**Insurance Claims Fraud Detection Using Machine Learning.**

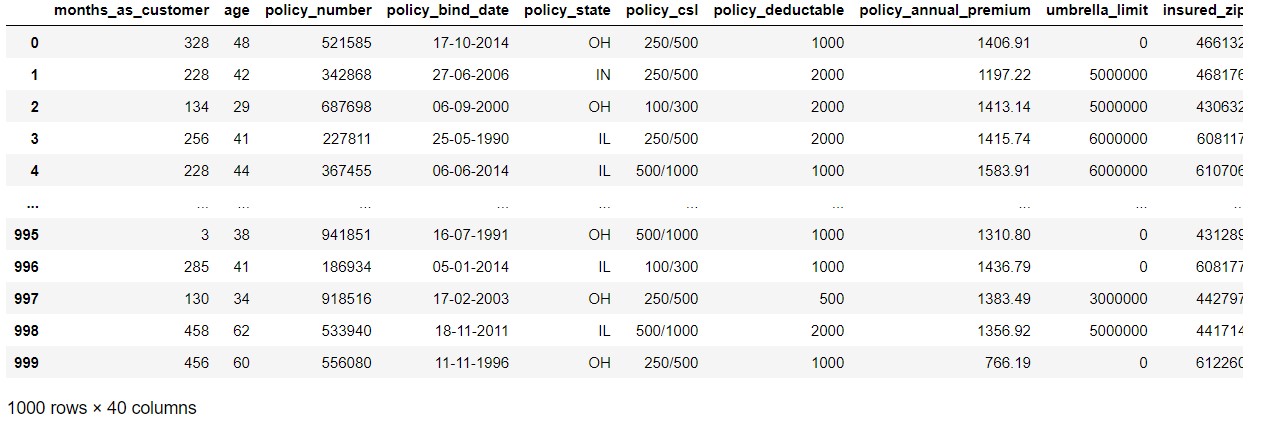
**Problem Definition:**

Business case: 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.

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

**Data Analysis:**

****

**There are 1000 rows and 40 columns in the dataset.**

The Independent Feature columns in the dataset are as follows:

months\_as\_customer: How many months the person is a customer

age: Age of the Customer

policy\_number: Unique number of policies

policy\_bind\_date: Time period between effective date of coverage and policy insurance.

Policy\_state: The state where policies are active

Policy\_csl: Policy Combined Single limit

Policy\_deductable: Amount paid before insurance company starts paying up.

Policy\_annual\_premiun: Annually paid premium

Umbrella\_limit: Provides excess limits and gives additional excess coverage

Insured\_zip: Zip code of the insured address

Insured\_sex: Gender

Insured\_education\_level: Education of the insured

Insured\_occupation: Occupation of the insured

Insured\_hobbies: Hobbies of the insured

Insured\_relationship: Relationship of the insured

Capital\_gains: Capital gains made from insurance

Capital\_loss: Capital loss incurred

Incident\_date: Date of the incident

Collision type: Which type of collision

Incident\_severity: Severity of the incident

Authorities\_contacted: whether authorities were contacted

Incident\_state: state where incident occure

Incident\_city: City where incident occure

Incident\_location: location of the incident

Incident\_hour\_of\_the\_day: Time on which incident occure

Number\_of\_vehicles\_involved: Number of vehicles involved in the incident.

Property\_damaged: Property is damaged or not

Bodily\_injuries: Severity of bodily injuries.

Witnesses: Number of witnesses

Police\_report\_available: Whether police report is available or not

Total\_claim\_amount: Total amount of claim

Injury\_claim: Injury claim amount

Property\_claim: Property claim amount

Vehicle\_claim: Vehicle claim amount

Auto\_make: Make of vehicle

Auto\_model: Model of vehicle

Auto\_year: Manufacturing year of vehicle

The target variable In the dataset are as follows:

Fraud Reported: Whether fraud reported as Yes or No

* **Next step is checking for null values:**

\_c39 has no usable data. Other columns appear to have no null values. So we will drop it.

* **Next step Is checking for unique numbers**

we can observe the column policy\_number and incident\_location have thousand unique values which means they have only one value count. So it is not required for prediction and we can drop it.

* **Next step is checking for value counts of each columns**

By looking at the value counts of each column we can realize that the columns umbrella\_remit, capital\_gain and capital\_loss contains more zero values around 79.8%, 50.8% and 47.5% respectively. I'm keeping the zero values in a capital gain and capital loss columns as it is since umbrella limit column has more than 70% of 0 values less drop it.

