TDS 3301
DATA MINING

INSURANCE PRODUCT RECOMMENDATION

Prepared by

Ann Choi Hua En, 1181300227, 0183165400
Oi Zhen Fan, 1181300513, 0193521017
Lee Xi Jie, 1161204459, 0162858045

1 Introduction

Through discussion between the members, we decide to construct a recommendation system for insurance to allow agents to optimize up-selling performances, by selecting customers who are most likely to subscribe an additional plan. We hope to perform a more efficient up-selling than classic marketing campaigns. To achieve this, our recommendation system consists of several classification, clustering and association prediction techniques for customized insurance plan. Therefore, our development of insurance plan recommendation systems have the capability to offer recommendations according to the diverse user coverage needs and financial constraints.

2 Exploratory Data Analysis

The given dataset consists of 26 attributes and 501 instances with the column names, in order to study and provide insights of the dataset for further data processing, different charts and graphs are plotted to study and understand the data. Besides in the beginning part, we firstly understand what all the features meaning in the dataset and understand those data. As the aim of this project is to recommend the plan to their respective targeted audience, thus, we knew that the attribute CUSTOMERNEEDS and PURCHASEDPLAN will be our point of interest. The overall work for Exploratory Data Analysis (EDA) is shown in the section Findings.

2.1 Data Preprocessing

After finishing studying the data, we make changes to the columns name to uppercase and this is particularly helpful for us to determine the features names in the coming steps and definitely improves the efficiency for us while we are coding. In the following section, we will discuss the more about pre-processing process.

```
Index(['AGE', 'GENDER', 'MARITALSTATUS', 'SMOKERSTATUS', 'LIFESTYLE',
       'LANGUAGESPOKEN', 'HIGHESTEDUCATION', 'RACE', 'NATIONALITY',
       'MALAYSIAPR', 'MOVINGTONEWCOMPANY', 'OCCUPATION', 'TELCO',
       'HOMEADDRESS', 'RESIDENTIALTYPE', 'NOOFDEPENDENT',
       'FAMILYEXPENSES(MONTH)', 'ANNUALSALARY', 'CUSTOMER_NEEDS_1',
       'CUSTOMER_NEEDS_2', 'PURCHASEDPLAN1', 'TRANSPORT', 'PURCHASEDPLAN2',
       'MEDICALCOMPLICATION'],
```

Figure 1: New Columns Name

2.1.1 Missing Value Treatment

To handle the missing value, firstly we categorize all features into numerical and categorical features. For AGE and NOOFDEPENDENT, we fill in the missing values by their mean values.

Subsequently, we handle the missing values of, ANNUALSALARY, SALARY(MONTH), and FAMILYEXPENSES(MONTH), by filling with their mean values which according to their RESIDENTIALTYPE. In such, we would reduce outliers as a person's affordability is predicted within a range and would be more accurate.

To deal with missing values of categorical features, we fill in the missing values with a string value "NotSpecified". This is because we can't merely fill in missing values as this would falsify and corrupt the data if we simply fill in the values. Besides, by simply filling the null values, it may impact the accuracy and reduce the veracity of the data sets.

[Before]

AGE	92
GENDER	0
MARITALSTATUS	142
SMOKERSTATUS	66
LIFESTYLE	0
LANGUAGESPOKEN	0
HIGHESTEDUCATION	0
RACE	115
NATIONALITY	145
MALAYSIAPR	0
MOVINGTONEWCOMPANY	0
OCCUPATION	145
TELCO	0
HOMEADDRESS	79
RESIDENTIALTYPE	0
NOOFDEPENDENT	94
FAMILYEXPENSES(MONTH)	121
ANNUALSALARY	156
CUSTOMER_NEEDS_1	0
CUSTOMER_NEEDS_2	0
PURCHASEDPLAN1	0
TRANSPORT	0
PURCHASEDPLAN2	0
MEDICALCOMPLICATION	0
SALARY(MONTH)	156

[After]

AGE	0
GENDER	0
MARITALSTATUS	0
SMOKERSTATUS	0
LIFESTYLE	0
LANGUAGESPOKEN	0
HIGHESTEDUCATION	0
RACE	0
NATIONALITY	0
MALAYSIAPR	0
MOVINGTONEWCOMPANY	0
OCCUPATION	0
TELCO	0
HOMEADDRESS	0
RESIDENTIALTYPE	0
NOOFDEPENDENT	0
FAMILYEXPENSES(MONTH)	0
ANNUALSALARY	0
CUSTOMER_NEEDS_1	0
CUSTOMER_NEEDS_2	0
PURCHASEDPLAN1	0
TRANSPORT	0
PURCHASEDPLAN2	0
MEDICALCOMPLICATION	0
SALARY(MONTH)	0

Figure 2: Missing Value Treatment

2.1.2 Feature Engineering

In this part, we performed feature engineering which is creating a new column called SALARY(MONTH). The values of SALARY(MONTH) is calculated by dividing ANNUALSALARY by 12 in order to divide the year into 12 months. By adding this new column, it will be easier for us to compare it with another column, which is FAMILYEXPENSES(MONTH).
Figure 3 shows that the new feature SALARY(MONTH) has been added into the dataframe.

	FAMILYEXPENSES(MONTH)	ANNUALSALARY	SALARY(MONTH)
0	10242.00000	119040.678571	9920.056548
1	6334.57265	73926.000000	6160.500000
2	4316.00000	140734.000000	11727.833333
3	4845.00000	119040.678571	9920.056548
4	9883.00000	98833.000000	8236.083333
5	6174.00000	119040.678571	9920.056548
6	7735.00000	119040.678571	9920.056548
7	2679.00000	166567.000000	13880.583333
8	4930.00000	51144.000000	4262.000000
9	5301.00000	127075.000000	10589.583333

Figure 3: Dataframe after feature extraction

2.1.3 Label Encoding

We convert all categorical values into numerical forms by using LabelEncoder in order to let the values fit into machine learning algorithms and to do a better job in predictions afterwards. Additionally, we could use label encoded datasets to perform correlation checks.

	0	1	2	3	4
AGE	35	25	27	33	28
GENDER	0	1	1	0	0
MARITALSTATUS	2	0	0	0	0
SMOKERSTATUS	0	0	1	2	2
LIFESTYLE	0	1	2	2	0
LANGUAGESPOKEN	0	1	0	0	0
HIGHESTEDUCATION	0	1	0	0	0
RACE	3	1	4	3	1
NATIONALITY	1	0	1	0	0
MALAYSIAPR	1	0	0	1	1
MOVINGTONEWCOMPANY	1	1	0	0	1
OCCUPATION	1	4	3	3	3
TELCO	2	3	0	2	3
HOMEADDRESS	3	2	1	4	2
RESIDENTIALTYPE	3	3	1	3	1
NOOFDEPENDENT	2	2	2	2	2
FAMILYEXPENSES(MONTH)	10242	6334.57	4316	4845	9883
ANNUALSALARY	119041	73926	140734	119041	98833
CUSTOMER_NEEDS_1	2	1	0	2	1
CUSTOMER_NEEDS_2	1	1	1	2	0
PURCHASEDPLAN1	2	2	2	2	0
TRANSPORT	0	0	0	0	0
PURCHASEDPLAN2	1	2	0	1	0
MEDICALCOMPLICATION	0	1	1	0	1
AGE_GROUP	(30, 40]	(20, 30]	(20, 30]	(30, 40]	(20, 30]
SALARY(MONTH)	9920.06	6160.5	11727.8	9920.06	8236.08

Figure 4: Label Encoded Dataframe

2.1.4 Data Transformation

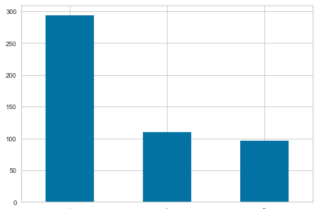
Furthermore, We perform data normalization for three particular numerical features by using min max scaler as shown in the figure 5. Normalisation is particularly helpful in classification problems as it helps give an equal weight to all our features and not let any one feature dominate the other. Also, normalization allows us to eliminate the units of measurement of the data and let us have a better understanding of the data set in order to use those three features compared with other features. In addition, normalization also protects our data. Besides, in the mix max scaler will transform the datas into min with 0, max with 1 and the rest will be in the range of 0,1.

