**ML-PROJECT BLOG**

**A diagram of machine learning

Description automatically generated**

**Name:** BALAKRISHNAN.N -DS2309

**Blog/Article:** Project on Insurance Fraud Detection

**Course:** P.G. Program in Data Science

**Internship:**Flip Robo Technologies

**Institute:** Data Trained Institution

# PROJECT: INSURANCE CLAIM FRAUD DETECTION USING MACHINE LEARNING



**PROBLEM IDENTIFICATION AND TARGET**

**Problem Statement:**

Insurance fraud is a huge problem in the industry. It's difficult to identify fraud claims. Machine Learning is in a unique position to help the Auto Insurance industry with this problem.

In this project, you 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.

**Target**

Demonstrate how you can create a predictive model that predicts if an insurance claim is fraudulent or not.

# ABSTRACT

Insurance fraud is a deliberate deception perpetrated against or by an insurance company or agent for financial gain. Fraud may be committed at different points by applicants, policyholders, third-party claimants, or professionals who provide services to claimants. Insurance agents and company employees may also commit insurance fraud. Common frauds include "padding" (inflating claims), misrepresenting facts on an insurance application, submitting claims for injuries or damage that never occurred, and staging accidents.

People who commit insurance fraud include:

* organized criminals who steal large sums through fraudulent business activities,
* professionals and technicians who inflate service costs or charge for services not rendered, and
* Ordinary people who want to cover their deductible or view filing a claim as an opportunity to make a little money.

Some insurance lines are more vulnerable to fraud than others. Healthcare, workers' compensation, and auto are generally considered the most affected insurance sectors.

# PROCESS FOLLOWED AND STEPS

* + Exploratory data analysis
  + Feature Engineering
  + Relation between different features/amenities of house and target column

i.e. sales price.

* + Correlation and identification of multicollinearity problem.
  + Identifying and Selecting best features that affect sales price.
  + Data pre-processing.
  + Model Initialization and evaluation.
  + Model Validation
  + Model Testing and Pipelining.
  + Prediction of test dataset values.

# IMPORTING LIBRARIES:

***Common libraries***

* import numpy as np
* import pandas as pd
* import matplotlib.pyplot as plt
* import seaborn as sns

***Libraries for splitting training and testing data Hyper parameter tuning***

* from sklearn.model\_selection import train\_test\_split,GridSearchCV

***Importing Algorithms***

* from sklearn.linear\_model import LogisticRegression
* from sklearn.tree import DecisionTreeClassifier
* from sklearn.ensemble import RandomForestClassifier
* from sklearn.metrics import confusion\_matrix
* from sklearn.neighbors import KNeighborsClassifier
* from sklearn.svm import SVC
* from sklearn.naive\_bayes import GaussianNB

***Importing metrics***

* from sklearn.metrics import accuracy\_score,confusion\_matrix,classification\_report
* from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

***Removing warnings***

import warnings warnings.filterwarnings('ignore')

%matplotlib inline

# DATASET DESCRIPTION SUMMARY AND LOADING DATASET

The dataset 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.

**File location:**

file\_loc= "https://raw.githubusercontent.com/dsrscientist/Data-Science-ML-Capstone-Projects/master/Automobile\_insurance\_fraud.csv"

**Reading CSV file:**

df = pd.read\_csv(file\_loc)

**FEATURE ENGINEERING**

**What is it and how we’ve used it?**

Feature engineering or feature extraction or feature discovery is the process of using domain knowledge to extract features from raw data.

As the date column in the data set is mentioned in ddmmyy format so we are splitting the date into days months and year and then checking the insurance claims on the basis of these individual days months and year just check if there is any particular relation in between them and the target column.

A screenshot of a computer program

Description automatically generated

# DATA ANALYSIS

As preliminary data analysis we are first inspecting the first and last 10 rows of the data set to check the different column values and row entries.

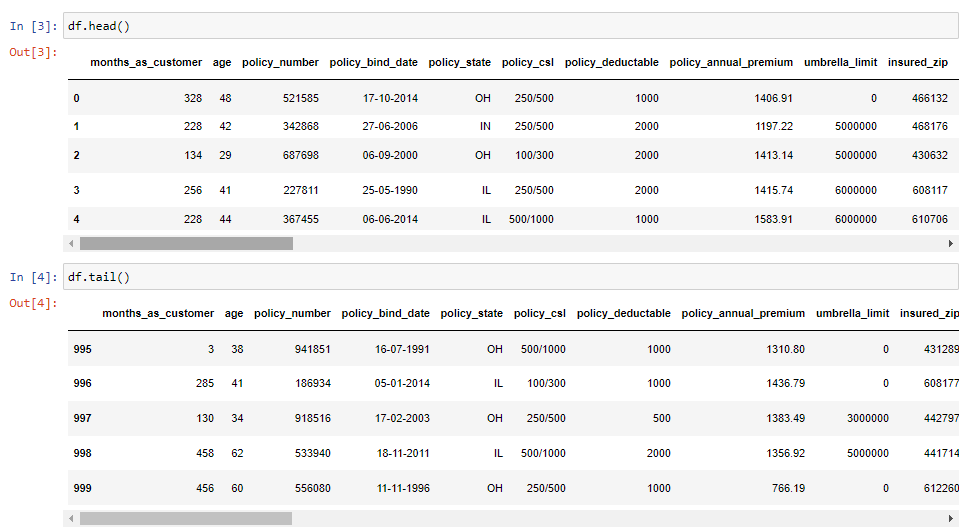
Using head and tail method dtypes method and isnal.sum method we are inspecting data types of columns and if null values are present in the columns.

