Prediction of Windspeed, Hotel Booking Demand and Credit Card Customer Segmentation using Machine Learning Algorithms

Gollamudi Sarath Chandra
School of Computing
National College of Ireland
Dublin, Ireland
x19193581@student.ncirl.ie

***Abstract*—In this paper, the machine learning algorithms Multiple Linear Regression, Support Vector Regression, Decision Tree, Naïve Bayes, and K-Means clustering using Principal Component Analysis (PCA) were implemented on three datasets and the results were compared. At first, for the wind speed prediction regression analysis has performed, and the results showed that multiple linear regression is giving a good RMSE value of 5.696 with a significant p-value than the SVM which has the RMSE value of 5.739, the lower the RMSE the better the model. Moreover, SVM requires more time to execute than the linear regression algorithm. Secondly, for the hotel booking demand, classification analysis has performed and the results showed that Decision Tree algorithm gives the best accuracy value of 79.66% with good precision, sensitivity and specificity percentages in other words with a well-balanced data than the Naïve Bayes with the accuracy of only 76.68% the higher the accuracy the better the model. Thirdly, for the credit card customer segmentation K-Means clustering algorithm has applied by using the dimensionality reduction PCA technique, and the clusters are measured by the total sum of squares with the K value 3 by the traditional elbow method.**

Keywords—Multiple Linear Regression, Support Vector Machine, Decision Tree, Naïve Bayes, and K-Means clustering with PCA.

I. INTRODUCTION

In today's world, machine learning is evident as one of the most influential technology for turning information into knowledge. In other words, finding the underlying pattern from the complex data to predict future events for complex decision making. As part of this, there are multiple forms of machine learnings which are supervised, unsupervised and semi-supervised with different approaches. In this report supervised and unsupervised learnings were covered.

II. OBJECTIVE

The main objective of this report is by using regression, classification and clustering algorithms on the datasets weather forecasting, hotel booking, and credit card customer transactions to predict and segregate the hidden data for making the right decision of future events. In the following sections, how this has been done and how the algorithms performed are explained clearly.

III. RELATED WORK

A. Dataset-1 (Windspeed Prediction)

With the decline of the environment and deficiency of conventional resources, renewable energy resources have been increasing. As a kind of non-pollution energy wind energy is increasing rapidly. In other words, generating wind energy with the help of wind turbines depends on wind speed. Below are the research papers which were done as part of this.

In this paper [1] M. A. MOHANDES et al. built a model predicting the wind speed using Artificial Neural Networks and Autoregressive modelling on weather forecasting data. The results showed that RMSE value of the neural network model is 1.87 whereas the AR model is of 2.88 and it says that the Neural Network had the best performance than the AR.

In an earlier study, the electronic system which was developed by Njau for predicting the wind speed patterns, it shows that there is a good agreement between the predicted and observed values [2].

Lalarukh and Yasmin in a statistical way of analysis by using an ARIMA model on the wind speed data of more than 2 years show that the forecast values of variance with a confidence interval of 95% can be acceptable for both short-term and long-term prediction [3].

In the paper [4] T.O Halawani with the help of his fellow colleagues performed a research on the prediction of wind by using the dataset which consists of more than 12 years data with the algorithms Support Vector machine and Multilayer perception, and the results showed that SVM performs well with a lowest MSE value of 0.0078 than the MLP on all orders with an MSE value of 0.0090.

In this paper [5] the average windspeed was forecasted using the fractional-ARIMA models. The reason for selecting this model is to incorporate long-range correlations that usually exist in the wind speed records. After a proper analysis with the model, the results showed good accuracy of the forecasting measured by the RFMSE value of 42%.

Sideratos G, Hatziargyriou ND in their paper [6] presents an advanced statistical method for wind power forecasting which depends critically on the volatility of wind. As part of their analysis, they have chosen the meteorological forecasts of wind speed. By using a combination of neural networks and fuzzy logic techniques on the data, the results showed that for each class the RBf network provides a different preliminary prediction. Later this is compared with the output of two RBf networks which are obtained by the NWPs between turbine power vs wind speed curve. This helps the fuzzy model to provide a quality indicator of wind speed, relating them to the wind direction. With this proposed method the model is succeeding for prediction and can be used effectively for operational planning in 1-48h ahead.

Forecasting the windspeed accurately is critical to the effective result of wind energy and the integration of wind power into the electric power grid. In this paper [7] Junyi Zhou, Jing Shi, and Gong Li used Least-Squares Support Vector Machine(LS-SVM) for one-step-ahead wind speed forecasting, considering the SVM parameters regularization and kernel. The results have shown that LS-SVM performs well at prediction values over the persistence model, where the RMSE values of LS-SVM are around 0.96 to 0.97 while that from the persistence model is 1.584.

B. Dataset-2 (Hotel Booking Demand)

In industries like tourism and travel, most of the research is carrying out on the Revenue management demand forecasting and prediction problems from the aviation industry in the format known as the Passenger Name Record (PNR). However, the remaining industries like hospitality, theme parks, etc., have different requirements. There is some research done on these industries by taking datasets with demand data to help in overcoming this limitation, and this helps in improving the hotel revenue management. Below are some papers which discussed the forecasting results.

Misuk Lee in this paper [8] develops a stochastic approach to the short-term forecasting of hotel bookings. On the other hand, by observing the strong correlations and using the Poisson mixture models and Naïve model the research has performed. The results have shown that the Mean Relative Absolute Error (MRAE) and Geometric Mean Relative Absolute Error (GMRAE) values are around 0.809 (80.9%) and 0.191 (19.1%) respectively. This means that it was concluded that Poisson Mixture models outperform the Naïve model as well as the standard NHPP model.

Dolores Romero and Morales Jingbo Wang in the paper [9] have described the forecasting rates for services of hotel revenue management using the machine learning algorithms. The following are the methods applied SAVG, AVG, Decision Tree, Random Forest, Support Vector Machine, and Kernel Logistic Regression (KLR). The forecasting results of these shows that SVM gives the smallest mean of SUM among the remaining algorithms. On the other hand, LR is the least accurate with reducing the error of AVG by 19.2% while other methods 22% (2.8%) more on average.

The uncertainty in the dynamics is characterised by their economic system which leads to taking bad decisions that affect the financial terms. This paper [10] explains the forecasting of the uncertain hotel room demand for each arrival date using the Holt-Winters method. These forecast helps the hotel management when to make rooms available and how revenue management improving. The forecasting has been divided into two parts Long-Term Forecasting (LTF) and Short-Term Forecasting (STF). After the stringent analysis, the results have shown that LTF when the arrival day is far, gives a very high estimation demand value before the arrival day. Eventually, the combination of both LTF and STF gave the right estimation output.

The important step in the decision-making process of hotel managers is occupancy rate forecasting, without accurate forecasting that might impact hotel financial performance. In this paper [11] machine learning algorithms performance was compared in order to find the best model. The algorithms are Ridge Regression, Kernel Ridge Regression, Multilayer Perception, and Radial Basis Function Networks using a time series data. Ridge regression model with quadratic features performs well than the other models, with a validation set MAPE of 8.2012% and a test set MAPE of 8.6561%.

With the web traffic volume data of an organisation to predict the hotel booking demand. In this [12] paper the authors have used the techniques ARMAX and compared with the ARMA counterparts. After a deep analysis in this the study has showed that ARMAX performs good with an average MAPE of 7.43% for four weeks ahead and 10.60% for eight weeks ahead over the ARMA model.

C. Dataset-3 (Credit Cardholder Segmentation)

Classifying the customers into groups who are having similar preferences is called segmentation. There are two types of technologies in the credit card industry which are market development and customer churn prediction model, below papers, have represented some of the important information about these segmentations.

Segregating the raw data into a useful and understandable way by using clustering methods is a statistical technique. In this paper [13] customers have been segregated into different groups using cluster analysis (k-means), and ANOVA analysis to test the stability of the clusters. The results have shown that after the number of clusters identified which is K = 4 for computation. The final cluster centres contained the mean values and interpreted in multi-dimensional related to market forecasting and planning, subsequently by using the ANOVA method null hypothesis is tested.

In order to improve customer service, the customer's behaviour must be identified. The paper [14] presents the rigorous clustering for customer segmentation and profiling using the algorithms K-Means and Density-Based Method (DBSCAN). At first, using the k value of 4 applied on the dataset and it has shown the WCSS value of 238.476. On the other hand, in order to remove the outliers which are existing and to improve the customer service DBSCAN method has used and the result showed a value of 278.

The banking system is very important in our daily life, the same for the bank as well where the customers are key factors for prosperity and development. In the paper [15] customer segmentation has done using the Principal Component Analysis (PCA), fuzzy k-means, and fuzzy k-medoids algorithms. The results have shown that k-medoids clustering performs very well with the threshold value of 0.24 than the other algorithms and finally, PCA correlates all the results.

In recent years credit card business is one of the important business for commercial banks. This paper [16] presents the customer segmentation using the algorithms K-means, FCM, MAJ, and a two-step model based on FSGA-FCEN which is an efficient and practical tool for customer segmentation. The results have shown that k-means performs well with an ocq value of 0.7823 then the other models.

