PROJECT REPORT ON

College Predictor System for IIT's and NIT's

Bachelor of Technology

COMPUTER SCIENCE SPECIALIZATION IN ARTIIFICIAL INTELLIGENCE

GURUGRAM UNIVERSITY, GURUGRAM, HARYANA, INDIA

Submitted by -

Satyam Pandey - 191030050061
Amanjeet Saroha - 191030050028
Kunal Singhal - 191030050032

February, 2023

ACKNOWLEDGEMENT

We take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guide, ,
for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We also take this opportunity to express a deep sense of gratitude to **MR. ASHOK KHANNA.** We would like to thank our friends and family for the support and encouragement they have given us during the course of our work.

Abstract :

"Prediction of the colleges is a very hectic task for the individual. So, we provide a model which predict the colleges as well as the courses with stream by evaluating the features data such as Institute type, Round Number, Quota, Pool, Category, Opening Rank and closing Rank."

Context :

Joint Entrance Examination – Main (JEE-Main), formerly All India Engineering Entrance Examination (AIEEE), is an Indian standardized 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 BTECH, 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.

IITs and NITs :

The Indian Institutes of Technology (IITs) are the globally appreciated engineering and technological institutes in India. IITs have maintained quality education and internationally acclaimed research facilities. IIT JEE Exam is the most popular engineering admission entrance test conducted in India.

National Institute of Technology (NITs) are premier engineering colleges in India offering admission to degree courses at both undergraduate and postgraduate level.

Introduction

College Predictor System for IIT's and NIT's is a web based application system in which students can register their marks along with their personal information. This helps to predict their admissions in colleges. Administrator can add the college details and the batch details. Using this Application, the entrance seat allotment becomes easier and efficient. The main advantage of the project is the computerization of the entrance seat allotment process. Administrator has the power for the allotment. Admin can add the allotted seats into a file and the details are saved into the system. The total time for the entrance allotment becomes lower and the allotment process becomes faster. It helps students to make right decisions for choosing their college. In which students can register with their personal as well as marks details to prediction the admission in colleges and the administrator can allot the seats for the students. Administrator can add the college details and the batch details. Using this Application, the entrance seat allotment became easier and can be implemented using system. The main advantage of the project is the computerization of the entrance seat allotment process. Administrator has the power for the allotment. Admin can add the allotted seats into a file and the details are saved into the system. The total time for the entrance allotment became lesser and the allotment process became faster. It helps student for making decision for choosing a right college.

1.1 SCOPE OF THE PROPOSED WORK

In this proposed project we designed a protocol or a model to predict the colleges. This system is capable of providing most of the essential features required to predict the best colleges around the country. As technology changes, it becomes difficult to track the Modeling and pattern of college admission. With the rise of machine learning, artificial intelligence and other relevant fields of information technology, it becomes feasible to automate this process and to save some of the intensive amount of labor that is put into predicting the colleges which are best for the individual.

2. SOFTWARE AND HARDWARE REQUIREMENT

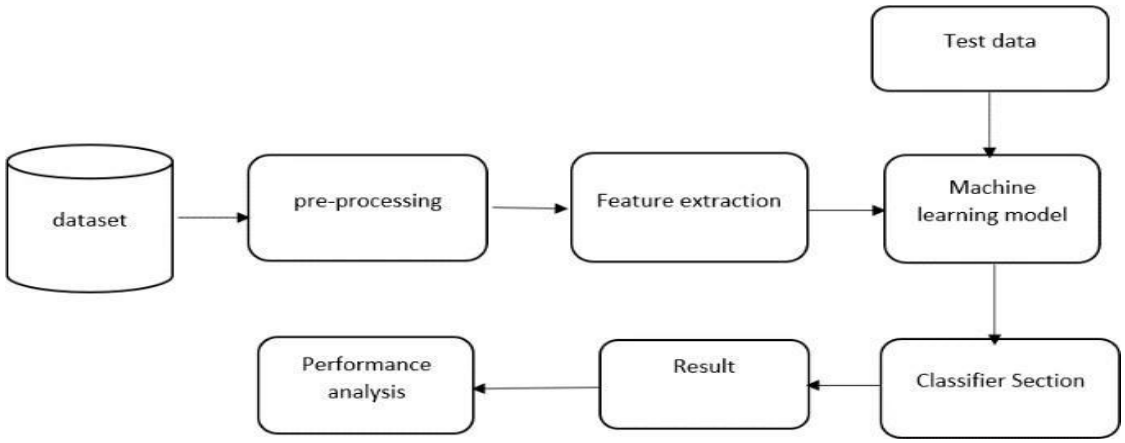
2.1 Hardware

- OS – Windows 7, 8 and 10 (32 and 64 bit)
- RAM – 4GB 2.2

2.2 Software

- Python
- Kaggle

3. System Architecture



4. Packages and Library :

Which are being used for data exploration, pre- processing and for using random forest and decision tree algorithms are :

