**IIT NIT CUTOFF ANALYSIS**

*A Internship report submitted in partial fulfilment of the requirements for the Award of Degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**INFORMATION TECHNOLOGY**

**By**

**Pathivada Arun**

**Regd. No:20B91A12F0**

Under Supervision of Mr. Gundala Nagaraju

Henotic Technology Pvt Ltd, Hyderabad

(Duration: 7th July, 2022 to 6th September, 2022)

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**SAGI RAMA KRISHNAM RAJUENGINEERING COLLEGE**

(An Autonomous Institution)

Approved by AICTE, NEW DELHI and Affiliated to JNTUK, Kakinada

CHINNA AMIRAM, BHIMAVARAM,

ANDHRA PRADESH

SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE

(Autonomous)

Chinna Amiram, Bhimavaram

DEPARTMENT OF INFORMATION TECHNOLOGY



**CERTIFICATE**

This is to certify that the Summer Internship Report titled “**IIT NIT CUTOFF ANALYSIS”** is the bonafide work done by Mr PATHIVADA ARUN bearing Register number 20B91A12F0 at the end of second year second semester at Henotic Technology Pvt Ltd, Hyderabad from 07.07.2022 to 06.09.2022 in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Information Technology.

|  |  |  |
| --- | --- | --- |
| **Department Internship Coordinator** | **Dean -T & P Cell** | **Head of the Department** |

Table of Contents

[**1.0** **Introduction** 4](#_Toc110502057)

[**1.1.** **What are the different types of Machine Learning?** 4](#_Toc110502058)

[**1.2.** **Benefits of Using Machine Learning in IITS AND NITS CUTOFFS**  4](#_Toc110502059)

[**1.2.1** **AI / ML Role in IITS AND NITS CUTOFFS** 5](#_Toc110502061)

[**2.0** **IITS AND NITS CUTOFFS ANALYSIS (your internship project)** 6](#_Toc110502062)

[**2.1.** **Main Drivers for AI Auto Quote Analysis** 6](#_Toc110502063)

[**2.2.** **Internship Project - Data Link** 7](#_Toc110502064)

[**3.0** **AI / ML Modelling and Results** 8](#_Toc110502065)

[**3.1.** **Your Problem of Statement** 8](#_Toc110502066)

[**3.2.** **Data Science Project Life Cycle** 8](#_Toc110502067)

[**3.2.1** **Data Exploratory Analysis** 9](#_Toc110502068)

[**3.2.2** **Data Pre-processing** 9](#_Toc110502069)

[**3.2.2.1.** **Check the Duplicate and low variation data** 9](#_Toc110502070)

[**3.2.2.2.** **Identify and address the missing variables** 10](#_Toc110502071)

[**3.2.2.3.** **Handling of Outliers** 10](#_Toc110502072)

[**3.2.2.4.** **Categorical data and Encoding Techniques** 10](#_Toc110502073)

[**3.2.2.5.** **Feature Scaling** 10](#_Toc110502074)

[**3.2.3** **Selection of Dependent and Independent variables** 10](#_Toc110502075)

[**3.2.4** **Data Sampling Methods** 10](#_Toc110502076)

[**3.2.4.1.** **Stratified sampling** 10](#_Toc110502077)

[**3.2.4.2.** **Simple random sampling** 11](#_Toc110502078)

[**3.2.5** **Models Used for Development** 11](#_Toc110502079)

[**3.2.5.1.** **Model 01** 11](#_Toc110502080)

[**3.2.5.2.** **Model 02** 11](#_Toc110502081)

[**3.2.5.3.** **Model 03** 11](#_Toc110502082)

[**3.2.5.4.** **Model 04** 11](#_Toc110502083)

[**3.2.5.5.** **Model 10** 11](#_Toc110502084)

[**3.3.** **AI / ML Models Analysis and Final Results** 12](#_Toc110502085)

[**3.3.1** **Different Model codes** 12](#_Toc110502086)

[**3.3.2** **Random Forest Python Code** 12](#_Toc110502087)

[**3.3.3** **Extra Trees Python code** 12](#_Toc110502088)

