Insurance Claims- Fraud Detection

## A step-by-step guide to insurance claim fraud detection!

In this article, I am going to walk you through how we can train a model that will help us to predict if an insurance claim is fraudulent or not. We will practice the machine learning workflow to predict fraudulent activity in auto industry.

Text

Description automatically generated

**Objective**

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

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

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

**About the Auto Insurance Fraud**

Insurance fraud is any act committed to defraud an insurance process. It involves hoaxing with an insurance company about a claim involving their personal or commercial motor vehicle. It can involve giving out misleading information or providing false documentation to support the claim.

**Data Set**

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

**Approach**

1. Importing the important libraries such as Pandas, Numpy, matplotlib, andseaborn
2. Performing **Exploratory Data Analysis** (EDA) to check pattern of the data set
3. Data pre-processing
4. Diving the data set into training and testing.
5. Creating Machine learning models such as Decision Tree classifier, KNeighbors classifier, Support vector machine classifier, and Naive bayes classifier.
6. Begging and boosting using ensemble techniques: Random Forest classifier, AdaBoost classifier, Gradient Boosting classifier, XGboost classifier
7. Hyperparameter tuning with Grid search CV
8. Plotting AUC/ROC curve.
9. Saving the model
10. Conclusion

**Importing the libraries:**

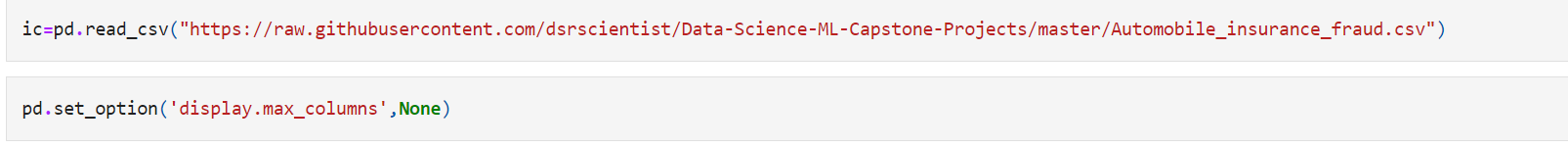
We will begin with importing the libraries such as Pandas, NumPy, matplotlib, and seaborn.

Graphical user interface

Description automatically generated with medium confidence

# Loading the data set: Since data is in form of an csv file, we have to use “pandas.read\_csv” to load the data and store it in data frame.

**Our data set will look like this:**



A picture containing table

Description automatically generatedTable

Description automatically generated

Table

Description automatically generated with medium confidenceTable

Description automatically generated with medium confidence

**Performing EDA:**

Exploratory Data Analysis (EDA) is the key process to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations.

**Benefits of EDA**

1. It gives us valuable insights into the data.
2. It helps us with feature selection (i.e using PCA)
3. Visualization is an effective way of detecting outliers.

Table

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Chart

Description automatically generated

In the above dataset:

1) The size of table is 1000 \* 40 i.e., no. of rows is 1000 and no. of columns are 40(including target).

2) Out of 40 columns, 19 columns are continuous in nature and rest 21 are object type.

3) Null values are present in only one column named \_c39 as we can see in the seaborn heatmap. Hence, we will drop this column.

4) In case of object data type, we will apply the encoding technique to convert the values in the numeric format.

## Univariate analysis

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated Chart

Description automatically generated with low confidence

Chart, pie chart

Description automatically generatedChart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

Chart, pie chart

Description automatically generated

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generatedChart, pie chart

Description automatically generated

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

In the data set, we have a Fraud\_reported column as a target variable, inside which output is in the form of Yes or No. This column is reflecting the chance of fraud being less as compared to fraud accounted. Auto year is exhibiting the Auto years' information, which have available years starting from 1995 to 2015. In 1995, 1999, 2002, 2003, 2005, 2006, 2007, 2009 ,2010 and 2011 huge amount of manufacturing was done as compared to other years. In addition, we have many auto models in the dataset, where Ram and wrangler model is the topmost occurring model as compared to other models. A3, Neon, MDX, Jetta, Pathfinder, AS, Malibu and Passat model also turning up exquisitely in the model column.