- **NumPy :** For simple arrays.
- **Pandas :** For reading the file.
- **SciKit :** Learn- for pre-processing.
- **Matplotlib or Seaborn :** For plotting and representing confusion matrix colour format.
- **Warnings :** To neglect the warnings appear in the output.
- **OS :** For Importing the dataset from the system

```
#imports

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

+ Code + Markdown

[2]:

```
import warnings
warnings.filterwarnings('ignore')
```

5. MODULES

- Data collection
- Data pre-processing
- Data exploration
- EDA and Data Visualization
- Model Generation
- Model Prediction
- Evaluation model

5.1 Data Collection

In this Project we used the data set of the IIT's and NIT's admission portal, Which was collected by this portal only and pre-processed by the model.

This step is concerned with selecting the subset of all available data that we will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called labelled data.

[4]:
```
# Read data in a dataframe

# A DataFrame is a data structure that organizes data
#into a 2-dimensional table of rows and columns, much like a spreadsheet.

df = pd.read_csv("/kaggle/input/iit-and-nit-colleges-admission-criteria/data.csv")
df.head()
```

[4]:

	year	institute_type	round_no	quota	pool	institute_short	program_name	program_duration	degree_short	category	opening_rank	closing_rank
0	2016	IIT	6	AI	Gender-Neutral	IIT-Bombay	Aerospace Engineering	4 Years	B.Tech	GEN	838	1841
1	2016	IIT	6	AI	Gender-Neutral	IIT-Bombay	Aerospace Engineering	4 Years	B.Tech	OBC-NCL	408	1098
2	2016	IIT	6	AI	Gender-Neutral	IIT-Bombay	Aerospace Engineering	4 Years	B.Tech	SC	297	468
3	2016	IIT	6	AI	Gender-Neutral	IIT-Bombay	Aerospace Engineering	4 Years	B.Tech	ST	79	145
4	2016	IIT	6	AI	Gender-Neutral	IIT-Bombay	Aerospace Engineering	4 Years	B.Tech	GEN-PWD	94	94

5.2 Data Pre-processing

Pre-processing is the process of three important and common steps as follows:

Formatting : It is the process of putting the data in a legitimate way that it would be suitable to work with. Format of the data files should be formatted according to the need. Most recommended format is .csv files.

Cleaning : Data cleaning is a very important procedure in the path of data science as it constitutes the major part of the work. It includes removing missing data and complexity with naming category and so on. For most of the data scientists, Data Cleaning continues of 80% of work.

Sampling : This is the technique of analyzing the subsets from whole large datasets, which could provide a better result and help in understanding the behavior and pattern of data in an integrated way.

[15]:
```
# Various degrees

Degree = pd.DataFrame(df["degree_short"].unique(), columns = {'Degree':0})
Degree
```

[15]:

	Degree
0	B.Tech
1	BSc
2	B.Tech + M.Tech (IDD)
3	Int MSc.
4	B.Arch
5	Int M.Tech
6	B.Pharm
7	B.Pharm + M.Pharm
8	BS + MS (IDD)
9	Int Msc.
10	B.Plan
11	Btech + M.Tech (IDD)
12	BSc + MSc (IDD)

[16]:
```
# Correcting the values