[**4.0** **Conclusions and Future work** 13](#_Toc110502089)

[**5.0** **References** 14](#_Toc110502090)

[**6.0** **Appendices** 15](#_Toc110502091)

[**6.1.** **Python code Results** 15](#_Toc110502092)

[**6.2.** **List of Charts** 15](#_Toc110502093)

[**6.2.1** **Chart 01: Share of each category in preparatory courses** 15](#_Toc110502094)

[**6.2.2** **Chart 02: Average opening and closing rank in each institute for courses** 15](#_Toc110502095)

[**6.2.3** **Chart 03:Average opening and closing rank in top 10 institute for courses** 15](#_Toc110502096)

[**6.2.4** **Chart 04: Average opening and closing rank in each round** 16](#_Toc110502097)

**Abstract**

* This project is mainly used to predict the cutoff marks and rank in prestigious exam jee advanced and jee mains.
* In this project we take a data set from Kaggle which contains multiple rows and columns.
* Based on the specific rows like institution type,gender,program name,program duration,category,opening and closing ranks.
* Firstly clean the data using data cleaning methods in python language.
* Then divide the categorical and target variables from the data.
* And assign values to the strings using label encoder method and binary encoder.
* Split the data into test and training data by using some modules in python pandas.
* Scaling must be done on the data for getting accurate results.
* Finally prepare some machine learning models to predict the results.
* Among all the algorithms select few algos for end user

# **Introduction**

With the increasing power of computer technology, companies and institutions cannowadays store large amounts of data at reduced cost. The amount of available datais increasing exponentially and cheap disk storage makes it easy to store data thatpreviously was thrown away. There is a huge amount of information locked up indatabases that is potentially important but has not yet been explored. The growing sizeand complexity of the databases makes it hard to analyse the data manually, so it isimportant to have automated systems to support the process. Hence there is the needof computational tools able to treat these large amounts of data and extract valuableinformation.

In this context, Data Mining provides automated systems capable of processinglarge amounts of data that are already present in databases. Data Mining is used toautomatically extract important patterns and trends from databases seeking regularitiesor patterns that can reveal the structure of the data and answer business problems. DataMining includes learning techniques that fall into the field of Machine learning. Thegrowth of databases in recent years brings data mining at the forefront of new businesstechnologies.

A key challenge for the insurance industry is to charge each customer an appropriate price for the risk they represent. Risk varies widely from customer to customer and a deep understanding of different risk factors helps predict the likelihood and cost of insurance claims. The goal of this program is to see how well various statistical methods perform in predicting auto Insurance claims based on the characteristics of the driver, vehicle and driver / vehicle coverage details.

A number of factors will determine BI claims prediction among them a driver's age, past accident history, and domicile, etc. However, this contest focused on the relationship between claims and vehicle characteristics well as other characteristics associated with the auto insurance policies.

## **What are the different types of Machine Learning?**

* **Supervised learning**

As its name suggests, is based on supervision. It means in the supervised learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output. Here, the labelled data specifies that some of the inputs are already mapped to the output. More preciously, we can say; first, we train the machine with the input and corresponding output, and then we ask the machine to predict the output using the test dataset.

* **Unsupervised learning**

Unsupervised learning is different from the Supervised learning technique; as its name suggests, there is no need for supervision. It means, in unsupervised machine learning, the machine is trained using the unlabeled dataset, and the machine predicts the output without any supervision.

* **Reinforcement learning**

**Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences, and improving its performance.** Agent gets rewarded for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards

* **Semi supervised learning**

**Semi-Supervised learning is a type of Machine Learning algorithm that lies between Supervised and Unsupervised machine learning**. It represents the intermediate ground between Supervised (With Labelled training data) and Unsupervised learning (with no labelled training data) algorithms and uses the combination of labelled and unlabeled datasets during the training period.

## **Benefits of Using Machine Learning in iit &nit cutoff prediction**

* Students are able to assume their college according to their rank in jee advanced and mains exams.
* By using machine learning ,we can predict for accurate results compared to another types of predictions.
* There are different types of machine learning models for prediction ,either classification or regression type of predictions.

