**Machine Learning & Data Mining with Python and Azure Machine Learning Studio **

**A Project report**

***Submitted by***

***ATITEBI ZAYNAB MORENIKEJI***

**(@00669741)**

***In fulfilment of***

**Machine Learning and Data Mining course work**

***For***

**MASTER OF SCIENCE**

***In***

**DATA SCIENCE**

**School of Computing, Science & Engineering**

**University of Salford, Manchester.**

**December, 2022**

**Table of Contents**

Task 1a: Classification with Python………………………………………………………………………….4

1. Introduction…………………………………………………………………………………………………4
   1. Aim and Objectives…………………………………………………………………………..4

2.0 Explanation and Preparation of Datasets……………………………………………………..4

3.0 Implementation……………………………………………………………………………………………8

4.0 Result Analysis and Discussion……………………………………………………………………..12

5.0 Conclusion……………………………………………………………………………………………………13

Task1b: Classification with Microsoft Azure Machine Learning Studio……………………..14

1.0 Introduction………………………………………………………………………………………………….14

2.0 Explanation and Preparation of Datasets………………………………………………………14

3.0 Implementation…………………………………………………………………………………………….14

4.0 Result Analysis and Discussion……………………………………………………………………….23

5.0 Conclusions and Recommendations……………………………………………………………….24

Task 2: Association Rules Mining………………………………………………………………………………..25

1. Introduction……………………………………………………………………………………………………25
   1. Aim and Objectives……………………………………………………………………………..25

2.0 Explanation and Preparation of Datasets………………………………………………………..26

3.0 Implementation………………………………………………………………………………………………27

4.0 Result Analysis and Discussion………………………………………………………………………..32

5.0 Conclusions…………………………………………………………………………………………………….32

Task 3: Clustering……………………………………………………………………………………………………….33

1.0 Introduction……………………………………………………………………………………………………33

2.0 Explanation and Preparation of Datasets………………………………………………………..33

3.0 Implementation………………………………………………………………………………………………34

4.0 Result Analysis and Discussion……………………………………………………………………..39

5.0 Conclusions………………………………………………………………………………………………….40

Task 4: Text Mining and Sentimental Analysis………………………………………………………….41

1.0 Introduction…………………………………………………………………………………………………41

2.0 Explanation and Preparation of Datasets…………………………………………………….41

3.0 Implementation………………………………………………………………………………………….43

4.0 Result Analysis and Discussion……………………………………………………………………44

5.0 Conclusion………………………………………………………………………………………………….44

References…………………………………………………………………………………………………………….46

**TASK 1a: CLASSIFICATION WITH PYTHON**

**Performance Evaluation of 2 Classification Algorithms on a Data Set**

**1.0 Introduction**

The Classification algorithm, a Supervised Machine Learning approach, is used to categorize new observations in light of training data. And in contrast to regression, classification produces a variable that is a category rather than a value. In classification, a program is trained using the available data set and then classifies additional observations into a number of classes or groups.

In this report, my aim is to classify a data set using 2 algorithms and then compare the performance of the 2 algorithms using some metrics. Here, python programming language was used and the F1 score metrics was considered majorly as we have an imbalanced class distribution.

**1.1 Aim and Objectives**

This report aims to evaluate the performance of K-Nearest Neighbor (KNN) and Neural Network (NN) Algorithms on a data set. To achieve this aim, the following objectives must be met.

1. Perform exploratory analysis of the data set
2. Categorize using KNN model.
3. Categorize using NN model.
4. Perform hyperparameter optimization of the models
5. Evaluate the performance of the 2 models.

**2.0 Explanation and Preparation of Datasets.**

The data set used in this project was downloaded from <https://archive.ics.uci.edu/ml/machine-learning-databases/blood-transfusion>. It is a transfusion data collected from a blood transfusion service center in Hsin-Chu City, Taiwan. The data set was suitable for classification problem because it has a target variable which states whether someone will donate blood in 2007 or not. This data set has 748 observations and 5 attributes. The first four attributes are the independent variables while the last attribute, the target variable is the dependent variable. Below is the definition of each variable accordingly.

**Table 1: Attributes Definition**

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **ATTRIBUTE** | **DATA TYPE** | **DEFINITION** |
| 1. | Recency (months) | Integer | Number of months since the last donation was done |
| 2. | Frequency (times) | Integer | Total number of times blood was donated |
| 3. | Monetary (c.c. blood) | Integer | Total amount of blood that was donated in cubic centimeter |
| 4. | Time (months) | Integer | Number of months since the first donation was done |
| 5. | Whether he/she donate blood in March 2007 | Integer | A binary number showing if the person donated in March 2007 or not. 1 signifies the person donated while 0 means the person didn’t donate. |

Exploratory analysis of the data set was carried out too. I described the data sets by checking the mean, standard deviations, minimum and maximum values of each attribute as seen below.

**Table 2: Descriptive Statistics**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Descriptive measures** | **Recency (months)** | **Frequency (times)** | **Monetary (c.c. blood)** | **Time (months)** | **whether he/she donated blood in March 2007** |
| **count** | 748.000000 | 748.000000 | 748.000000 | 748.000000 | 748.000000 |
| **mean** | 9.506684 | 5.514706 | 1378.676471 | 34.282086 | 0.237968 |
| **std** | 8.095396 | 5.839307 | 1459.826781 | 24.376714 | 0.426124 |
| **min** | 0.000000 | 1.000000 | 250.000000 | 2.000000 | 0.000000 |
| **25%** | 2.750000 | 2.000000 | 500.000000 | 16.000000 | 0.000000 |
| **50%** | 7.000000 | 4.000000 | 1000.000000 | 28.000000 | 0.000000 |
| **75%** | 14.000000 | 7.000000 | 1750.000000 | 50.000000 | 0.000000 |
| **max** | 74.000000 | 50.000000 | 12500.000000 | 98.000000 | 1.000000 |

