Acknowledgements

We are profoundly grateful to **Prof.Dipti.P.Rana** for her expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement.

We would like to express deepest appreciation towards **Dr. S.R.Gandhi**, Director, SVNIT, and Head of Department of Computer Engineering and whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

ABSTRACT

The purpose of this research is to construct a sales prediction model for retail stores using the predictive modelling approach. Using such a model for analysis, an approach to store management could be formulated. Several factors like seasonality of goods, their price points, loyalty of customers, assortment of goods, goods that are bought together, etc. play a major role in providing the competitive advantage. In this report we study the techniques to refine and improve the tools available to the retailers for demand forecasting, stock management, item recommendation systems for retailers, financial indicator of their business, visualizations of the demand variations and building loyalty programs so as to employ historical data to help their growth. The present study uses a years' worth of point-of-sale (POS) data from a retail store to construct a sales prediction model that, given the sales of a previous days, predicts the changes in sales on the following fortnight. This can be done effectively by analyzing the continuous stream of data, hence we use predictive modelling for this solution.

Keywords: Retail, Prediction, Analyze, Stock, Model, Demand, Visualize, Dataset, Patterns.

Contents

List of Figures

Chapter 1

Introduction

The retail business globally is expanding in leaps and bounds. With increasing competition, each retailer needs to correctly cope up with the impending demand. This also means that there is an increasing shift towards optimization and efficiency, and a shift away from excess and waste. One of the most valuable assets a company has today is the data generated by the customers and it has become popular to try and win business benefits from analyzing this data. Using data mining in such an application is common but expensive and needs huge amount of historical data, hence employed by only the top retailers in the world. Using this approach in small scaled industries is our aim, hence the focus would be on data generated by small time retailers and help improve their business strategies, at good scales of analysis.

1.1 Applications

To predict outcomes of a future event a machine learning model is exposed to the historical data, from which, it learns patterns that are used to predict the outcome. Within the data there might be patterns that could be used to guide a company in how to take decisions regarding marketing, organization and sales. Opportunities to better understand the underlying factors which play part in the daily sales can be seen.

The data analyzed in this project consists of the items bought, along with the date, its buying price, quantity bought, and several other related factors. Training and test data sets have been created to meet a certain benchmark accuracy set for the modelling.

There is a need to manage and track a large number of items across various categories, track consumers' shopping habits and above all, maintain a compelling brand and loyalty strategy that makes consumers to keep coming back. A huge section of cross marketing or stock recommendation will also be analyzed for the retailers to refine their business dealings resulting in more profit. Discounts and promotions are also decided after analyzing the profits earned seasonally due to the sale of certain products. Statistical analysis of profit and loss statements and good visualization of variations of purchases based on seasons, brands and trends can help the management decide how to steer their business in future.

1.2 Motivation

Historically these tasks have been based on intuition and current market trends, which often are short termed and frequently varying and not reliable. To enable the businesses to make a highly dependable prediction based on the past experiences as well as the current trends is the main aim of this project. Suggestion of several parameters to consider for stock management for optimized sales would be resulted. Predictive modelling is applied to the data and fetch results based on several rounds of algorithms. Hence enhancing the process of business decisions that can be taken by the manager.

The API and prototype created can be replicated for other data sets and faithful analysis can be done on several data sets of different retail stores with different categories. This report can be used a POC.

1.3 Objectives

Following are the project objectives :

1. Predict stock requirement on a fortnightly basis

2. Recommend products to be bought, in the future based on customer buying trends and global trends
3. Recommend to the customer : cross marketing and latest trend analysis , based on majority of items bought currently (personal level) and items bought by the population around (global analysis)
4. Statistical analysis of profit/loss statements for a Year-on-Year analysis
5. Visualization of :
    a. Variation in weekly purchases
    b. Variation in demand of individual items.
    c. Growth graph of business
    d. Future clustering of items

The data used is thoroughly anonymised for secure usage, and our models do not use any personal information for analysis.

1.4 Organization of Project Report

The further sections of this report are organized as follows. Chapter 2 describes the literature survey reviewed and analysis of different techniques used. It also provides an overview of the different approaches and algorithms used to build the base of models of similar application in research and in the current market. Comparison of the techniques and why the preference of one over the other is mentioned in this chapter. Chapter 3 describes the proposed framework and methodologies used in the project and the techniques employed. Description in detail of the steps in prepossessing data, building the model and analysis reports generated via the models along with the appropriate visualization techniques proposed for the same. Algorithms, application and solution described in a flow chart with theoretical examples of the same.

Chapter 2

Theoretical Background and Literature Survey

Using statistical techniques such as exponential smoothing, ARIMA regression models is required for this application. Newer approaches include the applications of advanced techniques such as neural network and data mining. In order to increase the forecasting performance hybrid models are developed. Hybrid models can be used to take advantages of different models for a new combined approach. These models seemed to be the most accurate in sales forecasting.

The predictive model, will allow retailers to decide the amount of safety stock that would be needed by the retailer to be kept in-house. This data would be generated by a automation script.

2.1 Research and Analysis

To develop a predictive model where the work is done on a continuous value data set various regression models can be used. Kris J, David and Bang Lee suggested a few models such as regression trees,principal components regression, multiplicative (power) regression, etc., to be used in this retail forecasting scenario.

1. **Regression trees**

They are a kind of decision trees that are used where there are too many features which interact in nonlinear ways. These types of data are computationally expensive if global model such as linear regression is used. In regression trees an alternative approach to nonlinear regression is to partition, the space into smaller regions, where the interactions are more manageable. Then the subdivisions are partitioned again, this is called recursive partitioning, until finally chunks of the spaces are resulted which are so tame that we can fit simple models to them.

In simple linear regression, a real-valued dependent variable Y is modeled as a linear function of a real-valued independent variable X plus noise

$$Y = \beta_0 + \beta_1 X \tag{2.1}$$

In multiple regression, there may be multiple independent variables

$$X_1, X_2, ...X_p = X. \tag{2.2}$$

$$Y = \beta_0 + \beta_T X \tag{2.3}$$

This works well as long as the independent variables each has a separate, strictly additive effect on Y, regardless of what the other variables are doing. It's possible to incorporate some kinds of interaction,

$$Y = \beta_0 + \beta_T X + \gamma_X X T \tag{2.4}$$

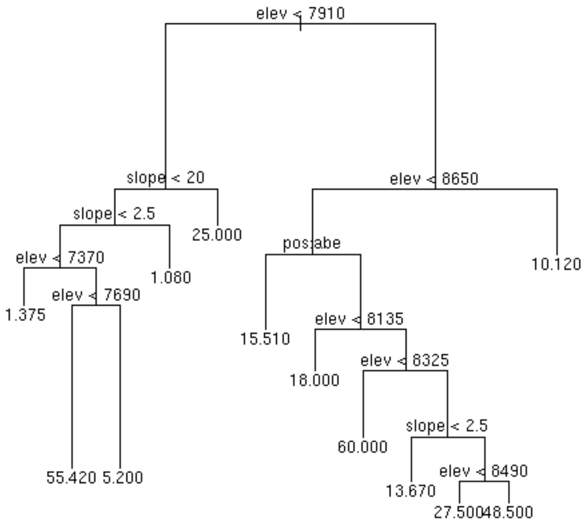


Figure 2.1: Regression Tree Example

2. Principal Component Regression

It is a regression based on principal component analysis, here the principal values are calculated and they are used as predictors in a linear regression model that used the least square procedure. This type of model is used when there are several co-linearity's between the attributes.

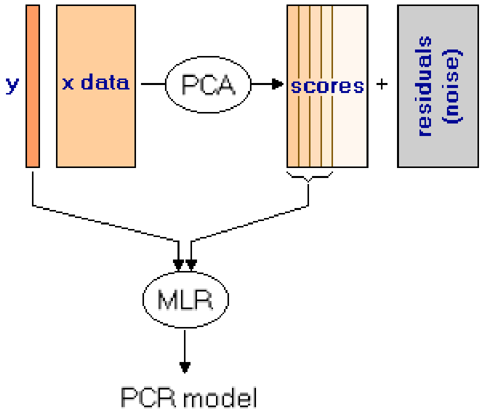


Figure 2.2: PCR Model Description

The Principal component analysis that is used for the model is a dimension-reduction tool that is used to reduce a large set of variables to a small set that still contains most of the information in the large set. PCA uses a linear combination of variables such that the maximum variance is extracted from the variables. It then removes this variance and uses a second linear combination which explains the maximum proportion of the remaining variance, and so on. This is called the principal axis method and results in orthogonal (uncorrelated) factors. PCA analyzes total (common and unique) variance.

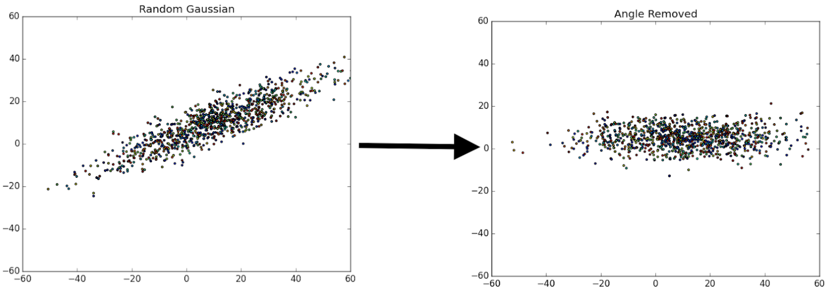


Figure 2.3: Graphical Description of PCA

3. **Multiplicative regression**

It is a type of regression in which the log of the data is taken for the prediction.For tuning the parameters of this model, 5-fold cross-validation on the training data was also used. To better increase the performance metrics evaluated, regression trees with bagging was used for all categories.

The k-fold validation technique is a validation technique machine learning that is used to have a good variation of the test data for the model that is trained to be tested on.

It works in a way that the dataset is broken into k sectors known as k folds and the k-1 percent terms of each sector is training data and the rest is test data.For example, setting k = 2 results in 2-fold cross-validation. In 2-fold cross-validation, we randomly shuffle the dataset into two sets d0 and d1, so that both sets are equal size (this is usually implemented by shuffling the data array and then splitting it in two). We then train on d0 and validate on d1, followed by training on d1 and validating on d0.

When k = n (the number of observations), the k-fold cross-validation is exactly the leave-one-out cross-validation.

The bagging technique is one of the most refined techniques that is a part of the ensemble learning. It breaks the data in such a way that most of the each part is trained by different models and then the aggregate of these models are taken, thus giving the best use properties by each model.

Using techniques like boosting and bagging has lead to increased robustness of statistical models and decreased variance.

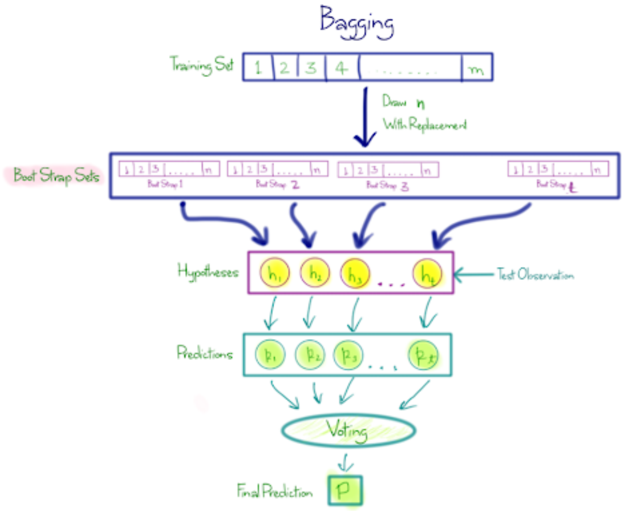


Figure 2.4: Bagging Technique

Another Paper by Xia, M., Wong, suggested that in order to deal with seasonality and limited data problems of retail products, a seasonal discrete grey forecasting model is introduced.

The authors suggested pre-processing the time series by using ANN model or fuzzy grey regression model.

1. **Artificial Neural Network Model**

It is one of a kind, it is a part of the Deep Learning algorithms and can be used for regression as well as classification, it has hidden layers that lead to its final output, it uses the concept of forward and backward propagation that are used to reduce the loss.

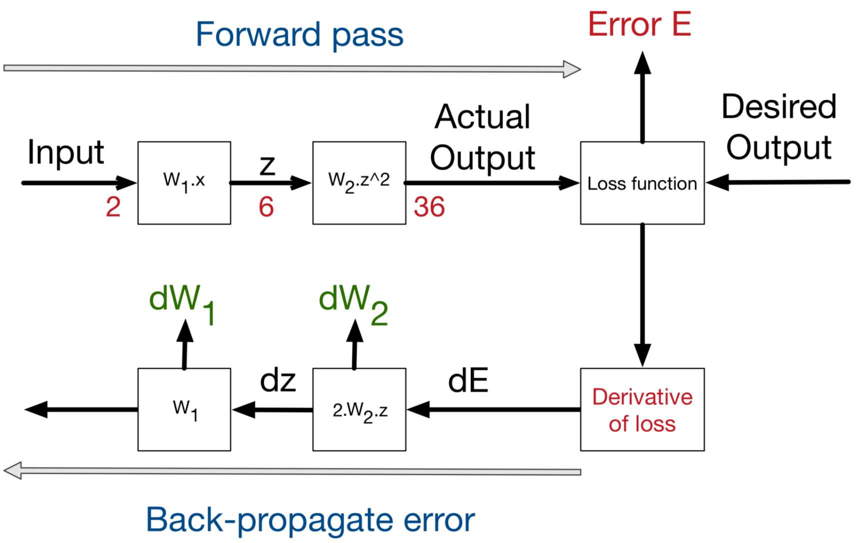


Figure 2.5: ANN Model Description

2. **Fuzzy grey regression model**

It is used for limited time series data such as just having 50 observational time-stamps as the dependent variable. The grey model with first-order differential equation and one dependent variable model is referred to as the grey model GM(1,1) (Deng 1986, 1989) and was introduced in management and engineering applications for solving limited time series data.