* **Next step is checking for datatypes**

we have extracted the month and year corner from both policy blind date and incident date so we can drop this column

* **Next step is checking for categorical columns**

The categorical columns are

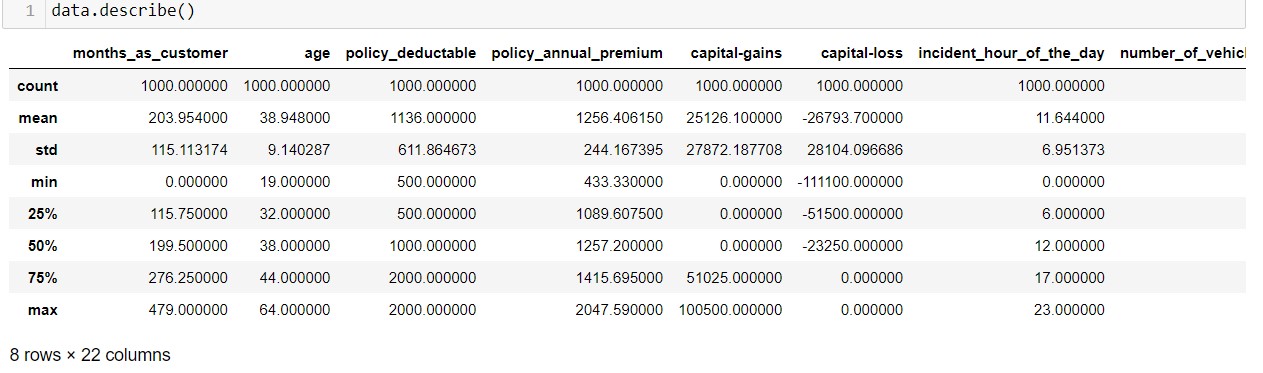
'policy\_state', 'insured\_sex', 'insured\_education\_level', 'insured\_occupation', 'insured\_hobbies', 'insured\_relationship', 'incident\_type', 'collision\_type', 'incident\_severity', 'authorities\_contacted', 'incident\_state', 'incident\_city', 'property\_damage', 'police\_report\_available', 'auto\_make', 'auto\_model', 'fraud\_reported'

* **Next step is checking for numerical columns**

The numerical columns are

'months\_as\_customer', 'age', 'policy\_deductable', 'policy\_annual\_premium', 'capital-gains', 'capital-loss', 'incident\_hour\_of\_the\_day', 'number\_of\_vehicles\_involved', 'bodily\_injuries', 'witnesses', 'total\_claim\_amount', 'injury\_claim', 'property\_claim', 'vehicle\_claim', 'policy\_bind\_day', 'policy\_bind\_month', 'policy\_bind\_year', 'incident\_day', 'incident\_month', 'csl\_per\_person', 'csl\_per\_accident', 'Vehicle\_Age'

