This feature derivation step helps in reducing the dimensionality of data (the sixteen categories would have needed to be one-hot encoded as will be described later) while at the same time keeps some sort of broad item classification criteria. As for visualizing the new category, it is observed from the figure that item categorized as food are the ones that are most sold and this comes to be reasonable when seeing that there are ten out of sixteen different types which are food as illus- trated in the **Figure 3.6**.

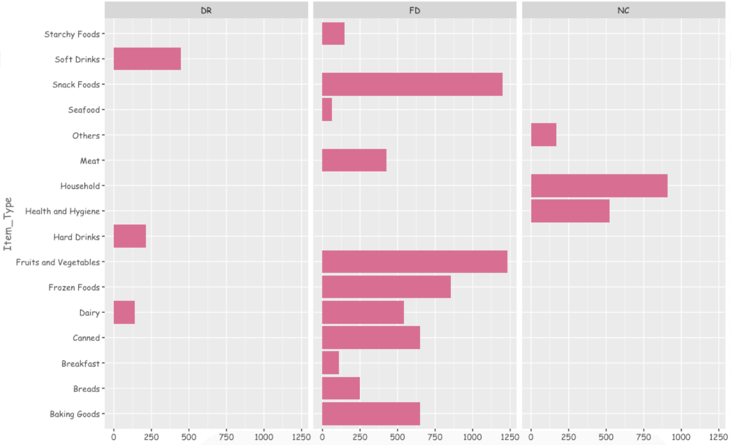


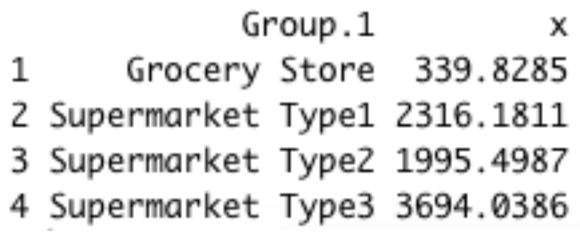
Figure 3.6 Item types according to the broader category

2- Outlet Operation Years: The year in which the store was established does not have a significant relationship with sales, but another useful variable can be de- rived, that is Outlet Operation Years which equals to establishment year subtracted from 2013 (The year to which all data belongs). This variable makes more sense because using it one can compare between, for example, outlets that have been operating for 14 and those that have been operating for four years.

3- Item MRP: From the scatter plot of Item MRP four distinct groups of prices were clearly observed, and that was a motivation to create a categorical feature called Item MRP Groups that represent the relationship between Item MRP and Item Outlet Sales in a good way.

4- Item Fat Content: Among item types there are Health and Hygiene, House- hold, and Other which ”Regular” or ”Low Fat” criteria does not apply and so another category is added, ”not-applicable”, as the Fat Content value for those non-eatable and non drinkable products. This step is not considered as a feature extraction but a refinement to the categories of an existing feature, which also contribute to the same bigger goal of better representation of the features for machine learning models.

5- Outlet Type: One suggested approach to deal with categorical features and make them better represent the data is to combine some levels that could be similar to each other. During applying exploratory data analysis to outlet type, it was observed that supermarket type1 has much larger expression than of type2 and type3 and they are both are located in tier3 type of cities, which could leads to considering combining the two types (2 and 3) as one type leaving the levels of the features as grocery, supermarket type1 and supermarket type2 only. However, looking at the mean value of item outlet sales it appears that it is very different in the two types and thus they had not been combined.



```
            Group.1         x
1     Grocery Store  339.8285
2 Supermarket Type1 2316.1811
3 Supermarket Type2 1995.4987
4 Supermarket Type3 3694.0386
```

Figure 3.7 Sales mean values in outlet types

3.4 One-hot Encoding

While there is the fact that most machine learning algorithms work better when supplied with numerical variable only, the dataset provided contains seven categori- cal variables and three numerical ones. Thus, categorical variables cannot be simply omitted as this would affect the performance of the predictive models. One approach is to convert them into their numerical form variables using the well-known one-hot encoding approach.
In case of provided data Label Encoding or Integral Encoding will not work and even could product unexpected results, because if the categories of item type for example would be encoded as ”1” for Breakfast, ”2” for Sea food, ”3” for canned, and so on, automatically the computer will treat the higher numbers as if their categories are higher in weight, but indeed there no exist such an ordered relationship between val- ues and

here comes one-hot encoding as a good solution for this conversion process. One-hot encoding utilizes a binary representation; if there are N distinct values of a categorical variables, they will mapped to N-1 features. Outlet size for example has three categories, when an outlet size is large, it will be represented by a ”1” to the newly created outlet size large and zeros for medium and small. Whereas when the value is medium it will be presented as a ”1” and zeros for large and small, etc. This method can also be called as the process of creating dummies for categorical variables. This applies to all categorical variables in the dataset.

After Applying one-hot encoding dimensions of the dataset has changed as it now contains 31 variables, instead of the initial 12 variables.

3.5 Power Transformation

During univariate EDA there were two numerical variables that proved to be right-skewed, Item Visibility and the target variable Item Outlet Sales. When the variable is skewed the accuracy of statistics techniques and some machine learning methods are effected as these algorithms assume that variables are normally distributed (a case when there is almost equal distribution of data points are above and below the mean value of the variable). One of the methods that is commonly used to transform left or right skewed numerical variable so as to be normally distributed is power transformation. For our data cube root transformation was applied. After transformation Item Outlet Sales, its values have become as close to the normal distribution where the mean value (11.992) almost represents middle point of the data as the minimum value is 3.217 and maximum value is 23.566. The same procedure was applied to Item Visibility and it has been normalized; minimum value is 0.1135, mean is 0.3849, and maximum value is 0.6899.

3.6 Numerical features Scaling

Scaling is useful when some type of ”unification” is needed between variables so all of their values fall between the same range and so can be weighted correctly. Some of the machine learning algorithms uses different distance calculations and the most common is Euclidean Distance where scaling become critical as varying magnitudes of variable causes problems, such as in k-nearest neighbours, performing Principle Component Analysis (PCA), and Naive Bayes. However, scaling techniques are not applied to the dataset because the algorithms utilized are not affected by this concept.

3.7 Correlation

One important aspect after performing exploratory data analysis and feature engineering is to start understanding the relationships between independent variables, what is called multicollinearity, and the relationships between them and the dependent variable as this will help in including and excluding variables according to their importance and impact on model performance. Correlation shows what the statistical relationship is and its direction whether positive, negative, or neutral. Some of these relationships are clearly observed from scatter plots, but looking into numbers will tell more. For example, from item outlet sales vs. item MRP scatter plot one can tell that a kind of positive relationship between the two variables does exist. And be-cause there are more than 25 variables in the data set it is very hard to tell from figures and plots how the data is correlated.
The indicator used to investigate relationships is the correlation coefficient which as goes closer to 1 emphasizes the existence of a positive relationship, while indicating a negative one when it approaches to -1.

Some derived relationships between independent variables and the outcomes are:

1. There is moderate positive correlation between Sales and MRP.
2. Moderate negative correlation between Sales and Outlet Identifier OUT019.
3. Weak positive correlation between Sales and Outlet Identifier OUT027 and also between Sales and Outlet Size Medium.
4. Weak negative correlation between Sales and Outlet Size Small.
5. Weak positive correlation between Sales and Supermarket Type1 and Type3.