Grey model GM(1,1) If an original time series set f0 is defined as

$$f^0 = \{ f_t^0 \big| t \in 1, 2, \ldots, n\} \qquad (1)$$

where t denotes the number of data observed in period t, then the AGO value f 1 t of the original time series f 0 t is obtained as

$$f_t^1 = \left(\sum_{k=1}^{t} f_k^0 \right) \quad t = 1, 2, \ldots, n. \qquad (2)$$

The grey model GM(k, N) (Deng 1986, 1989) is defined as Eq. (3) where k stands for the kth-order derivative of the dependent variables F1 t , and N stands for N variables (i.e. one dependent variable F1 t and N 1 independent variables X1 1(t), X1 2(t), . . . , X1 N1(t)).

$$\begin{aligned}&\frac{d^k F_t^1}{dt^k} + a_1 \frac{d^{k-1} F_t^1}{dt^{k-1}} + \cdots + a_{k-1} \frac{dF_t^1}{dt} + a_k F_t^1 \\ &\quad = b_1 X_1^1(t) + b_2 X_2^1(t) + \cdots + b_{N-1} X_{N-1}^1(t) \qquad (3)\end{aligned}$$

where a1, a2, . . ., ak and b1, b2, . . ., bN1 are unknown parameters. If k = 1 and N = 1, then the grey model GM(1,1) with first-order differential equation and one dependent variable model can be constructed as

$$\frac{dF_t^1}{dt} + aF_t^1 = b, \quad t = 1, 2, \ldots, n \qquad (4)$$

where a represents the unknown developed parameter, b represents the unknown grey controlled parameter, and F1 t is the dependent variable with AGO input value f 1 t . For solving model (4), the derivative d F1 t dt for the dependent variable is represented as

$$\frac{dF_t^1}{dt} = \lim_{h \to 0} \frac{F_{t+h}^1 - F_t^1}{h}, \quad \forall t \geq 1. \qquad (5)$$

2.2 Current Trends

Various trending algorithms that are commonly discussed and implemented in performing similar statement problems are Bagging and Boosting, LGBM's, Xgboost and SVR.

1. **Bagging and Boosting**

Boosting algorithms are some of the most popularly used algorithms, in all data science applications. These algorithms can immediately boost up the accuracy of the models, without much intensive preprocessing required. Many wrappers are readily available to identify the best parameters for fine tuning such models.

'Boosting' refers to a family of algorithms which convert weak learners to strong learners.Weak learners individually are not powerful enough to predict accurate results for prediction. All weak learners are stacked sequentially one after the other. The motivation behind boosting is to combine all weak learners, and to allow continuous learning for each individual weak learner by improving from the mistakes of the previous learner. The data set is divided into folds depending on the number of weak learners. Multiple iterations are applied. Once a weak learner predicts an output for a given fold it creates a weak prediction rule. The errors and mistakes made by this learner are learned by the next weak learner, thus improving the predictive knowledge of each learner, gradually minimizing the loss function.

$$Y = ax + b + e \tag{2.5}$$

Where e=error term.

For every wrong predictive rule that is generated (similar to errors made by the previous weak learner) this learner is heavily penalized. Different boosting algorithms include Adaboost, gradient tree boosting, XGBoost,etc.

2. **LGBM**

Light gradient boosting algorithm is a boosting algorithm which generates a decision by minimizing loss using gradient descent. Light GBM grows tree vertically while other algorithm grows trees horizontally that is level wise.Light GBM grows tree leaf-wise. Leaf-wise algorithm can reduce more loss than a level-wise algorithm.

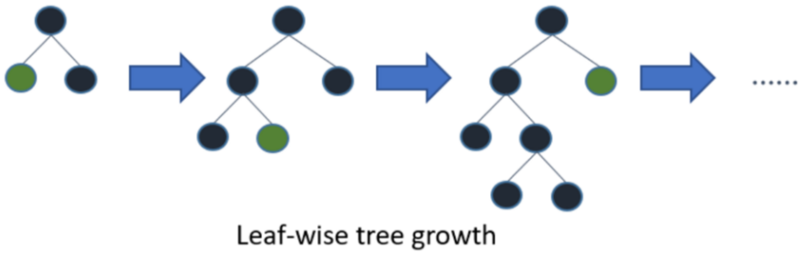


Figure 2.6: Vertical Tree Growth

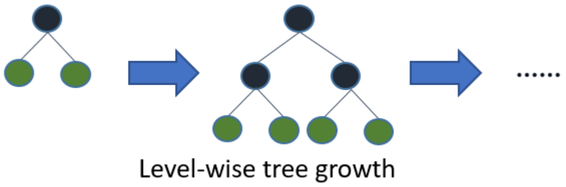


Figure 2.7: Horizontal Tree Growth

LGBM is very versatile, as it can be used for regression, binary classification and multiclass classification. This model can prove extremely useful, as it can handle large sizes of data giving an advantage of requiring less memory to train the model. To work with large amounts of data similar to the data set referred for this project, it is essentially important to be able to train models using GPU. Complying with this requirement specification, LGBM has the specialty of supporting GPU learning instantly. Other reasons for its popularity, is because it grows regression trees leaf wise, in turn helping to reduce the loss of content and focusing on increasing the accuracy of the result.

3. **XGBOOST**

Among-st other, more powerful models and often used models in various Kaggle challenges is XGBOOST. Xgboost allows to exponentially reduce the exponential speed and increase the model performance.
Xgboost is known as a regularized boosting technique which provides inbuilt regularization. Standard GBM implementation has no regularization, hence they cannot solve the problem of over-fitting as efficiently as XGBOOST.

Along with which it can be implemented in single, distributed, parallel and out of core computations. It also supports implementation on Hadoop.

Xgboost uses a greedy algorithm. It can find the best split in the regression tree over continuous features like time series, continuous price, and product quantity. Xgboost stops splitting nodes when it encounters a negative loss in the split. Xgboost is also aware of the sparsity pattern in the data hence only visiting the default direction in each node.

Xgboost has in built cross-validation wrapper which can be used at each iteration of the boosting process. This is used to exact optimum number of boosting iterations in a single run.

4. **SVR**

Support vector regression is supervised regression model which is used to provide accurate results using a margin tolerance (epsilon) threshold. It is very difficult to predict the absolute real number output, due to infinite possibilities. SVR uses various kernels like the Gaussian, RBF and linear kernels. A kernel is a function that measures the similarity between two inputs.

$$Higher - values = more - similarility \tag{2.6}$$

Hence the equation.
The RBF kernel defines similarity to the euclidean distance between the two inputs (similar to nearest neighbor). It is used to calculate the similarity of data points in a multidimensional information space by interpolating the data to a higher dimension, in order to measure the similarities between linearly separable data using the similarity function.

If the two inputs are right on top of each other, they get the maximal similarity of 1. If they are ”too far” away from each other, the RBF kernel just says they aren’t similar (returning a value near 0).

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right)$$

The gamma parameter controls how far away ”too far” is. If you are ”too far” you don’t impact the solution at all. If you aren’t ”too far” your voice still gets heard, even if its not as much as someone who is closer.

$$\gamma = \frac{1}{2\sigma^2}$$

SVR can thus help us determine accurate results precisely by lying in a given range.

5. **Handling time-series data**

There are two types of time-series data sets which are available for usage.

a. **Univariate time series data** - this data set only consists of one independent variable i.e. time and one dependent response variable. This data set is simple, easy to understand, can easily be plotted to compare with the expected results.

b. **Multivariate time series data** – this data set consists of more than one independent variable including time and one dependent response variable. This data set is generally more challenging than univariate data set. A great source of this data can be obtained, from UCI machine learning repository.

Univariate time series data can be easily used without modification to perform time-series forecasting. To predict response variables for multivariate time-series data, the dataset can be converted to a supervised learning algorithm. We can use the shift() function in Pandas to automatically create new framing of time series problems given the desired length of input and output sequences.

This method can be applied to both univariate and multivariate time series data. One of the other popularly used time series models are the:

a. **ARMA model**, AR stands for auto-regression and MA stands for moving average. Given by the formula,

$$x(t) = alpha * x(t–1) + error(t) \tag{2.7}$$

The AR formulation signifies x(t) as the current instance of is solely dependent on the previous instance. The alpha is a coefficient which we seek so as to minimize the error function.

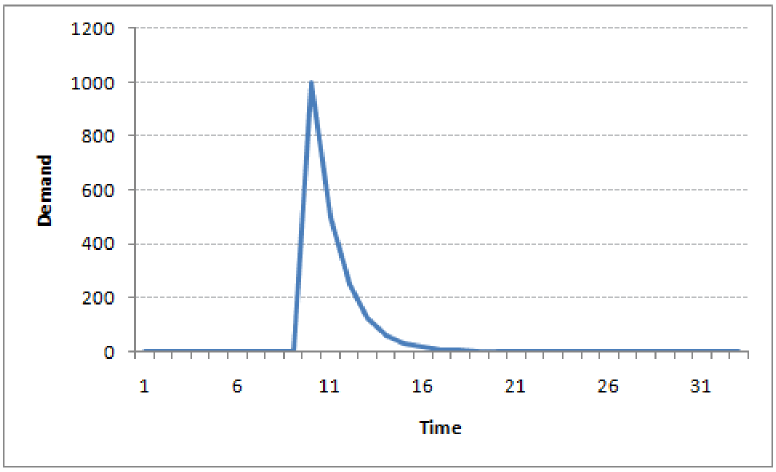


Figure 2.8: Variation Graph for ARMA Model

b. **Moving Average Time Series Model**- given by the formula

$$x(t) = beta * error(t-1) + error(t) \tag{2.8}$$

In MA model, noise / shock quickly vanishes with time whereas, the AR model has a much lasting effect of the shock.

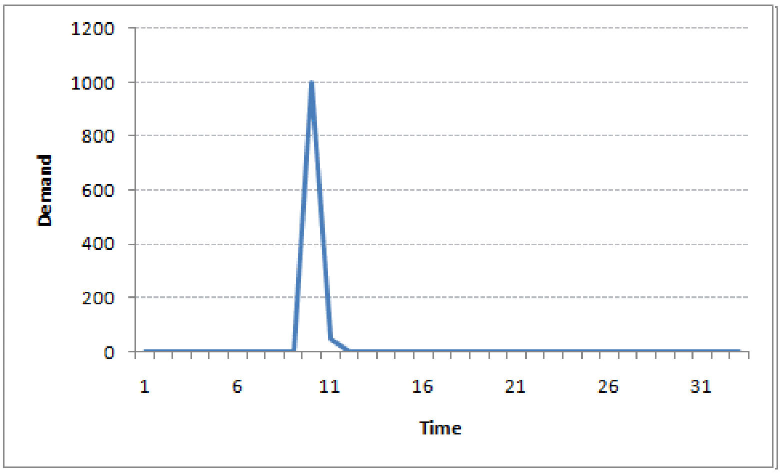


Figure 2.9: Variation Graph for MA Model

This concludes that covariance between x(t) and x(t-n) is zro for MA models. However, the correlation of x(t) and x(t-n) gradually declines with n becoming larger in the AR model.

c. **SAX** - One of most trending and optimum time-series algorithm used today is the SAX. Symbolic Aggregate approximation (SAX) algorithm is a popular and a simple way of transforming time series to a symbolic representation i.e. a sequence of symbols. The motivation of using SAX algorithm for time series data, is that after converting time series data to a sequence of symbols it can be applied onto traditional symbolic pattern mining algorithm such as sequential pattern mining and sequential rule mining algorithm.

   The input time series data can be a sequence of floating point decimal numbers or string value.

   It is necessary to indicate the ”separator” of the time-series data. This is used as a delimiter to separate data points in the input file using comma, semicolon, colon etc.

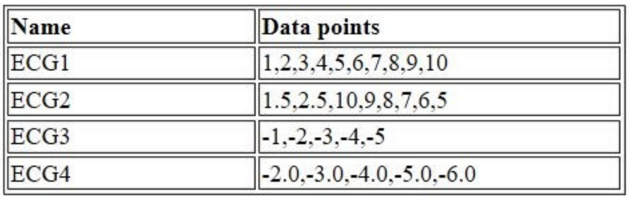
   Along with the input of time-series parameter additionally two other parameters are required :

   i. the number of segments (w)
   ii. number of symbols (v)

   The working of algorithm follows the following steps.

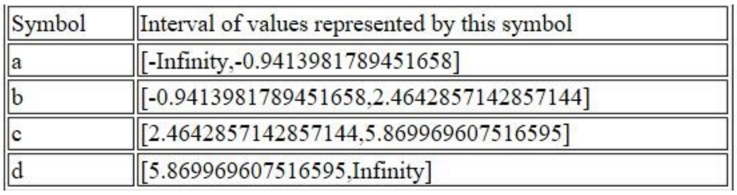
   i. Initially, the time series data is divided into ’w’ segments and each segment is replaced by its data points. This method is referred as piece-wise approximate aggregation (PAA).
   ii. Value of each segment is replaced by a symbol, as provided from the list given by the user. Respective symbol for intervals are chosen depending on the equal probability of the range (from interval) occurring under the normal distribution.