The common tool for market segmentation is cluster analysis. In the paper [17] it aims to discuss the possibility by integrating ANN and Multivariate analysis which involves k-means and Ward's method to find the final solution. After a proper analysis of the data the results have shown that the k-means method performs well than the remaining methods with the prediction result of 99.17% which is followed by the self-organizing feature maps. On the other hand, the result of Ward's method is at 88.14%.

The problem that arises in business is market segmentation for which clustering techniques are used. This paper [18] explores the ability of the Frequency-Sensitive Competitive Learning Algorithm (FSCL) to cluster the data in comparison with the k-means algorithm. After the analysis of the data, the results of this paper states that the error level of the k-means case was significant at the 0.01 level ($p<0.001$). In order to this, the percentage of misclassifications was higher for FSCL which means the error levels were ranged from 7.28% to 13.78%. This means that the k-means method performs well over FSCL.

The many business competitors across the world are highly concentrated on gaining new customers and retaining old ones. This paper [19] explains a strategy for targeted customers using the K-means algorithm with the help of the MATLAB program. After the program is trained using a z-score on a two-feature dataset acquired from a retail business. The results have shown that the k-means algorithm has a purity measure of 0.95 indicating 95% accurate segmentation of the customers.

IV. METHODOLOGY

This data mining analysis has performed based on the KDD methodology which helps in finding the knowledge from data, pattern recognition, data visualization, and evaluation. Below is the overall process flow diagram in short which explains shows the step-by-step procedure.

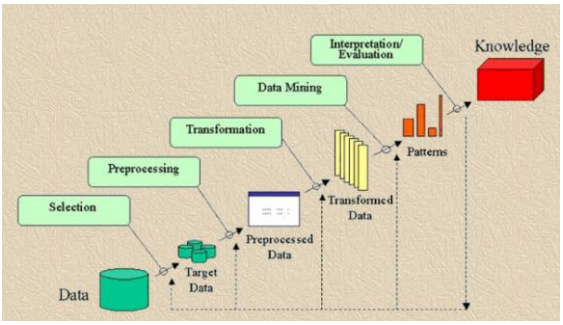


Fig-1: Steps of the KDD process

The same method has followed for all three datasets in this paper. Let us investigate the supervised and unsupervised learning analysis using the three datasets.

A. Dataset-1: Regression (Supervised Learning)

The objective of this regression analysis is to predict the windspeed with the weather forecasting data. As part of this, the two algorithms Multiple Linear Regression and Support Vector Machine (or Support Vector Regression) were used.

1) Data Selection:

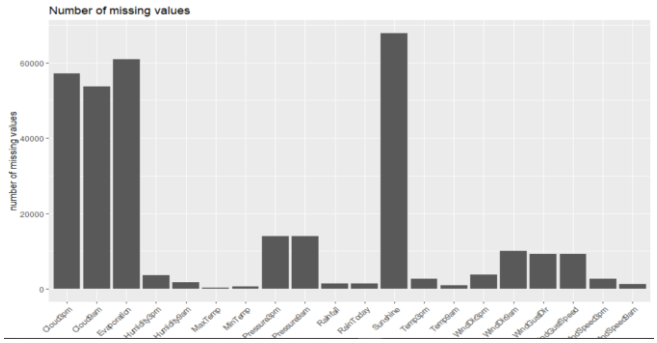
The dataset used for this analysis was taken from the Kaggle repository which is weather forecasting data of Australia and the goal of this analysis to predict the windspeed which is helpful for the wind turbines industry to generate renewable energy. The dataset contains data since 2007 which has 142194 records with 23 features.
https://www.kaggle.com/jsphyg/weather-dataset-rattle-package

2) Data cleaning and preprocessing:

Verifying the NA values: The verification has done by taking the missing percentages of each column, then the columns with a high percentage and which are not useful for the analysis were ignored. Below are the screenshot and graph of the missing values.

```
> windspeed3pm_naper <- data %>% summarise_each(list(~ sum(is.na(.)) / length(.) * 100))
> windspeed3pm_naper
  Date Location   MinTemp   MaxTemp  Rainfall Evaporation Sunshine WindGustDir WindGustSpeed WindDir9am
1    0        0 0.4479827 0.2264528 0.9887969    42.78903 47.69292    6.561504     6.519308   7.041838
  WindDir3pm WindSpeed9am WindSpeed3pm Humidity9am Humidity3pm Pressure9am Pressure3pm Cloud9am
1   2.656952    0.9480073    1.849599    1.2476    2.538803    9.855619    9.832411 37.73533
  Cloud3pm   Temp9am  Temp3pm RainToday RISK_MM RainTomorrow
1 40.15247 0.6357556 1.917113 0.9887969       0           0
```

Screenshot-1: Percentage of missing values of all columns



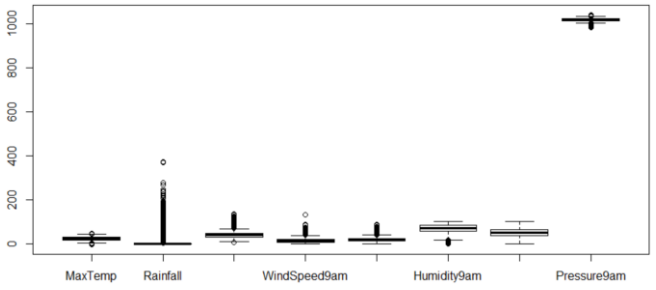
Graph-1: Missing values of the columns

From the graph above we could see that there are a lot of columns with missing values. So, to handle all these columns the NA values were replaced with the mean of the respective columns. Below is the screenshot of the count.

```
> for(i in 1:12){
+   windspeed3pm_2[is.na(windspeed3pm_2[,i]), i] <- mean(windspeed3pm_2[,i], na.rm = TRUE)
+ }
> #Checking NA values count
> colSums(is.na(windspeed3pm_2)) # 0
     MinTemp      MaxTemp     Rainfall WindGustSpeed  WindSpeed9am  WindSpeed3pm   Humidity9am
           0            0            0            0             0             0             0
 Humidity3pm   Pressure9am  Pressure3pm      Temp9am      Temp3pm
           0            0            0            0             0
```

Screenshot-2: Count of the missing values

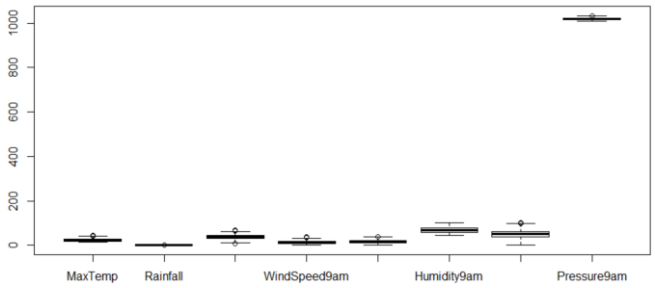
Verifying the Outliers: To find out the outliers in the data the boxplot has generated with all the columns and has seen that there are some outliers in the data. To handle these, using the traditional IQR formula the range has set between the upper and lower fence. Below are the screenshots of the boxplot and the output of the range values.



Boxplot-1: Outliers in the data

```
> Upfen_Humidity3pm <- 63.00+1.5*IQR_Humidity3pm
> Upfen_Humidity3pm #107
[1] 107
> summary(windspeed3pm_final1$Pressure9am)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  980.5  1013.5  1017.7  1017.7  1021.8  1041.0
> IQR_Pressure9am <- 1021.8 - 1013.5
> IQR_Pressure9am #8.3
[1] 8.3
> Upfen_Pressure9am <- 1021.8+1.5*IQR_Pressure9am
> Upfen_Pressure9am #1034.25
[1] 1034.25
> #Removing the outliers
> windspeed3pm_final_2 <- subset(windspeed3pm_final, MaxTemp<=43.65 & Rainfall<=2 & WindGustSpeed<=68.5 & W
indSpeed9am<=37 & WindSpeed3pm<=40.5 & Humidity9am<=122 & Humidity3pm<=107 & Pressure9am<=1034.25)
> View(windspeed3pm_final2)
Error in View : object 'windspeed3pm_final2' not found
> boxplot(windspeed3pm_final2)
Error in boxplot(windspeed3pm_final2) :
  object 'windspeed3pm_final2' not found
> boxplot(windspeed3pm_final_2)
> dim(windspeed3pm_final2)
Error: object 'windspeed3pm_final2' not found
> summary(windspeed3pm_final_2$MaxTemp)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -2.50   18.80   23.50   23.93   28.90   43.60
> IQR_MaxTemp <- 28.90 - 18.80
> IQR_MaxTemp #10.3
[1] 10.1
> Lofen_MaxTemp <- 28.90-1.5*IQR_MaxTemp
> Lofen_MaxTemp #12.75
[1] 13.75
```

Screenshot-3: Upper and Lower fence range of data



Boxplot-2: Final data without outliers

3) Data Transformation:

Collinearity Verification: In the above step we have removed some columns based on the analysis. In order to that, the correlation between the independent variables (IVs) has verified. Below is the screenshot of the correlation diagram.

