EECS 349: Machine Learning (Spring 2015)
Final Project Report
Group members: Corinna Wendisch (CWP734), Eureka Foong (ECF384), Mmachi Obiorah (MGO983)

Getting into grad school: Predicting acceptance to graduate engineering programs using machine learning

TASK

Having an advanced degree has become a necessity for securing a job in many fields. However, it is difficult for applicants to predict their likelihood of being accepted to these academic programs. We want to help prospective students decide which programs they should apply to given the likelihood of being accepted to the program. In this project, our task was to predict whether a student who applies to a graduate engineering program will be accepted or rejected based on the student's GRE scores, residential status, Grade Point Average (GPA) and program type (Master's or PhD).

DATA

We obtained data from GradCafe.com, a forum where prospective graduate students publicly share their acceptances and rejections from academic programs. Using a Python script, we scraped data from the engineering forum to obtain numeric attributes, such as GPA, GRE Quantitative score, GRE Verbal score, and GRE Analytical Writing score, and nominal attributes, such as a student's residential status (A: American; U: International, with US degree; I: International, without US degree), graduate program type (PhD, Masters), school name, program name, and result (rejected or accepted). We removed observations that did not have GRE scores and were left with 10571 observations out of an initial 22582 observations.

To clean the data, we first converted GRE scores on the 200-700 scale to scores on the latest 130-170 scale (effective August 1, 2011). We ensured school and program names were consistent as sometimes a school was referred to by different names (e.g. MIT and Massachusetts Institute of Technology). We marked unknown values as such and replaced unknown numeric values with the average of the attribute for the data set. To create training and test data sets, we randomized rows in the spreadsheet by creating a column of random numbers and sorting on this column. We used the top 70% of the observations for training, and the remaining 30% for testing.

METHODS

Our initial intuition was that different schools have varying standards of acceptance. Because there were too many schools to consider in the entire data set, we trained classifiers on two data sets: the first contains data from only 16 engineering schools with the highest number of observations, whereas the second is the entire dataset. We fed these data sets into Weka and tried several different classifiers to predict on the result attribute of instances in the test sets.

Training considering only the schools with most instances

Can we predict the likelihood of being accepted to a specific university given GPA, status, degree type, and GRE scores? We created a separate data set (3393 training instances, 1321 test instances) with the instances from the top 16 graduate schools and defined the nominal attribute **Top_School** = {austintexas, carnegiemellon, columbiauniversity, cornelluniversity, georgiatech, massachusettsinstitutetechnology, northwesternuniversity, princeton, purdueuniversity, stanforduniversity, ucberkeley, ucla, ucsandiego, umichiganannarbor, universitywashington, virginiatech}. We also added a numeric attribute, rank, that represents the relative rank of the school based on the U.S. News & World Report's article in 2016. In total, this dataset included 9 attributes: Top_School, Result, GPA, Writing_GRE, Status, Degree_Type, new_VerbalGRE, newQuantGRE, and Rank.

Training using the entire dataset

Can we predict the likelihood of being accepted to any graduate engineering school given GPA, status, degree type, and GRE scores? In this dataset, we did not include the program or school name for training. Our intention was to predict acceptance into any school at all given the other attributes. We partitioned the data into a training set with 70% of the data (7398 instances) and a test set with 30% of the data (3172 instances). We also tried using 10-fold cross-validation in Weka after combining our original training and test sets.

SOLUTION AND RESULTS

Across both datasets, we discovered that the Rotation Forest algorithm performed the best out of the classifiers. Below, we report on the accuracy of different classifiers at predicting values, either on the test set or using 10-fold cross validation.

Training considering only the schools with most instances

The baseline accuracy for the dataset (ZeroR) was 50.34%. The following decision tree classifiers performed the best on the test set, which only considered graduate schools with the most instances (Table 1).

Classifier	**Number of correctly classified instances**	**Accuracy**
Rotation Forest	941	71.2339 %
Classification via Regression	926	70.0984%
J48	910	68.8872%

Table 1. The accuracy of various classifiers at predicting the values of examples from the top schools dataset.

