**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 Precision and metrics showcasing model predictive Power. Also, Reason for Claims approval or rejection needs to be provided

**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, we first check 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 our dataset, so, the features in the dataset used in this project are:

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

In a dataset, we can distinguish 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'*

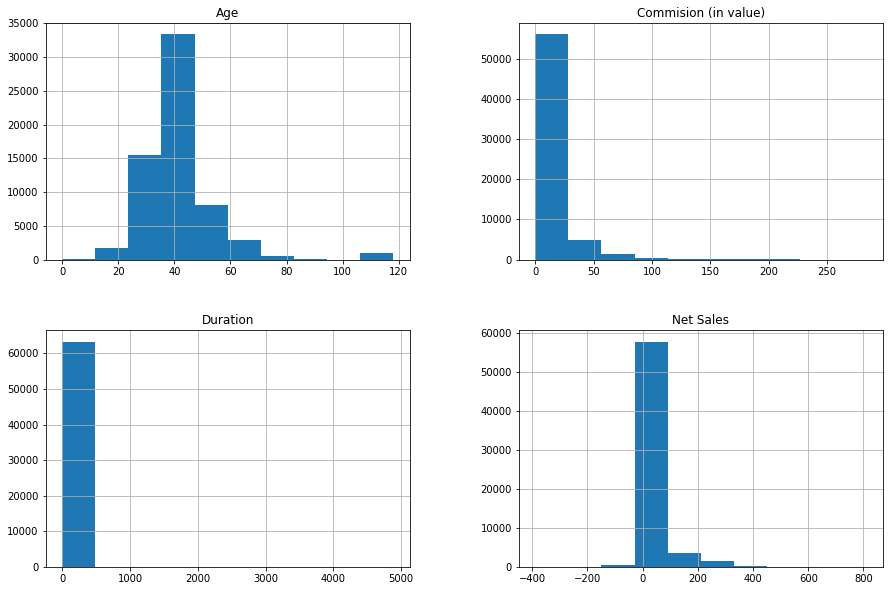
In the dataset, the continuous variables are:

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

* Traditionally *'Duration'* and *'Age'* should be continuous variables because
  + The skew 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 the feature of a dataset about which you want to gain a deeper understanding. Here, target variable is ‘Claim’

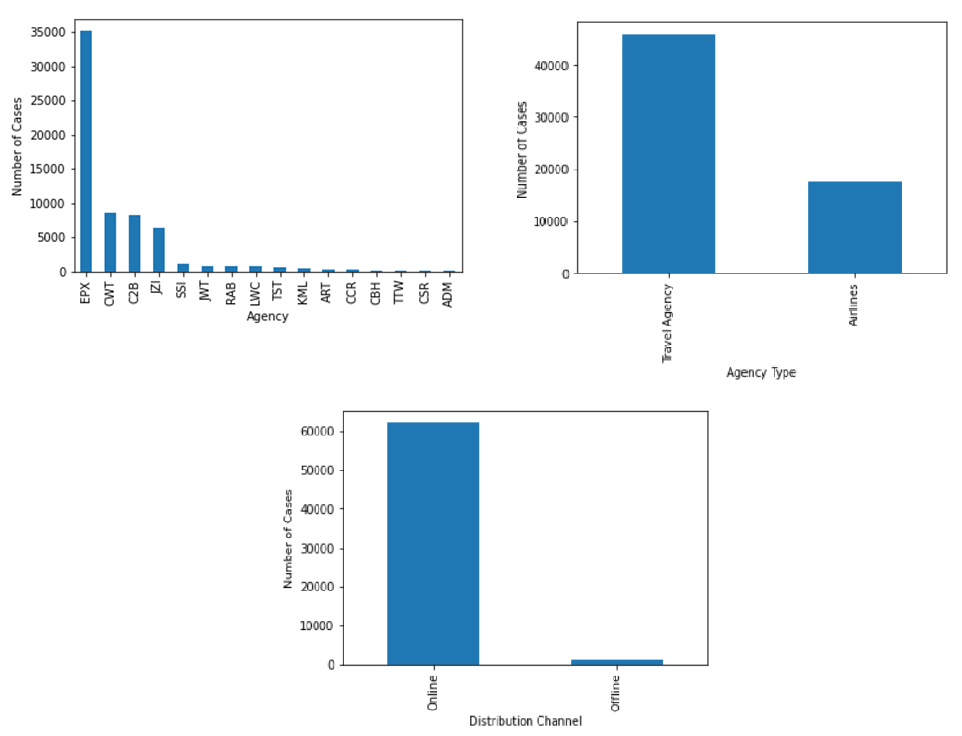
**Exploratory Analysis on Features**



From the above graphs, we can infer that:

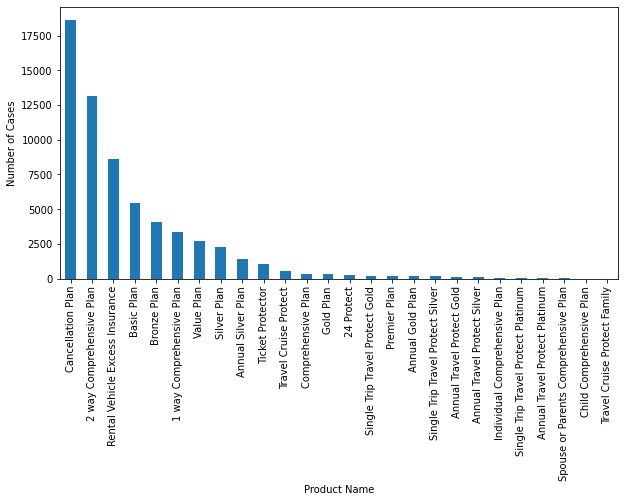
* The majority of people i.e. about 32500 of agency are of the age between 37 and 45, 16000 people are of age between 23 and 37, 8000 people are between the age group of 45-59, 3000 between 59-70, 2000 between 10-23 and rest between 70 and 120 leaving the gap of 94-114.
* Almost 57000 people agencies 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, hen 80-210 for 3000 cases.

From the graphs below:

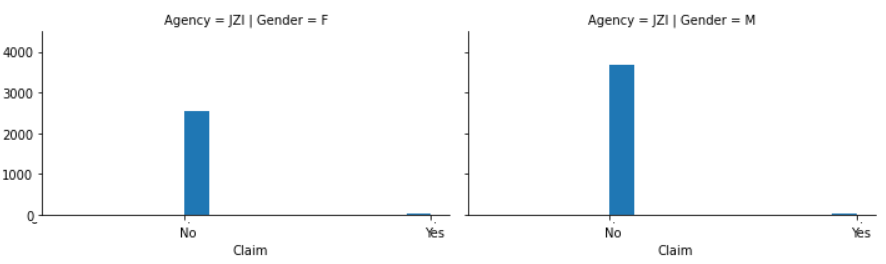


We, can clearly tell that, Agency EPX claims mostly, Travel Agencies apply more claims i.e. 70% while Airlines agencies apply less claims and almost all the cases except for very few have online distribution channel.

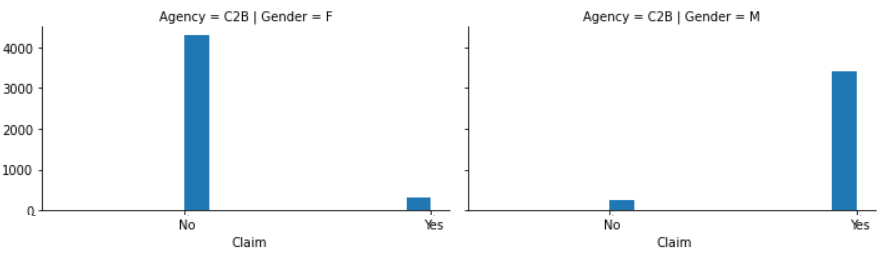
The below graph shows that the Cancellation plans are the more frequently used plan which is understandable as if the trip got cancelled, they'll hope for the getting the refund.

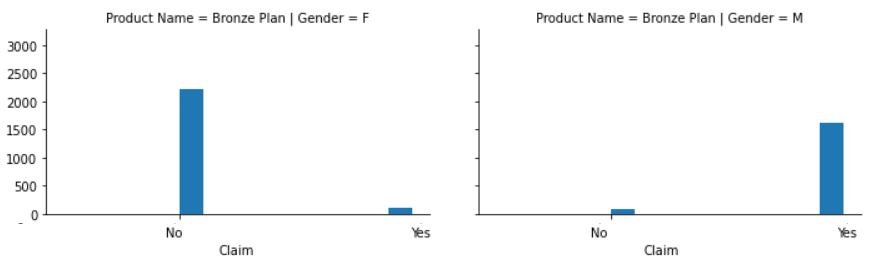


From the below graphs, we can conclude that, irrespective of the Gender, when the Agency is JZI, the claim is No. though the number of cases is more in case of gender is Male.



The graph given below 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.

