Graduate Admission Recommendation System: Project Report for CSCI-B565 Data Mining

Md. Lisul Islam
Indiana University
islammdl@indiana.edu

Abu Saleh Md. Noman
Indiana University
amdnoman@indiana.edu

Syed Mahbub Hafiz
Indiana University
shafiz@indiana.edu

1 Objective and Significance

Each and every year the graduate schools around the world are experiencing applications from thousands of students all across the globe intending to pursue graduate studies. According to a study by The Council of Graduate Schools (CGS), between fall 2011 and fall 2012 more than 461,000 students [1] enrolled for the first time in graduate certificate, education specialist, Masters, or doctoral programs (only in USA). This huge number of applicant requires to garner sufficient knowledge about the graduate schools before applying since the application process involves significant amount of time and money. The first step before applying to a graduate school is choosing the right school that fits the applicant's credentials from vast pool of choice. This poses a significant challenge because there exist no hard and fast algorithm that guides the student or provide the ordering of choices. In our project we tried to address this issue and provide and come with a solution in the form of a recommender system.

While searching for project idea we went through variety of options like: *Kaggle, KdNuggets, KDD* data and other online repositories. These sites involve lots of project ideas and existing dataset given a goal. But we wanted to explore something 'which is not already there'. Recommendation system based tools are becoming common nowadays with the enhancement of advanced machine learning and data discovery algorithms. However, we thought to approach a problem that is relatively new and where there is scope for doing something novel. So we came up with the idea of building an online graduate student's admission recommender system that will enable students to easily predict the range of universities they should be applying for providing their credentials. So the goal of this project is to understand how and what set of students apply for what sets of schools and based on the profile of a student what might be the admission decision given the school they are applying.

We believe this is important because there does not exist adequate guidance and helping for students when it comes to the point of applying for higher studies. The main motivation comes from the students who are applying from abroad namely international students. Most of the students who complete their undergrad plan to pursue their higher studies in reputed universities primarily located in North America. With the vast pool of information those are available online, it becomes difficult to data mine the required information and plan to apply accordingly. For the international students it is often perplexing to decide which set of schools should he/she will go for. For example, a lot of students from south-east Asia suffer from this dilemma. The data from different blogs/forums is often misleading. Moreover, since every application cost them money and time, it is often crucial to find out the best combination of schools which meet their needs. It is often the case that an international student is not familiar with the admission procedure hence apply to schools which are ambitious or below par according to their profile. Building a recommender system for them might not guarantee 100% of accuracy but certainly help them narrow down the search space in this regard.

2 Background and Related Works

The important concept for the graduate admission recommender system is to understand the process of admission. In most cases, student are asked to apply via a portal and provide their necessary details. This includes standardized test scores such as GRE, TOEFL, IELTS etc. and Undergrad CGPA, Undergraduate Institute, Letters of Recommendation (usually 3) and Statement of Purpose. It is also beneficial for a student if he/she has research publication in his/her area of research. Some schools require further writings such as: statement of diversity, research proposal etc. Based on these credentials the graduate admission committee of a given school decides whether or not to give the student an admission offer. Sometimes depending on the strength of the application materials, the students are provided with full/partial tuition fee waiver too. So it is very important from students point of view to realize what criteria do he/she needs to possess in order to get through. But it is observed that there is no sure-shot algorithm of forecasting this scenario. This is due to the amount of uncertainty and number of variables involved in a decision making. It is not very hard to argue that every year admission committee faces huge challenge to filter out competent candidates. It becomes a optimization problem when a student complements one weakness in his/her application by something else. For instance, if a student have low CGPA yet acceptable (due to unforeseen circumstances) in his undergrad studies but he complements that with excellent GRE score and very good SOP, then he might be in the same wavelength with another student who has moderate CGPA with moderate GRE scores. This can become even trickier if we incorporate all the other variables in question. That is why this problem is very interesting and potentially non-trivial!

Challenges: There are lots of challenges involved in our study. Few of those were:

- The relative importance of the features in decision making is not well defined. We tried to deduce a metric for the strength of an applicant given their profile but it is seemingly difficult to find one. There are myriads of questions involved: is GRE Quantitative more important than Verbal? How many GRE points worth one GPA point? The fact that one criteria can be compensated by the other makes it difficult to generalize.
- The dataset that we had is not representative of the whole mass since not every student is willing to reveal their information. So there might be some bias in the data.
- The data was extremely noisy. There was lot inconsistency such as: misspelling of name of schools, using parentheses, ambiguity, abbreviated version of names, capitalization issue plenty of missing values etc. This requires a lot of cleaning and eventually limits the performance of our analysis.

Although data mining in Academia and student performance analysis is commonplace there has been very few previous work in aforementioned area. Few websites like *Gradcafe*, *msinUS*, *Edulix* are helping students with previous and current records for admission. Also they are providing forum based help services for students. For example, Gradcafe allows students to see the credentials of students who have been admitted in the past for a vast range of schools. Edulix comes up with several other features i.e, Unisearch, UniSuggest, AdmitTrend etc. [7] analyzed $35,000$ entries of Grad school admission data using advanced data visualization techniques and deduced this is nothing but a number game! Another attempt from [8] shows an acceptance estimator for Computer Science graduate admission that provides an User Interface to estimate the range of schools. [9] adopts another approach: based on different credential it calculates a score that maps to a possible range of schools. [10] gives somewhat similar evaluation of profile with the following equation:

$$\begin{aligned} Total\ Score =& (CGPA * 10) + (GRE\ score/20) + (TOEFL\ score/40) \\ & + Number\ of\ publications + Number\ of\ years \\ & in\ related\ job\ experience \end{aligned} \tag{1}$$

Based on total score it shows list of ambitious, matched, and sure shot schools.

The authors of [17] adopted a distributed data mining approach and devised an algorithm called Global Rule Binary Search Tree (GRBST) that stores all the global information from local sites in a binary search tree. This BST is later used to predict the probability of a students' admission in college. Decision tree classifier and fuzzy c-means clustering was proposed in course and college prediction system in [16]. The system classifies the student and matches them to the corresponding study tracks according to their profile using FCM algorithm and C4.5 classification algorithm. Makkinje [15] analyzed the physics graduate admission data available online and tried to gather insight from existing trend based on the rank of different programs. In [11] the authors tried to address this issue with modeling a recommender system based on *KNN, Random Forest and SVM*. They trained the system with *Edulix* data for 45 schools and their test set analysis returned 10 top recommendation given a profile. In [12] the author trained their model and created a calculator for students to compare where they are likely to get accepted.

Another study attempted to address the issue from the opposite perspective (from the graduate schools point of view) [14]. They proposed a semi-automatic process called Least Weighted Distance (LWD) that ranks the applicants according to the strength of their profile. The authors of [13] proposed a novel system called HRSPCA which comprises of two cascaded hybrid recommenders working together on knowledge discovery rules. Although these systems show good results, they suffer from context dependency. Also few of them do not take into consideration all the parameters which we will try to overcome.

