Insurance Claim Processing in a Nutshell – Approval and Fraud Detection

Uncertainty and Risk in Life

Life is full of uncertainties and worries about the future. We can plan most of the things in our life, but still, there will be many uninvited and unexpected guests. Since we haven't invented the Time Machine yet, we can only be prepared for all these uncertainties in our life. **Accidents** are unintended, normally unwanted events. So, there is always the risk factor existing in life about these uncertainties. Risks can come from a variety of sources including accidents, uncertainty in international markets, legal liabilities, threats from project failures, credit risk, natural causes and disasters, deliberate attack from an enemy, or events of uncertain or unpredictable root cause.

Risk Management

We can manage these risks by identifying, evaluating, and prioritizing by coordinated and economical application of resources to minimize, monitor, and control the probability or impact of unfortunate events or to maximize the realization of opportunities.

Insurance

Insurance is a type of risk management, primarily used to enclose the risk of contingent or uncertain loss. Insurance is a way of protection from financial loss. An entity or company which provides insurance is known as an **insurer**, or **insurance company(Also known as** an **insurance carrier or an underwriter)**. An individual or entity covered under the policy is called an **insured**. If the insured experiences a loss which is potentially covered by the insurance policy, the insured submits a claim to the insurer for coverage or compensation for a covered loss or policy event.

We know how insurance plays a vital role in our life. Every policyholder is taking insurance towards financial losses as a part of risk management to face unexpected or uncertain events. So as the insurance claims are more important in the event of a covered loss. Even a fleeting time of delay in claim settlement will affect the insured in multiple ways.

Problem Definition

Why do Insurance Companies Take So Long for Insurance Claim Settlement?

Insurance companies may conduct a thorough investigation into an accident to find fault and liability. This is one of the reasons it may take a long time for insurance companies to pay out the sum. Damage reviews, contested claims, or even unfair claim settlement practices and fraud detections can also cause delays. This is always a time-consuming and a tough row to hoe.

Machine Learning in Claim Settlement and Fraud Detection

The traditional heuristic approaches for insurance claims fraud detection was either based on evaluation of claims towards framed rules that would define if the case needed to be sent for investigation or analysing the claims with a checklist on the scores for various indicators of fraud and using the aggregate of this score to determine if the case needs to be sent for investigation. The challenge with the above approaches is that they rely very heavily on manual interventions that lead to:

- Verification against a limited set of parameters based on heuristic knowledge, while other crucial factors are ignored.
- Inability to understand specific relationships between parameters
- Effort and time required for calibration of model for the periodic changing behaviours.
- Delay in claim settlement and inefficiency to detect the fraud claims.

These are the challenges from a traditional statistical perspective. Thus, insurance companies have begun to look at Machine Learning capabilities. Machine learning algorithms learn from historical fraud patterns and recognize them in future claims. It Is more effective than the heuristic and manual approaches, in respect of speed, and accuracy of the processing of the data for fraud detection. Also, ML algorithms can detect complex fraud features that human simply cannot detect.

Machine Learning is in a unique position to help the Automobile Insurance industry with this problem.

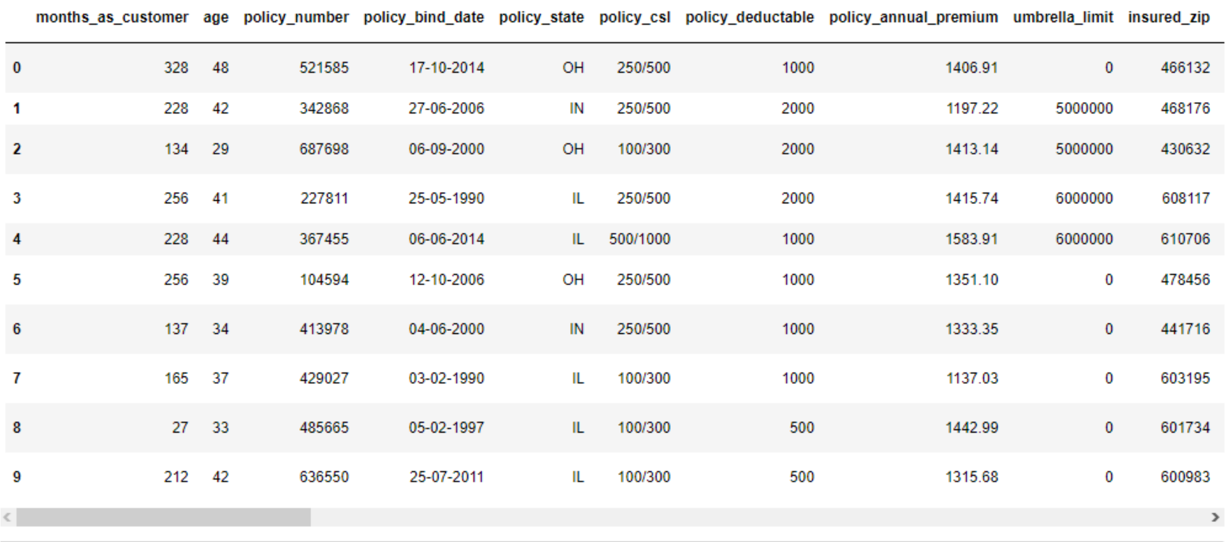
Dataset.

In this data analysis of insurance claim-approval and fraud detection, we **are provided a dataset which has the details of the automobile insurance policy along with the customer details. It also contains the information of the accident/incident based on which the claims have been made.**

Data Preparation

Loading the dataset.

```
#Loading the dataset


import pandas as pd
import  warnings
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns', None)

data = pd.read_csv('Automobile_insurance_fraud.csv')
data.head(10)
```



	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip
0	328	48	521585	17-10-2014	OH	250/500	1000	1406.91	0	466132
1	228	42	342868	27-06-2006	IN	250/500	2000	1197.22	5000000	468176
2	134	29	687698	06-09-2000	OH	100/300	2000	1413.14	5000000	430632
3	256	41	227811	25-05-1990	IL	250/500	2000	1415.74	6000000	608117
4	228	44	367455	06-06-2014	IL	500/1000	1000	1583.91	6000000	610706
5	256	39	104594	12-10-2006	OH	250/500	1000	1351.10	0	478456
6	137	34	413978	04-06-2000	IN	250/500	1000	1333.35	0	441716
7	165	37	429027	03-02-1990	IL	100/300	1000	1137.03	0	603195
8	27	33	485665	05-02-1997	IL	100/300	500	1442.99	0	601734
9	212	42	636550	25-07-2011	IL	100/300	500	1315.68	0	600983

Snapshot of the DataFrame

Our dataset includes data of 1000 automobile insurance claims, and it includes 39 features(Independent variables) and 1 target(Dependent Variable) about the insurance claim.(1000 rows and 40 columns). We have string, float, and integer type of data in the dataset.

