**“Insurance Fraud Claims Detection by Machine Learning”**

**Introduction**-Insurance fraud has been since the beginning of the insurance organization. When a person makes a false claim to get benefits to which they are not entitled is known as an insurance fraud.  Insurance fraud is a serious problem, the loss to insurance companies due to fraud results in the need to raise insurance rates, increased premium cost, trust deficit during the claims process. So, detection of fraud is a challenging problem for an insurance industry.

Fraudulent claims can be highly expensive for this sector and are very common for this sector. One of the main reasons being that it is not humanly possible to check each claim on all parameters. It will also take lot of time, energy, and money. So, we use machine learning to predict which claims are most likely to be fraudulent. And based on that, the companies can take quicker decisions and avoid great losses.

Why Machine Learning is needed-Registered Insurance Company would have the capacity to examine each case and detect whether it is genuine or not, but this approach is not only time consuming but also costly. As per the information gathered, the most efficient strategy so far, to detect Insurance fraud is, we can use computerized techniques. Data Scientists can use Machine Learning techniques to reduce human efforts. This will give edge over redundant manual processes.

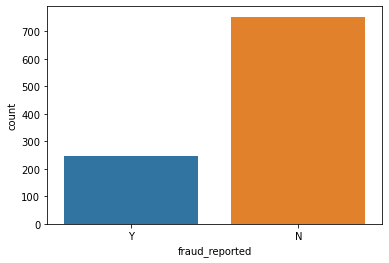
**Problem Definition**- First and most important task is to understand the data, make data from raw to featured, and completely understand each feature that affects the target variable (dependent variable). Target variable is binary, that is: fraud or not (yes or no). We have imported the data to jupyter (python environment), and on analyzing, it is evident that it is a binary classification model as the fraud reported is either YES or NO, so we will use the classification model to find out the r-result. We can understand the whole process as-

**Data Analysis-** The objective is to find out the fraudulent claims based on using classification models. The given dataset contains 1000 rows and 40 columns. It has object and integer data types and we can see there are no missing values in dataset. I have divided the dataset in to categorical and numerical features classifications to have more clarity. We have 21 columns for categorical values and 19 columns for numerical values.

I need to do data cleaning and feature corrections-

Just an overview gives the idea to drop ‘\_c39’ column as it won’t be affecting predictions, so I have dropped the same.

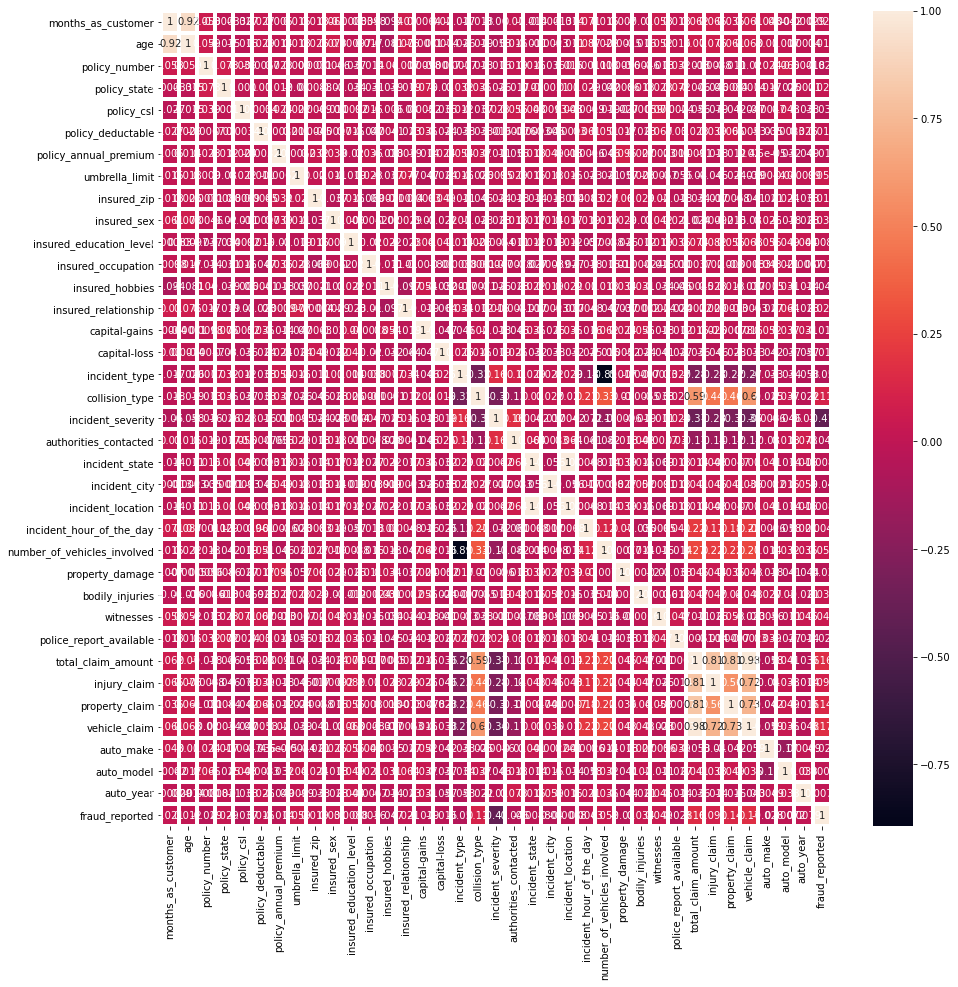
Target dataset is imbalanced hence either oversampling or under sampling has to be done to correct the same.