3 Methods

In this section we describe how the data is obtained, the methodology that we have adopted, and our evaluation strategy. We have gone through and studied several approaches from the current literature on this problem of graduate admission recommender systems. In the literature [5], the authors presented a hybrid recommender systems which is consisted of two recommender systems and predicts most probable college for a student based on his/her academic merits, background, student records, and the college admission criteria. While this study heavily focuses on college admission for students, it [6] focuses largely on admissions in computer science graduate program.

3.1 Data Collection

We have readily found several dataset that has extensively been collected and used in works such as [2], [3]. In the project [3], the authors have collected data from the website at [4]. Their data set presented in the work [2] contains 271,807 rows in total. Initially, we thought of collecting data from our classmates from B565 Data Mining, Spring 2016. We also had a plan of gathering data from our undergraduate institution (Bangladesh University of Engineering and Technology) for which we had prepared a questionnaire for the collection of graduate admission data. However, that plan did not work out so we had to stick to this dataset [2]. This data was collected by scraping the GradCafe website and the authors made the data and source code open. The data had some cleaning done already but we made some cleaning and processing for our purpose which will be discussed later.

In this data set, we had the following attributes defined in table 1 with corresponding relevant attribute type.

3.2 Data Preprocessing

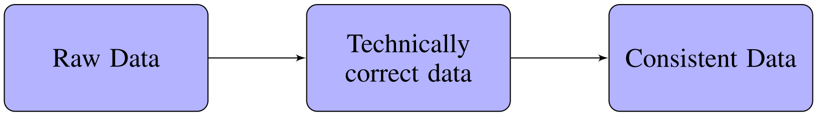
Data preprocessing plays an important role before modeling the data and mining relevant information out of it. The significance of data preprocessing is manifold:

- Real world data is dirty that means it is incomplete (lacking attribute values), noisy (contain outliers) and inconsistent. Before learning from this dataset they need to be accurately preprocessed.
- Only quality data can guarantee quality result. Also, having inconsistency in the dataset can result the model to crash or make it unstable.

  There are several data cleaning techniques that prevail. The basic steps include:

attribute name	attribute type	description
GPA	numerical	Candidate's self-reported undergraduate GPA [0-4.0]
greVerbal	integer	Candidate's self-reported GRE Verbal score [130-170]
greQuant	integer	Candidate's self-reported GRE Quantitative score[130-170]
greWriting	float	Candidate's self-reported GRE Writing score[0-6.0]
greSubject	integer	Candidate's self-reported GRE Subject Test score[0-900]
undergradInstitution	categorical	The undergraduate institution of the applicant.
season	categorical	Season could be either 'Spring' or 'Fall' followed by the year
major	categorical	The intended major.
degree	categorical	The degree to be earned. This field could be 'PhD' or 'MS'
IntendedUniversity	Nominal	Name of the university where the student is applying
decision	categorical	The decision being reported, either 'Accepted' or 'Rejected'

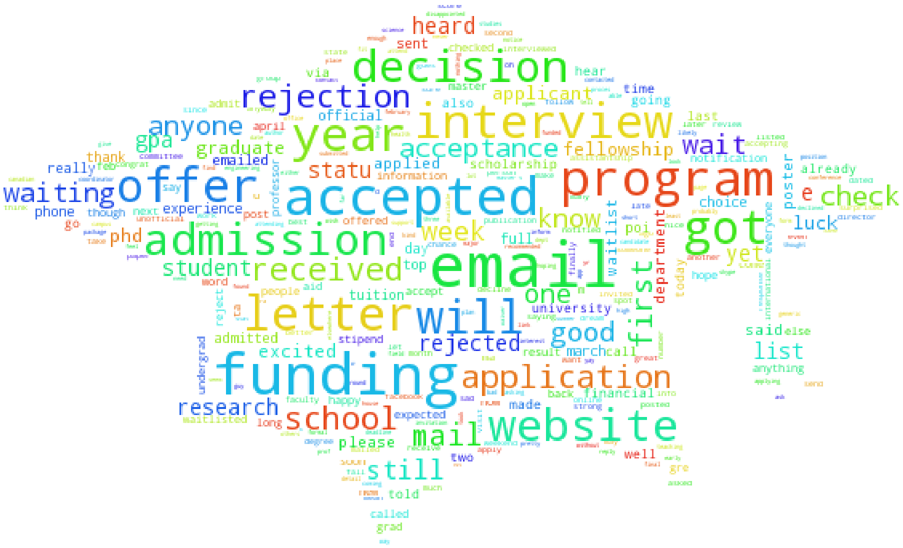
Table 1: Attributes of the data set



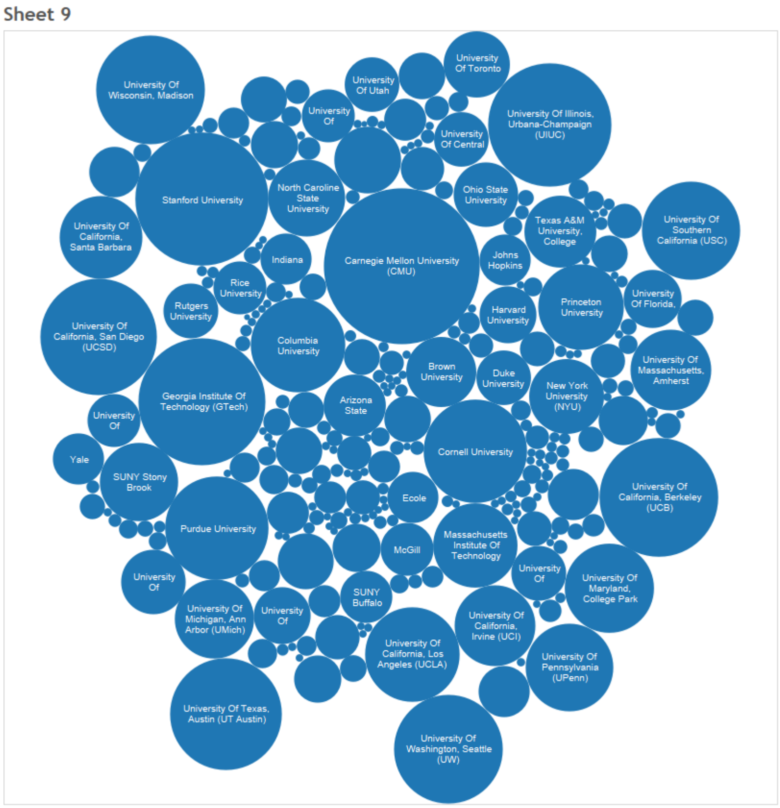
In general, we need to make data cleaning, data integration and transformation, data reduction etc. Each of these steps include a handful of strategies and techniques. We would like to focus only on those which we have used in our project.