In appearance, insured age lies between 18 to 65, however, 25-50 age group people making the more claims. We have one more column, where three states of policy named OH, IN, and IL are provided, giving the information of policy state. Umbrella limit also plays the vital role when it comes to insurance claims. In our dataset umbrella\_limit is zero in most of the cases. In few cases umbrella limit is 4000000, 3000000, 6000000, 7000000, 4000000 and 3000000 as well. Insured gender column demonstrate that both the genders are almost same in number in our data set, stating claim is being done by the both the genders equally. Multi vehicle and single vehicle incidents are present in most of the insurance claims in comparison to the parked cars and vehicle theft incident. Types of collisions are side collision, rear collision and front collision, that are being used by the people to make the claims. Incident severity column is showing the damage. Minor damage (35.4%), major damages (27.6) and total loss (28%) case, we get to see most of the time. Only few cases are present related to trivial damage (9.0).

City incident column presenting the cities’ data where the incident had taken place. Columbus, springfield, Arlington and northbend are the cities where number of incidents are higher.

## Bivariate analysis

Chart, bar chart

Description automatically generatedDiagram

Description automatically generated with low confidence

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

Chart, bar chart

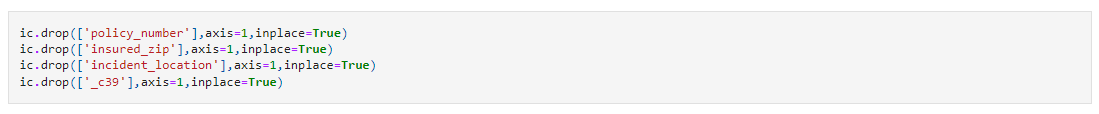
Description automatically generated

**From the above results we can conclude that:**

Fraud reported output shows that major the damage, higher the chances of fraud. Minor the damage, chances of fraud are less likely to arise, and in the trivial damage the chance of fraud is subordinate. In the Columbus and Arlington city fraud cases are higher as compared to other cities. Northbrook and Riverwood cities have less fraud cases reported. Fraud cases are present equally in all the three policy\_state and are being accounted in all the education level.

We can also observe high number of fraud cases in single vehicle collision and multi vehicle collision incident types. On the contrary, in the event of vehicle theft and parked car, the chance of fraud is very less. In the case of Rear collision, the chance of fraud is little bit higher and in the event of front and side collision, the chance of fraud is high and equal as well. If we focus the auto make and fraud reported column, we will find that Toyota, jeep, Nissan and Acura are claiming less fraud case as compared to others auto make. Rest of the auto makes are claiming almost equally fraud cases.

Above all, it is essential to observe the columns in data set to understand what it is stating and which sort of values its returning. While fulfilling the order, sometimes few columns are there which does not play important role to build machine learning model. If we do not remove these from our data set, then it messes up and does not provide the standard accuracy.



The Data set is providing the information of policy\_bind\_date and incident\_date, however we must separate the day, month and year for model so that our machine can grab it well

and encoding become easier.

Graphical user interface, text, application, email

Description automatically generated

As we see, 'months\_as\_customer’, and ‘age’ columns, comprised of huge number of unique values. If we give the data as simple as possible, our machine will receive the information accurately. This is smart way to achieve good accuracy, notwithstanding not the only way.