### **AI / ML Role in iit &nit cutoffs**

Machine Learning is a sub-set of artificial intelligence where computer algorithms are used to autonomously learn from data. Machine learning (ML) is getting more and more attention and is becoming increasingly popular in many other industries.

# **2.0 IIT AND NIT CUTOFFS**

* Nowadays ,competition for getting good college after intermediate increases gradually ,so there isa need for predicting college based on their rank and category.
* Some of the fields for prediction are id,year,institute \_type,quota,pool,institute\_sort,program\_name,program\_duration,degree\_short,category,opening rank,closing rank,is\_preperatory.

## **Main Drivers for iit &nit cutoff analysis**

Predictive modelling allows for simultaneous consideration of many variablesand quantification of their overall effect.

## **Internship Project - Data Link**

The internship project data has taken from Kaggle and the link is <https://www.kaggle.com/datasets/rumbleftw/iit-nit-data?select=data.csv>

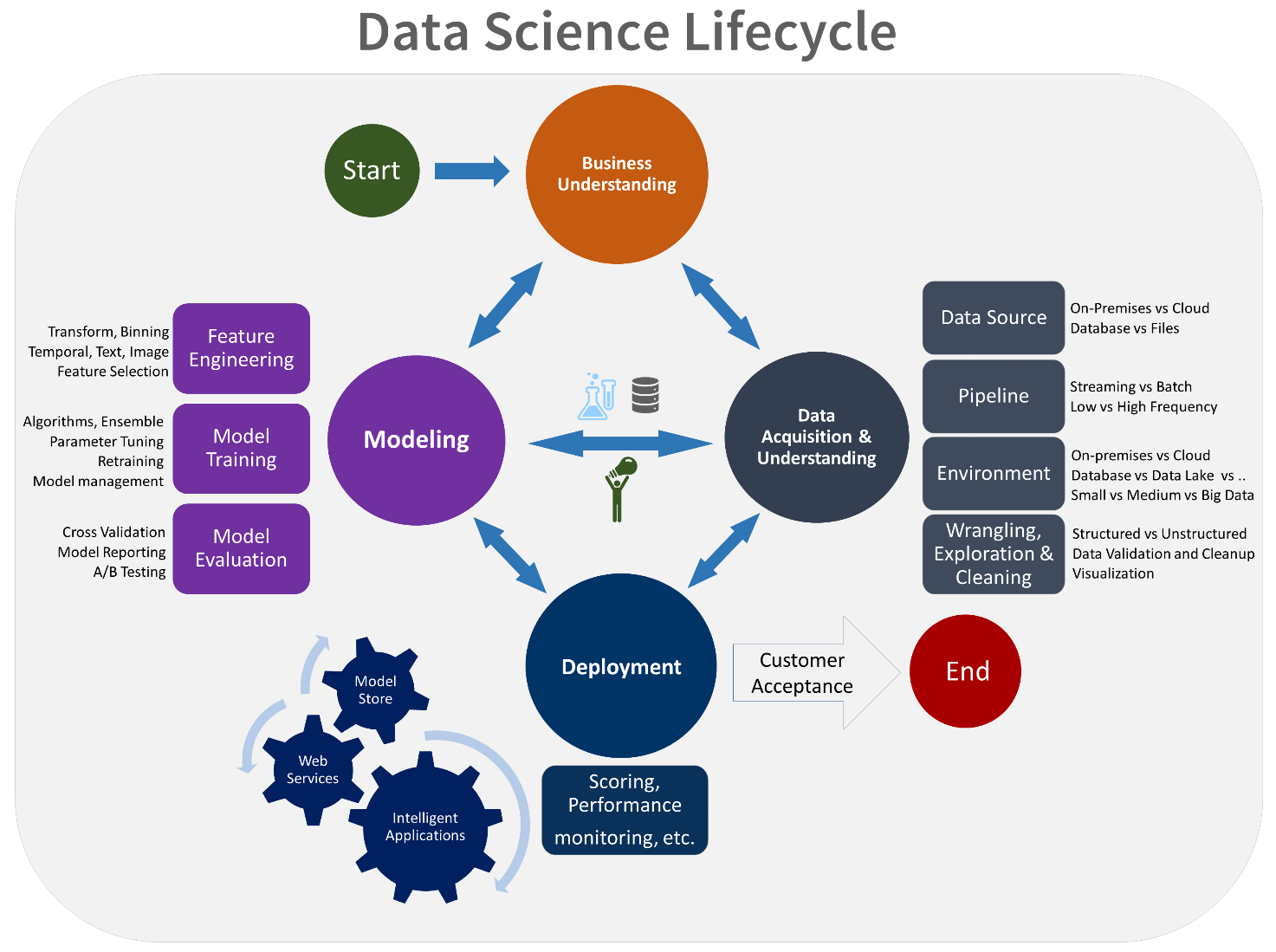
# **AI / ML Modelling and Results**

## **Your Problem of Statement**

* Joint Entrance Examination – Main (JEE-Main), formerly All India Engineering Entrance Examination (AIEEE), is an Indian standardised computer-based test for admission to various technical undergraduate programs in engineering, architecture, and planning across colleges in India. The exam is conducted by the JEE Apex Board for Admission for B.Tech, B.Arch, etc. programs in the premier technical institutes such as the National Institutes of Technology and Indian Institutes of Information Technology are based on the rank secured in the JEE-Main. It is usually conducted twice every year.
* In this dataset, the year-wise distribution of cutoffs for various IITs and NITs are collected and organized on the basis of stream, category, quota etc
* We had to predict the correct college and branch for their rank.

## **Data Science Project Life Cycle**

Data Science is a multidisciplinary field of study that combines programming skills, domain expertise and knowledge of statistics and mathematics to extract useful insights and knowledge from data.



### **Data Exploratory Analysis**

Exploratory data analysis has been done on the data to look for relationship and correlation between different variables and to understand how they impact or target variable.

### **Data Pre-processing**

We removed variables which does not affect our target variable(is\_preperatory)as they may add noise and also increase our computation time,we checked the data for anomalous data points and outliers. We did principal component analysis on the data set to filter out unnecessary variables and to select only the important variables which have greater correlation with our target variable.

