International Journal of Computer Applications Technology and Research
Volume 8–Issue 09, 379-384, 2019, ISSN:-2319–8656

Engineering College Admission Preferences Based on Student Performance

Dhruvesh Kalathiya
Student, M.Sc.(BDA)
MIT-WPU
Pune, India

Rashmi Padalkar
Student, M.Sc.(BDA)
MIT-WPU
Pune, India

Rushabh Shah
Student, M.Sc.(BDA)
MIT-WPU
Pune, India

Sachin Bhoite
Assistant Professor
Department of Computer Science
Faculty of Science
MIT-WPU
Pune, India

Abstract: As we know that after the 12th board results, the main problem of a student is to find an appropriate college for their further education. It is a tough decision to make for many students as to which college they should apply to. We have built a system that compares the student's data with the past admission data and suggests colleges in a sequence of their preference. We have used Decision Tree, Support Vector Classifier, Extra Tree Classifier, Naïve Bayes, KNN and Random Forest as our statistical model to predict the probability of getting admission to a college. It was observed that the performance of Random Forest was achieved highest among all.

Keywords: Decision Tree, Random Forest, KNN, Random Forest, Extra Tree Classifier, SVC, Probabilities

1. INTRODUCTION

Education plays a vital role in today's era. While we talk about career – a person's degree, course, university and the knowledge that he possesses – is the key factor on which the firm hires a fresher. As soon as a student completes his/her Higher Secondary Schooling, the first goal of any student is to get into an appropriate College so that he can get a better education and guidance for his future. For that, students seek help from many sources like online sites or career experts to get the best options for their future. A good career counselor charges a huge amount for providing such solutions. Online sources are also not as reliable as the data from a particular source is not always accurate. Students also perform their analysis before applying to any institution, but this method is slow and certainly not consistent for getting actual results and possibly includes human error. Since the number of applications in different universities for each year is way too high, there is a need to build up a system that is more accurate or precise to provide proper suggestions to students.

Our aim is to use machine learning concepts to predict the probability of a student to get admission into those preferred colleges and suggest a list of colleges in a sequence of the probability of getting admission to that specific college. The following are the steps that include the work we have done in sequence of implementation.

2. RELATED WORKS

One of the researchers has done work on predicting the university and students applying to explicit universities (Jay Bibodi).The first one is the University selection model, and the second one is a student selection model. They came across some issues like noisy data, unformatted text but after cleaning the data, they proceeded to 'model selection' with some important features. "University Selection Model" – A Classification problem with apriori probability output. They found out just two universities giving a higher probability of output. "Student Selection Model" – Classification using supervised learning like Linear and kernel, Decision Tree and Random Forest. Random Forest provided better accuracy than other algorithms i.e. 90% accuracy.[1]

There is one more researcher – Himanshu Sonawane – who has researched on 'Student Admission Predictor'. It is a system built to help students who are studying abroad. This system helps students find the best foreign universities/colleges based on their performance in GRE, IELTS, 12^{th}, Graduation Marks, Student Statement of purpose, Letter of Recommendation, etc. Based on this information, it recommends the best-suited university/college. They have used three algorithms: KNN (76% Accuracy), Decision Tree (80% Accuracy), Logistic Regression (68% Accuracy). In the case of a decision tree, accuracy was nearly the same for both pieces of training as well as testing datasets.[2]

From another research paper, we got to know what affects the likelihood of enrolling (Ahmad Slim – Predicting Student Enrolment Based on Student and College Characteristics). They have used machine learning to analyze the enrolment.

This work intends to provide decision-makers in the enrolment management administration, a better understanding of the factors that are highly correlated to the enrolment process. They have used real data of the applicants who were admitted to the University of New Mexico (UNM).In their dataset, they have different features like gender, GPA, parent's income, student's income. They had data issues like missing value and categorical variables. They have divided classification into two parts – classification at the individual level and classification at a cohort level. For classification at the individual level, the model was used to check the probability of enrolment and whether the applicant is enrolled or not. Logistic Regression (LR) provided an accuracy of 89% and Support Vector Machine (SVM) provided an accuracy of 91% which was used in the classification at an individual level. The total enrolment in 2016 was actually 3402 but the prediction was 3478 by using past year records (2015) using time series for classification at the cohort level. [3]

These researchers – Heena, Mayur, and Prashant from Mumbai – have used data mining and ML techniques to analyze the current scenario of admission by predicting the enrolment behavior of students. They have used the Apriori technique to analyze the behavior of students who are seeking admission to a particular college. They have also used the Naïve Bayes algorithm which will help students to choose the course and help them in the admission procedure. In their project, they were conducting a test for students who were seeking admissions and then based on their performance, they were suggesting students a course branch using Naïve Bayes Algorithm.[4]