Table

Description automatically generated

Chart, bar chart

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Chart, bar chart

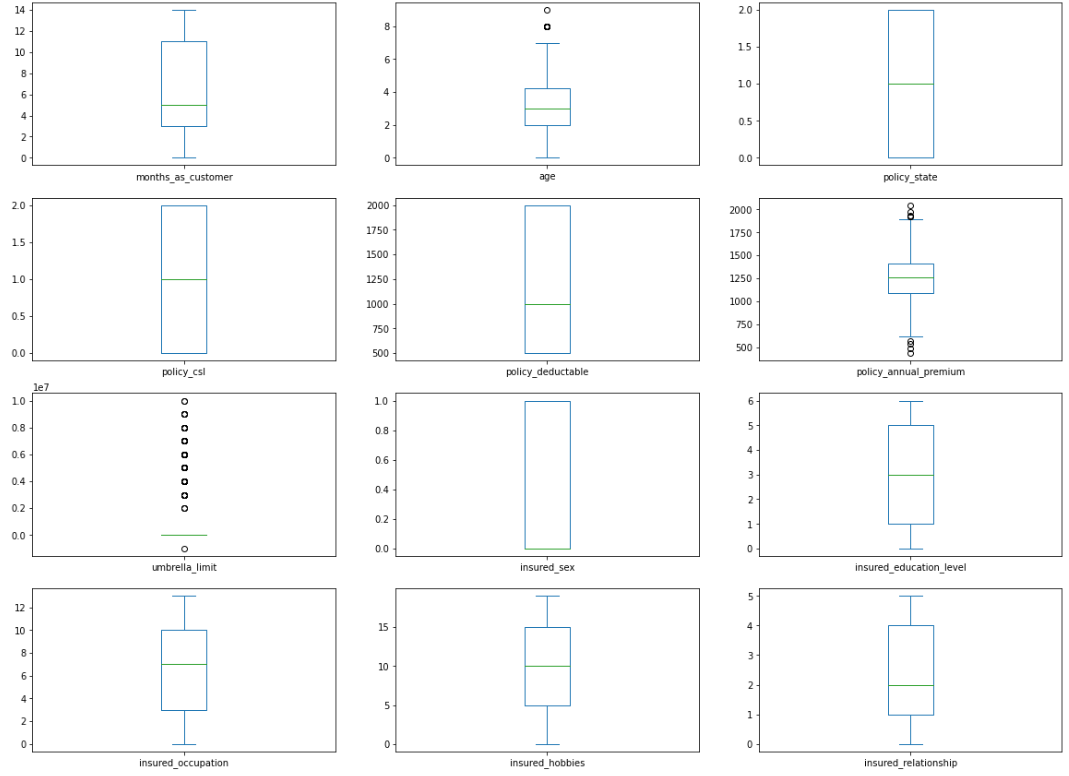
Description automatically generated

Encoding is one of the most important steps for machine learning model as we get some information in the form of object data type and for machine, it will be difficult to read the object data. To create the machine learning model, we need to covert these object data type to the numeric format.

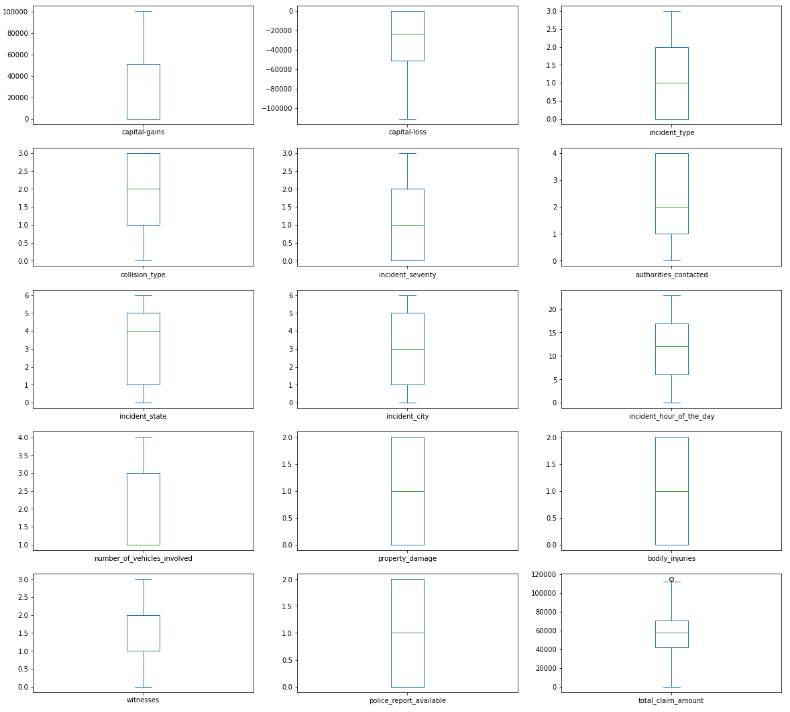
Let’s perform this task:

