Capstone Project Proposal

Prepared by: Piyush Agarwal
October 11, 2018
Proposal number: 001

Objective

To build a predictive model which can be used to predict the rate of admissions and enrollments based on the number of applications and the test scores accepted by the school. I will be using supervised machine learning algorithms like Random Forest, Neural Nets, etc. to build a robust model.

Domain Background

We all have been in the situation where we are searching for undergrad and grad schools to apply for. These days, there are a lot of schools around and everyone has a different acceptance criteria for admissions. While filling out our application we want to make sure that we choose a school with a higher probability of accepting our application, one because the application costs are high and second because we want to apply to a school which accepts our applications somewhere in the range of our test scores.
Supervised Machine Learning has proven itself repeatedly. It is being applied in many domains these days to find answers to complex questions based on the historical data. Many real world applications, which we use on a day-to-day basis, are using supervised learning in the background - think catching spam emails, predicting stock price movements, financial fraud detection, etc.

Problem Statement

Most of the information these days can easily be found by “Google-ing” but, think about a scenario in which you are applying to a school whose admissions rate is unknown or not available online yet (maybe because it’s a new school)? *Can we build a model which can predict the admissions rate of a new school based on other similar schools whose data is available? Can we extend this to the rate of enrollment?*

Datasets and Inputs

For the purposes of this model development I am going to use the *Integrated Postsecondary Education Data System (IPEDS)* data provided on the *National Center for Education Statistics (NCES)* webpage. This data is

updated annually and I'm planning on using the data for the years 2010 - 2016 (latest available). Each year's data is available in a Comma-Separated Values (CSV) format. There are a lot of columns in the data but I'm going to use only a few important ones required for the development of this model.

After initial data cleaning and munging, I will divide the data into 3 sets:

- The *train* set is going to be the one with most of the data (about 70%) and will be used for model development and training.
- The *test* set is going to have some data (about 20%) to test the trained model and to see how the model performs on data which it has never seen before during training. This testing could possibly be done repeatedly until I get a model which performs satisfactorily.
- Finally the *validation* set, with about 10% of the remaining data. This is going to be a hold-out set and will never be seen by the model either during training or testing phases. Once the final model is achieved, it will be validated on this set.

Solution Statement

The main goal of this project is to understand the relationship between the test scores accepted by a school and the rate of admissions for that school. Based on this relationship, we can determine the admissions rate of a school whose accepted test scores are known to us but not the admissions rate. *Through this model, I am also looking to analyze how good can we forecast the rate of change in the admissions over the years for a given school.*

Benchmark Model

I am planning to train few different Supervised Learning Algorithms to find the best model. A good modeling practice is to first get a baseline model working and then try to improve it from there by using different algorithms and tweaks. As a first step, I am planning to build a simple Linear Regression model and see how it performs on the test dataset. The output performance of this model will serve as a baseline for developing a better model either by using different machine learning algorithms like Random Forest, Neural Networks, etc. or by tweaking the linear model parameters and building on better features.

Evaluation Metrics

Evaluating a model performance is as necessary as building the model itself. For evaluating the performance of this regression model, I am going to use the statistical measure of ***coefficient of determination*** *(denoted as R^2 or r^2)***.** It is defined as the proportion of the variance in the dependent variable at is predictable from the independent

variables(s). Wikipedia defines coefficient of determination mathematically as below. When put in simple words - coefficient of determination shows the effect of variation in input features on the variation of the predicted feature. The best r^2 value a model can have is 1.0 - higher this value, the better our model. A negative r^2 value shows that the model is performing arbitrarily worse and needs some major work.

> A data set has n values marked $y_1,...,y_n$ (collectively known as y_i or as a vector $y = [y_1,...,y_n]^T$), each associated with a predicted (or modeled) value $f_1,...,f_n$ (known as f_i, or sometimes $\hat{y}_i$, as a vector f).
>
> Define the residuals as $e_i = y_i - f_i$ (forming a vector e).
>
> If $\bar{y}$ is the mean of the observed data:
>
> $$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$
>
> then the variability of the data set can be measured using three sums of squares formulas:
>
> - The total sum of squares (proportional to the variance of the data):
>
> $$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2,$$
>
> - The regression sum of squares, also called the explained sum of squares:
>
> $$SS_{\text{reg}} = \sum_i (f_i - \bar{y})^2,$$
>
> - The sum of squares of residuals, also called the residual sum of squares:
>
> $$SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$
>
> The most general definition of the coefficient of determination is
>
> $$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

Project Design

Below are the steps I have in mind to design this model:

1. Data Prep: analyze and clean the data. This step also involves getting the required input variables from the data in a format which can then be used for model development.
2. Data Splitting: split the data into train, test and validation sets. Then remove the target feature (admission rate) from the test and validation set for the model evaluation later.
3. Initial Analysis: use the train set to analyze the relationships between different variables. What variables are more important for predicting the target variable, is there any other important insights we can get from the data, etc.
4. Feature Generation: think and create any new derived variables which might be helpful in better prediction.

5. Benchmarking: train the initial benchmark Linear Regression model and evaluate its performance on the test data set.
6. Algorithm Selection: experiment with various other algorithms. I am planning to use Random Forest and/or Neural Networks (Deep Learning) for this model.
7. Model Evaluation & Tuning: evaluate the final model's performance on the test set. See if I can further improve the performance by some tweaking of the model parameters.
8. Final Model Evaluation: evaluate the model performance on the held-out validation set. Use this for reporting.
9. Reporting: create the final model report for submission.

References

- NCES webpage
- IPEDS data system
- Strengths and Weaknesses of IPEDS data
- Wikipedia for r2
- Scikit-learn metrics.r2_score