	FAMILYEXPENSES(MONTH)	ANNUALSALARY	SALARY(MONTH)
0	0.972320	0.467180	0.467180
1	0.480696	0.169964	0.169964
2	0.226724	0.610095	0.610095
3	0.293281	0.467180	0.467180
4	0.927151	0.334051	0.334051
...	...	...	...
495	0.291520	0.467180	0.467180
496	0.621666	0.113689	0.113689
497	0.102542	0.025265	0.025265
498	0.853800	0.467180	0.467180
499	0.627328	0.442750	0.442750

Figure 5: Three features after performing min max scaler

2.1.5 Handling Imbalanced Data

Subsequently, We decide to balance the dataset by using PURCHASEDPLAN1 and PURCHASEDPLAN2 as our target features and SMOTE with oversampling technique. The reason we choose oversampling from undersampling is because oversampling would not face the problem of loss of data as these data set instances weren't big. As we can see from Figure 6, the frequency of each value in PURCHASEDPLAN1 is very imbalanced before SMOTE. For PURCHASEDPLAN2 as we can see from Figure 8, although the data before SMOTE seems more balanced than PURCHASEDPLAN1, we still decide to balance it with preparation of increasing accuracy for further predicting modelling techniques.

[Before] 

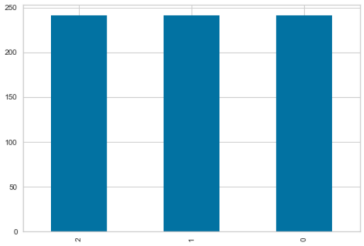
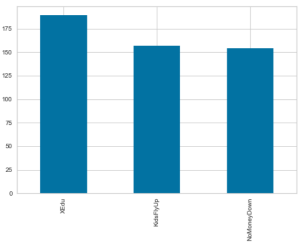
[After] 

Figure 6: Bar Chart for the data before and after SMOTE with PURCHASEDPLAN1 as the target feature

[Before] 

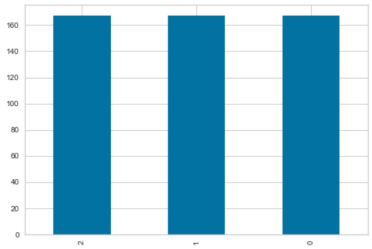
[After] 

Figure 7: Bar Chart for the data before and after SMOTE with PURCHASEDPLAN2 as the target feature

3 Feature Selection

The original dataset has been pre-processed which are filling NaN values, data transformation, provided imbalance treatment and encode the dataset into the form by Label Encoding. Besides than encoding, we perform the normalization to the numeric data using min-max normalization.

	0	1	2	3	4	5
AGE	35	25	27	33	28	44
GENDER	female	male	male	female	female	male
MARITALSTATUS	single	NotSpecified	NotSpecified	NotSpecified	NotSpecified	single
SMOKERSTATUS	NotSpecified	NotSpecified	frequent	once_in_a_while	once_in_a_while	once_in_a_while
LIFESTYLE	home	outdoor	pub_goer	pub_goer	home	home
LANGUAGESPOKEN	english	malay	english	english	english	english
HIGHESTEDUCATION	Bachelor	Diploma	Bachelor	Bachelor	Bachelor	Bachelor
RACE	malay	chinese	others	malay	chinese	NotSpecified
NATIONALITY	NotSpecified	Malaysian	NotSpecified	Malaysian	Malaysian	Malaysian
MALAYSIAPR	yes	no	no	yes	yes	yes
MOVINGTONEWCOMPANY	yes	yes	no	no	yes	no
OCCUPATION	employer	selfEmployed	privateEemployee	privateEemployee	privateEemployee	privateEemployee
TELCO	maxis	umobile	celcom	maxis	umobile	umobile
HOMEADDRESS	north_mal	east_mal	central_mal	south_mal	east_mal	central_mal
RESIDENTIALTYPE	terrace	terrace	condominium	terrace	condominium	terrace
NOOFDEPENDENT	2	2	2	2	2	3
FAMILYEXPENSES(MONTH)	10242	6334.57	4316	4845	9883	6174
ANNUALSALARY	119041	73926	140734	119041	98833	119041
CUSTOMER_NEEDS_1	PersonalSaving	PersonalRetirement	PersonalMedical	PersonalSaving	PersonalRetirement	PersonalRetirement
CUSTOMER_NEEDS_2	KidMedical	KidMedical	KidMedical	KidSaving	KidEducation	KidMedical
PURCHASEDPLAN1	SchoolAgain	SchoolAgain	SchoolAgain	SchoolAgain	COVIDFree	HomeSafe
TRANSPORT	driving	driving	driving	driving	driving	driving
PURCHASEDPLAN2	NoMoneyDown	XEdu	KidsFlyUp	NoMoneyDown	KidsFlyUp	KidsFlyUp
MEDICALCOMPLICATION	no	yes	yes	no	yes	yes
AGE_GROUP	(30.0, 40.0]	(20.0, 30.0]	(20.0, 30.0]	(30.0, 40.0]	(20.0, 30.0]	(40.0, 50.0]
SALARY(MONTH)	9920.06	6160.5	11727.8	9920.06	8236.08	9920.06

Figure 8: Dataframe before pre-process

	0	1	2	3	4	5
AGE	21.000000	31.000000	31.000000	36.000000	29.000000	33.00000
GENDER	0.000000	1.000000	0.000000	0.000000	1.000000	0.00000
MARITALSTATUS	1.000000	0.000000	2.000000	1.000000	2.000000	1.00000
SMOKERSTATUS	2.000000	3.000000	1.000000	2.000000	0.000000	2.00000
LIFESTYLE	2.000000	0.000000	0.000000	2.000000	2.000000	2.00000
LANGUAGESPOKEN	0.000000	1.000000	0.000000	1.000000	1.000000	0.00000
HIGHESTEDUCATION	0.000000	0.000000	2.000000	1.000000	0.000000	2.00000
RACE	3.000000	3.000000	3.000000	4.000000	3.000000	4.00000
NATIONALITY	1.000000	1.000000	1.000000	2.000000	2.000000	0.00000
MALAYSIAPR	1.000000	1.000000	1.000000	1.000000	1.000000	1.00000
MOVINGTONEWCOMPANY	1.000000	0.000000	0.000000	0.000000	0.000000	1.00000
OCCUPATION	0.000000	3.000000	2.000000	0.000000	3.000000	0.00000
TELCO	2.000000	2.000000	2.000000	2.000000	1.000000	0.00000
HOMEADDRESS	0.000000	4.000000	0.000000	0.000000	1.000000	1.00000
RESIDENTIALTYPE	3.000000	1.000000	1.000000	3.000000	0.000000	1.00000
NOOFDEPENDENT	2.000000	2.000000	2.000000	3.000000	2.000000	3.00000
FAMILYEXPENSES(MONTH)	0.565551	0.440063	0.440063	0.047685	0.251007	0.80385
ANNUALSALARY	0.467180	0.882430	0.358058	0.936630	0.459023	0.44275
CUSTOMER_NEEDS_1	1.000000	1.000000	1.000000	1.000000	0.000000	0.00000
CUSTOMER_NEEDS_2	2.000000	0.000000	0.000000	0.000000	2.000000	0.00000
TRANSPORT	0.000000	0.000000	0.000000	0.000000	1.000000	0.00000
PURCHASEDPLAN2	0.000000	0.000000	0.000000	2.000000	2.000000	0.00000
MEDICALCOMPLICATION	0.000000	1.000000	0.000000	0.000000	1.000000	1.00000