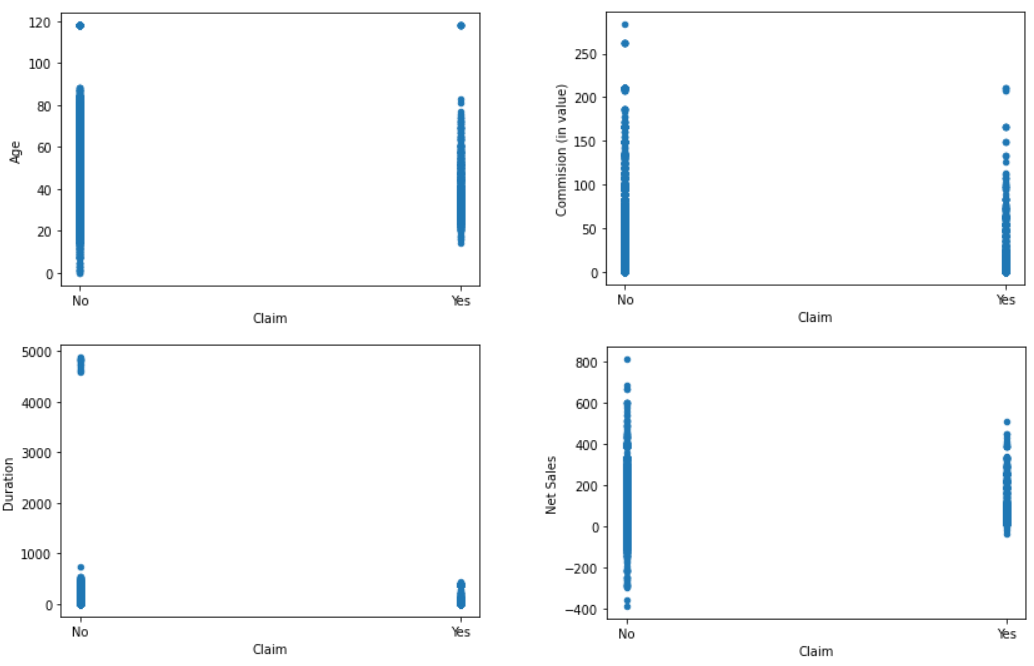




From the above graph, When the Product Name is Bronze, Cases of Claim being No is more when the applicant is Female, and Cases of Claim being Yes is more when gender is Male and vice versa.

From the below graphs, we can observe that,

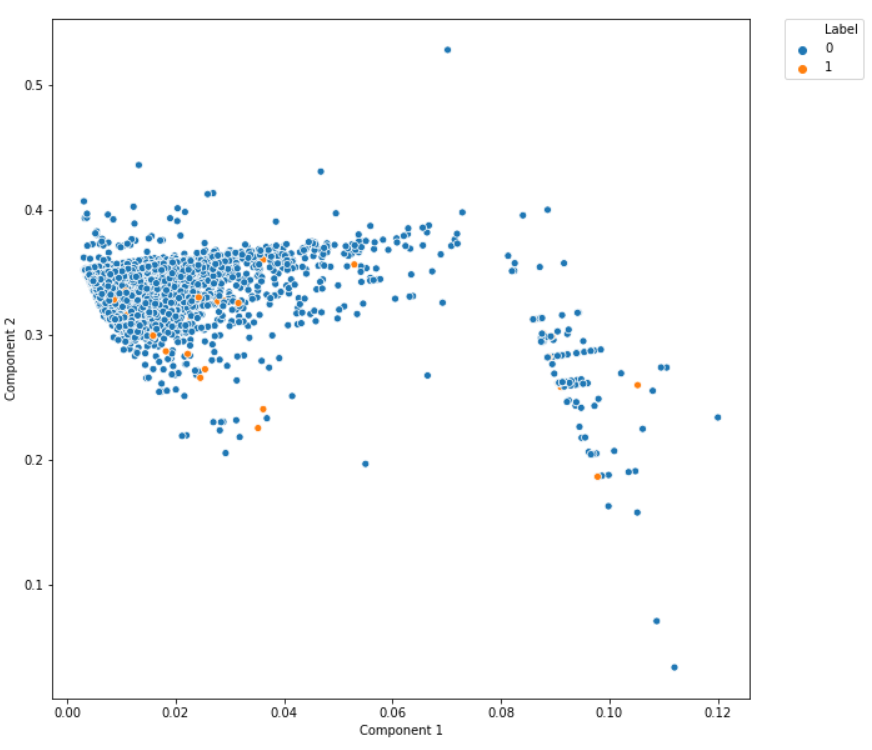
* People of age lying between 1 and 90 almost have both the cases of Yes and No of claims with the probability of claim be No more but there are outlier in each case having the age 118.
* There is the probability of claims being both Yes and No0 when the commission ranges form 0-200, but the probability 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 will always be No.