Dataset: A Bird's Eye View : Before beginning the cleaning of data, we tried to gain some insight about the data. It is better to have an initial look and identify any pattern in the data:

- The comment section of the data generated the following word cloud [7]:

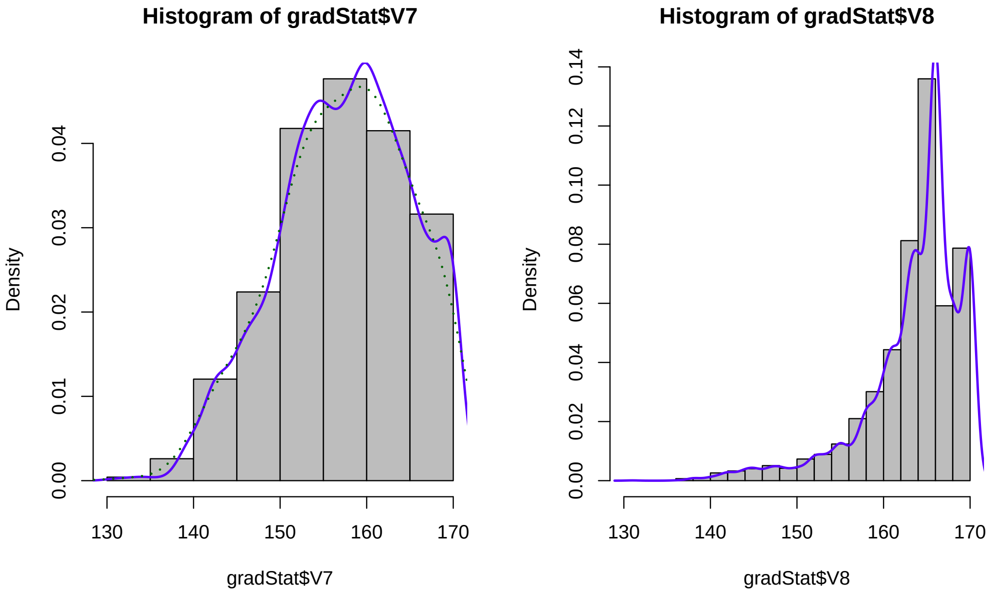


It is intuitive that it comprised of terms like accepted, email, funding, interview etc. Another analysis showed the packed bubbles of schools according to the CGPA of the students who got accepted:



3.png School. Size shows sum of F6. The marks are labeled by school.

This clearly shows the higher CGPA requirement for top-tier schools. The following illustration



Technical Correctness:

A typical technically correct dataset:

- can be directly recognized as belonging to a certain variable.

- is stored in a data type that represents the value domain of the real-world variable. In other words, for each unit, a text variable should be stored as text, a numeric variable as a number, and so on, and all this in a format that is consistent across the data set.

The training dataset was read into a dataframe in R using:

```
gradStat <- read.csv("cs_clean2.csv", header = TRUE)
```

The data fields are well defined, as described in previous subsection. No type conversion, character manipulation or character encoding issues were faced. So it was a technically sound data in the first place.

Data Consistency: Consistent data are technically correct data that are fit for statistical analysis. They are data in which missing values, special values, (obvious) errors and outliers are either removed, corrected or imputed. The data are consistent with constraints based on real-world knowledge about the subject that the data describe.

- We did encounter some missing values in the dataset. This is because few students might accidentally or intentionally skipped some fields. For features those are pivotal in the decision making process, such as greQuant, greVerbal, CGPA we simply omitted those data points. However, we allowed missing values for greWriting feature. Also we only considered the data points with target values 'accepted' or 'rejected' and ignored others such as 'interviewed', 'waiting' etc..

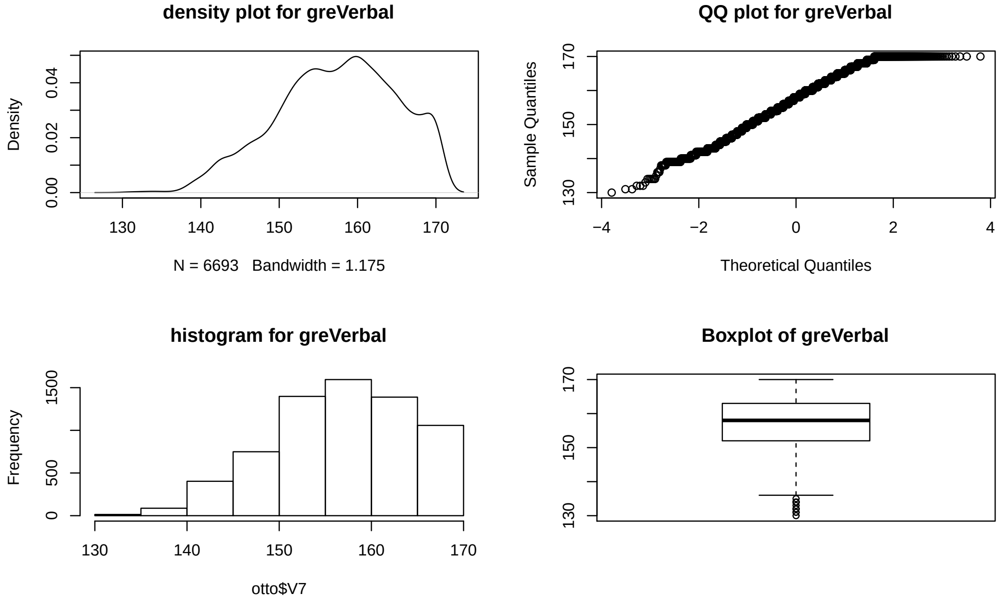
- We tried to detect outliers in the data per feature. Outliers is a set of extreme datapoints which lies far distance apart from usual distribution of data sample from a population. Upon close examination, we found several outliers in the data. Before removing them, first we need to define what exactly is an outlier with respect to a dataset. In statistics, we define:

  ```
  Q3+1.5IQR and Q1−1.5IQR as extreme points,
  Where Q3=3rd quantile
  Q1=First quantile
  IQR=Interquartile Range
  ```

  Any point beyond these ranges are called outliers. To detect outliers, we have drawn the boxplot of each feature and figured out the outliers. To detect and remove the outliers, we used the library caret in R. The following code snippet explains how it was done:

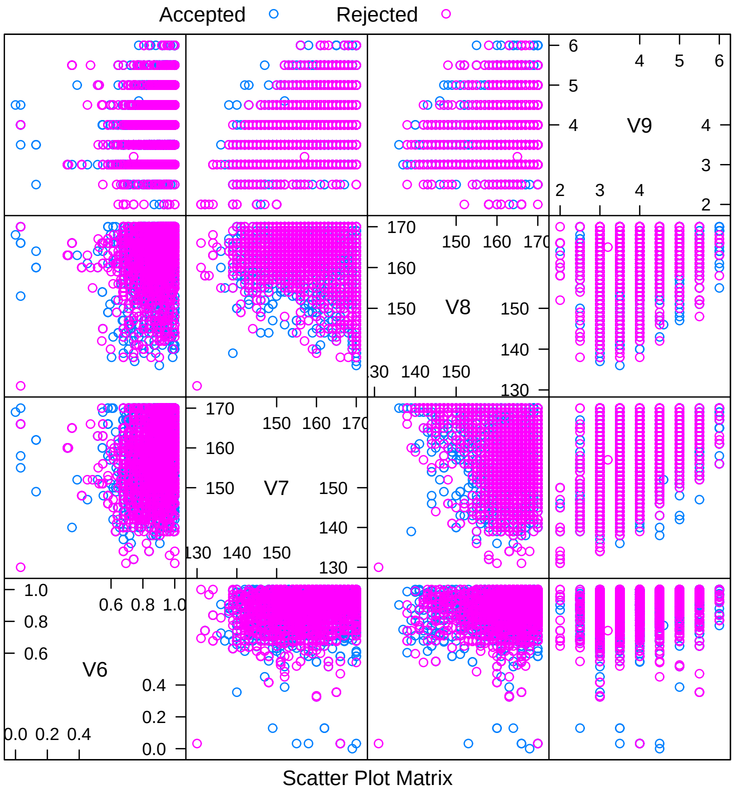
  ```
  for (i in 1:ncol(gradStat)-1){
  outlier_tf1 = outlier(gradStat[,i],logical=TRUE)
  find_outlier1 = which(outlier_tf1==TRUE,arr.ind=TRUE)
  gradStat_new = gradStat[-find_outlier1,]}
  ```