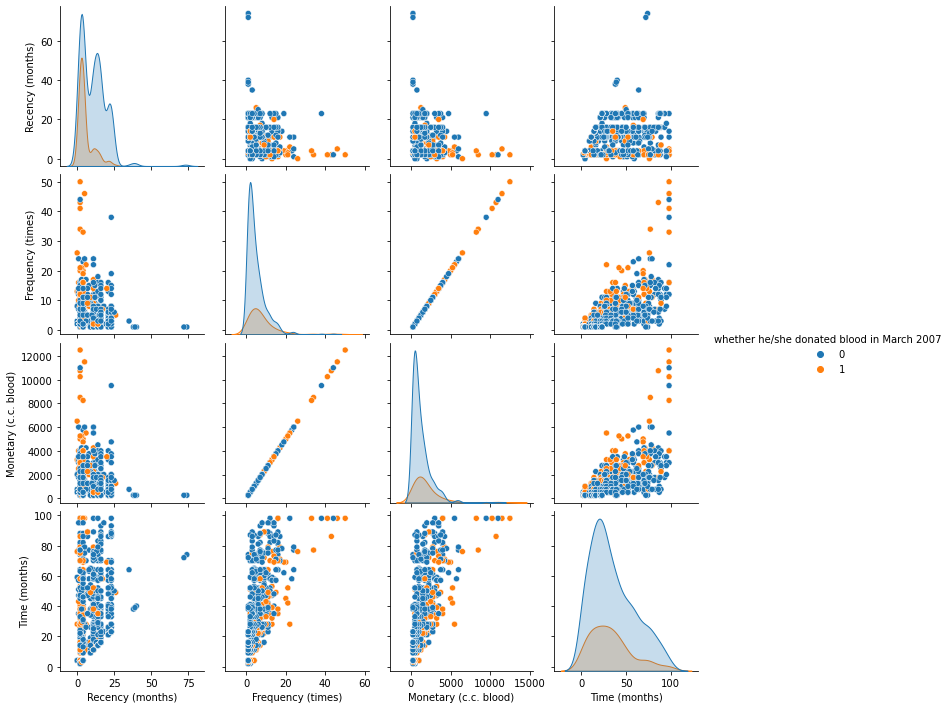
In addition to this, I confirmed the number of null observations, and then made a histogram plot, box plot and violin plot of all the independent variables. However, I will only be reporting the box plot as it reveals the mean and the outliers in the data set.

Graphical user interface, diagram

Description automatically generated

**Figure 1: Box plot of the independent variables.**

To see the relationship between the independent variables, I made a scattered plot of the variables while using the target as hue to differentiate the points (Figure 2). Also, I explored the dependent variable by counting the number of those who donated and those who do not donate blood. In addition to this, I normalized the variable and then plotted a pie chart to see the percentage of those who donated blood from those who didn’t donate (Figure 3).

****

**Figure 2: Scattered plot matrix.**

Chart, pie chart

Description automatically generated

**Figure 3: Chart of dependent variable**

**3.0 Implementation**

Classification algorithms can be divided into 2 major categories, the Liner models, and the non-linear models (Java T point, 2022). The liner models are the logistic regression and support vector machines while the non-linear models are KNN, Kernel SVM, Naïve Bayes, Decision tree classification and Random Forest classification. In this project, I considered the K-nearest neighbor and Neural network.

According to Knocklein (2019), Neural Networks are made up of an artificial network of parameters that the computer may learn from and fine-tune by examining fresh data. Each parameter, often known as a "neuron," is a function that outputs a result after taking one or more inputs. The following layer of neurons receives those outputs and uses them as inputs to perform its own function, producing more outputs. Once each layer of neurons has been considered and the input has reached the terminal neurons, their outputs are then passed on to the subsequent layer of neurons. The final output for the model is then produced by those terminal neurons. On the other hand, the KNN algorithm believes that related things are located nearby (Harrison, 2018). To put it another way, it assumes that related items are close to one another.

In a supervised machine learning, such as the classification algorithm, feature engineering needs to be performed on the data set. Hence, I started by defining my independent (X) variables and my dependent variable (y), then splitted the independent variables into train sets and test sets. Hence, I have I had 598 observations as the trained set and 150 observations as the test set. I then proceeded to using my model to predict the categories.

I tested four models all together in this project, however, I will be reporting only 2 of the models which is KNN and NN. I chose KNN over logistic regression and decision tree because KNN has a higher accuracy than the other 2 models. Then I compared it with the Neural network as it is a special type of classifier. While a Neural Network sets up algorithms so that it can make reliable decisions on its own, a Machine Learning model makes decisions based on what it has learnt from the data. Hence, the reason I am comparing it with a type of machine learning model (Goyal, 2022).

I started by importing the model and metrics from scikit-learn, then built a pipeline which houses the standardizer and the classifier model. Pipelining is crucial here because it expedites the process and aids in workflow organization. After this, I evaluated my model by computing its confusion matrix and then plotting it (figure 4). Confusion matrix, also known as error matrix, summarizes the results of the predictions, while including the total number of accurate and inaccurate predictions. And finally, I computed the report of the model (table 3).

A picture containing text, electronics, screenshot, display

Description automatically generated

**Figure 4: Confusion matrix for KNN Model**

**Table 3: KNN Metrics report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** | **Support** |
| **0** | **0.81** | **0.92** | **0.86** | **114** |
| **1** | **0.57** | **0.33** | **0.42** | **36** |
| **Accuracy** |  |  | **0.78** | **150** |
| **Macro avg** | **0.69** | **0.63** | **0.64** | **150** |
| **Weighted avg** | **0.76** | **0.78** | **0.76** | **150** |

Talking about the hyperparameter optimization, David (2020) stated that hyperparameters are various values used to regulate the learning process of an algorithm and have a big impact on how well machine learning model’s function. Finding the ideal combination of hyperparameter values to get the best performance on the data in the least amount of time is known as hyperparameter optimization. As such, I used the Grid search cross validation approach, as it is the most widely used approach for hyperparameter optimization. I specified a range between 1 and 25 and the number of jobs as -1 so that it can use all available processors to run the optimization. I also plotted the selected ranges against their accuracy, just as seen below.

**Chart, line chart

Description automatically generated**

**Figure 5: Cross validation result plot**

After the cross validation, I then used the best value of K to predict the model and printed the result.

**Table 4: Hyperparameter optimized report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** | **Support** |
| **0** | **0.82** | **0.94** | **0.87** | **114** |
| **1** | **0.63** | **0.33** | **0.44** | **36** |
| **Accuracy** |  |  | **0.79** | **150** |
| **Macro avg** | **0.72** | **0.64** | **0.65** | **150** |
| **Weighted avg** | **0.77** | **0.79** | **0.77** | **150** |

**A picture containing text, electronics, screenshot, display

Description automatically generated**

**Figure 5: CM for optimized model**

For the Neural network model, I started by installing and importing tensor flow, and then importing Keras from tensor flow. I also imported the layers and optimizers from Keras. It is important to note that Keras helps us builds prediction models.

Secondly, I defined the train and test set for the independent and dependent variables (X and y) and set a random seed number to have a producible result. I also defined my model by specifying the input layer and the hidden layer, then compiled the model (optimized during this process) and did a model fit while specifying the epochs and validation split. Then I evaluated the model and finally predicted it. At the long run, I was able to print out my report as shown below.

**Table 5: NN model result**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** | **Support** |
| **0** | **0.92** | **0.61** | **0.74** | **114** |
| **1** | **0.41** | **0.83** | **0.55** | **36** |
| **Accuracy** |  |  | **0.67** | **150** |
| **Macro avg** | **0.66** | **0.72** | **0.64** | **150** |
| **Weighted avg** | **0.80** | **0.67** | **0.69** | **150** |

Finally, I evaluated the 2 models by comparing the optimized results of the KNN and the NN models while specifying the accuracy, precision, recall and f1 scores.

