Department of Industrial and Management Engineering,

Indian Institute of Technology, Kanpur

MBA652A – Statistical Modelling for Business Analytics

Project 2: Travel Insurance Claim Status

Guided by:

DEVLINA CHATTERJEE

PhD (IISc Bangalore)

Submitted by- (Group 03):

ASHISH UBANA (18114005)

SAURABH GUPTA (19114010)

VIVEK PRAJAPAT (19114019)

PRADNESH LACHAKE (19114007)

INDEX

ACKNOWLEDGEMENT :

We are highly indebted to Prof. Devlina Chatterjee, for her guidance and continuous support in completing this project. It is because of the knowledge and skills acquired during the course work, along with her comprehensive style of teaching, that we are able to understand the subject in a better way and are able to complete this modelling project successfully.

1. INTRODUCTION

Travel insurance is an insurance product for covering unforeseen losses incurred while travelling, either internationally or domestically. Basic policies generally only cover emergency medical expenses while overseas, while comprehensive policies typically include coverage for trip cancellation, lost luggage, flight delays, public liability, and other expenses

Cost calculation

Travel insurance is risk-based, and takes into account a range of factors to determine whether a traveller can purchase a policy and what the premium will be. This generally includes destination countries or regions, the duration of the trip, the age of the travellers, and any optional benefits that they require coverage for such as pre-existing medical conditions, adventure sports, rental vehicle excess, cruising, or high-value electronics. Some policies will also take into account the traveller's estimated value of their trip to determine price.

Insurance companies take risks over customers. Risk management is a very important aspect of the insurance industry. Insurers consider every quantifiable factor to develop profiles of high and low insurance risks. Insurers collect vast amounts of information about policyholders and analyze the data.In this project we'll have to analyze the available data and predict whether to sanction the insurance or not using different machine learning classifier models.

1.1 Objective

The goal of this project is to determine if we can predict the claim status (Yes or No) from the various travel insurance-related attributes.

1.2 Methodology

The methodology for this project is as follows:

- Importing dataset, dataset inspection and data cleaning.
- Data Exploration and Data Visualisation.
- Address imbalance in the data.
- Train Probit and Logit model to do the prediction.
- Compute the metrics for the algorithm.
- Perform various tests for possible improvements.

1.3 Economic theory

The algorithms involve detection of relations between claims, detection of the missing observations, etc. In this way, the individual customer's portfolio is made. Forecasting the upcoming claims helps to charge competitive premiums that are not too high and not too low. It also contributes to the improvement of the pricing models. This helps the insurance company to be one step ahead of its competitor.

2. DATA

2.1 Data Source

The "Travel Insurance" data is provided by a third-party travel insurance servicing company based in Singapore.

2.2 Variables

The **dependent variable** is of object type is denoted by "**Claim**" in the data set which represents whether the insurance claim would be expected or not.

There are 10 **independent variables** shown below

```
Data columns (total 11 columns):
Agency                  63326 non-null object
Agency Type             63326 non-null object
Distribution Channel    63326 non-null object
Product Name            63326 non-null object
Claim                   63326 non-null object
Duration                63326 non-null int64
Destination             63326 non-null object
Net Sales               63326 non-null float64
Commision (in value)    63326 non-null float64
Gender                  18219 non-null object
Age                     63326 non-null int64
dtypes: float64(2), int64(2), object(7)
```

3. DESCRIPTIVE ANALYSIS

3.1 Statistical summary of data

	Duration	Net Sales	Commision (in value)	Age
count	63326.000000	63326.000000	63326.000000	63326.000000
mean	49.317074	40.702018	9.809992	39.969981
std	101.791566	48.845637	19.804388	14.017010
min	-2.000000	-389.000000	0.000000	0.000000
25%	9.000000	18.000000	0.000000	35.000000
50%	22.000000	26.530000	0.000000	36.000000
75%	53.000000	48.000000	11.550000	43.000000
max	4881.000000	810.000000	283.500000	118.000000

1. **Age:**
    - minimum: 0 which is possible.
    - maximum: 118, Way too old to travel.

Let's check what insurance companies say.

Insurance Companies do not give insurance over an age of 70. However there are few who provide insurance upto age of 99. So giving benefit of doubt to our Safe Travel Insurance travel company we will assume that any individual upto age 99 is valid anything above that needs to be treated.

Since the percentage of people whose age is more than 99 is greater than 0.05 we will not be deleting these data, we will instead compute age above 118 as 99 we will not be computing this with median or mean since that would manipulate the data. Thus giving the benefit of doubt to the data we will make it 99.

Also rather than dealing age as continuous variable we can convert it into groups so that it would be easier for us to classify which age group people are more likely to claim for insurance.

Function to make age groups as children, adults and senior is as follows

If age <= 21 : Child

Else age <= 50 : Adult

Else : Senior

2. **Commission:** The data for commission looks valid.
3. **Duration:**
    - minimum: -2
    - maximum: 4881

so now lets see what insurance companies say.