**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:

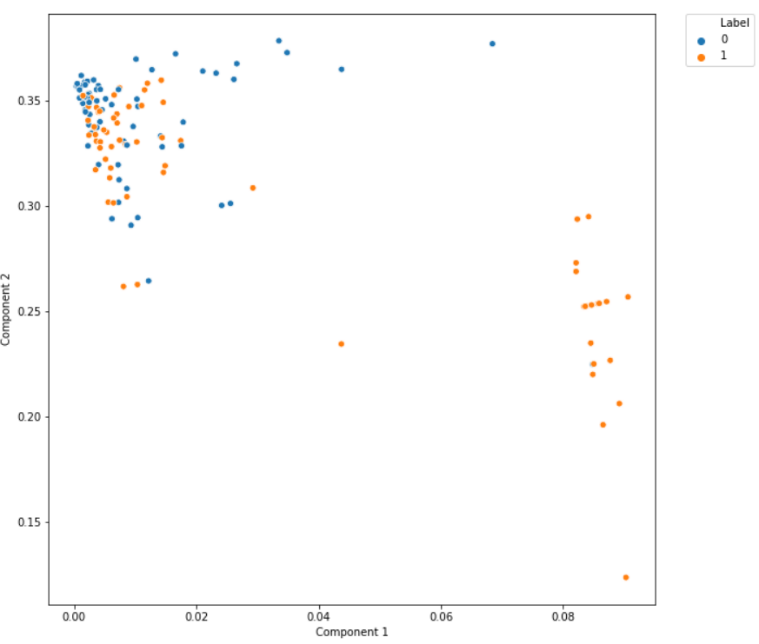


**Approaches tried for handling class imbalance**

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

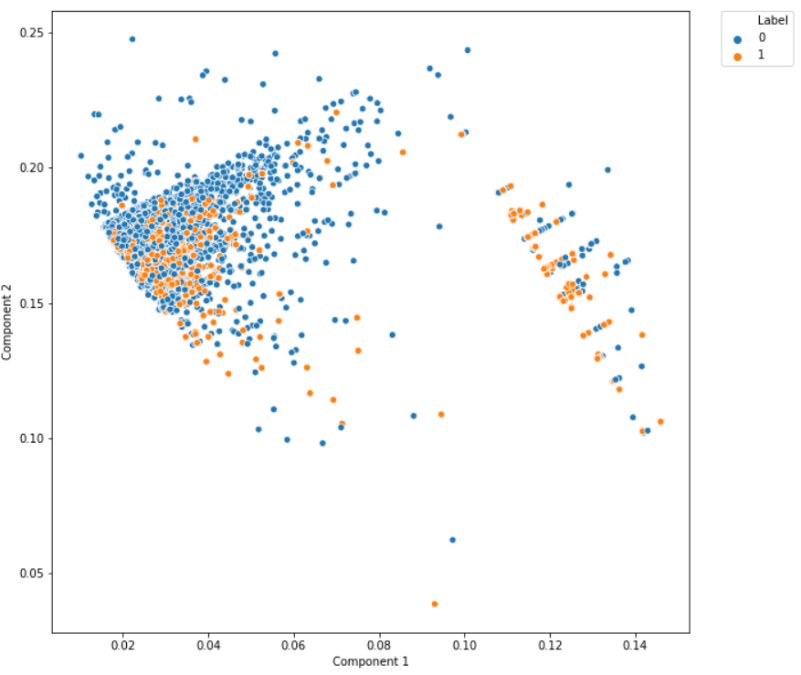
1. *Random under Sampling:* Random under sampling involves randomly selecting examples from the majority class to delete from the training dataset. This has the effect of reducing the number of examples in the majority class in the transformed version of the training dataset. This process can be repeated 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:



1. *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.

