MSc Business Analytics

MIS41270 Data Management and Mining

Data Analytics Report

Assignment: Insurance Marketing 2021

UCD Smurfit School of Business
Lecturer – Aoife D'Arcy

Group 12

Student Name:	**Student ID:**
Ebison George	20200231
Sidharth Mohapatra	20200224
Vaishnavi Gadekar	20200073

Table of Contents

BUSINESS UNDERSTANDING

Business Objective

Insure ABC is a general insurance company, which is planning to launch a new home insurance product in the coming months. A data-driven marketing strategy is to be formulated for this product, targeting the existing customers, who are most likely to purchase the new product.

Analytical Approach

As part of the analytical team of Insure ABC, we are required to analyze the historical customer data and provide solutions based on it. For this, we are using the Crisp-DM methodology. Following are the three analytical solutions that we have used to meet the requirements:

1. **Predictive Analytics**: We predicted which customer type is likely to buy the new home Insurance product, and what would be the appropriate communication channel.
    a. Clustering Model: For marketing purposes, the division of the data into different datasets depending on specific data attributes is needed, and in this case the clustering model is useful. We used a clustering algorithm, because there is no target variable, and an unsupervised learning approach is needed to be applied which will discover the patterns and information on its own.
    b. Linear SVC: This supervised algorithm is used for predicting the preferred communication channel.
2. **Descriptive Analytics**: The existing customer dataset is used to assess the relative importance of the different features in it. Exploratory Data Analysis (EDA) was carried out using descriptive analysis and the quality of data was assessed which helped us find the outliers, missing values, and data anomalies.
3. **Prescriptive Analytics**: Different marketing strategies can be prescribed to target different customer segments.

Available Resources

A training dataset is provided which will be used for building and training the model that would predict the preferred communication channel. And a scoring dataset is provided for assessing the accuracy of the built model. The split of the training and scoring data is approximately 75% and 25% respectively.

DATA UNDERSTANDING

The training dataset contains 4090 records, with 20 features. The preferred channel has been selected as the target variable, with respect to which the analytics has been carried out.

Exploratory Data Analysis

Descriptive Analytics of the Training data

	CustomerID	Age	MotorValue	HealthDependentsAdults	HealthDependentsKids
count	4090	4090	3361	2543	2543
mean	2604.479	41.391	23450.911	0.816	1.748
std	1498.310	15.986	11985.631	0.646	1.108
min	1	-44	-25686	0	0
25%	1295.25	22	14837	0	0
50%	2594.5	46	25045	1	2
75%	3908.75	50	32289	1	3
max	5200	210	325940	2	3

Table 1:Training Data Descriptive Statistics

Heat Map for correlation between the features in Training Data

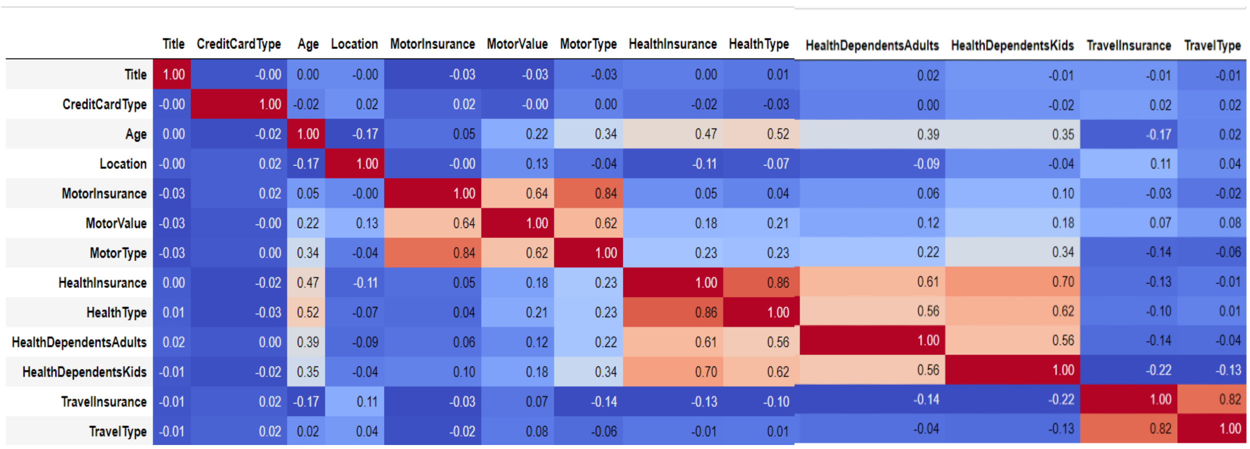


Figure 1:Feature Correlation

Level of Data

- Interval – Age, Id, MotorValue
- Binary – Credit card type, Healthinsurance, location
- Nominal – Gender, GivenName, HealthdependantAdults, HealthdependentKids, HealthType, MiddleInitial, PrefChannel

Data Quality Report

Feature	Data Quality Issue	Potential Handling Strategies
CreditCardType	Missing Values (17.65%)	Replace null by 0
Occupation	Missing Values (38.04%)	Replace null by 0
Gender	Inconsistent Labelling	Replacing ‘m’ and ‘f’ by ‘Male’ and ‘Female’ respectively.
Age	Outliers (low)	Corrected the values as per assumptions. Ex: -44 changed to 44, and 210 to 21.
MotorValue	Outliers (low) Missing values (17.82%)	Corrected the values as per assumptions. Ex: -4679 to 4679.
MotorType	Missing Values (17.82%)	Replace null by 0
HealthType	Missing Values (37.82%)	Replace null by 0
HealthDependentAdults	Missing Values (37.82%)	Replace null by 0
HealthDependentKids	Missing Values (37.82%)	Replace null by 0
TravelType	Missing values (48.45%)	Replace null by 0
PrefChannel	Inconsistent Labelling	Replacing ‘S’, ‘E’, and ‘P’ by ‘SMS’, ‘Email’ and ‘Phone respectively.