- Duration can never be -ve. so this definitely has to be computed.

- Duration of 4881.

Insurance companies do not give insurance over 180 days in India. Let's assume that there is an Annual plan in place then we can say that the maximum tenure of an insurance plan would be 365 days. Also given a fact that one can book a ticket 1 year in advance so we will say that the maximum duration of the insurance cannot be more than

- 731 days(365 + 366 days considering if it is a leap year)

In case the values are less than 0.05 we will go ahead and drop the data points else we will have to treat them.

Again the percentage of data showing duration less than 1 day and greater than 731 days are more than 5% so we will replace duration less than 1 day by mean and greater than 731 by 731 itself only.

4. **Net Sales:** Minimum has negative values. To check whether sales can be negative or not let's understand how Net Sales is computed.
   Net Sales = (The Value for which the insurance is sold - Any expenses incurred/Claim Amount paid)

   So it is possible that the Sales is showing negative if the Claim amount is paid, it can also be negative even if the claim amount is not paid incases where the insurance was sent for claim and got rejected. Now the expenses incurred for doing investigation of that claim would be more than the actual policy amount paid.

4. DATA EXPLORATION

	Agency	Agency Type	Distribution Channel	Product Name	Claim	Duration	Destination	Net Sales	Commision (in value)	Gender	Age
0	CBH	Travel Agency	Offline	Comprehensive Plan	No	186	MALAYSIA	-29.0	9.57	F	81
1	CBH	Travel Agency	Offline	Comprehensive Plan	No	186	MALAYSIA	-29.0	9.57	F	71
2	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	65	AUSTRALIA	-49.5	29.70	NaN	32
3	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	60	AUSTRALIA	-39.6	23.76	NaN	32
4	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	79	ITALY	-19.8	11.88	NaN	41

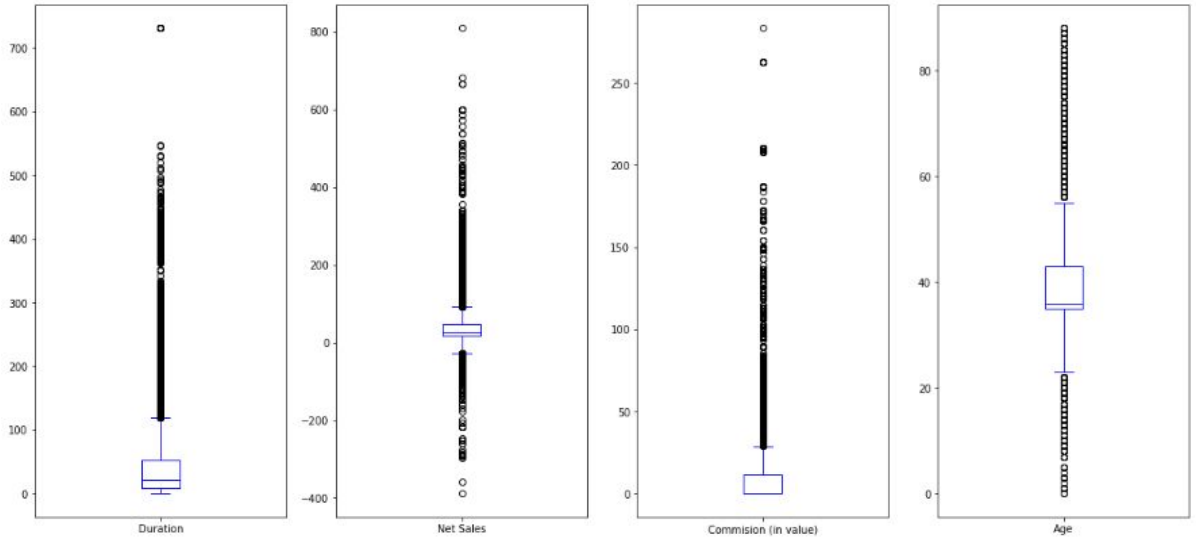
4.1 Checking null values

```
Agency                      0
Agency Type                 0
Distribution Channel        0
Product Name                0
Claim                       0
Duration                    0
Destination                 0
Net Sales                   0
Commision (in value)        0
Gender                  45107
Age                         0
dtype: int64
```