### **Check the Duplicate and low variation data**

Check duplicated values in the data by using some of the code in python syntax for finding duplicates is:Dataframe.duplicated(subset = None,keep = ‘first’)

### **Identify and address the missing variables**

* In given data set there are not having any missing values .we can check missing data in python by using the following syntax:Dataframe.isnull().sum()

### **Handling of Outliers**

* Outliers are data points which are far away from all the data points in the data set.those are handled by using some method in machine learning
* There are three types of outliers

1. Global outliers
2. Contextual outliers
3. Collective outliers

* In my data set there are no outliers.

### **Categorical data and Encoding Techniques**

* Given data set there are different types of data like categorical,numerical etc.we need to change the all categorical data by using some encoding techniques
* Some of encoding techniques are

1. Label encoder
2. Binary encoder

* By using label encoder we can able to change the string values into integer values from 0 to n
* By using binary encoder we can able to change only two types of strings only.

### **Feature Scaling**

* Feature Scaling is **a technique to standardize the independent features present in the data in a fixed range**. It is performed during the data pre-processing.
* There are two types of scaling techniques in machine learning

1. Standard scaling
2. Minmax scaling

* By using those scaling methods we can able to change the integers into some range .defaultly it changes between 0 and 1.

### **Selection of Dependent and Independent variables**

The dependent or target variable here is\_preperatory Target which tells us a particular person getting their dream college or not the target variable is selected based on our business problem and what we are trying to predict.

The independent variables are selected after doing exploratory data analysis and we used Boruta to select which variables are most affecting our target variable.

### **Data Sampling Methods**

The data we have is highly unbalanced data so we used some sampling methods which are used to balance the target variable so we our model will be developed with good accuracy and precision. We used three Sampling methods

### **Stratified sampling**

Stratified sampling randomly selects data points from majority class so they will be equal to the data points in the minority class. So, after the sampling both the class will have same no of observations.

It can be performed using strata function from the library sampling.

### **Simple random sampling**

Simple random sampling is a sampling technique where a set percentage of the data is selected randomly. It is generally done to reduce bias in the dataset which can occur if data is selected manually without randomizing the dataset.