  A detail look at some features helped us visualize the data:



The QQ-plot of greVerbal feature of dataset suggest that it is approximately normally distributed apart from the outlier which make the straight line to deviate. The skewness in the density plot is introduced by these outliers. The boxplot and histogram also refers to the same thing. The circles in the boxplot refer to the outliers which are beyond the whiskers. Careful analysis for each feature was performed similarly and noisy outliers were removed from the data.

- Identifying co-related features is another important aspect of data cleaning. It is better to get rid of the features which are strongly correlated.



A closer look at the pairwise scatter plot suggest that none of the features are strongly correlated.

- Since too many features can contaminate the forecast, we decide to omit the features with near-zero variance. It is observed that, these predictors may become zero-variance predictors when the data are split into cross-validation/bootstrap sub-samples or that a few samples may have an undue influence on the model. These ”near-zero-variance” predictors may need to be identified and eliminated prior to modeling. Zero-variance columns have identical values and therefore play no role in decision making. To omit them, we use the nearZeroVar function and identify such features:

```
library("caret")
nzv <- nearZeroVar(gradStat)
clean_data <- gradStat[, -nzv]
```

- For the purpose of modeling we performed few other modifications: few features such as major, degree, season were mapped to numeric values. Also the value of CGPA was 0-1 normalized and converted to 4 scale (in case it was in 10 scale). The old values of greQuant, greVerbal were converted to new scoring system according to the mapping provided by ETS.
- Treatment of university name attribute draws special attention. As people are asked to write the name on their own, hence same university gets dubbed by different name of strings. It is a great challenge to handle this issue. To tackle this first we use a heuristic that name of university can’t be greater than 80 characters. Using this filter we found less noise free university names. Next we search for distinct names among the filtered data objects. We obtained almost 430 unique names where same university has multiple forms. In next step we group all of these forms under one standard name. These process finds 300 standard university names.
- To make the university name attribute categorical, we decide to categorize them according to the world recognized *USnews* ranking. We choose first 150 universities from the ranking list and assign rank for each of the data object’s that attribute. As we have 300 unique university names, so there are some universities which are out of important ranking or their ranking is unavailable. For example, universities out of USA are fell in this group. We rank them all as 150+. Furthermore, we bucketize this 151 ranks into 16 categories. Where first category contains 1-10 ranked universities and 16 contains ranking-unavailable universities. Thus we convert university name of each data object into specific rank tier. This helps to employ Decision tree, Apriori algorithm, and Neural Network mining technologies.

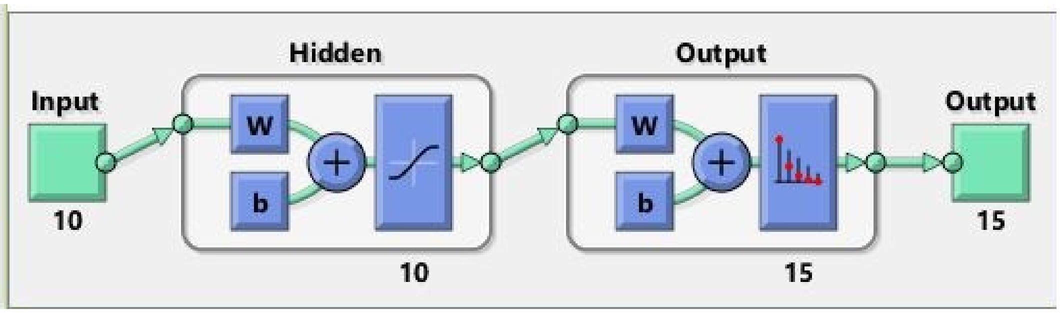


Figure 1: Multi-class (15 university tiers) classification Neural Net

All these preprocessing yields 6,626 data points whereas it was 27,823 initially.

3.3 Mining algorithms

If properly mined, we can extract useful insights from the graduate admission dataset which can be of paramount significance for the graduate study aspirants. We have implemented several popular data mining techniques to perceive different underlying concealed patterns present in the data set.

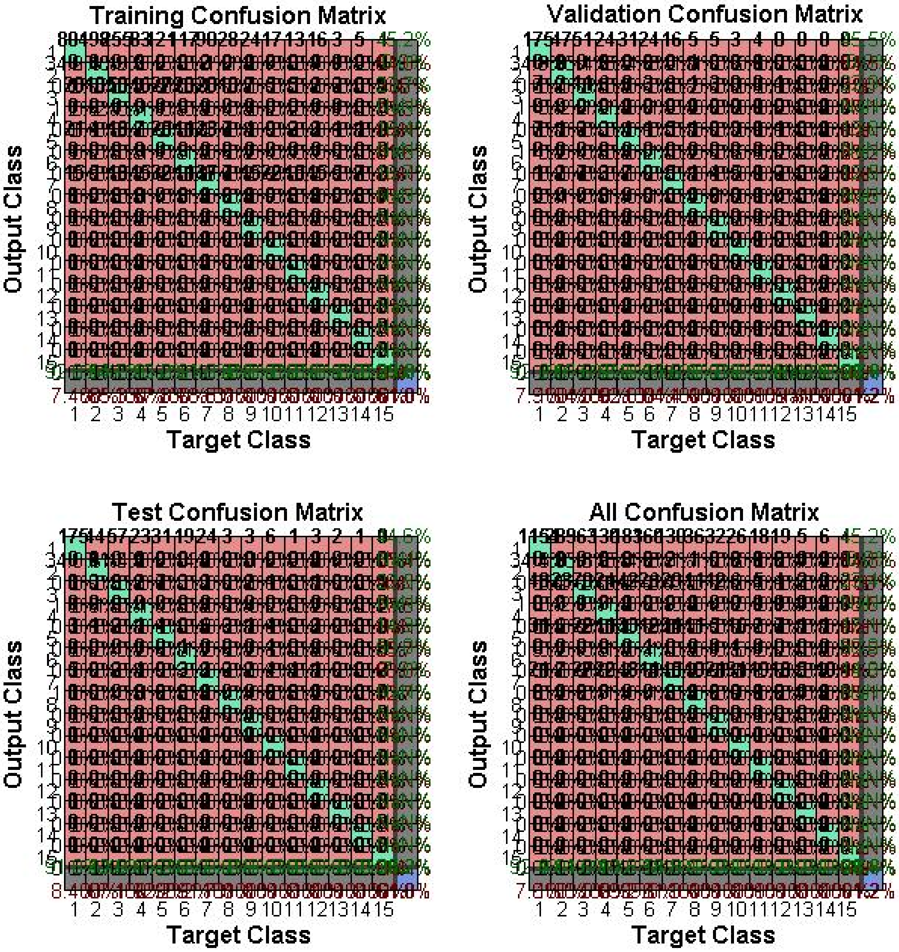


Figure 2: Confusion matrix for Multi-class (15 university tiers) classification Neural Net