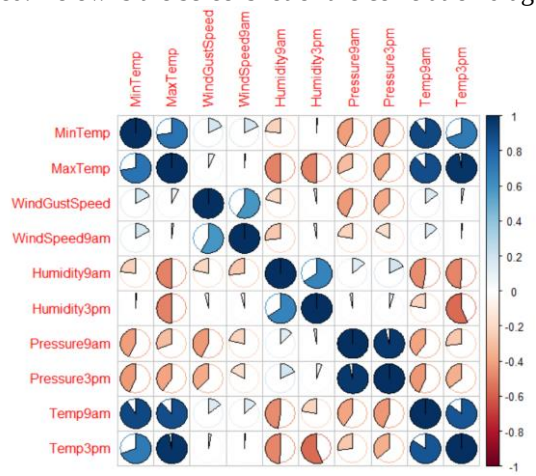


Fig-2: Correlation between IVs

From the above figure we could see that some of the variables are highly correlated (Pressure3pm-Pressure9pm, and MaxTemp, MinTemp, Temp9am, Temp3pm). This means that we can keep any one of those columns.

Representation of the final data: Below are the final columns and their respective datatypes with which algorithms are applied.

Feature	Description	Datatype
MaxTemp	The maximum temperature in degrees celsius	numeric
Rainfall	The amount of rainfall recorded for the day in mm	numeric
WindGustSpeed	The speed (km/h) of the strongest wind gust in the 24 hours to midnight	numeric
WindSpeed9am	Wind speed (km/hr) averaged over 10 minutes prior to 9am	numeric
WindSpeed3pm	Wind speed (km/hr) averaged over 10 minutes prior to 3pm	numeric
Humidity9am	Humidity (percent) at 9am	numeric
Humidity3pm	Humidity (percent) at 3pm	numeric
Pressure9am	Atmospheric pressure (hpa) reduced to mean sea level at 9am	numeric

Table-1: About Final Features

4) Evaluation with Data Mining Algorithms:

The algorithms Multiple Linear Regression and Support Vector Regression are used for the evaluation of the final data. As part of this analysis, the final data has divided into train and test dataset by taking the values randomly. The machine learning algorithms are applied on the train data and the output of this was compared with the test data to find the predicted vs actual values variance. Below is the output screenshot of the train and test data.

```
> set.seed(123) #To preserve the results every single time: without this, the data values split up 70/30% w
ill always be different
> indices <- sample(nrow(windspeed3pm_final2), 0.70 * nrow(windspeed3pm_final2))
> train_data <- windspeed3pm_final2[indices, ]
> test_data <- windspeed3pm_final2[-indices, ]
```

Screenshot-4: Train and Test data

Multiple Linear Regression Algorithm:

The multiple linear regression is nothing but predicting the dependent variable (DV) by using more than one independent variable (IV). In this analysis DV is windspeed3pm and the IVs are those in table 1. Firstly, this algorithm is applied on train data and below screenshots is the output of the algorithm.

```
> #Apllying Multiple Linear Regression Algorithm on the final data
> windspeed3pm_ml <- lm(WindSpeed3pm~., data=train_data)
> summary(windspeed3pm_ml) #R-squared:  0.4226

Call:
lm(formula = WindSpeed3pm ~ ., data = train_data)

Residuals:
     Min       1Q   Median       3Q      Max
-28.4733  -3.3088   0.3544   3.7726  20.9067

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -8.274969   5.713955  -1.448    0.148
MaxTemp        0.051458   0.005151   9.989  < 2e-16 ***
Rainfall      -0.344978   0.065669  -5.253 1.50e-07 ***
WindGustSpeed  0.423989   0.002910 145.705  < 2e-16 ***
WindSpeed9am   0.159345   0.003743  42.573  < 2e-16 ***
Humidity9am   -0.015382   0.002123  -7.244 4.42e-13 ***
Humidity3pm    0.059091   0.001704  34.681  < 2e-16 ***
Pressure9am    0.005110   0.005512   0.927    0.354
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.749 on 58603 degrees of freedom
Multiple R-squared:  0.4415,    Adjusted R-squared:  0.4414
F-statistic:  6617 on 7 and 58603 DF,  p-value: < 2.2e-16
```

Screenshot-5: Algorithm Output

We could see from the above results that the feature pressure9am is not significantly contributing to the DV and all other remaining columns are significant with a p-value <0.05 and the multiple R-squared values of 0.4415.

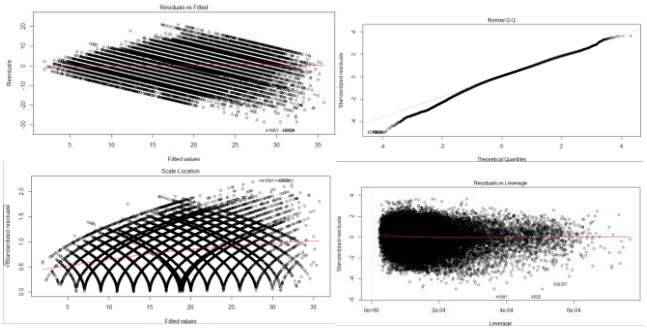
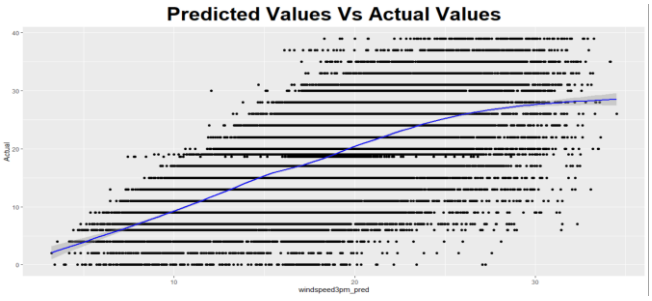


Fig-3: Residual Plots

Secondly, the output of the algorithm is applied to the test data to forecast the predicted values and to find the RMSE value of the algorithm. Below are the output and the graph of residual vs fitted values.

```
> rmsesvm = RMSE(windspeed3pm_pred, test_data$WindSpeed3pm)
> rmsesvm
[1] 5.696667
>
```

Screenshot-6: RMSE Value



Graph-2: Multiple Linear Predicted values vs Actual Values

From the above output console, we can see that the RMSE value of the Multiple Linear Regression algorithm is 5.696.

Support Vector Regression Algorithm:

The use of the SVR is to fit error within a certain threshold whereas in linear regression we try to minimize the error rate. This algorithm is also applied on the train and test dataset to find out the best algorithm. Firstly, below is the output screenshot of the algorithm.

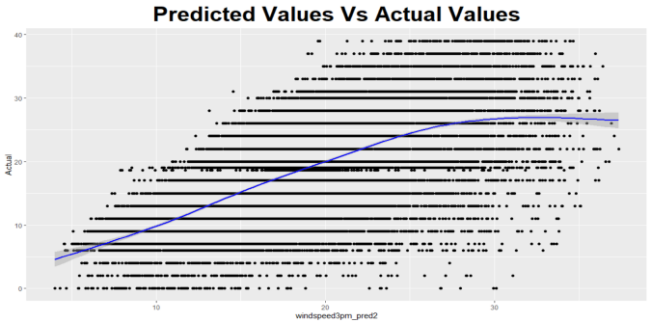
```
> windspeed3pm_svm <- svm(WindSpeed3pm~., data=traindata, kernel='linear', cost=1.0, epsilon=0.
1)
> summary(windspeed3pm_svm)

Call:
svm(formula = WindSpeed3pm ~ ., data = traindata, kernel = "linear", cost = 1,
    epsilon = 0.1)


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  linear
       cost:  1
      gamma:  0.1428571
    epsilon:  0.1


Number of Support Vectors:  51174
```

Screenshot-6: Algorithm Output

Secondly, the obtained output is applying to the test data to predict the forecasting values and to find out the RMSE value. Below are the output values and the screenshot of predicted vs actual values graph.

```
> rmsesvm = RMSE(windspeed3pm_pred2, testdata$WindSpeed3pm)
> rmsesvm
[1] 5.739581
```

Screenshot-7: RMSE Value



Graph-3: SVM Predicted values vs Actual Values

From the above output console, we can see that the RMSE value of the SVM algorithm is 5.739.

5) Interpretation of Results:

After observing both algorithm's output Multiple Linear Regression algorithm performs well over SVM with the RMSE values of 5.696 and 5.739. This means that the lower the RMSE value the better the model. On the other hand, the SVM algorithm took around 25mins to execute. It shows that SVM takes more time if the number of records is high. Considering all these aspects we are rejecting the SVM algorithm.

B. Dataset-2: Classification (Supervised Learning)

As mentioned above the same KDD methodology has followed for this analysis as well. The objective of this classification analysis is to predict the possibility of hotel booking between two hotels which are city hotel and resort hotel using the demand of the hotel. As part of this, the two algorithms Decision Tree and Naïve Bayes were used.