* **Next step is checking the describtion of the dataset**

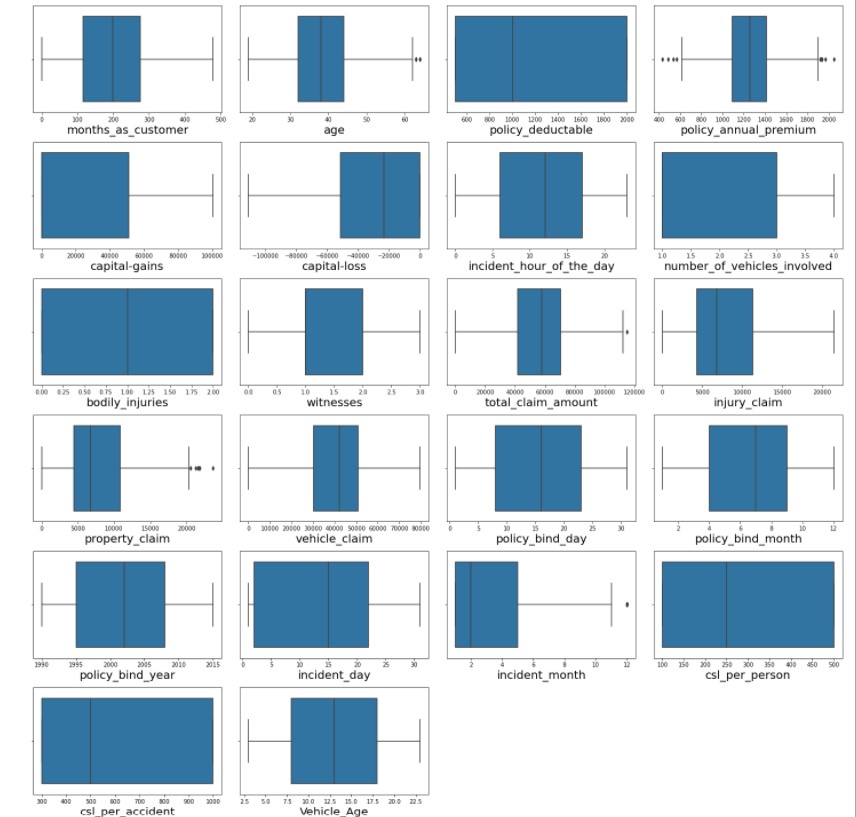


These give the statistical summary of the dataset

* Here the count of all columns is equal which means there are no missing values in dataset.
* In some of the columns like policy\_deductable, capital\_gain, injury\_clain etc we can observe the mean value is greater than the median which means the data inj those columns are skewed to the right.
* And in some of the columns like total\_claim\_amount, vehicle\_claim etc we can observe the median is greater than the mean which means the data in the columns are skewed to the left.
* And some of the columns have equal mean and median that means the data symmetric and is normally distributed and so skewness present.
* There is a huge difference in 75% and max it shows that huge outliers present in the columns.
* **Next step is Data Visualization**

In this step we will visualize different types of visualization for better understanding. In this step we will visualize the features with respect to target variable and features with respect to features.

* **Next step is checking for outliers**



We can find the outliers in the following columns.

* age
* policy\_annual\_premium
* total\_claim\_amount
* property\_claim
* incident\_month
* **Next step is Removing outliers**

In this step we remove outliers using zscore method and IQR method.  
We notice that using zscore we loose very less data so we consider zscore

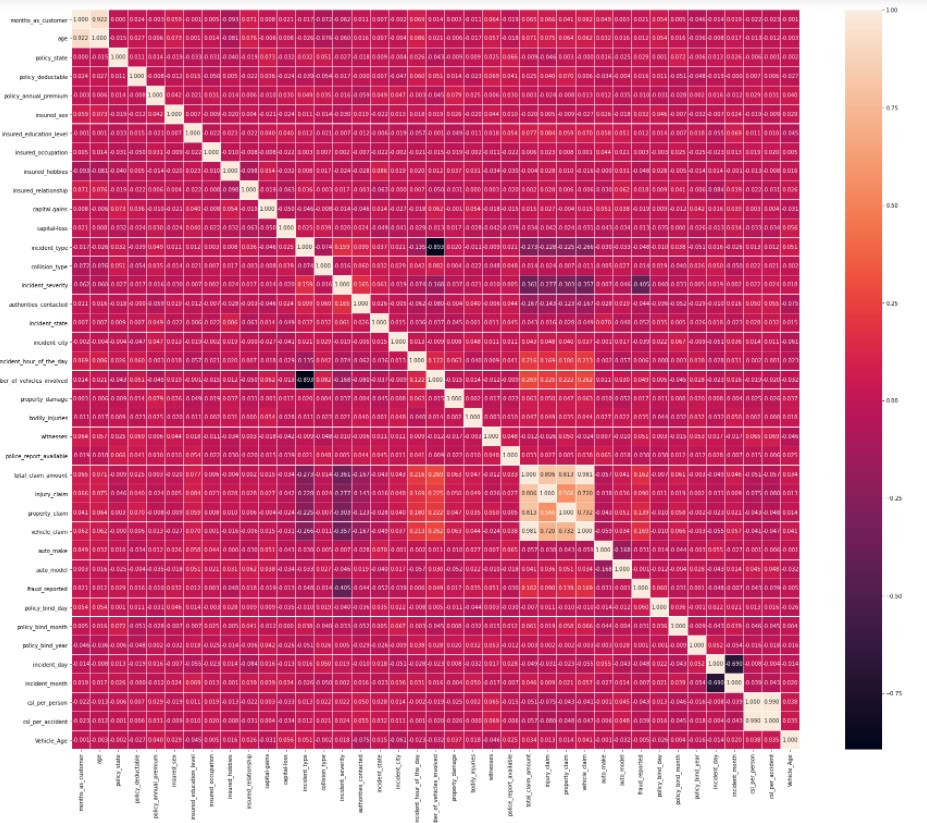
* **Next step is checking for skewness**