   Example of SAX conversion :



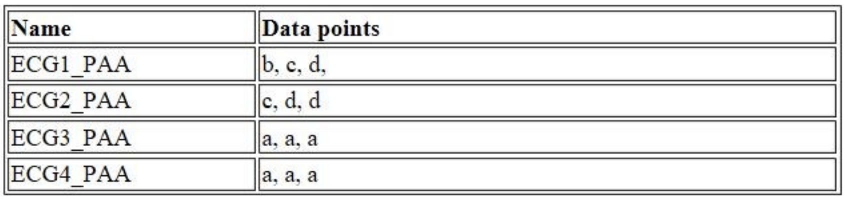
Name	Data points
ECG1	1,2,3,4,5,6,7,8,9,10
ECG2	1.5,2.5,10,9,8,7,6,5
ECG3	-1,-2,-3,-4,-5
ECG4	-2.0,-3.0,-4.0,-5.0,-6.0

Figure 2.10: Time Series Data



Symbol	Interval of values represented by this symbol
a	[-Infinity,-0.9413981789451658]
b	[-0.9413981789451658,2.4642857142857144]
c	[2.4642857142857144,5.869969607516595]
d	[5.869969607516595,Infinity]

Figure 2.11: Representing Symbols for Each Interval



Name	Data points
ECG1_PAA	b, c, d,
ECG2_PAA	c, d, d
ECG3_PAA	a, a, a
ECG4_PAA	a, a, a

Figure 2.12: SAX Result

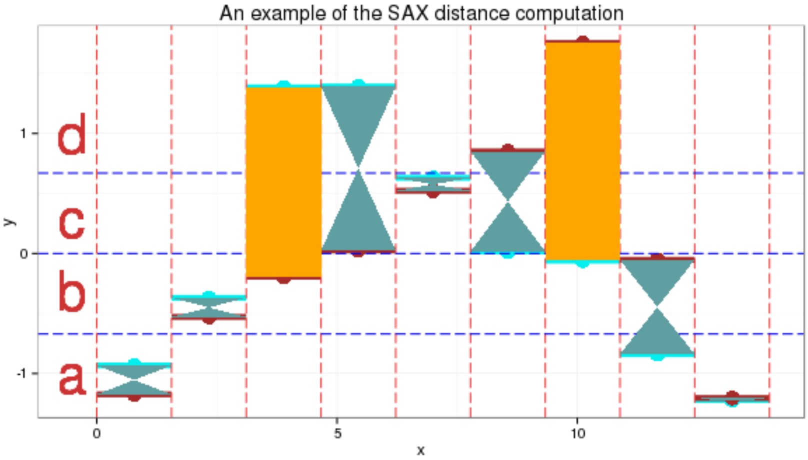


Figure 2.13: Distance Computation Graph

2.3 Data Set Selection

The data set that is appropriate for this project is the UC Irvine Data set. The timeline of the data set is for a year from 2010-2011 . This data set contains a mix of data of furniture, party products, utensils etc. Stock id, transaction id (encrypted) ,units of each product sold, time stamp, date given as DD/MM/YY, the price of each product are the attributes that have been provided by the data set.

1. InvoiceNo: Invoice number uniquely assigned to each transaction. If this code starts with letter ‘c’, it indicates a cancellation.
2. StockCode: Product (item) code uniquely assigned to each distinct product.
3. Description: Product (item) name.
4. Quantity: The quantities of each product (item) per transaction.
5. InvoiceDate: Invoice Date and time, the day and time when each transaction was generated.
6. UnitPrice: Unit price. Product price per unit in sterling.
7. CustomerID: Customer number uniquely assigned to each customer.
8. Country: Country name. The name of the country where each customer reside

The data set is a time-series data, consisting of exact time and date of items purchased. One of the key challenge faced was the generation of test data due to the presence of ONLY one year of data. This data consists of many transactions throughout the year.Due to the presence of multiple transactions everyday, to generate test data stratified k-fold will be used for each day, where 1 fold will be used as a part of the test day and rest for training data set.The data set consists of around 5,41,910 tuples.This data set consists of some of the most important attribute and that is the date, many new features such as seasons weekends, weekdays etc can be generated from this attribute.

2.4 Summary

Hence the model used for retail analysis by several researchers are regression trees, PCR, ANN and fuzzy grey .Some of the other models that can fit this data set are XGBOOST, LGBM and SVR. Data set analyzed for this project was collected from the UCI repository which held time series data.

Chapter 3

Proposed Framework

Proposing a plan to the project implementation, mining for patterns and knowledge and recommending the best strategies to increase the sales.

3.1 Software Requirements

1. **Python**

   Python is one of the most powerful high level programming language that is most commonly used for Machine Learning. Python contains free open-source libraries for efficiently pre-processing data, feature engineering, and for creating various models with the benefit of using easy syntax. Scipy and Numpy are two great libraries which are useful for writing algorithms with respect to linear algebra. Matplotlib is used render various graphs. They also allow us to understand usage of kernel methods (SVM) in Machine Learning. Scipy and Numpy will be used in this project to perform linear algebra required in data-preprocessing i.e. to replace missing value with a global constant or the mean of similar classes. They will also be used to construct various arrays, matrix and to perform algebraic functions like identifying mean, median, mode , min, max.

   Pandas is another software library that is used for Python programming language which allows to perform data manipulation and analysis. It provides data structures and operations for manipulating numerical tables and time series. It will be used to uploading the huge data set into data-frames which will can then be easily altered and manipulated as per the requirement.

Another such useful library in python is the Scikit-learn. It is a free software machine learning library for Python programming. It features inbuilt-functions and wrappers for performing classification, clustering, regression, dimensionality reduction, model selection and pre-processing. Scikit learn will be used to perform cross-validation, k-fold cross validation, calling wrappers for various models, fitting the model with appropriate hyper parameters by performing iterative hyper-parameter tuning and choosing various loss functions and accuracy measures to determining the performance of the model. Scikit learn's built in persistent model, pickle will also be used to save and load the model as when required to reduce the task of re-training and refitting the model when needed.

2. **R Studio**

RStudio is a free and open-source integrated development environment for R, a programming language for statistical computing and graphics. Programmers indulged into data science use various graphical libraries for visualization and to analyze various patterns and trends in the data set. R is one such programming language (more powerful for visualization than python) which provides numerous inbuilt libraries to perform the same effectively, by only writing small amount of code. All visualizations in the project, required to better understand the performance of models and showcase the patterns in the data set will be done using R-studio.

For visualization the Ggplot2 library of Rstudio will be used. Ggplot 2 is an enhanced data visualization package for R. It helps to create stunning multi-layered graphics with ease. Ggplot 2 allows programmers to plot specification at a high level of abstraction. Ggplot 2 is very flexible and user interactive as it provides theme system for polishing the plot appearance.

Various features of ggplot 2 are aesthetic mapping (i.e. position, color, fill, shape, line-type, size) and geometric objects (lines, points, box-plot). Various other graphs such as histograms, dot plots, strip plots, QQ plot which can be used to identify outliers, anomalies and see the variations in the data.

3. **Google Collaboratory**

   Colaboratory is a free cloud service used as a research tool for machine learning and research. This service is provided on the Google cloud which essentially helps programmers to optimally use the large storage space available. It also supports GPU, TPU ,CPU usage, which can be used to train models at low cost in minimum time, due to presence of multiple cores.

   Google Collab has an interface of Jupyter notebook environment that requires no setup to use. Google collab also ensures cyber security as every programmers individual code is executed in a virtual machine dedicated to their account.

4. **IDE Environment –Jupyter Notebook**

   The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. It can be effectively used for data cleaning, transformation, numerical simulation, statistical modeling, data visualization, and machine learning with live implementation and results.

3.2 Implementation Plan

Surveying the above research papers and analyzing the latest trends , for the UCI Online Retail data set the implementation guidelines are broadly categorized as data set collection, feature engineering and model optimization

3.2.1 Work-flow

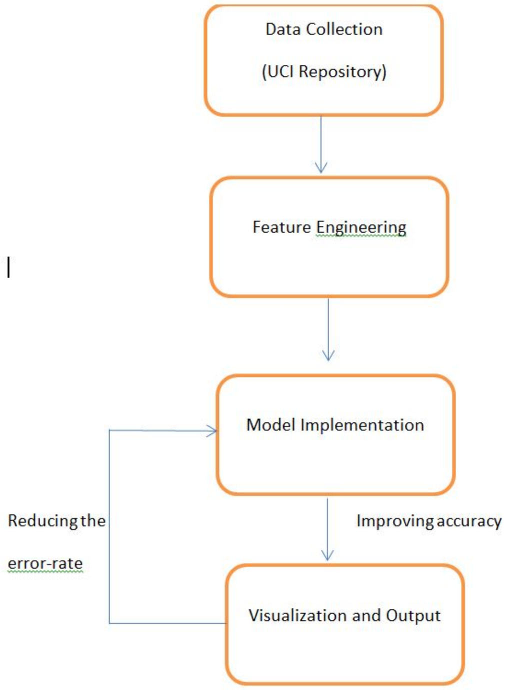


Figure 3.1: Work-Flow Diagram of The Project

3.2.2 Data Set Categories

Aggregate items at the categorical level - essentially aggregating all items that belong to that product category - and predicted demand for each category. This will help us to reduce the tedious task of predicting the demand required for each individual product.Rather, the product category will be predicted from which individual products will be autonomously predicted. This is depending on the weight of each product with respect to their respective categories.

These are hierarchical categories. A weighted average is taken for each item under the specific category. Once the total category is predicted , based on the weighted average calculated for each item, this stock value is distributed amongst the items .

3.2.3 Feature Engineering

One of the characteristics of data set is garbage in - garbage out. It means that if a dirty data is passed in a model we would get garbage values and one of the least accuracies . So in-order to get the best accuracies the concept of feature engineering is used. Feature engineering consists of feature extraction i.e. only including the required features and eliminating the irrelevant features, feature scaling which consists of standardization, normalization and centering.

The various strategies that are going to be followed for pre-processing are handling mission values, feature creation, feature segmentation, feature reduction.

1. **Handling Missing Values**: A database scan which will either remove the empty data rows or replace them with a global constant or with the mean of the relatable class that can be found out using inference based algorithm. The replacement or the removal would be done based on the feature importance. If there are less number of rows that are empty then we can eliminate them, otherwise we have to fill the rows with their global constant values.

2. **Feature Creation**

   a. **Segmenting the day into sessions** : We would be discretizing the time stamp into categories known as sessions of the day(e.g.: morning, night etc), in order to analyze the trends of the product market during different time periods of the day. This would even help in finding the anomalies in the data set, that might not be figured out through normal database scan. The data format will be in the form string data for the analysis that has to be done and will be converted to its numeric form for the prediction.

   b. **Analysis of past few days** : Another strategy includes finding out the variations in quantity sold for items over a range of past few days. This gives an important weightage in weekly or fortnightly analysis for the aggregated items.The Strategy includes making attributes like past few day sales for each of the aggregated items. The past analysis of the given product will allow us to identify the demand trend of the given item as a feature during the last 4 days, helping us use those weights to accurately predict the safety stock required for the product as suggested by Pablo Martin.

c. **Feature Segmentation** : Date is an important feature for stock prediction.The date attribute can be be broken down into the days of the week like Monday, Tuesday etc, after which it is converted into weekdays and weekends. This will also be used in the product analysis and will be used to create new attributes for the past daily or weekly sales of the product. This is even used to gain knowledge and insights on the seasons for a better forecast.

d. **Feature Reduction** : Looking from the optimization point of view for computation, the correlation between the attributes is used to reduce the number of attributes , if the attributes are highly correlated , the attributes are merged using the PCA algorithm that retains the property of both the attributes .

3.2.4 Model Implementation

This project will focus on models other than the time series models. A trade off between the time series and other models like regression trees, ANN, SVR, LGBM, XGBOOST would be reviewed based on the performance and accuracies of the respective models.

3.2.5 Outputs and Visualization

Given below are the proposed outputs and visualizations of the models.

Visualization

In order to decide, which features to use in the final data set for modeling, we want to get a feel for our data. And a good way to do this , is by creating different visualizations. It also helps with assessing your models later on, because the closer you are acquainted with the data's properties, the better you'll be able to pick up on things that might have gone wrong in your analysis.

Plots such as

1. **Visualization by country** - will help identify which population the data is coming from.

2. **Visualization of time across the month** - This is a distribution curve with which a frequency polygon visualization technique is used to identify the number of transactions recorded over a given span of time for a particular month. This will help us identify which days of the month will have maximum hike in sales. This can be used to analyze for special days of the year, especially part of festive months.

3. **Visualization by day and night** - Using a 2-dimensional-bin-plot we can compare a particular session of the day with other months of the year.

4. **Visualization for an item** - We can use a line chart to analyze the sales of each individual item throughout the year. This will help us conclude which item has minimum sales for the given season. The item which is sold the least can be placed during seasonal promotion of the year.This will also help in identifying the weight of the given item for a category.

5. **Visualization of repeat customers** - A pie chart can be used to identify the customers that have repeatedly bought from the store. This will help provide loyalty points and discounts to frequent customers.

Outputs

The primary output of this project is the forecasted stock value of the category in its numeric form. This value can later be used to suggest the safety stock required .The other secondary outputs would be the suggestion given on whether to keep or discontinue the item based on the analysis.

Chapter 4

Summary and Future Work

4.1 Summary

The background of this project will based on the economic strategies used for the retail stock management.These concepts would be then used to mine the data from the data set that we use for this project. This mining will be utilized to visualize the different scenarios of the stock variation. Analysis of the visualization will help us refine the model to remove anomalies, outliers and fine tune it. On refinement, predictions are made using the different algorithms and hybrid models will be implemented depending on performance and accuracy of the model.

Since only one year was fetched, k-fold stratified algorithm was applied for generating the test data.

4.2 Future Work

1. Implementation of the data preprocessing techniques.

2. Execution of different models and performance based trade

3. Statistical report of sales made throughout the year.

4. Cluster the most frequent bought objects.

Chapter 5

Data Preprocessing

Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing [8] is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing by transforming raw data into an understandable format. Data Preprocessing includes:

1. Data Cleaning

2. Data Transformation

3. Feature Engineering

5.1 Data Cleaning

Data cleansing [9] or data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or dataset and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.

Benefits of Data Cleaning:

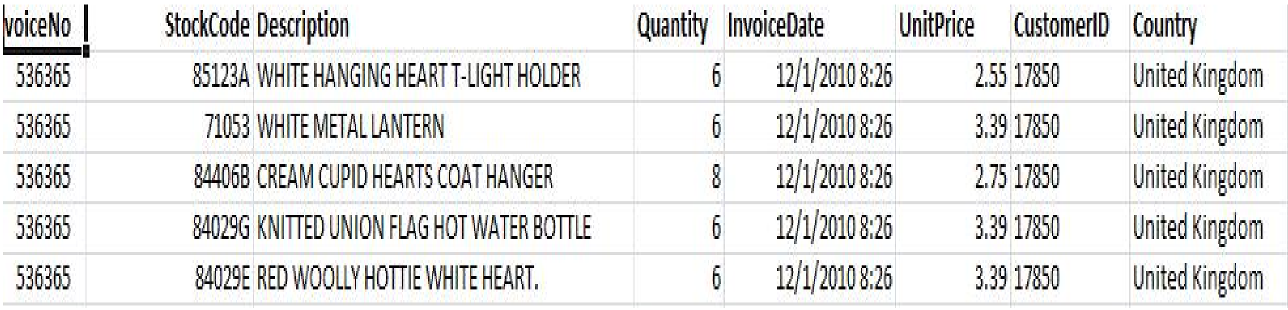
- Make accurate predictions
- Increases Productivity so that the employees can make best use of their working hours
- Reduces the size of database by removing redundancy.