1) Data Selection:

The dataset used for this analysis was taken from the Kaggle repository which is hotel booking demand and the goal of this analysis is to predict the possibility of booking which helps the hotel management for analyzing the revenue and rooms available to customers. The dataset contains 119391 records with 32 features.
https://www.kaggle.com/jessemostipak/hotel-booking-demand

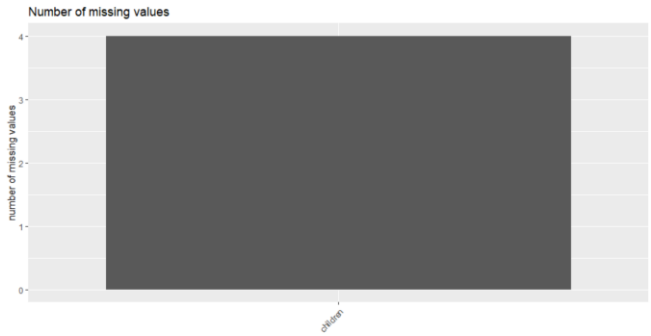
2) Data cleaning and preprocessing:

Verifying the NA values: The verification has done by taking the missing percentages of each column, then the columns with a high percentage and which are not useful for the analysis were ignored. Below are the screenshot and graph of the missing values.

```
> hotelbooking_per <- hotelbooking %>% summarise_each(list(~ sum(is.na(.)) / length(.) * 100))
> hotelbooking_per
  hotel iscanceled lead_time arrival_date_year arrival_date_month arrival_date_week_number
1     0          0         0                 0                  0                       0
  arrival_date_day_of_month stays_in_weekend_nights stays_in_week_nights adults   children
1                         0                      0                     0      0 0.003350364
  babies meal country market_segment distribution_channel is_repeated_guest
1      0    0       0              0                    0                 0
  previous_cancellations previous_bookings_not_canceled reserved_room_type assigned_room_type
1                      0                              0                 0                  0
  booking_changes deposit_type agent company days_in_waiting_list customer_type adr
1               0            0     0       0                    0             0   0
  required_car_parking_spaces total_of_special_requests reservation_status
1                           0                         0                 0
  reservation_status_date
1                       0
```

Screenshot-8: Missing Values Percentage

From the above output, we could see only one column is having with less percentage which is 'children'.



Graph-4: Missing values column

From the graph above we could see that there are four missing values. So, to handle all these, NA values were replaced with the mean of the respective column. Below is the screenshot of the count.

```
> #Replacing the NA with mean value
> hotelbooking$children[which(is.na(hotelbooking$children))] <- mean(hotelbooking$children, na.r
m = TRUE)
> colSums(is.na(hotelbooking)) #0
                          hotel                     iscanceled                      lead_time
                              0                              0                              0
              arrival_date_year             arrival_date_month        arrival_date_week_number
                              0                              0                              0
     arrival_date_day_of_month         stays_in_weekend_nights           stays_in_week_nights
                              0                              0                              0
                         adults                       children                         babies
                              0                              0                              0
                           meal                        country                 market_segment
                              0                              0                              0
           distribution_channel                 is_repeated_guest         previous_cancellations
                              0                              0                              0
previous_bookings_not_canceled                reserved_room_type             assigned_room_type
                              0                              0                              0
                booking_changes                   deposit_type                          agent
                              0                              0                              0
                        company                 days_in_waiting_list                  customer_type
                              0                              0                              0
                            adr       required_car_parking_spaces        total_of_special_requests
                              0                              0                              0
              reservation_status             reservation_status_date
                              0                              0
```

Screenshot-9: Count of the missing values

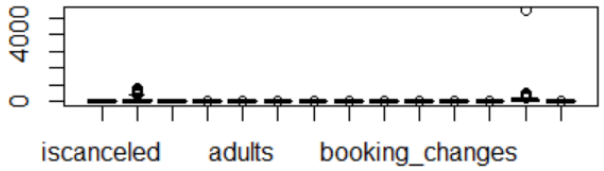
After replacing the NA values, the unwanted columns for this analysis have removed from the data and the outlier's verification has done on the remaining columns.

Converting all the character columns into factors:

```
> str(hotelbooking)
'data.frame':   119390 obs. of  26 variables:
 $ hotel                         : Factor w/ 2 levels "City Hotel","Resort Hotel": 2 2 2 2 2 2 2
 2 2 2 ...
 $ iscanceled                    : int  0 0 0 0 0 0 0 1 1 ...
 $ lead_time                     : int  342 737 7 13 14 14 0 9 85 75 ...
 $ arrival_date_year             : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
 $ arrival_date_month            : Factor w/ 12 levels "April","August",..: 6 6 6 6 6 6 6 6 6 6
 ...
 $ arrival_date_week_number      : int  27 27 27 27 27 27 27 27 27 27 ...
 $ arrival_date_day_of_month     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ stays_in_weekend_nights       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ stays_in_week_nights          : int  0 0 1 1 2 2 2 2 3 3 ...
 $ adults                        : int  2 2 1 1 2 2 2 2 2 2 ...
 $ children                      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ babies                        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ country                       : Factor w/ 178 levels "ABW","AGO","AIA",..: 137 137 60 60 60 6
0 137 137 137 137 ...
 $ market_segment                : Factor w/ 8 levels "Aviation","Complementary",..: 4 4 4 3 7 7
 4 4 7 6 ...
 $ is_repeated_guest             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_cancellations        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_bookings_not_canceled: int  0 0 0 0 0 0 0 0 0 0 ...
 $ reserved_room_type            : Factor w/ 10 levels "A","B","C","D",..: 3 3 1 1 1 1 3 3 1 4
 ...
 $ assigned_room_type            : Factor w/ 12 levels "A","B","C","D",..: 3 3 3 1 1 1 3 3 1 4
 ...
 $ booking_changes               : int  3 4 0 0 0 0 0 0 0 0 ...
```

Screenshot-10: Factors conversion

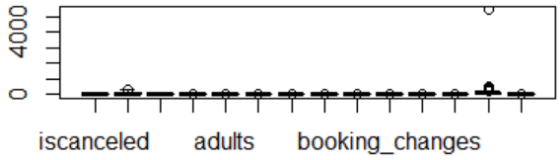
Verifying the Outliers: To find out the outliers in the data the boxplot has generated with all the columns and has seen that there are some outliers in the data. To handle these, using the traditional IQR formula the range has set between the upper and lower fence. Below are the screenshots of the boxplot and the output of the range values.



Boxplot-3: Outliers in the data

```
> #Removing Outliers
> boxplot(hotelbooking)
> summary(hotelbooking$lead_time)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0      18      69     104     160     737
> IQR_le_t <- 151-17
> IQR_le_t #134
[1] 134
> Upfen_le_t <- 151+1.5*IQR_le_t
> Upfen_le_t #352
[1] 352
> summary(hotelbooking$adr)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -6.38   69.29   94.58  101.83  126.00 5400.00
> IQR_adr <- 126.90 - 70
> IQR_adr #56.9
[1] 56.9
> Upfen_adr <- 126.90+1.5*IQR_adr
> Upfen_adr #212.25
[1] 212.25
```

Screenshot-11: *Upper fence range of data*



Boxplot-4: Final data without outliers

3) Data Transformation and EDA:

EDA: To know the insights present in the data, the below graphs were generated, and we could see the difference between the city and resort hotel bookings and cancelations.

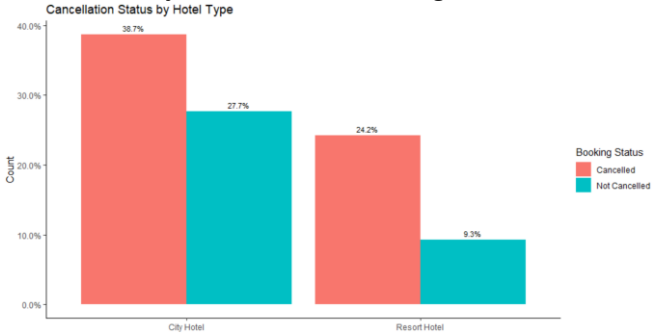


Fig-4: cancellation based on the hotel type

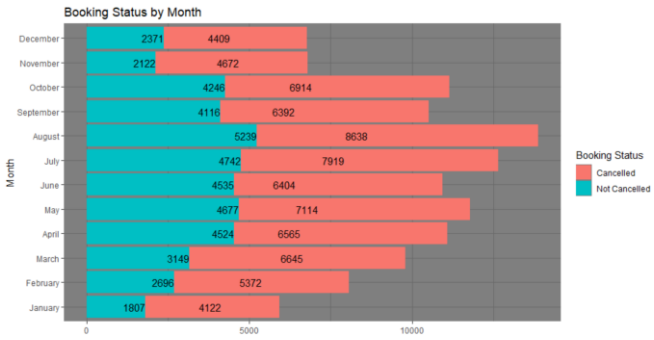


Fig-5: Month wise cancelations & booking count

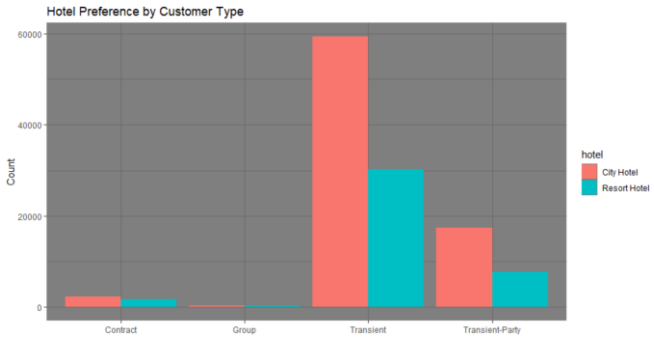


Fig-6: Hotel Preference by the customer type

Collinearity Verification: In the above steps we have removed some columns based on the analysis. In order to that, the correlation between the independent variables (IVs) has verified. Below is the screenshot of the correlation diagram.