As we can see that there are no null values in the data set that is a good sign but further we need to check the statistical description of the data set and we need to inspect the presence of outliers in the data set.

Segregating the categorical data types and numerical data types is a good idea and will help us in doing exploratory data analysis on the basis of data types of columns.

As categorical columns and numerical columns need to be inspected and analysed with different approaches.

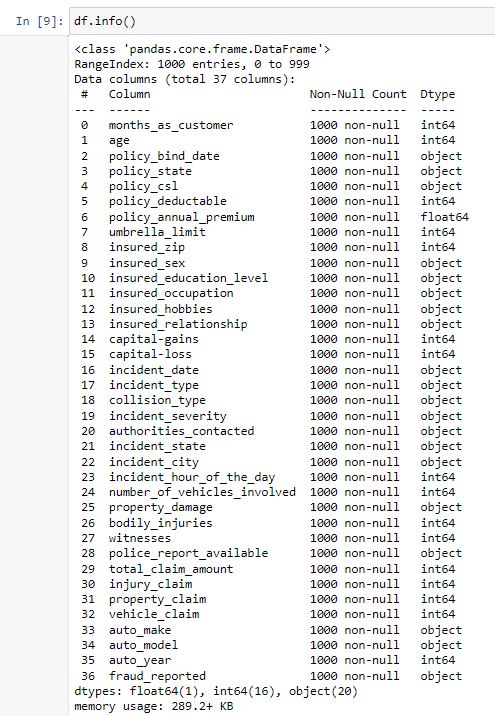
* First and last 5 rows:



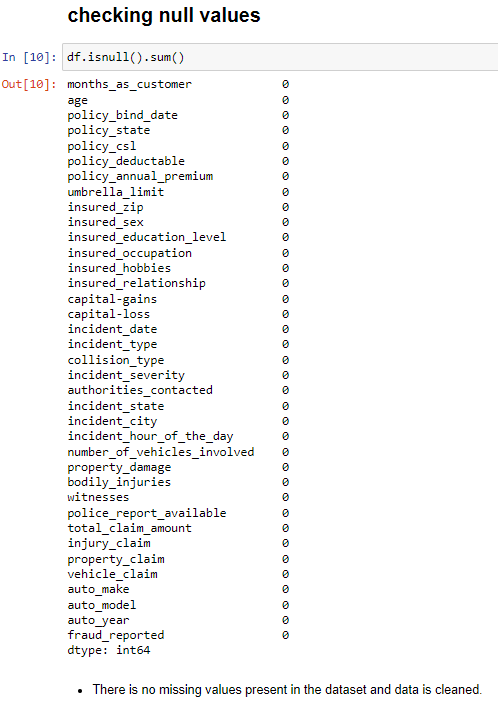
* Rows and columns in dataset: (1000, 40)
* Column names:

['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', 'incident\_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', 'fraud\_reported', '\_c39']

* + **Information of columns**:

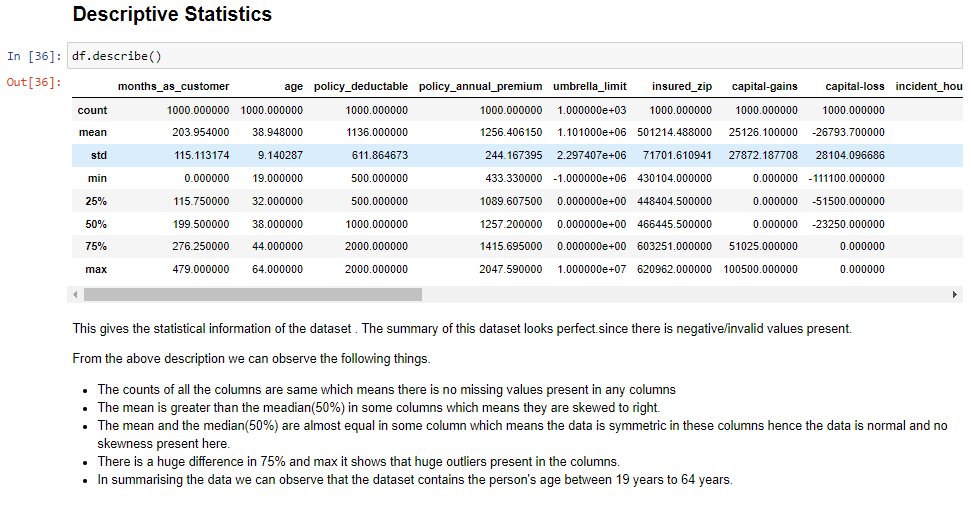


* **Null values in each column:**

****

# EXPLORATORY DATA ANALYSIS

* **Summary statistics of numerical columns:**

****

* **Summary statistics of categorical columns:**

A screenshot of a computer

Description automatically generated

* **Selecting categorical and numerical columns:**

A screenshot of a computer code

Description automatically generated

**Data Distribution Analysis:**

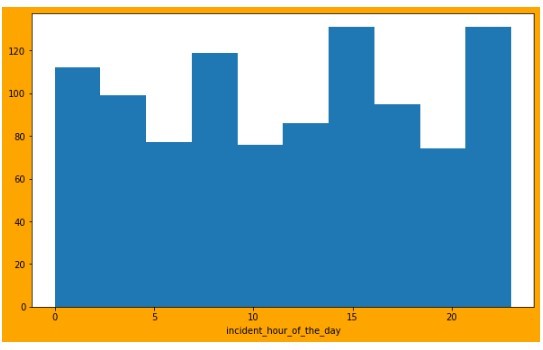
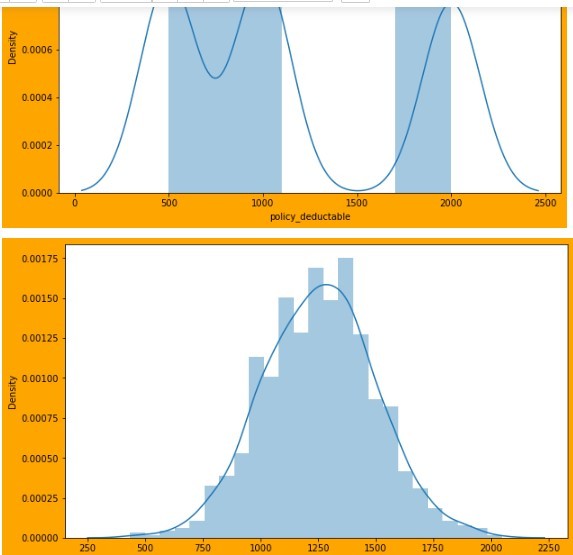
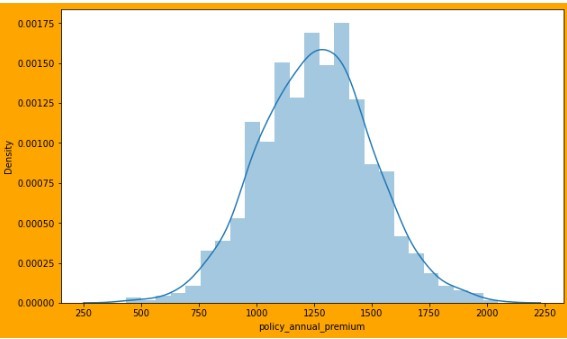
Data distribution of the data set is very important to determine the frequency distribution of the numerical columns it also helps us to analyze the data distribution of columns and find the skewness if it is present in the data set.

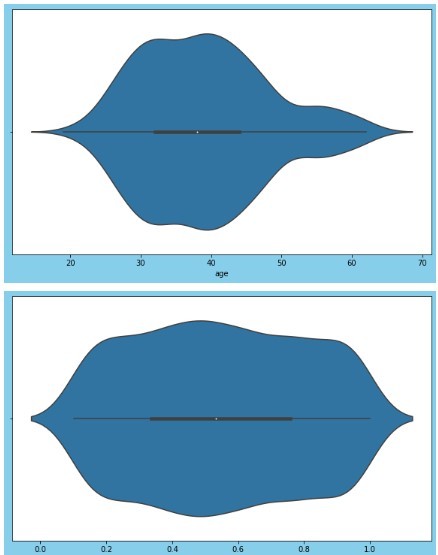
Here we have used histograms density plots box plots distribution plots and violent plots to analyze the data distribution and frequency of the data set at large.

Skewness can lead to large error values does they need to be figured out and further they need to be treated.

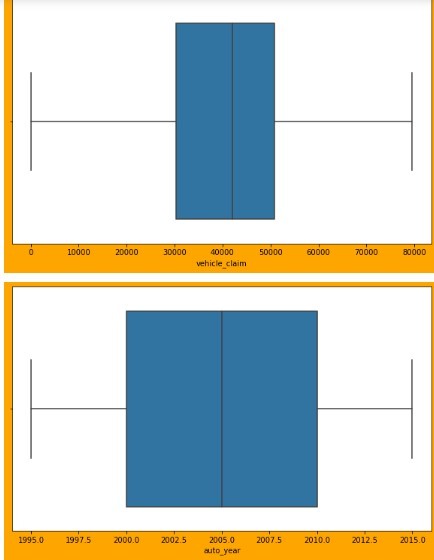
A white screen with a white text box

Description automatically generated with medium confidence

1. Histogram.
2. KdePlot
3. Distribution Plot
4. Violin plot.



**Outlier Analysis using BoxPlot**



# UNIVARIATE ANALYSIS

**What is it and how we’ve used it?**

Univariate Analysis is a type of data visualization where we visualize only a single variable at a time. Univariate Analysis helps us to analyze the distribution of the variable present in the data so that we can perform further analysis.

