mosely

Mini Project (IS65)

6th Semester

Kushagra Saxena 1MS18IS045

Under the guidance of

Dr.Vijaya Kumar B P

Professor, Dept. of ISE, RIT

RAMAIAH
Institute of Technology

RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)
MSR NAGAR, MSRIT POST, BANGALORE-560054

Acknowledgement

I have taken utmost efforts to successfully complete this project. However, it would not have been possible without the kind support and help of many individuals and the organization in general. I would like to extend my sincere thanks to all of them. I am grateful to our Institution, Ramaiah Institute of Technology with its ideals and inspiration for having provided me with the facilities, which has made this project a success.

I acknowledge the guidance and support extended by my guide, Dr.Vijaya Kumar B P, Professor, Department of ISE. His incessant encouragement and invaluable technical support has been of immense help. His guidance gave me the environment to enhance my knowledge and skills.

I take this opportunity to express my profound gratitude and deep regards to the Head of the Department and Professor Dr. Yogish H K. for his valuable information, exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him from time to time shall carry us a long way in the journey of life on which we are about to embark.

I would like to extend my hearty gratitude to Dr. N V R Naidu, The Principal, for the kind support and permission to use the facilities available in the Institute. I am obliged to all the teaching and non-teaching staff members of the ISE department, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of our project.

Kushagra Saxena 1MS18IS045

Abstract

Ever since the pandemic hit us, we are moving more and more towards a digital world. Almost every industry has had to move their day to day work online in some way or form. Educational institutes were no different. All the classes, tests, assignments had to move online to ensure continuation of a student's education during the pandemic.

However, this gave rise to the problem where student's would cheat on the tests and assignments given to them as teachers have had no way of proctoring them while they are at home. This has led to the students lacking knowledge as the task was not performed by them. Handing over plagiarised assignments is the main issue we want to eliminate.

By adhering strictly to a no-plagiarism-accepted policy as an institute, we will not only be able to make the evaluation procedure more robust and clear but also motivate the students to be self-sufficient. This will lead to clarity of the subject matter involved.

Table of Contents

1 Overview

An audit company is evaluating the cases where Insurance has been claimed by Agencies for various Products under Travel Insurance across Geographies. Using this data, it wants to build a predictive model which can identify beforehand whether Insurance will be claimed by such Agencies under the various scenarios. Utilizing model, the company also aims to highlight False claims and built an automated guidance tool for its Stakeholders for the reasons under which Claims are approved or rejected The Company is particularly interested in higher Recall. Also, Reason for Claims approval or rejection needs to be provided

