**BLOG/ARTICLE**

Insurance Claims- Fraud Detection

**Abstract** — Fraud detection is an important area of research in the Auto Insurance Companies due to its financial consequences arising mainly from investigation costs, revenue losses, and reputational risk. To solve this problem, most of the companies adopt Machine Learning based fraud detection models. Efficient fraud detection models improve the performance Insurance claim settlement. Key challenges in building an efficient fraud detection model include

 Data imbalance: skewed number of lesser fraudulent cases in comparison to the non-fraudulent cases,

 Selection of classification model: use of appropriate machine learning or deep learning models to identify fraud or non-fraud cases

In this work, we have used data-imbalance techniques and four classification models to meet these challenges. The performance of these models was measured using various metrics such as accuracy, sensitivity, specificity, F1-score and AUC ROC Curve

Code is available in the following link: <https://github.com/SharvariAK/EvaluationProject.git>

**Introduction/Problem Definition**

The dataset is a Business Case dataset in the Insurance Industry.  
Insurance fraud is a huge problem in the industry. It is difficult to identify fraud claims, here we can use Machine Learning to solve this problem.

In this project, we are provided a dataset which has the details of the insurance policy along with the customer details. It also has the details of the accident on the basis of which the claims have been made.

In this example, we will be working with some auto insurance data to demonstrate how we can create a predictive model that predicts if an insurance claim is fraudulent or not.

**Data Analysis**

It is a dataset of shape 4996 rows and 40 columns.

It is a mixed data of continuous and categorical data having null values in only one column i.e \_c39.

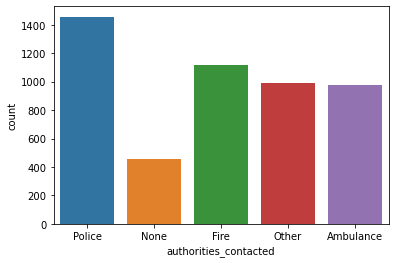
**Independent variable**:-months as customer, 'age', 'policy number', 'policy bind date', 'policy state', 'policy csl', 'policy deductable', 'policy annual premium', 'umbrella limit', 'insured zip', 'insured sex', 'insured education level', 'insured occupation', 'insured hobbies', 'insured relationship', 'capital-gains', 'capital-loss', 'incident date', 'incident type', 'collision type', 'incident severity',' authorities contacted', 'incident state', 'incident\_city',ncident\_location','incident\_hour\_of\_the\_day','number\_of\_vehicles\_involved', 'property damage', 'bodily injuries', 'witnesses', 'police report available', 'total claim amount', 'injury claim', 'property claim', 'vehicle claim', 'auto make', auto model', 'auto year', , '\_c39'

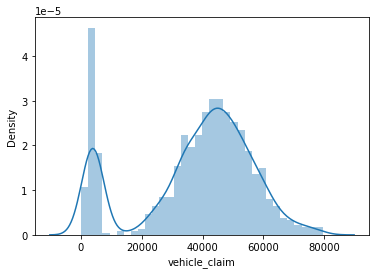
**Dependent Variable :**- fraud reported

**Exploratory Data Analysis Concluding Remark**

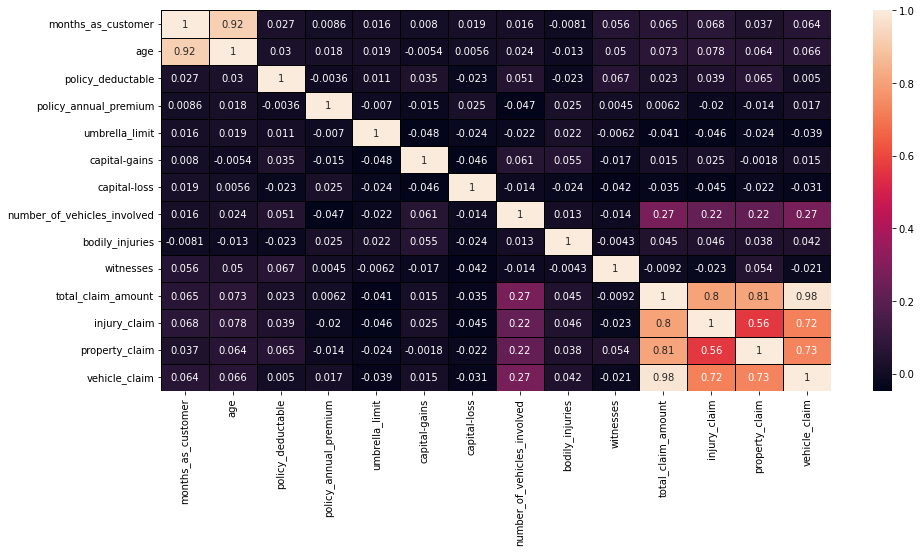
We will plot count plot for categorical variables and kde plot and distribution plot for continuous variables. We plotted for each and every variables.

For ex.





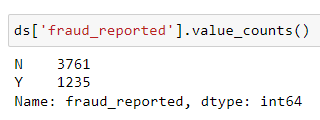
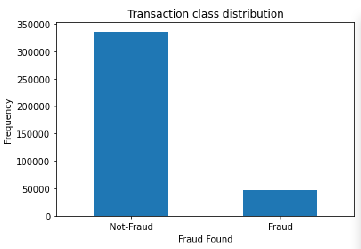
**HISTOGRAM:**



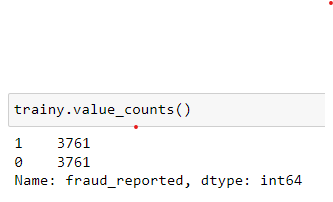
The fig.shows the histogram of the dataset for various numerical fields in the dataset.

**SMOTE Technique**

The Target variable is showing imbalance this will affect our accuracy, hence we will SMOTE technique to balance the data.

****

After applying SMOTE technique to balance the variable for better accuracy.

****

**PCA Technique**

As there are more number of columns i.e 40 we will reduce it to 10 columns with the help of PCA technique

**Removing Unnecessary Columns:**

ds=ds.drop(['policy\_number','insured\_zip','policy\_bind\_date','incident\_date','incident\_location','auto\_year','incident\_hour\_of\_the\_day'],axis=1)

**Pre-Processing Pipeline**

Dataset

Simplified data set

Classification Models

Performance

Metrics

* Sensitivity
* Specificity
* Precision
* Accuracy
* F1 Score
* AUC ROC CURVE
* Saving Model
* Feature Engineering
* Handling Missing Value
* Encoding

Train Test

**Building Machine Learning Models.**

As it is a predictive type model in a yes or no situation we will use Logistic Regression.

Other classifiers used:-

Random Forest Classifier

Decision Tree Classifier

Support Vector Classifier

Table Content:Performance of the Model using various classifying Models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Models** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| Logistic Regression. | o.6066 | 0.61 | 0.59 | 0.60 |
| Random Forest Classifier | 0.9940 | 0.99 | 1.00 | 0.99 |
| Decision Tree Classifier | 0.9906 | 0.99 | 0.99 | 0.99 |
| Support Vector Classifier | 0.5249 | 0.52 | 0.82 | 0.63 |

**Hyperparameter Tuning**: -For Increasing accuracy score we will use Hyperparameter Tuning

**Concluding Remarks.**

We will use Random Forest Classifier as our final model as it is giving maximum accuracy for testing and training data.

AUC ROC Curve is plot between True Positive Rate and False Positive Rate which is giving maximum area coverage which tells the model is working well.

At last, we will save the model with the help of joblib technique.

**AUC ROC Curve**