Table 2:Data Quality Report

DATA PREPARATION

Scaling and labeling the data

- The ‘GivenName’, ‘MiddleInitial’, ‘Surname’ columns are dropped as these features don’t have any dependency on the target variable.
- The ‘Occupation’ column is avoided as we can see from the data that each occupation covers only 1-5 members and most records are found to be missing for this feature.
- We can see that depending on the type of features and correlations between them there are a significant number of null values and categorical variables. For example, if the feature, ‘MotorInsurance’ is marked as yes then the ‘Motorvalue’ consists of a value, however, if the ‘MotorInsurance’ is marked as No, then the value is null. Therefore, we impute the null values wherever applicable using a scaled value between -1 to 1.
- ‘PrefChannel’, which is the target variable had 6 categories instead of 3, the extra three being ‘E’, ‘P’, and ‘S’. So, we labeled them as ‘Email’, ‘Phone’, and ‘SMS’ respectively.
- The Age column had highly variable data. Therefore, we scaled it using the MinMaxScaler.
- The Title column has ‘Mrs.’, ‘Mr.’, and ‘Ms.’ So we ignored gender as a column as we can map the gender from these titles.
- The other categorical variables with multiples classes can be encoded using a LabelEncoder as the categories are distinguishable and less in number. The Yes/No binary columns of ‘MotorInsurance’, ‘HealthInsurance’, and ‘TravelInsurance’ are encoded into a 1/0, as it is easier to train the model using numeric data instead of string data. Similarly, the ‘MotorType’, ‘HealthType’, and ‘TravelType’ which have different categories of insurances are encoded into integer values for making it easier to train the model. The same is done with the ‘PrefChannel’ column. ‘CreditCardType’ has 2 types – AMEX and Visa, which are also encoded into integer values, and the null values are replaced by 0 to have integer data. The encoding is written in the appendix for reference (page 16).

Analytics Base Table (ABT)

After preparing the data as per the relevance and our needs, the Analytics Base Table is created which will be used for modeling and prediction. The first five rows of the ABT are:

CustomerID	Title	CreditCardType	Age	Location	MotorInsurance	MotorValue	MotorType	HealthInsurance	HealthType	entsAdults	identsKids	TravelInsurance	TravelType	PrefChannel
1	2	1	0.26378	1	0	0.073049	0	0	0	0	0	1	3	2
2	3	1	0.346457	1	0	0.073049	0	1	1	2	3	0	0	1
4	3	1	0.248031	1	1	0.087041	1	0	0	0	0	0	0	2
5	1	2	0.358268	0	1	0.115691	2	1	1	1	2	1	2	1
11	2	2	0.385827	0	1	0.114585	2	1	2	2	3	0	0	0

Figure 2: ABT

MODELING

Customer Segmentation – Unsupervised Learning Approach

The new product has to be marketed to the existing customers which belong to various segments. Therefore, to make the marketing more organized the customers have to be categorized into different segments and each group can be targeted uniquely.

Principal Component Analysis

After the data cleaning and preparation stage, there are 15 features, out of one which is our target variable. To create the clusters, we had to decide which features would lead to maximum information gain. However, due to a large number of features, finding out the maximum information gain from the related columns would be difficult. Therefore, we used the Principal Component Analysis (PCA) technique which will algorithmically choose the most relevant features for creating the customer segments.

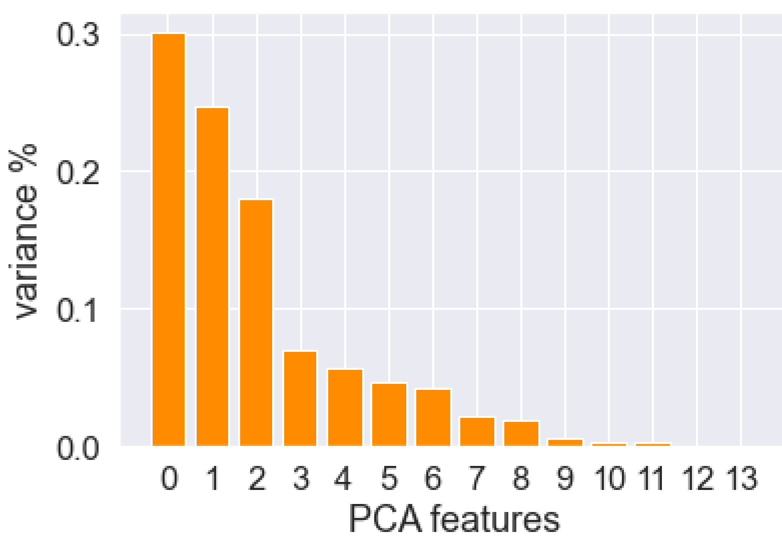


Figure 3: PCA showing relevant features

From the above graph, we can see that after the first three features, there’s a dip in the importance of the other categories. So, we consider the first three features (0, 1, 2) for creating the clusters. The features are selected by the algorithm by determining the most correlated features, so there’s no manual intervention required, and we get the most relevant features to create the clusters.

Elbow Graph

Further, to identify how many categories the customer base is to be divided into, the Elbow Graph technique was used. By specifying how many features to consider, and the PCA components’ data frame obtained from the PCA step, an Elbow Graph is plotted.