1.1 Approach to the problem

The below flowchart demonstrates the flow of the project

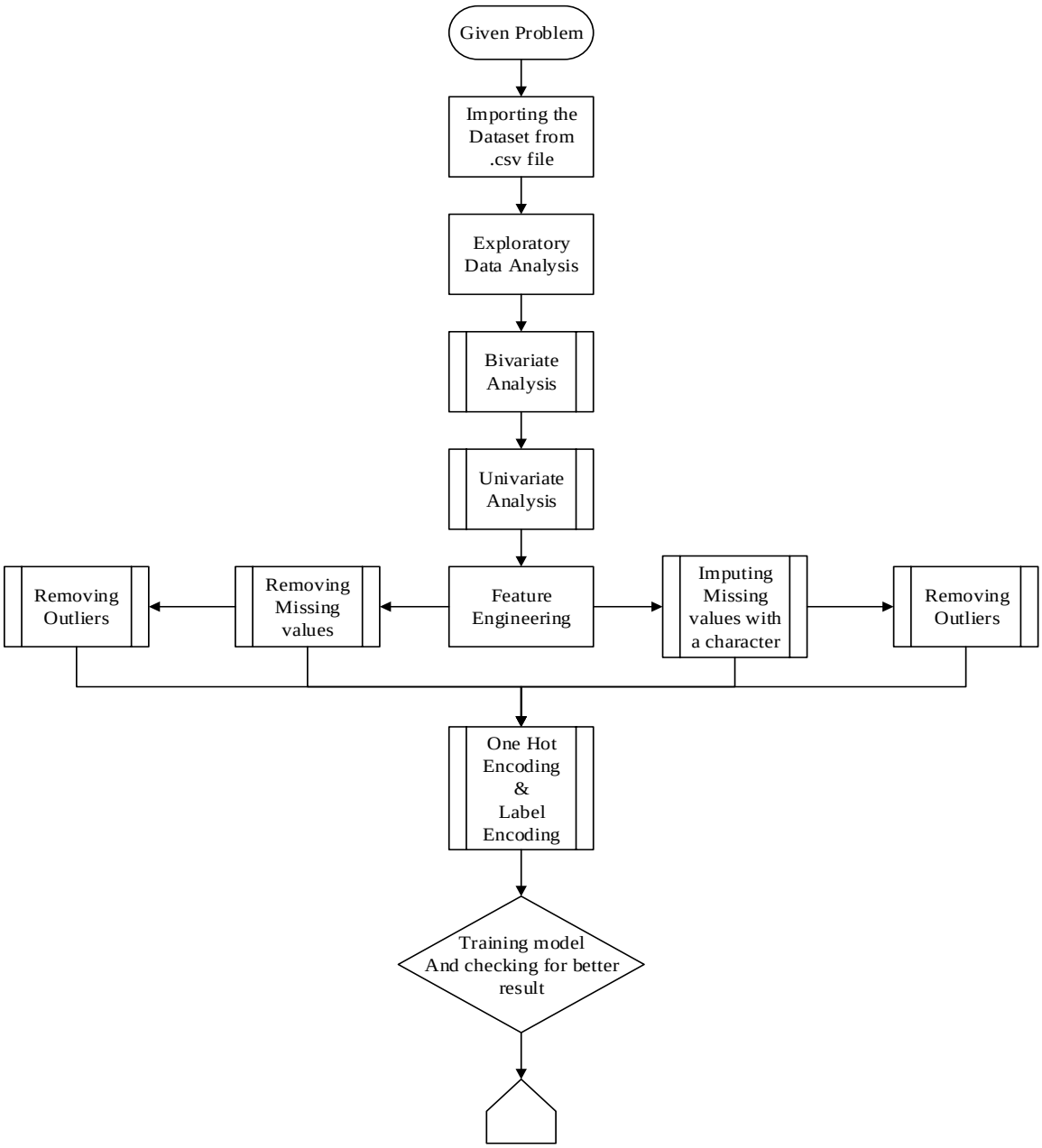

Figure 1- Flow Chart

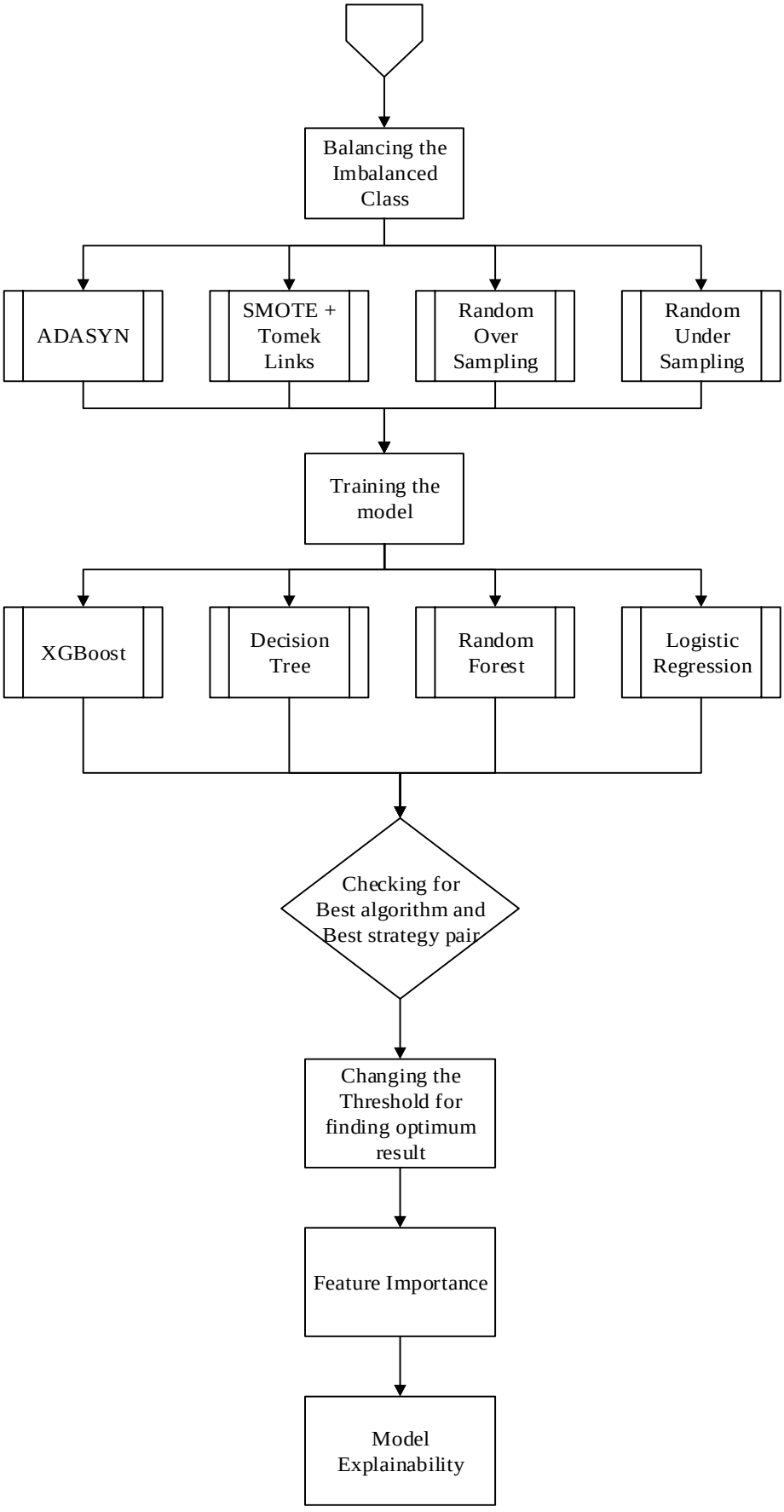

Figure 2 – Flow Chart

2 Technology Stack

Google Colab	Jupyter Notebook	Microsoft Word	Microsoft Excel
Microsoft Visio	Sci-kit learn	Imbalanced-learn	Pandas
Seaborn	Matplotlib	Numpy	GitHub

Table 1- Technology Stack Used in the project

3 Features Description

When given a dataset, here a labelled one with Claim being the Target Label/Dependent Label, rest of the fields become potential Independent Labels or Features.

From the "potential" list of features, it is first checked if any Primary Key record identification fields are present which were created only for Identification purposes. Such fields are removed from Features list as they have no usage in predicting Target Label.

Since, there is no such primary key in the dataset, so, the features in the dataset used in the project are:

Column Name	***dtype***	***Description***
Agency	object	contains the name of agency
Agency Type	object	contains the type of travel insurance agencies
Distribution channel	object	contains the distribution channel of travel insurance agencies
Product Name	object	contains the travel insurance products
Claim	object	contains whether insurance claim has been approved or not
Duration	int64	contains the duration of travel
Destination	object	contains the destination of travel
Net Sales	float64	contains the amount of sales of travel insurance policies
Commission(in value)	float64	is the commission received for travel insurance agency
Gender	object	is the gender of the insured person
Age	int64	is the age of the insured person

Table 2 - Features Description

In a dataset, there are two types of variables: categorical and continuous.

- *Categorical Variable*:
  In a categorical variable, the value is limited and usually based on a particular finite group.
- *Continuous Variable*:
  A continuous variable, however, can take any values, from integer to decimal.

In the dataset, the categorical variables are:

'Agency', 'Destination', 'Distribution Channel', 'Agency Type', 'Claim', 'Product Name', 'Gender', 'Duration', 'Age'

In the dataset, the continuous variables are:

'Net Sales', 'Commission (in value)'

- Traditionally *'Duration'* and *'Age'* should be continuous variables because
    - The skewness of *'Duration'* was coming as 23.17 which was not possible for a continuous variable.
    - *'Age'* was so less in number so it can't be a continuous variable.

Target Variable: The target variable of a dataset is a feature of a dataset which you want to gain a deeper understanding on. Here, target variable is 'Claim'

4 Exploratory Analysis on Features

4.1 Univariate Analysis

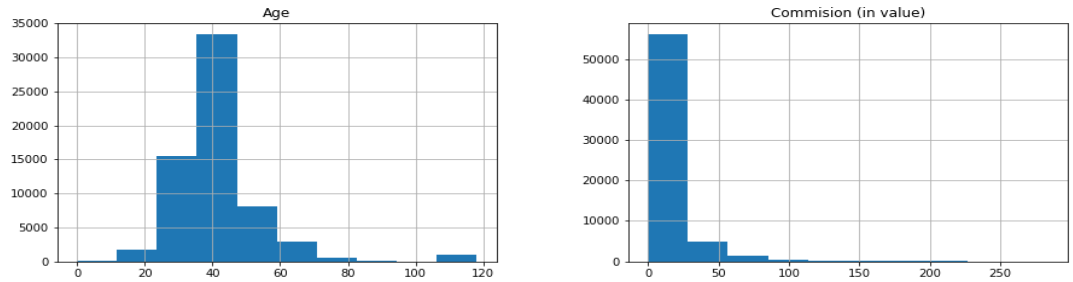

Figure 3 - Histogram Plot (Age and Commision in value)

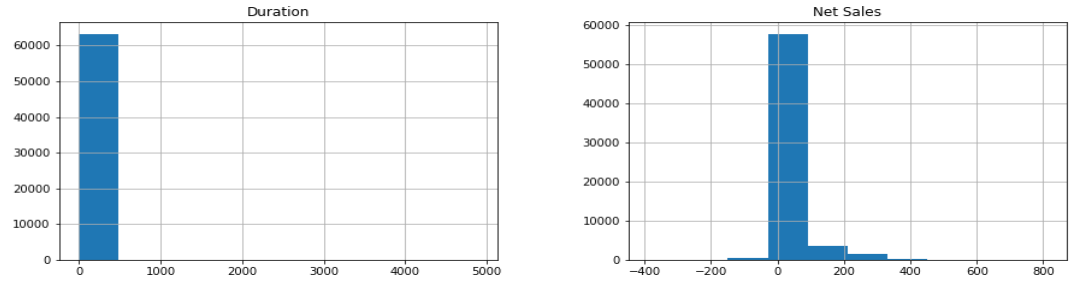

Figure 4- Histogram Plot (Duration and Net Sales)

From the Figure 3 and Figure 4, it can be inferred that:

- The majority of people i.e. about 32500 of agency are of the age between 37 and 45, 16000 people are of age between 23 and 37, 8000 people are between the age group of 45-59, 3000 between 59-70, 2000 between 10-23 and rest between 70 and 120 leaving the gap of 94-114.
- Almost 57000 people agencies have the commission of 0-25, and 5000 agencies have the commission of 25-55.
- The duration of travel of a person is lying between 1 and 450 for all the cases.
- In about 59000 cases, the net sales of the agencies lies between -170 to 80, when 80-210 for 3000 cases.