Features in dataset(Independent Variable)

There were no values in the column '_c39'. So, we have dropped this column from our dataset.

months_as_customer - Duration of insurance of the insured in months.

age - Age of the insured

policy_number - Unique number to identify the insurance policy of the insured

policy_bind_date - The date when the automobile insurance was came into force.

policy_state - State where the insurance policy was taken.

policy_csl - Combined single limits of policy

policy_deductable - Amount of expense that the insured have to pay out before making an insurance claim.

policy_annual_premium - Annual premium of the insurance policy

umbrella_limit - Additional coverage.

insured_zip - Zipcode of the insured

insured_sex - Gender of the insured

insured_education_level - Educational qualification of the insured

insured_occupation - Occupation of the insured

insured_hobbies - Hobbies of insured

insured_relationship - Relationship status of the insured

capital-gains - Capital gains of the insured

capital-loss - Capital loss of the insured

incident_date - Date of incident

incident_type - Type of accident or incident

collision_type - Type of collision

incident_severity - Severity of the damage in incident

authorities_contacted - Authorities contacted when the incident happened

incident_state - State in which the incident happened

incident_city - City in which the incident happened

incident_location - Location information of place in which the incident happened

incident_hour_of_the_day - The time of the incident - Hour of the day

number_of_vehicles_involved - Number of vehicles involved in collision

property_damage - Property damage caused by the incident bodily_injuries - Number of bodily injuries

witnesses - Number of witnesses of the incident

police_report_available - Availability of the police report

total_claim_amount - Total amount of claim towards the loss and damage

injury_claim - Total amount of claim towards the bodily injuries

property_claim - Total amount of claim towards property damaged in incident.

vehicle_claim - Total amount of claim towards the vehicle damaged in incident.

auto_make - Manufacturer of the vehicle

auto_model - Model of the insured vehicle

auto_year - Manufacturing year of the vehicle

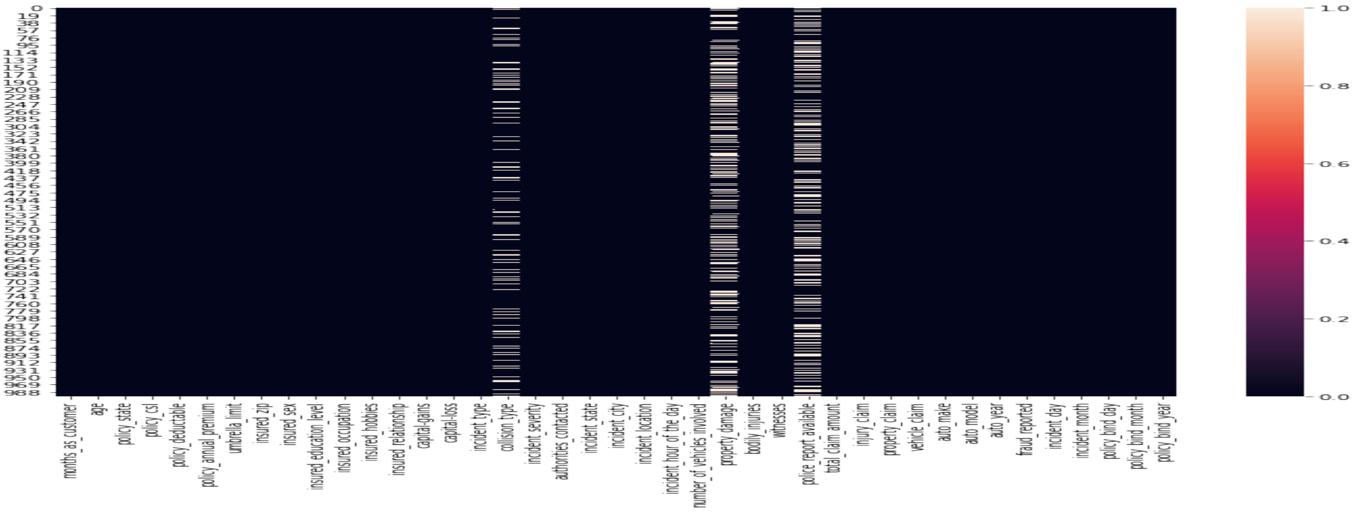
Target in dataset(Dependent Variable)

fraud_reported - Whether the claim is reported as fraud or not.

Since our target variable (‘fraud_reported’) is having binary output(i.e., there are only two outcomes : Yes or No), we will consider this as a classification problem use classification approaches and algorithms to build the predictive model.

The data was provided for the incidents happened in the year 2015. So, we dropped the column ‘incident_date’ from the column and added the day and month of the incident as new columns to the dataset. Also, we have dropped the column ‘policy_bind_date and added the day, month and year as new columns do the dataset.

Checking for Missing Value



Missing Values in the dataset

The columns ['collision_type', 'property_damage', 'police_report_available'] are having missing values.

Since the variables were categorical in nature, we imputed mode of the data of these columns for the missing values.

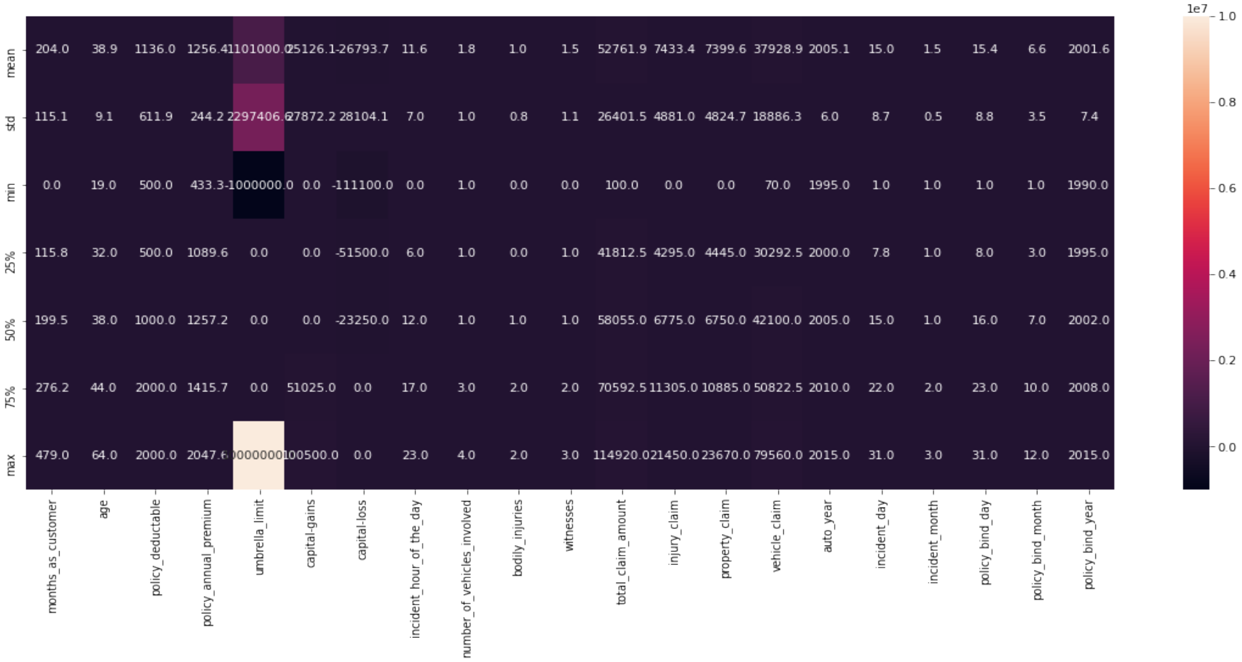
By removing the irrelevant columns, changing the datetime variables into readable formats for machine, and replacing the missing values with mode values are part of data processing, which enables us to build a powerful model.