In addition, most of multicollinearity that could be addressed is because one of the variables, for example Item MRP Groups Very High, is derived from another, Item MRP. Accordingly no special procedure applied to treat the issue as this multi- collinearity will not affect the predictions themselves. As for another example of is Outlet Establishment Year and Outlet Operation Years where there exists a perfect negative relationship between them, that is why Outlet Establishment Year will be deleted from the data set because it is no longer significant as a predictor. The correlation matrix shows these correlations and some initial decisions are taken regarding the more important features that will be supported with feature selection procedures followed which is described in the next section.

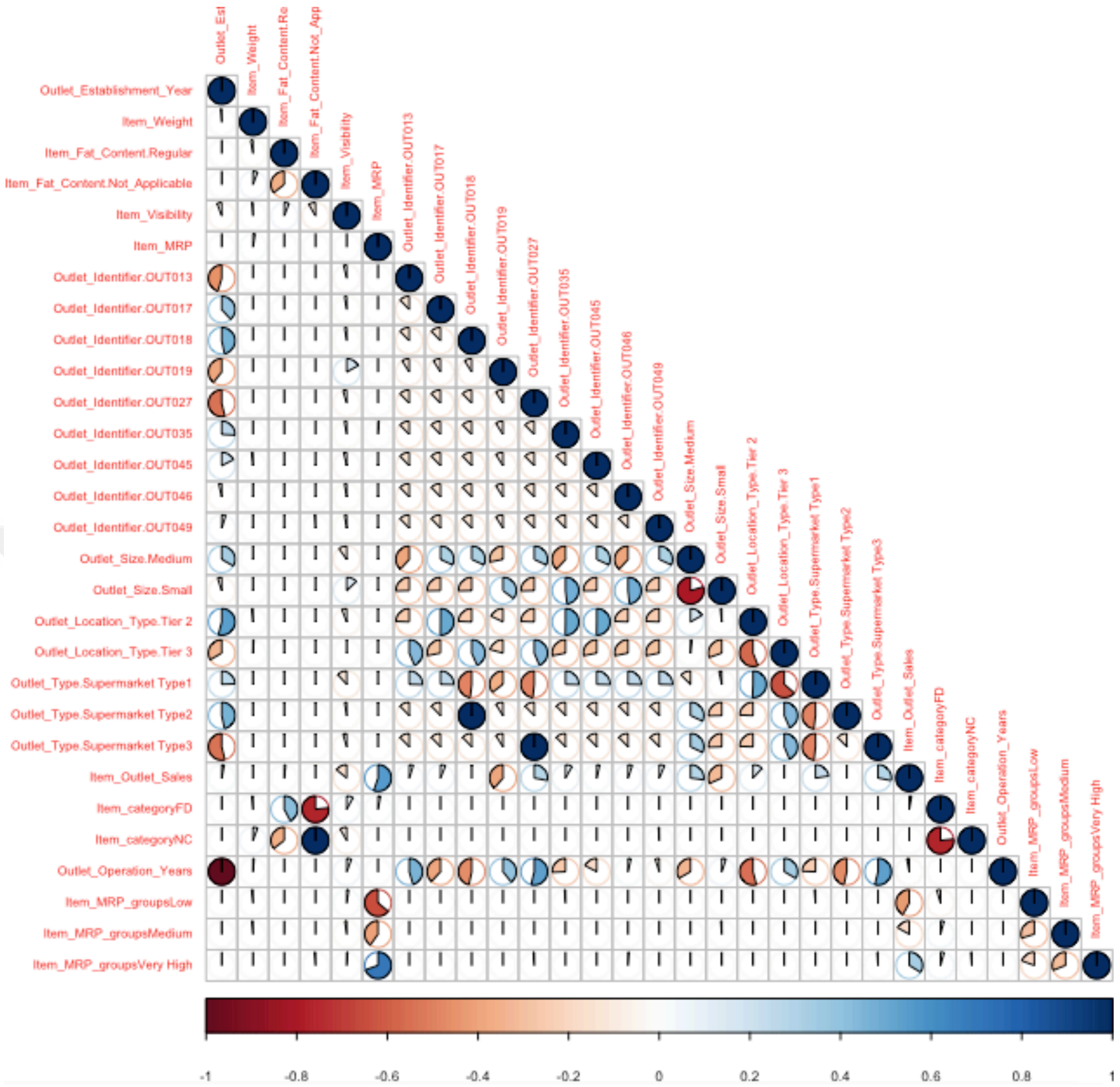


Figure 3.8 Correlation matrix

3.8 Feature Selection

Feature selection is a process that can be considered as the first step towards developing a machine learning model. The need for this process is that data sometimes can be large and probably contains some features that are irrelevant to the prediction process of the target variable and including these features in the predictors set could affect model performance negatively. In other words, the aim is to achieve improved performance or reduced error using correlated data. In addition, it reduces the time the algorithm takes for training and this is especially true when performing some sort of hyper-parameter optimization. Feature selection differs from dimensionality reduction techniques- although both have the same ultimate goal that is reducing unnecessary data- in that the later does so by creating new combinations of variables.

Feature selection algorithms are generally classified into three classes; filter methods, wrapper methods, and embedded methods. Filter methods such as calculating correlation coefficient scores involve ranking the features and make a decision whether to keep them in the data set or not. Basically, the correlations discussed in the pre- vious section was the first step in this feature selection process. Wrapper methods on the other hand, a subset of features are used to train a model and then adding or removing features from the selected sets based on the performance of the initial model. The traditional wrapper methods are forward selection, backward elimi- nation, and recursive feature elimination. Finally, in embedded methods feature importance scores are calculated as a part of a machine learning model development. The specific procedure followed in this experiment will be described in the performance evaluation chapter.

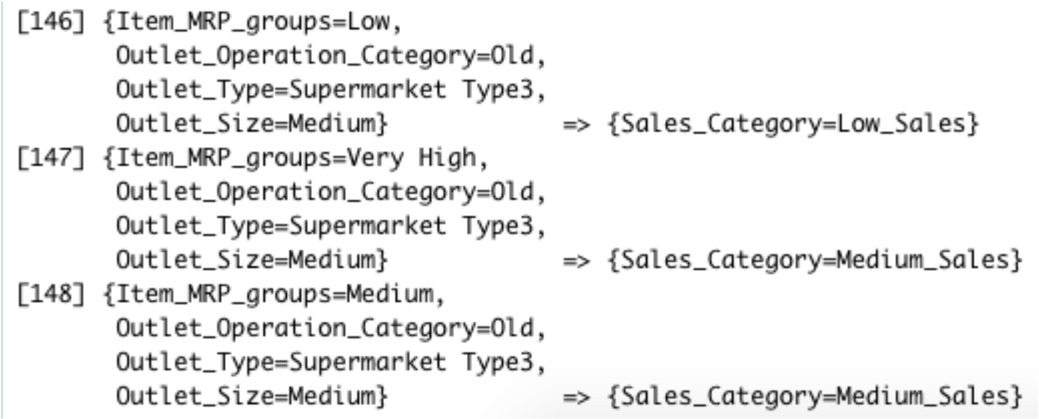
3.9 Association Rule Mining

Association Rule Mining is a procedure through which correlations between items/features can be discovered. As for the data used in this experiment, applying ARM will help in finding features that are correlated or that occur together. After previous steps of utilizing the correlation matrix and feature selection methods, some rules generated from the data can support part of the results which indicate that MRP and outlet type are important to the target variable item outlet sales.