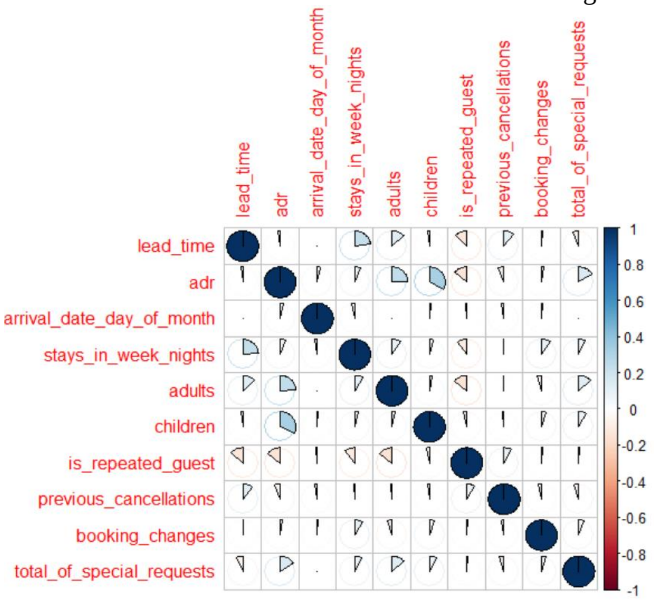


Fig-7: Correlation between IVs

From the above figure, we could see that there is no high correlation between the IVs. This means that we no need to remove any column and can apply the machine learning algorithms on this data.

Representation of the final data: Below are the final columns and their respective datatypes with which algorithms are applied.

Feature	Description	Datatype
iscanceled	Value indicating if the booking was canceled (1) or not (0)	integer
lead_time	Number of days that elapsed between the entering date of the booking into the PMS	integer
stays_in_week_n	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay a	integer
adults	Number of adults	integer
arrival_date_day	Day of arrival date	integer
children	Number of children	numeric
market_segment	Market segment designation. In categories, the term "TA" means "Travel Agents" and	factor
is_repeated_gue	Value indicating if the booking name was from a repeated guest (1) or not (0)	integer
previous_cancell	Number of previous bookings that were cancelled by the customer prior to the currer	integer
booking_changes	Number of changes/amendments made to the booking from the moment the bookin	integer
deposit_type	Indication on if the customer made a deposit to guarantee the booking. This variable	factor
adr	Average Daily Rate as defined by dividing the sum of all lodging transactions by the to	numeric
customer_type	Contract - when the booking has an allotment or other type of contract associated to	factor
total_of_special_	Number of special requests made by the customer (e.g. twin bed or high floor)	integer

Table-2: About Final Features

4) Evaluating with Data Mining Algorithms:

The algorithms Decision Tree and Naïve Bayes are used for the evaluation of the final data. As part of this analysis, the final data has divided into train and test dataset by taking the values randomly. The machine learning algorithms are applied on the train data and the output of this was compared with the test data to find the predicted vs actual values variance. Below is the output screenshot of the train and test data.

```
> #Splitting the final data into train and test datas
> set.seed(123) #To preserve the results every single time: without this, the data values split
 up 70/30% will always be different
> indices <- sample(nrow(hotelbooking), 0.70 * nrow(hotelbooking))
> train <- hotelbooking[indices, ]
> test <- hotelbooking[-indices, ]
```

Screenshot-12: Train and Test data

Decision Tree:

It is one of the predictive models in machine learning. This algorithm splits the data set based on several conditions and arrange the data in a tree structure. In this analysis, DV is 'iscanceled' and IVs are those shown in table 2. At first, this algorithm is applied to the train data, and below are the output screenshots.

```
> test <- hotelbooking[-indices, ]
> tree1 <- rpart(iscanceled ~ ., data=train)
> rpart.plot(tree1,nn=TRUE)
> summary(tree1)
Call:
rpart(formula = iscanceled ~ ., data = train)
  n= 75935

          CP nsplit rel error    xerror       xstd
1 0.20757440      0 1.0000000 1.0000427 0.001943743
2 0.03316389      1 0.7924256 0.7924537 0.002964565
3 0.02080224      3 0.7260978 0.7187811 0.003116255
4 0.01363950      5 0.6844933 0.6808901 0.003324333
5 0.01000000      6 0.6708538 0.6685134 0.003355120

Variable importance
            deposit_type           market_segment total_of_special_requests
                      52                       15                        9
           customer_type    previous_cancellations                lead_time
                       8                        6                        6
                     adr         is_repeated_guest                   adults
                       2                        1                        1

Node number 1: 75935 observations,    complexity param=0.2075744
  mean=0.3706591, MSE=0.2332709
  left son=2 (67494 obs) right son=3 (8441 obs)
  Primary splits:
      deposit_type             splits as  LRL,        improve=0.20757440, (0 missing)
      previous_cancellations   < 0.5    to the left,  improve=0.07128594, (0 missing)
      total_of_special_requests < 0.5    to the right, improve=0.06900580, (0 missing)
      lead_time                < 17.5   to the left,  improve=0.06437167, (0 missing)
      market_segment           splits as  LLLLRLL-,   improve=0.03584287, (0 missing)
```

Screenshot-13: Algorithm Output

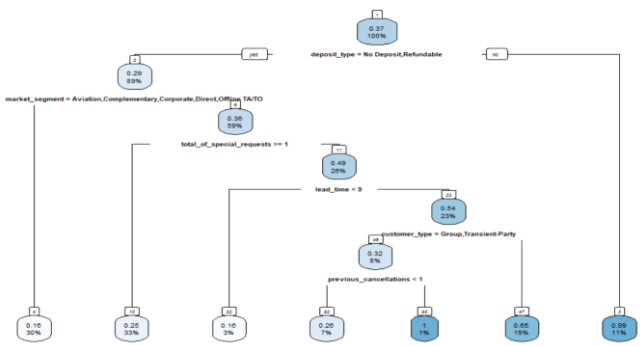


Fig-8: Tree plot of algorithm output

Secondly, the output of the algorithm is applied to the test data to find the **Accuracy, Kappa, Sensitivity, & Specificity** values of the algorithm. Since there is a small issue while generating the confusion matrix, I have used the table and calculated all the values using the formulas. Below are the outputs and the ROC graph.

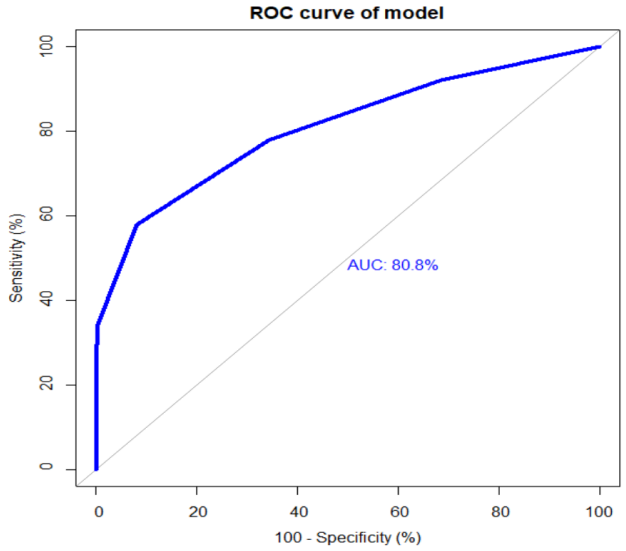
	accuracy1	kappa1	sensitivity	specificity	precision1	recall1
FALSE	0.7966004	0.5311103	0.9199242	0.5790061	0.7940466	0.9199242
TRUE	0.7966004	0.5311103	0.9199242	0.5790061	0.8038479	0.5790061

Screenshot-14: Accuracy, Sensitivity, Specificity output

From the above output we could see that the accuracy of the model is 79.66%. On the other hand, the sensitivity and specificity of the model are 91.99 and 57.9. This means that the data we have used for this analysis is a balanced one. To justify this, we will also check the ROC curve of this model. Following this, the kappa value of the model is 0.53.

ROC Curve:

It is used to measure the performance of this classification model. In this, there are two values we need to consider those are ROC and AUC. The probability curve is the ROC and the AUC tells how much the model is capable of distinguishing between classes. Here the higher the AUC the better the model is distinguishing between 'yes' and 'no' of the DV. Below is the output graph.



Graph-5: ROC and AUC of Decision Tree

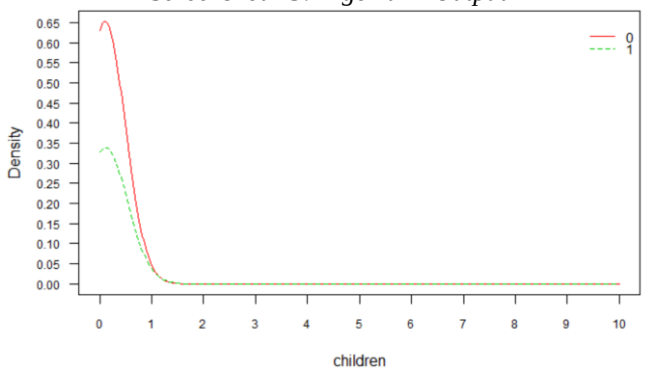
We could clearly see in the graph above; the AUC value is 80.8%. This means that our data is perfectly balanced.