All data distribution plots belong to analysing only one column at a time apart from them we have used count plot line plot and scatter plot strip plot and swarmplot to analyse one column at a time.

1. Count plot
2. Lineplot
3. Scatterplot

**BIVARIATE ANALYSIS:**

**What is it and how we’ve used it?**

Bivariate analysis is the simultaneous analysis of two variables. It explores the concept of the relationship between two variable whether there exists an association and the strength of this association or whether there are differences between two variables and the significance of these differences.

For analysing two columns at a time we have used categorical plots and box plot with you value as whether fraud reported or not to check indepth of frequencies for different columns at which fraud was reported or not.

A graph with different colored bars

Description automatically generated

**Multivariate Analysis**

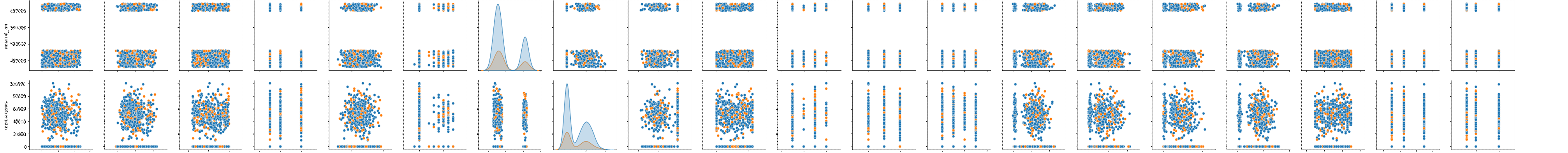
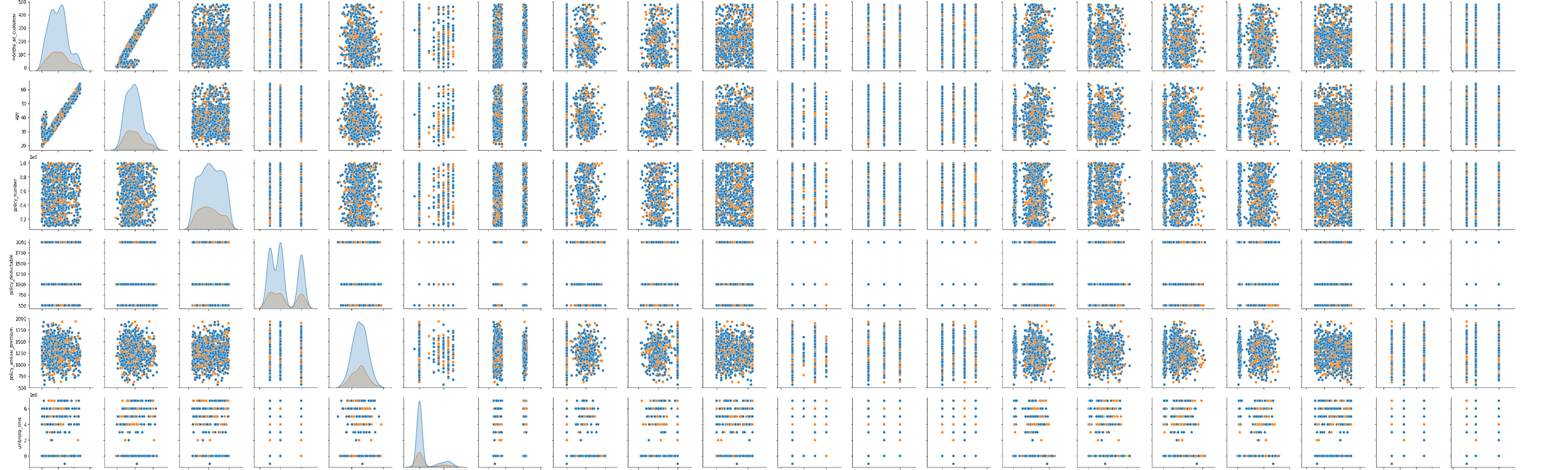
What is it and how we’ve used it?

It is an extension of bivariate analysis which means it involves multiple variables at the same time to find correlation between them. Multivariate Analysis is a set of statistical model that examine patterns in multidimensional data by considering at once, several data variable.

We have done multivariate analysis using pair plot and correlation plot such as heat

maps to understand the correlation between different columns and to check multi

collinearity problem as well.

