**Insurance Claims - Fraud Detection**

**Problem Definition**

Fraud is one of the critical issues of the insurance industry. The meaning of fraud in the insurance industry is to knowingly make a fake claim, exaggerating or adding items to it, or in any way being fraudulent to gain more than legal entity. Identifying these fraud claims is difficult which causes huge losses. Considering the huge amount of data, the Insurance industry is growing rapidly at a very fast rate. The conventional way of identifying the fraudulent claims will not work as the data size increases. It will become challenging to predict these claims as the new types of claims arise.

Machine Learning is in a unique position to help the Auto Insurance industry with this problem. The task is to create a predictive model that predicts if an insurance claim is fraudulent or not and create an auto insurance fraud detective model. Frauds are corrupt due to which companies face massive losses. To cut out these losses for the Insurance industries, we will build a Machine Learning model that helps in classifying auto insurance fraud. Machine Learning techniques allow us to enhance predictive accuracy, permitting the loss controls to accomplish higher scope with low false positive rates.

**Dataset Information**

The dataset consists of 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. Here, we are working with some auto insurance data to demonstrate a predictive model that predicts if an insurance claim is fraudulent or not. We have trained and built a Machine Learning model to help the Auto Insurance industry with this problem.

The dataset comprises 1000 rows and 40 columns in total including the output variable. “fraud\_reported” is the target variable in this example.

**Data Analysis**

**Data Collection**

The dataset consists of auto insurance claims from Ohio, Illinois and Indiana dated from 01/01/2015 to 01/03/2015. This information has been retrieved from an online data source (github.com). The data received was in the form of a csv file. Before performing any operations or transformations (feature engineering), the dataset comprises a total 39 input features and a target variable. We are not sure of the fact whether the data is of a single company or multiple insurance companies though the con is that the data sample is of a small size. Hence, we begin working with small datasets and create big ones as time goes on to succeed in order to profit a company’s revenue.

**Cleaning and Data Preparation**

Success rate in Machine Learning is achieved depending on the way the data is represented. Feature Engineering is a part of data cleaning and preparation which transforms raw data into variables that better the performance of the model to give accurate predictions. We identify the features relevant to finding the predictions as that of the actual output.

In this case, we have featured “policy\_number’ as the index variable and split variable “policy\_bind\_day” to retrieve “policy\_bind\_date”, “policy\_bind\_month” and “policy\_bind\_year” from the existing data to make the model understand data better. Feature “\_c39” consisted of all missing values that served no value to the dataset, hence the variable was deleted. Few uncertain “?” values found in the dataset were replaced with NaN values to fill the missing values using the “most\_frequent” strategy of the SimpleImputer() function. Encoding all string values using OrdinalEncoder() to convert it to floating values. Therefore, all the feature engineering procedures were used to get rid of the corrupt data and prepare a dataset with useful information.

**Data Analysis Using Visualizations**

Dividing information, understanding its patterns and correlations is the most significant part of analysing and preparing data. After this step, ML algorithms are applied for training the model to get predictions. For this dataset, we have divided data into two dataframes on the basis of the data types of its variables i.e. object or numeric data type.

**Visualization of the String Values**

For the string data, a countplot graph was used as it gives the frequency of the columns. Observations made while analysing data are -

* Female customers are more than the male customers.
* More than 700 people have been reported with No Fraud but around 250 People have been reported with Fraud.
* 300 customer's property has been damaged and 700 customer's property has not been damaged.
* Around 320 customer's police reports are available and 680 customer's police reports are not available.
* Incidents are high in states like SC, NY and WV.
* Rear Collision is the max collision type.
* The incident severity is Minor damage in most of the cases.
* Police authorities have been contacted the most.
* There are too many categories in columns like “auto\_model”, “auto\_make”, “incident\_location”, “incident\_date”, “insured\_occupation” & “insured\_hobies”. Hence, it is difficult to conclude any observation out of these columns.

**Visualization of Int/Float Values**

We have used distplot and scatter plot graphs to understand data with int/float values. Observations made are -

* Variables “policy\_bind\_year”, “policy\_bind\_month”, “policy\_bind\_day”, “auto\_ year” & “incident\_hour\_of\_the\_day” are showing values at a constant rate.
* Variable “umbrella\_limit” consists of a maximum number of 0 values.
* We could observe negative values in the “capital-loss” feature.
* “Capital-gains” consists of values ranging till positive 120000.
* “Capital-loss” consists of values ranging till negative 130000.
* Variables like “number\_of\_vehicles\_involved”, “bodily\_injuries” & “witnesses” have a nominal/categorical type of data.