Naïve Bayes:

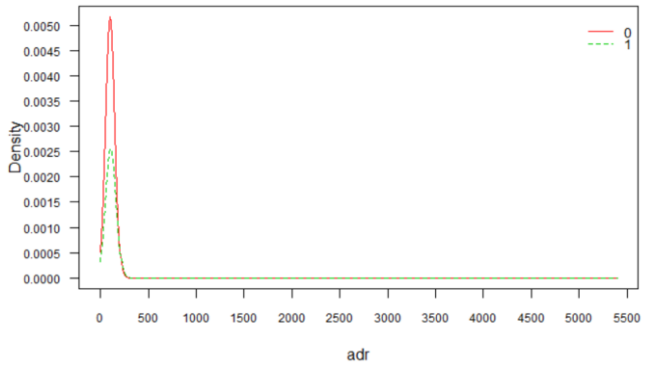
It is one of the classification techniques and this algorithm is based on the Bayes Theorem. This Naïve Bayes classifier assumes that the presence of a certain feature in a class is unrelated to any other feature. It is also applied on the train and test dataset to find out the best algorithm. Firstly, in order to execute this, all the integer variables are converted into factors, and below is the final output of the algorithm.

```
========================================= Naïve Bayes =========================================

 Call:
naive_bayes.formula(formula = iscanceled ~ ., data = train)

---------------------------------------------------------------------------------------------

Laplace smoothing: 0

---------------------------------------------------------------------------------------------

 A priori probabilities:

        0         1
0.6400904 0.3599096

---------------------------------------------------------------------------------------------

 Tables:

---------------------------------------------------------------------------------------------
 ::: lead_time (Categorical)
---------------------------------------------------------------------------------------------

lead_time            0            1
       0   8.086576e-02 1.022369e-02
       1   4.297922e-02 7.684918e-03
       2   2.575281e-02 4.974612e-03
       3   2.220336e-02 4.254151e-03
```

Screenshot-15: Algorithm Output



Graph-6: Algorithm plot of children categorical variable



Graph-7: *Algorithm plot of adr variable*

Like the above plots we have multiple plots for all the IVs, due to space concern all the plots are given in one fig as shown below.

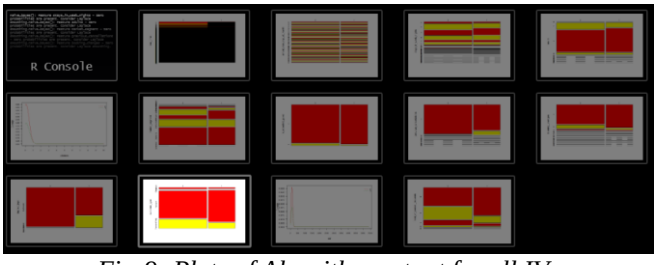


Fig-9: Plots of Algorithm output for all IVs

Secondly, the output of the algorithm is applied to the test data to predict forecasting values. In order to this, the confusion matrix table has generated to find out the accuracy, kappa, sensitivity, and specificity values. Below is the output screenshot.

```
> pre <- predict(hotelbooking_navie, test)
> confusionMatrix(table(pre, test$iscanceled))
Confusion Matrix and Statistics


pre     0     1
  0 20350  6290
  1  1804  6266

               Accuracy : 0.7668
                 95% CI : (0.7623, 0.7713)
    No Information Rate : 0.6383
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4526

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9186
            Specificity : 0.4990
         Pos Pred Value : 0.7639
         Neg Pred Value : 0.7765
             Prevalence : 0.6383
         Detection Rate : 0.5863
   Detection Prevalence : 0.7675
      Balanced Accuracy : 0.7088

       'Positive' Class : 0
```

Screenshot-16: confusion matrix table

In the above matrix table, we can see that the accuracy percentage of the algorithm is 76.68 and the corresponding kappa value is 0.45. On the other hand, the sensitivity and specificity of the model are 91.86 and 49.9. For Naïve Bayes we could not generate the ROC curve since the Naïve Bayes algorithm will take only factor and numeric variables as an input.

5) Interpretation of Results:

After observing both algorithm's output, the Decision Tree algorithm performs well over Naïve Bayes with the Accuracy values of 79.66 and 76.68. This means that the higher the Accuracy value the better the model.

C. Dataset-3: Clustering (Unsupervised Learning)

As mentioned above the same KDD methodology has followed for this analysis as well. The objective of this clustering analysis is to segment the customers based on their credit card usage and labeling the final clusters. As part of this, at first PCA technique was used which is dimensionality reduction technique on the dataset to reduce the IVs and the output of this is finally used for the clustering analysis with the K-Means algorithm.

1) Dataselection:

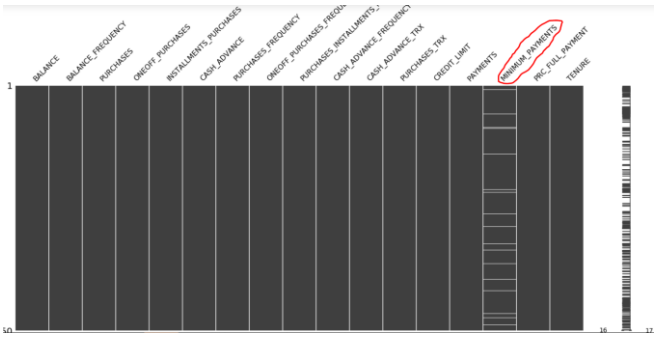
The dataset used for this analysis was taken from the Kaggle repository which is the credit card customer segmentation, and the goal of this analysis is to cluster the customers for analyzing the customer's behavior and offering the benefits or increasing the credit limit which will help to the growth of revenue of the banks. The dataset contains 9800 records with 18 features.
https://www.kaggle.com/arjunbhasin2013/ccdata

2) Data cleaning and preprocessing:

Verifying the NA values: The verification has done by taking the missing values of each column, then the columns with missing values were replaced with the mean of the respective column. Below are the screenshots and the graph of the missing values.

```
In [7]: #Verifying the missing values
        df.isnull().sum()

Out[7]: BALANCE                                  0
        BALANCE_FREQUENCY                        0
        PURCHASES                                0
        ONEOFF_PURCHASES                         0
        INSTALLMENTS_PURCHASES                   0
        CASH_ADVANCE                             0
        PURCHASES_FREQUENCY                      0
        ONEOFF_PURCHASES_FREQUENCY               0
        PURCHASES_INSTALLMENTS_FREQUENCY         0
        CASH_ADVANCE_FREQUENCY                   0
        CASH_ADVANCE_TRX                         0
        PURCHASES_TRX                            0
        CREDIT_LIMIT                             1
        PAYMENTS                                 0
        MINIMUM_PAYMENTS                       313
        PRC_FULL_PAYMENT                         0
        TENURE                                   0
        dtype: int64
```

Screenshot-17: Missing values count



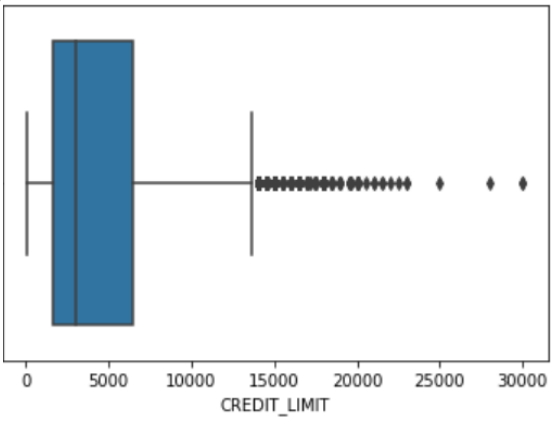
Graph-8: Missing values graph

```
In [10]: #No missing values in our data now
         df.isnull().sum()

Out[10]: BALANCE                                  0
         BALANCE_FREQUENCY                        0
         PURCHASES                                0
         ONEOFF_PURCHASES                         0
         INSTALLMENTS_PURCHASES                   0
         CASH_ADVANCE                             0
         PURCHASES_FREQUENCY                      0
         ONEOFF_PURCHASES_FREQUENCY               0
         PURCHASES_INSTALLMENTS_FREQUENCY         0
         CASH_ADVANCE_FREQUENCY                   0
         CASH_ADVANCE_TRX                         0
         PURCHASES_TRX                            0
         CREDIT_LIMIT                             0
         PAYMENTS                                 0
         MINIMUM_PAYMENTS                         0
         PRC_FULL_PAYMENT                         0
         TENURE                                   0
         dtype: int64
```

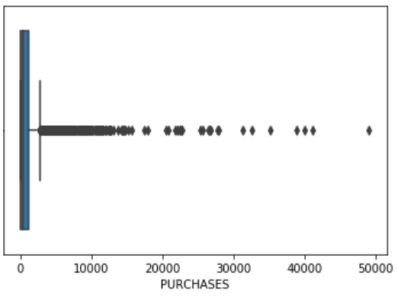
Screenshot-19: After replacing with Mean no NAs

From the above screenshot, we can see that there no missing values in our data now and later the outlier's verification has performed on this data.

Verifying the Outliers: To find out the outliers in the data the boxplot has generated using the seaborn for some columns and has seen that there are some outliers in the data. To handle these, using the traditional IQR formula the range has set. Below are the screenshots of the boxplot and the output of the range values.