	Name	Type
1	UniversityRank	Categorical
2	Program	Categorical
3	Degree	Categorical
4	Session	Categorical
5	Origin	Categorical
6	CGPA	Numerical
7	Verbal	Numerical
8	Quant	Numerical
9	AWA	Numerical
10	SubjectGRE	Numerical
11	**Decision**	Categorical

Table 2: Type of the attributes in Graduate Admission Recommender Data Set

3.3.1 ID3 Decision Tree Learning Algorithm

We have used ID3 decision tree learning algorithm to learn and discover the patterns and credentials needed for a candidate to get into the graduate schools at different US/Canadian/Chinese universi-

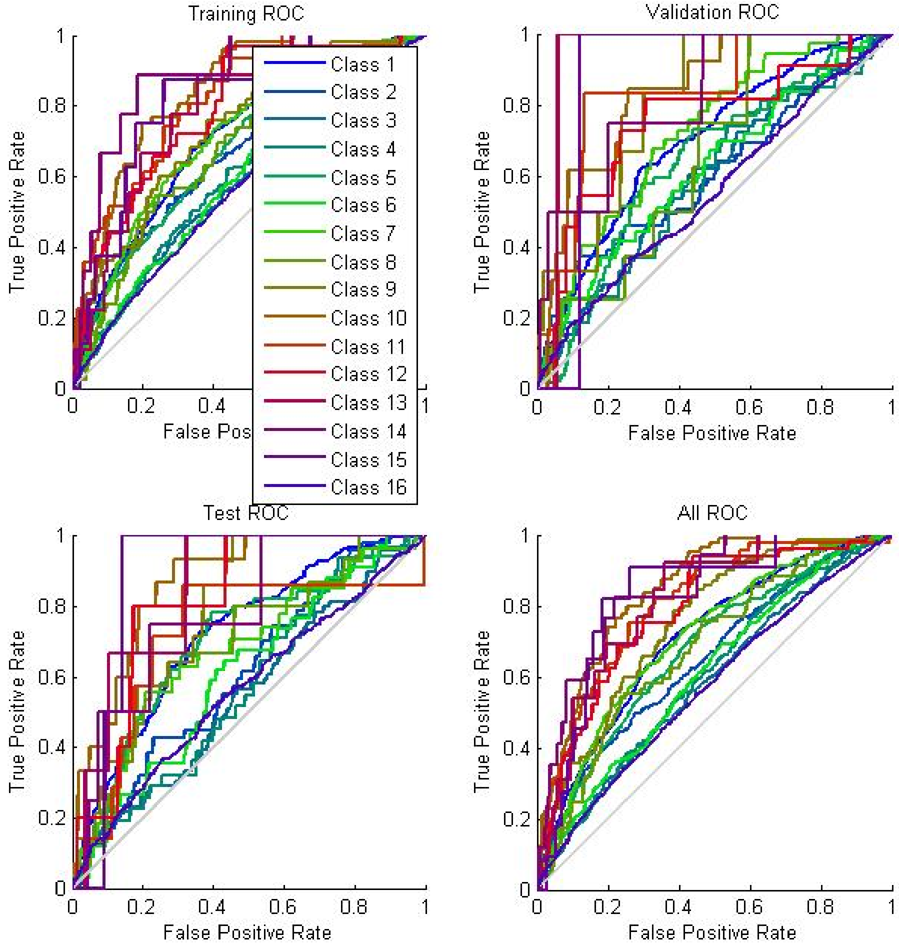


Figure 3: ROC curve for Multi-class (15 university tiers) classification Neural Net

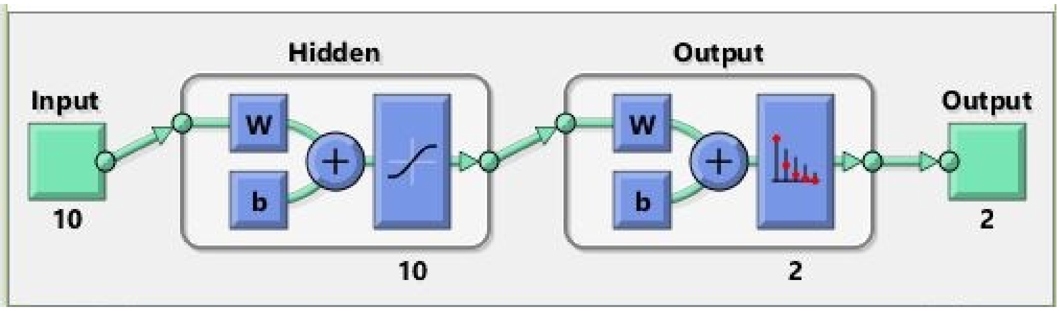


Figure 4: Binary class (Accepted or Rejected) classification Neural Net

ties. As we know, data set contains 10 attributes and one target variable. Out of the 10 attributes, 4 of them are numerical (continuous) and other 6 are categorical. As this is a classification task, we also have target variable which listed in boldface along with other attributes in Table 2. A brief description of the data set, way of data collection and data pre-processing is provided in the previous section.