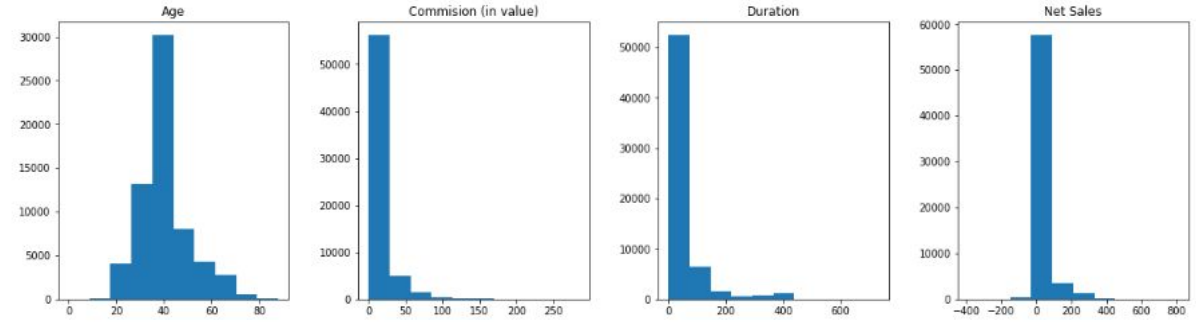
As clearly seen that most of the values in the gender column are null And we also understand that there are no other features that will help us find out whether the individual is male or female. Thus we will drop this feature.

5. Exploratory Data Analysis

5.1 Boxplot



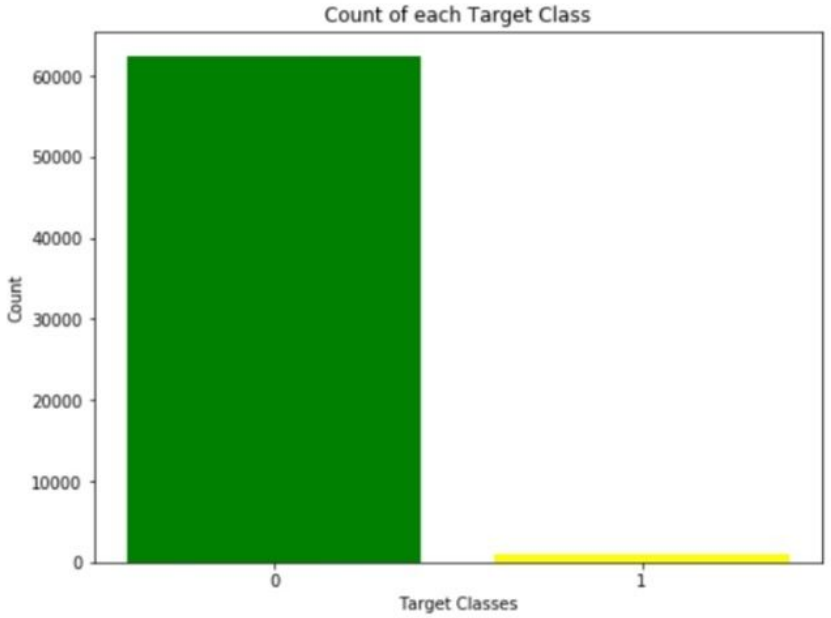
5.2 Histogram



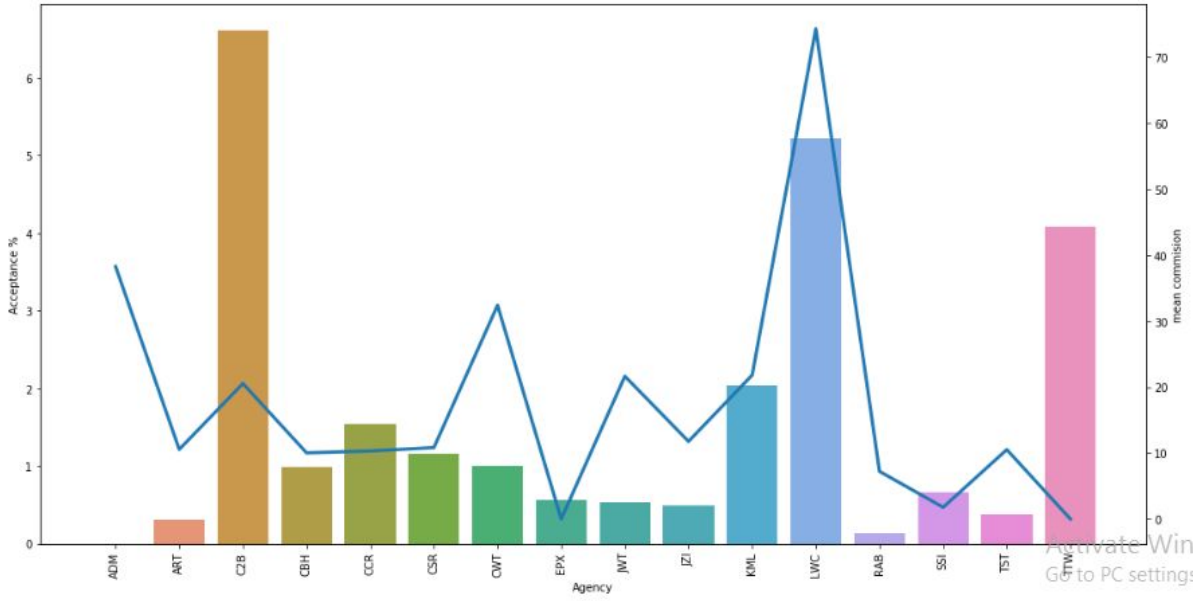
5.3 Checking the distribution of the claims:

Claim	
0	0.9854
1	0.0146

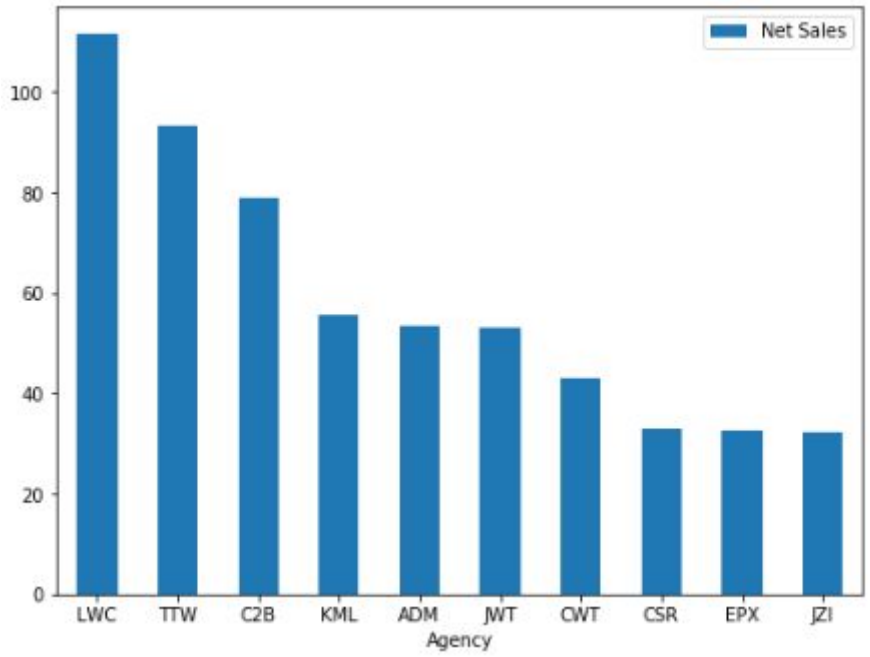
As it can be clearly seen that it's a case of class imbalance problem where the number of observations belonging to one class is significantly lower than those belonging to the other classes.



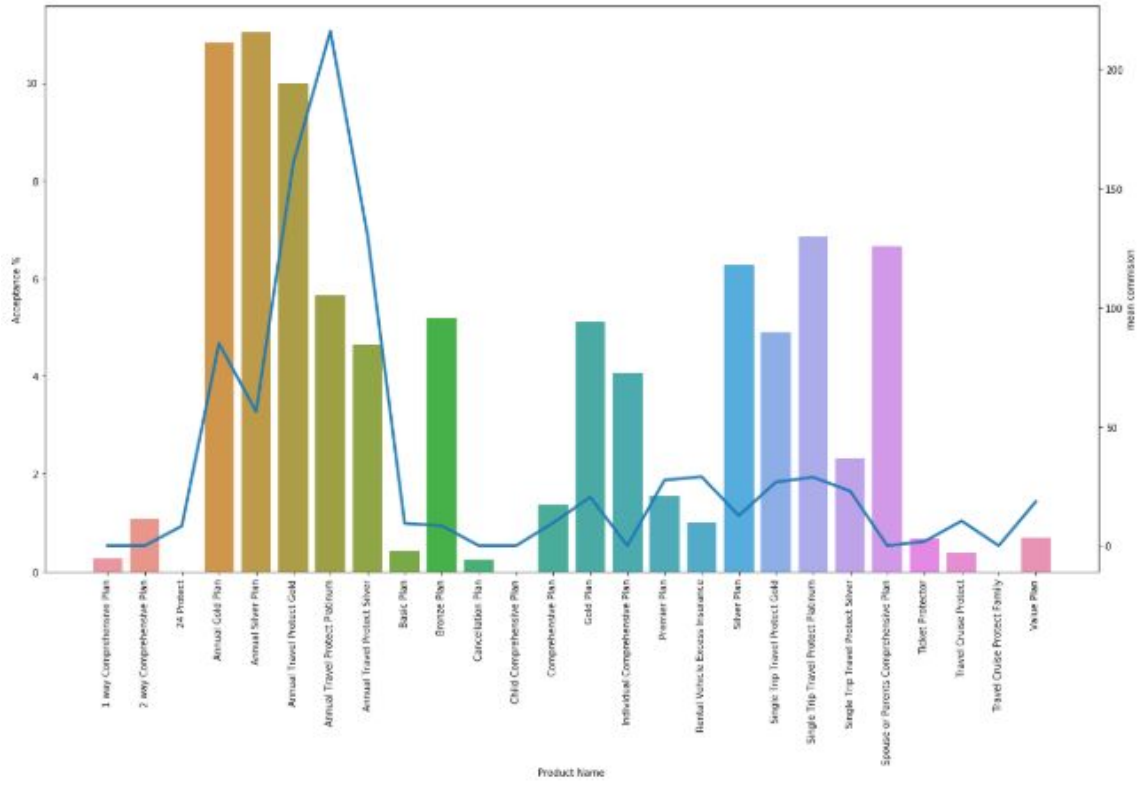
5.4 Plotting the Agencies with the Maximum Claims