**Table 6: comparison between KNN and NN models**

|  |  |  |
| --- | --- | --- |
|  | **KNN** | **NN** |
| **Accuracy** | **0.793333** | **0.666667** |
| **Precision** | **0.621579** | **0.405405** |
| **Recall** | **0.333333** | **0.833333** |
| **F1** | **0.436364** | **0.545455** |

**4.0 Result Analysis and Discussion**

The descriptive statistics on table 2 confirms that there are 748 observations in this data set. Picking the first attribute as a case study, it has a mean of 9.5, standard deviation of 8.09, minimum value of 0, and maximum value of 74 months. The Q1 of this attribute is 2.75, median is 7 and the Q3 is 14. This also applies to other attributes.

Figure 1 shows a box plot of each of the independent variable. It clearly shows the mean which is represented by the circle, and the outliers in the data set. It is obvious that only the time variable has no outlier, which means the number of months since the first donation was done all falls within a uniform range.

Figure 2 shows the relationship between each of the variables, and we can see that on figure 3, 76.20% of the observation didn’t donate blood in March 2007, while 23.80% of the observations donated blood in March 2007.

On figure 4, we can see the confusion matrix of the KNN model. Here, the **True negative** was 105, which means those that didn’t donate and were correctly predicted to have not donated. The **false positive** is 9, which means, those that didn’t donate but were falsely predicted to have donated. Also, the **False Negative** is 24, which means, those who donated but were false predicted to have not donated. And finally, we have the **True positive** to be 12, which means those who donated and were correctly predicted to have donated.

On table 3, we can see that the Accuracy of the model is 78%, but we can’t judge with this alone because we have an imbalanced class. And in this case, the F1 score is best in judging, which is the harmonic mean of the precision and recall. Hence, the F1 score for those who didn’t donate is 86% which is high compared to those who didn’t donate (42%).

After specifying some hyperparameters and then optimizing it, result shows that if the number of neighbors is specified as 24, we can have a better prediction. This can be seen on figure 5 as K=24 gives an accuracy of 79% compared to the accuracy of 78% that wasn’t optimized.

Also, table 4 further showed the improvement in the values of the F1 score as it improved from 86% to 87% for those who didn’t donate and from 42% to 44% for those who donated. This means that the optimization has a positive effect on the predicted values.

This is also evident in figure 5 as the new confusion matrix shows that the **True negative** has increase to 107, **false positive** has reduced to 7, **false negative** is still 24 and **true positive** is still at 12.

For the neural network, table 5 shows the result of the model and we can see that the accuracy of 67% is lower compared to that of KNN. Likewise, the F1 score for those who did not donate, was lower compared to that of KNN too. However, there was a higher f1 score of 55% for those who donate in comparison to the KNN model.

Finally, comparing the optimized metrics of the 2 models, we can see that the accuracy and precision of KNN was higher, while the recall and f1 scores of NN was higher.

However, looking at the target value, we can see from figure 2 revealed clearly that there is an imbalanced class. Hence, we can only conclude using the f1 score in this case.

**5.0 Conclusion**

Based on the results above, and judging by the f1 scores of both models, the Neural Network model would be recommended as the best classifier for this data set as it gave a higher score compared to the K-Nearest Neighbors.

**TASK 1b: CLASSIFICATION WITH MICROSOFT AZURE MACHINE LEARNING STUDIO**

**Performance Evaluation of 2 Classification Algorithms on a Data Set**

**1.0 Introduction**

Most explanations about classification have been made in the part a. Here, I will be using the Azure machine learning studio to perform this project.

In this report, I used the same transfusion data just as in part A and did a performance evaluation of the neural network and logistic regression model. Choice of logistic regression here was because it was the second model in python that gave a high accuracy after KNN. So, I wanted to see how best the logit model is in comparison to the Neural network.

**2.0 Explanation and Preparation of Datasets.**

Same data set was used as in part A. so the same explanation works for it. However, I changed the column title “whether he/she donated blood in March 2007” to “Target” for easy identification and use.

After launching the studio, I landed on the home page then went ahead to create a new azure machine learning resource. Here, I named the resource group and workspace name and then my deployment was done. Then I went ahead to launch the studio properly. At the compute tab, I created a new compute cluster and named the cluster too. The compute is just like the computer where we perform all the work and stores it for us.

At the Data tab, I click on create data set, named it, and uploaded it from the local files on my system. I also validated my data set and checked the schema to be sure they are all the right data types. Then, I finally created the data after all checks have been done. And then I viewed the descriptive analysis of the data set.

**3.0 Implementation**

The implementation was carried out at the designer tab. Here, a pipeline was created where all the processes were hosted. The pipeline name was changed, and I chose the cluster I wanted to work with, which was the initial cluster created.

I started my model building by dragging the **data set** to the workspace. Also, under the data transformation tab, I dragged the **split data** to the workspace and connected the output of the **data set** to the input of the **split data**. Then on the **split data**, I specified the train set to be 75% of the data set and chose a random split. I also stratified my split using the target column. Since I had no missing observations, I didn’t clean my data set.

The next thing I did was to normalize my data set, so I dragged the **normalize data** to the workspace and connected the result of the training set to the input of the **normalize data**. At the **normalize data**, I used the z scored transformation method, then selected all the columns except the target column to be normalized. I also picked **applied transformation** so that I can apply the normalization on my test set. Hence, I connected test result output from the **split data** to it and also the transformation output of the **normalize data** to it too.

After this, I needed to train the model, so I picked **train model** and dragged it to the workspace. Then I connected the transformed data output of the **normalize data** to the dataset input of the **train model**. I also introduced the **Two-class neural network model** and connected the untrained model output from it to the untrained model input of the **train model**. Then I edited the two class NN model by editing the hidden nodes as 100, number of learning iterations as 15 and random seeds as 0. This didn’t give any f1 score, so I kept on changing the hidden nodes value and I finally used 250 which gave me a good result unlike in python where I used 100. On the **train model**, I specified my label column as the target column.

I also did a model scoring by using the trained model to predict the testing set. So, I dragged the **score model** to the workspace, connected the trained model output from **train model** to the trained model input of the **score model**, and also the transformed data set from **apply transformation** to the dataset input of the **score model**.

Thereafter, I brought in the **evaluate model** and connected the output scored dataset from the **score model** to the input scored dataset of the **evaluate model**. Then I went ahead to submit my job. Below is the schema of the model (figure 1) and the evaluation result (figure 2)

Diagram

Description automatically generated

**Figure 1: Neural network model**

**Graphical user interface, chart, line chart

Description automatically generated**

**Figure 2: Evaluation report for NN**

N.B: For the Precision-recall curve, the closer it is to 90 degrees, the better the model. It shouldn’t be 45 degrees, which indicates a bad model.