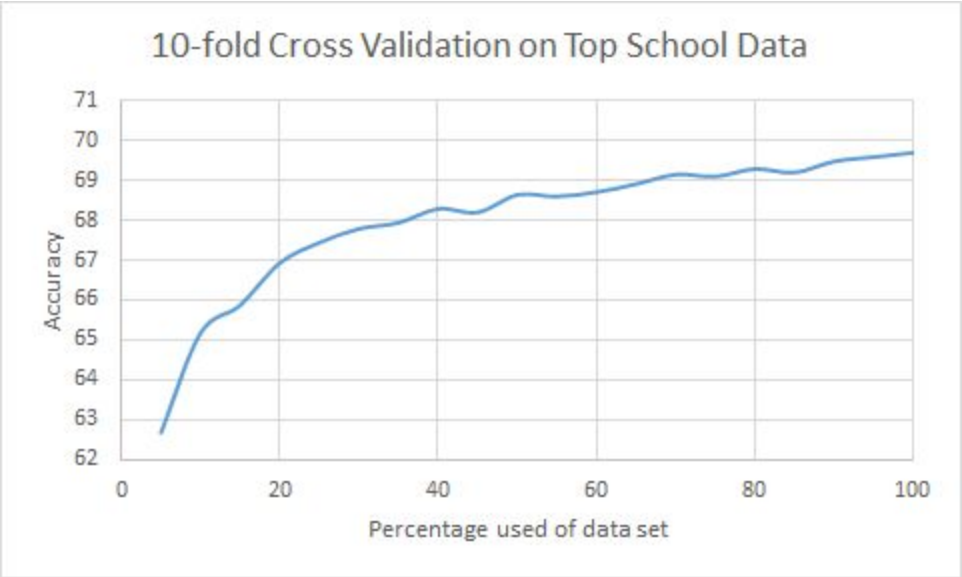

Figure 1. Learning curve for the top school data set, where we used 10-fold cross validation on the combined training and test sets. The underlying model is RotationForest.

Training using the entire dataset

The baseline accuracy for this dataset (ZeroR) was 56.27%. The following algorithms performed the best on the entire dataset, using 10-fold cross validation (Table 2).

Classifier	**Number of correctly classified instances**	**Accuracy**
Rotation Forest	6607	62.5071%
Classification via Regression	6567	62.1287%
J48	6534	61.8165%

Table 2. The accuracy of various classifiers at predicting the values of examples from the entire dataset, using 10-fold cross validation.

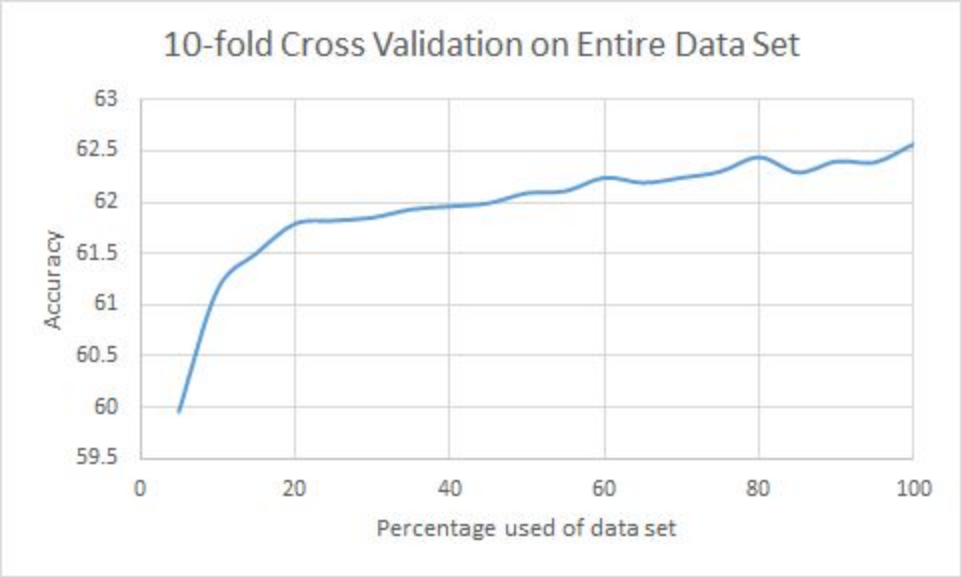

Figure 2. Learning curve for the entire data set, where we used 10-fold cross validation on the combined training and test sets. The underlying model is RotationForest.

When we partitioned the data into a training set with 70% of the data (7398 instances) and a test set with 30% of the data (3172 instances), we obtained a baseline accuracy (ZeroR) of 56.9%. The following algorithms were the most accurate at predicting values on the test set (Table 3).

Classifier	**Number of correctly classified instances**	**Accuracy**
Rotation Forest		
- *With J48, percentage removed: 50% (default)*	2002	63.1148%
- *With J48, percentage removed: 10%*	1992	62.7995%
Logistic Regression	1976	62.2951%
J48	1974	62.2320%

Table 3. The accuracy of various classifiers at predicting the values of examples from the test set.