The rules presented in **Figure 3.9** supports the existence of a positive relationship between MRP and item outlet sales; when MRP is low sales are low, when MRP is medium or higher the sales are medium (Most of the sales are either low or medium).

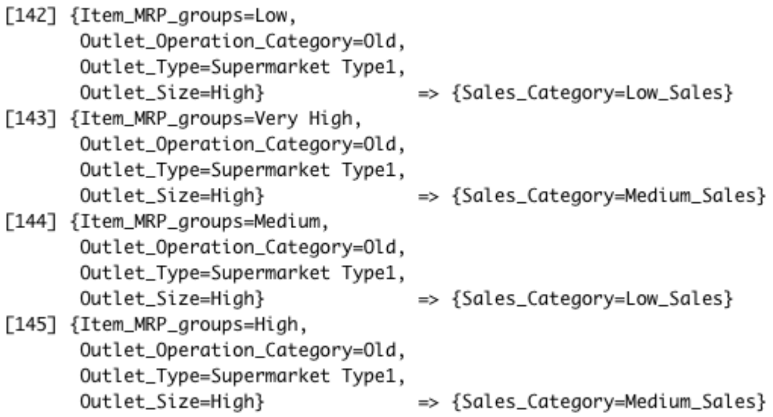
The affect of the supermarket type becomes clear when for instance MRP and sales of supermarket type 1 and type 3 are compared. It is observed that medium MRP in supermarket type 1 is correlated with low sales while medium MRP in supermarket type 3 is correlated with higher sales, **Figure 3.10**.

These findings are consistent with the results found in **Figure 3.11**. The table shows that total sales of supermarket type 3 are the highest compared to all other types of outlet.



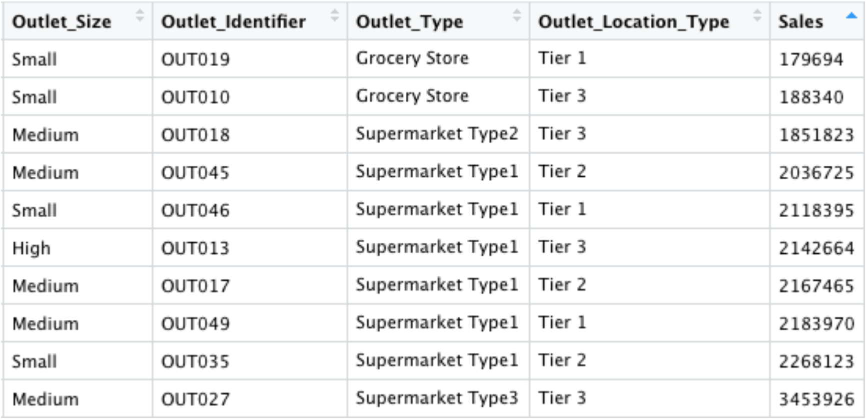
```
[146] {Item_MRP_groups=Low,
       Outlet_Operation_Category=Old,
       Outlet_Type=Supermarket Type3,
       Outlet_Size=Medium}          => {Sales_Category=Low_Sales}
[147] {Item_MRP_groups=Very High,
       Outlet_Operation_Category=Old,
       Outlet_Type=Supermarket Type3,
       Outlet_Size=Medium}          => {Sales_Category=Medium_Sales}
[148] {Item_MRP_groups=Medium,
       Outlet_Operation_Category=Old,
       Outlet_Type=Supermarket Type3,
       Outlet_Size=Medium}          => {Sales_Category=Medium_Sales}
```

Figure 3.9 Association rules – MRP



```
[142] {Item_MRP_groups=Low,
       Outlet_Operation_Category=Old,
       Outlet_Type=Supermarket Type1,
       Outlet_Size=High}            => {Sales_Category=Low_Sales}
[143] {Item_MRP_groups=Very High,
       Outlet_Operation_Category=Old,
       Outlet_Type=Supermarket Type1,
       Outlet_Size=High}            => {Sales_Category=Medium_Sales}
[144] {Item_MRP_groups=Medium,
       Outlet_Operation_Category=Old,
       Outlet_Type=Supermarket Type1,
       Outlet_Size=High}            => {Sales_Category=Low_Sales}
[145] {Item_MRP_groups=High,
       Outlet_Operation_Category=Old,
       Outlet_Type=Supermarket Type1,
       Outlet_Size=High}            => {Sales_Category=Medium_Sales}
```

Figure 3.10 Association rule mining – outlet type



Outlet_Size	Outlet_Identifier	Outlet_Type	Outlet_Location_Type	Sales
Small	OUT019	Grocery Store	Tier 1	179694
Small	OUT010	Grocery Store	Tier 3	188340
Medium	OUT018	Supermarket Type2	Tier 3	1851823
Medium	OUT045	Supermarket Type1	Tier 2	2036725
Small	OUT046	Supermarket Type1	Tier 1	2118395
High	OUT013	Supermarket Type1	Tier 3	2142664
Medium	OUT017	Supermarket Type1	Tier 2	2167465
Medium	OUT049	Supermarket Type1	Tier 1	2183970
Small	OUT035	Supermarket Type1	Tier 2	2268123
Medium	OUT027	Supermarket Type3	Tier 3	3453926

Figure 3.11 Sales ranking according to the outlets

3.10 Splitting the Data

A common approach in machine learning before start building any models is to split the original data set into three smaller sets usually called: training, validation and testing data sets. Training data is the set used to generate a model, where valida- tion and test data are used to evaluate how the model fits to new data set, in an unbiased way after the model has already ”learned” from training data and is ready to perform on unseen data. The splitting of the data was chosen to be of the follow- ing percentages of 0.80, 0.10, and 0.10 for the training, validation, and testing sets respectively. This ratio of splitting was chosen so that more samples are added to the training process as the data set is not big enough which could lower the training process efficiency. Another approach would be to split the data to mere training and testing. However, The validation phase is used to tune hyper-parameters so to enhance the model and thus when the algorithm is applied on testing it will be the final model and not further improved.

3.11 Building Predictive Models

3.11.1 Decision Trees

Decision Trees are supervised machine learning algorithms that have the advantage of being full compatible with all data types of which other algorithms lack and that makes it possible to use them without applying one-hot encoding to categorical variables. In this experiment, regression trees are utilized as the response variable is continuous. Decision trees build a regression model in the structure of a tree, and that is what makes them easy to understand and implement. The approach by which decision trees work is that they break down the complete data set represented in the root node of the tree into smaller and more homogeneous sets while at the same time building a decision tree in an incremental manner. Each internal node of a tree is a donation for a test on one feature, each branch represents an outcome of a test, and each leaf in the tree, or can be called as a terminal node, holds a decision which is in the regression case the average response of the observations.