- Can help avoid unnecessary increase in costs

Common Steps involved in data cleaning:

- Replacing missing values (by mean or mode)
- Removing empty tuples
- Removing Out-Dated data
- Removing non-sensible data
- Removing unimportant records
- Removing Duplicate records
- Normalizing the dataset
- Standardizing the dataset

Platform used in the project for Data Cleaning is Python (Jupiter Notebook).Original dataset contained 541910 instances with attributes namely InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID and Country.



voiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom

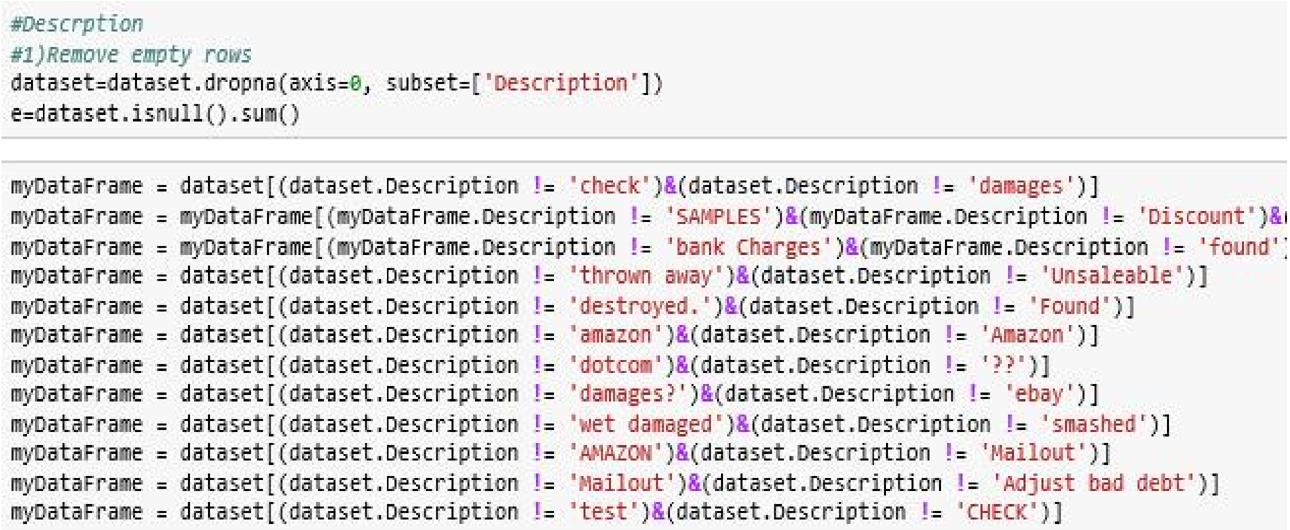
Figure 5.1: Original Data

Steps undertaken in the project for Data Cleaning (for particular attributes) are as follows:

For **Description:**

- Tuples with empty Description were removed.

- About 75 unique values of Description were found to be irrelevant (noisy data) like “thrown away”, “???missing”, “damages/dotcom”, “Discount”, “smashed”, etc. Hence, all tuples with any one of these values were removed.



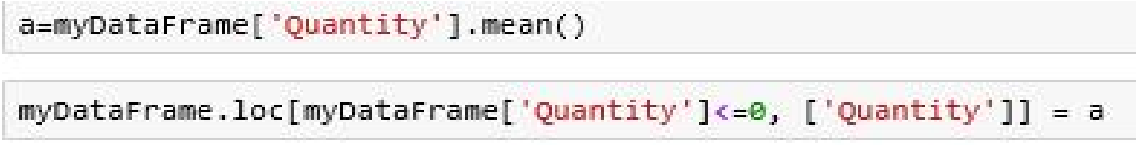
```
#Descrption
#1)Remove empty rows
dataset=dataset.dropna(axis=0, subset=['Description'])
e=dataset.isnull().sum()
```

```
myDataFrame = dataset[(dataset.Description != 'check')&(dataset.Description != 'damages')]
myDataFrame = myDataFrame[(myDataFrame.Description != 'SAMPLES')&(myDataFrame.Description != 'Discount')&
myDataFrame = myDataFrame[(myDataFrame.Description != 'bank Charges')&(myDataFrame.Description != 'found'
myDataFrame = dataset[(dataset.Description != 'thrown away')&(dataset.Description != 'Unsaleable')]
myDataFrame = dataset[(dataset.Description != 'destroyed.')&(dataset.Description != 'Found')]
myDataFrame = dataset[(dataset.Description != 'amazon')&(dataset.Description != 'Amazon')]
myDataFrame = dataset[(dataset.Description != 'dotcom')&(dataset.Description != '??')]
myDataFrame = dataset[(dataset.Description != 'damages?')&(dataset.Description != 'ebay')]
myDataFrame = dataset[(dataset.Description != 'wet damaged')&(dataset.Description != 'smashed')]
myDataFrame = dataset[(dataset.Description != 'AMAZON')&(dataset.Description != 'Mailout')]
myDataFrame = dataset[(dataset.Description != 'Mailout')&(dataset.Description != 'Adjust bad debt')]
myDataFrame = dataset[(dataset.Description != 'test')&(dataset.Description != 'CHECK')]
```

Figure 5.2: Data Cleaning step 1

For **Quantity:**

- Tuples which contained negative value for quantity were first converted to 0. Then the quantity value for these tuples were replaced with the mean of the of the whole column which was found to be equal to 7.



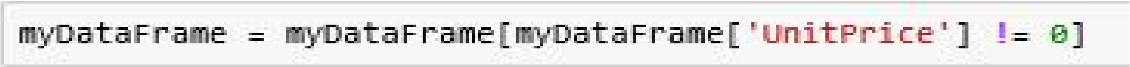
```
a=myDataFrame['Quantity'].mean()
```

```
myDataFrame.loc[myDataFrame['Quantity']<=0, ['Quantity']] = a
```

Figure 5.3: Data Cleaning step 2

For **UnitPrice:**

- Tuples which had a unit price of 0 were removed as they were anomalous. As they were only a couple of such abnormal instances they were simply ignored and not replaced by mean or mode.

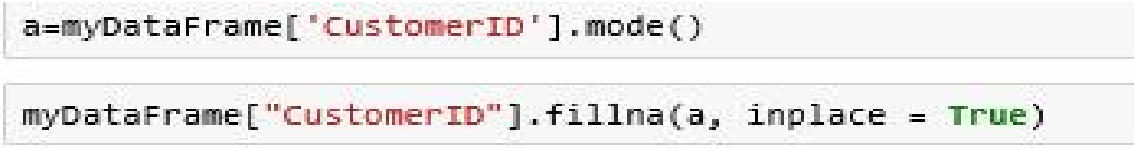


```
myDataFrame = myDataFrame[myDataFrame['UnitPrice'] != 0]
```

Figure 5.4: Data Cleaning step 3

For **CustomerID:**

- About 133626 instances were identified which had no CustomerID. As it would not have been possible to remove all these instances, they were replaced by the most common CustomerID (as it was a categorical attribute) found by taking mode of the column values.



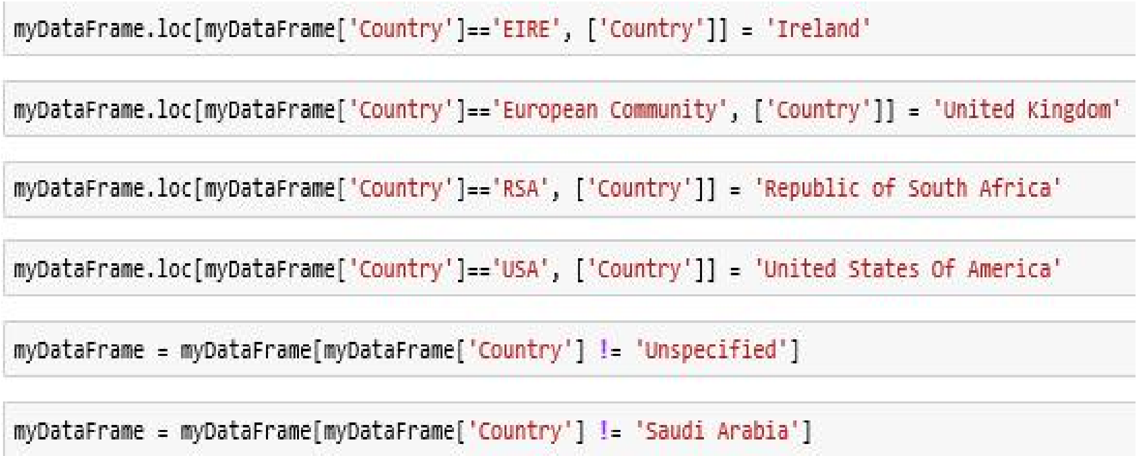
```
a=myDataFrame['CustomerID'].mode()
```

```
myDataFrame["CustomerID"].fillna(a, inplace = True)
```

Figure 5.5: Data Cleaning step 4

For **Country:**

- First, the country column was analyzed to see if there were any cities. Accordingly, the cities were replaced with the name of their country.
- Country values like RSA were replaced with their full form “Republic of South Africa” in order to maintain uniformity with other countries
- Tuples with unspecified country names were ignored as they were just few in number.
- Communities such as European Communities were replaced by the most common European Community that is United Kingdom
- Countries like Saudi Arabia which had very few records like 10 were removed as they would serve no useful purpose for our training.



```
myDataFrame.loc[myDataFrame['Country']=='EIRE', ['Country']] = 'Ireland'
```

```
myDataFrame.loc[myDataFrame['Country']=='European Community', ['Country']] = 'United Kingdom'
```

```
myDataFrame.loc[myDataFrame['Country']=='RSA', ['Country']] = 'Republic of South Africa'
```

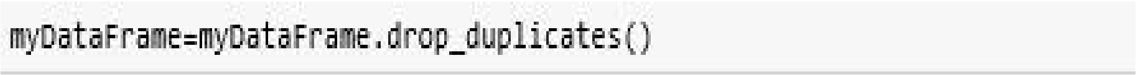
```
myDataFrame.loc[myDataFrame['Country']=='USA', ['Country']] = 'United States Of America'
```

```
myDataFrame = myDataFrame[myDataFrame['Country'] != 'Unspecified']
```

```
myDataFrame = myDataFrame[myDataFrame['Country'] != 'Saudi Arabia']
```

Figure 5.6: Data Cleaning step 5

- In addition Removing duplicate records were removed from the dataset.



```
myDataFrame=myDataFrame.drop_duplicates()
```

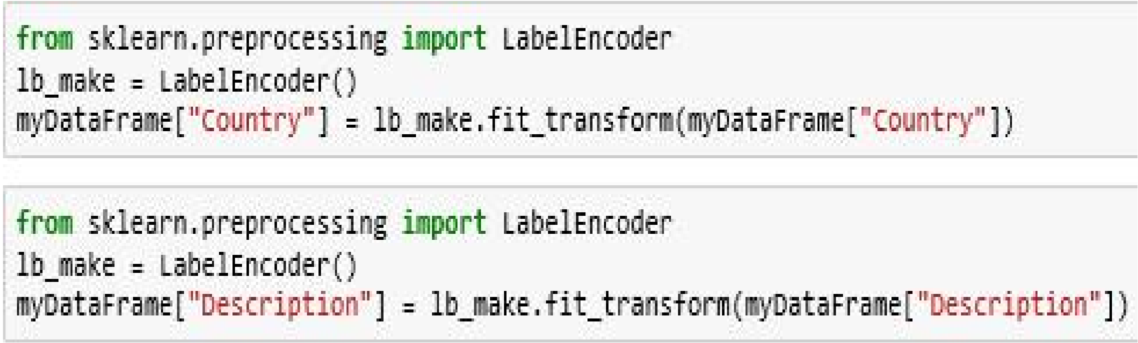
Figure 5.7: Data Cleaning step 6

5.2 Data Transformation

Data Transformation [10] is for transforming data from one form to another form. Steps Undertaken in The project for Data Transformation are as follows:

- **Converting Categorical Values To Numerical Form**

  This included giving numerical labels Categorical Attributes like Country and Description. An Inbuilt python Library called LabelEncoder was used for this purpose which assigned labels like 1, 2, 3... to each unique country depending on the order in which they appeared.



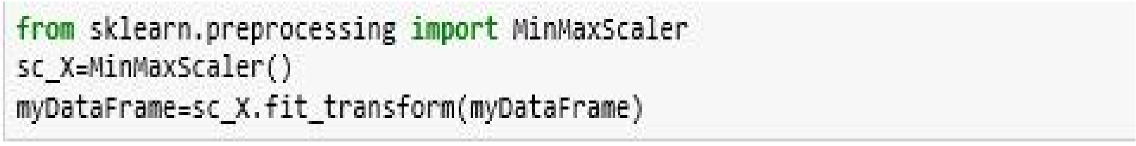
```
from sklearn.preprocessing import LabelEncoder
lb_make = LabelEncoder()
myDataFrame["Country"] = lb_make.fit_transform(myDataFrame["Country"])
```

```
from sklearn.preprocessing import LabelEncoder
lb_make = LabelEncoder()
myDataFrame["Description"] = lb_make.fit_transform(myDataFrame["Description"])
```

Figure 5.8: Data Transformation step 1

- **Normalization**

  Features with different units (unscaled) can lead to difficulties to visualize the data and more importantly, they can degrade the predictive performance of many machine learning algorithms. Unscaled data can also slow down or even prevent the convergence of many gradient-based estimators. Normalization helps to prevent this from happening by rescaling the features in order to standardize them. MinMaxScaler(), an inbuilt python library was used to scale features between one and zero.