The following features contain skewness:

* total\_claim\_amount
* vehicle\_claim
* incident\_month
* csl\_per\_accident
* **Next step is removing skewness**

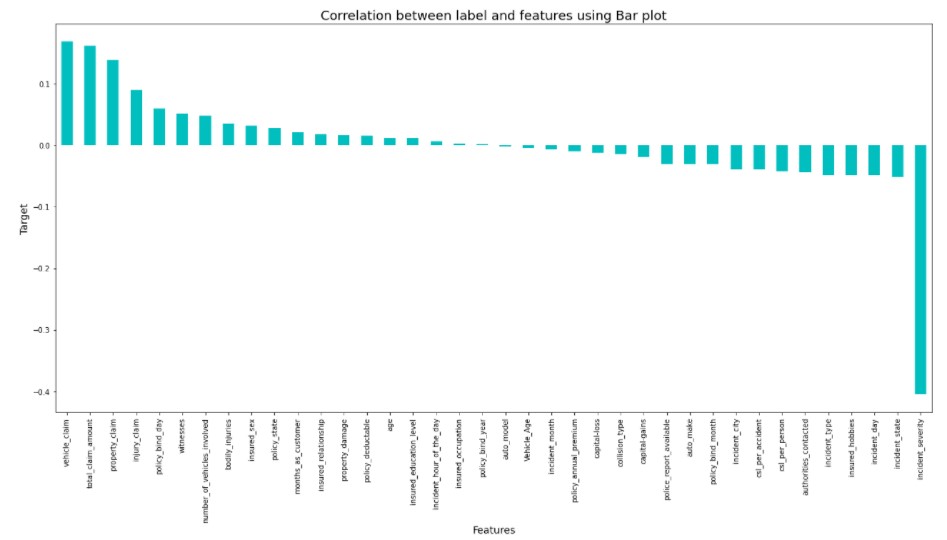
We removed the skewness by yeo-jonhson method and after that plotted the distribution graph the graph looks almost natural after removing the skewness.

* **Next step is checking for correlation**



This heatmap shows the correlation matrix by visualizing the data. We ccan observe the relation between one feature to other.

* This heatmap contain both positive and negative correlation
* We can observe the most of the columns ahe highly correlated with each other which vreates multicolinerity problem.
* We will check thew VIF value to overcome with this multicolinearity problem.



* **Next step is separating features and labels**

In this step we separate the feature columns and target columns.

* **Next step is standard scaler**

In this step we scaled the data

* **Next step is checking for VIF values**

In this step we will check that any multicollinearity exists in the data or not.  
As there is multicollinearity problem is present in few of columns, we solved it by dropping few of them.

* **Next step is Oversampling**

In this step we balanced the data using oversampling method and processed for modeling.

* **Next step is modelling**

In modeling we get 90.88% accuracy on the random state of 80

* **Next step is finding the best model**

1. By RandomForestClassifier we get 91.55% accuracy
2. By SVC we get 86.66% accuracy
3. By GradientBoostingClassifier we get 90.22% accuracy
4. By AdaBoostClassifier we get 88% accuracy
5. By BaggingClassifier we get 88% accuracy
6. By ExtraTreeClassifier we get 91.11% accuracy

* **Next step is checking for cross validation**

1. Cross\_validation\_Score Of RandomForestRegressor is: 0.868
2. Cross\_validation\_Score Of SVC is: 0.858
3. Cross\_validation\_Score Of GradientBoostingClassifier is: 0.8460000000000001
4. Cross\_validation\_Score Of AdaBoostClassifier is: 0.8460000000000001
5. Cross\_validation\_Score Of BaggingClassifier is: 0.8699999999999999
6. Cross\_validation\_Score Of ExtraTreeClassifier is: 0.9153333333333334

**The ExtraTreeClassifier is our Best Fit Model**

* **Next step is Hyper parameter Tuning**

After tuning we get89.11% Accuracy

* **Next step is plotting ROC AUC curve**

We plotted the ROC AUC curve

* **Next step is saving the model**

In this step finally we save the model