The biggest disadvantage of this algorithm is building complex trees that can not be generalized; over fitting. They also can be unstable if small variations in the data is presented.

Sticking with default values set by the algorithm may not be the best choice, and that is why hyper-parameter optimization is performed to improve the performance. There are nine parameters that are available for tuning but not all of them actually effects the results. Among these parameters is the cost complexity parameter which controls the growth of the tress by defining the minimal improvement needed at each split. setting cost complexity is tricky because very high values make the criteria for splitting nodes harder and thus less splits are performed, while too low values may have the ability to produce accurate models but could possibly suffer from over fitting. Criteria for splitting nodes is different in classification and regression trees. For regression trees best split is where minimum sum of squares is achieved in each partition. This is called a top-down greedy strategy as it makes best to choose the best split but does not guarantee an optimal complete decision tree, and the top- down notion indicated that process starts with the whole data and continue making splits going down to individual data points. In addition to cost complexity there is the max depth of the tree which also controls the growth of tree by limiting its size. Another two hyper-parameters are minimum split (min split), which defines the minimum number of observations to be accepted at each split task, and minimum bucket (min bucket), which indicates the minimum number of observations in a terminal node.

There are more than one algorithm that can be used to implement decision trees. In this experiment CART algorithm (stands for Classification and Regression Trees) is implemented using R implementation of the algorithm that is called rpart method that involves recursive partitioning of trees. It can be implemented in many ways; the first one is using rpart package itself with assigning the method argument to ”anova” indicates a regression task. Although the algorithm would make an intelligent guess of the task type depending on the outcome variable, it is always recommended to explicitly define the method. The other way of implementation is by utilizing caret package through the general function train and assigning the method to rpart, but the ultimate results will not be significantly different.

3.11.2 Random Forest

Random Forest is considered to be an improvement on decision tree algorithm that tries to overcome its weakness. Random forest algorithm is often compared to bagged trees with which random forest shares a common characteristic of adding a randomness to the tree building process. The main difference between the bagged decision trees and random forest is that later does not utilize a greedy strategy to decide on each split, but rather randomly selects from the whole set of variables provided according to a defined rule, which is m = p/3 for a regression tree, where p is number of variables, compared to m = p in the case of bagged decision trees meaning that the split variable at each split is to be searched among the set of p variables. This approach minimizes the correlation problem from which bagged de- cision trees suffer as their decision trees are not completely independent from each other.

One of the common features between bagging and random forest is using Out-Of- Bag (OOB) samples to provide a good estimate of the accuracy of the model, but it is not a good practice for this study to use OOB samples as it has an objective of comparing models to each other where it is better to test the performance on the same validation set.

There are a number of hyper-parameters that can be tuned in a random forest algorithm, and they could slightly differ from one package to another. The most important hyper-parameter to optimize which have the biggest affect on the perfor- mance is mtry and is common between all packages, which describes, as mentioned above, the candidate set of variables at each split. The number of trees to is more required for the purpose of variable importance estimates and it would stop improv- ing the model performance as it reaches a ”big enough” number, and that is why in most packages it is not a parameter to be tuned because the focus is more on hyper-parameters that would perform better or worse if they set too low or too high. Caret package, which stands for classification and regression training, is the most popular software implementation that does not implement the random forest algo- rithm itself but instead an interface to the algorithm. This interface can be identified by assigning the method argument to ”ranger”, a faster implementation of random forest and a more suitable one for high dimensional data sets, and it was used to build the model. Ranger stands for random forest generator. Using caret, only mtry parameter can be tuned, as be default ”train” function crosses three options of mtry. However, by defining a tune grid additional hyper-parameters such as minimal node size which sets basically the depth of trees, maximum number of nodes, and split rule that is defined as ”variance” in most regression problems. There are a number of more hyper-parameters to be tuned, such as the maximum number of terminal nodes (maxnode), which also somehow a controller of the tree complexity as the more nodes equated to deeper trees.

3.11.3 Extreme Gradient Boosting (Xgboost)

Three concepts are involved in Xgboost algorithm; extreme, gradient, and boosting. Starting from basics boosting is one of the systematic ensemble methods aims to converting weak learners (regression trees in this case as this is a tree-based Xgboost model; there is also a linear type) into stronger learners in order to obtain more ac- curate predictions. Unlike bagging algorithms such as random forest where trees are independently built, boosting builds them in a sequential fashion so each model can learn from the ”mistakes” made by the previous models. Through boosting, the model learns from all training data points that have incorrect output produced by a previous model, these are called the ”hard” examples. What distinguishes gradient boosting from other boosting algorithms, such as Adaboost, is the way in which it identifies hard examples, which is by calculating large residuals computed in the previous iterations, and then fitting the newly built model to the errors of the previous model. Finally the ”Extreme” term simply refers to the computational efficiency provided by the model compared to other variations. The main advantage of xgboost algorithm over gradient boosting lies in its provision for regularization As with other models developed in the previous steps, there are a number of hyper- parameters to be tuned for an xgboost model. At first comes number of trees (also called as number of rounds) is handled. A good approach when specifying this pa- rameter is to set a large enough number, lets say 10000 trees, with also specifying another parameter called early stopping rounds, which as the name suggest will stop growing the trees if the model stopped improving in the last 50 rounds. Another hyper-parameter related with the regularization feature of this algorithm, which imposes constraints on the learning process so more complex models are not built to avoid overfitting, is called the lambda parameter, also known as the shrinkage parameter and is called eta within the model, simply a learning rate identifier that takes values between 0 and 1. Next is the maximum depth of the tree and is by default set to six but is also tuned by building a grid space to search for the optimal value of max depth. Reduction in the loss function is controlled using gammma pa- rameter which is difficult to identify its best value as it can be any number from 1 to infinity being very dependent on the rest set of parameters used, but values around 20 are extremely high and can cause overfitting and are used in special cases, and it is recommended to start with a gamma value of 5. Ratios of samples and columns to be used when constructing new trees are also specified and are both having a default value of 1. All mentioned parameters are called Booster Parameters, meaning that they are related to the booster chosen for the model, trees in this case, and that they would be different if the booster is of a linear nature.

Other parameters are related to the learning task itself such as the objective which set as ”reg:linear”. The type of Booster itself is a general parameter that needs to be set at the beginning as gbtree or gblinear.

The model was implemented through caret’s advanced interface using the general train function which enables utilizing a grid search for parameters tuning which is more time effective than implementing xgboost package and initiating for loops for the possible values of parameters (eta, gamma, etc...) and testing them against each configuration.