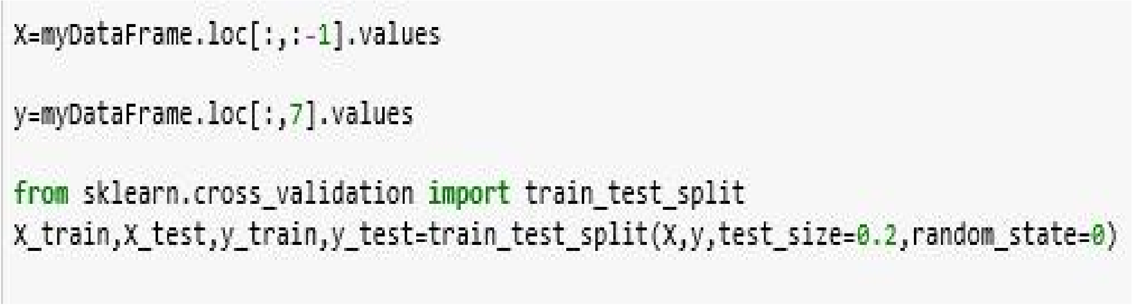


```
from sklearn.preprocessing import MinMaxScaler
sc_X=MinMaxScaler()
myDataFrame=sc_X.fit_transform(myDataFrame)
```

Figure 5.9: Data Transformation step 2

Splitting into Train and test set

Apart from data transformation, Train_test_split, an inbuilt Python Library was used to split the dataset into training set and test set randomly with training set having 80 percent of the records and test set having 20 percent of the records.



```
X=myDataFrame.loc[:,:-1].values

y=myDataFrame.loc[:,7].values

from sklearn.cross_validation import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
```

Figure 5.10: TrainTestSplit

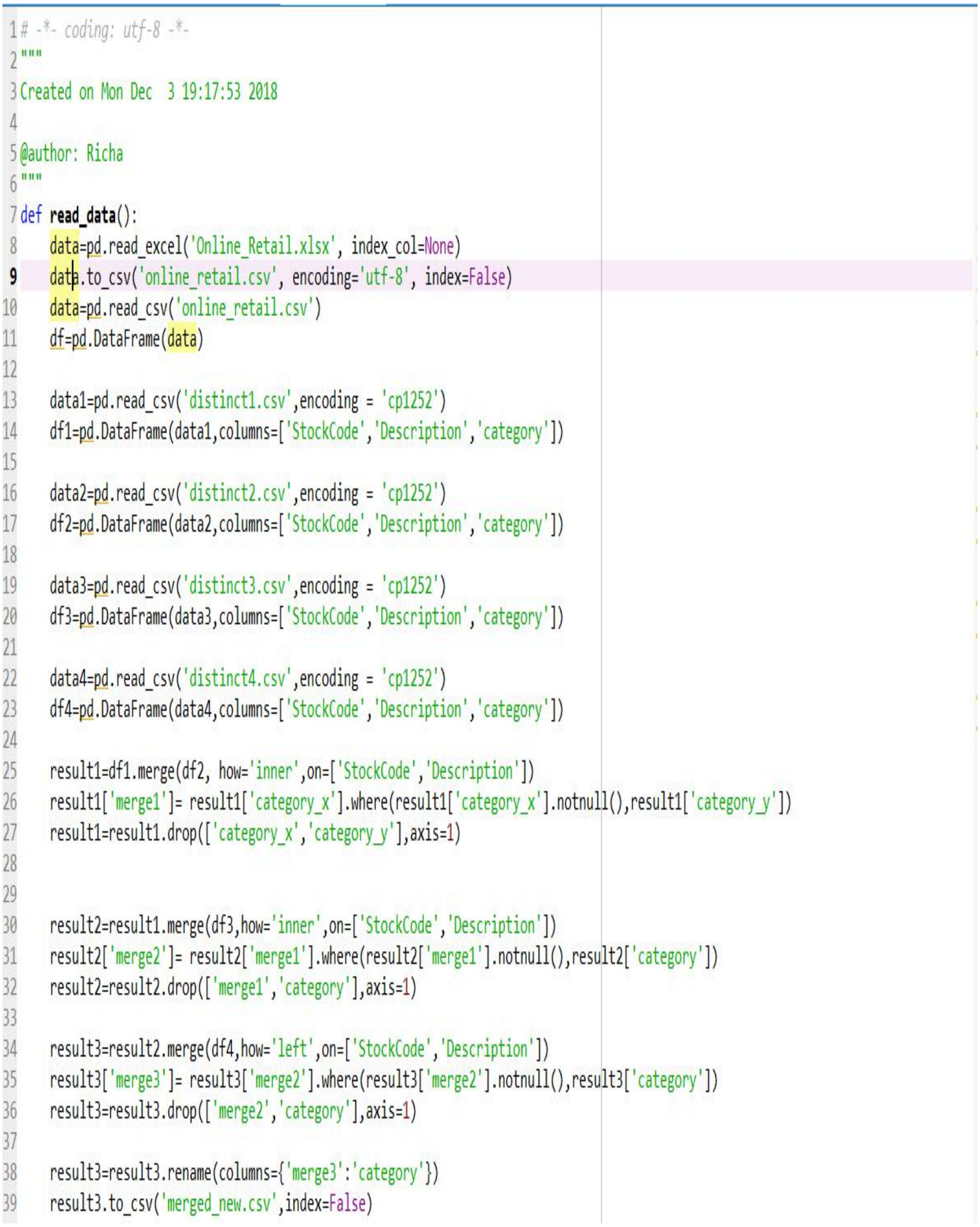
The total number of instances after cleaning the data set were 538373.

5.3 Feature Engineering

Feature Engineering is an important phase of the Predictive Modeling project. It involves using the domain knowledge to create relatable features. This is necessary to generate comprehensive knowledge of the data as a result of the models. Features are augmented and processed. Incubating features allows the model to derive higher accuracy and knowledge from the data. Feature engineering involves using domain knowledge of data to create features. The entire feature engineering process was successfully executed using the Anaconda Navigator on the python 3.6 platform [12].

5.3.1 Labeling item-set

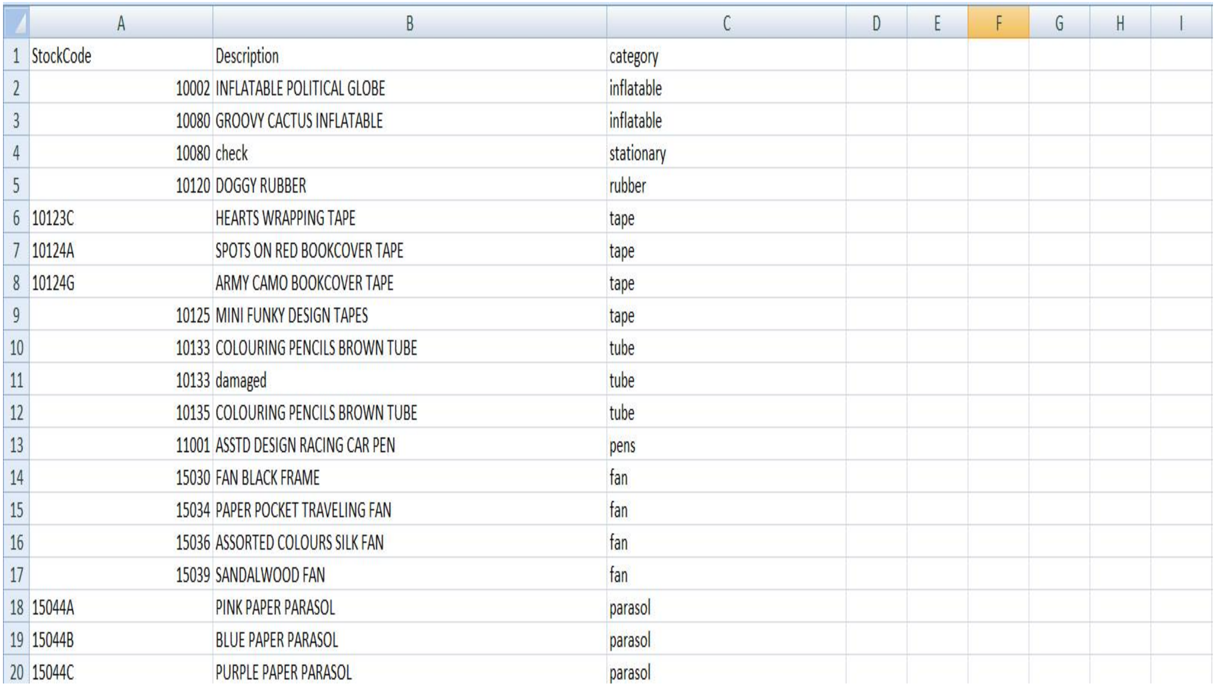
The dataset contains 5.5 lakh transactions! Notifying the retailer of the sales required for each item would be a tedious task, for a retailer to analyse at the end of each week. A better alternative to resolving his conflict is to generate categories which in turn will determine the quantity of each item.



```python
# -*- coding: utf-8 -*-
"""
Created on Mon Dec  3 19:17:53 2018

@author: Richa
"""
def read_data():
    data=pd.read_excel('Online_Retail.xlsx', index_col=None)
    data.to_csv('online_retail.csv', encoding='utf-8', index=False)
    data=pd.read_csv('online_retail.csv')
    df=pd.DataFrame(data)

    data1=pd.read_csv('distinct1.csv',encoding = 'cp1252')
    df1=pd.DataFrame(data1,columns=['StockCode','Description','category'])

    data2=pd.read_csv('distinct2.csv',encoding = 'cp1252')
    df2=pd.DataFrame(data2,columns=['StockCode','Description','category'])

    data3=pd.read_csv('distinct3.csv',encoding = 'cp1252')
    df3=pd.DataFrame(data3,columns=['StockCode','Description','category'])

    data4=pd.read_csv('distinct4.csv',encoding = 'cp1252')
    df4=pd.DataFrame(data4,columns=['StockCode','Description','category'])

    result1=df1.merge(df2, how='inner',on=['StockCode','Description'])
    result1['merge1']= result1['category_x'].where(result1['category_x'].notnull(),result1['category_y'])
    result1=result1.drop(['category_x','category_y'],axis=1)


    result2=result1.merge(df3,how='inner',on=['StockCode','Description'])
    result2['merge2']= result2['merge1'].where(result2['merge1'].notnull(),result2['category'])
    result2=result2.drop(['merge1','category'],axis=1)

    result3=result2.merge(df4,how='left',on=['StockCode','Description'])
    result3['merge3']= result3['merge2'].where(result3['merge2'].notnull(),result3['category'])
    result3=result3.drop(['merge2','category'],axis=1)

    result3=result3.rename(columns={'merge3':'category'})
    result3.to_csv('merged_new.csv',index=False)
```

Figure 5.11: Code for merging categories

These items were categorized into their respective categories by each team member individually. Each member was given 1250 items to categorize. Efficiently, each member's work was merged together to generate the attribute of category. Each member's contribution is merged by using the inner and left outer join. Left outer join is used to ensure all keyed items of the former dataframe matches with those of the last to perform intersection of categories. While merging if a null item of either the member's dataframe is determined the not null cell of the other is considered.



	A	B	C	D	E	F	G	H	I
1	StockCode	Description	category						
2	10002	INFLATABLE POLITICAL GLOBE	inflatable						
3	10080	GROOVY CACTUS INFLATABLE	inflatable						
4	10080	check	stationary						
5	10120	DOGGY RUBBER	rubber						
6	10123C	HEARTS WRAPPING TAPE	tape						
7	10124A	SPOTS ON RED BOOKCOVER TAPE	tape						
8	10124G	ARMY CAMO BOOKCOVER TAPE	tape						
9	10125	MINI FUNKY DESIGN TAPES	tape						
10	10133	COLOURING PENCILS BROWN TUBE	tube						
11	10133	damaged	tube						
12	10135	COLOURING PENCILS BROWN TUBE	tube						
13	11001	ASSTD DESIGN RACING CAR PEN	pens						
14	15030	FAN BLACK FRAME	fan						
15	15034	PAPER POCKET TRAVELING FAN	fan						
16	15036	ASSORTED COLOURS SILK FAN	fan						
17	15039	SANDALWOOD FAN	fan						
18	15044A	PINK PAPER PARASOL	parasol						
19	15044B	BLUE PAPER PARASOL	parasol						
20	15044C	PURPLE PAPER PARASOL	parasol						

Figure 5.12: Output of merging categories

As a whole, the model is trained with input data which in turn predicts the quantity of each category that is required. Quantity of each item in respective category is weighted using weighted average using the cumulative frequency of items in each category.

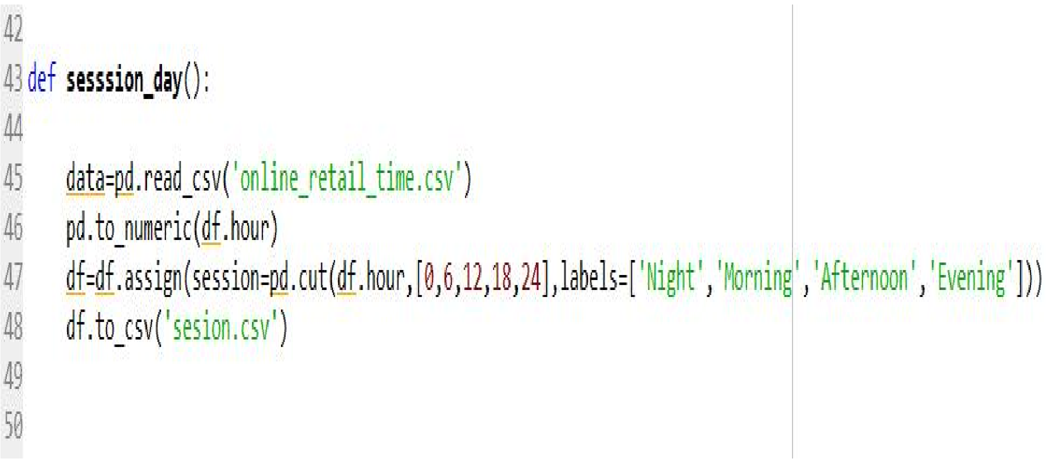
5.3.2 Feature Extraction

For better analysis of the data, the date attribute is broken down into year, month, week, day and hour. These categories would allow to explicitly specify the session of the day when the transaction occurred, to identify whether the particular day is a weekday, weekend, previous sales of the item from the past

3 weeks.