Statistical Summary

Summary statistics is a part of descriptive statistics that **summarizes and provides the gist of information about the dataset.**

Describe of the Data

The describe function provides a set of information about the dataset including the mean, median, standard deviation, percentile(25%,50%,75%), minimum and maximum value of the numerical columns in the dataset. This will give us an insight about where the mean lies, whether the data is skewed etc.



Describe of the Numerical Data

- The graph shows that most of the variables are having asymmetry of a distribution of data, which also known as skewness. i.e., the data in the columns are not normally distributed.
- The columns ['age', 'policy_annual_premium', 'umbrellla_limit', 'total_claim_amount', 'property_claim'] were having huge difference in maximum value and the 75%. That means possible outliers, present in the data of these columns. In statistics, an outlier is a data point that is quite different from other observations.

Encoding the Categorical Variables

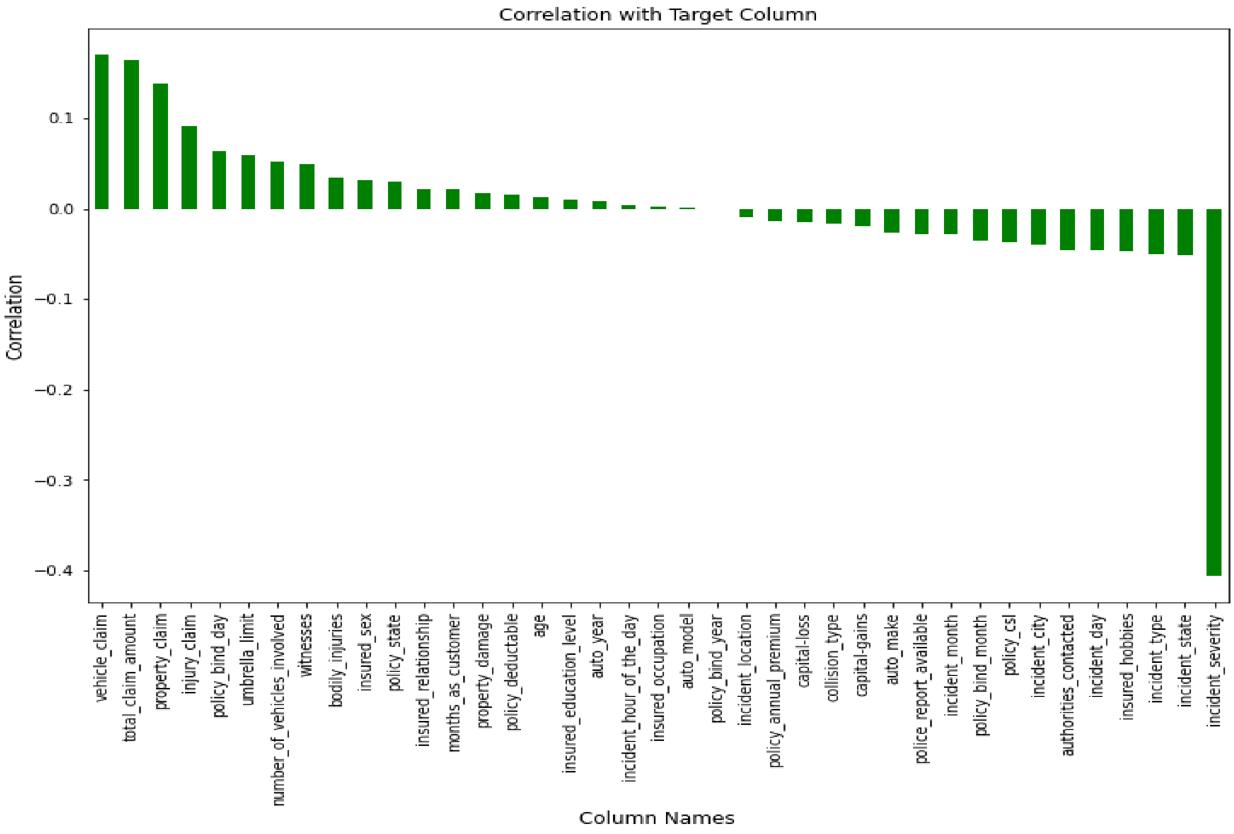
Since most ML Algorithms only accept numerical variables, preprocessing the categorical variables becomes a necessary step. We need to convert these types of categorical featurers to numbers(encoded) such that the model can understand and extract valuable information.

Scikit learn is a powerful library which contains many data science tools including the encoders. We have used ordinal encoder for the features(independent categorical variables) and label encoder for Target variable(dependent categorical variable).

```
from sklearn.preprocessing import OrdinalEncoder
from sklearn.preprocessing import LabelEncoder
onc = OrdinalEncoder()
lnc = LabelEncoder()
data['fraud_reported'] = lnc.fit_transform(data['fraud_reported'].values.reshape(-1,1))    #Label encoding for target or label.
for i in data.columns:
   if data[i].dtypes =='object':
        data[i] = onc.fit_transform(data[i].values.reshape(-1,1)).astype('int64')    #Ordinal Encoding for the features.
```

Correlation

In statistics, correlation or dependence is any statistical relationship(causal or not) between two random variables or bivariate data. Since we have more than 40 features(independent variables), we can check the correlation of them with the target variables.