Figure 9: Dataframe after pre-process

After preprocessing, we use the processed datasets to for the feature selection. First and foremost, train test split is applied to our target data, y and y2 which are PURCHASEDPLAN1 and PURCHASEDPLAN2. Besides, a function is created to identify the ranking from the features as shown below.

```
def ranking(ranks, names, order=1):
    minmax = MinMaxScaler()
    ranks = minmax.fit_transform(order*np.array([ranks]).T).T[0]
    ranks = map(lambda x: round(x,2), ranks)
    return dict(zip(names, ranks))
```

3.1 Feature Importance Study

In the features selecting, we perform the Boruta and RFE to identify which features are the most important for us to do the classification and the machine learning techniques. Besides, RFE is easy to be configured and used. It will select those features in the training dataset that are most relevant in predicting the target variable. Furthermore, boruta is a wrapper algorithm which build around by the random forest algorithm. In the boruta, it tries to capture all the significant features to the target data. In the beginning, we fit the X and y, PURCHASEDPLAN1 into the feat-selector and RFECV to obtain the score of the features. The figure 10 below shows that the features score after we fit the X and y. From the figures, we can obtain that what are the least significant and important features for the target data.

[Boruta]

	Features	Score
0	AGE	1.0
3	SMOKERSTATUS	1.0
21	PURCHASEDPLAN2	1.0
17	ANNUALSALARY	1.0
16	FAMILYEXPENSES(MONTH)	1.0

---------Bottom 5----------

	Features	Score
8	NATIONALITY	0.20
6	HIGHESTEDUCATION	0.13
1	GENDER	0.07
10	MOVINGTONEWCOMPANY	0.00
20	TRANSPORT	0.00

[RFE]

---------Top 5----------

	Features	Score
0	AGE	1.0
1	GENDER	1.0
21	PURCHASEDPLAN2	1.0
19	CUSTOMER_NEEDS_2	1.0
18	CUSTOMER_NEEDS_1	1.0

---------Bottom 5----------

	Features	Score
3	SMOKERSTATUS	1.0
2	MARITALSTATUS	1.0
22	MEDICALCOMPLICATION	1.0
10	MOVINGTONEWCOMPANY	0.5
20	TRANSPORT	0.0

Figure 10: Features Score for PURCHASEDPLAN1

After that, we perform the same process for the PURCHASEDPLAN2 as our target data 2 to get the important features scores. So we fit the X2 and y2(PURCHASEDPLAN2) into the feat-selector and RFECV to obtain the score of the features. The figures 11 below shows the least significant and important features for the target data. Hence by this process, we will usually drop the not important features that obtain from the feature selector, then use the processed X, and X2 for the classification and other prediction techniques.

[Boruta]

Using the smote dataset for Feature Selection

---------Top 5----------

	Features	Score
16	FAMILYEXPENSES(MONTH)	1.00
17	ANNUALSALARY	0.95
0	AGE	0.91
7	RACE	0.86
13	HOMEADDRESS	0.82

---------Bottom 5----------

	Features	Score
1	GENDER	0.18
10	MOVINGTONEWCOMPANY	0.14
22	MEDICALCOMPLICATION	0.09
9	MALAYSIAPR	0.05
21	TRANSPORT	0.00

[RFE]

---------Top 5----------

	Features	Score
0	AGE	1.0
1	GENDER	1.0
20	PURCHASEDPLAN1	1.0
19	CUSTOMER_NEEDS_2	1.0
18	CUSTOMER_NEEDS_1	1.0

---------Bottom 5----------

	Features	Score
2	MARITALSTATUS	1.00
22	MEDICALCOMPLICATION	0.75
10	MOVINGTONEWCOMPANY	0.50
21	TRANSPORT	0.25
9	MALAYSIAPR	0.00

Figure 11: Features Score for PURCHASEDPLAN2

3.2 Optimal Features Selection

Hence from the figure above, we can obtain the most important features and drop those that with least score from the result. From our studies, those with least score which might not that suitable to be put into the training set to predict the y. So in our case, the we drop the TRANSPORT, GENDER, MOVINGTONEWCOMPANY, HIGHESTEDUCATION and MOVINGTONEWCOMPANY, MEDICALCOMPLICATION, MALAYSIAPR, GENDER for X and X2 respectively. Besides for the data that without smote we have do the feature selection as well in order to determine the accuracy before and after smote for the classfication.

```
#Optimal Feature Set used to predicting PurchasePlan1 (After Smote)
Index(['AGE', 'MARITALSTATUS', 'SMOKERSTATUS', 'LIFESTYLE',
    'LANGUAGESPOKEN', 'RACE', 'NATIONALITY', 'MALAYSIAPR',
    'OCCUPATION', 'TELCO', 'HOMEADDRESS', 'RESIDENTIALTYPE',
    'NOOFDEPENDENT', 'FAMILYEXPENSES(MONTH)', 'ANNUALSALARY',
    'CUSTOMER_NEEDS_1', 'CUSTOMER_NEEDS_2', 'PURCHASEDPLAN2',
    'MEDICALCOMPLICATION'],
dtype='object')
```

```
#Optimal Feature Set used to predicting PurchasePlan2 (After Smote)
Index(['AGE', 'MARITALSTATUS', 'SMOKERSTATUS', 'LIFESTYLE',
    'LANGUAGESPOKEN', 'HIGHESTEDUCATION', 'RACE', 'NATIONALITY',
    'OCCUPATION', 'TELCO', 'HOMEADDRESS', 'RESIDENTIALTYPE',
    'NOOFDEPENDENT', 'FAMILYEXPENSES(MONTH)', 'ANNUALSALARY',
    'CUSTOMER_NEEDS_1', 'CUSTOMER_NEEDS_2', 'PURCHASEDPLAN1'],
dtype='object')
```

```
#Optimal Feature Set used to predicting PurchasePlan1 (Before Smote)
Index(['AGE', 'GENDER', 'MARITALSTATUS', 'SMOKERSTATUS',
    'LIFESTYLE', 'LANGUAGESPOKEN', 'HIGHESTEDUCATION', 'RACE',
    'NATIONALITY', 'OCCUPATION', 'TELCO', 'HOMEADDRESS',
    'RESIDENTIALTYPE', 'NOOFDEPENDENT', 'FAMILYEXPENSES(MONTH)',
    'ANNUALSALARY', 'CUSTOMER_NEEDS_1', 'CUSTOMER_NEEDS_2',
    'PURCHASEDPLAN2'],
dtype='object')
```

4 Data Mining Techniques

After finishes features selection, we decided to go into classification, clustering and association rules mining. The main reason we go into the classification steps is because it will assign the features in a collection to the target data. Besides, the goal of the classification is also to predict the target data for each case in the data. Furthermore, association rule mining in helps us to analyze the customer behaviors. In association rules mining, it plays a significant part in analyzing customer and product clustering in order to recommend the product to the other users. Moreover, in the clustering technique, it will usually cluster the groups of objects which are similar to each other and differentiate it from other clusters. For example, Cluster A and Cluster B will have different group of user who tends to buy different sorts of product and it will be clustered and separates for a clearer view.