Boxplot-5: Outliers of IV CREDIT_LIMIT



Boxplot-5: Outliers of IV PURCHASES

```
#IQR range to remove outliers
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3-Q1
print(IQR)

BALANCE                             1925.858120
BALANCE_FREQUENCY                      0.111111
PURCHASES                           1070.495000
ONEOFF_PURCHASES                     577.405000
INSTALLMENTS_PURCHASES               468.637500
CASH_ADVANCE                        1113.821139
PURCHASES_FREQUENCY                    0.833334
ONEOFF_PURCHASES_FREQUENCY             0.300000
PURCHASES_INSTALLMENTS_FREQUENCY       0.750000
CASH_ADVANCE_FREQUENCY                 0.222222
CASH_ADVANCE_TRX                       4.000000
PURCHASES_TRX                         16.000000
CREDIT_LIMIT                        4900.000000
PAYMENTS                            1517.858151
MINIMUM_PAYMENTS                     693.348888
PRC_FULL_PAYMENT                       0.142857
TENURE                                 0.000000
dtype: float64
```

Screenshot-20: Range of outliers

```
df = df[~((df < (Q1 - 1.5 * IQR)) |(df > (Q3 + 1.5 * IQR))).any(axis=1)]
```

Screenshot-20: IQR Range for removing the outliers

As shown in the above screenshot the range has set for the data and the outliers were removed.

3) Data Transformation:

Collinearity Verification: In the above steps we have removed the outliers and NAs. In order to that, the correlation between the independent variables (IVs) has verified. Below is the screenshot of the correlation diagram.

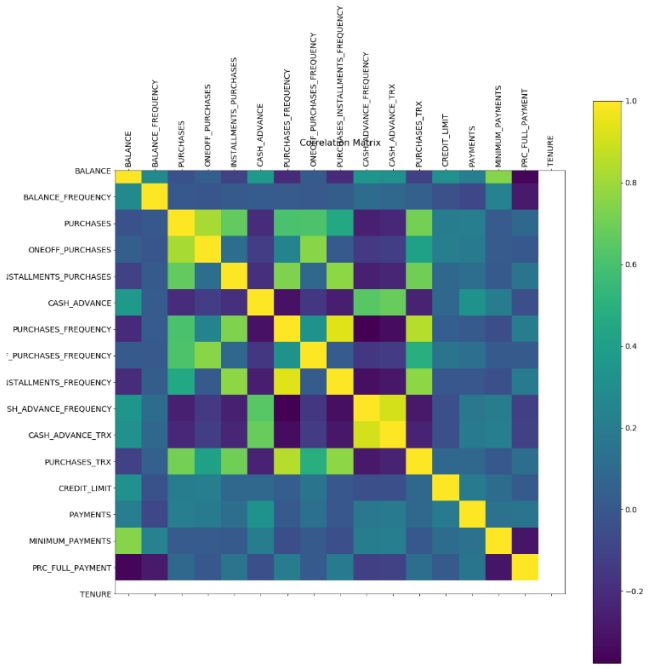


Fig-10: Correlation between IVs

From the above correlation plot, we can see that there is a high correlation between the variables purchases, purchases_ frequency, oneoff_purchases, cash_advance_frequency, cash_advance_trx, and purchases_installments_frequency. So, one variable was removed from the highly correlated.

Representation of the final data: Below are the final columns and their respective datatypes with which algorithms are applied.

Feature	Description	Datatype
BALANCE	Balance amount left in their account to make pu	float
BALANCE_FREQUENCY	How frequently the Balance is updated, score be	float
PURCHASES	Amount of purchases made from account	float
INSTALLMENTS_PURCHASES	Amount of purchase done in installment	float
CASH_ADVANCE	Cash in advance given by the user	float
PURCHASES_FREQUENCY	How frequently the Purchases are being made, s	float
ONEOFF_PURCHASES_FREQU	How frequently Purchases are happening in one	float
CASH_ADVANCE_TRX	Number of Transactions made with "Cash in Ad	integer
PURCHASES_TRX	Numbe of purchase transactions made	integer
CREDIT_LIMIT	Limit of Credit Card for user	float
PAYMENTS	Amount of Payment done by user	float
MINIMUM_PAYMENTS	Minimum amount of payments made by user	float
PRC_FULL_PAYMENT	Percent of full payment paid by user	float
TENURE	Tenure of credit card service for user	integer

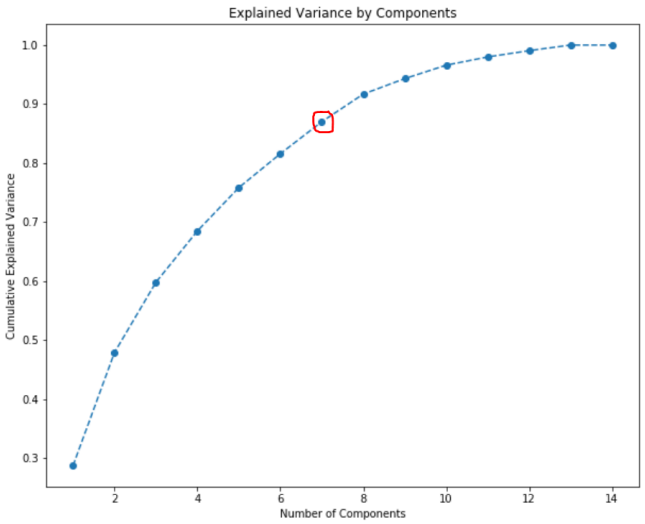
Table-3: About Final Features

4) Evaluating with Data Mining Algorithms:

The algorithm K-Means clustering with Principal Component Analysis (PCA) technique was used prior to data segmentation. As part of this analysis, the PCA technique has applied to the final dataset and the output of this was used for the K-Means clustering.

Principal Component Analysis (PCA):

It is a dimensionality reduction technique that can be used to reduce the number of features as well as the records so that the visualizing of segmentation will be more understandable. At first, all the data of the data frame was standardized in order to apply the PCA. Secondly, the number of components that we need to keep was finalized based on the cumulative variance plot. Below is the output screenshot.



Graph-9: Cumulative graph

The thumb rule is to preserve more than 80% of the variance. From the graph above, we can see that a total 7 components explaining the 80% variance. So, the PCA has performed with these chosen 7 components and then the final scores have taken for the further clustering analysis. Below is the output of the PCA.

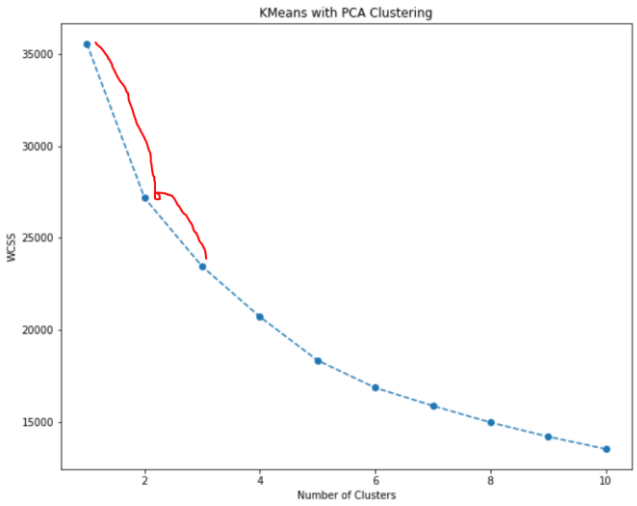
```
#Taking the calculated resulting components scores from our dataset
pca_scores = pca.transform(df_std)
pca_scores
```

```
array([[-0.61397124, -2.82977558,  0.22596338, ...,  0.34562928,
         1.81799651,  1.05655247],
       [-1.04407518, -1.45394666, -0.82184052, ..., -0.45399018,
        -0.02893586, -0.65528631],
       [ 1.23667279,  0.10315826, -1.20504418, ...,  1.06912205,
         0.22075893, -0.10619112],
       ...,
       [ 2.52014525, -2.75232847,  2.16147784, ...,  0.67741433,
         0.08907789, -1.10441771],
       [ 0.94501841, -2.73051765,  1.12051612, ...,  0.80074086,
        -0.91853027, -1.6489079 ],
       [ 1.25267635, -0.29669079,  0.13420901, ..., -0.85495727,
        -0.13522493,  0.39265172]])
```

Screenshot-21: Scores of PCA

K-Means Clustering:

K-Means clustering is one of the unsupervised learning algorithms. It is used to discover the underlying pattern in data by grouping similar points. In other words, it identifies the K number of centroids and allocates every point to the nearest centroid. At first, the algorithm is tested by taking the number randomly, and then the number of final clusters that we need to use for K-Means has decided using the **Elbow** method. Below is the output screenshot of the Within Cluster Sum of Squares (WCSS) plot.