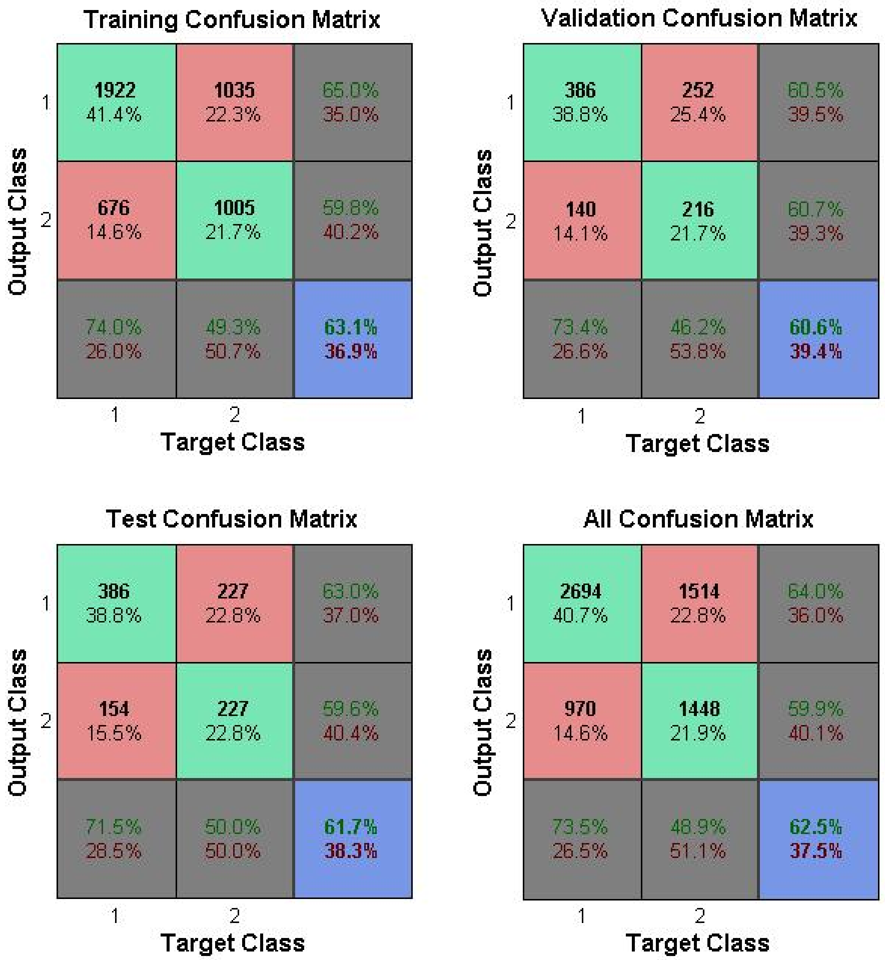


Figure 5: Confusion matrix for Binary class (Accepted Class 1 or Rejected Class) classification Neural Net

Splitting Criterion We have tried out both *Entropy* and *Gini Index* as splitting criterion for the nodes in the decision tree. In any node, the data set is split upon different values of categorical attributes and threshold values of continuous attributes and evaluate each split. Finally, we keep the best split as the testing condition corresponding to that node.

Pruning We also adopted some measures to prune the tree so that it does not grow indefinitely and results in overfitting. As the Occams Razor states simpler model or smaller decision tree is more likely to give us low generalization errors. We have set a threshold, t to a particular value and we do not grow the tree further if the nodes contains data of size less than the threshold. In that case, we assign the label of majority class as the node label and make the corresponding node as leaf of the tree.

Cross Validation and Evaluation Strategy We have used 10-fold cross validation to train and test the data set. We calculated the accuracy, recall, precision. f-measure of this experiment. We report these values in coming Section 4. Each of the experiment, done with decision tree learning, have used 10-fold cross validation strategy to train and test different evaluation criterion.

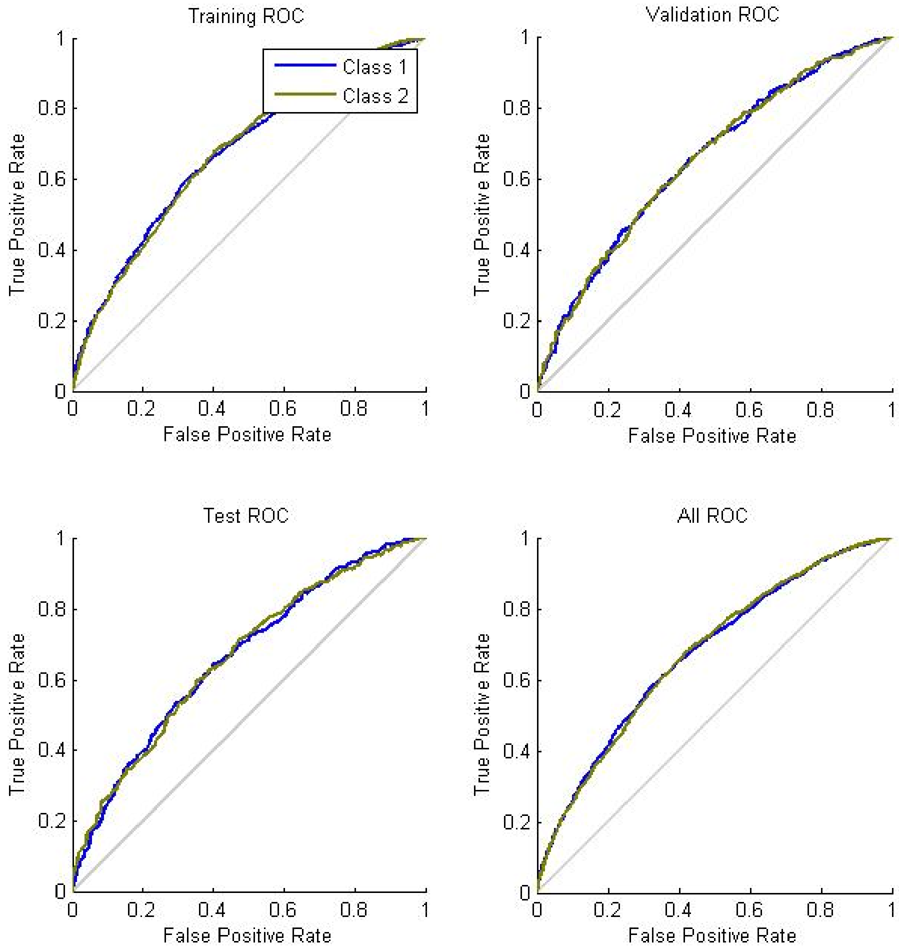


Figure 6: ROC curve for Binary class (Accepted Class 1 or Rejected Class 2) classification Neural Net

3.3.2 Association Rule Mining Using Apriori Algorithm

To extract hidden pattern, knowledge we employed association rule mining technique. We have used Apriori Algorithm to discover interesting rules. There are 16 university rank buckets, 5 programs, 5 degrees, 15 sessions. Moreover, cgpa, GRE scores are first normalized to [0,1] range and then discretize into 10 groups; hence these 4 attributes each has 10 categories. As most of the data object s don't have subject GRE scores, so this value can mislead the pattern mining, that's why we don't consider this attribute in final rule mining. Last two attributes origin and decision has 4 and 2 categories. In summary, we transfer each attribute into several categories. In specific, there are 81 distinct items in the input of apriori algorithm.

We have employed both $F_{k-1} \times F_1$ and $F_{k-1} \times F_{k-1}$ method here. We have used the threshold of minimum support of 20%, 30%, and 40%. Though we have tried with other support percentages, but in that case we didn't find any association rule, so we stopped testing further values. Beside, for confidence we investigated 30%, 40%, and 50%. We also tested *lift* interestingness to identify more surprising rules from the processed dataset. The obtained rules and results are demonstrated and discussed in Sections 4.

3.3.3 Neural Networks

We have used the Neural Network support from Matlab. To feed this tool, we have to perform further processing upon the already available dataset form. As we have several categorical attributes and several numerical attributes, so we convert the categorical attributes to ordered ones. To acquire this, we used several heuristic to make a logical and acceptable orderings. More specifically, the university rank categorical attribute is already ordered. However, programs, degrees, sessions origins, decisions these categorical attributes are heuristically ordered. For example, we weight CS program more than that of ECE program. Similarly, most recent session, PhD degree, Accepted decision, American origin are weighted higher than other. After these conversion, we perform Z-score normalization over these and then train a neural net and classify according to our requirements. In particular we have two kinds of target variables in this regard. Both are explained in the following paragraphs.