4.1 Association Rule Mining

As we mention above, association rule mining can help us to analyse a customer behaviors. This data set consists of the customer historical purchase plan. So we apply the association rules mining to analyse the pattern of the historical purchased plan. In the association rules, it contains the if-then patterns which are antecedent and consequent. Besides, we used support to indicate how frequent the items appear in the data, meanwhile for the confidence it will indicate how true is the if-then rules. In our project, we used the apriori algorithm to generate the item sets and the rules. To the best of our knowledge, apriori algorithm reduce the number of rules which we can set the min-support and threshold in the parameter based on the algorithm below to we can get the most interest support and threshold in order to show the rules studies by [2], [3], [6] , [7] and [5]

```
support=rules["support"].values
confidence=rules["confidence"].values
import matplotlib.pyplot as plt
for i in range (len(support)):
   support[i] = support[i] + 0.0025
   confidence[i] = confidence[i] + 0.0025
plt.scatter(support, confidence, alpha=0.5, marker="*")
plt.xlabel('support')
```

```
plt.ylabel('confidence')
plt.show()
```

4.2 Classification Models

We decided to use various classification techniques which is random forest, k-nearest-neighbors (KNN), naive bayes and support-vector machine (SVM) as our choice of classification algorithms. To obtain the best fit model to our data set, we experimented and applied different algorithms by tuning the parameter to determine the best of parameter which fits our classifier. For the classification report, since it has the multilabel we need included the multiclass function in the parameter, studies by [4]

Each classifiers has its own principle and works different from each other. Random forest is an ensemble method, it comprise numbers of decision trees; for the KNN, it is a supervised and usually focuses on finding the similarities between the observations. Furthermore, the Naive Bayes, a supervised classifier, which works as a probabilities estimator by generating the probabilities for each of the class. Lastly the SVM, it tends to find the best split between the categories.

Therefore, in our project, we separate the target data which are PURCHASEDPLAN1 and PURHCASEDPLAN2 into the y and y2 respectively and also dropping the least significant features from the feature selection process to classify them. After several experiments, we conclude that the random forest performs the best overall. In the findings sections we will explain more on the methods' accuracy.

4.3 Clustering

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters. Furthermore, in our clustering process, we perform the K-means clustering which is an unsupervised learning as it will help us to find the groups without pre-defined the label to the data. In this process, it helps

us to assign the data point to each of the K groups based on their features similarities. Moreover, K-means Clustering helps in our project as it will characterize the clusters based on their behaviors. In terms of choosing the number of K, we have used the algorithms below to help us to identify the best K to use in our project studies by [1].

```
distortions = []
for i in range(1,11):
    km = KMeans(
        n_clusters=i, init='random',
        n_init=10, max_iter=300,
        tol=1e-04, random_state=0
    )
    km.fit(X)
    distortions.append(km.inertia_)
plt.plot(range(1, 11), distortions, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Distortion')
plt.show()
```

After applying all the machine learning algorithms, we can now measure the algorithms' accuracy and the fitness of model to the data. In clustering, we uses the Silhouette coefficient to obtain the fitness value. Silhouette coefficient is usually a metric that is used to calculate the goodness of a clustering technique.

5 Findings

In this section, we will explain more on the result and the discussion from our project from the beginning to the end.

5.1 Exploratory Analysis

5.1.1 Target Data

First, we plotted a stack bar chart to see the frequency of each values in PURCHASEDPLAN 1 and 2 with CUSTOMERNEED 1 and 2 separately.

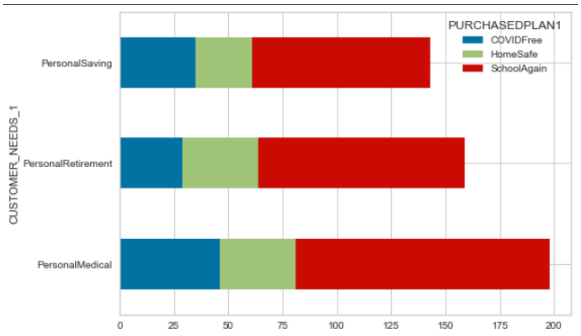


Figure 12: Stacked bar chart of PURCHASEDPLAN1 and CUSTOMERNEED1

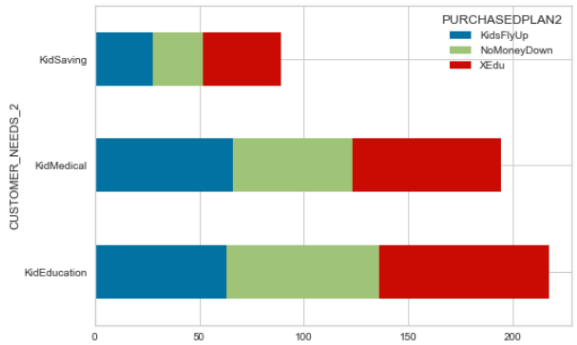


Figure 13: Stacked bar chart of PURCHASEDPLAN2 and CUSTOMERNEED2

We could observe that SchoolAgain are bought more frequently than others in all the CUSTOMERNEED1. While PURCHASEDPLAN2 shows a more consistent plot as each plan are bought equally in a balanced ratio.

5.1.2 Bar Plot

Below were some of the attributes with their respective bar plot. Some attribute could be clearly see the unbalanced between the data.