For the logistic regression, I followed the same process, however, instead of the **two class Neural Network**, I used the **two-class logistic regression model** and below is my workflow (Figure 3) and the evaluation report (Figure 4).

Diagram

Description automatically generated

**Figure 3: Logistic regression model**

**Graphical user interface, chart

Description automatically generated**

**Figure 4: Evaluation report for Logit Regression**

For the hyperparameter optimization, I deleted the **train model** and replaced it with the **Tune model hyperparameter**, connected the out put untrained model of the **two-class NN** to the input untrained model of the **Tune model hyperparameter**, output transform data from the **Normalize data** to the input training data set of the **Tune model hyperparameter**, and the output trained best model of the **Tune model hyperparameter** to the input trained model of the **Score model**. I then edited the learning rate of the two class NN as 0.001; 0.01; 0.1; 0.2; 0.4 and the number of hidden nodes as 400.

I repeated the same process for the logit regression by changing the **Train model** to **Tune model hyperparameter**. Also, for the **tune model hyperparameter**, I specified the sweeping mode to entire grid, metric performance to f score, and label column as Target. Under the two-class logistic model, I specified the optimization tolerance as 0.00000001, and regularization weight as 0.01; 0.1; 1.0; 10; 50.

Below is the workflow and the evaluation report of the 2 optimized models.

Diagram

Description automatically generated

**Figure 5: Hyperparameter workflow for NN**

Chart, line chart

Description automatically generated

**Figure 6: Hyperparameter evaluation result for NN**

Diagram

Description automatically generated

**Figure 7: Hyperparameter workflow for Logit regression**

**Graphical user interface, chart

Description automatically generated**

**Figure 8: Hyperparameter evaluation report for Logit regression**

**4.0 Result Analysis and Discussion**

While precision-recall curves are useful for datasets with imbalances, ROC curves are appropriate when observations are evenly distributed among each class. Hence, we focus on the precision -recall curves and the f1 scores of the reports as we have an imbalanced class in this data set.

Figure 2 shows the evaluation report for the neural network model. High precision is correlated with a low false positive rate, likewise a high recall is correlated with a low false negative rate. A high area under the curve denotes both high recall and high precision. This means a perfect classifier’s curve would be at 90 degrees. However, in the case of figure 2, the area under the curve is low and almost at baseline. In addition to this, the f1 score is 31% which is a bit low too compared to what I got with the python tool.

Figure 4 shows the precision recall curve for the logistic regression model. And it can be observed that the curve is almost at the diagonal which isn’t too bad. However, the f1 score of 12.2% is quite low too which depict it isn’t a good model for this data set.

Talking about the hyperparameter optimization of the 2 models, it can be observed that the area under the precision recall curve for neural network improved after increasing the hidden layers from 250 to 400. And the f1 score also increased from 31% to 38.3%.

Also, for the logistic regression model, there was no significant improvement in the curve. However, the f1 score increased from 12.2% to 22.6%.

It was generally observed that the 2 models didn’t really do well in the Azure machine learning studio compared to their performance on the same data with the Python package.

**5.0 Conclusions and Recommendations**

Based on the above results, the following conclusion and recommendations can be made.

1. Judging by the optimized result, the Neural network model tends to be more suitable for the classification of this data in the Azure machine learning studio compared to the logistic model.
2. The python resource is a better platform as it gave a better result compared to the azure machine learning studio.
3. Although Azure studio was easier to use as it is a drag and drop platform, However, the python package is preferable as it is flexible and gave a better classification result.
4. In subsequent research, I would recommend we use varying values of hidden layers for the neural network so we can get a better result as the limited time I have didn’t give me the opportunity to do that.
5. I will also recommend for subsequent research that balancing techniques such as SMOTE, upsampling, downsampling should be done on the training set before model fitting. This could probably improve the result.

**TASK 2: ASSOCIATION RULES MINING**

**Identifying Frequently Purchased Items In A UK-Based Online Retail Store**

**1.0 Introduction**

In the world today, a lot of retail outlets are now being strategic about how they attract their customers to buy more products, including buying things they never planned for. For instance, when you go to the popular Jumia online store mainly based in Nigeria, and you intend to buy a pair of shoes. While searching for the shoe, you realize the system is prompting you to also buy a dress because previous customers who bought that same pair of shoes also ordered the dress. And in the long run, you end up buying the dressed. Then Jumia gets more sales. The machine learning principles that such company such as Jumia devised to attract more sales is called the Market Basket Analysis.

Retailers utilize market basket analysis, which is a data mining approach to boost sales by better understanding clients’ buying habits (TechTarget, 2022). To identify product groups and items that are most likely to be bought together, evaluating big data sets, such as purchase histories is necessary.

Association rules are used in market basket analysis to forecast the possibility that two products will be bought together, by keeping track of how frequently various things appear together and looking for relationships that happen far more frequently than is reasonable. In addition to this, one of the algorithms deployed by machine learning to achieve this association rule is called the Apriori Algorithm.

In this project, I intend developing association rules that would help me identify products that are frequently purchased together in a UK-based online retail store, by people resident in Germany. To achieve this, I used an online retail data set sample, filtered for those purchased from Germany alone, grouped the invoice numbers with their various descriptions to see the most frequently bought items, then performed the market basket analysis to know if the items occur together by coincidence or not, using the lift as a yard stick.

**1.1 Aim and Objectives**

The aim of this report is to identify the top 10 items that are bought together in UK-based online retail store. And to achieve this, the following objectives must be met.

1. To perform data wrangling.
2. To group invoice numbers with their respective items.
3. To determine top 10 frequently occurring item.
4. To identify the top 10 items occurring together using the lift as a criterion.

**2.0 Explanation and Preparation of Datasets.**

The dataset used for this association rule was carefully selected to ensure it is applicable to this machine learning problem. It is titled “Online Retail Data set” and was downloaded from <https://archive.ics.uci.edu/ml/datasets/online+retail> . This international data set includes every transaction made between December 1, 2010, and December 9, 2011 by a UK-based company. It has 541,909 observations with 8 attributes and no missing values. The attributes have both real and integer characteristics. The data set has transaction details from 38 countries, and I only analyzed that of Germany alone. I chose Germany as it happens to be the second most occurring country where the online transactions took place, with a total transaction of 9495 (UK happens to be the first, but I didn’t choose it because I was interested in analyzing dataset for other countries except UK, as the online company is already a UK-based).

Below is a table showing the description of the variables in this data.