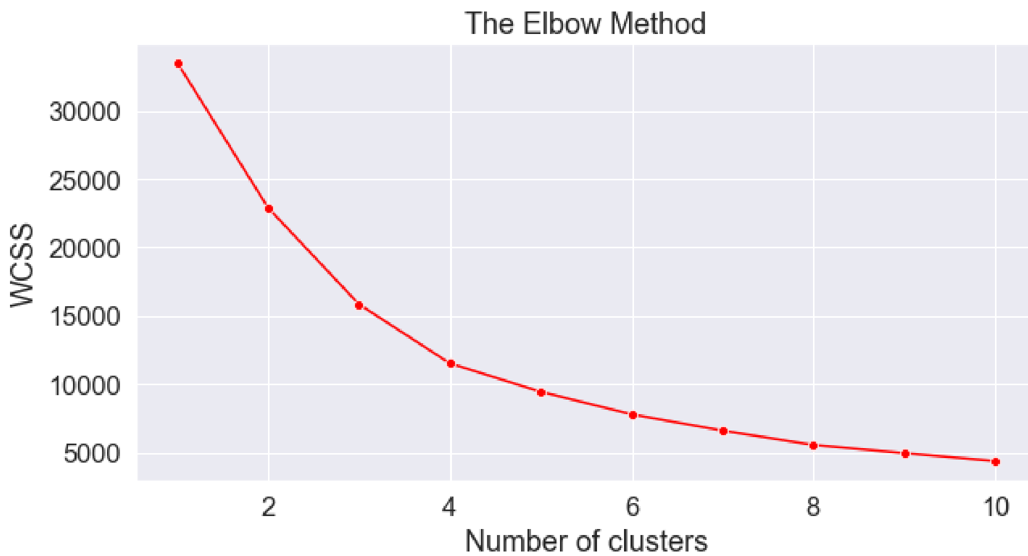


Figure 4: Elbow Graph

From the above Elbow Graph, it is clear that the elbow is closer to 3 clusters.

K-means Clustering

After finding the most distinguishable features and the number of categories to distinguish them in, a clustering algorithm is used to create a model. K-means clustering algorithm is used to cluster the customer base. The value of K is 3 which is derived from the Elbow Graph.

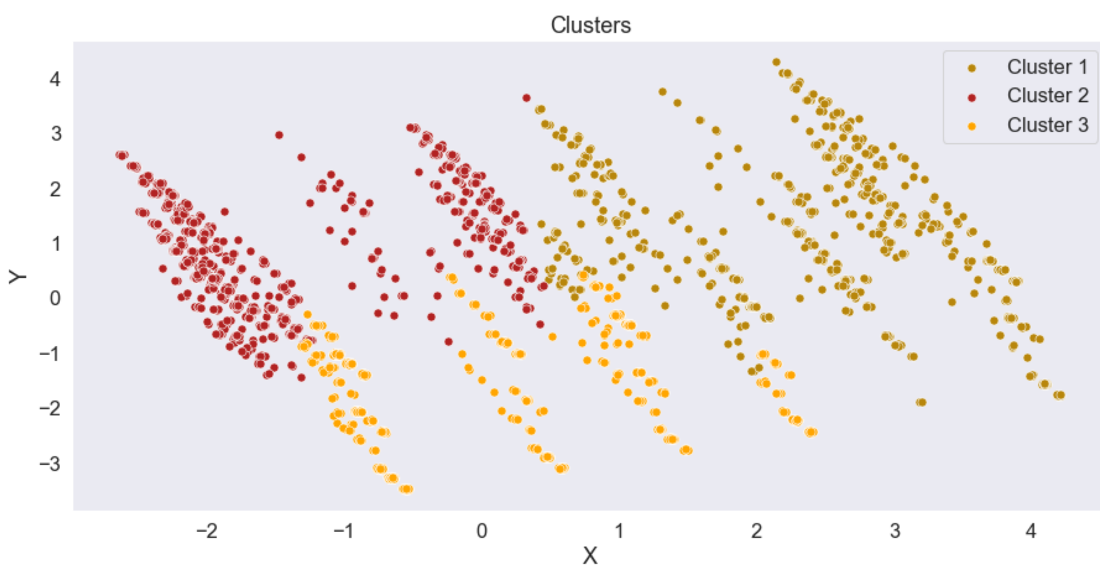


Figure 5: Clusters

The customers have been segmented into three groups labeled as 0, 1, 2 clusters.

Now, the correlation of the clustered data was compared with the features to check which features were highly relevant to the cluster like “Title”, “CreditCardType”, “Location”, “TravelInsurance”, and “TravelType”. This information was further used while deducing a marketing strategy.

Prediction Models – Supervised Learning Approach

Selection of the most accurate predictive model:

Several models were considered and evaluated to predict the optimum preferred channel for the customer segments that have been created. The accuracy and precision scores were calculated on the training data, and they were the deciding criteria to choose a supervised prediction model.

Following are the accuracy and prediction scores based on which the model was selected.

Model Type	Accuracy Score	Precision Score
Logistic Regression	63.73	38.36
Decision Tree	54.05	31.04
Linear SVC	67.15	41.06
Gaussian NB	20.91	29.89
K-Nearest Neighbor	60.70	36.33
Random Forest Classifier	64.90	38.36

Table 3:Model Selection

From the above table, we can see that the highest accuracy and precision scores belong to the Linear SVC model. Therefore, to score the data, Linear SVC is the most appropriate model.

Model validation using training data

Using the training dataset, we got the following results:

	Preferred Channel		
Clusters	**Email**	**Phone**	**SMS**
0	683	813	115
1	494	414	111
2	591	338	531

Table 4:Cluster-channel mapping

After segmenting the customer base into clusters 0, 1, and 2 we use the Linear SVC model on the training data to determine what is the segregation of the preferred communication channel for each cluster. The above table gives us an insight into how many people in each cluster prefer to be contacted via Email, Phone, and SMS.