5.5 Plotting the Agencies with the Maximum Net Sales



5.6 Plotting the Product Name with the % Acceptance



6. MODELS

6.1 Probit Model

A probit model (also called probit regression), is a way to perform regression for binary outcome variables. Binary outcome variables are dependent variables with two possibilities, like yes/no[3]. The word “probit” is a combination of the words probability and unit; the probit model estimates the probability a value will fall into one of the two possible binary (i.e. unit) outcomes. It uses the inverse standard normal distribution as a linear combination of the predictors. The binary outcome variable Y is assumed to have a Bernoulli distribution with parameter p (where the success probability is $p \in (0,1)$).It is estimated using Maximum Likelihood Estimator (MLE).

We will be using the “statsmodel” package for Probit regression. First we break our dataset into dependent variable and independent variables. We will use "Claim" as our response variable and all the remaining variables as predictors. Then we use the statsmodels function to fit our Probit regression with our response variable and design matrix. The statsmodels package is unique from other languages and packages as it does not include an intercept term by default. We will set constant value manually.

The statsmodels package automatically includes p values and confidence intervals for each coefficient. For those that are familiar with objects, the probit model is stored as a probit model object in Python.

6.2 Logit Model

The logit model uses something called the cumulative distribution function of the logistic distribution. Similar to the Probit model this function will take any number and rescale it to fall between 0 and 1.

The logistic regression uses a function called the sigmoid function, it’s an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

1 / (1 + e^-value)

Logistic regression equation:

*y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))*

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.

*p(X) = e^(b0 + b1*X) / (1 + e^(b0 + b1*X))*

The coefficients (Beta values b) of the logistic regression algorithm must be estimated from your training data. This is done using maximum-likelihood estimation.

6.3 ROC(Receiver Operating Characteristics):

In order to understand the performance of a model with a binary classification problem, we can use the ROC curve to evaluate the performance. Using test data which is splitted from the given data set, a confusion matrix is created for various values of threshold. The true positive rate (TPR) and false positive rate (FPR) from the confusion matrix is then used to plot the ROC curve.

The TPR defines how many correct positive results occur among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test.

An ROC space is defined by FPR and TPR as x and y axes, respectively, which depicts relative trade-offs between true positive (benefits) and false positive (costs). Since TPR is equivalent to sensitivity and FPR is equal to 1 − specificity, the ROC graph is sometimes called the sensitivity vs (1 − specificity) plot. Each prediction result or instance of a confusion matrix represents one point in the ROC space.

6.4 Multicollinearity

In our analysis, we are using VIF (Variance Inflation Factor) to check the multicollinearity, which assesses how much the variance of an estimated regression coefficient increases if your predictors are correlated. Following table shows the VIF of all the independent variables.

	Features	VIF
2	Distribution Channel	9.64
0	Agency	8.08
5	Net Sales	5.45
3	Product Name	4.00

1	Agency Type	3.97
6	Commision (in value)	3.51
4	Duration	3.06
7	Age Group	1.25

VIF value with 1 means that there is no correlation among the predictor and the remaining predictor variables, and hence the variance is not inflated at all. VIFs exceeding 10 are signs of serious multicollinearity requiring correction. Considering VIF threshold value as 10, we have taken every independent variable to train the model.

To check the multicollinearity, a heatmap of correlation matrix for the predictors variables has been plotted. The following diagram shows the correlation matrix.

We fitted regression models between a dependent variable and independent variables. We formulated 4 different models to understand the relationship of Claim with other factors like 'Agency', 'AgencyType', 'Distribution Channel', 'ProductName', 'Duration','Net Sales', 'Commision (in value)', 'Age Group Etc. The list of models formulated in this study are listed below:

Model 1: Probit Regression

Here we are taking all the independent variables, i.e. 'Agency', 'AgencyType', 'Distribution Channel', 'ProductName', 'Duration','Net Sales', 'Commision (in value)', 'Age Group' to train the probit model. The summary of probit results are as follows:

$$Z = -0.755 * \text{'DistributionChannel'} - 0.4738 * \text{'AgencyType} - 0.1131 * \text{'Agency'} - 0.0802 \text{'Age Group'} + 0.0039 * \text{'Net Sales'} + 0.0033 * \text{'Commision (in value)'} - 0.0011 * \text{'ProductName'} - 0.0004 \text{'Duration'} + 0.4678 \text{ (const)}$$

$$Prob(Claim = 1 / X) = \Phi(Z)$$

Probit Regression Results

Dep. Variable:	Claim	**No. Observations:**	74399
Model:	Probit	**Df Residuals:**	74390
Method:	MLE	**Df Model:**	8
Date:	Sat, 13 Jun 2020	**Pseudo R-squ.:**	0.1850
Time:	01:00:06	**Log-Likelihood:**	-26790.
converged:	True	**LL-Null:**	-32870.
Covariance Type:	nonrobust	**LLR p-value:**	0.000
Dep. Variable:	Claim	**No. Observations:**	74399

