**Auto Insurance Fraud claim Detection**

**by Machine Learning**

**Name-Astha Rai**

**Section-ML&AI**

**Roll no.-98**

**University roll no.-2015236**

**Introduction:-**

The goal of this project is to build a model that can detect auto insurance fraud. The challenge behind fraud detection in machine learning is that frauds are far less common as compared to legit insurance claims. This type of problems is known as imbalanced class classification.  
Frauds are unethical and are losses to the company. By building a model that can classify auto insurance fraud, I am able to cut losses for the insurance company. Less losses equates to more earning.

**Insurance Fraud:-**

Insurance fraud is an act committed to defraud an insurance process. It occurs when a claimant attempts to obtain some benefit or advantage they are not entitled to, or when an insurer knowingly denies some benefit that is due.

There are two types of Insurance Fraud:-

* Soft Insurance Fraud
* Hard Insurance Fraud

**Motivation:-**

* India is one of the biggest market for the insurance companies across the world.
* It is estimated that Indian insurance industry loses close to $6 billion insurance fraud in India.
* FBI which is also an insurance industry in USA,also estimates that the total cost of insurance fraud to be more than $40 billion annually.
* Hence the Insurance industry has an urgent need to develop capability that can help identify potential frauds with high degree of accuracy.
* The goal of this project is to build a model that can detect auto insurance fraud. The challenge behind fraud detection in machine learning is that frauds are far less common as compared to legit insurance claims.

**Executive Summary:-**

The main aim of this project was to build a machine learning model that could correctly classify insurance frauds.

Several models were tested with different methods of handling imbalance datasets. The top models were also fitted and tested with different ensembles.

The final fitted model is a weighted XGBoost which yielded an F1 score of 0.72 and a ROC AUC score of 0.84.

Prior to modelling, the data was cleaned and exploratory data analysis was conducted. After which, the data was pre-processed for the modelling. After modelling, the models were evaluated, and the best fitted model was selected using the F1 score and the ROC AUC score.

**Dataset used:-**

**Insurance\_claims.csv**

This dataset was labelled with n=1000 samples. It consist of 1000 auto incidents and auto insurance claims from Ohio, Illinois and Indiana from 01 January 2015 to 01 March 2015. Before any cleaning or feature engineering, the data set has a total of 39 variables.

**Steps involved:-**

* Data Cleaning
* Data Analysis
* Losses by Claims
* Pre-Processing
* Baseline Score
* Modelling
* Ensemble Models
* Final Model: Weighted XGBoost
* Evaluation

**Data Cleaning:-**

In the data cleaning, every features' percentage missing of missing values, number of unique values, and percentage of biggest category were considered.

There were date variables which where extracted as well. A new variable 'policy\_bind\_year was extracted from policy\_bind\_date using regex to reduce the number of categories the variable had.

**Data Analysis:-**

Exploratory data analysis was conducted started with the dependent variable, Fraud\_reported. There were 247 frauds and 753 non-frauds. 24.7% of the data were frauds while 75.3% were non-fraudulent claims.

**Losses by Claims:-**

I created a variable that measure how much claims minus how much premiums were paid by a client to indicate losses by claim. a positive will indicate a loss while a negative will be a profit. Every time a claim is more than the total premiums paid by a client; it is a loss for the insurance company.

**Pre-Processing:-**

The DV, fraud\_reported was coded 1 for fraud and 0 for non-fraud.Six interaction terms were created. Nominal variables were one-hot encoded, and the data set was split into 75% train and 25% test set, stratified on fraud reported.

**Baseline Score:-**

As our dataset is imbalance, accuracy is not a good measure of success. A high accuracy can be achieved by a poor model that only selects the majority class, hence, not detecting and measuring the accuracy of classifying the class of interest. In fact, predicting only the majority class will give an accuracy of 75%, specificity of 100% but a sensitivity of 0%.