We used this method to split the dataset into train dataset which contains 70% of the total data and test dataset with the remaining 30% of the data.

### **Models Used for Development**

* DecisionTreeClassifier()
* DecisionTreeClassifier(max\_features = ‘auto’)
* ExtraTreeClassifier(random\_state=722370420)
* KNeighborsClassifier()
* SVC(probability=True)
* DecisionTreeClassifier(random\_state=147384717...)
* DecisionTreeRegressor(criterion='friedman\_ms...)
* LGBMClassifier()
* GaussianNB()
  + - 1. **DecisionTreeClassifier()**
* Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome
* In a Decision tree, there are two nodes, which are the **Decision Node** and**Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
* The decisions or the test are performed on the basis of features of the given dataset
* It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions***.***
* It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

### **ExtraTreeClassifier()**

* **Extremely Randomized Trees Classifier(Extra Trees Classifier)** is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output it’s classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest.
* Each Decision Tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, Each tree is provided with a random sample of k features from the feature-set from which each decision tree must select the best feature to split the data based on some mathematical criteria (typically the Gini Index). This random sample of features leads to the creation of multiple de-correlated decision trees
* To perform feature selection using the above forest structure, during the construction of the forest, for each feature, the normalized total reduction in the mathematical criteria used in the decision of feature of split (Gini Index if the Gini Index is used in the construction of the forest) is computed. This value is called the Gini Importance of the feature. To perform feature selection, each feature is ordered in descending order according to the Gini Importance of each feature and the user selects the top k features according to his/her choice.

### KneighboursClassifier()

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
* It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

### **SVC(support vector machine)**

* Support vector machines (SVMs) are powerful yet flexible supervised machine learning methods used for classification, regression, and, outliers’ detection. SVMs are very efficient in high dimensional spaces and generally are used in classification problems. SVMs are popular and memory efficient because they use a subset of training points in the decision function.
* The main goal of SVMs is to divide the datasets into number of classes in order to find a **maximum marginal hyperplane (MMH)** which can be done in the following two steps
* Support Vector Machines will first generate hyperplanes iteratively that separates the classes in the best way.
* After that it will choose the hyperplane that segregate the classes correctly.

### DecisionTreeRegressor

* Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

### LGBM Classifier()

* LightGBM is a gradient boosting framework based on decision trees to increases the efficiency of the model and reduces memory usage.
* It uses two novel techniques: Gradient-based One Side Sampling and Exclusive Feature Bundling (EFB) which fulfills the limitations of histogram-based algorithm that is primarily used in all GBDT (Gradient Boosting Decision Tree) frameworks. The two techniques of GOSS and EFB described below form the characteristics of LightGBM Algorithm. They comprise together to make the model work efficiently and provide it a cutting edge over other GBDT frameworks
* Gradient-based One Side Sampling Technique for LightGBM:
* Different data instances have varied roles in the computation of information gain. The instances with larger gradients(i.e., under-trained instances) will contribute more to the information gain. GOSS keeps those instances with large gradients (e.g., larger than a predefined threshold, or among the top percentiles), and only randomly drop those instances with small gradients to retain the accuracy of information gain estimation. This treatment can lead to a more accurate gain estimation than uniformly random sampling, with the same target sampling rate, especially when the value of information gain has a large range.