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	0.4678	0.054	8.679	0.000	0.362	0.573
Agency	-0.1131	0.003	-43.228	0.000	-0.118	-0.108
Agency Type	-0.4738	0.016	-29.910	0.000	-0.505	-0.443

Distribution Channel	-0.7550	0.046	-16.396	0.000	-0.845	-0.665
Product Name	-0.0011	0.001	-0.939	0.348	-0.003	0.001
Duration	-0.0004	9.76e-05	-4.213	0.000	-0.001	-0.000
Net Sales	0.0039	0.000	21.030	0.000	0.004	0.004
Commision (in value)	0.0033	0.000	7.849	0.000	0.002	0.004
Age Group	-0.0802	0.009	-9.021	0.000	-0.098	-0.063

Model 2: Logit Regression

The second model is trained on logit regression using all the independent variables. The summary of the results are as follows:

Z = - 1.293 'DistributionChannel' - 0.8496* 'AgencyType - 0.2082* 'Agency' -0.1415* 'Age Group' + 0.0067* 'Net Sales' + 0.007* 'Commision (in value)' - 0.0039* 'ProductName' - 0.0013 'Duration' + 0.9276 (const)*

Prob(Claim = 1 / X) = F (Z)

F(z) = 1 / (1 + e^-z)

Logit Regression Results

Dep. Variable:	Claim	**No. Observations:**	74399
Model:	Logit	**Df Residuals:**	74390
Method:	MLE	**Df Model:**	8
Date:	Sat, 13 Jun 2020	**Pseudo R-squ.:**	0.1851
Time:	01:00:07	**Log-Likelihood:**	-26786.
converged:	True	**LL-Null:**	-32870.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.9276	0.097	9.541	0.000	0.737	1.118
Agency	-0.2082	0.005	-41.885	0.000	-0.218	-0.198
Agency Type	-0.8496	0.030	-28.393	0.000	-0.908	-0.791
Distribution Channel	-1.2930	0.085	-15.164	0.000	-1.460	-1.126
Product Name	-0.0039	0.002	-1.731	0.083	-0.008	0.001
Duration	-0.0013	0.000	-6.837	0.000	-0.002	-0.001
Net Sales	0.0067	0.000	19.025	0.000	0.006	0.007
Commision (in value)	0.0070	0.001	9.077	0.000	0.005	0.009
Age Group	-0.1415	0.016	-8.719	0.000	-0.173	-0.110

The confidence interval after t-test for the independent variable should not contain zero to reject the null hypothesis. But from both of the above models,i.e. Probit and logit independent variable ‘Product Name’ contains zero in the t test interval of [0.025] to [0.975], so we can’t reject the null hypothesis. It’s p value is also higher than the significance level.

Based on the above facts we have concluded that the independent variable “Product Name” is not statistically significant. So, we have decided to drop this variable and model the data again.

Model3: Probit Regression

After dropping 'Product Name' from the data set we have trained the probit regression model. The summary of the model is as follows:

Z = - 0.7483 'DistributionChannel'- 0.469* 'AgencyType - 0.1134* 'Agency' -0.0808*'Age Group' + 0.0039* 'Net Sales' + 0.0031* 'Commision (in value)' -0.0004 'Duration' + 0.4482 (const)*

Prob(Claim = 1 / X) = Φ (Z)

Probit Regression Results

Dep. Variable:	Claim	**No. Observations:**	74399
Model:	Probit	**Df Residuals:**	74391
Method:	MLE	**Df Model:**	7
Date:	Sat, 13 Jun 2020	**Pseudo R-squ.:**	0.1850
Time:	01:00:07	**Log-Likelihood:**	-26790.
converged:	True	**LL-Null:**	-32870.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	0.4482	0.050	9.027	0.000	0.351	0.545
Agency	-0.1134	0.003	-43.573	0.000	-0.118	-0.108
Agency Type	-0.4690	0.015	-31.292	0.000	-0.498	-0.440
Distribution Channel	-0.7483	0.045	-16.462	0.000	-0.837	-0.659
Duration	-0.0004	9.38e-05	-4.115	0.000	-0.001	-0.000
Net Sales	0.0039	0.000	22.030	0.000	0.004	0.004

Commision (in value)	0.0031	0.000	8.360	0.000	0.002	0.004
Age Group	-0.0808	0.009	-9.118	0.000	-0.098	-0.063

Model 4: Logit Regression

The results of the logit regression model is as follows

Z = - 1.2745* 'DistributionChannel' - 0.8346* 'AgencyType - 0.2086* 'Agency' -0.1441* 'Age Group' + 0.0068* 'Net Sales' + 0.0064* 'Commision (in value)' - 0.0012 'Duration' + 0.8626 (const)