Graph-10: WCSS Graph

From the graph above, we can tell that the maximum variance is explained in the first 3 clusters, and from the 4th cluster, it is much smoother. Based on this the K value 3 has taken and applied to the data. Below is the output screenshot.

```
df_seg_pca_kmeans
```

IES_TRX	CREDIT_LIMIT	...	PRC_FULL_PAYMENT	TENURE	0	Component 1	Component 2	Component 3	Component 4	Component 5	Component 6	Cluster
2	1000.0	...	0.000000	12	-0.613971	-2.829776	0.225963	0.594309	0.345629	1.817997	1.056552	2
1	1200.0	...	0.000000	12	-1.044075	-1.453947	-0.821841	0.300060	-0.453990	-0.028936	-0.655286	2
12	2300.0	...	0.000000	12	1.236673	0.103158	-1.205044	-0.912280	1.069122	0.220759	-0.106191	0
5	7000.0	...	0.000000	12	0.501062	-0.131656	-0.529873	1.298132	0.577022	-1.097185	0.347469	2
6	2000.0	...	0.000000	12	0.970665	-1.357372	0.730451	1.720365	-0.924208	1.925687	0.898234	2
...	...	...	...	...	...	...	...	...	...	...	...	...
1	1400.0	...	0.000000	12	-0.099165	-0.574565	-0.861116	0.639367	-0.825604	0.244802	-0.617941	2
12	1500.0	...	0.000000	12	1.941182	-1.249858	-0.852126	-1.497664	0.612388	-0.242982	0.316067	0
14	1000.0	...	0.333333	12	2.520145	-2.752328	2.161478	-0.795749	0.677414	0.089078	-1.104418	0
6	1000.0	...	0.300000	12	0.945018	-2.730518	1.120516	-1.006469	0.800741	-0.918530	-1.648908	2
12	1000.0	...	0.000000	12	1.252676	-0.296691	0.134209	-1.405147	-0.854957	-0.135225	0.392652	0

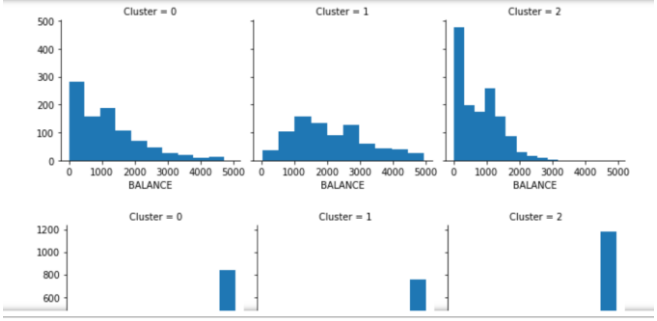
Screenshot-22: K-Means Clusters Output

In the above output, we could see that cluster numbers are assigned to all the records in the data frame.

5) Interpretation of Results:

The final three clusters are labelled based on the behaviour of all the IVs and the mean value of those respective columns. Below are the output screenshots.

```
#Interpretation of results
for c in df_seg_pca_kmeans:
    grid = sns.FacetGrid(df_seg_pca_kmeans, col='Cluster')
    grid.map(plt.hist, c)
```



Graph-11: Graphs of all IVs of 3 clusters

```
df_seg_pca_kmeans.groupby('Cluster').mean()
```

Cluster	BALANCE	BALANCE_FREQUENCY	PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEOFF_PURCHA
0	1162.364496	0.987098	1021.427784	444.614782	194.002230	0.803584	
1	2104.230134	0.988316	136.614284	38.612729	1152.262929	0.118829	
2	761.748012	0.969247	226.783397	84.976454	132.384308	0.298596	

Screenshot-23: Mean table of the clustering data frame

From the above mean table based on the purchases and credit limit column, all the clusters are labelled. Below are the outputs of all the components with labelling.

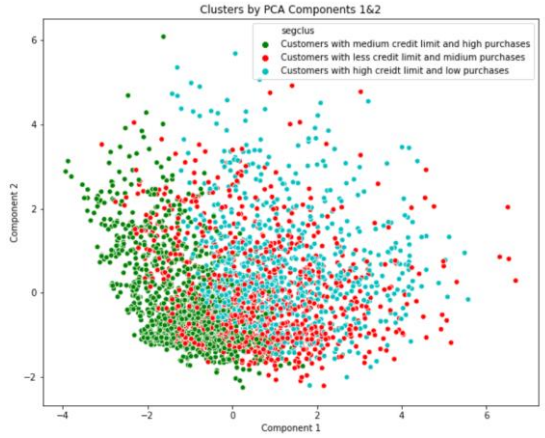


Fig-11: Clusters of Component 1 & 2

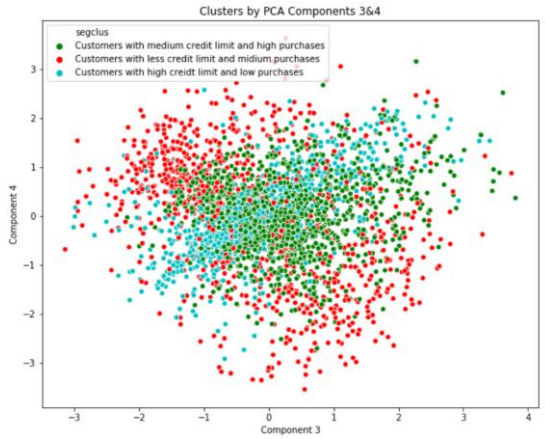


Fig-12: Clusters of Component 3 & 4

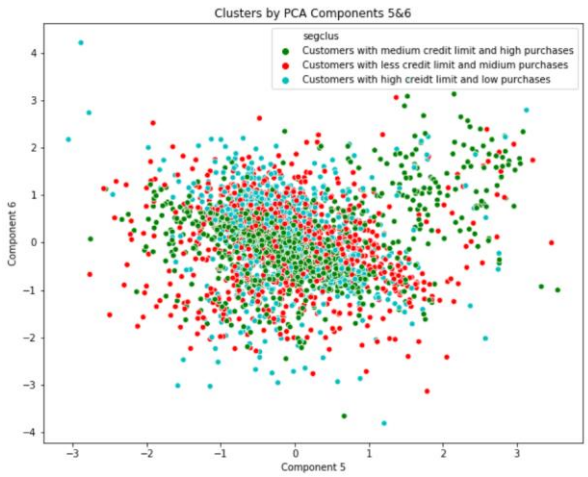


Fig-12: Clusters of Component 3 & 4

From the above all three figures we could clearly see the different types of customers based on the usage of their credit cards.

V. CONCLUSION AND FUTURE WORK

This research was performed on the six machine learning algorithms which cover both supervised and unsupervised learnings. As part of supervised learning, the regression and classification analysis has performed. At first, Among the regression algorithms, the Multiple Linear Regression performs well over SVR (SVM) with lower RMSE value for the 'windspeed prediction dataset'. The future work on this could be trying with the larger dataset to improve the performance of the SVM algorithm. Secondly, among the classification algorithms, the Decision Tree performs well over Naïve Bayes with higher accuracy for the 'Hotel demand dataset'. Future work could be improvising the decision tree algorithm by using boosting and bagging techniques. Thirdly, the unsupervised analysis has performed by the K-Means clustering algorithm with PCA technique on the credit card segmentation dataset and labelled the final clusters based on the usage of the credit cards. Future work could be analysed by taking the high volume of data to identify a greater number of clusters.

REFERENCES

[1] Njau, E. C., An electronic system for predicting air temperature and wind speed patterns. Renewable Energy, 1994, 4(7), 793-805.

[2] Mohandes M, Rehman S, Halawani TO. A neural networks approach for wind speed prediction. Renewable Energy 1998;13(3):345–54.

[3] Lalarukh K, Yasmin ZJ. Time series models to simulate and forecasthourly averaged wind speed in Quetta, Pakistan. Solar Energy1997;61(1):23–32.

[4] Mohandes MA, Halawani TO, Rehman S, Hussain AA. Support vectormachines for wind speed prediction. Renewable Energy 2004;29(6):939–47.

[5] K.-R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, V. Vapnik Predicting time series with support vector machines Advances in kernel methods—support vector learning, MIT Press, Cambridge, MA (1999), pp. 243-254

[6] Sideratos G, Hatziargyriou ND. An advanced statistical method for windpower forecasting. Power Systems 2007;22(1):258–65.

[7] J Zhou, J Shi, G Li - Energy Conversion and Management, 2011 – Elsevier.

[8] Misuk Lee, Modeling and forecasting hotel room demand based on advance booking information 2017.11.04 science direct

[9] Dolores Romero and Morales Jingbo Wang, Forecasting cancellation rates for services booking revenue management using data mining 2009.06.006 science direct

[10] Rajopadhye et al., 2001 M. Rajopadhye, M.B. Ghalia, P.P. Wang Forecasting uncertain hotel room demand Information Sciences, 132 (1) (2001), pp. 1-11

[11] Advances in Artificial Intelligence - IBERAMIA 2016, 2016, Volume 10022 ISBN : 978-3-319-47954-5 illiam Caicedo-Torres, Fabián Payares

[12] Yang Yang, Bing Pan, Haiyan SongFirst Predicting Hotel Demand Using Destination Marketing Organization's Web Traffic Data Published August 21, 2013

[13] KR Kashwan, CM Velu - International Journal of Computer Theory …, 2013 - researchgate.net

[14] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-hard Euclidean sum-of-squares clustering," Machine Learning, vol. 75, pp. 245-249, 2009.

[15] MS Kamal, MK Uddin - icmime-ruet.ac.bd

[16] W Gao, H Jia, R Yan - 4th International Conference on …, 2015 - atlantis-press.com

[17] Kuo et al., 2002 R.J. Kuo, L.M. Ho, C.M. Hu Integration of self-organizing feature map and K-means algorithm for market segmentation

[18] P.V.S. Balakrishnan, M.C. Cooper, V.S. Jacob, P.A. Lewis Comparative performance of the FSCL neural net and K-means algorithm for market segmentation EJOR, 93 (1996), pp. 346-357

[19] CP Ezenkwu, S Ozuomba, C Kalu - 2015 – Citeseer.