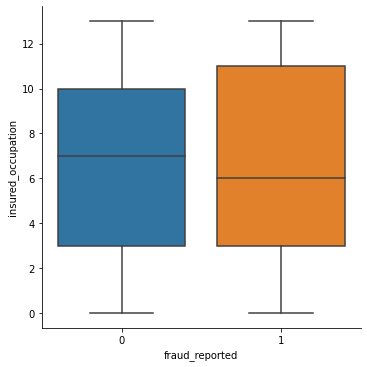
‘Policy bind date’ and ‘incident date’ column has been separated in date, month and year column, and I have also dropped the policy bind date and the incident date column.

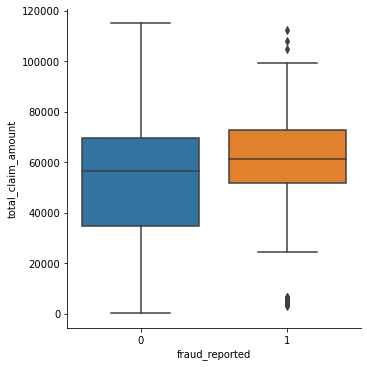
I need to convert categorical features to numerical ones for building machine learning models so using Label Encoder from sklearn to convert categorical features into numerical data.

To establish correlation among attributes, I have applied ‘correlation’ matrix and plotted graph. Target data is highly correlated with total claim amount and property claim than other features.



By plotting factor plot and count plot between the insured occupation and the fraud reported, it is observed that more claims are reported with high insured occupations. Similarly, fraud reported cases are high with total claim amount.





I used boxplot (Subplot) method to have a complete glance and found that there are outliers presents in the dataset and with distribution plots got distribution of data of all attributes at one glance then plotting individual graphs, applied this for boxplot and distribution plot to capture all images at one shot.

**EDA Concluding Remarks-**

After importing dataset to jupyter environment, I did data cleaning, formatting and feature engineering for the dataset. The data set comprises of 1000 rows and 40 columns, and is a mix of object and integer type. There are 21 categorical and 19 numerical columns. I have dropped \_c39 column as it is of no relevance.

I also found that the target data set is imbalanced, and oversampling is required. There are outliers present in dataset. Analysis of the factor plot shows that higher the number of insured occupation and total claim amount is directly proportional to the claims reported.

Correlation analysis also supports the same. Distribution plots exhibits that data are skewed.

**Preprocessing Pipeline-**

1 Applied skew to dataset to find out the skewness.

2 Applied ‘get dummies’ to police report available and property damage.

3 Corrected NaN and Infinity values present in dataset.

4 With Z score application removed outliers, and data loss is just 2% so it can be considered.

5 Data Standardization- from sklearn.preprocessing imported standard scaler for standardization of dataset.

6 Oversampling- since the target data set is very imbalanced so I had to do either over sampling or under sampling, but I opted for over sampling so from imblearn.oversampling imported SMOTE to normalize the target data set.