### GuassianNB()

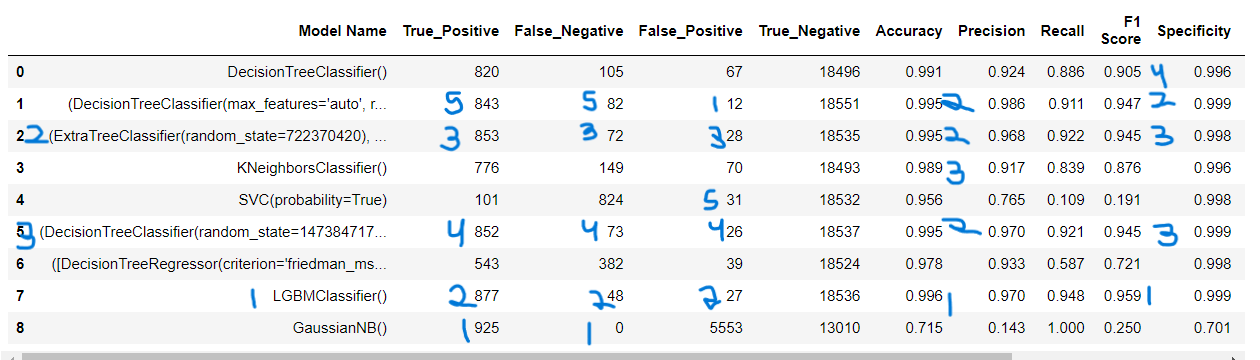
* Naive Bayes is a probabilistic machine learning algorithm that can be used in several classification tasks. Typical applications of Naive Bayes are classification of documents, filtering spam, prediction and so on. This algorithm is based on the discoveries of Thomas Bayes and hence its name.
* The name “Naïve” is used because the algorithm incorporates features in its model that are independent of each other. Any modifications in the value of one feature do not directly impact the value of any other feature of the algorithm. The main advantage of the Naïve Bayes algorithm is that it is a simple yet powerful algorithm.
* It is based on the probabilistic model where the algorithm can be coded easily, and predictions did quickly in real-time. Hence this algorithm is the typical choice to solve real-world problems as it can be tuned to respond to user requests instantly. But before we dive deep into Naïve Bayes and Gaussian Naïve Bayes, we must know what is meant by conditional probability.

# **Conclusions and Future work**

The model results in the following order by considering the model accuracy, F1 score and RoC AUC score.

1. **LGBM Classifier** with Stratified and Random Sampling
2. **Extra Trees Classifier** with Simple Random Sampling
3. **Decision Tree Classifier** with Simple Random Sampling

We recommend model - **LGBM Classifier** with Stratified and Random Sampling technique as a best fit for the give n BI claims dataset. We considered Random Forest because it uses bootstrap aggregation which can reduce bias and variance in the data and can leads to good predictions with claims dataset.

****

The future work to evaluate the “to predict results” in auto Insurance by using classification methods.

# **References**

* Geeks for geeks website
* Kaggle website for data set
* Some other websites for machine learning concepts

# **Appendices**

## **Python code Results**

Final result:

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import ExtraTreesClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.svm import SVC

from sklearn.ensemble import BaggingClassifier

from sklearn.ensemble import GradientBoostingClassifier

import lightgbm as lgb

# Create objects of classification algorithm with default hyper-parameters

ModelDC = DecisionTreeClassifier()

ModelRF = RandomForestClassifier()

ModelET = ExtraTreesClassifier()

ModelKNN = KNeighborsClassifier(n\_neighbors=5)

ModelSVM = SVC(probability=True)

modelBAG = BaggingClassifier(base\_estimator=None, n\_estimators=100, max\_samples=1.0, max\_features=1.0,bootstrap=True,bootstrap\_features=False,oob\_score=False,warm\_start=False,n\_jobs=None, random\_state=None, verbose=0)

ModelGB = GradientBoostingClassifier(loss='deviance', learning\_rate=0.1, n\_estimators=100, subsample=1.0,criterion='friedman\_mse',min\_samples\_split=2,min\_samples\_leaf=1,min\_weight\_fraction\_leaf=0.0, max\_depth=3, min\_impurity\_decrease=0.0,min\_impurity\_split=None, init=None, random\_state=None,max\_features=None,verbose=0,max\_leaf\_nodes=None, warm\_start=False, validation\_fraction=0.1, n\_iter\_no\_change=None, tol=0.0001, ccp\_alpha=0.0)

ModelLGB = lgb.LGBMClassifier()

ModelGNB = GaussianNB()

# Evalution matrix for all the algorithms

MM = [ModelDC, ModelRF, ModelET, ModelKNN, ModelSVM, modelBAG, ModelGB, ModelLGB, ModelGNB]

for models in MM:

# Fit the model

models.fit(x\_train, y\_train)

# Prediction

y\_pred = models.predict(x\_test)

y\_pred\_prob = models.predict\_proba(x\_test)