1. Sessions of the day

Each day is divided into sessions of the day. Sessions are night,morning, afternoon, and evening. This is done in order to visualize and analyse which part of the day incurs more sales as a whole as well as for a particular item. Using this technique would help developing particular market schemes for a given category of item to be sold, depending on its highest demand time.

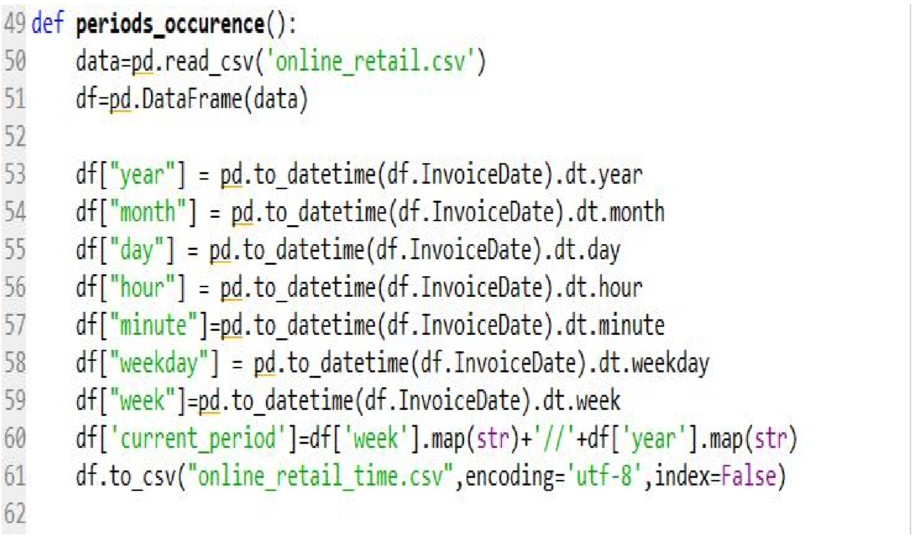


```
def sesssion_day():

    data=pd.read_csv('online_retail_time.csv')
    pd.to_numeric(df.hour)
    df=df.assign(session=pd.cut(df.hour,[0,6,12,18,24],labels=['Night','Morning','Afternoon','Evening']))
    df.to_csv('sesion.csv')
```

Figure 5.13: Code for session of the day

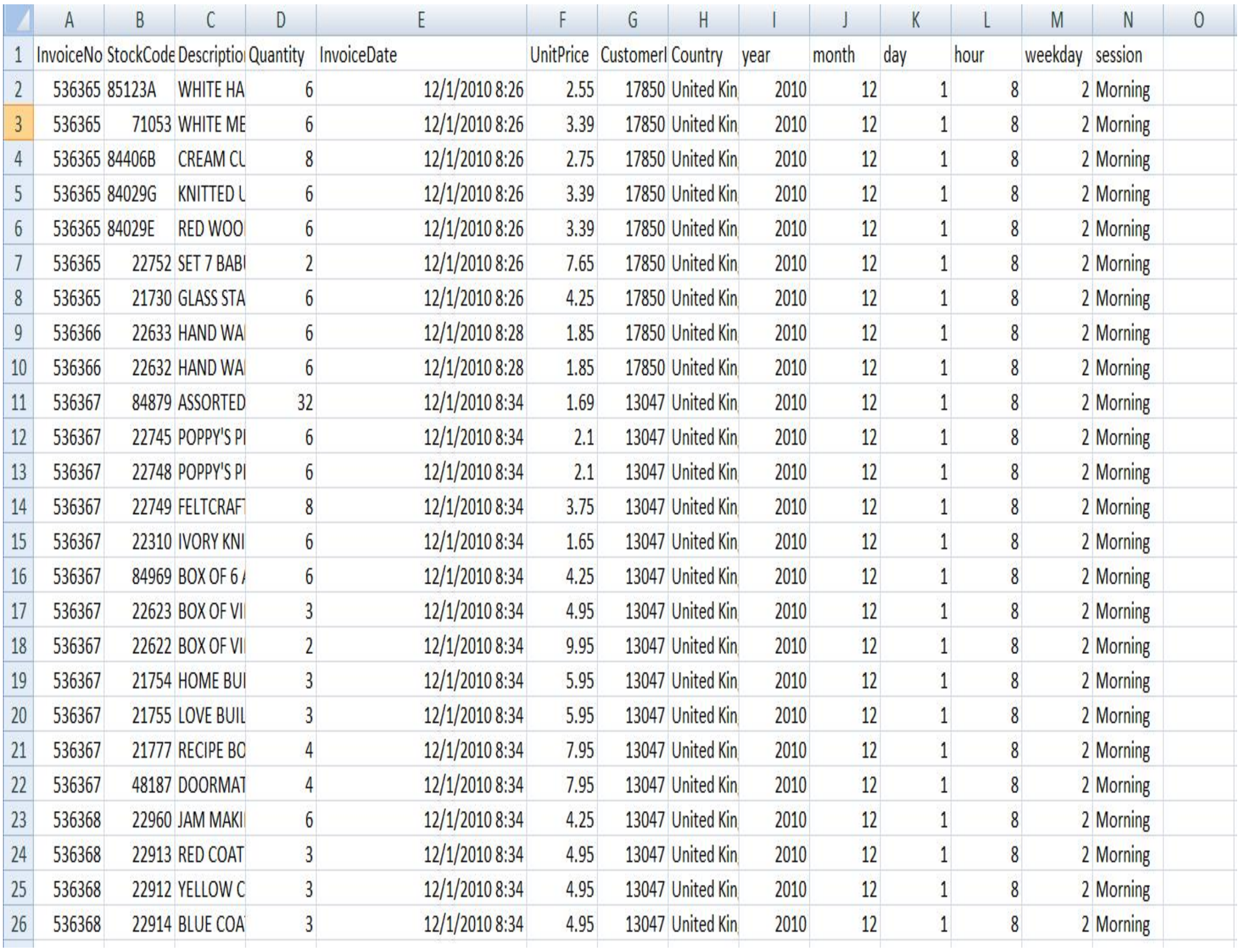
2. Weekday

The date is also classified into day of the week. All dates are divided into year, month, day, hour etc by converting the InvoiceDate column into a datetime type. This would help to identify whether the particular day is a weekday or a weekend



```
def periods_occurence():
    data=pd.read_csv('online_retail.csv')
    df=pd.DataFrame(data)

    df["year"] = pd.to_datetime(df.InvoiceDate).dt.year
    df["month"] = pd.to_datetime(df.InvoiceDate).dt.month
    df["day"] = pd.to_datetime(df.InvoiceDate).dt.day
    df["hour"] = pd.to_datetime(df.InvoiceDate).dt.hour
    df["minute"]=pd.to_datetime(df.InvoiceDate).dt.minute
    df["weekday"] = pd.to_datetime(df.InvoiceDate).dt.weekday
    df["week"]=pd.to_datetime(df.InvoiceDate).dt.week
    df['current_period']=df['week'].map(str)+'//'+df['year'].map(str)
    df.to_csv("online_retail_time.csv",encoding='utf-8',index=False)

```

Figure 5.14: Code for dividing year, month, day, hour, minute

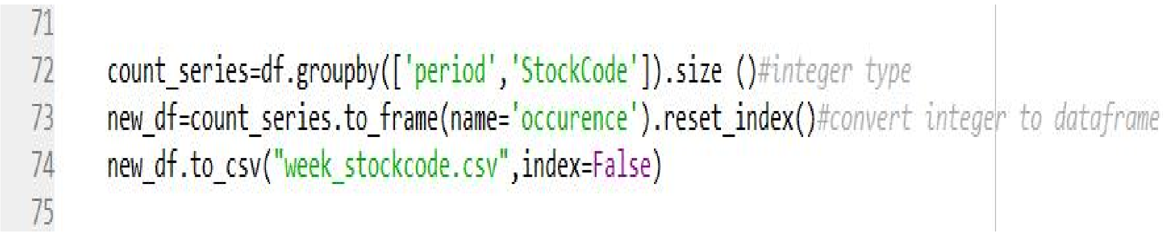


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	InvoiceNo	StockCode	Descriptio	Quantity	InvoiceDate	UnitPrice	CustomerI	Country	year	month	day	hour	weekday	session	
2	536365	85123A	WHITE HA	6	12/1/2010 8:26	2.55	17850	United Kin	2010	12	1	8	2	Morning	
3	536365	71053	WHITE ME	6	12/1/2010 8:26	3.39	17850	United Kin	2010	12	1	8	2	Morning	
4	536365	84406B	CREAM CU	8	12/1/2010 8:26	2.75	17850	United Kin	2010	12	1	8	2	Morning	
5	536365	84029G	KNITTED U	6	12/1/2010 8:26	3.39	17850	United Kin	2010	12	1	8	2	Morning	
6	536365	84029E	RED WOO	6	12/1/2010 8:26	3.39	17850	United Kin	2010	12	1	8	2	Morning	
7	536365	22752	SET 7 BAB	2	12/1/2010 8:26	7.65	17850	United Kin	2010	12	1	8	2	Morning	
8	536365	21730	GLASS STA	6	12/1/2010 8:26	4.25	17850	United Kin	2010	12	1	8	2	Morning	
9	536366	22633	HAND WA	6	12/1/2010 8:28	1.85	17850	United Kin	2010	12	1	8	2	Morning	
10	536366	22632	HAND WA	6	12/1/2010 8:28	1.85	17850	United Kin	2010	12	1	8	2	Morning	
11	536367	84879	ASSORTED	32	12/1/2010 8:34	1.69	13047	United Kin	2010	12	1	8	2	Morning	
12	536367	22745	POPPY'S PI	6	12/1/2010 8:34	2.1	13047	United Kin	2010	12	1	8	2	Morning	
13	536367	22748	POPPY'S PI	6	12/1/2010 8:34	2.1	13047	United Kin	2010	12	1	8	2	Morning	
14	536367	22749	FELTCRAF	8	12/1/2010 8:34	3.75	13047	United Kin	2010	12	1	8	2	Morning	
15	536367	22310	IVORY KNI	6	12/1/2010 8:34	1.65	13047	United Kin	2010	12	1	8	2	Morning	
16	536367	84969	BOX OF 6 /	6	12/1/2010 8:34	4.25	13047	United Kin	2010	12	1	8	2	Morning	
17	536367	22623	BOX OF VI	3	12/1/2010 8:34	4.95	13047	United Kin	2010	12	1	8	2	Morning	
18	536367	22622	BOX OF VI	2	12/1/2010 8:34	9.95	13047	United Kin	2010	12	1	8	2	Morning	
19	536367	21754	HOME BUI	3	12/1/2010 8:34	5.95	13047	United Kin	2010	12	1	8	2	Morning	
20	536367	21755	LOVE BUIL	3	12/1/2010 8:34	5.95	13047	United Kin	2010	12	1	8	2	Morning	
21	536367	21777	RECIPE BO	4	12/1/2010 8:34	7.95	13047	United Kin	2010	12	1	8	2	Morning	
22	536367	48187	DOORMAT	4	12/1/2010 8:34	7.95	13047	United Kin	2010	12	1	8	2	Morning	
23	536368	22960	JAM MAKI	6	12/1/2010 8:34	4.25	13047	United Kin	2010	12	1	8	2	Morning	
24	536368	22913	RED COAT	3	12/1/2010 8:34	4.95	13047	United Kin	2010	12	1	8	2	Morning	
25	536368	22912	YELLOW C	3	12/1/2010 8:34	4.95	13047	United Kin	2010	12	1	8	2	Morning	
26	536368	22914	BLUE COA	3	12/1/2010 8:34	4.95	13047	United Kin	2010	12	1	8	2	Morning	