3.11.4 Support Vector Regression (SVR)

SVM Regression or simply Support Vector Regression is one of the margin maximization algorithms; it tries to find the best hyper plane with the largest margin; biggest distance between boundary/separation line and closest data points to it. Data points that lie in the margin of separation are called support vectors from where the algorithm takes its name. It also depends on epsilon-intensive loss func- tion, and thus unlike other algorithms where all data points in the training set are considered, SVR focuses on those points that are epsilon-deviated from the hyper plane. The more data points in the hyper plane the better the model the more it is robust to outliers and the better can be generalized. According to the way the algorithm works the ultimate objective is not to minimize the error as in simple linear regression but to try to guarantee that errors do not exceed the threshold. Utilizing kernel trick is essential for this algorithm with the objective of transform- ing non linear separable data set without blowing up dimensionality that happens when adding polynomial terms such as square and cubic in a high dimensional data. This allows to support linear and non linear regression because each training data point is representing it own dimension. In other words, when the kernel transforms data into a higher dimensional feature it allows for performing a linear separation. There are four types of kernels available in the package; Linear, Polynomial, Radial- based (Radial Based Function RBF, or Gaussian kernel), and Sigmoid kernels. In the implementation of this experiment, a radial-based kernel was chosen because it is often recommended when there is a non linear relationship between independent and dependent variables and it is also advised as a default choice for regression problems.

The algorithm was implemented using e1017 package, an interface to libsvm the fast implementation of support vector machine algorithm, which allows for grid search with cross validation. The model is built using svm function that will ”understand” from the type of the output variable that it is a regression problem. Tuning a sup- port vector machine model, given a Radial-based kernel, is basically done by trying different values of its two important hyper-parameters; epsilon and cost. Starting with epsilon, which defines the width of the hyper plane, and it represents the mar- gin of tolerance for the model, and it takes a default value of 0.1. In addition there is the cost tuning parameter that helps in avoiding over-fitting and which can be described as the weight of penalty of making errors and its default value is 1.

These were the four predictive models used for the sales prediction problem. Decision trees, random forest, and extreme gradient boosting are similar in that they are tree-based algorithms although differ in their degree of complexity and interpretability as random forest and xgboost in their essence involve conventional ensembling methods; bagging and boosting. In the following section the implemen- tation is continued describing ensembling of multiple but different models.

3.11.5 Ensemble Modeling Technique

Stacking is the method of ensemble learning chosen to combine the models devel- oped for the prediction task. The first step towards building the stacking ensemble is to define a list of the base learners that are to be combined. These learners are already defined to be the same four algorithms used in this experiment. Next step is to specify a meta learner (the combiner model). A simple generalized linear model is selected as the meta learner. Training the ensemble goes through the following steps:

Each model in the list of base learners (L) is trained on the training data set.

1. A re-sampling technique is performed, which is a k-fold cross validation, on these base learners.
2. The N cross-validated predicted values from base learners are combined form- ing a new matrix of N x L. (N = number of rows in the training data set). The new matrix with the original target vector is called the ”level-one” data.
3. The meta learner is trained on the level-one data. Finally, the resulted ensem- ble model consists of the L base learners and the meta learner, can then be used to make predictions on the validation and testing data sets.

Ensemble learning was implemented using caretEnsemble package that have three main functions. The first function aims to create a list of models to be trained on the same training data. The other two functions are used to build ensembles based on the models that formed the list with the first function. caretStack in particular is the function which builds an ensemble according to the same way described above, by utilizing another model (the meta learner) in order to combine the resulted predictions from the caret list created.

Chapter 4: Performance Evaluation

4.1 Performance Evaluation

Predictive models evaluation is a crucial integral part of the modeling process; it tells to which extent this model can be trusted for predicting future outcomes. For this experiment, evaluation the performance of models developed in the previous stages is the last step of the whole predictive modeling process through which mod- els are analyzed so as to identify how accurate the results are and the percentage of error produced. Performance evaluation metrics calculate what can be called as ”scores” for each model to show how it performed on the testing and validation (”unseen data”) after iterations of training (”learning”) on the training samples. In other words, testing how the model would perform on future data after looking at the historical ones. Of course a must step towards this process is to have testing and/or validation samples because evaluating on the training would be misleading as it could be gamed by the model by memorizing (”storing”) the results and achieve good results while also avoiding the two common problems of overfitting and under fitting. Here comes the difference between the two errors; one is the training error that is produced due to fitting the model to the training dataset, where test error when model is fitted to the test or validation dataset. It is normal that the train error is always smaller than test error. Training and testing error can be also referred to as in sample and out of sample error respectively.

Over-fitting is when the algorithm using which the model is being trained cap- tures all of the noise in the training data and thus the model over-fits these data points to an extent which makes it fail to predict new data points; it loses the gener- alization property. On the other hand, under fitting is the opposite case where the model fails to capture the noise or pattern in the training data and it does not fit the training data. One of the reasons why a model would be an under fitting one is because it is too simple where necessary data preprocessing steps, such as missing data and outliers handling, are not done.

4.2 Resampling

Re-sampling process aims to enhancing the learning process of the model. As the objective of predictive modeling is to apply one algorithm to the training data and then test the model on unseen data, resampling helps by estimating how the model would perform on that new data without actually using it; referred to as test error estimation.

This procedure involves refitting the model on different samples within the exact training data.

One of the well-known resampling method is called Cross Validation and especially its most common version K-fold Cross Validation. In this type of resampling the total number of observation is divided into k number of folds, usually k equals to 5 or 10. In each iteration one fold is sided away to act as the testing set and the model is trained on the other k-1 folds. This process is repeated until each fold acts as the test set once. Finally, the average test error of the k folds is calculated and presented as the estimated test error. Cross validation is often applied when tuning the hyper-parameters of the models in order to avoid overfitting problem.

4.3 Model Tuning

One common approach followed as an attempt to enhance models performance was tuning hyper-parameter to select the most effective model parameters for the pre- diction task. The algorithm of model tuning process can be described as follows and it explains how cross validation is incorporated in the model building process:

1. Before starting the process data is split into training, validation and testing sets.
2. Next step is to define different combinations of possible values of hyper-parameters to be tested.
3. A for loop is initiated and the next steps will be repeated for each set of parameters.
   3.A Start another for loop for each cross validation split.
   3.B Fit the model on the remaining k folds that are not used for validation.
   3.C Model is tested on the validation fold.
   3.D Evaluate the model performance and this is the end of the inner for loop for the first split of cross validation.
   3.EThe average performance is calculated for the first split in the cross validation process and this is the end of the outer fop loop for the first set of parameters.

4. Now that the for loop is finished, the best set of parameters is chosen to fit the final model to the training data and then evaluate the performance of that final model.

4.4 Metrics

As predictive modeling handles two types of problems; classification and regression, there exist different metrics for each of them. The major difference between the evaluation of the two categories is that in regression problems there are continuous variables with which the model is dealing, and they can be used to calculate the error between actual values and the predicted ones. However, in classification problems there are correctly and incorrectly classified outputs which are used to compare the predicted to the actual values.

With classification problems there are a number of metrics and among them the four most common metrics are: Classification Accuracy, Confusion Matrix, Area Under Curve (AUC), and Log Loss. However, there are a number of other eval- uation metrics used for regression models. It is important to use more than one metric because one number can not be relied upon for deciding on the best model and for comparisons between more than one algorithms or a more than one version of the same algorithm. This is basically because one metric can compensate for the limitation of the others.

The first question that one of the metrics provides an answer for is how well the model fits data? The question can be answered using R-squared or coefficient of determination referred to as R2. This metric is defined as the percentage of the response variable variation explained by a regression model. This variation is also called the Sum of Squares and it describes how data points are deviated from the mean which is itself a measurement of central tendency. R-squared always takes a value between 0 and 1 and the closer the value to 1 the better the model because it means that more variance is explained by the model.