# Print the model name

print('Model Name: ', models)

# confusion matrix in sklearn

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import classification\_report

# actual values

actual = y\_test

# predicted values

predicted = y\_pred

# confusion matrix

matrix = confusion\_matrix(actual,predicted, labels=[1,0],sample\_weight=None, normalize=None)

print('Confusion matrix : \n', matrix)

# outcome values order in sklearn

tp, fn, fp, tn = confusion\_matrix(actual,predicted,labels=[1,0]).reshape(-1)

print('Outcome values : \n', tp, fn, fp, tn)

# classification report for precision, recall f1-score and accuracy

C\_Report = classification\_report(actual,predicted,labels=[1,0])

print('Classification report : \n', C\_Report)

# calculating the metrics

sensitivity = round(tp/(tp+fn), 3);

specificity = round(tn/(tn+fp), 3);

accuracy = round((tp+tn)/(tp+fp+tn+fn), 3);

balanced\_accuracy = round((sensitivity+specificity)/2, 3);

precision = round(tp/(tp+fp), 3);

f1Score = round((2\*tp/(2\*tp + fp + fn)), 3);

# Matthews Correlation Coefficient (MCC). Range of values of MCC lie between -1 to +1.

# A model with a score of +1 is a perfect model and -1 is a poor model\

from math import sqrt

mx = (tp+fp) \* (tp+fn) \* (tn+fp) \* (tn+fn)

MCC = round(((tp \* tn) - (fp \* fn)) / sqrt(mx), 3)

print('Accuracy :', round(accuracy\*100, 2),'%')

print('Precision :', round(precision\*100, 2),'%')

print('Recall :', round(sensitivity\*100,2), '%')

print('F1 Score :', f1Score)

print('Specificity or True Negative Rate :', round(specificity\*100,2), '%' )

print('Balanced Accuracy :', round(balanced\_accuracy\*100, 2),'%')

print('MCC :', MCC)

# Area under ROC curve

from sklearn.metrics import roc\_curve, roc\_auc\_score

print('roc\_auc\_score:', round(roc\_auc\_score(actual, predicted), 3))

# ROC Curve

from sklearn.metrics import roc\_auc\_score

from sklearn.metrics import roc\_curve

logit\_roc\_auc = roc\_auc\_score(actual, predicted)

fpr, tpr, thresholds = roc\_curve(actual, models.predict\_proba(x\_test)[:,1])

plt.figure()

# plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit\_roc\_auc)

plt.plot(fpr, tpr, label= 'Classification Model' % logit\_roc\_auc)

plt.plot([0, 1], [0, 1],'r--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver operating characteristic')

plt.legend(loc="lower right")

plt.savefig('Log\_ROC')

plt.show()

print('-----------------------------------------------------------------------------------------------------')

#----------------------------------------------------------------------------------------------------------

new\_row = {'Model Name' : models,

'True\_Positive' : tp,

'False\_Negative' : fn,

'False\_Positive' : fp,

'True\_Negative' : tn,

'Accuracy' : accuracy,

'Precision' : precision,

'Recall' : sensitivity,

'F1 Score' : f1Score,

'Specificity' : specificity,

'MCC':MCC,

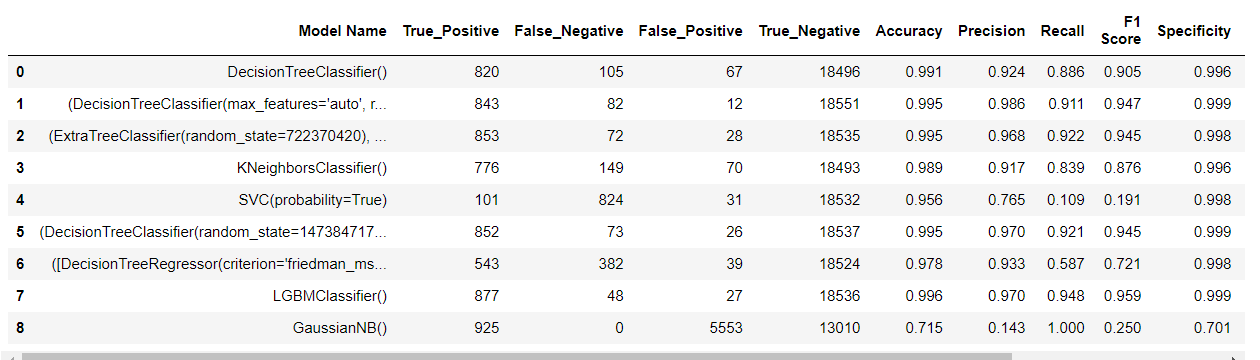
'ROC\_AUC\_Score':roc\_auc\_score(actual, predicted),