DISCUSSION

RotationTree is an ensemble method that trains several decision trees independent from one another. For each tree, a different set of attributes is considered in a rotated attribute space. The Rotation Forest classifier may have performed best for our data set because it considers

the possibility that we may be missing key variables in our dataset. These variables, such as the quality of an applicant's essay or their co-curricular activities, might account for diversity within the sample, which may have confused other classifiers. For example, it might be the case that students with both high and low GPAs were accepted to graduate programs, and that another feature, such as the quality of an applicant's essay, provides more knowledge about whether a student is accepted or not. In other words, it seems like our dataset is missing several key features that predict the likelihood of acceptance to graduate school, and the Rotation Forest algorithm simply accounts for this the best.

Besides that, we observed overall low accuracies with all of the classifiers we tried. This might be because our sample is biased and may not be able to predict the likelihood of acceptance for all prospective graduate students. In order to determine how representative the contributors to GradCafe are of all applicants, we compared some of the attributes of our dataset to average values within a population of college students in 2010 and GRE test takers in 2013 and 2014 (Table 4). On average, the students in our sample had higher GPA and GRE scores in all areas. Furthermore, there was a significantly higher percentage of international students in our dataset: 62.72% compared to only 36% of students who completed the GRE between July 2013 and June 2014. Given that our sample is more international and performs better academically than the average college student, this model may not be able to predict the likelihood of acceptance for all graduate school applicants.

Attribute	**Average within dataset**	**Average within overall population**	**Source**
GPA	3.64	3.15	Mean of average GPA at private and public colleges and universities in 2010 http://www.gradeinflation.com/tcr2010grading.pdf
GRE - Analytical Writing	3.8	3.5	Average GRE scores of test takers between July 2013 and June 2014 https://www.ets.org/s/gre/pdf/snapshot_test_taker_data_2014.pdf
GRE - Verbal	156.5	150.2	Average GRE scores of test takers between July 2013 and June 2014 https://www.ets.org/s/gre/pdf/snapshot_test_taker_data_2014.pdf
GRE - Quantitative	163.8	152.5	Average GRE scores of test takers between July 2013

			and June 2014 https://www.ets.org/s/gre/pdf/snapshot_test_taker_data_2014.pdf
Percentage of international students	62.72%	36% (refers to non-U.S. citizens)	A survey of GRE test takers between July 2013 and June 2014 https://www.ets.org/s/gre/pdf/snapshot_test_taker_data_2014.pdf

Table 4. A comparison of the average GPA and GRE scores, and the percentage of international students within the data set and within the population of graduate school applicants.

LIMITATIONS AND FUTURE WORK

One of the limitations of our project is that we have used a relative small dataset that may only be representative of prospective students who use GradCafe. As we discussed above, our sample is, on average, skewed towards international students, and performs better academically than other graduate school applicants. Furthermore, because we chose to disregard the school and program attributes in our entire dataset, some instances contain the same attribute values, but different labels. For example, if one student applied to multiple schools, this would appear in our dataset as several identical instances with different labels. By ignoring the school and program attributes, this may have resulted in false positives and redundancies.

In the future, we would like to extend our work by first obtaining additional attributes that may be more informative than our current attributes. For instance, it might be useful to obtain ratings of prospective student's essay scores. We would also benefit from including more data from domestic students, so that our sample is more representative of the prospective student population. In addition, we would use a classifier with probability outputs, such as super vector machine or logistic regression. Lastly, we would like to obtain more data in general to improve the accuracy of our model.

CONCLUSION

In conclusion, we created a model using the Rotation Forest algorithm that predicts with some accuracy whether a student is accepted or rejected to an engineering school and program, given their GPA, GRE scores, status, and type of degree. Nevertheless, there is considerable room for improvement on this model that may be addressed in future work by collecting additional attributes, such as an applicant's essay and extracurricular experience.

RESPONSIBILITIES OF TEAM MEMBERS

- Mmachi: Scraped data from GradCafe.com, cleaned and experimented with the entire dataset in Weka, uploaded the functioning model to the project website
- Eureka: Cleaned and experimented with the entire dataset in Weka, characterized the bias within our sample of instances, wrote final report
- Corinna: Cleaned the scraped dataset from GradCafe.com, cleaned and experimented with top schools and Mechanical Engineering datasets, generated learning curves for final report