One more researcher has made a project for helping students in suggesting them best-suited colleges in the USA based on his/her profile. He has collected the data from online sources which was reported by students. He has used 5-6 algorithms for his project. Naïve Bayes was one of them which gave the highest accuracy among all of them. He has predicted students' chances(probabilities) of getting admission in 5 different universities in the USA.[5]

Other researchers were predicting the student admission (Students' Admission Prediction using GRBST with Distributed Data Mining - Dinesh Kumar B Vaghela). They have used the Global Rule Binary Search Tree (GRBST). While searching, they identified some problems like maintaining a single database for all the colleges were difficult. This paper has two phases i.e. training phase and testing phase. In the training phase, the J48 algorithm was used for all local sites. In the testing phase, Users can interact with the system with the help of the application layer. They have used consolidation techniques in two ways i.e. using If…Then… rules format and Decision Table. They have also used binary search tree construction. After applying this technique, they have found the time complexity of generating the Binary Search Tree from the Decision table is very less and also this BST has efficient time complexity to predict the result. They conclude that data mining techniques can be useful in deriving patterns to improve the education system. [6]

GRADE system developed to help graduate admission committee at the University of Texas at Austin Department of Computer Science (UTCS) by Austin Waters and Risto Miikkulainen Department of Computer Science 1 University Station C0500, University of Texas, Austin, TX 78712. This system first reads the applicant's files from the database and encodes as a high-dimensional feature vector and then a logistic regression classifier is trained on that data. It then predicts the probability of binary classification. The feature vector encoding of a student's file indicates whether the applicant was rejected or admitted. The system was used to predict the probability of admissions committee accepting that applicant or not but, in our model, we are trying to make it easy for the applicants to understand whether they should apply to that college or not.[7]

3. DATA EXTRACTION AND TRANSFORMATION

We have achieved our goals step-by-step to make the data steady, fitting it into our models and finding out suitable algorithms of machine learning for our System.

This step contains mainly – Data Extraction, Data Cleaning, Pre-processing, removing unwanted columns, feature selection, label encoding. These steps are shown in Figure 1.

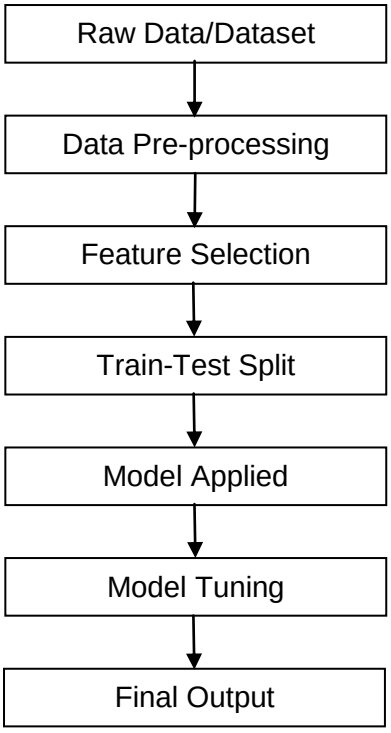


Figure 1: Architecture

3.1 Dataset

Knowing about this use case we need past admission data of multiple colleges to work on. We have extracted data from three different colleges which includes information about a student's academic scores and the reservation category he falls in. Data has been mined from college registries. We have extracted 2054 records that include 13 attributes.

Attributes of the dataset are:

First Name, Last Name, Email ID, Gender, Address, Date of Birth, Category, S.S.C. Percentage, H.S.C Percentage, Diploma Percentage, Branch, Education Gap, and Nationality.

1. Data Preprocessing

Data preprocessing is an important task in machine learning. It converts raw data into clean data. Following are techniques, we have applied on data: -

- **Missing Values –** Missing Value are those values that failed to load information or the data itself was corrupted. There are different techniques to handle missing values. One of which we have applied is deleting rows because some of the rows were blank and they may mislead the classification.
- **Label Encoder –** This is one of the most frequently used techniques for the categorical variable. Label encoder converts labels into a numeric format so that the machine can recognize it. In our data, there are many attributes which are categorical variable like gender, category, branch.
- **Change in data type –** Some attributes didn't include proper input. For example, the Nationality attribute included values like Indian, India, IND which all meant the same country. For that purpose, we needed to change such values into a single format. 'Object' data type values in some attributes had to be changed into 'float' data type. Some records included CGPA for S.S.C scores so we converted those records into a percentage. We made all these changes so that it doesn't affect our accuracy.
- **Drop Columns –** As per domain knowledge, we removed some columns which were not needed in our model.