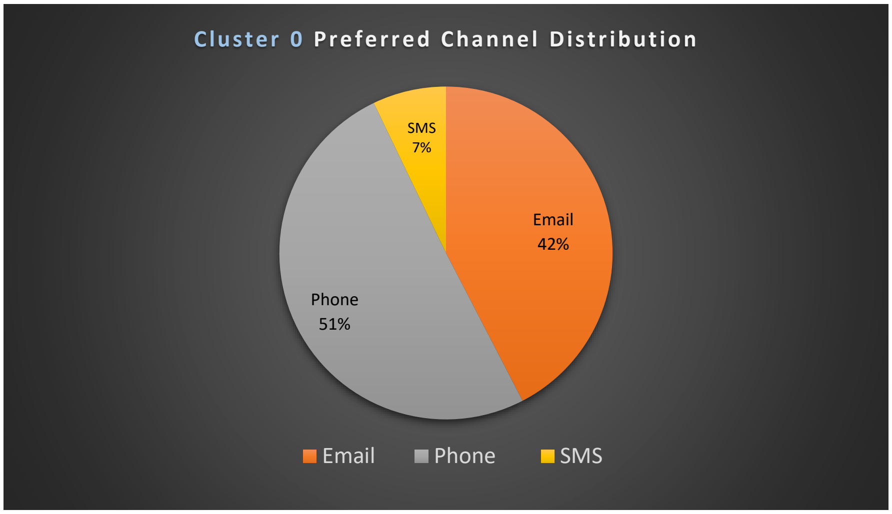


Figure 6: Cluster 0 Preferred Channel Distribution

It is evident that more than half the customers in cluster 0 prefer to be contacted via Phone. The second most preferred is Email with 42% of the customers in this segment opting for it. Only 7% of the entire customer base in Cluster 0 prefer to be contacted via SMS

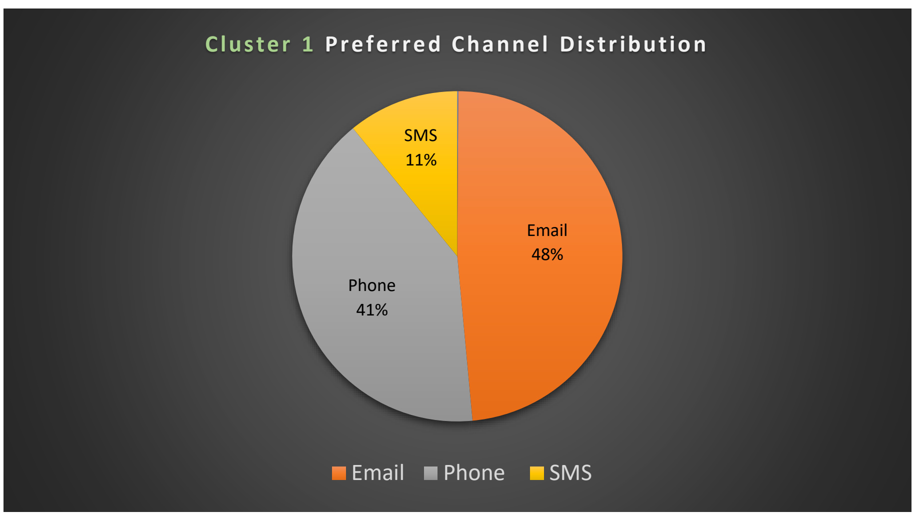


Figure 7:Cluster 1 Preferred Channel Distribution

In cluster 1, almost half of the customers (48%) prefer to be contacted via Email, followed by Phone (41%). Similar to cluster 0, SMS is the least preferred communication channel with only 11% opting for it.

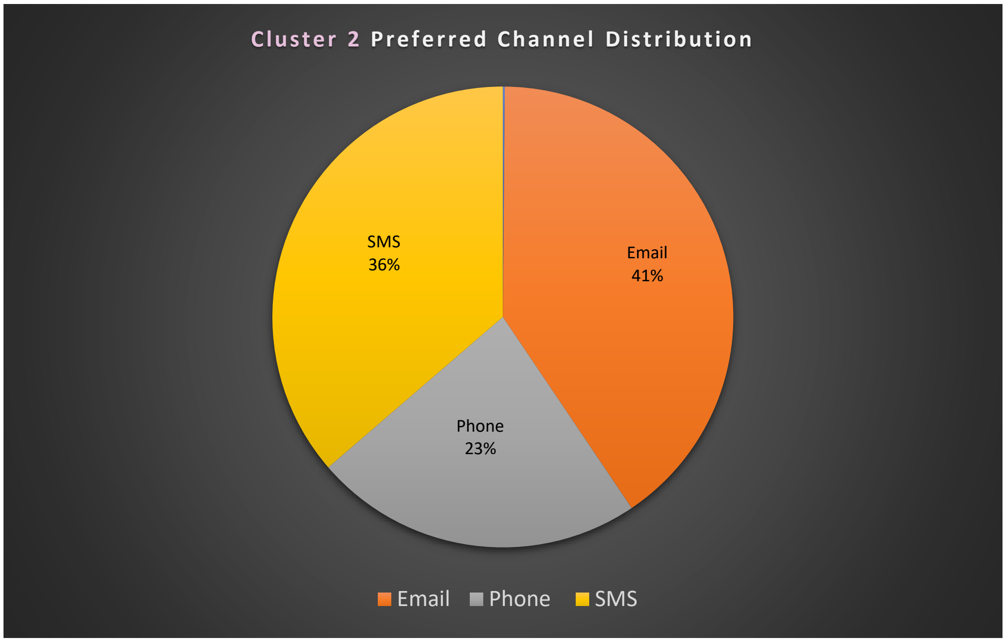


Figure 8: Cluster 2 Preferred Channel Distribution

In Cluster 2 again Email is most preferred, but SMS is the second most favored communication mode with 36% of customers going for it. Lastly, Phone is preferred by 23% of the customers in this particular customer segment.

In conclusion, we can say that in clusters 1 and 2, Email is the most preferred communication channel. If we look at the ratio of the numbers in each cluster divided by each preferred channel, we can say that in clusters 0 and 1, Email and Phone have the greatest number of customers, and SMS is the least significant mode of communication. There is a more balanced ratio in cluster 2 when it comes to this particular customer segment divided by how they should be contacted.