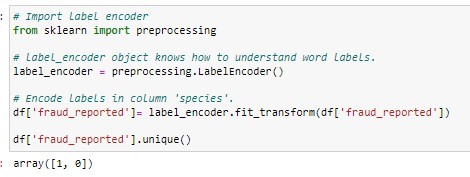


**Preprocessing**

**What is it and how we’ve used it?**

Label Encoding refers to **converting the labels into a numeric form so as to convert them into the machine-readable form**. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

Using label encoder to encode label column:



Removing outliers values using **Z-Score method**

**What is it and how we’ve used it?**

Z-score tells how many standard deviations away a given observation is from the mean. For example, a Z score of 2.5 means that the data point is 2.5 standard deviation far from the mean. And since it is far from the center, it's flagged as an outlier/anomaly.

A screenshot of a computer

Description automatically generated

Here we have chosen absolute z score value equal to 2.8 so all the values lying in 2.8 times standard deviation will be removed.

**Checking correlation and addressing Multicollinearity problem:**

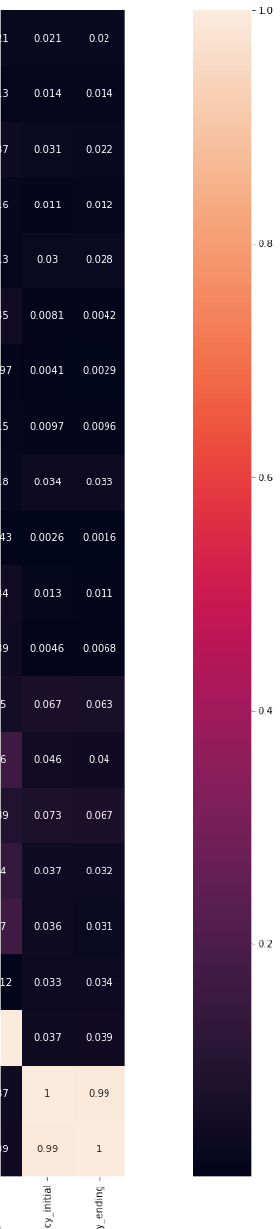
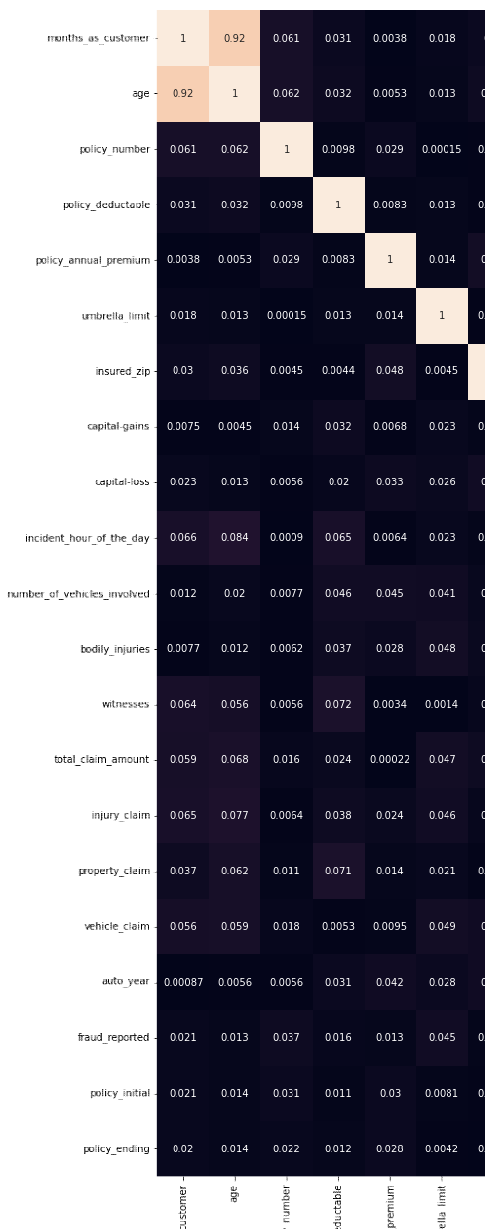
**What is it and how we’ve used it?**

**Heatmap analysis:**

A Heat map is a graphical representation of multivariate data that is structured as a matrix of columns and rows.

Heat maps are very useful in describing correlation among several numerical variables, visualizing patterns and anomalies.

Using heat map the correlation values of different columns show that vehicle claim property claim injury claim and total claim amount columns are highly correlated with each other thus it would be necessary to use variance inflation factor that will help us to measure the amount of multi collinearity in the data set columns.



**Multicollinearity check using variance Inflation Factor:**

**V**

**What is it and how we’ve used it?**

**A variance inflation factor (VI ) is** a measure of the amount of multicollinearity in

**F**

regression analysis**. Multicollinearity exists when there is a correlation between**

**n**

**multiple independent variables in a multiple regression model. This can adversely affect the regression results.**

Vif values of different columns show that the multi collinearity problem exist in the data set that needs to be corrected before further model initialization and testing.