From the Figure 5 and Figure 6 below:

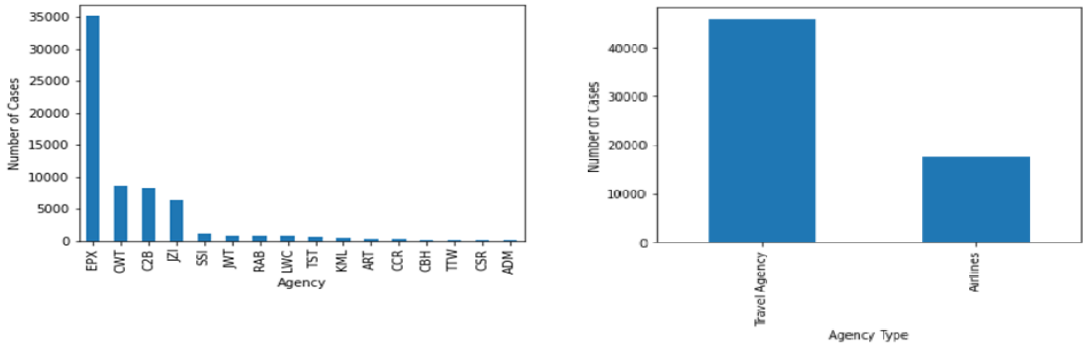

Figure 5- Graph showing 'Agency' and 'Agency Type' distribution

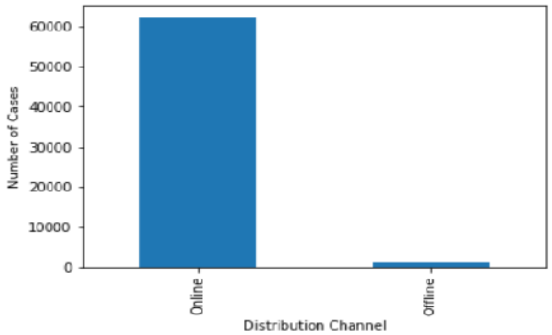

Figure 6- Graph showing the 'Distribution Channel' Distribution

It can be clearly concluded that, Agency EPX claims mostly, Travel Agencies apply more claims i.e. 70% while Airlines agencies apply less claims and almost all the cases except for very few have online distribution channel.

Figure 7 shows that the Cancellation plans are the more frequently used plan which is understandable as if the trip got cancelled, they'll hope for a refund.

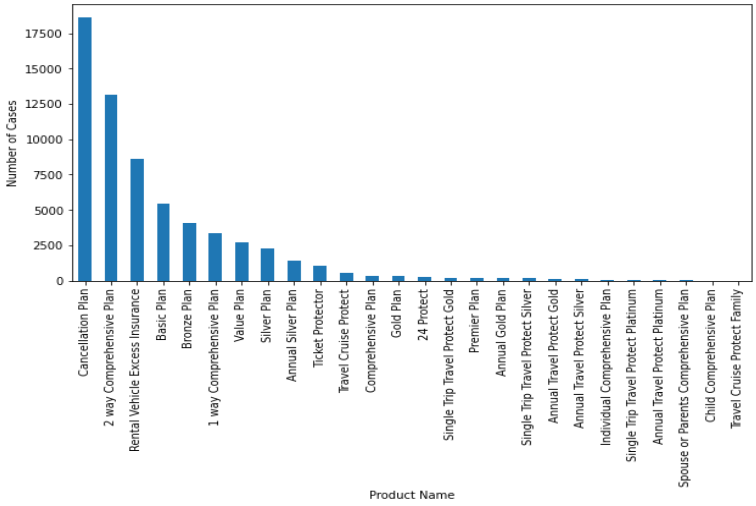

Figure 7- Graph showing the distribution of 'Product Name'

4.2 Bivariate Analysis

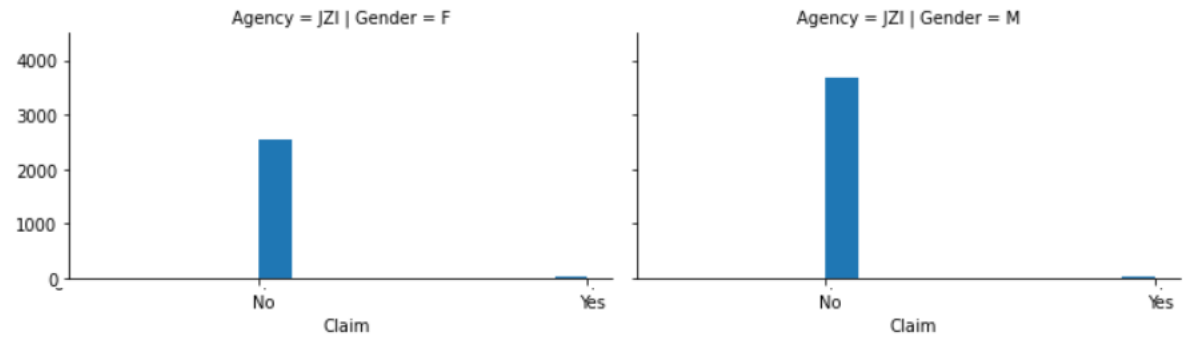

Figure 8- Facet Grid plot between 'Agency' and 'Gender'

From the Figure 8, it can be derived that, irrespective of the Gender, when the 'Agency' is 'JZI', the Claim is 'No'. Though the number of cases is more in case of gender is 'Male'.

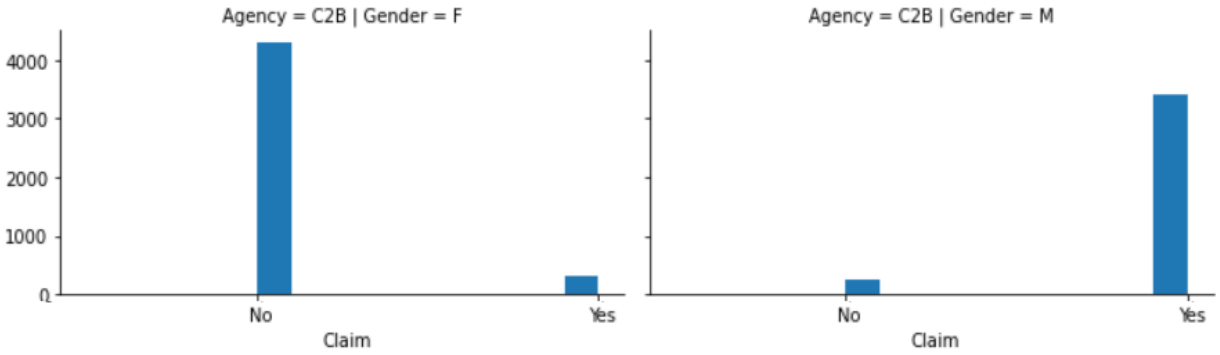

Figure 9- Facet Grid plot between 'Agency' and 'Gender'

Figure 9 shows, when the Agency is 'C2B', cases of Claim being 'No' is more when the applicant is Female, and cases of Claim being 'Yes' is more when Gender Male.