Figure 5.15: Output of splitting the date

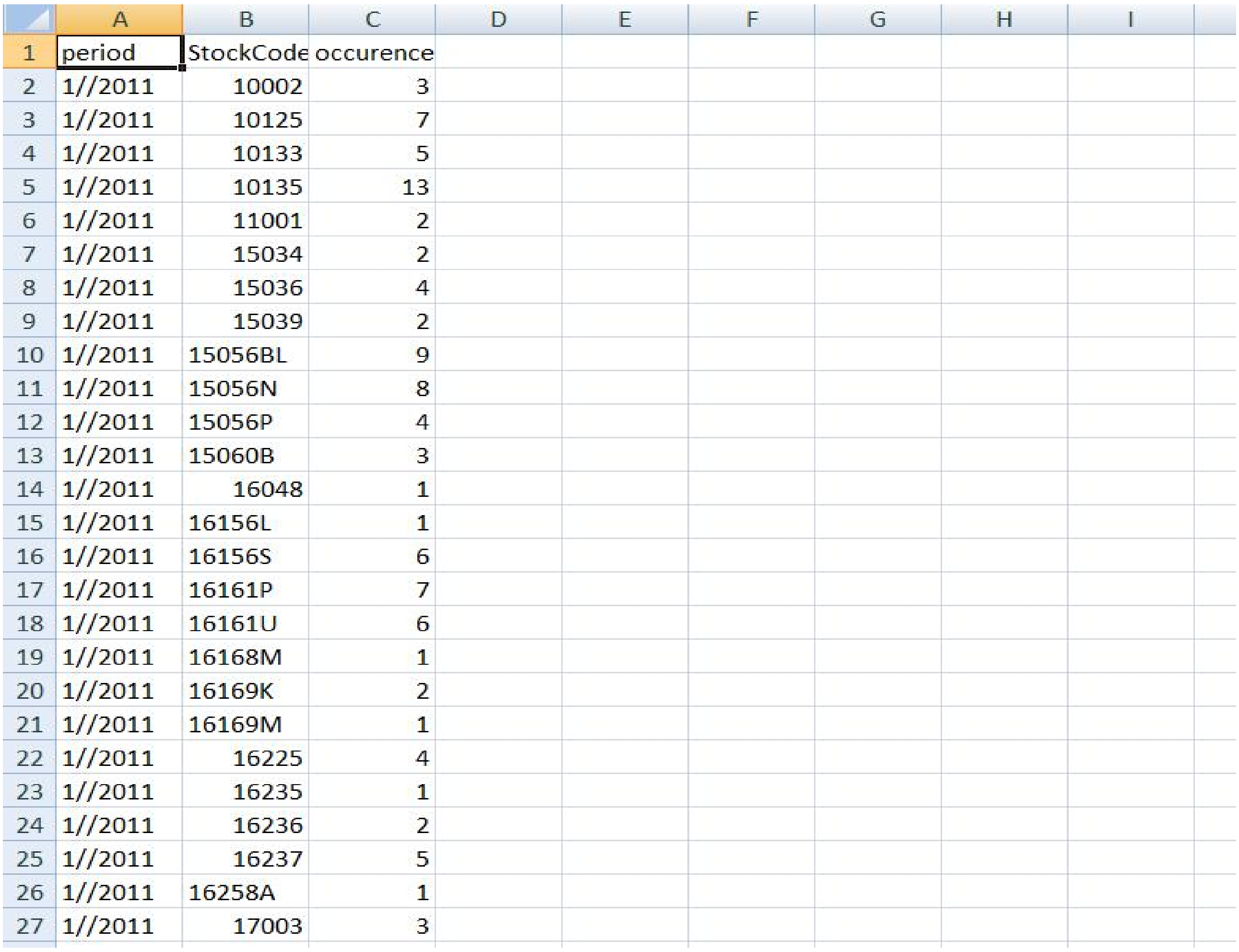
3. Past Week Sales

It is important to analyze the past week sales for a given item, in order to predict the requirement of a particular item for the current week with respect to its sales in the past few weeks. This would provide an additional incite to the model, to identify the sudden rise or drop in sale of an item. This sudden change in the behavior of its sales can be due to an indirect un-identified attribute [13]. The code to produce the output is shown in Fig. 4.16.



```

count_series=df.groupby(['period','StockCode']).size ()#integer type
new_df=count_series.to_frame(name='occurence').reset_index()#convert integer to dataframe
new_df.to_csv("week_stockcode.csv",index=False)

```

Figure 5.16: Code for occurrence in each week



	A	B	C
1	period	StockCode	occurence
2	1//2011	10002	3
3	1//2011	10125	7
4	1//2011	10133	5
5	1//2011	10135	13
6	1//2011	11001	2
7	1//2011	15034	2
8	1//2011	15036	4
9	1//2011	15039	2
10	1//2011	15056BL	9
11	1//2011	15056N	8
12	1//2011	15056P	4
13	1//2011	15060B	3
14	1//2011	16048	1
15	1//2011	16156L	1
16	1//2011	16156S	6
17	1//2011	16161P	7
18	1//2011	16161U	6
19	1//2011	16168M	1
20	1//2011	16169K	2
21	1//2011	16169M	1
22	1//2011	16225	4
23	1//2011	16235	1
24	1//2011	16236	2
25	1//2011	16237	5
26	1//2011	16258A	1
27	1//2011	17003	3

Figure 5.17: Output of frequency of each item

Chapter 6

Visualization

Visualization is a very important data mining tool that is helpful to create human interpretable visuals from complex data and model outputs which play a pivotal role in decision making. A brief introduction to Tableau is followed by its importance as a data visualization tool. Next section covers the visualizations derived from the retail data set in hand followed by the conclusions drawn from it.

6.1 Introduction to Tableau

Data visualization is a general term that describes any effort to help people understand the significance of data by placing it in a visual context. Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier with data visualization software.

The visualization and interpretation of data is becoming an important skill in today's business world. Indeed, the inclusion of data visualizations and dashboards allows analysts to make informed decisions and arrive at logical conclusions in a short amount of time given the information which is needed. This is a critical part of the evolution of business intelligence to what is now known as the area of business analytics.

Developing effective visualizations is crucial for understanding and communicating data in a variety of contexts. Tableau is a tool for creating customized, interactive visualizations. Tableau can also be used to create effective scientific visualizations in the context of public health, seismic engineering, etc.

Interactivity can be introduced into Tableau visualizations by utilizing filters (for example, using a filter to select only a specific year to show on the chart) and tooltips, which show a message when the cursor is placed over a data object (for example, showing information about a point on a scatterplot). Multiple visualizations can be combined on dashboards, which allow users to get a more complete view of their data by placing several charts in one place. Stories are another way of combining visualizations, presenting multiple visualizations one at a time in a sequence, similar to a slideshow. Tableau visualizations can be created using many different data types, including both static data sets (like an Excel file) and dynamic data sets (like a server or database). When Tableau is connected to a dynamically updating data source, the visualizations provide a live, automatically updating view of the data. Thus, Tableau is useful for creating dashboards that reflect the current status of a system or data set.

6.2 Visualizing the retail data set

Here, the aim is to build effective visualizations for the given retail data set in order to provide a major tool for business analytics and gain insight into an information space by mapping data onto graphical primitives, provide qualitative overview of large data sets, search for patterns, trends, structure, irregularities, relationships among data, help find interesting regions and suitable parameters for further quantitative analysis and provide a visual proof of computer representations derived.

Several dimensions and measures were put against each other to create the visualizations in Tableau that have been explained below. A lot of perspective about current and future scope of the project has been derived from the same.

1. This map provides an overview of the sales made by the company, country-wise. It represents the quantity of items sold in each country and also how that quantity compares to the quantity sold in the other countries represented by colour gradient in the map.

This helps in giving a perspective of which market the sales should focus on, where a new venture would be profitable, where the demand would rise in the future and how these countries can be segregated into markets and focused on in a group rather than focusing on each country. This can also help with dynamic data, in case a promotion is launched all over, we can track its effect over the countries and according plan the future plans.

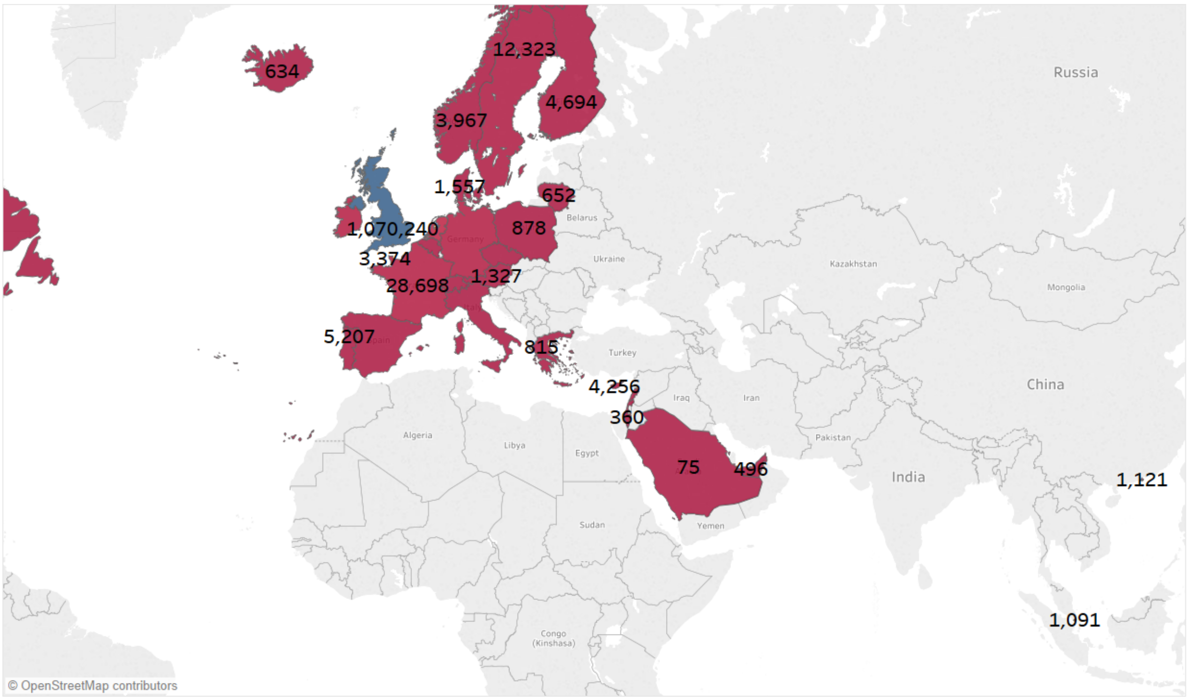


Figure 6.1: Map plot representing quantity sold per country

2. This map represents the sales made in each category in each country. an overview of the sales made by the company, country-wise. The sales are also compared and colour gradient is applied to view which countries have more sale of those items.

   This helps in giving a perspective of which market the sales of a particular item should focus on and which countries the category is not getting sold at all. Also sales and promotions on the categories in the particular countries can be visualized dynamically.

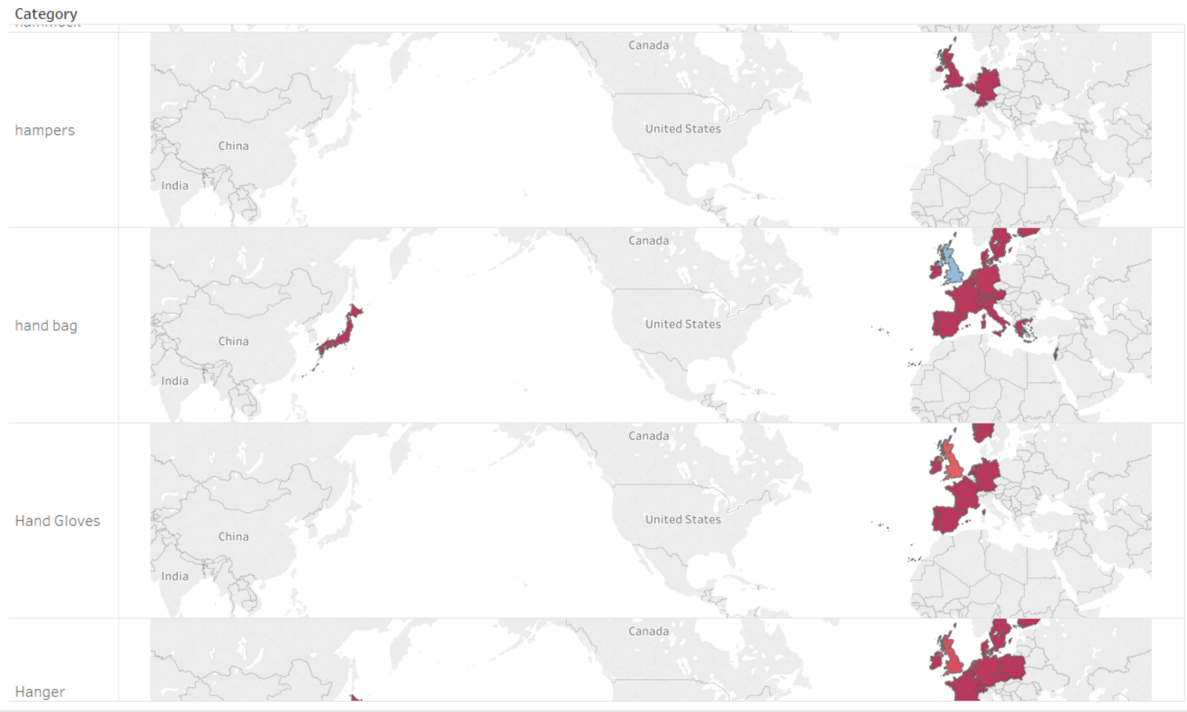


Figure 6.2: Map plot representing countries where each category of products was sold

3. The bar graph below compares over quantity of each product sold, till date. This helps in deciding which categories are the most popular or frequently bought items overall and they have to be handled separately as compared to the categories that are sold less frequently but that doesn't mean they aren't useful. A time-stamp analysis done later will further help in the aim.

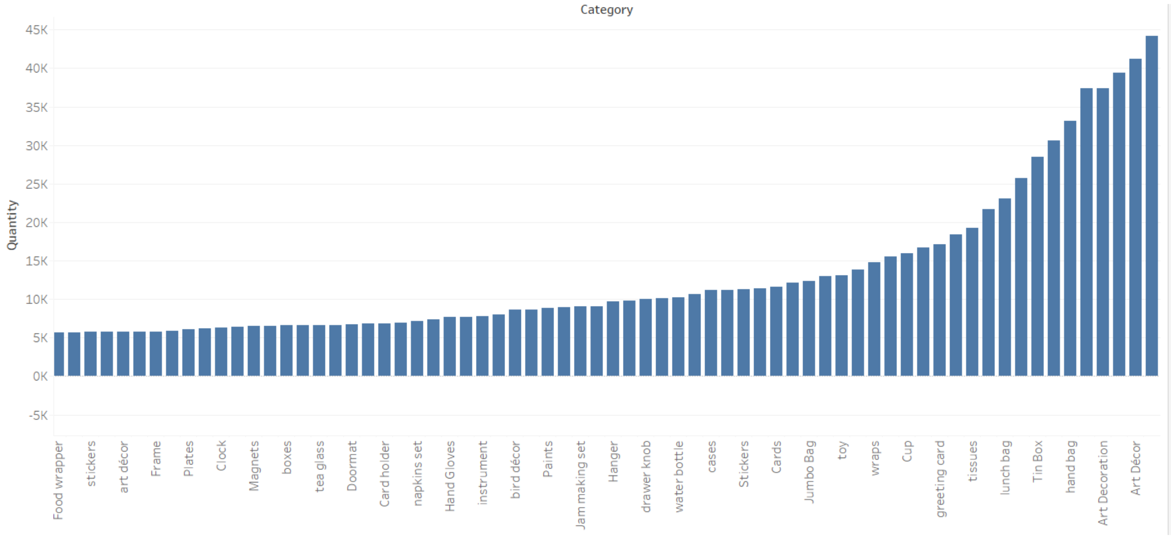


Figure 6.3: Bar graph plot representing quantity of each product sold on the whole.