**EDA Concluding Remarks**

Performing complete EDA on the data (cleaning, integrating and transforming of data), we get a dataset with 1000 rows and 40 columns. Concluding observations are -

* The standard deviation of the variables in the dataset are very huge which means that the values in these columns are largely scattered and are not near to the mean values. They are very far away from their mean values.
* The values inside the dataset ranges from high negative values to high positive values. The value ranges are very high within the dataset.
* The min & max values in every feature have huge range differences.
* Understanding data properly is difficult due to the huge number of columns.
* The most negatively correlated variable to the target variable is “incident\_severity”.
* The most positively correlated variable to the target variable is “vehicle\_claim”.
* High positive correlation was not observed in the data with respect to the target variable.
* There are 22 features which are positively correlated with the Target Variable.
* There are 17 features which are negatively correlated with the Target Variable.

**Pre-Processing Pipeline**

**Skewness Correction**

Considering threshold value as +/-0.5 as the range for skewness, we could see skewness in features such as “umbrella\_limit”, “insured\_zip”, “property\_damage”, “police\_report\_available”, “total\_claim\_amount”, “vehicle\_claim” & “fraud\_reported”.

The skewness for the required columns were resolved using power\_transform function.

**Outliers Detection and Correction**

Maximum outliers were observed in the “umbrella\_limit” feature and in few other variables as well. These outliers were resolved using the z-score technique. The information loss after the removal of outliers was found to be 2% which is not much of a huge information loss. The shape of the dataset before outliers removal was 1000 rows & 40 columns and that after removal came out to be 980 rows & 40 columns.

**Normalization**

As the values in the dataset have high ranges, it becomes complex for an ML model to understand and read data, hence data training becomes difficult which is not a proper way to deal with data to achieve good accuracy and get accurate predictions. Therefore, it is very important to normalize/standardize data which means getting data within a certain range to have proper understanding of data. In this example, we have used StandardScaler() technique to normalize the data which brings data between the range of 0 to 1.

**Applying PCA on data**

Translating the given data from higher dimension into lower dimension is called PCA (Principle Component Analysis). It is used to reduce the number of attributes and select the attributes which contain the maximum information (here, it is 98% of variance). It is helpful to use PCA (dimensionality-reduction technique) before using ML algorithms as fewer the attributes, less will be the attributes for feature selection for performing iterations and hence leads to faster computations.

**Building Machine Learning Models**

**Logistic Regression**

Logistic Regression is a supervised learning algorithm. It is a predictive regression analysis which is conducted when there are binary values in the target variable. The f1 score, precision and recall for this algorithm was found to be at 100% accuracy.

**Decision Tree Classifier**

Visualisation view of the Decision Tree is in the form of graphs. Decision tree divides the main data set into a subset of trees that consists of choices and results. Node of each tree depicts a choice and the edges depicts the decision. The main dataset is categorised into 80% training data and 20% testing data. A model is built with a training dataset which predicts the accuracy. This model is applied on the test data and the predicted accuracy is validated. The model gives us a prediction in the form of a Yes/No for this data.

**Cross Validation** technique was applied on both Logistic Regression and Decision Tree Classifier algorithms to overcome the issues of overfitting and underfitting. The mean cross validation score was obtained to compare both the algorithms and check which algorithm works best and gives highest accuracy for the given data. Since both algorithms predicted 100% accuracy, Logistic Regression was used for further hypertuning of data.

Use of **GridSearchCV** techniquewas made to hypertune the algorithm with the best found parameters i.e. “max\_iter” at 1 and “penalty” as l1 to achieve the best accuracy.

**Concluding Remarks**

* The f1 score, precision and recall for both Logistic Regression and Decision Tree Classifier is 100%. Hence, the accuracy score for both the algorithms is also 100%.
* Comparing the cv score and accuracy score of both algorithms, the accuracy achieved was 100%. Therefore, Logistic Regression was selected as the model.
* Performing hyperparameter tuning on the selected algorithm with the best found parameters using the GridSearchCV, the predicted accuracy comes out to be 100%.
* Hence, we have achieved 100% accuracy for the model after trying different algorithms and different testing techniques on the data.