df.loc[df["degree_short"] == "Int Msc.", "degree_short"] = "Int MSc."
```

[17]:
```
df["degree_short"].nunique()
```

[17]: 12

5.3 Data Exploration

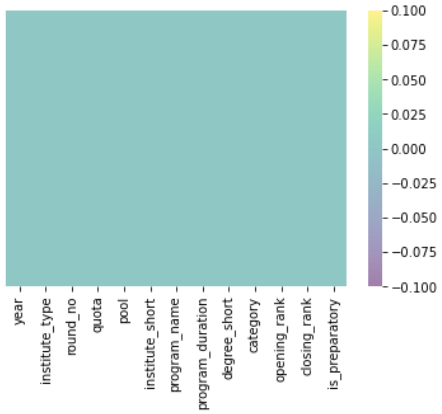
[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64958 entries, 0 to 64957
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   year              64958 non-null  int64 
 1   institute_type    64958 non-null  object
 2   round_no          64958 non-null  int64 
 3   quota             64958 non-null  object
 4   pool              64958 non-null  object
 5   institute_short   64958 non-null  object
 6   program_name      64958 non-null  object
 7   program_duration  64958 non-null  object
 8   degree_short      64958 non-null  object
 9   category          64958 non-null  object
 10  opening_rank      64958 non-null  int64 
 11  closing_rank      64958 non-null  int64 
 12  is_preparatory    64958 non-null  int64 
dtypes: int64(5), object(8)
memory usage: 6.4+ MB
```

[19]:
```
# Checking the null values in dataset

sns.heatmap(df.isnull(),yticklabels=False,cmap='viridis', alpha = 0.5)
```

[19]: <AxesSubplot:>



[12]:
```
# Unique values in pool

Pool = pd.DataFrame(df["pool"].unique())
Pool
```

[12]:

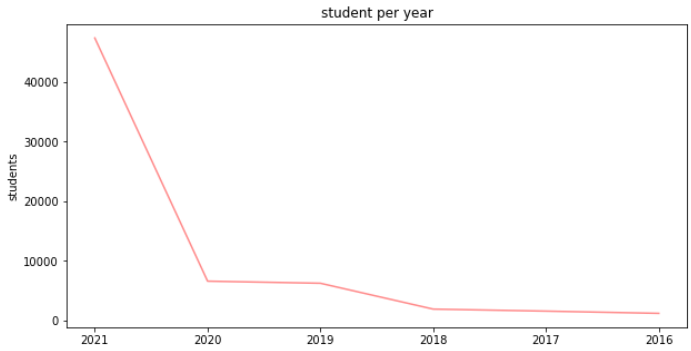
	0
0	Gender-Neutral
1	Female-Only

5.4 Exploratory Data Analysis and Visualization

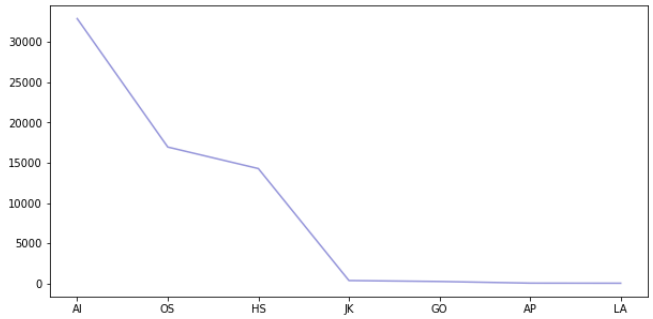
Data Visualization is the method of representing the data in a graphical and pictorial way, data scientists depict a story by the results they derive from analyzing and visualizing the data. The best tool used is Tableau which has many features to play around with data and fetch wonderful results.

```
plt.figure(figsize=(10,5))
plt.title('student per year')
sns.lineplot(x=['2021','2020','2019','2018','2017','2016'],y=df['year'].value_counts(), color = "#ff1a1a"
plt.ylabel('students')
```

[90]: `Text(0, 0.5, 'students')`

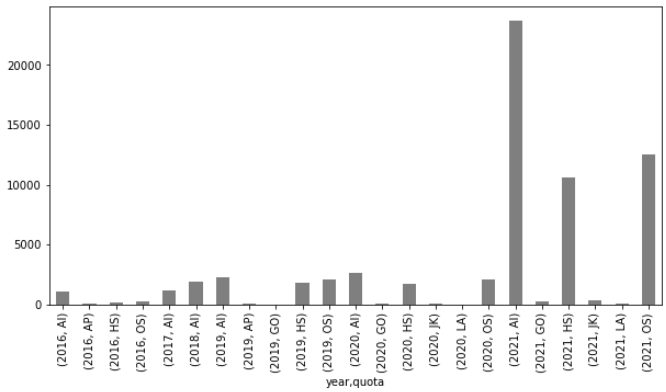


[92]:
```
plt.figure(figsize=(10,5))
plt.plot(max_quota, color = "#2e2eb8", alpha = 0.5)
plt.show()
```



[93]:
```
#Yearly quota study

plt.figure(figsize=(10,5))
year_club = df.groupby(['year', 'quota']).size().plot(kind = 'bar', color = "#000000", alpha = 0.5)
```



2 What is the most optimum Opening and closing rank in overall years??

---------- **Opening Ranks** ----------

[94]:
```
avg_opening_rank = df['opening_rank'].mean(axis = 0)
avg_open_rank = round(avg_opening_rank)
print("Average opening rank over the years has been - ", avg_open_rank)
```

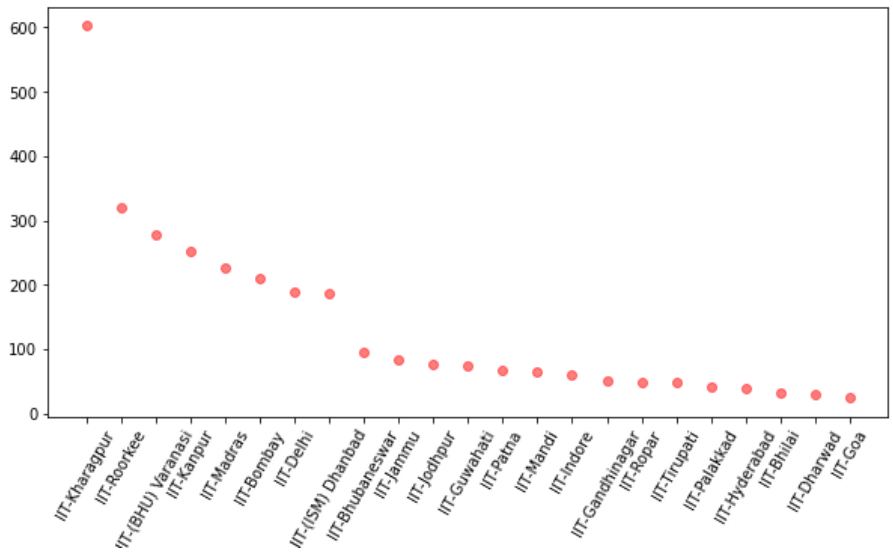
```
Average opening rank over the years has been -  8260
```

[95]:
```
max_opening_rank = df['opening_rank'].max(axis = 0)
max_open_rank = round(max_opening_rank)
print("Max opening rank over the years has been - ", max_open_rank)
```

```
Max opening rank over the years has been -  1082601
```

[100]:
```
# Which universities/colleges provide preparatory courses (represented by 1)

plt.figure(figsize=(10,5))
prep_true = df.loc[df['is_preparatory'] == 1, 'institute_short'].value_counts()
plt.plot(prep_true, 'o', color = "#ff0000", alpha = 0.5)
plt.xticks(rotation = 60)
plt.show()
```



5.5 Feature extraction

Feature extraction is the process of studying the behavior and pattern of the analyzed data and draw the features for further testing and training. Finally, our models are trained using the Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random forest. These algorithms are very popular in text classification tasks.

[106]:
```
# Changing Institute type

Pred_College['institute_type'] = [0 if x == 'IIT' else 1 for x in Pred_College['institute_type']]
Pred_College['institute_type'].unique()
```

[106… array([0, 1])

[107]:
```
# importing library for encoding

from sklearn.preprocessing import LabelEncoder
```

[108]:
```
# Labeling the quota values

le = LabelEncoder()
Pred_College['quota'] = le.fit_transform(Pred_College['quota'])
Pred_College['quota'].unique()
```

[108… array([0, 3, 6, 1, 2, 4, 5])

```
# Selecting the target data

y = Pred_College[['institute_short', 'program_name', 'degree_short']]
```

```
# Selecting the featured data

X = Pred_College[['institute_type', 'round_no', 'quota', 'pool',
       'category', 'opening_rank', 'closing_rank']]
```

💔 Splitting the training and testing data

```
# Spilliting the data into training and testing dataset

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.8)
```

5.6 Model Generation

```
# Importing library for mdoel generation

from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
```

```
model.fit(X_train, y_train)
```

[116… RandomForestClassifier()

5.7 Model Prediction

```
# predicting the dataset

y_pred = model.predict(X_test)
y_pred
```

[117… array([['NIT-Calicut', 'Electronics and Communication Engineering',
'B.Tech'],
['IIT-Mandi', 'Computer Science and Engineering', 'B.Tech'],
['NIT-Durgapur', 'Chemical Engineering', 'Btech + M.Tech (IDD)'],
...,
['NIT-Hamirpur', 'Engineering Physics', 'B.Tech'],
['NIT-Patna', 'Architecture', 'B.Arch'],
['NIT-Rourkela', 'Electrical and Electronics Engineering',
'B.Tech']], dtype=object)

[118]:
```
# prediction

Prediction = pd.DataFrame(model.predict([[0, 6, 0, 0, 0, 838, 1841], [1, 6, 0, 0, 0, 1000, 1841]]),
Prediction
```

[118…

	College	Branch	Degree
0	IIT-Bombay	Aerospace Engineering	B.Tech
1	NIT-Warangal	Computer Science and Engineering	B.Tech

5.8 Evaluation model

Model Evaluation is an essential part of the model development process. It helps to find the best model that represents our data and how well the selected model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can effortlessly generate overoptimistically and over fitted models. To avoid overfitting, evaluation methods such as hold out and cross-validations are used to test to evaluate model performance. The result will be in the visualized form. Representation of classified data in the form of graphs. Accuracy is well-defined as the proportion of precise predictions for the test data. It can be calculated easily by mathematical calculation i.e. dividing the number of correct predictions by the number of total predictions.

[67]:
```
# total accuracy of the model

Mean_Accuracy = (Accuracy1 + Accuracy2 + Accuracy3) / 3
Mean_Accuracy
```

[67]: 0.8193247126436781

Here we evaluate our model and find that it is really good and a realistic model. Firstly, we extract the individual Accuracy of the target data and then conclude the mean accuracy of the target data.

Accuracy of Institute Name - 75.3%

Accuracy of Program Name - 75.6%

Accuracy of degree - 93.8%

After calculating the individual accuracy, we will find out the total or mean accuracy of the model.

Accuracy of the Model - 81.6%

---

6. ALGORITHM :

6.1 Random Forest

Random forest is a supervised machine learning algorithm based on ensemble learning. Ensemble learning is an algorithm where the predictions are derived by assembling or bagging different models or similar model multiple times. The random forest algorithm works in a similar way and uses multiple algorithm i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

6.1.1 Advantages of using random forest :

The random forest algorithm is not biased and depends on multiple trees where each tree is trained separately based on the data, therefore biasedness is reduced overall. It's a very stable algorithm. Even if a new data point is introduced in the dataset it doesn't affect the overall algorithm rather affect the only a single tree. It works well when one has both

categorical and numerical features. The random forest algorithm also works well when data possess missing values, or when it's not been scaled properly. Thus, using this Random forest algorithm and decision trees algorithm we have extracted the accurate percentage of detection of fraud from the given dataset by studying its behavior. A confusion matrix is basically a summary of prediction results or a table which is used to describe the performance of the classifier on a set of test data where true values are known. It provides visualization of an algorithm's performance and allows easy identification of classes. Thus, resulting in the computing of most performance measures by giving insights not only the errors being made by the classification model but also tells the type of errors being made. Trained Data and Testing Data is represented in a confusion matrix which portrays :

- **TP :** True Positive which denotes the real data where customers are subjected to fraud and are used for training and were accurately predicted.
- **TN** : True Negative denotes the data which was not predicted and doesn't match with the data which was subjected to the fraud.
- **FP** : False Positive is predicted but there is no possibility of the data to be subjected to the fraud.
- **FN :** False Negative is not predicted but there is an actual possibility of the data who is subjected to fraud.

7. CONCLUSION

Hence, we have acquired the result of an accurate value of Prediction of college's i.e. 0.8193247126436781 (81.9%) using a random forest algorithm with new enhancements. In comparison to existing modules, this proposed module is applicable for the larger dataset and provides more accurate results. The Random forest algorithm will provide better performance with many training data, but speed during testing and application will still suffer. Usage of more pre-processing techniques would also assist.

8. REFERENCES

1. **This data is provided by scraping the website : *https://cutoffs.iitr.ac.in/***
2. **Modules and Libraries : *https://scikit-learn.org/stable/***
3. **Also help from *https://Youtube.com***