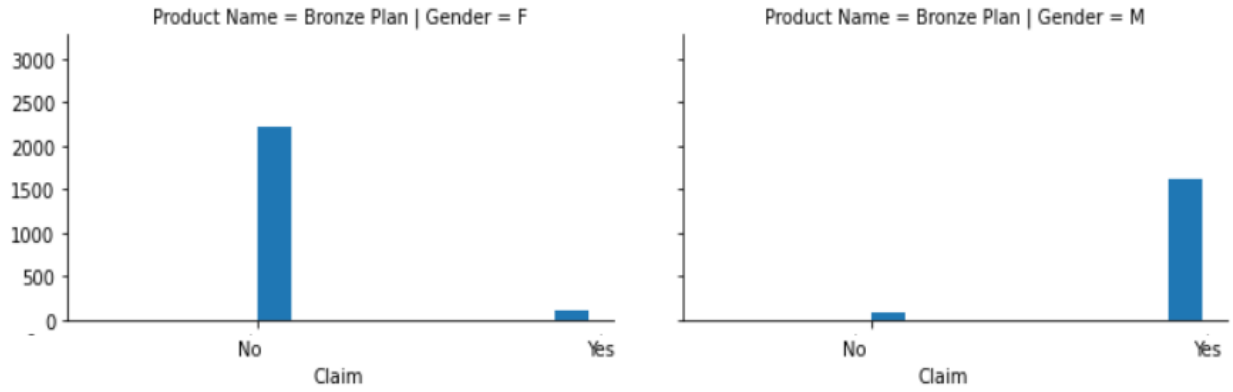

Figure 10 Facet Grid plot between 'Product Name' and 'Gender'

From the Figure 10, when the Product Name is Bronze, cases of Claim being 'No' is more when the applicant is Female, and cases of Claim being 'Yes' is more when gender is Male.

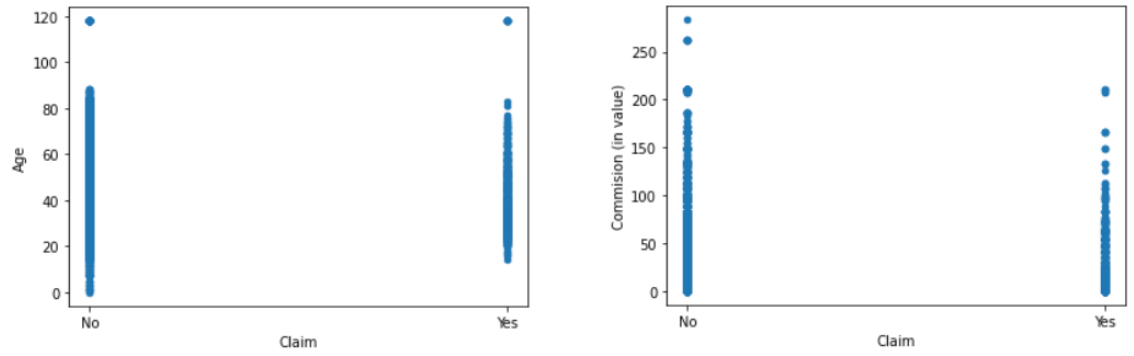

Figure 11- Scatter Plot w.r.t 'Claim'

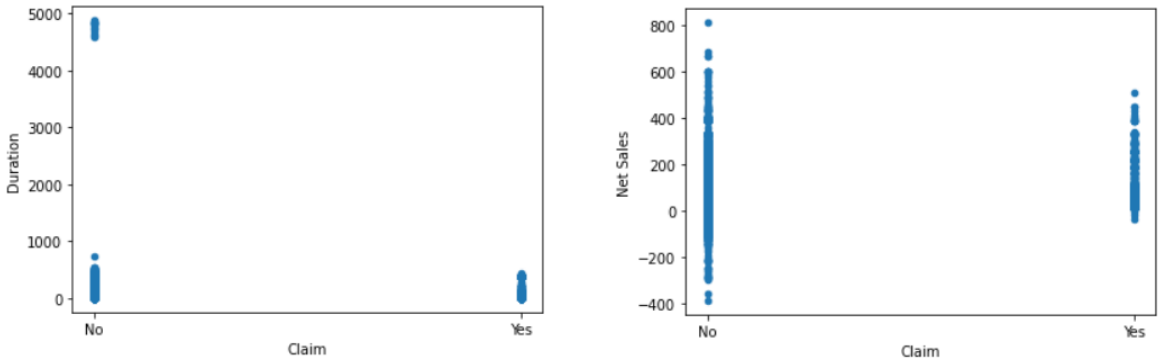

Figure 12- Scatter Plot w.r.t 'Claim'

From the Figure 11 and Figure 12, it is observed that,

- People of age lying between 1 and 90 almost have both the cases of 'Yes' and 'No' of claims with the probability of claim be No more but there are outlier in each case having the age 118.
- There is the probability of claims being both 'Yes' and 'No' when the commission ranges form 0-200, but the probability of the model predicting it as No is more as the number of data points is more when there Claim is 'No'.
- When the Duration of travel is too high i.e. more than 4500, then the claim will always be 'No'.

When the net sales is between 0 to -400 and 560-800, the claim was found to be 'No'

5 Feature Engineering

5.1 Missing values handling

The approaches for handling missing values are:

- Impute Categorical with Mode , Continuous with Median or Mean (if Outliers are handled)
- Use the previous or next value for the column (Pandas *.fillna* method -*{ffill ,bfill}*) when data shows a trend
- Utilize other fields to derive the value. Observations via bi-variate, multivariate analysis
- Impute (Categorical data only) an unseen/dummy constant value

In the given dataset, there are missing values only in the feature `'Gender'`. So, to handle the missing values in Gender feature, we have gone for the 4th approach i.e. by imputing a constant 'N' where the value is missing.

5.2 Outliers Handling

The outlier handling can be done on only continuous variables, the approaches for handling outliers are:

- Flooring & Capping the Outliers using Quantiles
- Transformations - Logarithmic or Square Root
- Replacement using Median Values
- Removing the Outliers using IQR/Confidence Intervals
- Removing the records having outliers if there are very less outliers compared to the total size of the dataset, so that it deleting those records will not have any impact on the further performance of the model.

Here, in the dataset, outliers are present in the two features i.e. `'Age'` and `'Duration'`. For handling the outliers in the `'Age'`, the records with age greater than 100 are removed. As for duration, the records having duration 0 or less than 0 are removed.

5.3 Encoding

After the feature engineering is done, the categorical features are encoded via one hot encoding and the label encoding is done on the target variable and the numeric columns remain as it is.

- *Label Encoding*: Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering.
- *One Hot Encoding*: One-Hot Encoding is another popular technique for treating categorical variables. It simply creates additional features based on the number of unique values in the categorical feature. Every unique value in the category will be added as a feature.

6 Class imbalance problem

Classification algorithms works best when the number of instances of each classes are roughly equal. When the number of instances of one class far exceeds the other, problems arise. In imbalanced cases standard classifier algorithms have a bias towards classes which have large number of instances. They tend to only predict the majority class data. The features of the minority class are treated as noise and are often ignored. Thus, there is a high probability of misclassification of the minority class as compared to the majority class.
In the case of this project the Label 0 class [Claim Rejected] constitute about 98.53% and the Label 1 class [Claim Accepted] constitute about 1.47%.
The dataset used in this project is highly imbalanced having a ratio of 98.53: 1.47 proportional to Label 0: Label 1. And, because of this high imbalance, it is necessary to balance the class for the proper predictions by the model.