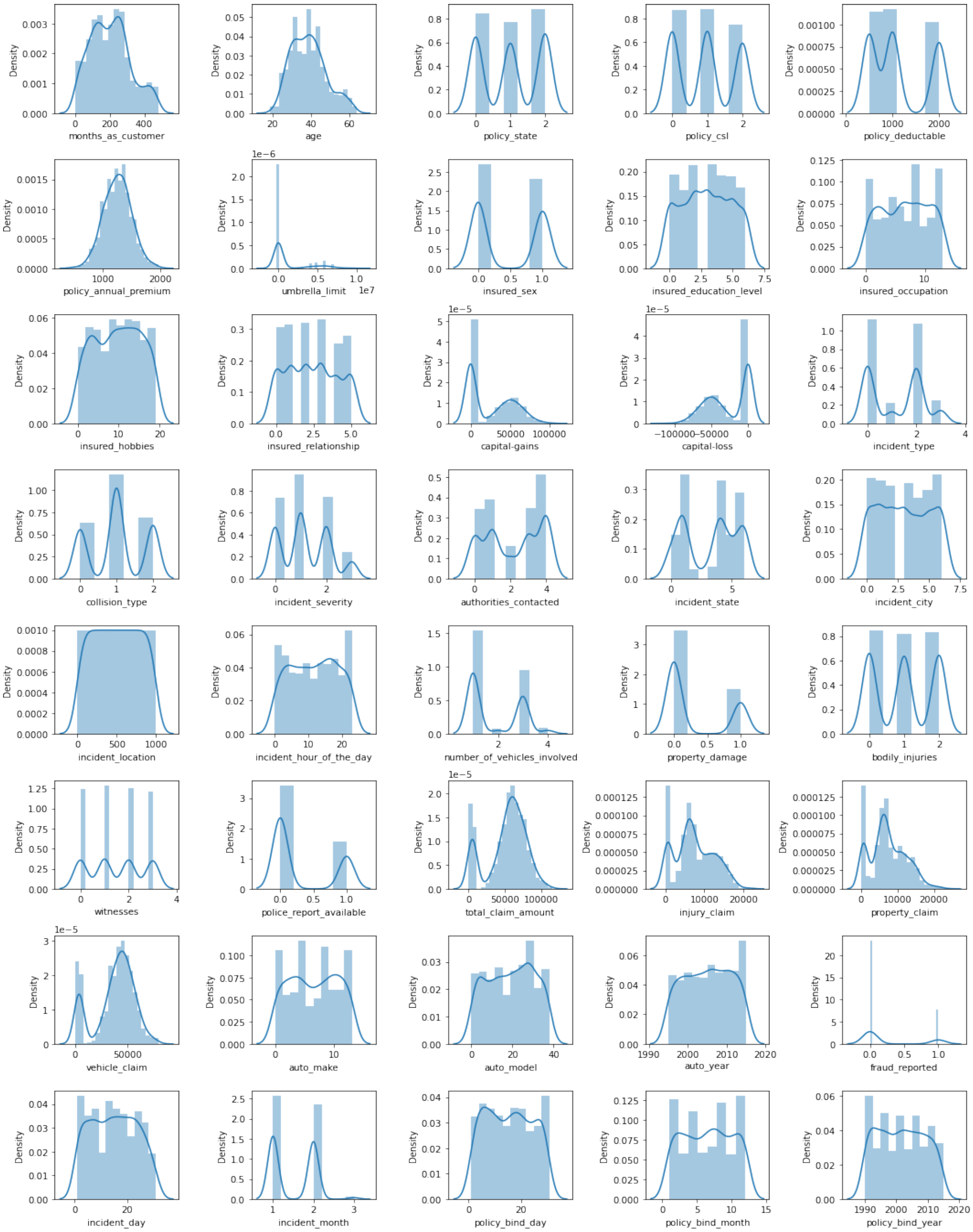


Correlation with Target Variable

Observations:

- The columns [‘vehicle_claim’, ‘total_claim_amount’, ‘property_claim’, ‘injury_claim’, ‘policy_bind_day’, ‘umbrella_limit’, ‘number_of_vehicles_involved’, ‘witnesses’, ‘bodily_injuries’, ‘insured_sex’, ‘policy_state’, ‘insured_relationship’, ‘months_as_customer’, ‘property_damage’, ‘policy_deductable’, ‘age’, ‘insured_education_level’, ‘auto_year’, ‘incident_hour_of_the_day’, ‘insured_occupation’, ‘auto_model’] are having positive correlation to the target variable ‘fraud_reported’.
- The column ‘vehicle_claim’ is having the highest positive correlation with the target variable ‘fraud_reported’, while the column ‘incident_severity’ is having highest negative correlation with the target variable ‘fraud_reported’.
- The column ‘auto_model’ is having the least positive correlation with the target variable ‘fraud_reported’ and the column ‘policy_bind_year’ is having least negative correlation to the target variable ‘fraud_reported’.

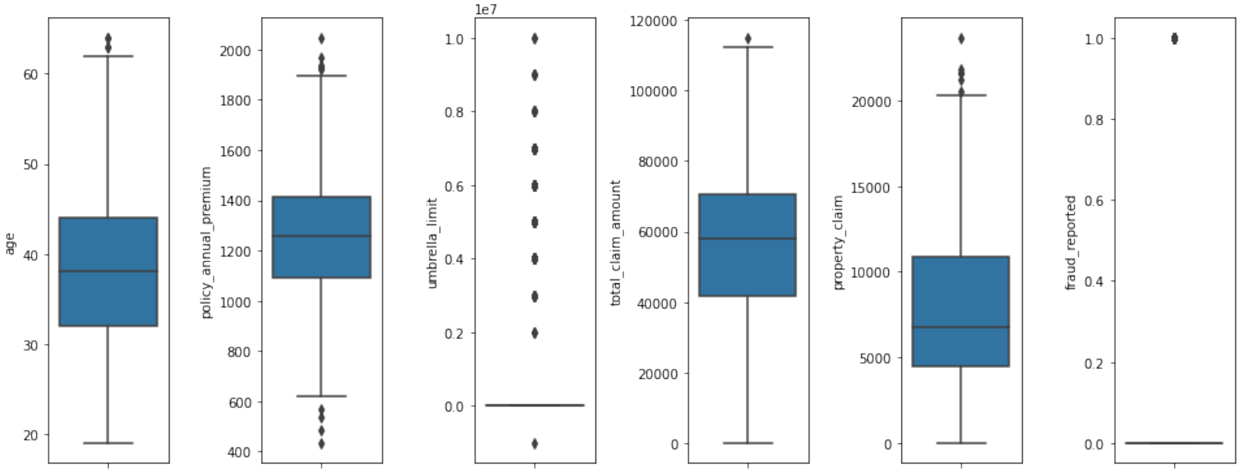
Distribution of data in columns(Checking skewness of data)



Skewness of Data

We can see that none of the columns were having normally distributed data. That means skewness is present in the data distribution of all the variables in the dataset.