VALIDATION AND EVAULATION

Scoring the data

Out of the 6 predictive analysis models considered, Linear SVC has the highest accuracy and precision. Therefore, we used the Linear SVC to train the model using the training dataset for predicting what are the preferred communication channels for the customer segments. This trained model is then used for scoring the new dataset, where we predicted which customer belongs to what cluster and what would be the most appropriate mode of communication to approach them with the marketing strategy.

Descriptive statistics for the scored data

	CustomerID	Age	MotorValue	HealthDependentsAdults	HealthDependentsKids	PrefChannel
count	1091	1091	903	692	692	1091
mean	2569.260	41.475	23638.007	0.811	1.842	0.357
std	1520.131	14.984	14476.005	0.650	1.091	0.480
min	3	-48	-16888	0	0	0
25%	1303	22	15116	0	2	0
50%	2579	47	24658	1	2	0
75%	3896.5	50	32259	1	3	1
max	5189	77	320280	2	3	1

Table 5:Scored Data Descriptive Statistics

We have 1091 rows in the scoring dataset, with the mean Age and MotorValue as 41.47 and 23638.007 respectively. The mean of HealthDependentAdults is 0.811 with a minimum of 0, and a maximum of 2 dependent adults. On average, the dataset contains at least one Health Dependent Adults. Similarly, for HealthDependentKids, the average dependency is 2. The age is distributed between three quartiles, with the lowest quartile being 22, the median is 47 and the upper quartile is having the age of 50. Likewise, MotorValue is distributed as 15116, 24658, 32259 in the lowest, median, and the upper quartile range.

From the descriptive statistics, majoritarily Email and Phone is the preferred communication channel. In the first 50%, the Preferred Channel (PrefChannel) is Email (0, refer appendix), and in the last 25% is Phone (1, refer appendix) and in those quartiles, the ages are 22, 47, and 50 respectively. From these statistics, it can be inferred that younger people prefer emails, and older people would prefer Phones as the mode of communication, and the marketing team can advertise their products via these media respectively.

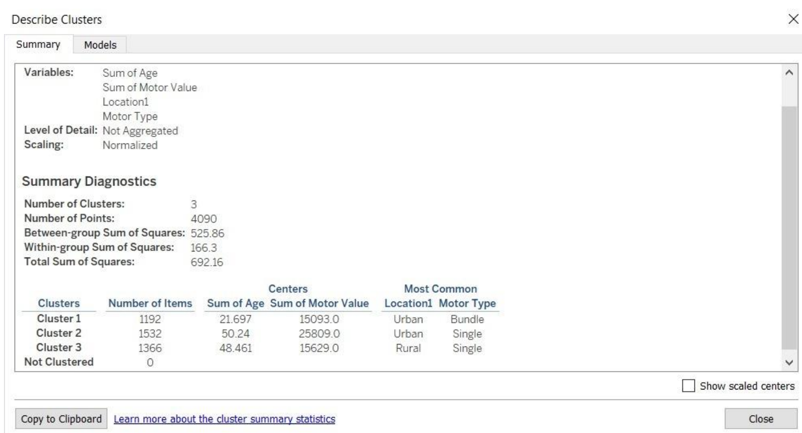
MARKETING STRATEGY

Customers can be categorized into different segments based on their purchasing behavior and their age. By performing the clustering technique, we have segmented the customer base into three different groups, which can be targeted and then approached depending on the preferred channel for marketing our new Home Insurance Product.

Following are the cluster descriptions from Python and Tableau, which provide us with the customer segments and what their attributes are:

PrefChannel	Title	CreditCardType	Location	TravelInsurance	TravelType	Cluster	PrefChannel
0	1.708987	1.229672	0.922967	0.606277	1.526391	0.814551	0.0
1	1.679487	1.184615	0.000000	0.394872	1.492308	0.741026	1.0

Figure 9: Correlated features


Describe Clusters

Summary | Models

Variables: Sum of Age
Sum of Motor Value
Location1
Motor Type
Level of Detail: Not Aggregated
Scaling: Normalized

Summary Diagnostics

Number of Clusters: 3
Number of Points: 4090
Between-group Sum of Squares: 525.86
Within-group Sum of Squares: 166.3
Total Sum of Squares: 692.16

		Centers		Most Common	
Clusters	Number of Items	Sum of Age	Sum of Motor Value	Location1	Motor Type
Cluster 1	1192	21.697	15093.0	Urban	Bundle
Cluster 2	1532	50.24	25809.0	Urban	Single
Cluster 3	1366	48.461	15629.0	Rural	Single
Not Clustered	0				

Show scaled centers

Copy to Clipboard | Learn more about the cluster summary statistics | Close

Figure 10:Tableau Insights

The marketing strategies that we can devise based on the cluster definitions and the prediction of the preferred channels are as follows:

Upselling with discount

Middle-aged customers having higher Motor values can be targeted. From *Figure 9,* we can see that middle-aged customers in Cluster 2 have a high motor value than the rest, and they have opted for the single Motor type. Therefore, we can lure them by tempting them with a discount and upsell the new Home Insurance Product.

From *Figure 8*, we know that they are located in urban areas, hence the most appropriate way to market the new product would be via email.

Promotional Offer

"You pay for 6 months, we got you covered for a year!"

All the existing customers of Insure ABC under the age of 25 can be approached with a promotional offer which will give them a year's coverage at half the price for the new Home Insurance product.

From Figure 9, the customers below age 25 lie in Cluster 1, who are located in the urban areas, thus they can be made aware of this offer via email.

Target by location

The customers living in urban areas can be targeted. The real estate values of these areas are higher than those in the rural areas, therefore, generating more revenue by charging higher premiums to the segment belonging to the urban areas.