**Modelling:-**

Hyperparameter tuning and selection was done for all the models using RandomizedSearch. Due to the number of parameters and models that were ran, RandomizedSearch is a faster more efficient choice as compared to gridsearch.

Five different classifiers were used in this project:

* Logistic regression
* K-nearest neighbors
* Random Forest
* XGBoost
* AdaBoost

**Ensemble Models:-**

Ensemble models in machine learning combine the decisions from multiple models to improve the overall performance and stability of the predictions.

Before ensembling, correlations of the predictions were ran. XGB, Random forest and AdaBoost have high correlation, perhaps as they all are CARTs (classification and regression tress).

The tree models selected are:

* Logistic regression with SMOTE (F1: 0.41)
* KNN with bootstrapping (F1:0.42)
* Weighted XGBoost (F1: 0.72)

**Final Model:-**

The final fitted model is the weighted XGBoost on the dataset with no oversampling.

**Weighted XGBoost**

XGBoost stands for extreme gradient boost. XGBoost is a form of gradient boosted CART.

The Gradient Boosting algorithm involves three elements:

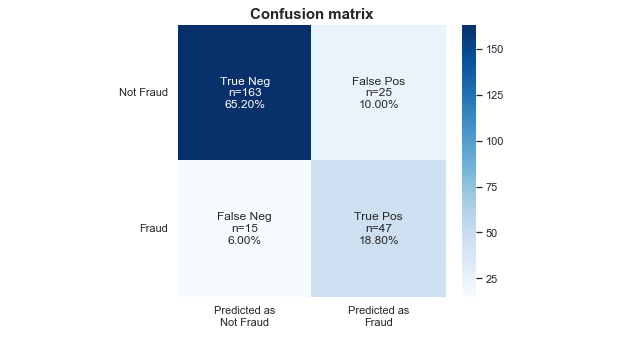
* A loss function to be optimized, such as cross entropy for classification or mean squared error for regression problems.
* A weak learner to make predictions, such as a greedily constructed decision tree.
* Additive model is used to add weak learners to minimize the loss function.

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

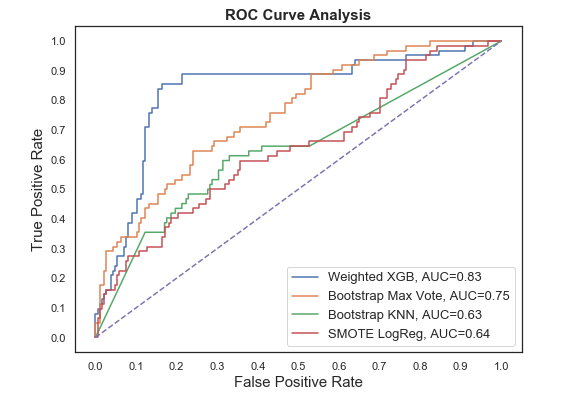
**Evaluation:-**

The number of cases for each class of the test set is shown in the confusion matrix. The y-axis shows the actual classes while the x-axis shows the predicted classes.

* True negative refers to non-fraud cases that are classified as non-fraud cases (161 cases, which makes up 64.40% of the test set's size).
* True positive refers to fraud cases that are correctly classified as fraud cases (50 cases, which makes up 20.00% of the test set's size).
* False negative are fraud cases that are classified as non-fraud cases (12 cases, which makes up 4.80% of the test set's size).
* False positive are non-fraud cases that are classified as fraud cases (27 cases, which makes up 10.80% of the test set's size).



The ROC curve below summarizes how well our model is at balancing between the true positive rate(sensitivity) and the false positive rate(1-specificity).



**Conclusion:-**

The best and final fitted model was a weighted XGBoost what yelled a F1 score of 0.72 and a ROC AUC of 0.84. The model's F1 score and ROC AUC scores were the highest amongst the other models. In conclusion, the model was able to correctly distinguish between fraud claims and legit claims with high accuracy.