Probable application decision when university rank tier is chosen by the student: In this case the interested student gives his credential values for each attribute and interested university rank tier. Then our trained neural net will give him either Accepted or Rejected decision after trained over already presented dataset. So we have two classes, this binary classification.

Probable universities where the student should apply: In this circumstance, student just gives his necessary credentials for the attributes and our trained second neural net will propose him/her a university tier out of 16 university rank tiers where each tier contains 10 universities except the last one. If the student applies any number of the universities of the resultant tier then his admission acceptance probability is comparatively higher according to our present dataset.

One thing to mention here, as many data objects has 16th university tier; hence, we observed that it dominates other classes and decrease the expected behavior of the neural net. So in our final experiment we discard those data objects which has 16th university rank class, so for this special case the number of data objects of the active dataset reduced to 3500 only. Therefore, this is a multi-class classification problem, more specifically we have 15 classes of different university tiers.

4 Results

4.1 Evaluation of Decision tree methodology

To evaluate the performance of our ID3 decision tree learning algorithm, we constructed the confusion matrix given in table 3. As stated in the previous section, we have tested both entropy and gini index as splitting criterion and the accuracy, precision and recall are reported for each splitting criterion in table 4. We also plugged in different values of pruning threshold, t and took the one that give maximizes the accuracy. If t is set to 100, that implies we will not grow the tree further from any node that contains data set of size less than 100. Accuracy, Precision and Recall for different values of t are reported in table 5

	Predicted Class = Yes	Predicted Class = No
Actual Class = Yes	TP(True Positive)	FN(False Negative)
Actual Class = No	FP(False Positive)	TN(True Negative)

Table 3: Confusion Matrix

Splitting Criterion	Accuracy	Precision	Recall
Entropy	60.42	63.01	68.23
Gini Index	61.48	64.59	68.30

Table 4: Accuracy, Precision and Recall with Entropy and Gini Index as Splitting Criterion

From the table, we can see gini index as splitting criterion works slightly better than entropy. Table 5 shows us that accuracy is insensitive to the value t but higher value of t tends to produce better

t	Accuracy	Precision	Recall
20	60.27	63.47	66.94
50	60.12	63.78	66.21
100	60.12	63.16	66.29
150	60.88	64.16	69.19
200	59.97	62.09	71.78
500	60.88	62.35	73.002
1000	60.42	60.24	80.83
3000	58.16	58.63	84.19

Table 5: Accuracy, Precision and Recall for different values of t

Figure 7: Whole Decision Tree Produced by ID3

recall. Recall represents percentage of positive instances that are classified correctly as positive. As, we have more positive instances than negative in the data set, so higher value of t will ensure higher recall and results in under fitting, hence reduce the accuracy. So, we picked 100 as the final value of t. Visualization of the decision tree produced for this data set is given in Figure 7 and 8.

4.2 Interesting rules extracted by Apriori

After discussion of 3.3.2, here we are explaining our obtained results from the implementation. We have tested with different levels of support, confidence and lift. The result is shown in Table 6. In this table, we have 9 cases where lots of interesting rules are observed. Most of them have significant interpretation. All of the investigated rules are outputted in file. We are mentioning some of them here:

[90%<quant<100%, 3.6<cgpa<4.0]=>[Program CS]; that means in the given dataset who is applied for Computer Science program, has 90% above GRE quant score and CGPA of close to 4.0.

[3.6<cgpa<4.0, Decision Accepted]=>[Program CS]; when someone applied for Computer Science program having CGPA of 3.6 to 4.0 range, then his application decision is accepted most of the cases.

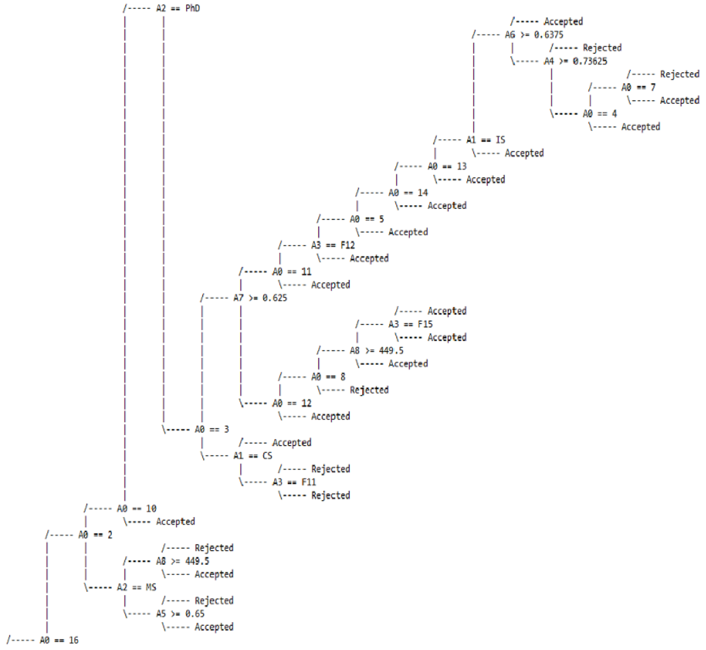
[Degree PhD, 90%<quant<100%]=>[Program CS]; Most of the student who applied for CS PhD has GRE quant score of close to 100%.

Similarly some other rules from hundreds:

[3.2<cgpa<3.6]=>[Program CS]

[AWA 3.6]=>[Program CS]

[Degree MS, Decision Accepted]=>[Program CS]



```
                        /----- A2 == PhD
                        |       |                                                               /----- Accepted
                        |       |                                                       /----- A6 >= 0.6375
                        |       |                                                       |       |       /----- Rejected
                        |       |                                                       |       \----- A4 >= 0.73625
                        |       |                                                       |               |               /----- Rejected
                        |       |                                                       |               |       /----- A0 == 7
                        |       |                                                       |               |       |       \----- Accepted
                        |       |                                                       |               \----- A0 == 4
                        |       |                                                       |                       \----- Accepted
                        |       |                                               /----- A1 == IS
                        |       |                                               |       \----- Accepted
                        |       |                                       /----- A0 == 13
                        |       |                                       |       \----- Accepted
                        |       |                               /----- A0 == 14
                        |       |                               |       \----- Accepted
                        |       |                       /----- A0 == 5
                        |       |                       |       \----- Accepted
                        |       |               /----- A3 == F12
                        |       |               |       \----- Accepted
                        |       |       /----- A0 == 11
                        |       |       |       \----- Accepted
                        |       |       /----- A7 >= 0.625
                        |       |       |       |                       /----- Accepted
                        |       |       |       |               /----- A3 == F15
                        |       |       |       |               |       \----- Accepted
                        |       |       |       |       /----- A8 >= 449.5
                        |       |       |       |       |       \----- Accepted
                        |       |       |       /----- A0 == 8
                        |       |       |       |       \----- Rejected
                        |       |       |       \----- A0 == 12
                        |       |       |               \----- Accepted
                        |       \----- A0 == 3
                        |               |       /----- Accepted
                        |               \----- A1 == CS
                        |                       |       /----- Rejected
                        |                       \----- A3 == F11
                        |                               \----- Rejected
                /----- A0 == 10
                |       \----- Accepted
        /----- A0 == 2
        |       |               /----- Rejected
        |       |       /----- A8 >= 449.5
        |       |       |       \----- Accepted
        |       \----- A2 == MS
        |               |       /----- Rejected
        |               \----- A5 >= 0.65
        |                       \----- Accepted
/----- A0 == 16
```