Originally, the distribution of class labels is as given below:

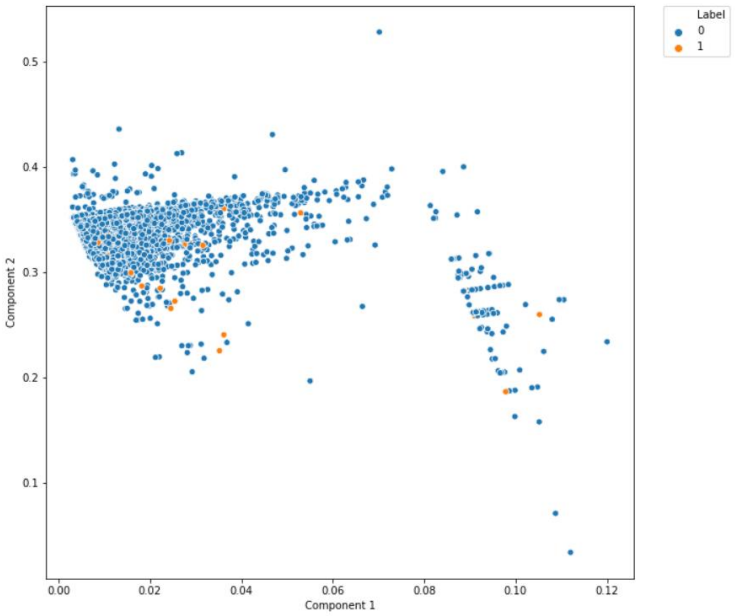

Figure 13- Distribution of Class Labels – No sampling

6.1 Approaches tried for handling class imbalance

Following techniques were applied on the dataset to balance the imbalance in the class:

6.1.1 Random under Sampling

Random under sampling (RUS) involves randomly selecting examples from the majority class to delete from the training dataset. This has the effect of reducing the number of examples in the majority class in the transformed version of the training dataset. This process can be repeated until the desired class distribution is achieved, such as an equal number of examples for each class.

After RUS, the distribution of class label is:

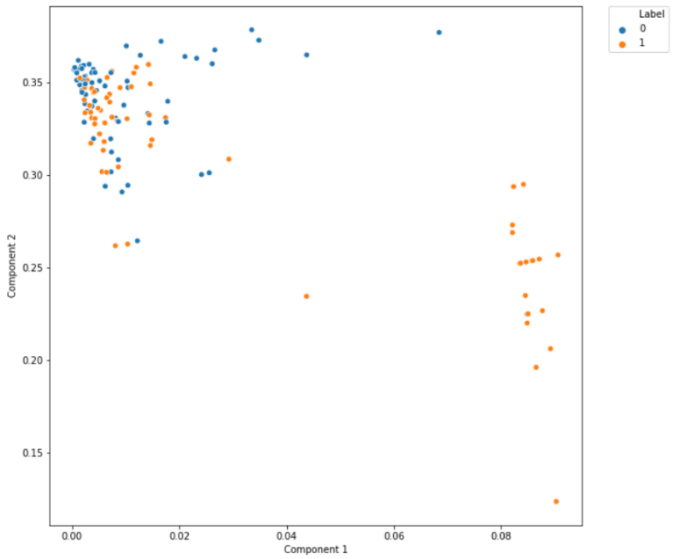

Figure 14-- Distribution of Class Labels – Random Under Sampling

6.1.2 Random over Sampling:

Random oversampling involves randomly duplicating examples from the minority class and adding them to the training dataset. Examples from the training dataset are selected randomly with replacement. This means that examples from the minority class can be chosen and added to the new "more balanced" training dataset multiple times; they are selected from the original training dataset, added to the new training dataset, and then returned or "replaced" in the original dataset, allowing them to be selected again.

After ROS, the distribution of class label is:

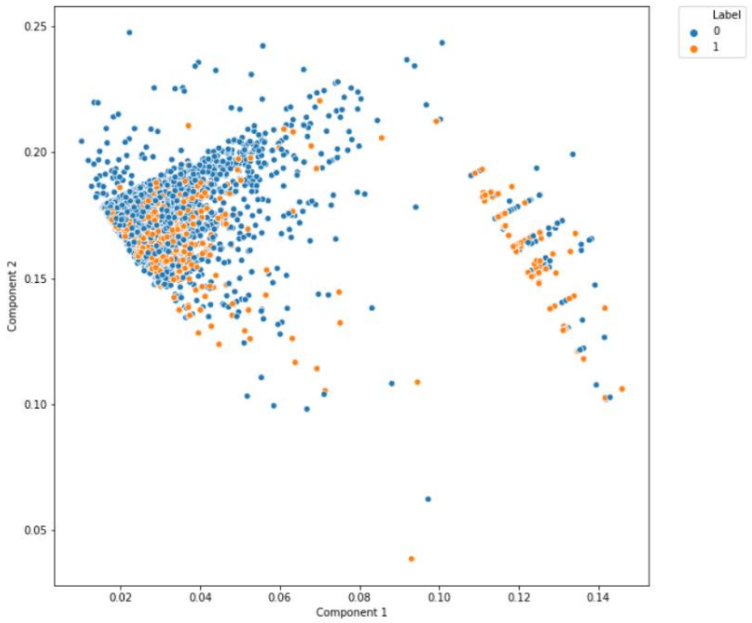

Figure 15- - Distribution of Class Labels – Random Over Sampling

6.1.3 SMOTE + Tomek Links

Synthetic Minority Over-sampling Technique(SMOTE) uses over-sampling approach in which the minority class is over-sampled by creating "synthetic" examples based upon the existing minority observations rather than by over-sampling with replacement.

Tomek Links removes unwanted overlap between classes where majority class links are removed until all minimally distanced nearest neighbor pairs are of the same class.

After SMOTE + Tomek Links, the distribution of class label is:

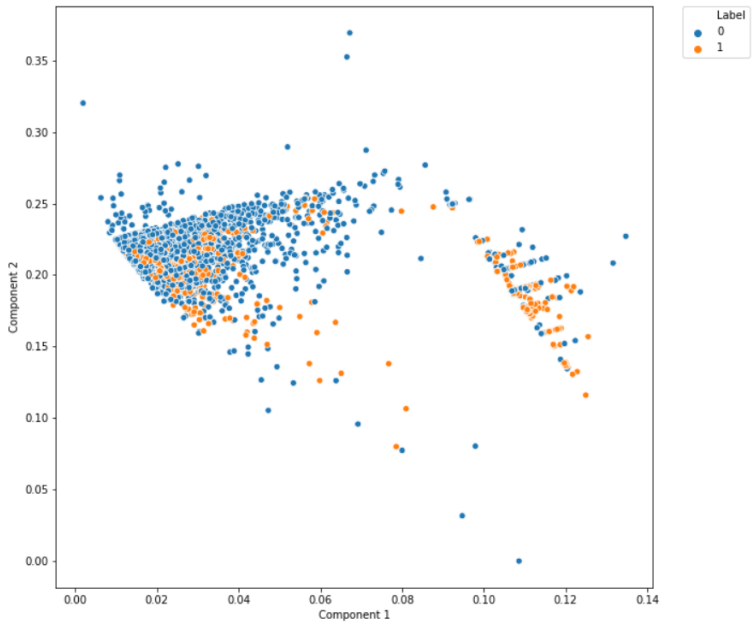

Figure 16- Distribution of Class Labels – SMOTE + Tomek Links

6.1.4 ADASYN

The essential idea of ADASYN is to use a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data is generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn.