Two eliminate the multicollinearity problem in the data prep processing stage we have used power transformation and we have done principle component analysis to select the best principal components for predicting the target column that is whether fraud was reported or not in a particular insurance claim.

A screenshot of a computer

Description automatically generated

Using correlation function we can see that columns namely umbrella limit vehicles involved bodily injuries witnesses claim amount injury claim property claim and vehicle claim are positively correlated with the target column that is fraud reported.

**Feature Importance**

**Feature (variable) importance** indicates how much each feature contributes to the model prediction**. Basically, it determines the degree of usefulness of a specific variable for a current model and prediction.**

**A screenshot of a computer

Description automatically generated**

In building this model we have followed two approaches with variations in:

1. Outlier removal.
2. PCA selection.
3. Variation in train test split size.
4. Opting Oversampling and Undersampling techniques.

**Feature Selection and Data Preprocessing**

* 1. **Feature and Label Selection:**

A screenshot of a computer

Description automatically generated

* 1. **Scaling Data using Power Transformer:**

In statistics, a power transform is a family of functions applied to create a monotonic transformation of data using power functions. It is a data transformation technique used to stabilize variance, make the data more normal distribution-like, improve the validity of measures of association (such as the Pearson correlation between variables), and for other data stabilization procedures.

A screenshot of a computer

Description automatically generated

**Analyzing Principal component using PCA:**

**What is it and how we’ve used it?**

Principal component analysis (PCA) is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss. It does so by creating new uncorrelated variables that successively maximize variance.

**Splitting Features and Labels into training and testing dataset:**

**What is it and how we’ve used it?**

The train\_test\_split() method is used to split our data into train and test sets. First, we need to divide our data into features (X) and labels (y). The dataframe gets divided into X\_train,X\_test , y\_train and y\_test. X\_train and y\_train sets are used for training and fitting the model.

Further we are splitting our data set and we are keeping 30% data result for the testing part and 70% data for the training part.

A screenshot of a computer program

Description automatically generated

**Treating imbalanced class using SMOTE and Near Miss methods:**

**What is it and how we’ve used it?**

SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem.

It aims to balance class distribution by randomly increasing minority class examples by replicating them.

By checking the unique value counts of the target column we can see that the values in the target column are categorical in nature and does they need to be and encoded. Does we are using label and coder to encode the label column that's simply converts the categorical value is to numerical values.

By checking the value counts of each class in label we can see that there are 5004 value counts for class where no fraud is reported. But we have only 178 value counts for which fraud is reported and this weekend conclude that the data set is imbalanced and does we need to correct this imbalance data set with the help of over sampling technique. That's we are using smoke that is synthetic minority over sampling technique with the help of which we will balance the data set.

A screenshot of a computer

Description automatically generated

**Building Machine Learning Models and Model Performance**

Thus after splitting the data set and balancing the data set we are proceeding towards model initialization part and further we will train and test our model and we will check the accuracy of our model and then we will select the model that is giving us the best accuracy further we will also plot the aucroc curve to evaluate the performance in classification models.

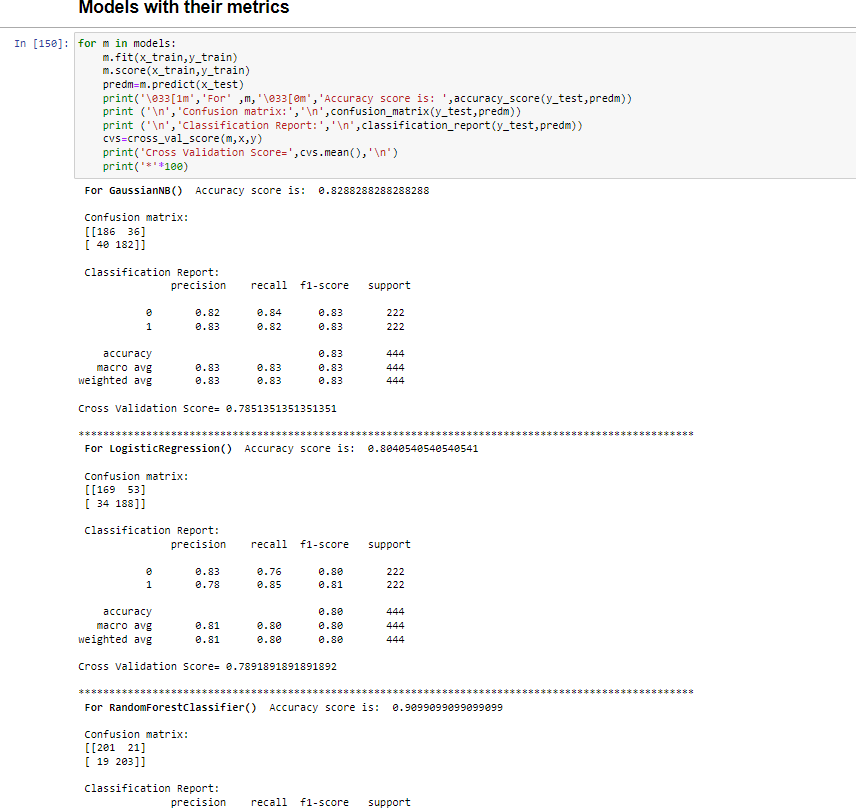
In all this will be initializing logistic regression model decision tree classifier model random forest classifier model k neighbors classifier model support vector classifier model gradient boosting classifier model at a post classifier model and nav base classifier model.