Presence of Outliers in Data



Checking the Presence of Outliers using Boxplot

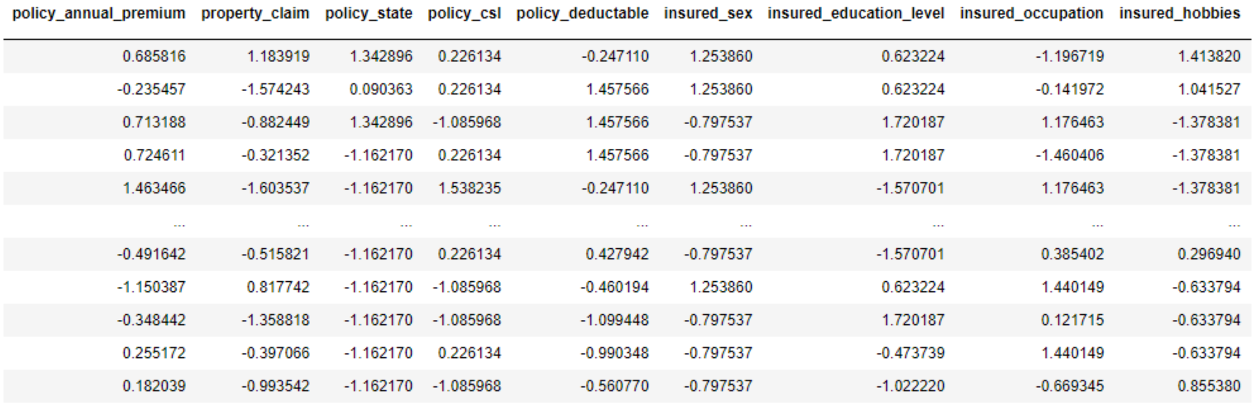
- These were the columns which were having outliers present in the data. ['age', 'policy_annual_premium', 'umbrella_limit', 'total_claim_amount', 'property_claim', 'fraud_reported'].
- But the column ‘fraud_reported’ is our target column and it is categorical in nature(Only two outcomes). So, if we consider the data as outlier, we may have to remove one whole category from the data. It will make our data biased towards one outcome.

Data Cleaning

Data cleaning is the process of correcting or removing incorrect, incorrectly formatted, corrupted, duplicate, or incomplete data from a dataset. These are the measures we took to clean our data, so that the machine can train and learn the about the variable relations and help us in the predictions.

- Replaced missing values(With mode values of variables for categorical variables)
- Removed imbalance of the data
  Because the dataset was having more data for policies which were not identified as fraud. But we need the data as balanced and shouldn’t be biased over one outcome. Thus, we over sampled the data for policies which were reported as fraud. Now the data is not biased towards any outcomes and both the outcomes are having equal importance.
- Removed skewness of data
- Removed outliers from the numerical data
- Feature scaling of the data
  Scaling or ***Feature Scaling*** is the process of changing the scale of certain features to a common one. This is typically attained through **standardization or normalization**(scaling techniques).
  In this dataset, we have used standardization as scaling technique. This helps us to standardize the data while preserving the uniqueness of each value in the dataset.
- Removed the Multicollinearity of the Variables.

Checking multicollinearity of a dataset is a part of PCA technique(Principal Component Analysis) which allows us to examine the correlation of dependent variables towards each other. If a dependent variable is highly correlated with another dependent variable, that means both are similarly correlated to the target variable. We don’t have to add many ingredients that gives the same flavor when we cook. Just like that we remove the multicollinearity from the dataset to make it clean and precise for the machine to learn.



policy_annual_premium	property_claim	policy_state	policy_csl	policy_deductable	insured_sex	insured_education_level	insured_occupation	insured_hobbies
0.685816	1.183919	1.342896	0.226134	-0.247110	1.253860	0.623224	-1.196719	1.413820
-0.235457	-1.574243	0.090363	0.226134	1.457566	1.253860	0.623224	-0.141972	1.041527
0.713188	-0.882449	1.342896	-1.085968	1.457566	-0.797537	1.720187	1.176463	-1.378381
0.724611	-0.321352	-1.162170	0.226134	1.457566	-0.797537	1.720187	-1.460406	-1.378381
1.463466	-1.603537	-1.162170	1.538235	-0.247110	1.253860	-1.570701	1.176463	-1.378381
...	...	...	...	...	...	...	...	...
-0.491642	-0.515821	-1.162170	0.226134	0.427942	-0.797537	-1.570701	0.385402	0.296940
-1.150387	0.817742	-1.162170	-1.085968	-0.460194	1.253860	0.623224	1.440149	-0.633794
-0.348442	-1.358818	-1.162170	-1.085968	-1.099448	-0.797537	1.720187	0.121715	-0.633794
0.255172	-0.397066	-1.162170	0.226134	-0.990348	-0.797537	-0.473739	1.440149	-0.633794
0.182039	-0.993542	-1.162170	-1.085968	-0.560770	-0.797537	-1.022220	-0.669345	0.855380

Final Dataset

Building the Model

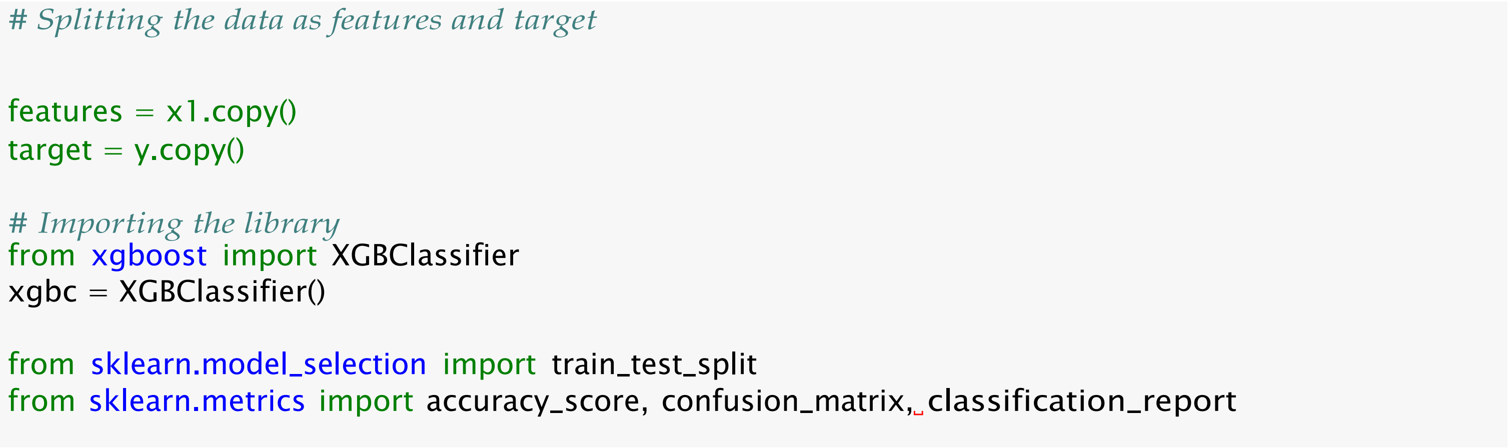
We have preprocessed our dataset and now it’s time to let the machine do its work. For that we must train the machine with the right algorithm. **Machine Learning algorithms** are part of AI, which uses an assortment of accurate, probabilistic, and upgraded techniques that empower computers to pick up from the past point of reference and perceive patterns that are difficult to perceive from massive, noisy, or complex datasets.