2. Feature Selection

As we proceed further, before fitting our model we must make sure that all the features that we have selected contribute to the model properly and weights assigned to it are good enough so that our model gives satisfactory accuracy. For that, we have used 4 feature selection techniques: Lasso, Ridge, F1 Score, Extra Tree Classifier.

Lasso, Ridge and F1 Score were removing the features that I needed the most and Extra Tree Classifier was giving me an acceptable importance score. Which is shown below.

Extra Tree Classifier:

Extra Tree Classifier is used to fit a randomized decision tree and uses averaging to improve the predictive accuracy and control over-fitting. We have used this to know the important features of our data.

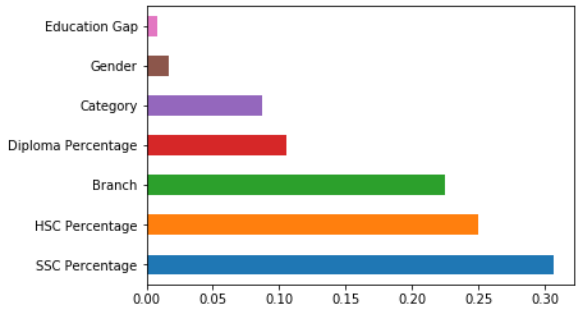


Figure 2: Feature Selection using Extra Tree Classifier

As we can see, my Feature Selection model is giving more importance to S.S.C. The percentage is not appropriate.

So, in this case, our domain knowledge is also helpful to make decisions for this type of situation.

4. EXPLORATORY DATA ANALYSIS

As we saw in feature selection, some features which seemed not so important were contributing to our model. So, to understand those features, we need to do exploratory analysis on this data.

We did exploratory analysis on a few features by grouping and plotting it on graphs.

EDA on Gender Column:

By grouping the gender and plotting the admissions in different colleges as per their gender, we identified some relations between the student's admission and his or her gender. As shown in Figure 4 – For different gender, most students lied in different bins for different colleges. Even for different colleges, we are getting different bell curves. Looking at this we can confirm that the gender column is contributing to our model.

For Extra Tree Classifier, Gender contributes to model – 1.3092%.

EDA on Category Column:

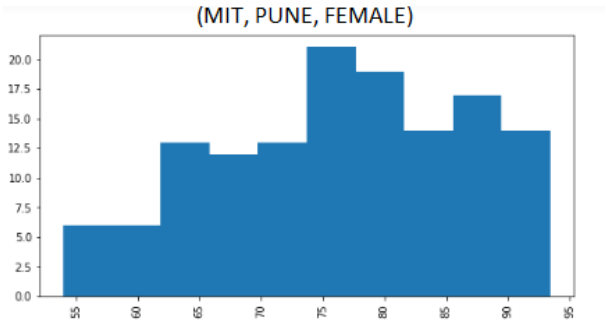
By grouping the category and calculating the percentage of students who got admissions with respect to their categories is shown in Figure 3 – For different categories, we calculated the percentage of students that lie in each category. This percentage of students was matching to reservation criteria as per Indian laws. This shows that the Category column is contributing to our model.

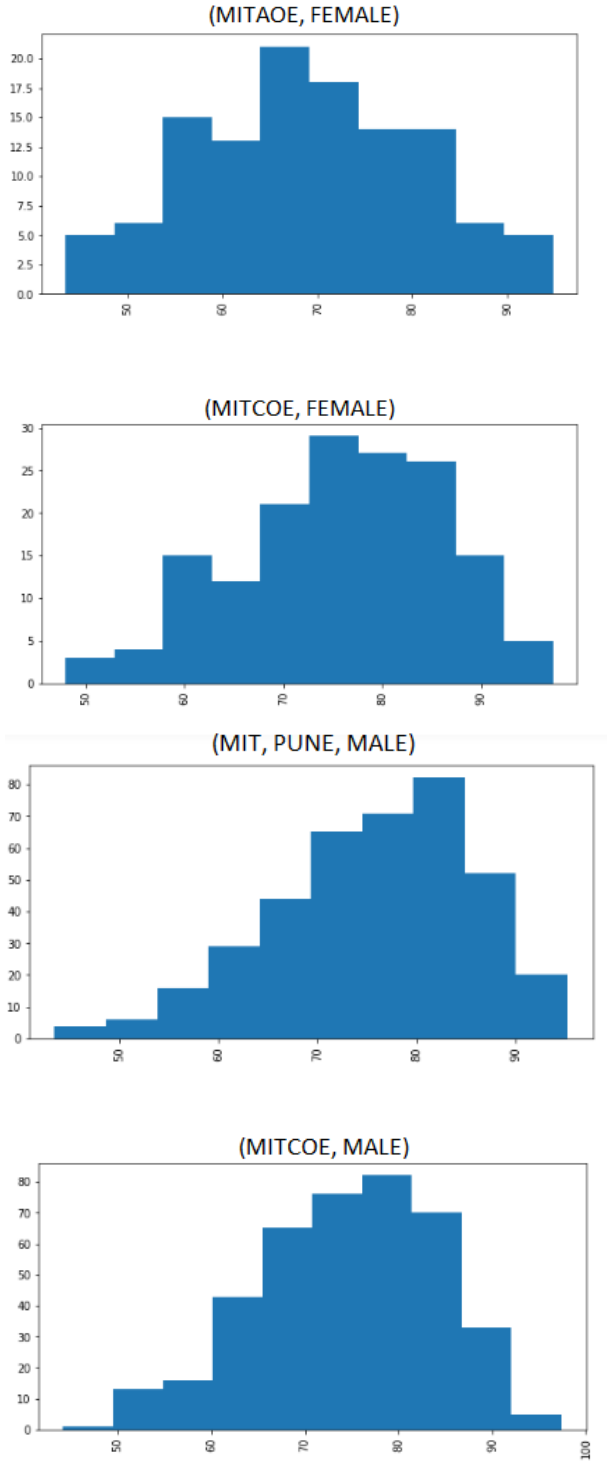
For Extra Tree Classifier, Category contributes to model – 9.6582%.

```
In [5]: df.groupby("Category")["Sr. no."].count()/df.shape[0]*100

Out[5]: Category
        NT(B)      1.850950
        NT(C)      3.214808
        NT(D)      1.899659
        OBC       19.629810
        OPEN      55.577204
        SBC        1.802241
        SC         8.426693
        ST         2.143205
        VJ(A)      1.802241
        select     3.604481
        Name: Sr. no., dtype: float64
```

Figure 3: Admission with respect to Category





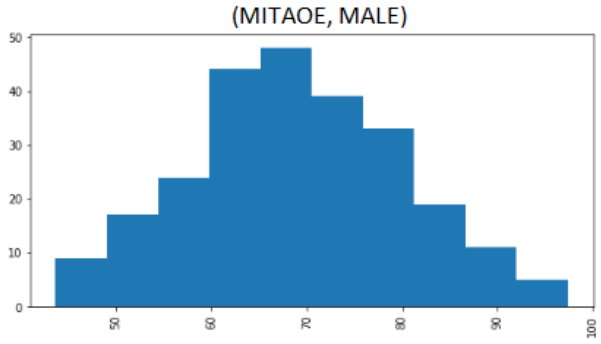


Figure 4: Admission with respect to Gender

5. RESULT AND DISCUSSION

After removing all the noise from the data and after selecting appropriate features for our model, the next step is to find out the best model which gives us more accuracy for train and test both. But before that, we must split our data into 2 parts as we don't have any testing dataset right now.
So, we have divided this modeling section into 3 parts:

1. Train-Test Split
2. Modeling
3. Tuning Model

5.1 Train-Test Split

The training data set is used to create the model while testing the data set is used to qualify the performance. Training data's output is available to model while test data is unseen data. So, in our data, we have split data into 70% for training data and 30% for testing data because it makes the classification model better. while the test data makes the error estimate more accurate.

5.2 Modeling

Following are models, we have applied to check which model gives better accuracy:

- **Support Vector Classifier (SVC):**
  This algorithm is used for the classification problem. The main objective of SVC is to fit the data you provide, returning a "best fit" hyperplane that divides or categorizes your data. From there, when obtaining the hyperplane, you'll then feed some options to your category to examine what the "predicted" class is.

- **Decision Tree:**
  A decision tree is non-parametric supervised learning. It is used for both classification and regression problems. It is a flowchart-like structure in which each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The path between root and leaf represents classification rules. It creates a comprehensive analysis along with each branch and identifies decision nodes that need further analysis.

- **Random Forest:**
  Random Forest is a meta estimator that uses the number of decision trees to fit the various subsamples

drawn from the original dataset. we can also draw the data with replacement as per the requirements.

- **K-Nearest Neighbors (KNN):**
  K-Nearest Neighbors (KNN) is a supervised learning algorithm that is used to solve regression as well as classification problems. Where 'K' is the number of nearest neighbors around the query. It is simple to implement, easy to understand and it is a lazy algorithm. The lazy algorithm means it does not need any training data points for a model generation [8]. All the training data is used in the testing phase. This makes the training faster and the testing phase slower and costlier. By costly testing phase we mean it requires more time and more memory.

- **Naïve Bayes:**
  A Naive Bayes Classifier is a supervised machine-learning algorithm that uses Bayes' Theorem, in which the features are statistically independent. It specifies multiple uses of probability theories and statistics. By simple machine learning problem, where we need to teach our model from a given set of attributes (in training examples) and then form a hypothesis or a relation to a response variable. Then we tend to use this to predict a response, given attributes of a replacement instance.

- **Extra Tree Classifier:**
  The main objective of Extra Tree Classifier is randomizing tree building further in the context of numerical input features, where the choice of the optimal cut-point is responsible for a large proportion of the variance of the induced tree.

We used all these models to fit our data and checked the accuracy which is shown in Figure 5.

Model	Accuracy
Support Vector Classifier (SVC)	47.79%
Decision Tree	56.13%
Random Forest Tree	58.87%
K-Nearest Neighbors (KNN)	52.35%
Naïve Bayes	42.29%
Extra Tree Classifier	58.33%

Figure 5: Accuracy for Different Models

5.3 Model Tuning:

As we can see our accuracy is not going beyond 60%. for that reason, we have tuned our model. Tuning is the method for increasing a model's performance without overfitting the data or making the variance too high. Hyperparameters disagree from other model parameters therein they're not learned by the model automatically through training ways. Following are the models, we applied to check which model gives better accuracy:

- **XGBoost:**
  XGBoost stands for eXtreme Gradient Boosting. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance [9]. Using this we have achieved 64% accuracy.
- **AdaBoost:**
  AdaBoost is one of the first boosting algorithms to be adapted in solving practices. AdaBoost helps you combine multiple "weak classifiers" into one "strong classifier". AdaBoost is best used to boost the performance of all trees on binary classification issues. Using this we have achieved 61% accuracy.

We have used XGBoost and AdaBoost for just improving our accuracy. Accuracy of XGBoost is higher and it improves our accuracy by 6%.

But as our problem statement suggests, we do not need accuracy as we are just calculating the probabilities for getting the admission in all the colleges and referring top probabilities to that student.

6. CONCLUSIONS

The objective of this project is achieved in this process flow which will be used by students to identify the appropriate colleges based on his/her performance. The main aspects of students which are taken under are their 10^{th}, 12^{th} percentages and diploma percentage too if applicable. Besides of that gender, category, Education gap and branch in which student wants to get admission are also contributing to admission. The final model for our project is Random Forest as it is giving a satisfactory output.

As we looked at our data, and we observed that this is just the data of students who took admission. There is no data of neither rejected students nor we have students' choice of college. We can ask students for their choice for college to get better data for accuracy. We were also about to consider the address of the student, as we know that different seats are reserved for those students who belong to different states. But in our data, most of the address columns are not filled properly so we removed that column. So, for that, we can keep dropdown buttons on Online Application Form for cities, states, and countries so that we get proper data for this. We also can ask for entrance exam results which can help us predict more accurately.

7. REFERENCES

[1] Bibodi, J., Vadodaria, A., Rawat, A. and Patel, J. (n.d.). "Admission Prediction System Using Machine Learning". California State University, Sacramento.

[2] Himanshu Sonawane, Mr. Pierpaolo Dondio. "Student Admission Predictor". School of Computing, National College of Ireland. unpublished.

[3] A. Slim, D. Hush, T. Ojah, T. Babbitt. [EDM-2018] "Predicting Student Enrollment Based on Student and College Characteristics". University of New Mexico, Albuquerque, USA.

[4] Heena Sabnani, Mayur More, Prashant Kudale." Prediction of Student Enrolment Using Data Mining Techniques". Dept. of Computer Engineering, Terna Engineering College, Maharashtra, India.

[5] Bhavya Ghai. “Analysis & Prediction of American Graduate Admissions Process”. Department of Computer Science, Stony Brook University, Stony Brook, New York.

[6] Dineshkumar B Vaghela, Priyanka Sharma. “Students' Admission Prediction using GRBST with Distributed Data Mining”. Gujarat Technological University, Chandkheda.

[7] Austin Waters, Risto Miikkulainen. “GRADE: Machine Learning Support for Graduate Admissions”. University of Texas, Austin, Texas.

[8] https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

[9] https://www.meetup.com/Big-Data-Analytics-and-Machine-Learning/events/257926117/