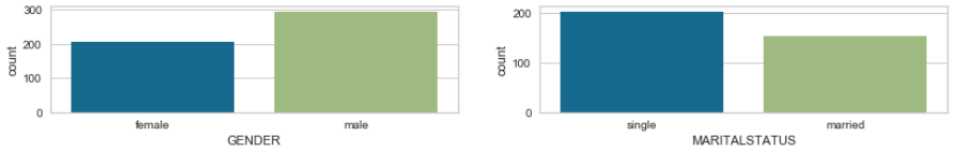


Figure 14: Bar Plot of Raw Data 1

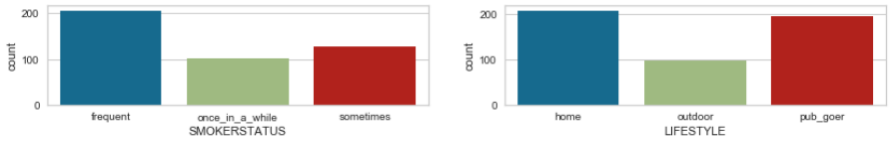


Figure 15: Bar Plot of Raw Data 2

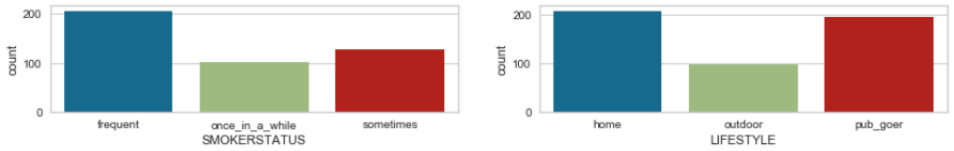


Figure 16: Bar Plot of Raw Data 3

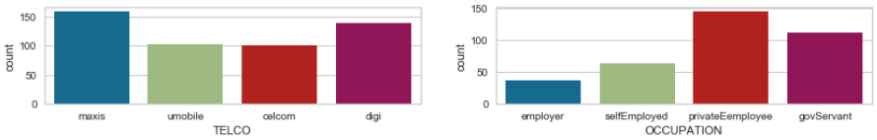


Figure 17: Bar Plot of Raw Data 4

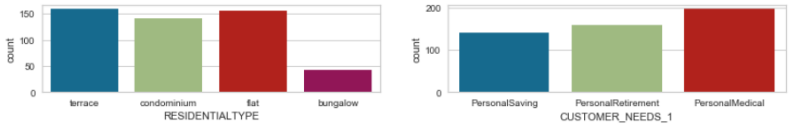


Figure 18: Bar Plot of Raw Data 5

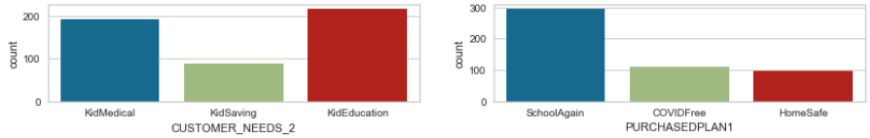


Figure 19: Bar Plot of Raw Data 6

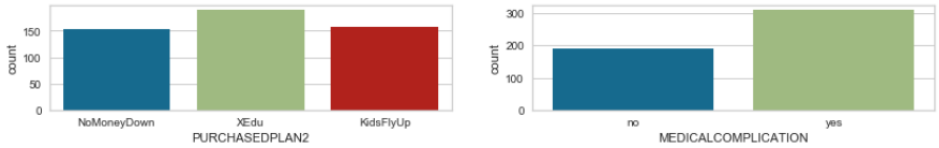


Figure 20: Bar Plot of Raw Data 7

5.1.3 Crosstab

This plot below shows the relationships between CUSTOMERNEED1, 2 and PURCHASEDPLAN1, 2. We could see that KidSaving is the least needed while SchoolAgain is most bought as compared with PURCHASEDPLAN2.

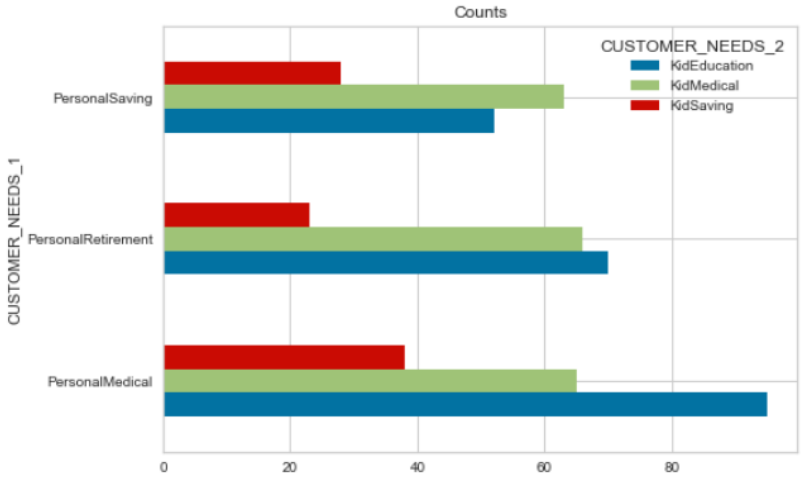


Figure 21: Crosstab of CUSTOMERNEED1 and CUSTOMERNEED2

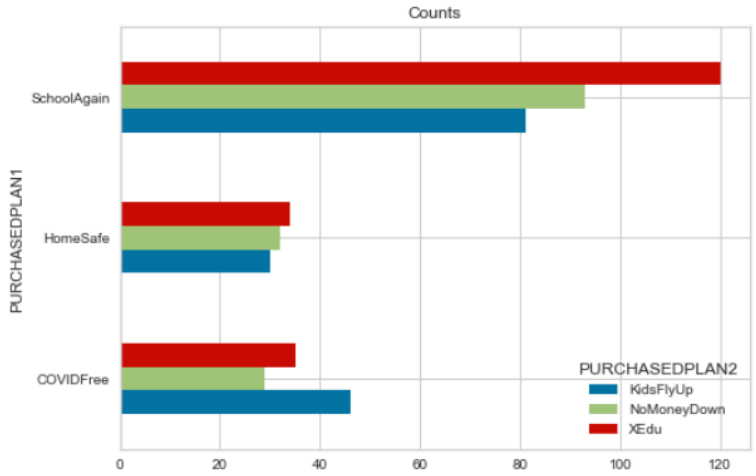


Figure 22: Crosstab of PURCHASEDPLAN1 and PURCHASEDPLAN2

Hence to further understand between PURCHASEDPLAN1 and the PURCHASEDPLAN2, we join the customer needs with the purchaseplan to provide us a better visualization for the understanding. The figures below shows the the combination of CUSTOMERNEED1 + PURCHASEDPLAN1 and CUSTOMERNEED2 + PURCHASEDPLAN2.

In Figure 23 and 24, we can clearly see that the the customer who bought the PersonalMedical tends to buy SchoolAgain instead of buying

Homesafe. Besides there is least numbers of customers who buy homesaving while their needs is homesafe. Furthermore from the graph we can see that if the customer need is KidEducation, they tend to buy Xedu instead of KidsFlyUp. From Figure 24, we can see that KidEducation is the most bought for CUSTOMERNEEDS2 and PURCHASEDPLAN2.