**Table 1: list of variables in the data set**

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Attribute** | **Data type** | **Definition** |
| 1. | InvoiceNo | Nominal | This is the Invoice number and It is a six-digit integral number provided to each transaction specifically. |
| 2. | StockCode | Nominal | This is the item code, a five-digit integral number that is assigned specifically to each individual product. |
| 3. | Description | Nominal | This is the product name of the items. |
| 4. | Quantity | Numeric | This is the number of each item in a single transaction. |
| 5. | InvoiceDate | Numeric | The date and time that each transaction was created are listed here. |
| 6. | UnitPrice | Numeric | This is the unit price for the product in Sterling. |
| 7. | CustomerID | Nominal | Each customer is given a 5-digit integral number called a customer ID. |
| 8. | Country | Nominal | This is the nation in which each consumer is located |

Having selected only transactions that took place in Germany, I re-indexed the transactions so that they can follow a correct order. Also, from the information that came with the data set, I realized that some transactions which have InvoiceNo starting with “C” were cancelled transactions. Hence, I listed out those cancelled transactions by first converting the InvoiceNo column to strings and then dropping the cancelled transactions. Thereafter, I was able to get only transactions that aren’t cancelled as that is what we are interested in. And at the long run, a total of 9042 observations are the uncancelled transactions that were analyzed in this project.

I also double checked to ensure there were no null values in the data set. And I ensured I stripped off all white spaces from the description column. And finally, I did a descriptive analysis of the dataset.

**3.0 Implementation**

As it was stated earlier in the introduction, market basket analysis (MBA) is used to find buying patterns in any retail environment (Smartbridge, 2022). Simply said, MBA seeks out the product combinations that appear in transactions the most frequently. It establishes connections and patterns among purchases which is modelled in a form of a conditional statement such as; IF you buy (Fish, meat), you will buy (Cooking oil). Here, the fish and meat is known as the **Antecedent** of the rule while the cooking oil is the **Consequent** of the rule.

The Probability that the antecedent will take place, that is, a customer will buy fish and meat is called the **Support** of the rule. The probability that a customer will purchase the cooking oil on the condition of purchasing fish and meat is referred to as the **confidence** of the rule. Let’s say fish and meat are item set X, while cooking oil is item set Y, the confidence tells us the proportion of transactions where the presence of item X results in the presence of item Y.

Hence, we say the **lift** of the rule is the ratio of the support of X and Y of the rule occurring together, divided by the probability that the two would occur together if they were independent. If the lift is greater than 1, this implies that the two items are found together more often than one would expect by chance. MBA look for rules with lifts greater than 1 and with high support and confidence values. All the above instances given is what is called the Association rules mining in statistics and the algorithm used to achieve this in the Python package is the Apriori Algorithm.

Banoula (2022) stated that Apriori algorithm is used to find items and significant correlations that appear the most frequently in a dataset. In this project, I deployed the use of the apriori algorithm to detect the set of items that are frequently bought together in a UK-based online store, while limiting my findings to customers who bought from Germany alone.

I began by importing the required libraries, including the apriori library, then loaded my data set. I also did some data preprocessing steps such as checking the data types present in the data, viewing the head of the data, and doing a value count of the countries in the data. While cleaning the data, I ensured I filtered out other countries, while leaving transactions of Germany alone. In addition, I re-indexed the remaining rows so we can have an ordered data.

From the data source, we were meant to know that some transactions were cancelled. Hence, I filtered out all cancelled transactions as we only want to work on transactions that were successful. I ensured I also checked the data for presence of null values and stripped off the description column of all white spaces. And finally, I did a descriptive statistical analysis of the quantity of items bought.

**Table 2: Descriptive Statistics of the data set**

|  |  |
| --- | --- |
| **Metrics** | **Quantity** |
| count | 9042.000000 |
| mean | 13.189892 |
| std | 17.640855 |
| min | 1.000000 |
| 25% | 6.000000 |
| 50% | 10.000000 |
| 75% | 12.000000 |
| max | 600.000000 |

To be able to achieve our first objective, I grouped the Invoice number and description based on the quantities and then summed the transactions. And I was able to represent this with a bar chart by showing the top 10 most bought items just as seen below.

A picture containing background pattern

Description automatically generated

**Figure 1: Chart showing top 10 most bought items**

It is important to note that the Apriori algorithm works with binary, hence, I transformed the transactions into binary form, then plotted a bar chart showing the frequency of occurrence of the top 10 items in all transactions.

A picture containing bar chart

Description automatically generated

**Figure 2: Chart showing the top 10 most occurring items**

Now to the Market Basket Analysis of this data set. In this analysis, I used a minimum support of 2%. I chose this value because I have a large dataset and realized using less than 2% support level would make my data take too long to load and I have just a limited time to submit my report. Also, I can’t use too high than that because it won’t give me enough rules to mine. Hence, the reason for sticking with 2% minimum support level.

Using this support level on the apriori algorithm, I was able to get about 915 rows of item sets that met with this cut off (Table 3). And with this, I was able to mine about 2438 association rules, while ensuring the minimum lift threshold was 1 as that is what we are interested in. And finally, I sorted the rules based on their confidence and lift, while focusing on the 10 top rules (Figure 4).

**Table 3: Item sets that meets the minimum support cut off**

|  |
| --- |
| **S/N Support itemsets** |
| 0 0.024070 (10 Colour Spaceboy Pen) |
| 1 0.021882 (12 Pencil Small Tube Woodland) |
| 2 0.032823 (3 Hook Hanger Magic Garden) |
| 3 0.041575 (3 Piece Spaceboy Cookie Cutter Set) |
| 4 0.024070 (36 Pencils Tube Red Retrospot) |
| -------------------------------------------------------------------------------------------------- |
| 910 0.021882 (Round Snack Boxes Set Of4 Woodland , Set Of 2... |
| 911 0.035011 (Round Snack Boxes Set Of4 Woodland , Round Sn... |
| 912 0.024070 (Round Snack Boxes Set Of4 Woodland , Round Sn... |
| 913 0.035011 (Woodland Charlotte Bag, Round Snack Boxes Set... |
| 914 0.024070 (Round Snack Boxes Set Of 4 Fruits , Round Sna... |
| 915 Rows × 2 Columns |