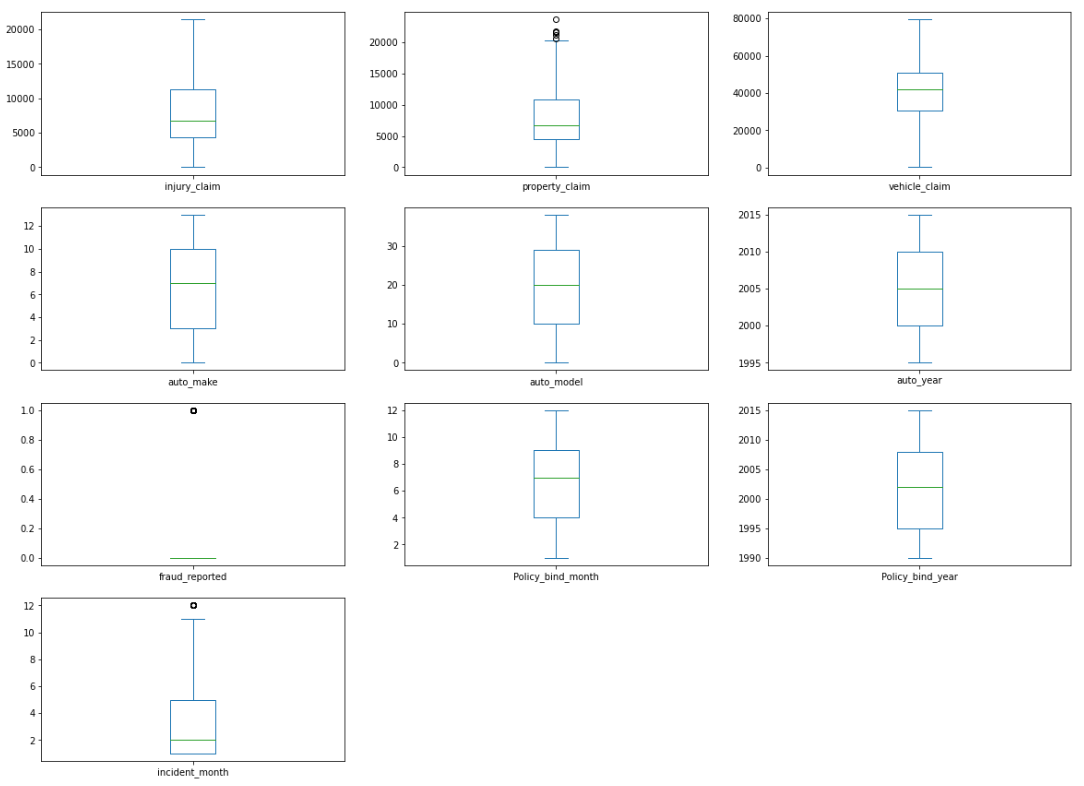
Table

Description automatically generated

**Outliers:**

Outliers are the values that look different from the other values in the data. Outliers in the data may causes problems during model fitting (esp. linear models). Outliers may inflate the error metrics which give higher weights to large errors (example, mean squared error, RMSE). There are many ways to remove the outliers, but we will use Z score method. If the z score of a data point is more than 3, it indicates that the data point is quite different from the other data points. Such a data point can be an outlier.





As we can see in the above box plots, only few columns have outliers, it means data are not highly laid out from the mean. We checked outliers and found it in policy\_annual\_premium and property\_claim column. Standard deviation of this column is high which shows the possibility of outliers present.

## Checking skewness:

Chart, line chart

Description automatically generated

**skewness** is a measure of the asymmetry of the [probability distribution](https://en.wikipedia.org/wiki/Probability_distribution) of a [real](https://en.wikipedia.org/wiki/Real_number)-valued [random variable](https://en.wikipedia.org/wiki/Random_variable) about its mean. The skewness value can be positive, zero, negative, or undefined.

*negative skew*: The left tail is longer; the mass of the distribution is concentrated on the right of the figure.

*positive skew*: The right tail is longer; the mass of the distribution is concentrated on the left of the figure.

Table

Description automatically generated

Now, let’s find the correlation of the columns with the target variables and we will also see the correlation between the columns, however in essence, it’s important to know what correlation is.

Correlation measures the relationship between two variables. It normally refers to the degree to which a pair of variables are linearly related. If increment in the values of one column leads to the increment of other column values, then it indicates the positive correlation b/w the columns and conversely, negative correlation means when one variable decreases so does the other. A negative correlation means that the variables move in opposite directions.

If value of one variable is increasing and another remains same, so this type of correlation is called zero correlation. A value of zero indicates that there is no relationship between the two variables.