**Building Machine Learning Models-**

The next task is the selection of the model, which model is suitable according to our data. We firstly understand whether we are doing a regression problem or classification problem. In our case this is a classification problem. (either the claim is fraud or not) we have several models for this like logistic regression, random forest classification, decision tree classification, support vector etc. In our case we use random forest to get a higher accuracy.Since the problem statement is binary classification in nature, so imported Logistic regression model from sklrean.linear,Random Forest classifier from sklearn.ensemble,Decision Tree classifier from sklearn.tree and support vector form sklearn.svm as following-

from sklearn.linear model import Logistic Regression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

Then imported accuracy score from sklearn metrics to get accuracy score of the model, which is as follows-

from sklearn.metrics import accuracy score

imported train, test, split from sklearn metrics to classify the dataset in to train data & test data so that we can upload the engineered data set for prediction, which is as follows-

from sklearn.metrics import confusion matrix,classification\_report

from sklearn.model\_selection import train\_test\_split

Now when we classified data in to training and testing data, and run logistic regression models, Decision tree classification, Random forest classification and support vector machine then we get accuracy score of each model for given dataset, as in this case, we got following accuracy scores-

Accuracy score of Decision Tree classifier- 86%

Accuracy score of Random Forest classifier-90%

Accuracy score of Support vector-86%

From above it is clear that random forest model is giving 90% prediction but at this stage it is not right to conclude because higher efficiency of model can be because of over fitting.

So, to check this we will import cross validation score, as follows-

from sklearn.model\_selection import cross\_val\_score

and we will get Cross validation score for each model, and we get following result-

CV score for Decision Tree classifier-84%

CV score for Random Forest classifier-88%

CV score for Support vector-86%

And then I calculated the difference of each model’s accuracy score and cross validation score, and got two models for final consideration which are Random Forest classifier and Support vector.

Now to improve further efficiency for this models need to do hyperparameter tuning.

So, imported GridsearchCV from sklearn as follows-

from sklearn.model\_selection import GridSearchCV

now to do the hyper parameter tuning for support vector-

selected parameters for the model and run to get best parameters, and then applied those shortlisted parameters to get result for the model, which is as follows-

* parameters={'C':[0.1,1,10,100,500],

'kernel':['poly','rbf','sigmoid']

}

* for the model the best parameters are-

**{'C': 10, 'kernel': 'rbf'}**

and with above Support Vector model is at 88% of accuracy.

Similarly did hyper parameter tuning for Random Forest model-

selected parameters for the model and run to get best parameters, and then applied those shortlisted parameters to get result for the model, which is as follows-

* parameters ={'n\_estimators':[100],

'max\_features':['auto','sqrt'],

'max\_depth':[4,5,6,7,8],

'criterion':['gini','entropy']}

* for the model the best parameters are-

**{'criterion': 'entropy',**

**'max\_depth': 8,**

**'max\_features': 'auto',**

**'n\_estimators': 100}**

And on above shortlisted parameters Random Forest model is working at 89% efficacy.

Saving The Model-

I have imported the ‘JobLib’ to save the model.

**Conclusion-**

The target data is binary classified, so after getting the dataset, performed data cleaning, formatting, feature correction and exploratory data analysis. The dataset checked for skewness, removed outliers with the help of z score, standardized the dataset with standard scaler and with the help of SMOTE did over sampling of target column, and now dataset is ready for machine learning model building.

Since problem statement is of the nature of binary classification, I applied the classification model, checked the accuracy scores and to avoid overfitting/underfitting, I also analyzed the cross-validation scores for each model. And then did hyperparameter tuning for selected models.

And finally concluded that in this case “Random Forest Classifier” is the more accurate model.

And with the help of this, we can achieve-

1. Reduction in number of fraud claims.
2. Lowers claim handling cost
3. More efficiently manage claim severity.
4. Detection of early claims in the claim life cycle is paramount to managing overall claims costs.

Hence, we can say that the use of machine learning approaches for insurance companies is very beneficial and efficient in the claim’s fraud detection process, and it is beneficial since-

1. All the claims which are not genuine or suspected can be detected using ML.
2. By machine learning we can give a featured structure to the dataset or can process data in a short interval of time.
3. Machine learning works better with an abundance of data or historical data. Machine Learning Technique will help to detect the similarities or differences between the multiple data behaviors. This would require proper model training to detectwhich transaction is fraud or genuine, then the machine automatically learns their behavior and puts the case in the bucket in which they fit correctly, and we can easily make predictions on the new or fresh transaction that either are fraud or genuine.
4. The main importance lies in efficiency. Since machines work faster than manual inspection done by human intervention. Then we can say that use of machine learning techniques in detecting frauds or potential fraud is a time saving and efficient approach.