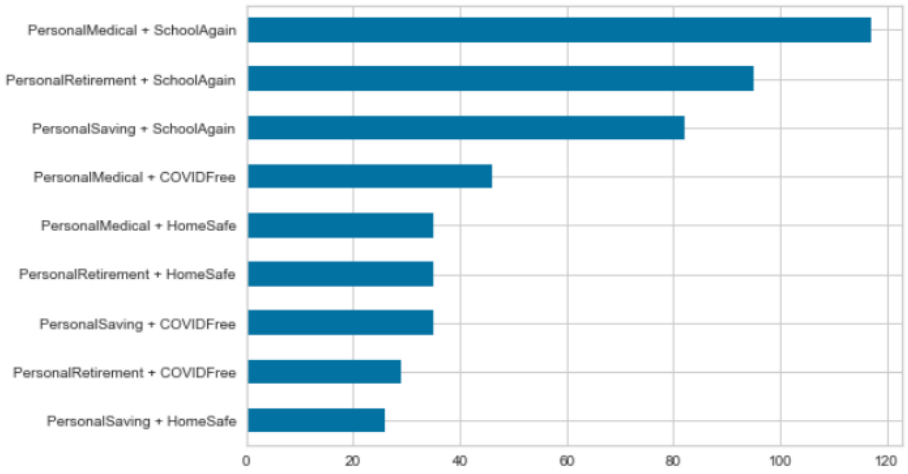


Figure 23: Crosstab of CUSTOMERNEED1 and PURCHASEDPLAN1

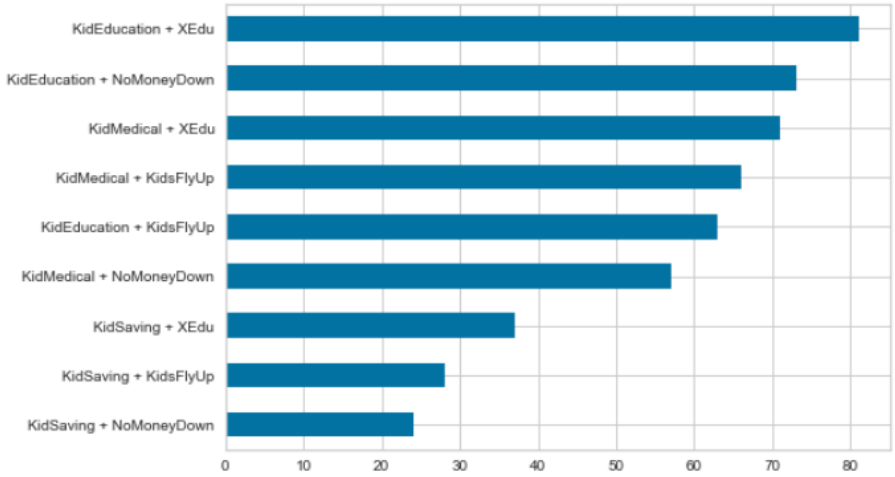


Figure 24: Crosstab of CUSTOMERNEED2 and PURCHASEDPLAN2

5.1.4 Box Plot

Furthermore, the box plot shown in belows are the customer's purchase ability within the age range. We could conclude that most buyers are around 120000, except the 10-20 age group, we suspect that this may be due to independence of finance as they may still be studying. Boxplot 3 and 4 shows the expenses and number of dependents of a family in PUCHASEDPLAN1 and 2. Both graphs allows the employers to decide on the prices of the plan.