These algorithms help us to make the predictions. So, we train the algorithms with our preprocessed data and test to see if they are providing the right results. This will be done by cross validating the predicted results with our actual results(‘fraud_reported’ status). If the algorithm is predicting the results accurately, we can make sure that the machine has learnt the patterns which enables to make the predictions.

XGBoost for Classification

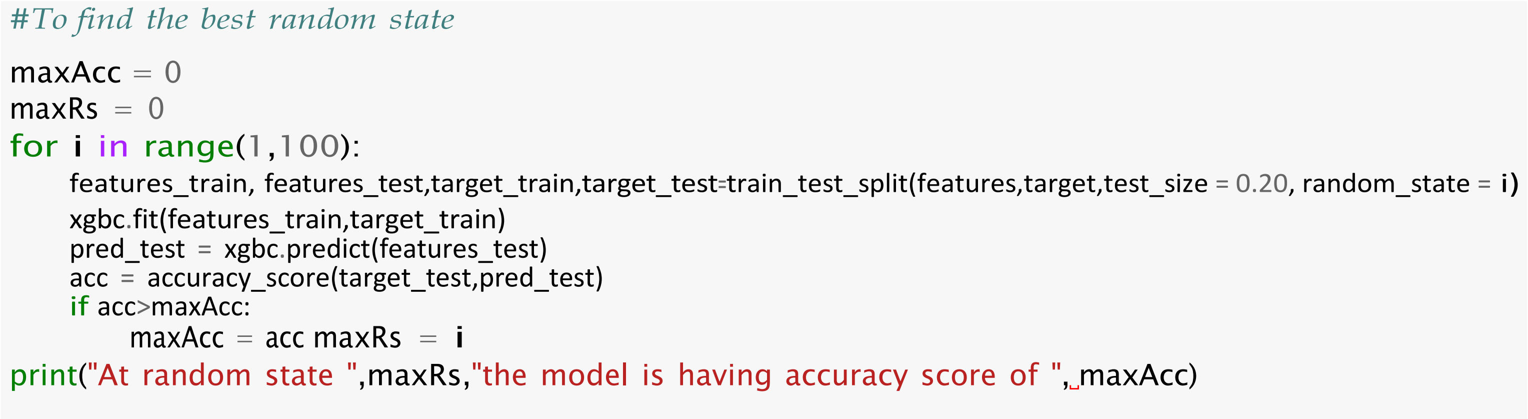
XGBoost is a powerful machine learning library that contains algorithms, which has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data. It became well known in the Machine Learning challenges(competitions) after its use in the winning solution of the Higgs Machine Learning Challenge. XGBoost is an adaptation of gradient boosted decision trees designed for performance and speed. Its speed and performance are unmatched and always surpass any other algorithms intended for supervised learning activities. Initially it was written in C++ as a command line application, later the library has its APIs in other languages like R, Python and Julia. It is an ensemble learning algorithm that uses Decision Tree as base learners, just like Random Forest Algorithm(Machine Learning Algorithm).

We have used XG Boost Classifier algorithm for machine learning of our dataset.



```
# Splitting the data as features and target


features = x1.copy()
target = y.copy()

# Importing the library
from xgboost import XGBClassifier
xgbc = XGBClassifier()

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

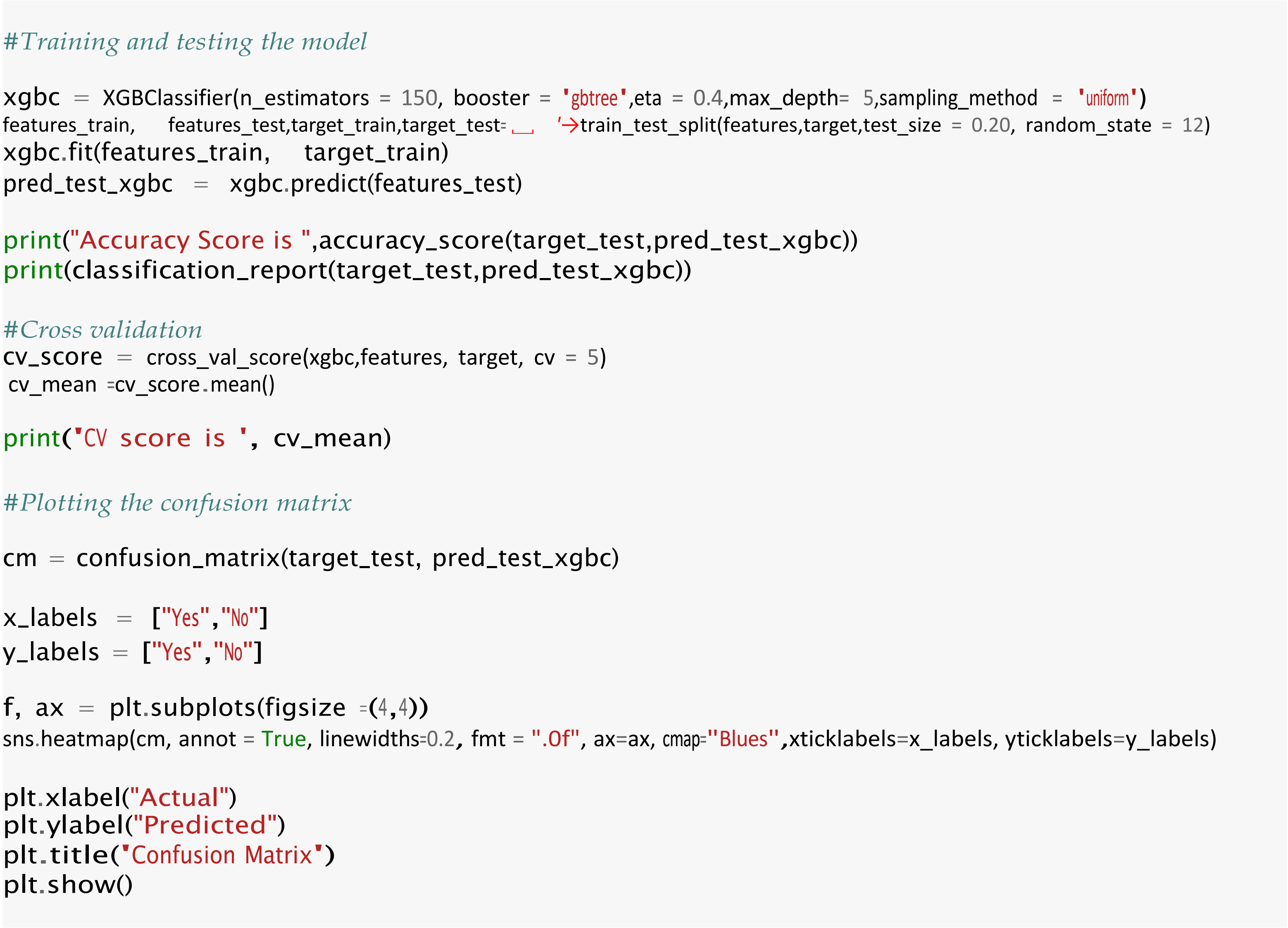
Importing Libraries for Algorithm and metrics



```
#To find the best random state
maxAcc = 0
maxRs = 0
for i in range(1,100):
    features_train, features_test,target_train,target_test=train_test_split(features,target,test_size = 0.20, random_state = i)
    xgbc.fit(features_train,target_train)
    pred_test = xgbc.predict(features_test)
    acc = accuracy_score(target_test,pred_test)
    if acc>maxAcc:
        maxAcc = acc maxRs = i
print("At random state ",maxRs,"the model is having accuracy score of ", maxAcc)
```

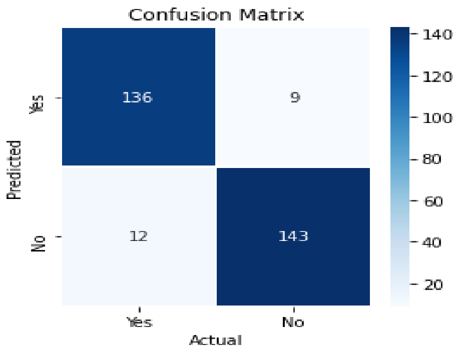
Finding the Best Random State

Training and Testing the Model



```
#Training and testing the model

xgbc = XGBClassifier(n_estimators = 150, booster = 'gbtree',eta = 0.4,max_depth= 5,sampling_method = 'uniform')
features_train,   features_test,target_train,target_test= ␣   '→train_test_split(features,target,test_size = 0.20, random_state = 12)
xgbc.fit(features_train,   target_train)
pred_test_xgbc  =  xgbc.predict(features_test)

print("Accuracy Score is ",accuracy_score(target_test,pred_test_xgbc))
print(classification_report(target_test,pred_test_xgbc))

#Cross validation
cv_score = cross_val_score(xgbc,features, target, cv = 5)
 cv_mean =cv_score.mean()

print('CV score is ', cv_mean)

#Plotting the confusion matrix

cm = confusion_matrix(target_test, pred_test_xgbc)

x_labels  = ["Yes","No"]
y_labels = ["Yes","No"]

f, ax = plt.subplots(figsize =(4,4))
sns.heatmap(cm, annot = True, linewidths=0.2, fmt = ".Of", ax=ax, cmap="Blues",xticklabels=x_labels, yticklabels=y_labels)

plt.xlabel("Actual")
plt.ylabel("Predicted")
plt.title('Confusion Matrix')
plt.show()
```

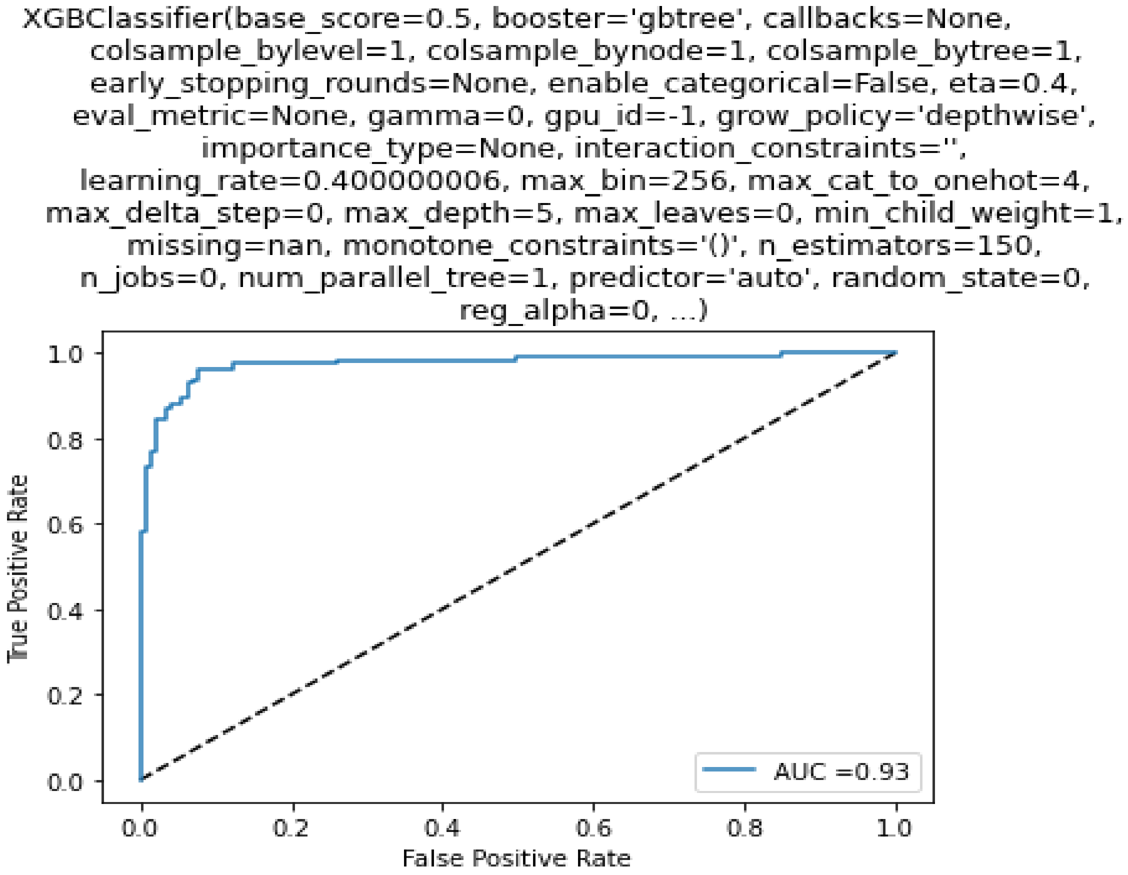
Model Building



Confusion Matrix

After testing, cross validations hyper parameter tunings, the XG Boost classifier model is performing well with an accuracy score of 93% with a cross validation mean score of 87.1%. That means our model is now trained to make the predictions with an accuracy of 93%.

AUC ROC Curve



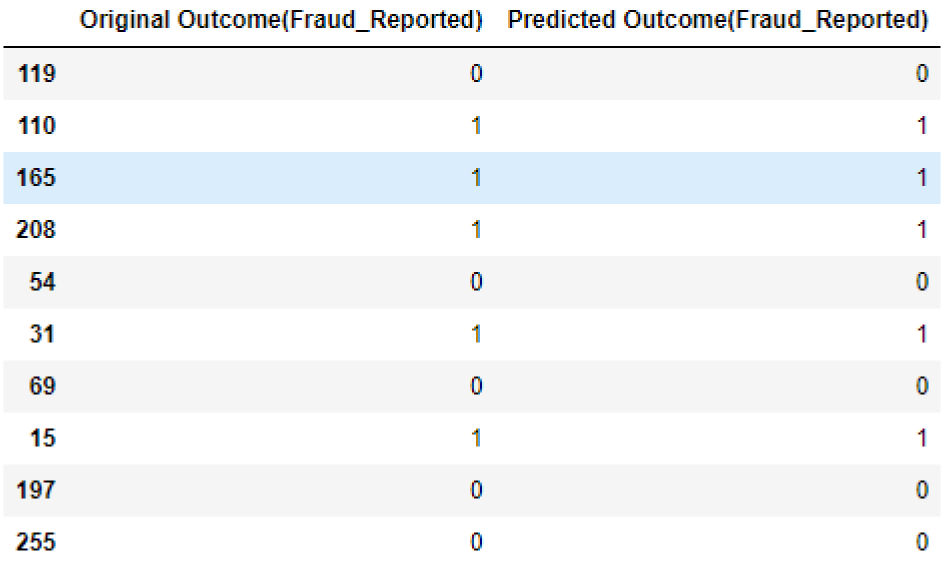
```
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
      colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
      early_stopping_rounds=None, enable_categorical=False, eta=0.4,
     eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
            importance_type=None, interaction_constraints='',
    learning_rate=0.400000006, max_bin=256, max_cat_to_onehot=4,
  max_delta_step=0, max_depth=5, max_leaves=0, min_child_weight=1,
        missing=nan, monotone_constraints='()', n_estimators=150,
    n_jobs=0, num_parallel_tree=1, predictor='auto', random_state=0,
                          reg_alpha=0, ...)
```

AUC ROC CURVE and Score

The XGB classifier model is providing an AUC(Area Under the Curve) score of 93%. That means most of the data is lying under the curve which is a good indicator that our model is performing well.

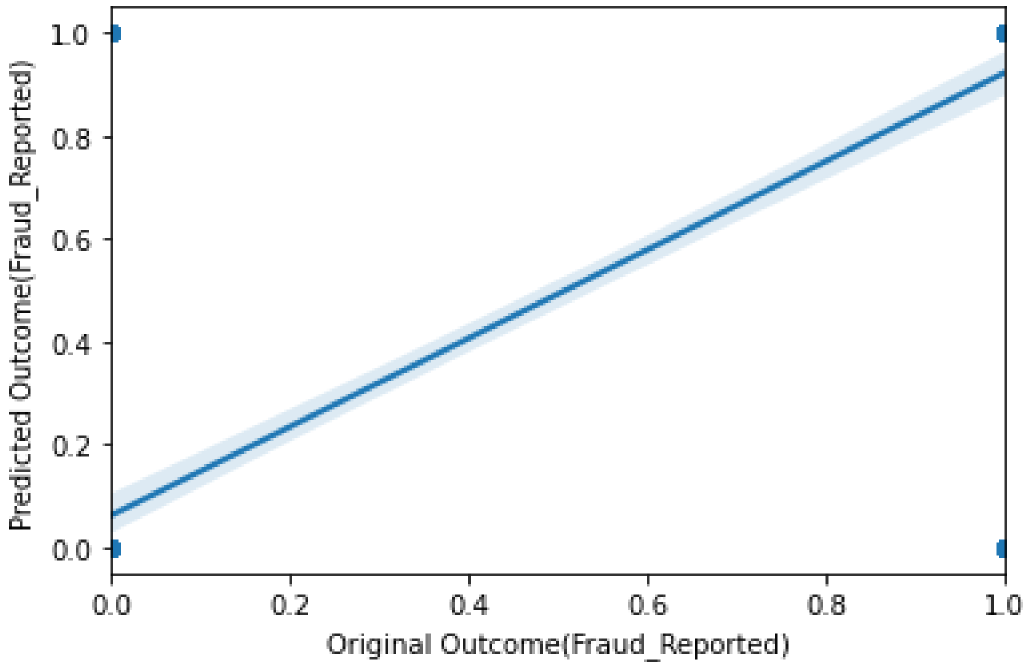
Conclusion

After all the data cleaning, preprocessing, processing, we built a powerful machine learning model with XG Boost algorithm, which can predict whether an insurance claim is fraud or not. Now let’s predict the outcome with the model to see how it is performing with the predictions.



	Original Outcome(Fraud_Reported)	Predicted Outcome(Fraud_Reported)
119	0	0
110	1	1
165	1	1
208	1	1
54	0	0
31	1	1
69	0	0
15	1	1
197	0	0
255	0	0

Model Prediction and Actual Outcomes



Prediction Accuracy Graph

We can see that the sample predictions with XGBoost classifier model, is accurate and didn't create any errors. That says, our model is performing well with predictions.

Summary

Processing the insurance claims and fast processing of the settlement was a time consuming and effort taking with the traditional heuristic manual approach. But now with the help of machine learning and A.I. we made the process easier with more efficiency and accuracy. Now we can predict the fraudulent insurance claims using the following information about the insurance claims.

* incident_type * bodily_injuries * insured_education_level
* incident_state * incident_location * capital-loss * policy_bind_month
* auto_year * policy_bind_year * policy_annual_premium * insured_occupation
* policy_bind_day * number_of_vehicles_involved *vehicle_claim
* property_claim * injury_claim * incident_severity * authorities_contacted
* incident_hour_of_the_day * incident_month * collision_type
* police_report_available * auto_model * insured_relationship
* auto_make * witnesses * policy_csl * insured_sex * age
* incident_day * insured_hobbies * incident_city * property_damage
* capital-gains * policy_state * policy_deductable

By automation and technology integration, the insurance companies can automate the insurance claim settlement and with machine learning and A.I , they can easily detect the fraudulent claims. This will not only just save the time and reduce cost for the company , but it will help the fast processing of the settlement which will help the insured who can get the financial aid at the right time for meeting the financial losses incurred in the incident.