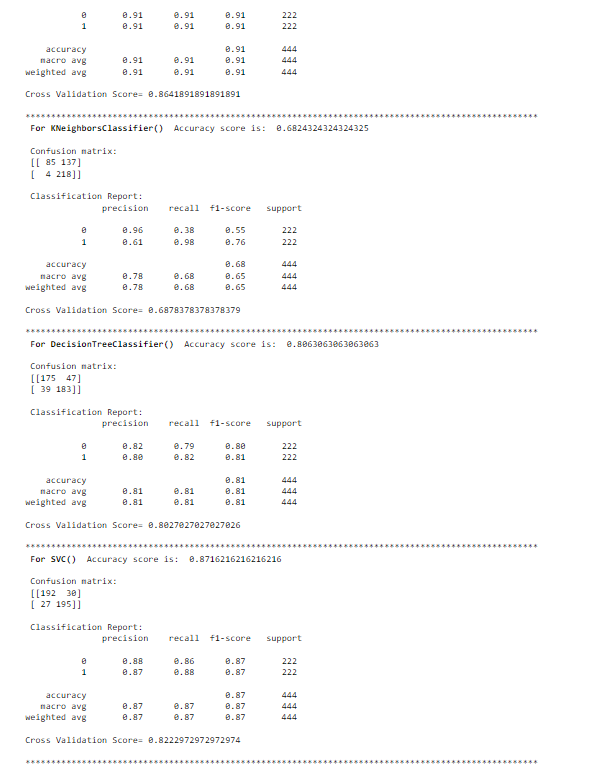
We will also do hyperparameter tuning of all these models and will do the cross validation using k folds method.

Screenshots of all initialized models are indicated below:

**Different modes are built to classify Insurance cases being fraudulent or genuine:**

1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest Classifier
4. K-Neighbours Classifier
5. Support Vector Classifier
6. Gradient Boosting Classifier
7. GaussianNB



****

**Hyperparameter tuning**

For Cross Validation using K-Fold method:

**What is it and how we’ve used it?**

Cross-validation is usually used in machine learning for improving model prediction when we don’t have enough data

k-fold cross-validation; we first shuffle our dataset so the order of the inputs and outputs are completely random. We do this step to make sure that our inputs are not biased in any way. Then, we split the dataset into *k* parts of equal sizes.

**Hyperparameter tuning using:**

1. GridSearchCV (**What is it and how we’ve used it?)**

GridSearchCV **is a technique for finding the optimal parameter values from a given set of parameters in a grid. It's essentially a cross-validation technique.**

1. Randomized Search CV (**What is it and how we’ve used it?)**

**Random search is** a technique where random combinations of the hyperparameters are used to find the best solution for the built model**.**

**Hyperparameter tuning for Best model**

**A screenshot of a computer program

Description automatically generated**

A screenshot of a computer code

Description automatically generated

**Model Evaluation and Final Result**

After performing the hyper parameter tuning of all the models we can conclude that the model that is performing the best is the logistic regression model. In it we have defined

a Param grid with L1 and l2 penalty values and different c values with different Max iterations. Further will be using grade search CV that will be fitting three folds for each of 160 candidates totally 480 fits and the best estimators that we have got is the C value equal to 1438.449 with random state equal to 1.

**Model Evaluation using Auc Roc curves:**

**What is it and how we’ve used it?**

AUC-ROC is the valued metric used for evaluating the performance in classification models. The AUC-ROC metric clearly helps determine and tell us about the capability of a model in distinguishing the classes. The judging criteria being - Higher the AUC, better the model. AUC-ROC curves are frequently used to depict in a graphical way the connection and trade-off between sensitivity and specificity for every possible cut-off for a test being performed or a combination of tests being performed. The area under the ROC curve gives an idea about the benefit of using the test for the underlying question. AUC - ROC curves are also a performance measurement for the classification problems at various threshold settings.

All the models are performing well but for this particular case logistic regression model performed the best.

We will conclude our search for the best model by plotting ROC auc curve.

A screen shot of a graph

Description automatically generated

Finding Best random state for our model:

A screenshot of a computer

Description automatically generated

**Understanding Final Result parameters:**

**Confusion matrix**

A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of

the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa – both variants are found in the literature.

The name stems from the fact that it makes it easy to see whether the system is confusing two classes (i.e. commonly mislabeling one as another).

**Classification Report**

What does a classification report show?

A Classification report is used to **measure the quality of predictions from a classification algorithm**. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report as shown below.