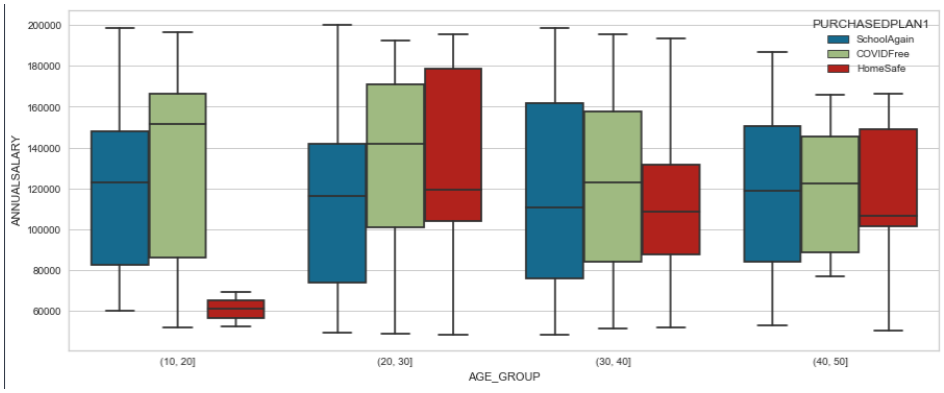


Figure 25: Boxplot of AGE, ANNUALSALARY and PURCHASEDPLAN1

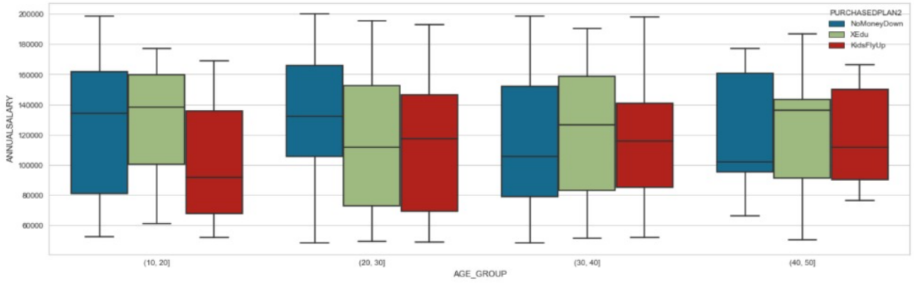


Figure 26: Boxplot of AGE, ANNUALSALARY and PURCHASEDPLAN2

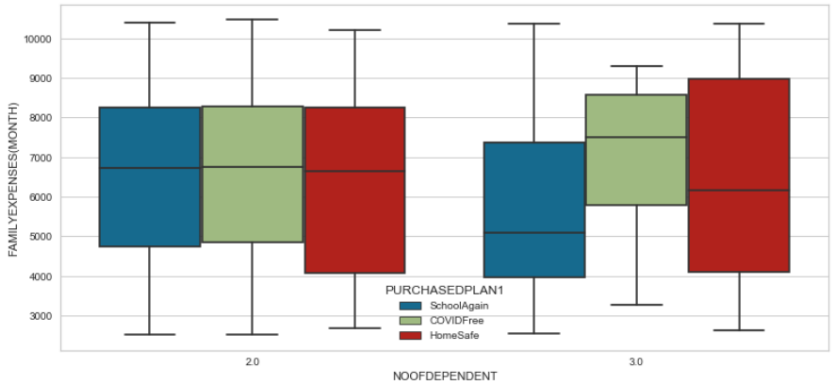


Figure 27: Boxplot of NOOFDEPENDENT, FAMILYEXPENSES and PURCHASEDPLAN1

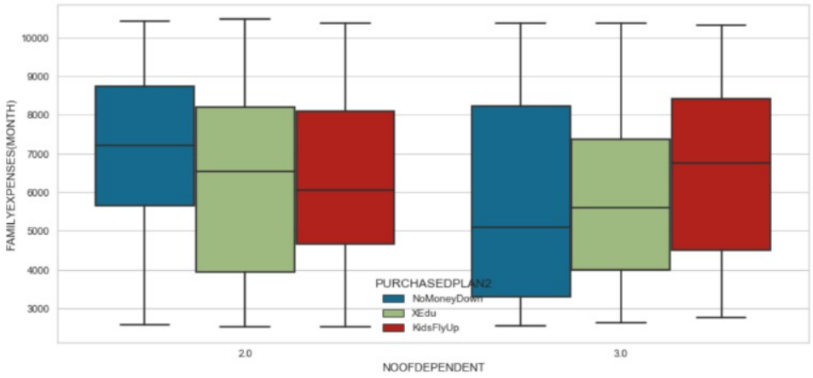


Figure 28: Boxplot of NOOFDEPENDENT, FAMILYEXPENSES and PURCHASEDPLAN2

5.2 Result of Association Rules Mining

The figures 29 and 30 below show the rules of association rules mining which we set the minimum support as 0.05 and the threshold as 0.05 for CUSTOMERNEEDS1 and PURCHASEDPLAN1. For example in the rules 1, the meaning of the support, confidence and the lift which are supported as an indication on how frequently the item appears, confidence indicates how that the if-then statement is true and the lift means how many times the if-then statement is expected to be found true.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
12	(PersonalMedical)	(SchoolAgain)	0.396	0.588	0.234	0.590909	1.004947	0.001152	1.007111
13	(SchoolAgain)	(PersonalMedical)	0.588	0.396	0.234	0.397959	1.004947	0.001152	1.003254
15	(PersonalRetirement)	(SchoolAgain)	0.318	0.588	0.190	0.597484	1.016130	0.003016	1.023562
14	(SchoolAgain)	(PersonalRetirement)	0.588	0.318	0.190	0.323129	1.016130	0.003016	1.007578
16	(PersonalSaving)	(SchoolAgain)	0.286	0.588	0.164	0.573427	0.975215	-0.004168	0.965836

```
(Rule 1) PersonalMedical -> SchoolAgain
Support: 0.234
Confidence: 0.591
Lift: 1.005
====================================
(Rule 2) SchoolAgain -> PersonalMedical
Support: 0.234
Confidence: 0.398
Lift: 1.005
====================================
(Rule 3) PersonalRetirement -> SchoolAgain
Support: 0.19
Confidence: 0.597
Lift: 1.016
====================================
(Rule 4) SchoolAgain -> PersonalRetirement
Support: 0.19
Confidence: 0.323
Lift: 1.016
====================================
(Rule 5) PersonalSaving -> SchoolAgain
Support: 0.164
Confidence: 0.573
Lift: 0.975
====================================
```

Figure 29: Rules of CstomerNeeds1 and PurchasePlan1

The figures below show the rules of association rules mining which we set the minimum support as 0.05 and the threshold as 0.05 for CUSTOMERNEEDS2 and PURCHASEDPLAN2.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
4	(XEdu)	(KidEducation)	0.378	0.434	0.162	0.428571	0.987492	-0.002052	0.990500
5	(KidEducation)	(XEdu)	0.434	0.378	0.162	0.373272	0.987492	-0.002052	0.992456
2	(NoMoneyDown)	(KidEducation)	0.308	0.434	0.146	0.474026	1.092226	0.012328	1.076099
3	(KidEducation)	(NoMoneyDown)	0.434	0.308	0.146	0.336406	1.092226	0.012328	1.042806
11	(XEdu)	(KidMedical)	0.378	0.388	0.142	0.375661	0.968199	-0.004664	0.980237

```
(Rule 1) XEdu -> KidEducation
Support: 0.162
Confidence: 0.429
Lift: 0.987
====================================
(Rule 2) KidEducation -> XEdu
Support: 0.162
Confidence: 0.373
Lift: 0.987
====================================
(Rule 3) NoMoneyDown -> KidEducation
Support: 0.146
Confidence: 0.474
Lift: 1.092
====================================
(Rule 4) KidEducation -> NoMoneyDown
Support: 0.146
Confidence: 0.336
Lift: 1.092
====================================
(Rule 5) XEdu -> KidMedical
Support: 0.142
Confidence: 0.376
Lift: 0.968
====================================
```

Figure 30: Rules of CstomerNeeds2 and PurchasePlan2

5.3 Model Accuracy

Table 1: Comparing Two Different Methods For PURCHASEDPLAN1

Accuracy (%)					
Average	Precision	Recall	F1 Score	Train	Test
Random Forest	83	82	83	100	83
Naive Bayes	55	55	55	59	55
KNN	69	64	61	70	64
SVM	76	76	75	93	76

Table 2: Comparing Two Different Methods For PURCHASEDPLAN2

Accuracy (%)					
Average	Precision	Recall	F1 Score	Train	Test
Random Forest	40	40	39	100	40
Naive Bayes	44	44	43	52	43
KNN	48	48	44	55	48
SVM	44	43	42	86	43

Table 3: Comparing Two Different Methods For PURCHASEDPLAN1 No SMOTE

Accuracy (%)					
Average	Precision	Recall	F1 Score	Train	Test
Random Forest	45	56	43	100	56
Naive Bayes	32	57	41	59	57
KNN	42	50	43	59	50
SVM	32	57	41	99	57

From our experiments, we can see that the Random Forest Classifier has obtained for the highest accuracy which is 0.83 compared to the naive bayer, knn and the svm for our target data 1. Besides, we have also experimented the accuracy of PURCHASEDPLAN1 without smite the data set. In the table, the result clearly shows that the accuracy after smote is much higher than the accuracy without smote. Furthermore, for our target data 2, it get a low accuracy for all the classifiers, hence we have concluded the target data 2, X2 is not suitable for use to predict to y2, PURCHASEDPLAN2. Meanwhile since the classifier of the target data 1 is high, we also make a conclusion that the X is suitable use to predict the y, PURCHASEDPLAN1 after smote the data set.

5.4 Clustering Analysis

The figures 31, 32 and 33 below show that we cluster the PURCHASED-PLAN1 based on the customer age, salary and the expenses. Figure 31 shows the left side which is before clustering and the right side which has been clustered. From the plot we can see that after cluster it has specified how much the salary per month of the customer tends to buy which plan of the insurance product. While the following plot (Figure 32) shows the clustering result based on age and salary. After clustering, we have conduct the calculation for the silhouette score and a accuracy of 0.5965 as shown below.

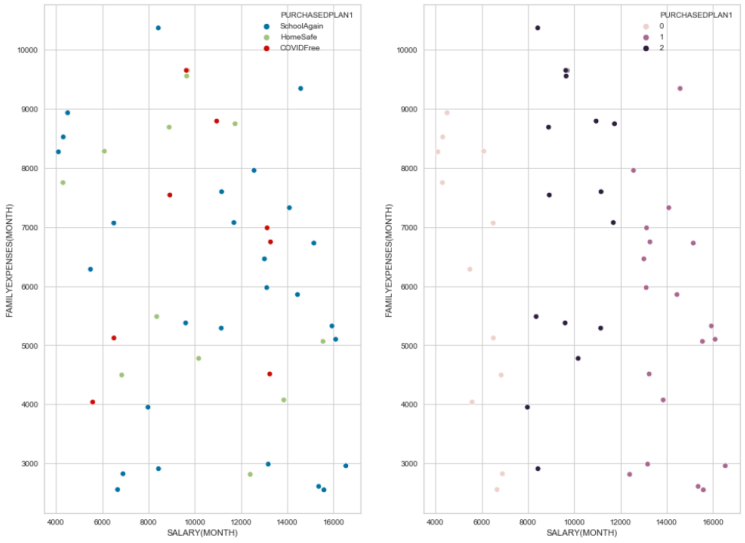


Figure 31: Before and After Clustering based on the Salary and Expenses

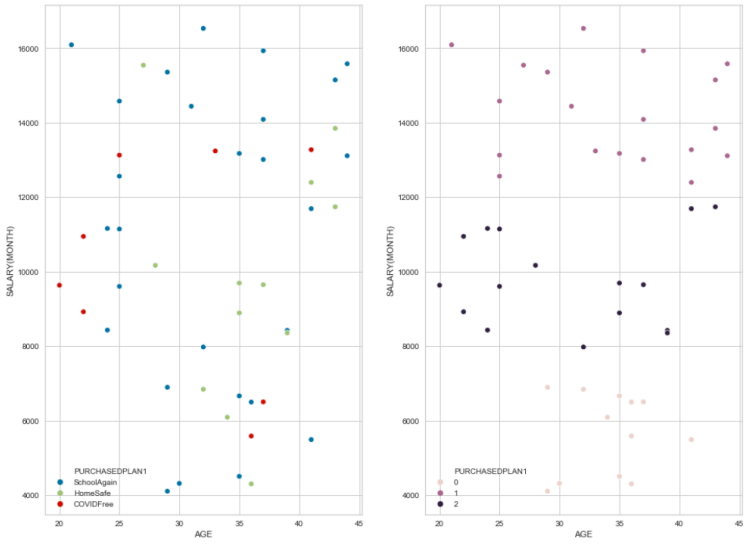


Figure 32: Before and After Clustering based on the Age and the Salary

```
0.5965359432896206
```