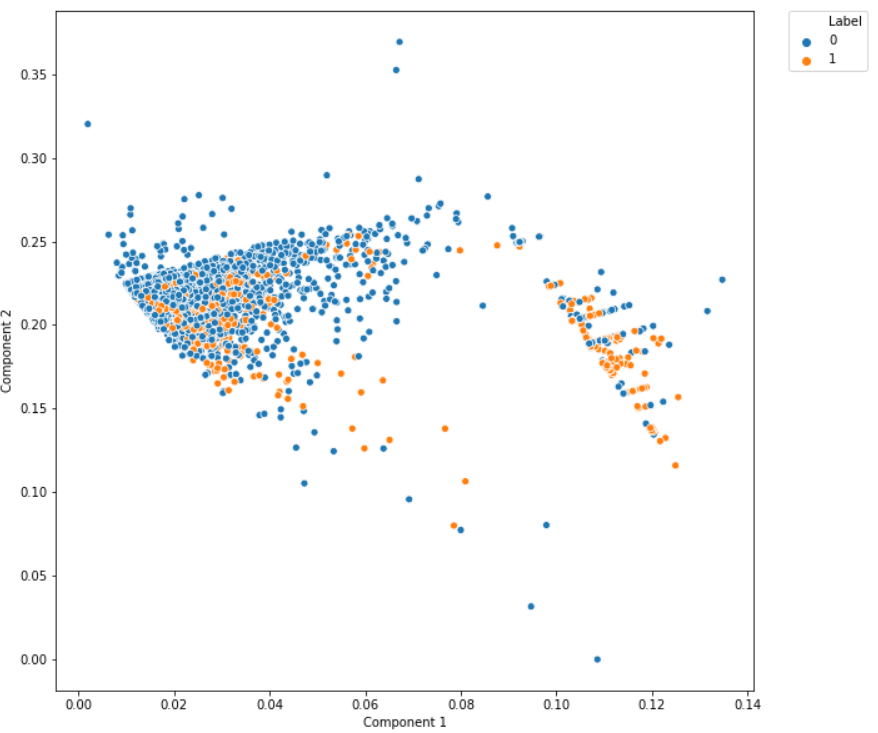
After ROS, the distribution of class label is:



1. *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.

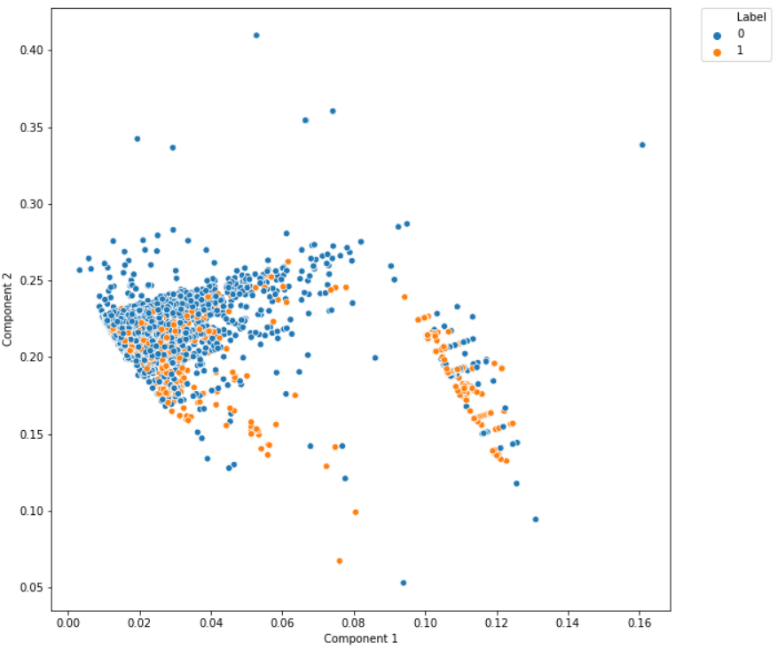
Tomek Links removes unwanted overlap between classes where majority class links are removed until all minimally distanced nearest neighbor pairs are of the same class.

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



1. *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:



The best results was obtained from Random Under Sampling

**Feature Engineering (Missing values, outliers, OHE)**

The approaches for handling missing values are:

* 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.

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 our 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.

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.

* 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.

**Results**

* Accuracy: Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.
* 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).
* 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.
* 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.
* 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 class labels be c and ¬c. Let x be a test instance.

* + True Positive (TP): Let the true class label of x be c. If the model predicts the class label of x as c, then we say that the classification of x is true positive.
  + False Positive (FP): Let the true class label of x be ¬c. If the model predicts the class label of x as c, then we say that the classification of x is false positive.
  + True Negative (TN): Let the true class label of x be ¬c. If the model predicts the class label of x as ¬c, then we say that the classification of x is true negative.
  + False Negative (FN): Let the true class label of x be c. If the model predicts the class label of x as ¬c, then we say that the classification of x is false negative.

**Modeling Efforts**

* 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.
* 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 our project, we use Grid Search for hyperparameter tuning via GridSearchCV.

* 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 0.6

* Final model chosen

Random Under Sampling with hyperparameter tuning was found to be the best model

**Reasoning lines for model prediction**

**Conclusion and Git Repository**

**Future learning**