4. The line graph is plotted with quantity of each category sold - month wise. Hence it can be used to visualize the sales trends of each category, in each

season or quarter, and stocks can be accordingly ordered.

Certain categories can be termed regular and certain can be predicted specifically based on the pattern hence generated.

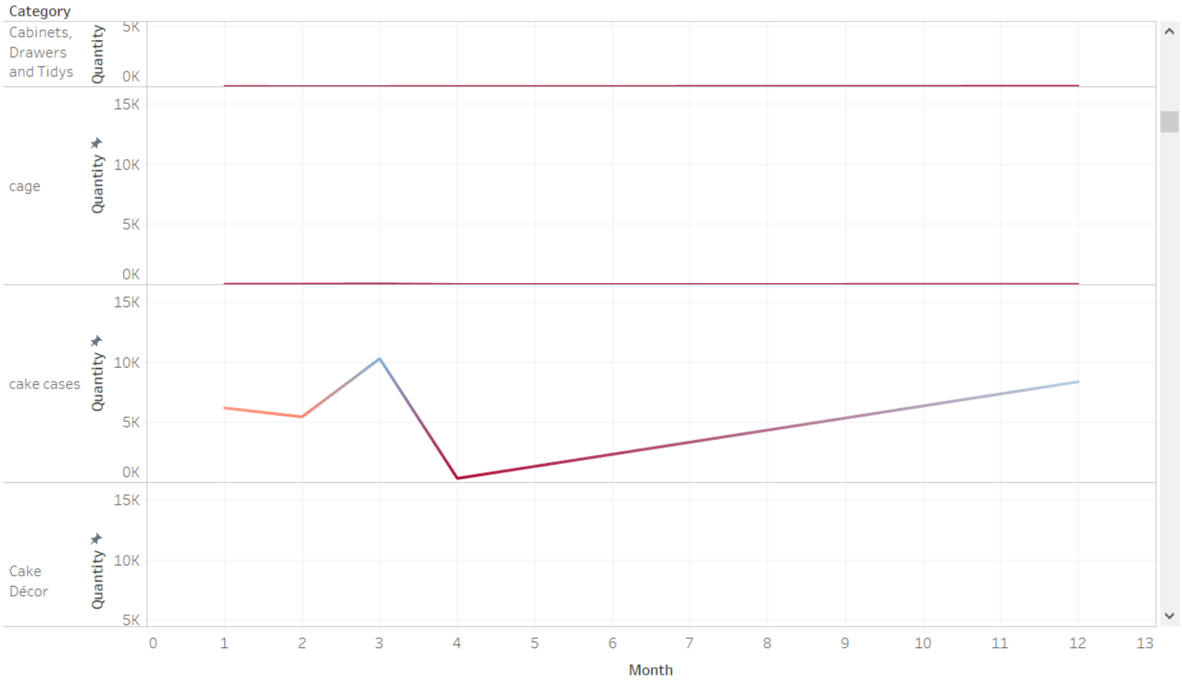


Figure 6.4: Line graph representing quantity of each category sold month-wise.

5. This line graph is a basic analysis of the sales made by the company in each month. This is for basic statistical output for the customers the get a general picture of the business and a comparison can be generated to show how the sales have been over the years. Also a seasonal or quarterly comparison can be made.

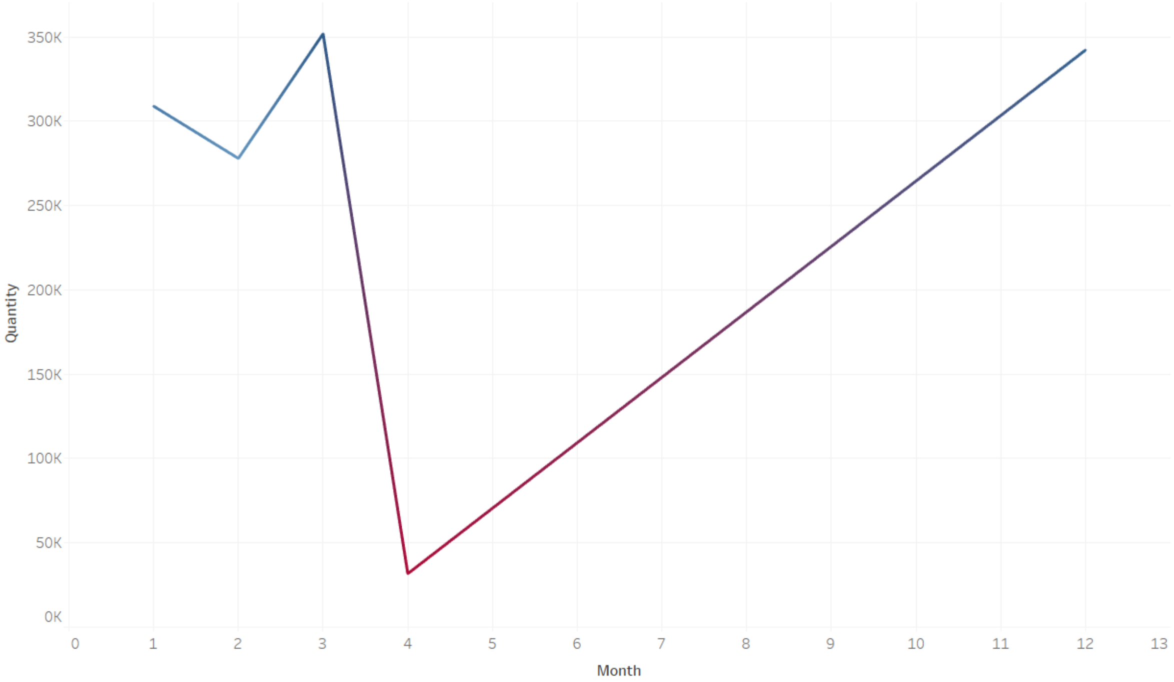
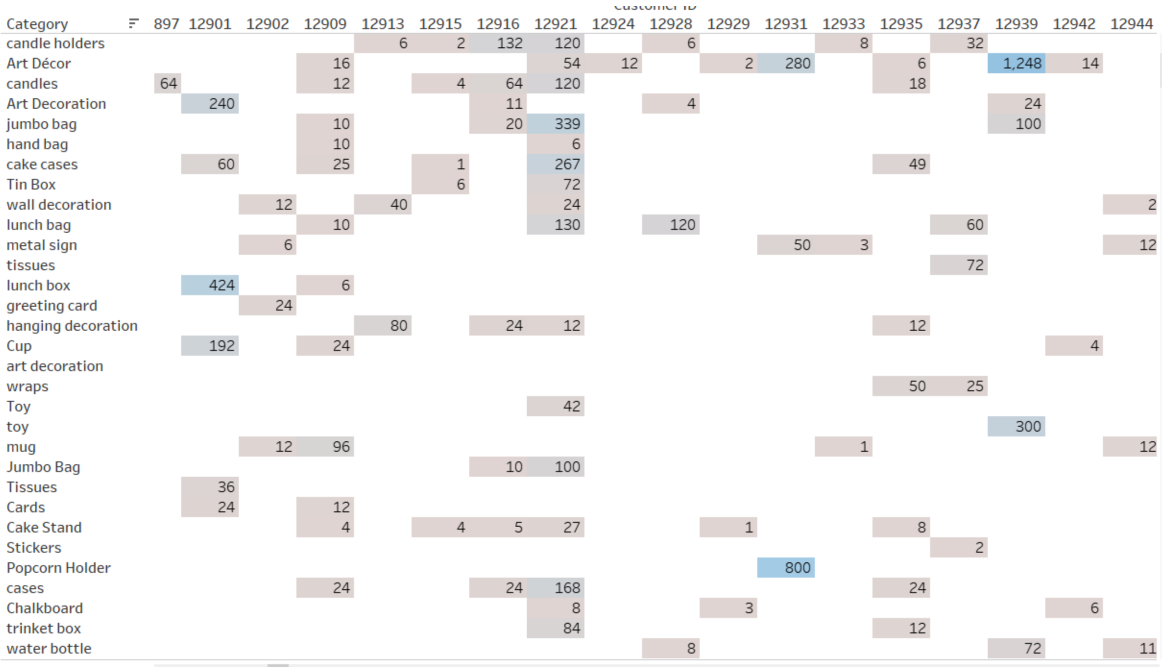


Figure 6.5: Line graph representing overall sales made month-wise.

6. This horizontal bars graph represents the quantity of each category bought by each customer. This helps in organizing sales and promotions for each category and also helps in personalized marketing for each customer. Frequent categories and rare items can also be found out using the same graph.



Category	897	12901	12902	12909	12913	12915	12916	12921	12924	12928	12929	12931	12933	12935	12937	12939	12942	12944
candle holders					6	2	132	120		6			8		32			
Art Décor				16				54	12		2	280		6		1,248	14	
candles	64			12		4	64	120						18				
Art Decoration		240					11			4						24		
jumbo bag				10			20	339								100		
hand bag				10				6										
cake cases		60		25		1		267						49				
Tin Box						6		72										
wall decoration			12		40			24										2
lunch bag				10				130		120					60			
metal sign			6									50	3					12
tissues															72			
lunch box		424		6														
greeting card			24															
hanging decoration					80		24	12						12				
Cup		192		24													4	
art decoration																		
wraps														50	25			
Toy								42										
toy																300		
mug			12	96									1					12
Jumbo Bag							10	100										
Tissues		36																
Cards		24		12														
Cake Stand				4		4	5	27			1			8				
Stickers															2			
Popcorn Holder												800						
cases				24			24	168						24				
Chalkboard								8			3						6	
trinket box								84						12				
water bottle										8						72		11

Figure 6.6: Horizontal bars graph representing quantity of each category bought by each customer.

7. This Pie chart represents the quantity of items sold in each country as a comparative share in the pie chart.

   It represents which country has a bigger share in the business, and which market is gaining the biggest revenue. Products of those regions can be focused and new items can be tested in those areas first to see the market reaction.

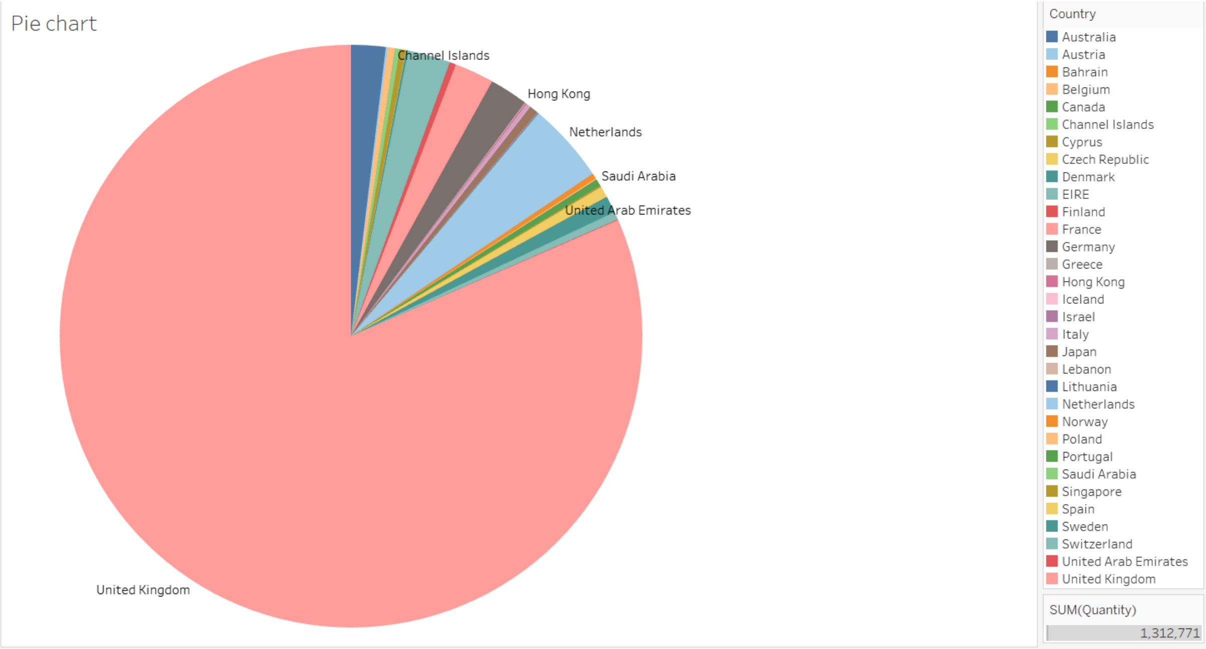


Figure 6.7: Pie chart representing proportion of each country in the business done world wide.

6.3 Summary

Today, Tableau has become the number one data visualization and business intelligence tool available in the market. It is used by maximum people to derive valuable insights from their data. The great advantage that tableau offers is that there is no need for the users to work with the powerful tool to have any sort of technical knowledge or programming skills. These aforesaid visualizations helped in finding future scopes and draw deeper insights into the data set, which will be helpful in drawing the accurate results from the data models and further compare results with the current results.

References

[1] *Analytics for an online retailer: Demand forecasting and price optimisation*; Ferreira,Kris J., Bin Hong Alex Lee, and David Simichi-Levi

[2] *Managing editor and resident data scientist, Reinventing the retail industry through machine and deep learning*; Daneil D. Gutierrez

[3] *A hybrid intelligent model for medium-tern sales forecasting in fashion retail supply chains using extreme learning machine and harmony search algorithm*; Wong,W.K.,Guo, Z.X.

[4] *Fuzzy Regression Model*; Arnold F. Shapiro

[5] `http://www.stat.cmu.edu/~cshalizi/350-2006/lecture-10.pdf\`

[6] `https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/\PrincipalComponentsRegression.pdf\`

[7] `https://www.neuraldesigner.com/blog/retail-store-sales-forecasting`

`https://www.techopedia.com/definition/14650/data-preprocessing\`