**Various elements of classification report:**

Precision — *What percent of your predictions were correct?* Recall — *What percent of the positive cases did you catch?* F1-score — *The weighted average of Precision and Recall* support —*The number of occurrences of each given class*

Let’s understand them in depth:

The report shows the main classification metrics precision, recall and f1-score on a per- class basis. The metrics are calculated by using true and false positives, true and false negatives. Positive and negative in this case are generic names for the predicted classes. There are four ways to check if the predictions are right or wrong:

1. **TN / True Negative:** when a case was negative and predicted negative
2. **TP / True Positive:** when a case was positive and predicted positive
3. **FN / False Negative:** when a case was positive but predicted negative
4. **FP / False Positive:** when a case was negative but predicted positive

**Precision – What percent of your predictions were correct?**

Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives.

**TP – True Positives FP – False Positives**

**Precision – Accuracy of positive predictions. Precision = TP/(TP + FP)**

from sklearn.metrics import precision\_score print("Precision score:

{}".format(precision\_score(y\_true,y\_pred)))

**Recall – What percent of the positive cases did you catch?**

Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.

**FN – False Negatives**

Recall: Fraction of positives that were correctly identified. Recall = TP/(TP+FN)

from sklearn.metrics import recall\_score print("Recall score:

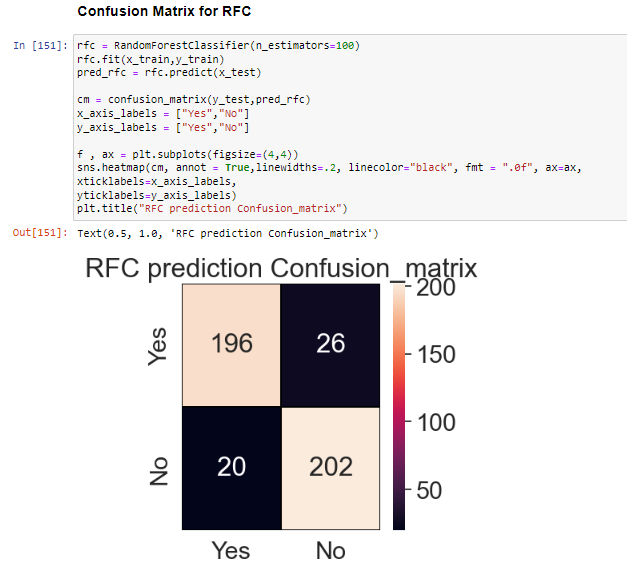
{}".format(recall\_score(y\_true,y\_pred)))

**F1 score – What percent of positive predictions were correct?**

The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy. **F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)**

Accuracy score of our logistics regression model on the training dataset is coming out to be 100 percent and accuracy score of our model on test dataset is 91 %.

With a precision of 91 % our model was able to predict whether an insurance claim was legit

****

**Pipelining**

**What is it and how we’ve used it?**

A Machine Learning pipeline is a process of automating the workflow of a complete machine learning task***.*** It can be done by enabling a sequence of data to be transformed and correlated together in a model that can be analyzed to get the output. A typical pipeline includes raw data input, features, outputs, model parameters, ML models, and Predictions. Moreover, an ML Pipeline contains multiple sequential steps that perform everything ranging from data extraction and pre-processing to model training and deployment in Machine learning in a modular approach. It means that in the pipeline, each step is designed as an independent module, and all these modules are tied together to get the final result.

**Save the prediction file.**

**A screenshot of a computer program

Description automatically generated**

**Remarks**

In this project we build have built a classification model that detects and insurance fraud. This model can be used to detect the frauds and does it can prevent and reduce losses in insurance companies.

The major challenge in this data set and fraud detection is that because of imbalanced data set the values of class level category representing fraud detected were very less as compared to no fraud detected values.

We developed 7 class fire models mainly logistic regression, k-nearest neighbor, random forest, decision tree, GaussianNB classifier, gradient boosting classifier, support vector classifier respectively. All these models were tested after performing the hyper parameter tuning and cross validation using k- fold cross validation method.

Other notable steps include over sampling with SMOTE, hyper parameter tuning and plotting ROC-AUC curve of the models. The best and final fitted model was a logistic regression model with hyperparameter tuning done that give ROC AUC values of 91. The model performs well as compared to other models. In conclusion the model was able to correctly distinguish between fraud claims and legit claims with average accuracy.

The analysis has limitations as well. Due to lack of sample size and imbalance nature of data set can we eliminated that could lead to even better classification and prediction.

Further more intense hyperparameter tuning and other approaches could have been utilized.

**References**

Following references have been fruitful in understanding the problem statement and framing of our research and development of model to predict Fraud cases.

* [**www.google.com**](http://www.google.com/)
* **https://**[**www.wikipedia.org/**](http://www.wikipedia.org/)
* **https://**[**www.kdnuggets.com**](http://www.kdnuggets.com/)
* **https://learning.datatrained.com/myaccount/#/course/58339/lesson/**