Figure 8: Magnified view of part of the decision tree produced by ID3

[Origin International]=>[Program CS]

[3.6<cgpa<4.0]=>[Program CS]

[Degree PhD]=>[Program CS]

[University Ranking Tier 16]=>[Program CS]

Table 6: Experimental results of Apriori association rule mining

Min Sup	# of Cand IS by F_1 meth	# of Freq IS by F_1 meth	# of Cand IS by F_{k-1} meth	# of Freq IS by F_{k-1} meth	# of Closed Frequent IS	# of Max Frequent IS	Min Confidence	# of Conf pruned Assoc Rules	# of Lift (1.0) pruned Assoc Rules
20%	679	74	282	74	74	36	30%	186	100
20%	679	74	282	74	74	36	40%	157	100
20%	679	74	282	74	74	36	50%	112	100
30%	106	27	88	27	27	18	30%	36	18
30%	106	27	88	27	27	18	40%	36	18
30%	106	27	88	27	27	18	50%	31	18
40%	67	13	48	13	13	8	30%	8	0
40%	67	13	48	13	13	8	40%	8	0
40%	67	13	48	13	13	8	50%	8	0

4.3 Neural network results

For the first case where we have two classes for application acceptance, the trained neural network is like Figure4. From the figure it is clear that we used 10 hidden layers, our input dataset has

10 attributes and we have 2 class output. Here we achieved around 62% accuracy, which confusion matrix is shown in Figure 5. The corresponding ROC curve is demonstrated in Figure 6. So from the acquired accuracy we can claim that this neural net can be used for future student profile to tell him/her whether a particular university tier is suitable or not.

For the Second case where we have 15 classes (university tiers) for forecasting most suitable university tier, the trained neural network is like Figure1. From the figure it is straightforward that we used 10 hidden layers as well, our input dataset has 10 attributes and we have 15 classes as output. Here we achieved around 40% accuracy, which confusion matrix is shown in Figure 2, for enormous class overlapping the image is not that much clear though. The corresponding ROC curve is demonstrated in Figure 3.

In addition, here in second case we observed drop in accuracy to only 40%, which is because of the shortage of data. Like when we drop the data oject rows which has 16th university tier, the size of dataset becomes 3,500, which is not enough to get an acceptable accuracy. So we think when we will be able to collect more data from volunteers our accuracy for this case will be improved more. This case is necessary because one of our recommendation system goal is to help student where they should apply, where their application acceptance is higher. In this case, when a student gives his/her profile the neural net outputs a university tier of 10 specific universities. So we observed a trade-off here, when we try to mitigate the domination of 16th university tier, to get unbiased classification, the dataset size drops significantly. Consequently the trained neural net on small dataset illustrated lower accuracy.

5 Conclusions

As we all know, graduate studies is the doorway for the inception of research by brightest minds across the globe. The aim of this project was to take a first step towards solving the problem faced by plenty of students planning for graduate studies. The analysis of the result shows that we had notable success towards that goal. The accuracy (more than 60%) we achieved in decision tree based technique, where we can suggest a student whether his/her given credentials will put him/her in 'Accepted' state or not against his admission application to certain university, is significantly high and can be further improved by overcoming the challenges discussed earlier. Moreover, in the same case of Neural Network based classification acquires 62% accuracy, which is also a remarkable achievement of our combined effort to pre-process the data by various justifiable means of cleansing and employing several logical heuristics in different steps of the methodologies. Additionally, by association rule mining by Apriori algorithm observes hundreds of interesting rules and knowledge which are interpret-able according to the real life scenario of graduate admission.

This research is still in development process and can be directed to future research. First, we have not taken into consideration different important parameters i.e. statement of purpose (SOP) and letter of recommendations. It requires a high level natural language processing to determine how much effect these might have. Second, the number of publications as well as quality of publication plays an important role in decision making. That was ignored too. Third, the size of the dataset is too small to call this a very good recommender. We plan to recruit more data in future that can help us build a even better predictor.

6 Individual Tasks

For the purpose of clarity and efficiency we adopted a divide and conquer strategy. We divided the project into tasks and the group members were responsible for equal amount of work as per their strength. Moreover, we had set milestone and arrange periodic group meetings to remain updated. A typical approach included:

- Data collection
- Data understanding and visualization
- Data preprocessing
- Model building

- testing and evaluation
- Further development and deployment (if possible)
- Report writing.

Since we are three of us, the individual task list was (not limited to):

Md Lisul Islam- Data Collection, preprocessing, Coding, Model Building, Testing, Report writing.
Abu Saleh Md Noman- Data collection, Data preparation and visualization, Model building, Report writing.
Syed Mahbub Hafiz- Data collection, preprocessing, Coding, Model Building, Testing, Report writing.

References

[1] cgsnet.org/us-graduate-schools-report-slight-growth-new-students-fall-2012.

[2] https://github.com/deedy/gradcafe_data.

[3] http://cseweb.ucsd.edu/~jmcauley/cse255/reports/fa15/026.pdf.

[4] http://www.edulix.com/.

[5] http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6416521&queryText=Graduate%20admission%20Recommender%20system&newsearch=true.

[6] http://dl.acm.org/citation.cfm?id=1226772.

[7] D. Das. The grad school statistics we never had. Available: http://debarghyadas.com/writes/the-grad-school-statistics-we-never-had/.

[8] Acceptance estimator for computer science graduate admissions. Available: http://www.cs.utep.edu/nigel/estimator/

[9] cherukuri ajay. Profile evaluation for ms in us and phd in usa. Available: http://www.msinus.com/content/profile-evaluation-ms-us-261/#.VtkHYpwrLt5

[10] Profile evaluation. Available: http://www.higherstudyabroad.com/select/profile/profile-evaluation/

[11] R. Swaminathan.University recommender system for graduate studies in usa. Available: http://cseweb.ucsd.edu/ jmcauley/cse255/reports/fa15/026.pdf

[12] N. Ward, The (un) predictability of computer science graduate school admissions, Communications of the ACM, vol. 50, no. 3, pp. 104106, 2007.

[13] A. H. M. Ragab, A. F. S. Mashat, and A. M. Khedra, Hrspca: Hybrid recommender system for predicting college admission, in Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on. IEEE, 2012, pp. 107113.

[14] O. Olivo, A least weighted distance approach to the doctoral student selection problem, 2009.

[15] J. A. Makkinje, An analysis of physics graduate admission data, arXiv preprint arXiv:1504.03952, 2015.

[16] S. V. K. Kumar and S. Padmapriya, An efficient recommender system for predicting study track to students using data mining techniques.

[17] D. B. Vaghela and P. Sharma, Students admission prediction using grbst with distributed data mining.