**Table 4: Top 10 Association Rules**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S/N** | **Antecedents** | **Consequents** | **Antecedent Support** | **Consequent Support** | **Support** | **Confidence** | **Lift** | **Leverage** | **Conviction** |
| 0 | (Dolly Girl Childrens Cup) | (Dolly Girl Childrens Bowl) | 0.024070 | 0.026258 | 0.024070 | 1.0 | 38.083333 | 0.023438 | inf |
| 1 | (Dolly Girl Childrens Cup, Postage) | (Dolly Girl Childrens Bowl) | 0.021882 | 0.026258 | 0.021882 | 1.0 | 38.083333 | 0.021307 | inf |
| 2 | (Pink Vintage Spot Beaker) | (Blue Vintage Spot Beaker) | 0.024070 | 0.030635 | 0.024070 | 1.0 | 32.642857 | 0.023333 | inf |
| 3 | (Postage, Sweetheart Ceramic Trinket Box) | (Strawberry Ceramic Trinket Box) | 0.024070 | 0.052516 | 0.024070 | 1.0 | 19.041667 | 0.022806 | inf |
| 4 | (Dinosaur Party Bag + Sticker Set) | (Woodland Party Bag + Sticker Set) | 0.021882 | 0.067834 | 0.021882 | 1.0 | 14.741935 | 0.020398 | inf |
| 5 | (Jumbo Bag Woodland Animals, Jumbo Bag Pink Po... | (Jumbo Bag Red Retrospot) | 0.026258 | 0.078775 | 0.026258 | 1.0 | 12.694444 | 0.024190 | inf |
| 6 | (Jumbo Bag Pink Polkadot, Postage) | (Jumbo Bag Red Retrospot) | 0.028446 | 0.078775 | 0.028446 | 1.0 | 12.694444 | 0.026206 | inf |
| 7 | (Jumbo Bag Woodland Animals, Jumbo Bag Pink Po... | (Jumbo Bag Red Retrospot) | 0.024070 | 0.078775 | 0.024070 | 1.0 | 12.694444 | 0.022174 | inf |
| 8 | (Woodland Charlotte Bag, Jumbo Bag Red Retrospot) | (Jumbo Bag Woodland Animals) | 0.024070 | 0.100656 | 0.024070 | 1.0 | 9.934783 | 0.021647 | inf |
| 9 | (Woodland Charlotte Bag, Postage, Jumbo Bag Re... | (Jumbo Bag Woodland Animals) | 0.021882 | 0.100656 | 0.021882 | 1.0 | 9.934783 | 0.019679 | inf |

**4.0 Result Analysis and Discussion**

Based on the descriptive analysis result (table 2), we can see that after the data wrangling, 9042 observations made it to the analysis. The average (mean) of all the quantities of the transaction was 13.189892, and the measure of how dispersed the data is in relation to the mean (standard deviation) is 17.640855. Also, we can see that at least, 1 quantity of all the listed items was purchased and the highest number of quantities any of the item could have in the whole transaction was 600. In addition, we can see that 25% of the purchased items had quantities that was below 6, half of the purchased items had quantities that was below 10 and 75% had quantities that was below 12.

Looking at figure 2, we can see that Postage occurred the most in the whole transactions with its frequency hitting above 350. This means out of about 9042 transactions, postage occurred more than 350 times (above 3.87%). This explanation also applies to all other items on the chart.

As it has been stated earlier that a lift of 1 is our interest, because if the lift is greater than 1, it means the items in the association rule would be found more often together. Hence, in table 4, we can see that the topmost association rule with Dolly Girl Children’s Cup as the Antecedent and Dolly Girl Children’s Bowl as the consequence, has a lift of 38.083333 with a support of 1. It means that the probability that a customer from Germany who buys Dolly Girl Children’s Cup would buy Dolly Girl Children’s bowl is very high and tops the list of all the transactions that took place in Germany. This is also the case of Dolly Girl Children’s Cup, Postage and Dolly Girl Children’s Cup as they have the same confidence and lift. It is also worth noting that all the top 10 association rules have a confidence of 1 and the frequency that the rule makes an incorrect prediction (Conviction)

**5.0 Conclusions**

Based on the above analysis, the following conclusions can be made.

1. Since postage had the highest frequency in all transactions, more postage should be made available for people in Germany to purchase.
2. The UK-based company should ensure they place Dolly Girl Children’s Cup, Postage and Dolly girl Children’s bowl side by side as they happened to be purchased more together by people residing in Germany.
3. In addition to this, Items that are similar should be placed side by side for easy purchase as the trend shows customers bought similar items more.
4. I am sure this algorithm would help the UK-based company make more sales in the nearest future if we can analyze transactions for other countries too.

**TASK 3: CLUSTERING**

**Performance Evaluation Of 2 Clustering Methods On A Selected Data Set**

**1.0 Introduction.**

In life, a lot of things looks similar and there is a need to group them together so that they can be well organized. In machine learning, the act of grouping similar data together is called clustering. It is an unsupervised type of machine learning, and its task is to break data sets into a certain number of clusters so that the data points within each cluster share common traits (Sharma, 2022).

In this project, my aim is to evaluate the performance of 2 clustering algorithms on a transfusion data set. This data set has an independent variable that states whether a person will donate blood in 2007 or not. The data set titled “Transfusion.data” was used and was downloaded from UCI machine learning repository site (<https://archive.ics.uci.edu/ml/machine-learning-databases/blood-transfusion>). Using other attributes, I used K-means and Agglomerative clustering algorithms to cluster the data set. I was also able to get the optimal number of clusters. And finally, I compared the two cluster algorithms using metrics such as the Silhouette score, completeness, and homogeneity. Based on physical appearance (graph) and the metrics, K-means happens to be a better cluster algorithm for this data set.

**2.0 Explanation and Preparation of Datasets.**

Explanation about the data set has been done in task 1. However, in this part of the project, we are only interested in the 4 dependent variables as we are using them to cluster the data. As such, the variables used in this project are Recency (months), Frequency (time), Monetary (c.c. blood) and Time (months).

After importing the required libraries and the data set, I performed some data preprocessing. I viewed the head of the data, checked some information about the data, then dropped the independent variable, which is the target variable in this data set. I also made an histoplot of the four dependent variables to have a proper view of them (Figure 1).

In addition to this, I standardized the data using the Standard Scaler library. Standardizing a data ensures all data we are working with are on a uniform scale by eliminating the mean and scaling to unit variance. I also performed dimensionality reduction using both t-distributed stochastic neighbor embedding (t-SNE) and Principal component Analysis (PCA) methods. However, I ended up using the PCA as it is the most commonly used method and also gave me a better-looking cluster than the t-SNE. Dimensionality reduction helps to facilitates data compression, enabling the data to occupy less storage space and require less processing power.

Chart, histogram

Description automatically generated

**Figure 1: Histoplot of the dependent variables**

**3.0 Implementation**

As it was stated earlier that clustering helps to group similar data together, and there are different types of clustering algorithm that could be used. Madhulatha (2012) grouped clustering algorithm into 2 types namely, Hierarchical and Partitional. While partitional algorithms decide all clusters at once, hierarchical techniques find subsequent clusters utilizing previously established clusters. In this report, Agglomerative algorithm (Hierarchical) and K-means (partitional) was used. When there are a definite number of classes, K-Means is employed. On the other hand, when there are an unknown number of classes, agglomerative is used (Das, 2020).

According to Scikit-learn (2022), there are some metrics that helps measure the performance of a clustering algorithm. Some of them includes Rand index, Mutual Information, Homogeneity, completeness, V-measure, and Silhouette coefficient.