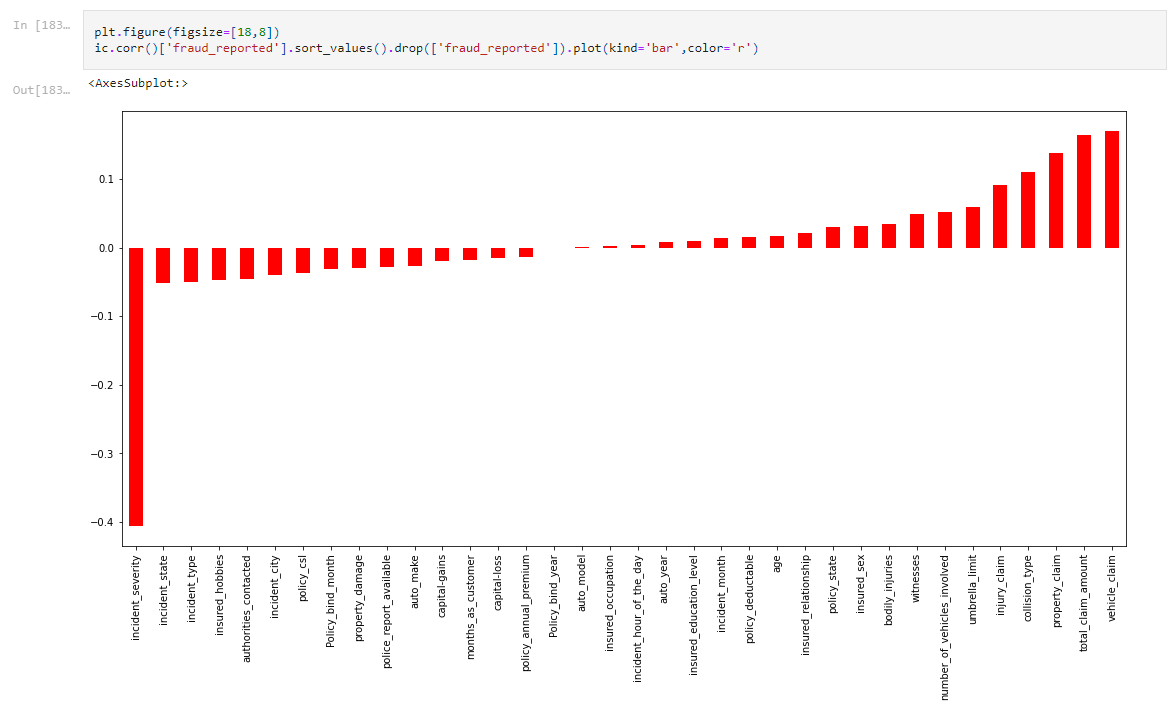
The correlation coefficient can never be less than -1 or higher than 1.

Table

Description automatically generated

Graphical user interface

Description automatically generated



From the above observations we can conclude that:

1) In our data set, no column is making a strong bond with the target column

2) Vehicle claim, total\_claim\_amount, property\_claim, collision\_type, injury\_claim, umberela\_limit, numbers\_of\_vehicle\_involved, witness, badly\_injuries, insured\_sex, policy\_state, insured\_relationship and age column shapisng a positive correlation with the target column.

3) Incident\_severity, incident\_state, incident\_type, insured\_hobbies, incident\_city, policy\_csl, policy\_bind\_month, property\_damage, police\_report\_avilable and auto\_makeand capital\_gain columns have the negative correlation with the target columns.

4) policy\_annual\_premimum, capital\_loss, policy\_bind\_year, incident\_hour\_of\_the\_day, auto\_year, insured\_education\_level columns have zero correlation with the target column.

5) As we can see the total\_claim\_amount and property claim graph, both the columns are directly proportional to each other signifying if one increases so does the other.

6) Total\_claim\_amount and injuri\_claim columns are building positive relation with each other.

7) vehicle claim and total\_claim\_amount will increase in the same direction indicates both are positively correlated with each other

**Data Preprocessing**

Data preprocessing is the crucial part of the machine learning since in this section, we remove the outliers, skewness, data scaling, adopting variance inflation factor for removing those columns which are highly correlated with each other and gives the same output. Adopting SMOTE for balancing the target variable

1. For outliers, Z score method is used.
2. To remove the Skewness, power transformation method is used.
3. For data scaling standard scaler technique is used.
4. In order to balance the target variable, SMOTE technique is used.

# Building Machine Learning Models

In most essential respect, it will become important to know which type of machine learning model we are going to construct. It depends on the target. Here we are embracing classification model since our target column is categorical in nature.

For classification model, some matrix we are going to find like:

\* Confusion matrix

\* Accuracy Score

\* Classification report

### Models

* Decision Tree classifier
* K Neighbors classifier
* Support vector machine classifier
* Naive bayes classifier

### For bagging and boosting:

* Random Forest classifier
* AdaBoost classifier
* Gradient Boosting classifier
* XGBoost classifier

Now we will train several Machine Learning models and compare their results. For achieving the desired results, first we will import the Classification models, Classification metrics, Ensemble methods, Cross validation score and Grid search cv for hyperparameter tuning.

Graphical user interface, text, application

Description automatically generated

In order to create ML model, it is significant to find out the best random state, the accuracy is distinct in different random state, and we need to find that particular state where the highest accuracy will be handed by model.