Prob(Claim = 1 / X) = F (Z)

F(z) = 1 / (1 + e^-z)

Logit Regression Result

Dep. Variable:	Claim	**No. Observations:**	74399
Model:	Logit	**Df Residuals:**	74391
Method:	MLE	**Df Model:**	7
Date:	Sat, 13 Jun 2020	**Pseudo R-squ.:**	0.1850
Time:	01:00:08	**Log-Likelihood:**	-26788.
converged:	True	**LL-Null:**	-32870.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	0.8626	0.090	9.627	0.000	0.687	1.038
Agency	-0.2086	0.005	-42.122	0.000	-0.218	-0.199
Agency Type	-0.8346	0.029	-29.173	0.000	-0.891	-0.779

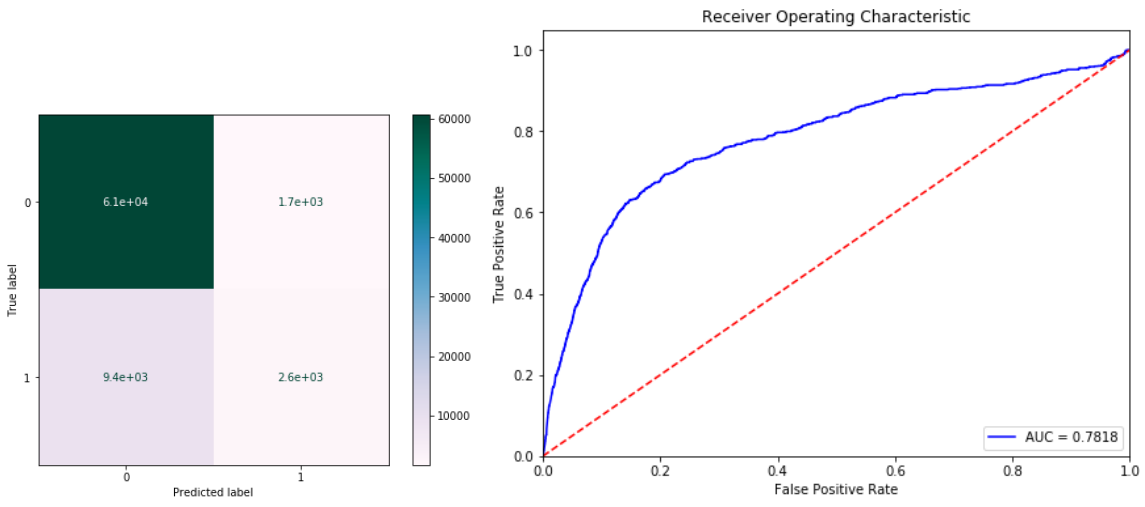
Distribution Channel	-1.2745	0.084	-15.088	0.000	-1.440	-1.109
Duration	-0.0012	0.000	-6.632	0.000	-0.002	-0.001
Net Sales	0.0068	0.000	19.853	0.000	0.006	0.008
Commision (in value)	0.0064	0.001	9.248	0.000	0.005	0.008
Age Group	-0.1441	0.016	-8.922	0.000	-0.176	-0.112

Now p values of all the independent variables are under significant level. So we have concluded that all the variables are statistically significant.

Confusion matrix and ROC Curve:

A Model comparison using confusion matrix and ROC curve has been done for Model 2 and Model 4.

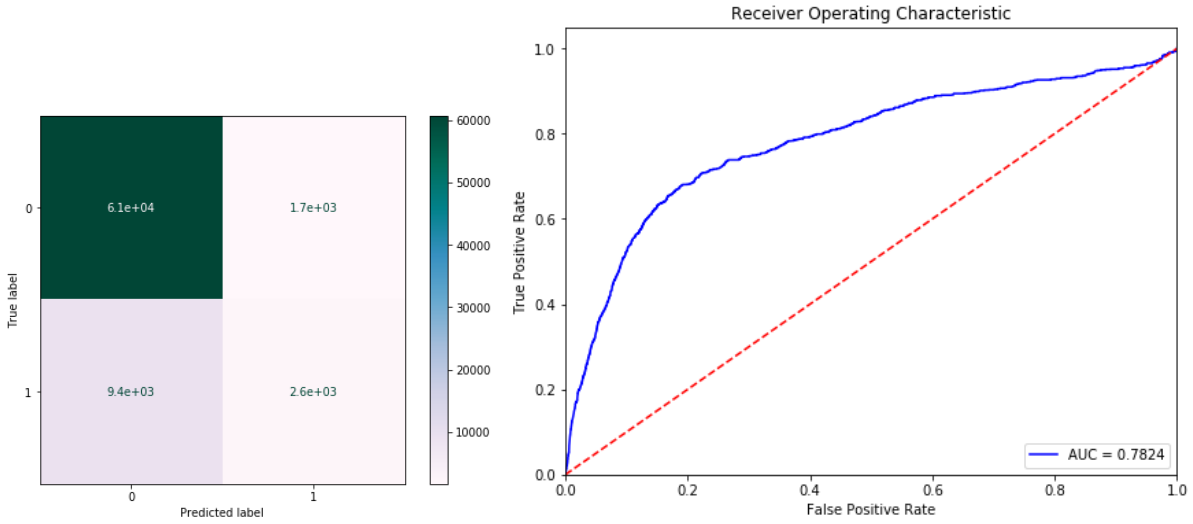
'Agency', 'AgencyType', 'DistributionChannel','ProductName', 'Duration','Net Sales', 'Commision (in value)', 'Age Group'