After ADASYN, the distribution of class label is:

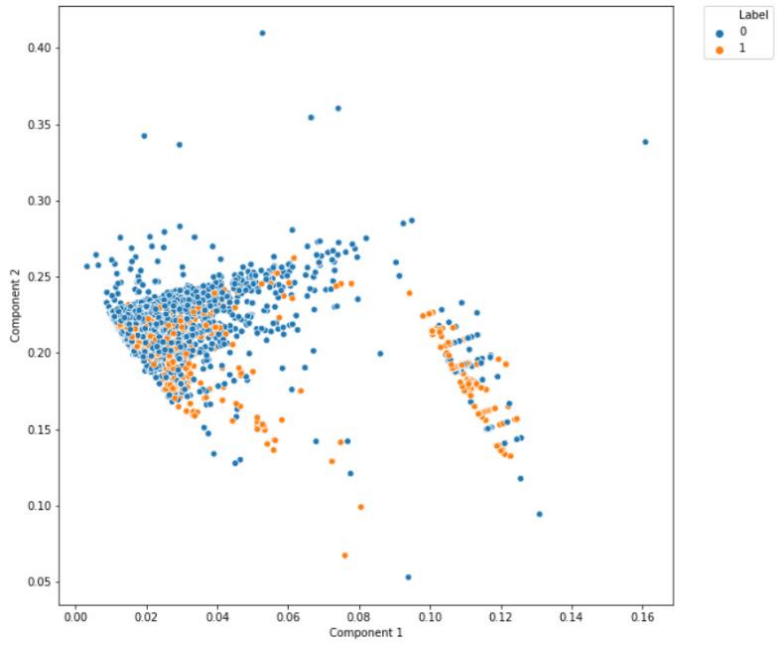

Figure 17- Distribution of Class Labels – ADASYN

7 Results

7.1 Confusion Matrix:

A confusion matrix is used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. A confusion matrix is a table that categorizes predictions according to whether they match the actual value.

Let the binary class labels be '1' and '0'. Let x be a test instance.

		Predicted	
		'0'	**'1'**
Actual	**'0'**	*True Negative*	*False Positive*
	'1'	*False Negative*	*True Positive*

Table 3- Confusion Matrix

- *True Positive (TP):* Let the true class label of x be '1'. If the model predicts the class label of x as '1', then the classification of x is True Positive.
- *False Positive (FP):* Let the true class label of x be '0'. If the model predicts the class label of x as '1', then the classification of x is False Positive.
- *True Negative (TN):* Let the true class label of x be '0'. If the model predicts the class label of x as '0', then the classification of x is True Negative.
- *False Negative (FN)*: Let the true class label of x be '1'. If the model predicts the class label of x as '0', then the classification of x is False Negative.

7.2 Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

7.3 Recall

Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (a small number of FN).

$$Recall = \frac{TP}{TP + FN}$$

7.4 Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate.

$$Precision = \frac{TP}{TP + FP}$$

7.5 F1 score

It is used as a statistical measure to rate performance. In other words, an F1-score is a mean of an individual's performance, based on two factors i.e. precision and recall.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

8 Modeling Efforts

8.1 Model chosen

Several Machine learning algorithms were used to train the model:

- ***Decision Tree***: A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter.
- ***Random Forest***: It is an ensemble tree-based learning algorithm. The Random Forest Classifier is a set of decision trees from randomly selected subset of training set. It aggregates the votes from different decision trees to decide the final class of the test object.
- ***Logistic Regression***: Logistic regression is used when the dependent variable is binary (0/1, True/False, Yes/No) in nature. Even though the output is a binary variable, what is being sought is a probability function which may take any value from 0 to 1. It is a statistical model that predicts the probability of an outcome that can only have two values.
- ***XGBoost Classifier***: XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks.

				Accuracy	Precision	Recall	F1 Score	AUROC Score
With Column 'Gender'	Label Encoding	Without Hyperparameter Tuning	Decision Tree	97%	4%	4%	4%	51%
			Random Forest	98%	6%	1%	2%	50%
			Logistic Regression	98%	0%	0%	0%	50%
		With Hyperparameter Tuning	Decision Tree	96%	6%	8%	7%	53%
			Random Forest	83%	5%	68%	10%	75%
			Logistic Regression	77%	4%	74%	8%	76%
	One Hot Encoding	Without Hyperparameter Tuning	Decision Tree	97%	6%	6%	6%	52%
			Random Forest	98%	5%	1%	2%	50%
			Logistic Regression	98%	0%	0%	0%	50%
		With Hyperparameter Tuning	Decision Tree	90%	6%	44%	11%	66%
			Random Forest	85%	6%	65%	11%	75%
			Logistic Regression	78%	5%	79%	9%	78%
Without Column 'Gender'	Label Encoding	Without Hyperparameter Tuning	Decision Tree	97%	5%	6%	6%	52%
			Random Forest	98%	0%	0%	0%	50%
			Logistic Regression	98%	0%	0%	0%	50%
		With Hyperparameter Tuning	Decision Tree	95%	5%	13%	7%	55%
			Random Forest	83%	5%	66%	10%	75%
			Logistic Regression	83%	5%	67%	10%	75%
	One Hot Encoding	Without Hyperparameter Tuning	Decision Tree	97%	5%	6%	6%	52%
			Random Forest	98%	0%	0%	0%	50%
			Logistic Regression	98%	0%	0%	0%	50%
		With Hyperparameter Tuning	Decision Tree	94%	8%	27%	12%	61%
			Random Forest	84%	6%	66%	11%	75%
			Logistic Regression	80%	5%	76%	10%	78%

Figure 18- Model Performance with and without column 'Gender'

8.2 Hyperparameter tuning

It is the process of selecting the parameters for the model which gives the best results. It can be done via two ways: Random Search and Grid Search.

- *Grid Search*: Grid search is where you pick x number of values that are evenly spaced along each axis. This forms a grid — hence the name.
- *Random Search*: Random search is when x-squared number of values are picked randomly.

In the project, Grid Search was used for hyperparameter tuning and was implemented using GridSearchCV.

The best values selected by GridSeacrhCV after Hyperparameter Tuning *in* case of Random Forest are

Parameter	Values
'class_weight'	'balanced'
'criterion'	'entropy'
'max_depth'	30
'max_leaf_nodes'	40
'min_samples_split'	3
'n_estimators'	100
'random_state'	42

Table 4- Selected Values after Hyperparameter Tuning

				Accuracy	Precision	Recall	F1 Score	AUROC Score
Label Encoding	Without GridSearchCV	No Sampling	Decision Tree	97.05	4.68	5.24	4.94	51.61
			Logistic Regression	98.52	0.00	0.00	0.00	49.99
			Random Forest	98.34	6.06	0.87	1.52	50.33
		Under Sampling	Decision Tree	65.26	2.63	62.88	5.05	64.09
			Logistic Regression	82.44	5.28	64.62	9.76	73.67
			Random Forest	73.95	3.89	70.74	7.39	72.37
		Over Sampling	Decision Tree	97.11	7.36	8.29	7.80	53.37
			Logistic Regression	83.49	5.57	64.19	10.25	73.98
			Random Forest	97.79	5.34	3.05	3.88	51.12
	With GridSearchCV	No Sampling	Decision Tree	96.76	6.34	8.73	7.35	53.40
			Logistic Regression	98.51	0.00	0.00	0.00	49.99
			Random Forest	85.67	6.28	62.88	11.42	74.44
		Under Sampling	Decision Tree	69.89	3.28	68.55	6.27	69.23
			Logistic Regression	76.92	3.24	51.09	6.10	64.19
			Random Forest	81.34	5.30	69.43	9.86	75.47
		Over Sampling	Decision Tree	97.15	6.82	7.42	7.11	52.95
			Logistic Regression	80.07	4.23	58.07	7.88	69.23
			Random Forest	84.35	5.94	65.06	10.89	74.85

Figure 19- Model Performance when class balancing approach is RUS and ROS with and without hyperparameter tuning in Label Encoded Dataframe