Residuals play an effective role in model assessment and they can be calculated for every single data point. And generally if there is a small number of residuals it can be said that the model is good at predicting data and vice versa. But this approach is not practical and instead residuals are condensed into a value repre- sented by a single metric. The second metric which is widely used in forecasting and regression context is Root Mean Square Error (RMSE). RMSE basically measures the standard deviation of the points that are far from the regression line, or residuals (i.e. prediction errors). According to the definition this metric answers the question of how closely are the actual values to the predicted ones? The smaller the value of RMSE the better the model fits the data. However, there does not exist a pre-defined threshold or value to decide on whether a particular RMSE value is good or not as this to be evaluated based on the context of the problem, data volume, and many other factors.
Third metric is Mean Absolute Error (MAE) which is considered as one of the simplest, most intuitive, ,and easily interpretable metrics out there to evaluate models. In essence,

MAE is a description of residuals magnitude so it is as simple as the absolute difference between actual values and residuals. Again, a lower values of MAE indicated a better model and the larger it gets the weaker the performance of the model is.

The forth and last metric used for this experiment is Mean Absolute Percent- age Error (MAPE) which is a very common metric to use in forecasting problems, and is also considered as a weighted version of the previous metric MAE. MAPE is calculated by dividing the absolute error by the actual value.

4.5 Experiment Plan

Studying the effect of every single factor or adjustment contributing to the perfor- mance of the model can be done in so many different ways. In addition, by not setting strict rules to the experiment through which predictive modeling is investi- gated in this thesis would led to a mess in the results. For the mentioned reasons and for the sake of clarity in displaying the results and the comparison of models, an experiment plan was developed as explained below:

1. The Basic models Step: In this step, the four models of the four chosen algorithms are developed with their default setting and withe the full set of predictors as inputs to them, without performing any hyper-parameter opti- mization. developing the models is followed by building an ensemble of them, again in their basic form without intervening into the learning process.

2. The Tuning Step: The second version of the models to be tested is the tuned models. The aim is to measure the affect of tuning models' hyper- parameters on their performance. Each model will have its most important hyper-parameter tuned as described in the implementation phase chapter. An ensemble of the tuned model is then built.

3. The Feature Selection Step: By now all of the models were developed with the full set of predictors fed to them (28 features), but as discussed be- fore, sometimes it could be of a better approach to build the model with a selected subset of features. The most common feature selection methods were explained in the previous chapter. However, the approach to feature selection followed in this thesis work is as follows:

- Filter methods: There is a number of filter methods that can be used for feature selection. Among them are information gain, which is also called entropy, gain ratio, and symmetric uncertainty were chosen. These meth- ods are called mutual information because they measure how important the inclusion of exclusion of a feature is. All of them find weights of the features basing on their correlation with continuous class. Based on the ranking given by each algorithms of the three, the top ten features are selected to form a list.

- Wrapper methods: Forward selection differs from backward elimination in that forward selection starts with an empty model and adds features that improve the model performance iteratively until the addition of a new feature is not improving the performance any more. While backward elimination start with a model containing the complete set of predictors then remove a non-significant feature in each iteration to improve the model performance. Implementing backward elimination resulted in a sub set of 25 features, which considered as a bigger than needed set of predictors, and this list were not used. Instead, forward selection resulted in a list of nine predictors ranked as important with respect to the response variable. On the other hand, re cursive feature elimination was applied to produce a list of selected features. This method is a greedy optimization algorithm used to find the best subset of feature through creating repeatedly models and recording the best or worst performing feature at each iteration. Then, the following model is constructed with the left features until all the features are exhausted. Finally, it ranks the features based on the order of their elimination. Again, ten of the best performing methods were chosen and added to the list.

- Embedded methods: Random forest is chosen to decide on the best performing features, that is the ones best contributing to the accuracy of the model, while the model is being developed. The final list of the ten most im- portant features is also created.

Each of the applied methods resulted in a list of important features. In all cases there were ten features but with forward selection (function sequential forward selection from mlr package) where it produced a list of 9 features. There were some common features in each list and some ones that are unique to the method. The union of all this features are listed in a final subset containing sixteen features. After this, the intersection is also recorded, that is the features that are common between all of the algorithms and there were five features. Now that we have the union and intersection as two different subsets, models are developed again using the selected sub of features.

4.6 Making predictions

The next step in the methodology followed after developing the predictive models based on the different iteration stated in the experiment plan is to use these models by making the predictions. Using the different versions of developed models values of Item Outlet Sales are predicted. Each prediction function produces a vector of the predicted sales using the particular model. For the prediction process, the first argument to the function is the fitted model object and the second argument is the set of predictor variables to be used for predicting the sales. In this thesis work the original train dataset was splitted further into a validation set and a testing set. This means for each model there is two different predicted values; one with the predictors of the validation set and another with the predictors of testing test. This explains why later on each performance evaluation metric has two values for a given model.

Chapter 5: Results and Conclusion

5.1 Results and Conclusion

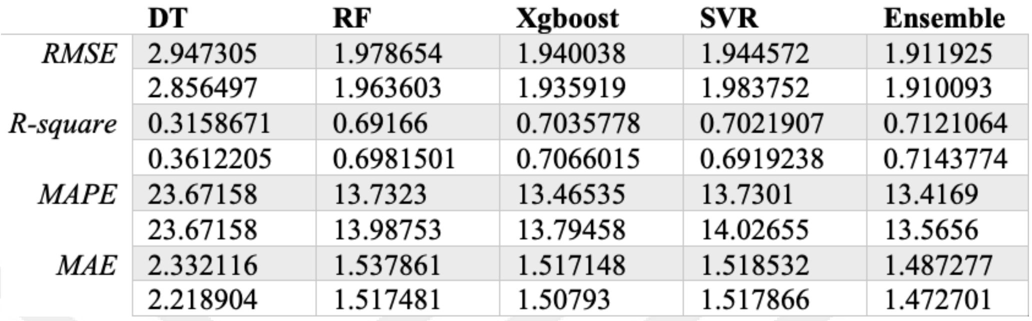
The last step in the methodology followed was the performance evaluation and comparison between the different configuration of models. In this step, after the models were developed and the prediction were made, summaries and conclusions are to be presented in this chapter in order to identify the best and worst performing models.

The results are presented according to the experiment plan described in the pre- vious chapter. Thus, the resulting measurements of each models are presented in four separate tables where each table represents conditions under which the models were developed. Each Table below shows the values associated with each perfor- mance evaluation metric (RMSE, R-square, MAPE, and MAE) for each one of the four models (Decision Tree model, Random Forest model, Extreme Gradient Boost- ing model, Support Vector Machine Regression model) and the ensemble model. Each metric has two lines showing the results of validation and testing sets.

Table 5.1 shows the result and performance of the models with their default hyper-parameter and the basic configuration. From RMSE perspective, Xgboost algorithm is associated with the lowest score of 1.935, while slightly higher values with the two models RF and SVR. In respect with R2 scores, again Xgboost best performed with a vale of 0.706 that is slightly better than SVR's value of 0.702, but both of the algorithms shows very close values. RF performance is slightly weaker with a value of 0.698. MAPE and MAE values are consistent with the ratio of error for all of the models; ranging between 13.456 percent for Xgboost to 14.026 percent for SVR model, and again the ensemble model has the lowest percentage of error of 13.416.