Graphical user interface, text, application

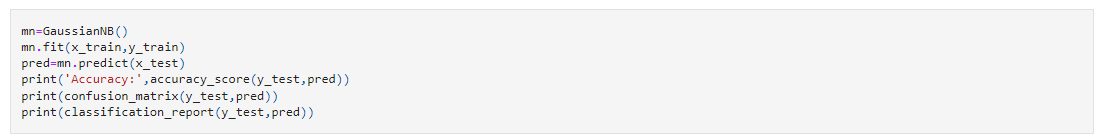
Description automatically generated

**Rectangle

Description automatically generated with low confidenceLogistic Regression**

A picture containing rectangle

Description automatically generated **K Neighbors classifier**



**Naive bayes(GussianNB)**

Shape, rectangle

Description automatically generated with medium confidence

**Decision tree classifier**

Shape, rectangle

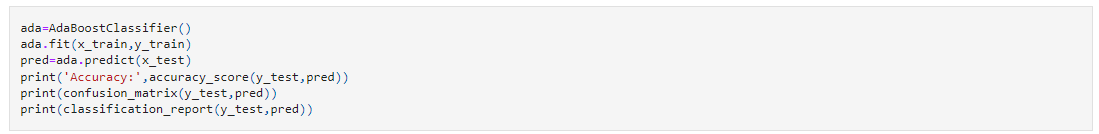
Description automatically generated with medium confidence

**Support vector machine**

Rectangle

Description automatically generated with low confidence

**Random Forest**



**AdaBoost**

Graphical user interface

Description automatically generated with medium confidence

**Gradient Boosting**

**Shape, rectangle

Description automatically generated with medium confidence**

**XG boost**

**Comparison of all the models:**

Graphical user interface, text, application, email

Description automatically generated

## Chart, bar chart Description automatically generated

### We are choosing Support vector machine classifier as its demonstrating model accuracy scores; model performance and cross validation scores are higher as compared to others. Accuracy of the model and performance of the model are directly proportional and hence better the performance of the model, more accurate are the predictions.

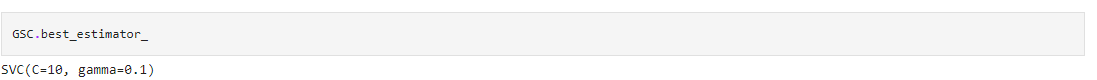
### Model accuracy is: 91.71

### Cross validation score: 88.72

### Difference: 2.99

Graphical user interface

Description automatically generated with medium confidence**Hyperparameters with Grid search CV:**

In the Machine learning model, we train the different models to achieve highest accuracy possible. Although we select one of the best models, yet there are chances for improvement. Hence, our aim is to improve the model in every possible way, therefore we can use hyperparameters by using Graphical user interface, application, Teams

Description automatically generatedGrid search CV.

Grid Search CV is the process of performing hyperparameter tuning in order to determine the optimal values for a given model. Once we set appropriate values for these hyperparameters using grid search CV, the performance of a model can improve significantly. We have performed with Grid search CV, and we got our best model Support vector machine classifier with 95.39% accuracy.

**AUC-ROC Curve**:

It is one of the most important evaluation metrics for checking any classification model’s performance. The AUC-ROC metric clearly helps to determine and tell us about the capability of a model in distinguishing the classes. The judging criteria being - Higher the Chart, line chart

Description automatically generatedAUC, 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.

## ROC AUC Score:

The ROC AUC Score is the corresponding score to the ROC AUC Curve. It is simply computed by measuring the area under the curve, which is called AUC.



We got the 91.71 % AUC-ROC score.

Now we will see the prediction of our model, it is salient step, and we eagerly want to see prediction result. If our prediction data is near or qual to the original values it will be substantial, and we will only be gratified if we achieve the best accuracy.

Graphical user interface, text, application

Description automatically generated

**Summary**

We started with the data exploration where we got a feeling for the dataset, checked about missing data and learned which features are important. During this process we used seaborn and matplotlib to do the visualizations. During the data preprocessing part, we computed missing values, converted features into numeric ones, grouped values into categories and created a few new features. Afterwards we started training 9 different machine learning models, picked one of them (SVC) and applied cross validation on it. Then we used grid search cv for tuned its performance through optimizing it’s hyperparameter values. Then we plot the AUC-ROC curve and found the AUC-ROC score. In the end we checked the original values and predicted values and got all the prediction values up to the mark.