				Accuracy	Precision	Recall	F1 Score	AUROC Score
One Hot Encoding	Without GridSearchCV	No Sampling	Decision Tree	97.06	6.81	7.86	7.30	53.12
			Logistic Regression	98.53	0.00	0.00	0.00	50.00
			Random Forest	98.30	5.12	0.87	1.49	50.31
		Under Sampling	Decision Tree	67.11	2.97	67.68	5.70	67.39
			Logistic Regression	77.83	4.69	72.92	8.81	75.41
			Random Forest	74.77	4.16	73.36	7.87	74.08
		Over Sampling	Decision Tree	96.90	5.90	7.42	6.57	52.82
			Logistic Regression	78.53	1.81	72.48	9.02	75.55
			Random Forest	97.69	6.04	3.93	4.76	51.50
	With GridSearchCV	No Sampling	Decision Tree	86.30	5.67	53.27	10.26	70.03
			Logistic Regression	80.09	4.23	58.07	7.89	69.24
			Random Forest	85.44	6.18	62.88	11.26	74.33
		Under Sampling	Decision Tree	76.57	4.00	65.06	7.54	70.90
			Logistic Regression	78.92	4.95	73.36	9.27	76.18
			Random Forest	80.30	5.09	70.30	9.49	75.37
		Over Sampling	Decision Tree	85.52	4.33	42.35	7.86	64.21
			Logistic Regression	78.77	4.86	72.48	9.12	75.67
			Random Forest	84.71	5.96	63.75	10.91	74.38

Figure 20- Model Performance when class balancing approach is RUS and ROS with and without hyperparameter tuning in One Hot Encoded Dataframe

ONE HOT ENCODING WITH HYPERPARAMETER TUNING							
			Accuracy	Precision	Recall	F1 Score	AUROC Score
No Sampling		Logistic Regression	80.55	5.12	69.86	9.55	75.29
		Random Forest	85.40	6.17	62.88	11.24	74.31
Under Sampling		Logistic Regression	78.92	4.95	73.36	9.27	76.18
		Random Forest	80.25	5.07	70.30	9.47	75.35
Over Sampling		Logistic Regression	78.77	4.86	72.48	9.12	75.67
		Random Forest	84.27	5.94	65.50	10.90	75.02
SMOTE + Tomek	0.3	Logistic Regression	81.91	5.34	67.68	9.91	74.90
		Random Forest	86.39	6.44	61.13	11.66	73.95
	0.5	Logistic Regression	79.77	5.72	72.05	9.47	75.97
		Random Forest	86.39	6.40	60.69	11.58	73.73
	0.7	Logistic Regression	79.59	5.02	72.05	9.40	75.88
		Random Forest	86.49	6.49	61.13	11.74	74.00
ADASYN	0.3	Logistic Regression	81.53	5.23	67.68	9.72	74.71
		Random Forest	86.22	6.32	60.69	11.46	73.65
	0.5	Logistic Regression	79.55	4.96	71.17	9.27	75.42
		Random Forest	86.49	6.45	60.69	11.66	73.78
	0.7	Logistic Regression	79.59	5.02	72.05	9.40	75.88
		Random Forest	86.30	6.44	61.57	11.67	74.12

Figure 21-- Model Performance when class balancing approach is RUS, ROS, SMOTE+Tomek Links and ADASYN with hyperparameter tuning in One Hot Encoded Dataframe

(XGBoost) ONE HOT ENCODING WITH HYPERPARAMETER TUNING						
		Accuracy	Precision	Recall	F1 Score	AUROC Score
No Sampling		98.53	0.00	0.00	0.00	50.00
Under Sampling		81.34	5.27	68.99	9.80	75.26
Over Sampling		83.99	6.05	68.12	11.11	76.17
SMOTE + Tomek	1	91.44	8.36	48.45	14.26	70.27
	0.8	92.79	8.68	41.04	14.34	67.30
	0.3	96.68	11.73	19.21	14.56	58.52
	0.1	98.15	13.75	4.80	7.11	52.17
ADASYN	1	90.00	8.09	55.45	14.12	73.03
	0.8	91.16	8.40	50.65	14.41	71.21
	0.3	96.52	11.76	20.96	15.07	59.30
	0.1	98.47	0.00	0.00	0.00	49.97

Figure 22- XGBoost Classifier Performance with ROS, RUS, SMOTE + Tomek Links and ADASYN

Among all these the best performance was observed in Random Forest Algorithm with Random Under Sampling method for class imbalance

8.3 Thresholds selection

Traditionally the predicted probabilities of the class is selected by the threshold of 0.5, but by changing the threshold the variation in the result can be seen.
Different threshold from 0.5 to 0.95 with the difference of 0.05 was chosen and the variation was recorded.
The best result was obtained with the threshold of 0.65

Since the problem was of class imbalance and that too of the ratio of 1.4: 98.6, so the optimum result was considered with respect to Recall.

Recall and Precision are in contradictory when one increases the other decreases, the selection for the recall or precision is fairly dependent on the nature and need of the hour. Hence, it is appropriate to maximise the recall and keeping precision's value fairly considerable.

The final optimum result was found having at least 3.5% of precision and maximum recall.

Random Under Sampling with Hyperparameter Tuning and cross validation = 10														
Thersh	TP	TN	FP	FN	Total	TP (%)	TN(%)	FP(%)	FN(%)	Accuracy	Precision	Recall	Label 0	Label 1
0.50	160	12572	2785	69	15586	69.87	81.86	18.14	30.13	81.69	5.43	69.87	15357	229
0.55	165	11972	3385	64	15586	72.05	77.96	22.04	27.95	77.87	4.65	72.05	15357	229
0.60	177	11264	4093	52	15586	77.29	73.35	26.65	22.71	73.41	4.15	77.29	15357	229
0.65	190	10220	5137	39	15586	82.97	66.55	33.45	17.03	66.79	3.57	82.97	15357	229
0.70	199	8892	6465	30	15586	86.90	57.90	42.10	13.10	58.33	2.99	86.90	15357	229
0.75	214	6981	8376	15	15586	93.45	45.46	54.54	6.55	46.16	2.49	93.45	15357	229
0.80	223	4271	11086	6	15586	97.38	27.81	72.19	2.62	28.83	1.97	97.38	15357	229
0.85	228	1370	13987	1	15586	99.56	8.92	91.08	0.44	10.25	1.60	99.56	15357	229
0.90	229	0	15357	0	15586	100.00	0.00	100.00	0.00	1.47	1.47	100.00	15357	229
0.95	229	0	15357	0	15586	100.00	0.00	100.00	0.00	1.47	1.47	100.00	15357	229

Random Under Sampling with Hyperparameter Tuning and cross validation = 3														
Thresh	TP	TN	FP	FN	Total	TP (%)	TN(%)	FP(%)	FN(%)	Accuracy	Precision	Recall	Label 0	Label 1
0.50	159	12572	2785	70	15586	69.43	81.86	18.14	30.57	81.68	5.40	69.43	15357	229
0.55	165	11996	3361	64	15586	72.05	78.11	21.89	27.95	78.03	4.68	72.05	15357	229
0.60	177	11276	4081	52	15586	77.29	73.43	26.57	22.71	73.48	4.16	77.29	15357	229
0.65	189	10335	5022	40	15586	82.53	67.30	32.70	17.47	67.52	3.63	82.53	15357	229
0.70	203	8724	6633	26	15586	88.65	56.81	43.19	11.35	57.28	2.97	88.65	15357	229
0.75	215	6257	9100	14	15586	93.89	40.74	59.26	6.11	41.52	2.31	93.89	15357	229
0.80	222	3840	11517	7	15586	96.94	25.00	75.00	3.06	26.06	1.89	96.94	15357	229
0.85	228	1004	14353	1	15586	99.56	6.54	93.46	0.44	7.90	1.56	99.56	15357	229
0.90	229	0	15357	0	15586	100.00	0.00	100.00	0.00	1.47	1.47	100.00	15357	229
0.95	229	0	15357	0	15586	100.00	0.00	100.00	0.00	1.47	1.47	100.00	15357	229

Figure 23- Performance of Random Forest with RUS technique for class imbalance with different thresholds and cross validation 10 and 3

8.4 Final model chosen

Random Forest algorithm with hyperparameter tuning for model training and Random Under Sampling for treating imbalanced data at a threshold of 0.65 was found to be give the best result.
Results obtained from the above model are:

TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
190	10150	5207	39	66.34%	3.52%	82.96%	6.75%

Table 5- Result of the final model after applying threshold

9 Feature Importance

Feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable. Feature importance scores play an important role in a predictive modeling project, including providing insight into the data, insight into the model, and the basis for dimensionality reduction and feature selection that can improve the efficiency and effectiveness of a predictive model on the problem.