Decision trees are out of the competition with this configuration; the results shows very weak fitting model and a high values of errors. Finally, the ensemble model successfully enhanced the R2 score and recorded a value of 0.714 and therefor decreased the RMSE score to 1.910.

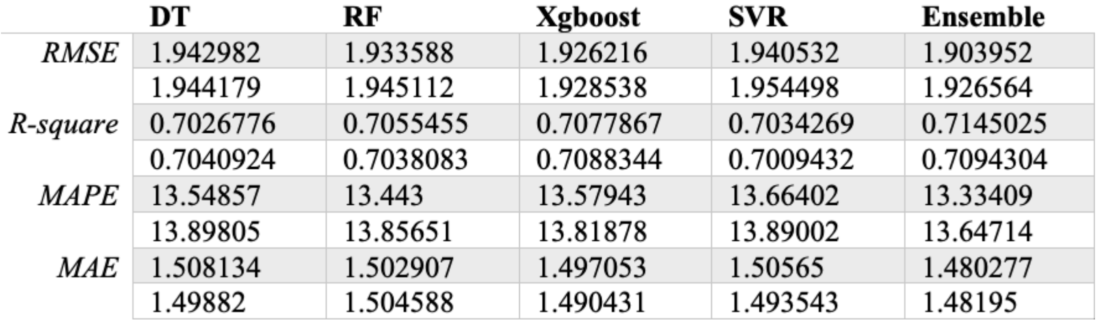
Table 5.1 Models performance – full set of feature – tuning not applied



	DT	RF	Xgboost	SVR	Ensemble
RMSE	2.947305	1.978654	1.940038	1.944572	1.911925
	2.856497	1.963603	1.935919	1.983752	1.910093
R-square	0.3158671	0.69166	0.7035778	0.7021907	0.7121064
	0.3612205	0.6981501	0.7066015	0.6919238	0.7143774
MAPE	23.67158	13.7323	13.46535	13.7301	13.4169
	23.67158	13.98753	13.79458	14.02655	13.5656
MAE	2.332116	1.537861	1.517148	1.518532	1.487277
	2.218904	1.517481	1.50793	1.517866	1.472701

Performing hyper-parameters tuning had a slightly good impact on decision trees algorithm in particular as it recorded values for the metrics that fall into the same range of the other algorithms. On the other hand, hyper-parameter optimization has also improved the performance of the remaining three algorithms but to a very small extent compared to decision trees. With performing tuning optimization, the algorithms performed at the same level with an R2 square of around 0.70, and the ensemble model successfully outperformed them and recorded a value of 0.71, **Table 5.2**. However, in other cases (i.e different types of data sets), applying hyper-parameter tuning should result in more remarkable imrpovments.

Table 5.2 Models performance – full set of features – tuning applied

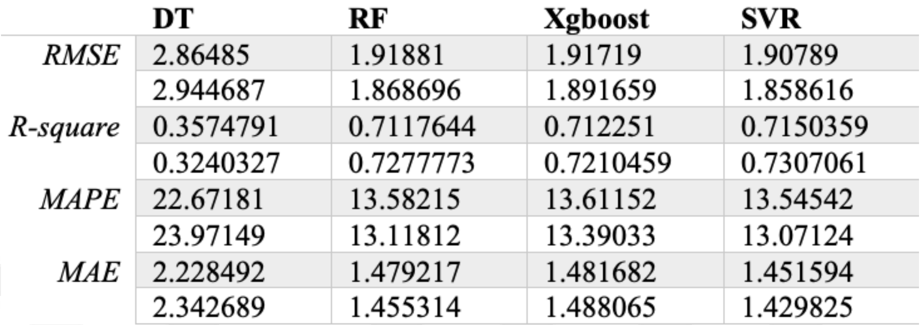


	DT	RF	Xgboost	SVR	Ensemble
RMSE	1.942982	1.933588	1.926216	1.940532	1.903952
	1.944179	1.945112	1.928538	1.954498	1.926564
R-square	0.7026776	0.7055455	0.7077867	0.7034269	0.7145025
	0.7040924	0.7038083	0.7088344	0.7009432	0.7094304
MAPE	13.54857	13.443	13.57943	13.66402	13.33409
	13.89805	13.85651	13.81878	13.89002	13.64714
MAE	1.508134	1.502907	1.497053	1.50565	1.480277
	1.49882	1.504588	1.490431	1.493543	1.48195

Table 5.3 presents the results associated with models developments using the subset of features selected (union approach; using 16 features). Looking at the R2 scores, it can be observed that RF and Xgboost have almost identical performance with recorded values around 0.72, while SVR model performed better with an R2 value of 0.73. Consequently,

RMSE score has been reduced best by SVR model to 1.858, while RF and Xgboost have reduces the score to 1.868 and 1.891 respectively.

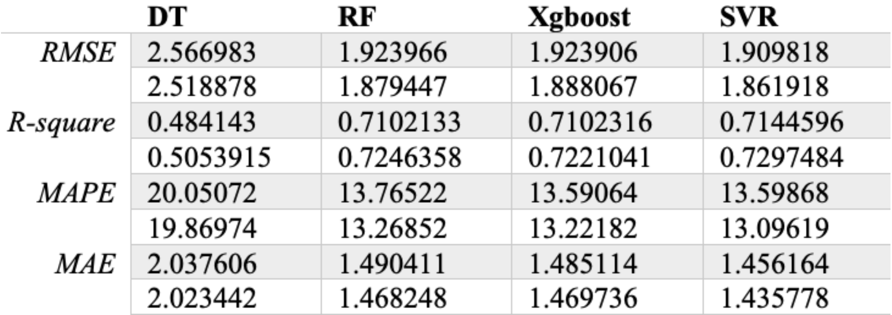
Table 5.3 Models performance – subset of features (16, the union)



	DT	RF	Xgboost	SVR
RMSE	2.86485	1.91881	1.91719	1.90789
	2.944687	1.868696	1.891659	1.858616
R-square	0.3574791	0.7117644	0.712251	0.7150359
	0.3240327	0.7277773	0.7210459	0.7307061
MAPE	22.67181	13.58215	13.61152	13.54542
	23.97149	13.11812	13.39033	13.07124
MAE	2.228492	1.479217	1.481682	1.451594
	2.342689	1.455314	1.488065	1.429825

Finally, **Table 5.4** presents the results associated with models developments using the subset of features selected (intersection approach; using only five features). The first observation is regarding DT model that has improved compared to the previous versions of the algorithm itself, while at the same time it is still considered a poor performance with an R2 value of 0.505, and thus reduced error measurements of 2.518, 19.869, 2.023 for RMSE, MAPE, and MAE respectively. Comparing R2 values of the three remaining models to the R2 values in Table 3, we can see that they are very close to each other; around 0.72 for RF and Xgboost and almost 0.73.