'Balanced Accuracy':balanced\_accuracy}

EMResults = EMResults.append(new\_row, ignore\_index=True)

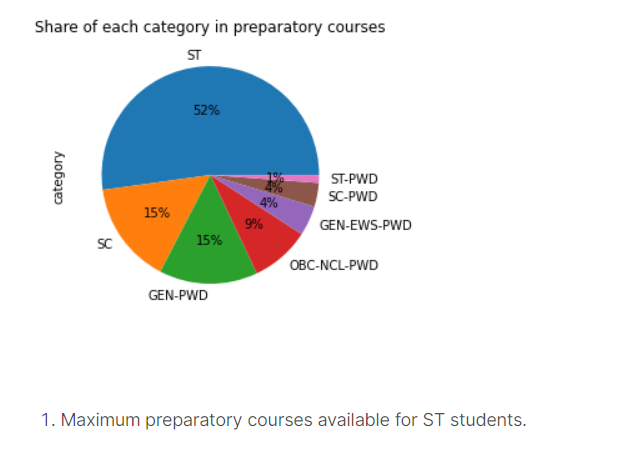
#----------------------------------------------------------------------------------------------------------

Output:

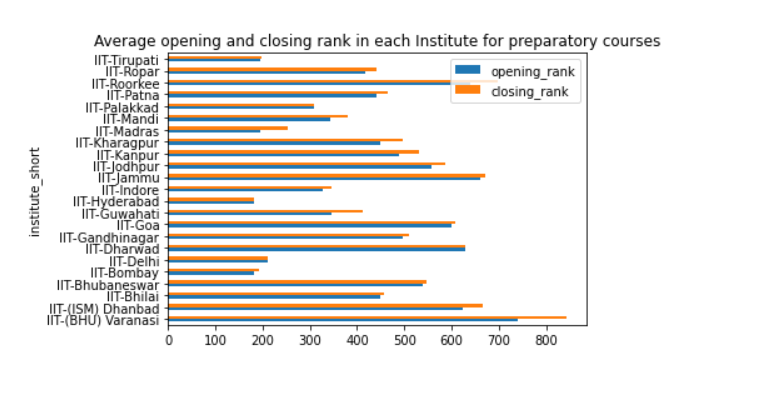


## **List of Charts**

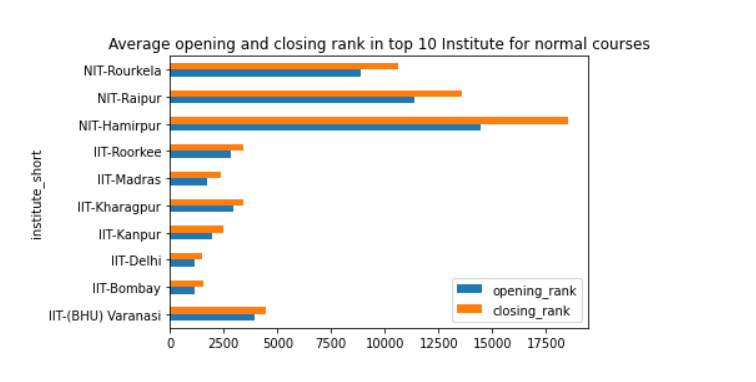
### **Chart 01: Share of each category in preparatory courses**



### **Chart 02: average opening and closing rank in each institute for preparatory courses**



### **Chart 03:****average opening and closing rank in top 10 institute for normal courses**



### **Chart 04:average opening and closing rank in each round**