Here, in the model the top 10% of the total important features in the one hot encoded dataset are:

Index	Percent	Top 10% Features
186	0.112815	Agency_C2B
1	0.107123	Net Sales
2	0.102839	Commision (in value)
0	0.091515	Duration
8	0.071606	Gender_N
128	0.055608	Destination_SINGAPORE
9	0.046254	Agency Type_Airlines
3	0.042710	Age
191	0.040356	Agency_EPX
10	0.035146	Agency Type_Travel Agency
168	0.030754	Product Name_Cancellation Plan
162	0.018845	Product Name_Annual Silver Plan
192	0.016095	Agency_JZI
159	0.015508	Product Name_2 way Comprehensive Plan
6	0.015302	Gender_F
166	0.014085	Product Name_Basic Plan
167	0.012395	Product Name_Bronze Plan
149	0.009834	Destination_UNITED STATES
194	0.009397	Agency_LWC

Table 6- Top 10 Percent Important Features

10 Reasoning lines for model prediction

From the above given important features in Table 6, the reasoning for the given prediction of the model can be given on the basis of top 5 important features.

From, these features (Categorical), if a feature, say '*Parent Feature_Feature Value'* is 1, then it that feature is present and the reasoning line would be '*Parent feature was found to be Feature Value*'

For the features (Continuous), it the value is more than its mean then it is '*large'* and '*small'* in other case.

Some of the Reasoning lines are:

- *Agency Type was found to be Travel Agency while Net Sales, Commision (in value) and Duration value was small',*

- *'Agency Type was found to be Travel Agency while Commision (in value) and Duration value was large & Net Sales value was small',*
- *'Agency Type was found to be Travel Agency while Net Sales and Duration value was large & Commision (in value) value was small',*
- *'Agency Type was found to be Travel Agency while Duration value was large & Net Sales and Commision (in value) value was small',*
- *'Commision (in value) value was large & Net Sales and Duration value was small',*
- *'Agency Type was found to be Travel Agency while Net Sales value was large & Commision (in value) and Duration value was small',*
- *'Destination was found to be SINGAPORE, Agency Type was found to be Travel Agency while Net Sales, Commision (in value) and Duration value was small',*
- *'Net Sales, Commision (in value) and Duration value was small',*

Reasoning	TP	TN	FP	FN
Agency Type was found to be Travel Agency while Commision (in value) and Duration value was large & Net Sales value was small	0.0	275.0	33.0	0.0
Agency Type was found to be Travel Agency while Commision (in value) value was large & Net Sales and Duration value was small	2.0	754.0	84.0	6.0
Agency Type was found to be Travel Agency while Duration value was large & Net Sales and Commision (in value) value was small	0.0	1343.0	32.0	9.0
Agency Type was found to be Travel Agency while Net Sales and Commision (in value) value was large & Duration value was small	7.0	204.0	400.0	4.0
Agency Type was found to be Travel Agency while Net Sales and Duration value was large & Commision (in value) value was small	3.0	622.0	116.0	5.0
Agency Type was found to be Travel Agency while Net Sales value was large & Commision (in value) and Duration value was small	2.0	1219.0	81.0	11.0
Agency Type was found to be Travel Agency while Net Sales, Commision (in value) and Duration value was large	9.0	91.0	301.0	0.0
Agency Type was found to be Travel Agency while Net Sales, Commision (in value) and Duration value was small	0.0	4784.0	45.0	21.0
Agency was found to be C2B, Destination was found to be SINGAPORE while Commision (in value) and Duration value was large & Net Sales value was small	1.0	0.0	17.0	0.0
Agency was found to be C2B, Destination was found to be SINGAPORE while Commision (in value) value was large & Net Sales and Duration value was small	6.0	0.0	98.0	0.0
Agency was found to be C2B, Destination was found to be SINGAPORE while Duration value was large & Net Sales and Commision (in value) value was small	7.0	5.0	84.0	0.0

Figure 24- Reasoning Results with contribution in each bucket [TP, TN, FP, FN]

11 Conclusion and Git Repository

- ❖ In the dataset the acceptance of the Claim was in minority as compared to the rejection of the insurance claim. Due to the inherit nature of the data there was a class imbalance problem.
- ❖ In the early stage of the model after trying all the possible approaches (Balancing the Imbalanced Class, Encoding, ML Models, Hyperparameter Tuning) the maximum obtained Precision was about 5-6% and Recall was about 72%
- ❖ XGBoost classifier was able to give a maximum of 13% precision but the Recall went down to 5%, and the company was interested in model resulting to the maximum Recall.
- ❖ After further exploring the techniques to get the most optimum results a decision was taken to maximize the recall with accepting the 1-2% hit of precision.
- ❖ The Final Machine Learning Algorithm chosen to build the model was Random Forest with Hyperparameter Tuning and technique to treat the imbalanced class chosen was Random Under Sampling.
- ❖ Applying threshold selection technique to the trained model, a maximum Recall of 82% was obtained with a considerable 3.5% of Precision when the threshold selected was 0.65.

❖ Using the final model, Features Importance was done in which the top 10% of the input features was selected which was contributing the most in the prediction, enlisted in Table 6.

❖ Using the top 5 features a Reasoning line was generated which'll be helpful in explainibility part of the model.

- *'Agency was found to be C2B, Destination was found to be SINGAPORE while Commision (in value), Duration and Net Sales value was large'*, this was the reasoning line contribution to the maximum number of True Positive Case
- *'Agency was found to be EPX while Commision (in value), Duration and Net Sales value was small'*, this was the reasoning line contribution to the maximum number of True Negative Case
- *'Agency was found to be C2B, Destination was found to be SINGAPORE while Commision (in value), Duration and Net Sales value was small'*, this was the reasoning line contribution to the maximum number of False Positive Case
- *'Agency was found to be EPX while Commision (in value), Duration and Net Sales value was small'*, this was the reasoning line contribution to the maximum number of False Negative Case

❖ **GitHub Repository** : The link to the notebooks containing the final project code is : https://github.com/Internship-BVoc/InsuranceClaimPrediction/tree/master/_Main

12 Future learning

- **GAN can be used treat the imbalanced class.**

  Imbalance Data is leading to lower precision. Additional work can be carried out towards Class Imbalance handling Using Generative Adversarial Networks to produce Synthetic data for minority Class

- **Neural Nets can be used as Model training algorithm.**

  Neural Nets can be used to Train the model to see improvement in overall performance metrics.

- **Derive Features from existing features**

  It can be done by analyzing False Positive (FP) & False Negative (FN) cases to improve the precision

- **Model Interpretability**

  To enhance the explainibility of the model one can utilize packages such as SHAP (**SH**apley **A**dditive ex**P**lanations), LIME (**L**ocal **I**nterpretable **M**odel-Agnostic **E**xplanations), ELI5 (Explain Like I'm 5).

- **Interactive Visualization to see the classification data**

  Apply more interactive visualization tools to view and see classification of data, tools like t-SNE (t-distributed **S**tochastic **N**eighbor **E**mbedding)