AUC(Area under Curve) = 0.7818

F1 score = 0.8198

'Agency', 'AgencyType', 'DistributionChannel', 'Duration','Net Sales', 'Commision (in value)', 'Age Group'



AUC(Area under Curve) = 0.7824

F1 score = 0.82

We can compare from the above ROC plots, AUC has been slightly increased from 0.7818 to 0.7824 after dropping the statistically insignificant variable '*Product Name'*.

6.5 CONCLUSIONS

- From the exploratory data analysis we have found that our data is highly imbalanced, So we have used F1 score to check the model performance.
- Both the classification models Probit and Logit have almost the same performance.
- We have found 'Product names' is statistically insignificant.
- From the final model following are the features listed in their decreasing order of their significance
    - 'DistributionChannel'
    - 'AgencyType'
    - 'Agency'
    - 'Age Group'
    - 'Net Sales'
    - 'Commision (in value)"Duration'

Omitted Variable Bias

- Medical History
- Purpose of Travelling (Adventure Sports, etc)
- Any formal proofs of stolen personal belonging.
- Consumption of Alcohol and any other stuff.

7 REFERENCES:

1. Probit Regression in R, Python

2. Travel insurance (From Wikipedia, the free encyclopedia (https://en.wikipedia.org/wiki/Travel_insurance)

3. statsmodels.discrete.discrete_model.Probit (link)

4. Probit Model (Probit Regression): (https://www.statisticshowto.com/probit-model/)

8. Python Code

```
#import libraries
import pandas as pd
import numpy as np

#importing libraries for graphical representation
import matplotlib.pyplot as plt
import seaborn as sns
from pandas.plotting import scatter_matrix
import matplotlib.patches as mpatches
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline

#import data
travel=pd.read_csv("travel insurance.csv")
travel.info()

#  function for data preprocessing
def data_pre_processing(df):
    # Add a new column in the Database as Age Group

    df['Age Group'] = df['Age'].map(lambda x: age_convert(x))

    # Dropping Feature Gender
    df.drop('Gender',axis =1, inplace=True)
   # Since the minimum duration that any travel can have is 1 day thus we impute it by the
column median.
    df['Duration'][df['Duration'] < 0] = df['Duration'].median()

    # As we observed duration of any travel cannot be more than 731 we will impute it as 731.
    df['Duration'][df['Duration'] > 731] = 731

    # replacing the values that is greater than 99 with the mean of Senior Age
    df['Age'][df['Age'] > 99] = df[df['Age Group'] == 'Senior']['Age'].mean()
X = dataset_oversampled[['Agency','Agency Type','Distribution Channel','Product
 Name','Duration','Net Sales','Commision (in value)','Age Group']]
Y = dataset_oversampled["Claim"]
```

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# Create a dataframe that will contain the names of all the feature variables and their
 respective VIFs
vif = pd.DataFrame()
vif['Features'] = X.columns
vif['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
import statsmodels.api as sm
X = dataset_oversampled[['Agency','Agency Type','Distribution Channel','Product
Name','Duration','Net Sales','Commision (in value)','Age Group']]
Y = dataset_oversampled["Claim"]
X = sm.add_constant(X)

Probit = sm.Probit(endog=Y, exog = X)

result = Probit.fit()
# print(Probit.cdf(X))
result.summary()
X = dataset_oversampled[['Agency','Agency Type','Distribution Channel','Product
Name','Duration','Net Sales','Commision (in value)','Age Group']]
Y = dataset_oversampled["Claim"]
X = sm.add_constant(X)

logit = sm.Logit(endog=Y, exog = X)

result = logit.fit()

result.summary()




from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X, Y)

y_pred = classifier.predict(X)

from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(classifier, X, Y , cmap='PuBuGn')  # doctest: +SKIP
plt.show()
```

```
#ROC curve code
import sklearn.metrics as metrics

probs = classifier.predict_proba(X)
preds = probs[:,1]

fpr, tpr, threshold = metrics.roc_curve(Y, preds)
roc_auc = metrics.auc(fpr, tpr)


import matplotlib.pyplot as plt
plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b', label = 'AUC = %0.4f' % roc_auc)
plt.legend(loc = 'lower right')

plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0, 1])
plt.ylim([0.0, 1.05])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```