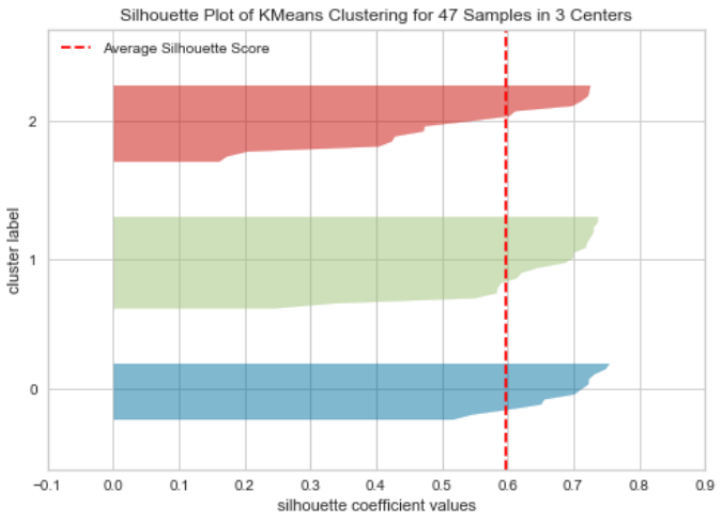


Figure 33: Silhouette Score of the Clustering

6 Deployment

In our project, the work is visualized via streamlit. In our Web App, we can select which section we want to access from the sidebar. We also can select which row of data from the original data frame we want to view. Furthermore, we are able to select which graph to be viewed by selecting the feature. In addition, we can select different data mining techniques as shown in Figure 34. Moreover, we are free to change the classification algorithms towards both of our target data. In our Web App allows to view data before or after clustering in order to have a clear comparison. Ultimately, we can select which target data to be performed under Association Rule Mining

The prototype can be accessed at https://dmpf.herokuapp.com/
Sample screenshots are shown below:

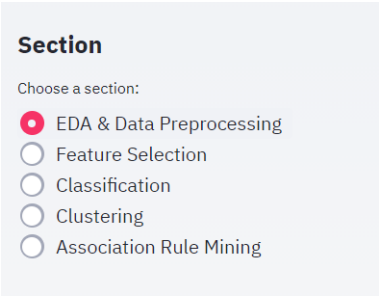


Figure 34: Web App sidebar

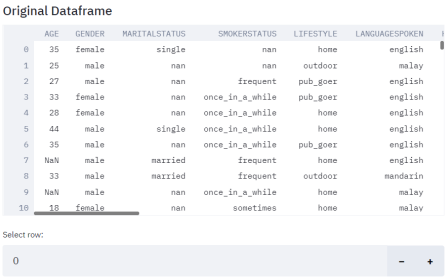


Figure 35: Original Dataframe and Row Selection

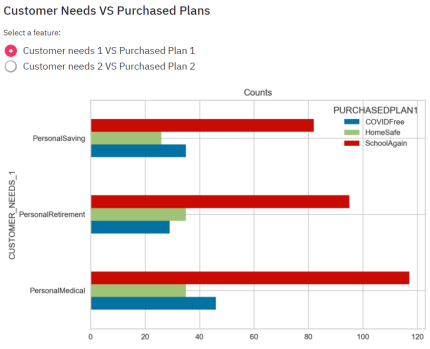


Figure 36: Graph Selection

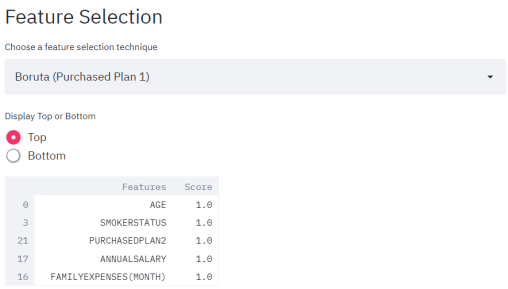


Figure 37: Feature Selection Technique Selection

Figure 38: Classification Techniques Selection

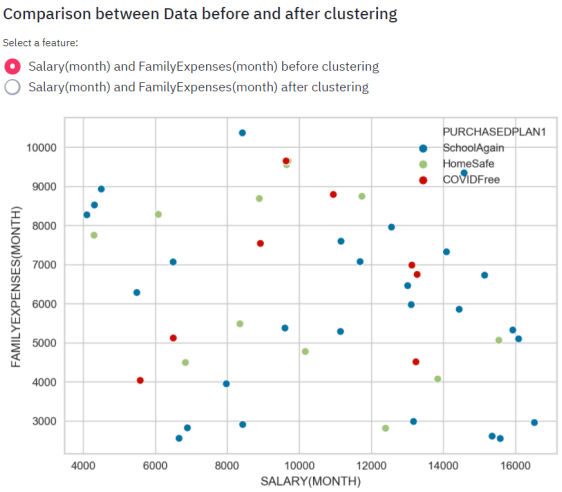


Figure 39: Selection of data before or after clustering

Figure 40: Target Data and View Selection for Association Rule Mining

7 Conclusion

This project provides us an opportunity to hands-on and apply those techniques we learnt in data mining. In this report, it has outlined the process of us while doing our experiment and justification.

First and foremost, we need to understand the data mining pipeline such as preprocessing, mining and visualization. Besides, we must understand all the features in the dataset before we apply those data mining techniques into it. For example, in the pre-processing, we need to find out what the target data of us and what we need to do with it. Nevertheless it is also important that if we simply fill in those NaN values, it will give us a bad performance in the classification if we do not do the pre-processed correctly.

Moverover, before we apply the machine learning techniques into the dataset. The things we do is to select the important features out and only apply the techniques in. For example, in this project, we have applied the Boruta and RFE to help us to select the features and we drop the least significant features at the end to generate our optimal feature set for applying machine learning techniques.

Furthermore, we conduct the experiment with different algorithms in order to get our accuracy and determine which classifier works the best for us and we obtain the highest which is 0.83 by random forest compared to others classification. Not only classification we did, we do for the association rules mining in our project which will help us to analyse the purchase pattern of the customer behaviors and generate the rules based on the apriori algorithms. Nevertheless, the clustering data mining technique has also been applied to cluster the data into numbers of groups based on their features similarities.

In a nutshell, in this project we learned a lot about how a data mining pipeline works and of course we are sure there is improvement but for now this output has been our best effort.

References

[1] *Data Mining - Cluster Analysis*, 2020 (accessed September 20, 2020).

[2] Harsh. *Association Analysis in Python*, 26 September 2019.

[3] Usman Malik. *Association Rule Mining via Apriori Algorithm in Python.*

[4] Javaid Nabi. *Machine Learning — Multiclass Classification with Imbalanced Dataset*, 23 December 2018.

[5] owygs156. *How to Create Data Visualization for Association Rules in Data Mining*, 2018 (accessed September 20, 2020).

[6] rasbt. *TransactionEncoding.*

[7] Margaret Rouse. *What are Association Rules in Data Mining (Association Rule Mining)?*, 2020 (accessed September 20, 2020).