Zuccarelli (2021) reported that each point in a cluster is compared against points in nearby clusters using the **silhouette score** and the score ranges between -1 and 1. In addition to this, similar clustering would have a high **adjusted rand index** and if every cluster in the clustering result only contains data points that belong to one class, then it is **homogeneous**. **Mutual information** on the other hand measures how similar two labels for the same data are to one another. And finally, a totally **complete** clustering is one in which every data point from the same class is grouped together into a single cluster. All these measures were put into consideration in this project when concluding the best type of clustering algorithm for the data used.

After all data preprocessing has been done, including dimensionality reduction, I plotted the result of the two extracted components PCA1 and PCA2 from the Principal component Analysis (PCA). I also did the same with the t-SNE, and it was at this point I picked the PCA as my choice for the dimensionality reduction as the plot looks more appealing than the t-SNE.

**Chart, scatter chart

Description automatically generated**

**Figure 2: PCA dimensionality reduction plot**

**Chart, scatter chart

Description automatically generated**

**Figure 3: t-SNE dimensionality reduction plot.**

Before I could use the K-means algorithm, I needed to specify the optimal number of clusters. Banerji (2021) mentioned 2 methods of getting the optimal K which are the elbow curve method and the Silhouette analysis. In this project, I used the elbow curve method, Silhouette score and the Calinski Harabasz score method, and plotted a graph of each of them to physically view them.

Chart, line chart

Description automatically generated

**Figure 4: Plot of Elbow curve**

**Chart, line chart

Description automatically generated**

**Figure 5: Plot of Silhouette score**

**Chart, line chart

Description automatically generated**

**Figure 6: Plot of Calinski Harabasz score.**

Since we also know there are 2 groups in the data set, then Silhouette & Calinski score also proved to us that we can only get 2 optimal clusters in the group, Hence, I settled for the 2 clusters and then went ahead to cluster my data.

While specifying the number of clusters as 2 and random state as 0, I clustered my data set using K-means algorithm and I fit predicted the clusters with the previously scaled dependent variables. Using the PCA dimensionality reducer, I plotted the new predicted clusters of the data set and compared it with the true value (figure 7). I repeated the same step using the Agglomerative algorithm, while specifying the number of clusters and the affinity as Euclidean. Then I plotted both the K-mean, agglomerative and the true value side by side so that I can physically judge which algorithm best looks like the true value.

**Chart, scatter chart

Description automatically generated**

**Figure 7: K-means and true value plot**

**Chart, scatter chart

Description automatically generated**

**Figure 8: K-means, agglomerative and True value plots.**

After this, I compiled all metrics into a single pipeline and evaluated the performance of each of the cluster algorithm using Homogeneity, completeness, v-measure, Adj. Rand index, Adj. mutual information and Silhouette score. And finally, I plotted a chart displaying the metrics for each of the algorithms.

**Table 1: performance evaluation of clustering algorithms**

|  |  |  |
| --- | --- | --- |
| **Metrics** | **KMeans** | **Hierarchical** |
| **Homogeneity** | 0.008563 | 0.006435 |
| **Completeness** | 0.008637 | 0.005550 |
| **V-measure** | 0.008600 | 0.005960 |
| **Adj. Rand Index** | 0.048001 | 0.028561 |
| **Adj. Mutual Information** | 0.007380 | 0.004833 |
| **Silhouette Score** | 0.430032 | 0.386147 |

**Chart

Description automatically generated**

**Figure 9: Chart of metrics for clustering algorithm**

**4.0 Result Analysis and Discussion**

As stated earlier, I chose the PCA dimensionality reduction over the t-SNE because the PCA plot looks more appealing compared to the later. Looking at figures 4, 5 and 6, we have the number of clusters as 5, 2 and 2 respectively. But in the case of the data set, the target already pointed towards 2 sets of groups, which is, those who will donate blood and those who won’t. This is the reason I chose the Silhouette score which gave an optimal number of clusters as 2.

From figure 8, the plots of the K-means, agglomerative and true value reveals clearly how the data was clustered. Visually looking at the 2 predicted plots in comparison to the true value, the K-means clustering looks more similar to the true value plot unlike the agglomerative clustering.

Judging by the metrics, we can see from table 1 and figure 9 that in all measures, K-means has metric values that is higher than the agglomerative algorithm. Silhouette score takes values between -1 and 1 and the closer the values to 1, the better the cluster. We can see from the visuals that K-means has a value of 0.430032 which is closer to 1 compared to 0.386147 for agglomerative. Also, both homogeneity and completeness have values ranging between 0 and 1, and the closer the result to 1, the better it is. And from the result, it is obvious that K-means has values of 0.008563 and 0.008637 for homogeneity and completeness respectively, which is high, compared to the values for agglomerative. This also applies to other measures.

**5.0 Conclusions**

At the end of this comparison, while judging by the physical appearance of the plots and the metrics, I can say that K-means algorithm gave a better cluster for the transfusion data compared to the agglomerative (Hierarchical) algorithm because its values were higher than the later.

**TASK 4: TEXT MINING AND SENTIMENT ANALYSIS**

**Understanding the Emotions of Tourists based on analysis of their Reviews**

**1.0 Introduction**

A lot of websites, applications and so many technological frameworks have a review section where the publishers want to get feedbacks about them. In this tech generation, hotels, restaurant, and other public places are not left out of this race. After getting the reviews of customers, the company needs to analyze, and interpret them to be able to bring out useful insight and promote their companies better, while also satisfying the customers. This is where text mining and sentimental analysis comes in.

Text mining is the process of converting unstructured text into a structured format, with the purpose of identifying significant patterns and fresh insights (IBM Cloud Education, 2020). On the other hand, Sentimental Analysis helps to find positive or negative emotions in words.

In this project, my aim is to do a text mining and sentimental analysis of reviews from 30 hotels/restaurant in Patong, Phuket, so as to know how they feel about the place. To achieve this, I ensured I cleaned the reviews by removing stop words and lemmatizing it, then used a word cloud to view the top 20 most occurring words from the review. I also checked the polarity of the review and its subjectivity. And I can say that there were more positive reviews as the sentimental analysis gave me about 85.6% positive polarity. This means that tourist had more positive emotions/feelings about the restaurants/hotels around Patong.

**2.0 Explanation and Preparation of Datasets**

In this report, the data set used (tourist\_accommodation\_reviews) was provided by the course instructor. The data set has 53644 observations and 5 attributes namely (ID, Review date, Location, Hotel/Restaurant name and Review). I reviewed 30 hotels/restaurants in the data set while focusing on hotels/restaurants around Patong Beach. Patong beach is found in Phuket, a city in Thailand. My choice of Patong is not far-fetched as it happens to be the longest and most populous beach in Phuket, as well as one of the most visited (McMurray, 2022). Hence, I wish to know what is attracting people to the place through text mining and how positive the review is via sentiment analysis.