The two segmented preferred channels 0 and 1 (refer appendix) can be classified based on location. Email represented as 0 corresponds to location 1 i.e., Urban (refer appendix). Therefore, we can deduce that people who are located in urban areas are more likely to prefer email as their preferred mode of communication. Furthermore, people who are located in the urban areas are likely to be financially well off and would be owning houses that have a significantly higher value than rural areas. Therefore, the insurance company can charge more premiums from the customer located in these areas.

Now for the people residing in the rural areas, it is observed from the above information that the phone is the preferred channel. Therefore, any advertising can be directed by the marketing department through this channel.

Cross-selling

The majority of our customers are frequent travelers and they already have travel insurance, we can launch a marketing campaign with the slogan ***"You travel, we got you covered"*** for all the customers who worry about the safety of their house while traveling. From *Figure 8*, it can be seen that for customers having Travel Type as Business (values closer to 2, refer to appendix), the preferred communication channel is Email (0, refer appendix). While for Backpacker (values closer to 1, refer appendix), those customers can be approached via Phone (1, refer appendix).

Loyalty Program

Approaching loyal customers, who have taken more than one type of Insurance from Insure ABC company by giving them a discount if they buy the new Home Insurance Product. Apart from the data-driven strategy, people who have purchased the bundle insurance packages are assumed to be loyal customers who are happy with our service, moreover, it shows the purchasing behavior of the customer which shows that the customers who are purchasing the bundle packs are risk-averse and would like to be safe in terms of any financial loss that may occur as a result of not taking insurance.

Therefore, these customers can be tempted with other bundle offers which give them a bundle or a combo package with a discounted pricing.

WORK LOG

DATE	TASKS	MINUTES OF MEETING	PARTICIPATION
02-04-2021	Introduction Meeting	Understanding the business objective, Understanding the data, Understanding the concept of clustering and formulating a set of hypotheses	All
04-04-2021	Exploratory Data Analysis	Explore the data and strategies on how to deal with the outliers and missing data and create a data quality report	Ebison, Sidharth
07-04-2021	Data Preparation	Scaling and Labelling of data, formulation of Analytics Base Table	Vaishnavi
08-04-2021	Data modelling: Customer Segmentation (Using Python)	Principal Component Analysis	Ebison
		Elbow graph	Sidharth
		K-means Clustering (Unsupervised learning)	Vaishnavi
11-04-2021	Prediction Models – Supervised learning approach (Using Python)	Selection of the most accurate predictive model	Sidharth, Vaishnavi
		Model validation using training data	Ebison
13-04-2021	Validation and evaluation	Scoring the data	Sidharth
		Descriptive statistics for the scored data	Ebison, Vaishnavi
15-04-2021	Marketing strategy	Data-Driven marketing strategy deduction	All

CONTRIBUTION

Name	Student ID	Contribution
Ebison George	20200231	33.33%
Sidharth Mohapatra	20200224	33.33%
Vaishnavi Gadekar	20200073	33.33%

APPENDIX

1. Legend of Encoded values
   After encoding the fields, the following are the labels for each categorical variable:
   a. Title:

0	1	2	3
Dr.	Mr.	Mrs.	Ms.

   b. CreditCardType:

0	1	2
Null	AMEX	VISA

   c. Location:

0	1
Rural	Urban

   d. MotorInsurance:

0	1
No	Yes

   e. MotorType:

0	1	2
Null	Bundle	Single

   f. HealthInsurance:

0	1
No	Yes

   g. HealthType:

0	1	2	3
Null	Level1	Level2	Level3

   h. TravelInsurance:

0	1
No	Yes

   i. TravelType:

0	1	2	3	4	5
Null	Backpacker	Business	Premium	Senior	Standard

   j. PrefChannel:

0	1	2
Email	Phone	SMS

2. Python Code:

```
#!/usr/bin/env python
# coding: utf-8

# ## Importing necessary packages

# In[1]:


import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
#from sklearn.impute import KNNImputer
from scipy.stats import chi2_contingency
from sklearn.svm import LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency
from sklearn import metrics
#from sklearn.feature_extraction.text import TfidfVectorizer
#tfidf = TfidfVectorizer()
import seaborn as sb
from sklearn.cluster import KMeans
import xlrd
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.decomposition import PCA
import warnings
import pickle
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from sklearn.decomposition import PCA
```

```
labelEncoder = LabelEncoder()
get_ipython().run_line_magic('matplotlib', 'inline')
scaler = MinMaxScaler()
warnings.filterwarnings('ignore')
sb.set(context="notebook", palette="Spectral", style = 'darkgrid' ,font_scale = 1.5, color_codes=True)


# ## Reading data

# In[2]:


df = pd.read_csv('QUB_Insurance_Data_Assignment_Training.csv')


# In[3]:


df1 = pd.read_csv('QUB_Insurance_Data_Assignment_Scoring.csv')


# ## Cleaning and Preprocessing of data
#
# 1. We can see that there are null values,categorical variables, other variables that provide the same data and 'PerfChannel' having 3 different categories that denote the same channels.
# 2. Therefore we impute the null values and ignore the columns with correlated data and also avoid the 'Occupation' column with string data as we can see that from the data, each occupation only covers 1-5 people and most rows are found to be None.
# 3. The other categorical variables with multiples classes can be encoded using a LabelEncoder as the categories are distinguishable and less in number.
# 4. The function module can be used for the scoring dataset as well.

# In[5]:


def preprocessing(df):
  try:
    df=df.fillna(0)
    df=df.set_index('CustomerID')
    #target=df['PrefChannel']
    df=df.drop(['GivenName','MiddleInitial','Surname','Occupation','Gender'],axis=1)
    df['Title'] = labelEncoder.fit_transform(df['Title'].astype(str))
```

```
        #df['Gender'] = labelEncoder.fit_transform(df['Gender'].astype(str))
        df['CreditCardType'] = labelEncoder.fit_transform(df['CreditCardType'].astype(str))
        df['Location'] = labelEncoder.fit_transform(df['Location'].astype(str))
        df['MotorInsurance'] = labelEncoder.fit_transform(df['MotorInsurance'].astype(str))
        df['MotorType'] = labelEncoder.fit_transform(df['MotorType'].astype(str))
        df['HealthInsurance'] = labelEncoder.fit_transform(df['HealthInsurance'].astype(str))
        df['HealthType'] = labelEncoder.fit_transform(df['HealthType'].astype(str))
        df['TravelInsurance'] = labelEncoder.fit_transform(df['TravelInsurance'].astype(str))
        df['TravelType'] = labelEncoder.fit_transform(df['TravelType'].astype(str))
        df['MotorValue'] = scaler.fit_transform(df['MotorValue'].values.reshape(-1,1))
        df['Age'] = scaler.fit_transform(df['Age'].values.reshape(-1,1))
        return df
    except Exception as e:
        print("Exception in preprocessing(): ",e)
        return None


# In[6]:


df=preprocessing(df)


# In[7]:


label=df['PrefChannel']
df=df.drop(['PrefChannel'],axis=1)


# In[8]:


df.head()


# ## Finding correlation between variables

# In[9]:


corr=df.corr()
corr.style.background_gradient(cmap='coolwarm').set_precision(2)
```

```
# 1. We can see that correlations can be mapped between features corresponding to Motor Insurance, Health Insuranc and Travel Insurance and a positive correlation between Health and Age.

# ## Clustering into categories using Unsupervised Learning
#
# 1. PCA Component Analysis: This is a dimensionality reduction technique that would find the features that are most important to distinguish categories and reduce the dimensionality of the dataset.
# 2. Elbow-Graph: This is used to identify the number of distinguishable clusters
# 3. Clustering: We use K-Means clustering.

# In[10]:


def clustering(k,data):
    try:
        kmeans = KMeans(n_clusters = k, init = 'k-means++', random_state = 42)
        model = kmeans.fit(data)
        clusters = model.predict(data)
        X = np.array(data)
        plt.figure(figsize=(15,7))
        sb.scatterplot(X[clusters == 0, 0], X[clusters == 0, 1], color = 'darkgoldenrod', label = 'Cluster 1',s=50)
        sb.scatterplot(X[clusters == 1, 0], X[clusters == 1, 1], color = 'firebrick', label = 'Cluster 2',s=50)
        sb.scatterplot(X[clusters == 2, 0], X[clusters == 2, 1], color = 'orange', label = 'Cluster 3',s=50)
        #sb.scatterplot(X[clusters == 3, 0], X[clusters == 3, 1], color = 'yellow', label = 'Cluster 4',s=50)
        print(model.cluster_centers_)
        centers = np.array(model.cluster_centers_)
        plt.scatter(centers[:,0], centers[:,1], marker="x", color='g')
        plt.grid(False)
        plt.title('Clusters')
        plt.xlabel('X')
        plt.ylabel('Y')
        plt.legend()
        plt.savefig('Clusters')
        plt.show()
        return model,clusters
    except Exception as e:
        print("Exception during KMeans clustering - ",e)
```

In[11]:

```
def PCAnalysis(data):
    try:
        pca = PCA(n_components=len(list(data.columns)))
        principalComponents = pca.fit_transform(data)
        PCA_components = pd.DataFrame(principalComponents)
        features = range(pca.n_components_)
        plt.bar(features, pca.explained_variance_ratio_, color='darkorange')
        plt.xlabel('PCA features')
        plt.ylabel('variance %')
        plt.xticks(features)
        plt.savefig("PCA")
        return PCA_components
    except Exception as e:
        print("Exception during PCA - ",e)
```

In[12]:

```
def Elbow_Graph(n,PCA_components):
    try:
        wcss = []
        for i in range(1, 11):
            kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
            kmeans.fit(PCA_components.iloc[:,:n])
            wcss.append(kmeans.inertia_)
        plt.figure(figsize=(10,5))
        sb.lineplot(range(1, 11), wcss,marker='o',color='red')
        plt.title('The Elbow Method')
        plt.xlabel('Number of clusters')
        plt.ylabel('WCSS')
        plt.savefig('ElbowGraph')
        plt.show()
        return None
    except Exception as e:
        print("Exception during plotting Elbow Graph - ",e)
```

In[14]:

```
PCA_components=PCAnalysis(df)


# 1. We can find that after the first three features, there is a dip in importance of the other categories
# 2. We run the Elbow Graph check to find distingushable clusters.

# In[15]:


Elbow_Graph(3,PCA_components)


# 1. We can find the elbow to be closer to 3 clusters.
# 2. Hence we categorise the data into 3 categories.

# clustering_model,clusters = clustering(3,PCA_components.iloc[:,:3])

# ## Writing the clustered categories and encoding target varible for predictive analysis

# In[18]:


label=label.replace('E','Email').replace('P','Phone').replace('S','SMS')


# In[19]:


#temp['Cluster']=pd.Series(clusters,index=df.index)
df['Cluster']=pd.Series(clusters,index=df.index)


# In[20]:


df['PrefChannel']=label
df['PrefChannel'] = labelEncoder.fit_transform(df['PrefChannel'].astype(str))


# In[21]:
```

```
df['PrefChannel'].value_counts()


# ## Model training and analysis using Training Data

# In[22]:


df.drop(['PrefChannel'],axis=1).apply(lambda x: x.corr(df.PrefChannel))


# In[24]:


df.drop(['Cluster'],axis=1).apply(lambda x: x.corr(df.Cluster))


# In[299]:


corr=X.corr()
corr.style.background_gradient(cmap='coolwarm').set_precision(2)


# In[424]:


df.apply(lambda x: chi2_contingency(pd.crosstab(x,df.PrefChannel)))


# In[336]:


df.columns


# In[483]:


X=df.drop([      'Age','MotorValue','MotorInsurance','MotorType',      'HealthInsurance',
'HealthType', 'HealthDependentsKids','HealthDependentsAdults', 'PrefChannel'],axis=1)
y=df['PrefChannel']
```

```
# In[484]:


X_train, X_test, y_train, y_test = train_test_split(X, y,random_state=0)


# ## Logistic Regression

# In[485]:


logreg = LogisticRegression()
LRModel=logreg.fit(X_train,y_train)
y_pred = LRModel.predict(X_test)
print("Accuracy Score:",accuracy_score(y_test, y_pred)*100)
score=precision_score(y_test,y_pred,average='macro')
print("Precision is", score*100)


# ## Decision Tree

# In[486]:


DTree = DecisionTreeClassifier()
DTModel=DTree.fit(X_train,y_train)
y_pred = DTModel.predict(X_test)
print("Accuracy Score:",accuracy_score(y_test, y_pred)*100)
score=precision_score(y_test,y_pred,average='macro')
print("Precision is", score*100)


# ## Gaussian NB

# In[487]:


GNB = GaussianNB()
GNBModel=GNB.fit(X_train,y_train)
y_pred = GNBModel.predict(X_test)
print("Accuracy Score:",accuracy_score(y_test, y_pred)*100)
score=precision_score(y_test,y_pred,average='macro')
print("Precision is", score*100)
```

```
# ## Random Forest Classifier

# In[488]:


clf = RandomForestClassifier(n_estimators = 100)
RFCModel=clf.fit(X_train,y_train)
y_pred = RFCModel.predict(X_test)
print("Accuracy Score:",accuracy_score(y_test, y_pred)*100)
score=precision_score(y_test,y_pred,average='macro')
print("Precision is", score*100)


# ## Linear SVC

# In[493]:


lsvc = LinearSVC()
LSVC=lsvc.fit(X_train, y_train)
y_pred = LSVC.predict(X_test)
accuracy=accuracy_score(y_test,y_pred)
print("Accuracy Score:",accuracy_score(y_test, y_pred)*100)
score=precision_score(y_test,y_pred,average='macro')
print("Precision is", score*100)


# ## K-Nearest Neighbours

# In[490]:


knn = KNeighborsClassifier(n_neighbors=5)
KNNModel=knn.fit(X_train,y_train)
y_pred = KNNModel.predict(X_test)
print("Accuracy Score:",accuracy_score(y_test, y_pred)*100)
score=precision_score(y_test,y_pred,average='macro')
print("Precision is", score*100)


# ## Cross Validation and Selection
```

```
# In[491]:


models = [
    RandomForestClassifier(n_estimators=200, max_depth=3, random_state=0),
    LinearSVC(),
    KNeighborsClassifier(n_neighbors=5),
    DecisionTreeClassifier(),
    GaussianNB(),
    LogisticRegression(random_state=0),
]
CV = 5
cv_df = pd.DataFrame(index=range(CV * len(models)))
entries = []
for model in models:
    model_name = model.__class__.__name__
    accuracies = cross_val_score(model, X_train, y_train, scoring='accuracy', cv=CV)
    for fold_idx, accuracy in enumerate(accuracies):
        entries.append((model_name, fold_idx, accuracy))
cv_df = pd.DataFrame(entries, columns=['model_name', 'fold_idx', 'accuracy'])


sb.boxplot(x='model_name', y='accuracy', data=cv_df)
sb.stripplot(x='model_name', y='accuracy', data=cv_df,
              size=8, jitter=True, edgecolor="gray", linewidth=2)
plt.show()


# In[492]:


cv_df.groupby('model_name').accuracy.mean()


# ## Model Selected : Linear SVC
#
# ### Classification Report and Confusion Matrix

# In[494]:


print(classification_report(y_test,y_pred))
```

In[495]:

```
cf_matrix=metrics.confusion_matrix(y_test, y_pred)
sb.heatmap(cf_matrix/np.sum(cf_matrix), annot=True,
           fmt='.2%', cmap='Blues')
```

Comparing Categories clustered to target variable

In[496]:

```
print(pd.crosstab(df.Cluster, df.PrefChannel))

fig = plt.figure(figsize=(4,4))
df.groupby('PrefChannel')['Cluster'].count().plot.bar(ylim=0)
plt.show()
```

1. We can find that the categorization doesnt prove significant enought for the prediction of channels.
2. Hence we clean the scoring data and use the features as well for predictive analysis.

Retraining the model using entire Training Data

In[497]:

```
lsvc = LinearSVC()
LSVC=lsvc.fit(X, y)
```

In[499]:

```
scoring_df=preprocessing(df1)
```

In[500]:

```
PCA2=PCAnalysis(scoring_df)
```

```
# In[501]:


scoring_df.head()


# In[502]:


result=clustering_model.predict(PCA2.iloc[:,:3])


# In[503]:


#temp1['Cluster']=result
scoring_df['Cluster']=result
#df1['Cluster']=result


# In[504]:


scoring_df=scoring_df.drop([     'Age','MotorValue',     'MotorType',     'HealthInsurance',
'HealthType',
'HealthDependentsKids','MotorInsurance','HealthDependentsAdults'],axis=1)


# In[505]:


predictions= LSVC.predict(scoring_df)


# In[513]:


scoring_df['PrefChannel']=predictions


# In[507]:
```

```
df1['PrefChannel']=predictions

# In[510]:

df1.to_csv("Results.csv",index=True)

# In[511]:

df1['PrefChannel'].value_counts()

# In[514]:

result=scoring_df.groupby(scoring_df['PrefChannel']).mean()

# In[515]:

result

# In[ ]:
```