Table 5.4 Models performance – subset of features (5, the intersection)



	DT	RF	Xgboost	SVR
RMSE	2.566983	1.923966	1.923906	1.909818
	2.518878	1.879447	1.888067	1.861918
R-square	0.484143	0.7102133	0.7102316	0.7144596
	0.5053915	0.7246358	0.7221041	0.7297484
MAPE	20.05072	13.76522	13.59064	13.59868
	19.86974	13.26852	13.22182	13.09619
MAE	2.037606	1.490411	1.485114	1.456164
	2.023442	1.468248	1.469736	1.435778

In general, Random Forest, Xgboost, and Support Vector Machine have performed well and resulted in the best fitting on the data. To conclude on the per formance of the algorithms, it is important to combine the evaluation of the performance with an understanding to the data itself. For example, with much larger samples and data sets, the algorithms are able to learn more from bigger training data.

Another important aspect regarding the data itself is the type of information recorded and used as predictors highly affects the performance of algorithms; when sales are the target for a prediction tasks, it would be of great benefit to record some other important features that would have a more direct relationship with the target variable. Those feature are, for example, the number of clients or visitors on a given day. Another feature is whether or not a particular item is on sales or not, or more generally whether or not a discount policy is applied or offers are provided. Sales would absolutely be affected by these feature and thus it worth for a company to record such piece of more informative features.

Based on the results presented on the four tables, the forth one includes the highest R2 values and the most reduced errors, and this has to do with what has been mentioned on the subject of whether or not the set of predictors are infor- mative or not. Models could have achieved better performance using only a subset of features (16 features) out of the original complete set of 28 features. Feature selection impact was better than performing hyper-parameter optimization, which is computationally very expensive, and the effect of ensemble learning, and also the affect of two of them combined (the last column of Table 2). Not only limited to the union approach for feature selection but also the intersection approach has resulted in a very close results to the union approach.

The results of feature selection achieved an improvement of 3(%) in the R2 value compared to the first configuration of the models, and by 2(%) compared to the tuned models. In other words, we could predict sales with a better level of accuracy using only five features.

Chapter 6: REFERENCES

A random forest method for real-time price forecasting in New York electricity market. (2014). IEEE.

A. Lahouar, J. B. (2015). Day-ahead load forecast using random forest and expert input selection. *Energy conversion and Management, 103*.

A.Bastos, J. (2010). Forecasting bank loans loss-given-default. *Journal of Banking & Finance, 34*.

Benjamin M. Van Doren, K. G. (2018). A continental system for forecasting bird migration. *Science, 361*.

Caroline Perssona, P. B. (2017). Multi-site solar power forecasting using gradient boosted regression trees. *Solar Energy, 150*.

Chi-Jie Lu, T.-S. L.-C. (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems, 47*.

Cliona Ni Mhurchu, S. O. (2007). The price of healthy eating: cost and nutrient value of selected regular and healthier supermarket foods in New Zealand. *THE NEW ZEALAND MEDICAL JOURNAL, 120*.

Crude oil price forecasting using XGBoost}. (2017). IEEE.

Data mining for short-term load forecasting. (2002). IEEE.

G. Santamaria-Bonfil, A. R.-B. (2016). Wind speed forecasting for wind farms: A method based on support vector regression. *Renewable Energy, 58*.

Halil IbrahimErdal, O. K. (2013). Advancing monthly streamflow prediction accuracy of CART models using ensemble learning paradigms. *Journal of Hydrology, 477*.

Hyun-Chul Kim, S. P.-M. (2003). Constructing support vector machine ensemble. *Pattern Recognition, 36*.

João Mendes Moreira, A. M. (2012). Ensemble Approaches for Regression: A Survey. *ACM Computing Surveys*.

John G. Wacker, R. R. (2002). Sales forecasting for strategic resource planning. *International Journal of Operations & Production Management, 22*.

Josef Kittler, F. R. (Ed.). (2000). Ensemble Methods in Machine Learning. Springer-Verlag London, UK ©2000.

Justin Heinermann, O. K. (2016). Machine learning ensembles for wind power prediction. *Renewable Energy, 89*.

Kenton B. Walker, L. A. (1991). Management Forecasts and Statistical Prediction Model Forecasts in Corporate Budgeting. *Journal of Accounting Research, 29*.

Kuan-Yu Chen, C.-H. W. (2007). Support vector regression with genetic algorithms in forecasting tourism demand. *Tourism Management, 28*.

L.Benali, G. F. (2019). Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components. *Renewable Energy, 132*.

Libbrecht, M. W. (2015). Machine learning in genetics and genomics. *Nature Reviews Genetics, 16*.

Lu, C.-J. (2014). Sales forecasting of computer products based on variable selection scheme and support vector regression. *Elsevier journal, 128*.

Maciej Zięba, S. K. (2016). Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems with Applications, 58*.

New Ensemble Machine Learning Method for Classification and Prediction on Gene Expression Data. (2006). *IEEE.*

Pao-Shan Yu, S.-T. C.-F. (2006). Support vector regression for real-time flood stage forecasting. *Journal of Hydrology, 328*.

Prasun Das, S. C. (2007). Prediction of retail sales of footwear using feedforward and recurrent neural networks. *Neural Computing and Applications, 17*.

S. Ferlito, G. A. (2017). Comparative analysis of data-driven methods online and offline trained to the forecasting of grid-connected photovoltaic plant production. *Applied Energy, 205*.

Sonal S. Pandya, R. V. (2016). French Roast: Consumer Response to International Conflict—Evidence from Supermarket Scanner Data. *THE MIT PRESS JOURNALS, 98*.

SongLi, L. G. (2016). An ensemble approach for short-term load forecasting by extreme learning machine. *Applied Energy, 170*.

Tin ST, M. C. (2007). Supermarket sales data: feasibility and applicability in population food and nutrition monitoring. *Nutrition Reviews*.

William Kew, J. B. (2015). Greedy and Linear Ensembles of Machine Learning Methods Outperform Single Approaches for QSPR Regression Problems. *Molecular Informatics, 34*.

Xing Chen, L. H. (2018). EGBMMDA: Extreme Gradient Boosting Machine for MiRNA-Disease Association prediction. *Cell Death & Disease, 9*.

Ye Ren, L. Z. (2016). Ensemble Classification and Regression – RecentDevelopments, Applications and Future Directions. *IEEE Computational Intelligence Magazine, 11*.

YonghongPeng. (2006). A novel ensemble machine learning for robust microarray data classification. *Computers in Biology and Medicine, 36*.

CURRICULUM VITAE

Personal Information

Name Surname	: Judi Sekban
Place and Date of Birth	: 24/11/1994 – Saudi Arabia, Al Madinah

Education

Undergraduate Education	: Information Technology and computing
Graduate Education	:Computer Engineering
Foreign Language Skills	: Arabic, English, Turkish

Work Experience

TRANSLA TOR (FREELANCER)
HAYKAL MEDIA, UAE | JAN 2016 - JUN 2019
convert written (Harvard Business Review articles and researches on different topics) and audible (interviews) materials English-Arabic
PROCUREMENT OFFICER
ALOGAL Y MEDICAL COMP AN Y, KSA | MAR 2016 – AUG 2017

Contact:

Telephone	: 0090 537 296 20 70
E-mail Address	: joudialfattal@gmail.com