Some data preprocessing and cleaning was done on the data set to make it available for use. First, I listed all locations available in the dataset, then selected data in Patong alone as my choice of location. Also, I viewed the hotels in Patong, and I can confirm that there are 164 distinct hotels there, based on the information from the data set. From these 164 hotels, I randomly picked 30 hotels to work on so as to avoid any form of bias in the analysis using the numpy library. Then, I dropped other attributes, leaving only the Hotel names and the reviews as those are our interest. I also re-indexed the new selections to ensure orderliness. Then, I made a plot of the top 20 hotels based on their frequency of occurrence in the review.

Chart, bar chart

Description automatically generated

**Figure 1: Top 20 most occurring hotels**

Talking about the data cleaning process, I defined the lemmatizer using an English pipeline. Then using a pipeline, I used a regular expression to extract only alphabets in the review, then joined the extracted letters. I also converted the words to lower case, removed stop words, lemmatized the key words and then joined the lemmatized words. **Stop words** are filler words with little or no meaning in a sentence. Also, words are **lemmatized** to return them to their root words. All these steps are important to help us reduce the bulk of words to be processed and ease analysis.

**3.0 Implementation**

After performing all the data preprocessing and cleaning, I checked the word frequency in a sparse matrix format using the count vectorizer. The frequency of each word that appears in the reviews is used by the count vectorizer to convert a given text into a vector. Each cell's value is the number of words in that specific text sample. Thereafter, I checked the top 20 most used words and plotted it on a chart.

Chart, bar chart, histogram

Description automatically generated

**Figure 2: Top 20 Most Used Words**

In other to be able to represent the most frequent words used in the review on the word cloud, I joined the text in the reviews then created a function to build the word cloud. And finally, I embedded it in a “comment” image.

Text

Description automatically generated

**Figure 3: Word Cloud Of Most Frequent Words**

Also, for the Sentimental Analysis, text blob was installed. It’s a library that helps process textual data. Then, I defined Polarity and Subjectivity of the analysis using the lambda function. Polarity expresses how positive, negative, or neutral the emotion of a text is while Subjectivity demonstrates how arbitrary the text is. After this, I viewed the polarity and subjectivity of each review, classed the polarity as positive, negative, or neutral and finally, counted the frequency of occurrence of each emotion after normalizing it.

**Table 1: Polarity classification of the review**

|  |
| --- |
| **positive 0.856347** |
| **Neutral 0.078638** |
| **negative 0.065015** |
| **Name: Classification, dtype: float64** |

**4.0 Result Analysis and Discussion**

Figure1 reveals the 20 topmost frequently occurring hotel and De Mario happens to be the hotel people visited most with over 250 reviews. Also, No. 6 Restaurant is the second most visited place with about 200 reviews while Home Kitchen, Bar & Bed is the 3rd most visited place with about 100 reviews.

Talking about the most used words, we can see from figure 2 that “food” topped the list with 2334 mentions, next to “good” with 1793 mentions and “restaurant” with 1174 mentions. If “good” could be the second on the list, then I want to believe the reviews about most of the hotels/restaurants are great and their food were fantastic

The sentimental analysis also gave a positive result about the review. After classifying the review as positive if polarity is greater than or equal to 0.05, negative if less than or equal to -0.05 and neutral if equal to zero, result shows that 85.63% of the reviews were positive, 7.86 % were negative while 6.5% were neutral. This is a great feet as majority of the tourist were satisfied with the services they got from the hotels/restaurants around Patong. This is also a good deal for the business owners as this review means more customers would want to visit there and patronize them.

**5.0 Conclusion**

Based on the above results, we can conclude that the hotels/restaurants in Patong are outstanding as the reviews were excellent. “Food”, “good”, and “restaurant” showing as the highest used words means their meals were top notch and they have an outstanding service in their restaurants. This positivity would increase the revenue of businesses in Patong as more people would also like to have the good experience. No wonder Patong was said to have been the most populous place in Phuket.

**References**

# Banerji, A. (2021). K-mean: Getting the optimal number of clusters. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/05/k-mean-getting-the-optimal-number-of-clusters/>

# Banoula, M. (2022). What Is Apriori Algorithm in Data Mining: Everything You Need to Know. SimpliLearn. <https://www.simplilearn.com/tutorials/machine-learning-tutorial/apriori-algorithm-in-data-mining>.

# Das, V. K. (2020). K-means clustering Vs Hierarchical clustering. Global Tech Council. <https://www.globaltechcouncil.org/clustering/k-means-clustering-vs-hierarchical-clustering/>

# David, D. (2020). Hyperparameter Optimization techniques to improve your machine learning model’s performance. freecodecamp.org/news/hyperparameter-optimization-techniques-machine-learning/.

# Goyal, K. (2022). Machine learning vs Neural Networks: what is the difference UpGrad. <https://www.upgrad.com/blog/machine-learning-vs-neural-networks/#:~:text=A%20Neural%20Network%20arranges%20algorithms,interaction%20in%20the%20early%20stages>.

# Harrison, O. (2018). Machine Learning Basics with K-Nearest Neighbors Algorithm. Towards Data Science. <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>.

# IBM Cloud Education (2020). Text Mining. <https://www.ibm.com/cloud/learn/text-mining#:~:text=Text%20mining%2C%20also%20known%20as,meaningful%20patterns%20and%20new%20insights>.

# 

# Java T point (2022). Classification Algorithm in Machine Learning. <https://www.javatpoint.com/classification-algorithm-in-machine-learning>.

# Knocklein, O. (2019). Classification Using Neural Networks. Towards Data Science. <https://towardsdatascience.com/classification-using-neural-networks-b8e98f3a904f>.

# Madhulatha, T.S. (2012). An Overview On Clustering Methods. IOSR Journal of Engineering. Apr. 2012, Vol. 2(4) pp: 719-725. <https://arxiv.org/ftp/arxiv/papers/1205/1205.1117.pdf>.

# McMurray, S. (2022). 15 Top-Rated Beaches Phuket. Planet ware. <https://www.planetware.com/thailand/top-rated-beaches-in-phuket-tha-1-47.htm>.

# Scikit-learn: Machine Learning in Python (2011). Clustering. <https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation>.

# Sharma, R. (2022). What is Clustering and different types of Clustering Methods. UpGrad. <https://www.upgrad.com/blog/clustering-and-types-of-clustering-methods/>

Smartbridge. (2022). Market Basket Analysis 101: Anticipating customer behavior. <https://smartbridge.com/market-basket-analysis-101/>.

TechTarget. (2022). Market Basket Analysis. <https://www.techtarget.com/searchcustomerexperience/definition/market-basket-analysis>.

Zuccarelli, E. (2021) Performance metrics in machine learning. Toward Data Science. <https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation>