Data Insight: Top-100 National Universities

Yanbang Wang

Introduction:

US is one of the countries that have most prestigious universities in the world. And there are numerous ranking websites that are devoted to collecting, presenting, and comparing the information (data) of those top universities. However, the data usually spread around within the website, and are not well formatted from data analysis perspective. Meanwhile, a fascinating fact to people outside the country is that here exist a large number of top private universities, whose operations are either completely or mostly independent of government funding. This is very different from the rest of the world, where government takes the main responsibility to provide stable financial support to high education system.

Goal:

This project aims to provide an insight about the top national universities in the country, by first collecting university metadata from a university ranking website, formatting them into .csv file. Based on the collected data, machine learning algorithms are applied in order to answer an ultimate question: what determines the amount of endowment that a university (either private or public) annually receives.

Data Sources:

1. US News National University Rankings:
   https://www.usnews.com/best-colleges/rankings/national-universities
   Metadata about a university are embedded in the website's HTML pages.
   Important variables about a university are: name, ranking, address, number of undergraduate students, ZIP code, university type (private/public), founded year, setting (city/urban/suburban/rural), yearly endowment, and URLs linking to the original page of those data.

2. The College Scorecard API endpoint: http://api.data.gov/ed/collegescorecard/v1/schools
   Full documentation: https://collegescorecard.ed.gov/data/documentation/
   This is the one of the entrances to the gorgeous education system data set maintained by US Department of Education. It presents all-round data about any registered university on a span of around 20 years. The response of this GET API are in JSON format. This API is used to provide complementary data on top of the data acquired from US News Ranking Website. In particular two variables are queried and returned: completion rate (percentage of students that are able to graduate within 150% of normal program period), and annual cost (including tuition fee and insurance).

Data Manipulation Methods:

1. The first step of the project is to scrape data from US New Best National University Ranking website. Relevant HTML pages include an overview list (containing brief information) located at https://www.usnews.com/best-colleges/rankings/national-universities. From there you can find a dedicated URL for each university. Through this URL, you will be able to bring up a HTML page that incorporate most of the data you need in this project. In terms of techniques, this project uses BeautifulSoup to parse the HTML pages. A class called Unvs is defined in order to store the metadata of a university. In fact, Unvs's __init__() method only takes a BeautifulSoup tag as the parameter and it does all the scraping job internally. The program instantiates a Unvs object for each of the 100 universities. Another technique involved is a simple cache system which is embedded into the whole process to significantly reduce the times of webpage requests and avoid IP blocking. The cache is in .json format and is designed to store all the webpages in this step and JSON response (from API) in the next step. The only malformed data that needs special care is "thumbnails URL" variable, because it was noticed that some universities in the list do not have its own thumbnails. A lot of conversions is done internally in Unvs object, including type conversion and key word extraction using regular expression.

2. After scraping, api_get() method is called on the list of Unvs objects (thus for 100 times in total). This method constructs a GET query based on the university name, address, and ZIP code, make inquiries about completion rate and annual cost, and join the two data sources together into the Unvs object. While constructing the query itself and parsing the response does not involve many techniques, this step is the most time-consuming part in the whole project, because there is a lot of information mismatches between the US News website and the database of API server. Here list two types of common mismatch that consumes a lot of time and effort to debug and fix:
   1) University Name Mismatch: the university's name given by US News website does not completely match with any of the university's name in API server's database, or matches more than one university's name in API server's database.
   2) Universities that are next to each other sometimes has identical 5-digit ZIP code.

   Sometimes only one of them happens in a query, sometimes both. The unpredictable patterns almost always throw an exception in the program and take a lot of time to examine the mismatched output and debug.

   The only way to solve problems of this kind is to manually adjust the query so that it matches The data stored in the API server. Please go to mismatch_handle() function for details.

3. All the data acquired previously are output to a .csv file, which serves three purposes:
   1) It serves as another cache, so that the program does not have to go through all steps of data collection in order to perform model training at the last step. If a .csv file already exists, it simply read it into a list of Unvs objects and returns the specific data that are need to train models.
   2) It can be read by Weka, which is a data mining and knowledge discovery software that manipulates the data and makes it easier to examine the correlations among multiple variables.
   3) It is also a preliminary visualization of the data collected so far.

4. Model training and result prediction (and visulization) is the last step of the project. The training set is a list of vectors in the following format

   <rank, #undergraduates, type, history, setting, completion rate, annual cost, endowment amount>

   Only setting and type are nominal attributes, others are numeric attributes in float numbers.

   Basically, two models are trained: 1. Decision tree, in particular, the CART tree. 2. Adaboost with decision tree being the kernel. 90% of the data are used for training, and the rest are for testing. Next, in order to visualize the outcome, prediction results both in the training and testing part are plotted against ground truth to see the accuracies of Adaboost model. The last step is to visualize the decision tree model, which is kind of straightforward because the tree can be (trained and) displayed directly on Weka.

   We do not go into much detail of this step in the report, because this is not our course's focus. The detailed implementation can be found in the attached code.

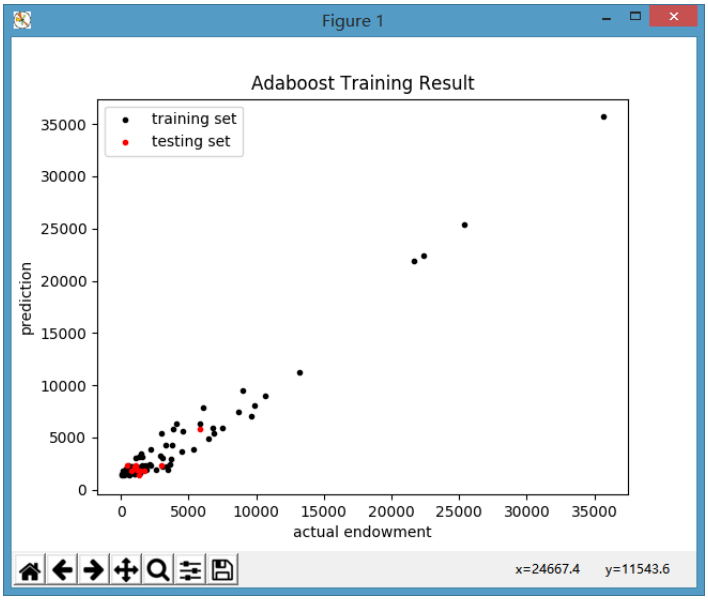
Outcomes and Findings:

1. We generate a .csv file that stores the meta data of top-100 universities in the country:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	name	rank	address	thumbnail	n_ug	page_url	zip	type	year_founded	setting	endowment	completion_rate	annual_cost
2													
3	Princeton University	1	Princeton, NJ	https://ww	5400	https://ww	8544	Private, Coed	1746	Suburban	$21.7 billion +	0.9684	59665
4													
5	Harvard University	2	Cambridge, MA	https://ww	6710	https://ww	2138	Private, Coed	1636	Urban	$35.7 billion	0.9759	62250
6													
7	University of Chicago	3	Chicago, IL	https://ww	5941	https://ww	60637	Private, Coed	1890	Urban	$6.1 billion +	0.9168	67572
8													
9	Yale University	3	New Haven, CT	https://ww	5472	https://ww	6520	Private, Coed	1701	City	$25.4 billion +	0.9678	63970
10													
11	Columbia University	5	New York, NY	https://ww	6113	https://ww	10027	Private, Coed	1754	Urban	$9.0 billion +	0.9518	66604
12													
13	Massachusetts Institute o	5	Cambridge, MA	https://ww	4524	https://ww	2139	Private, Coed	1861	Urban	$13.2 billion	0.9193	61030
14													
15	Stanford University	5	Stanford, CA	https://ww	7034	https://ww	94305	Private, Coed	1885	Suburban	$22.4 billion	0.9385	62363
16													
17	University of Pennsylvani	8	Philadelphia, P	https://ww	10019	https://ww	19104	Private, Coed	1740	Urban	$10.7 billion +	0.9534	64200
18													
19	Duke University	9	Durham, NC	https://ww	6609	https://ww	27708	Private, Coed	1838	Suburban	$6.8 billion +	0.9495	63999
20													
21	California Institute of T	10	Pasadena, CA	https://ww	979	https://ww	91125	Private, Coed	1891	Suburban	$2.2 billion +	0.9084	60990
22													
23	Dartmouth College	11	Hanover, NH	https://ww	4310	https://ww	3755	Private, Coed	1769	Rural	$4.5 billion	0.9414	65133
24													
25	Johns Hopkins University	11	Baltimore, MD	https://ww	6117	https://ww	21218	Private, Coed	1876	Urban	$3.3 billion +	0.9369	63509

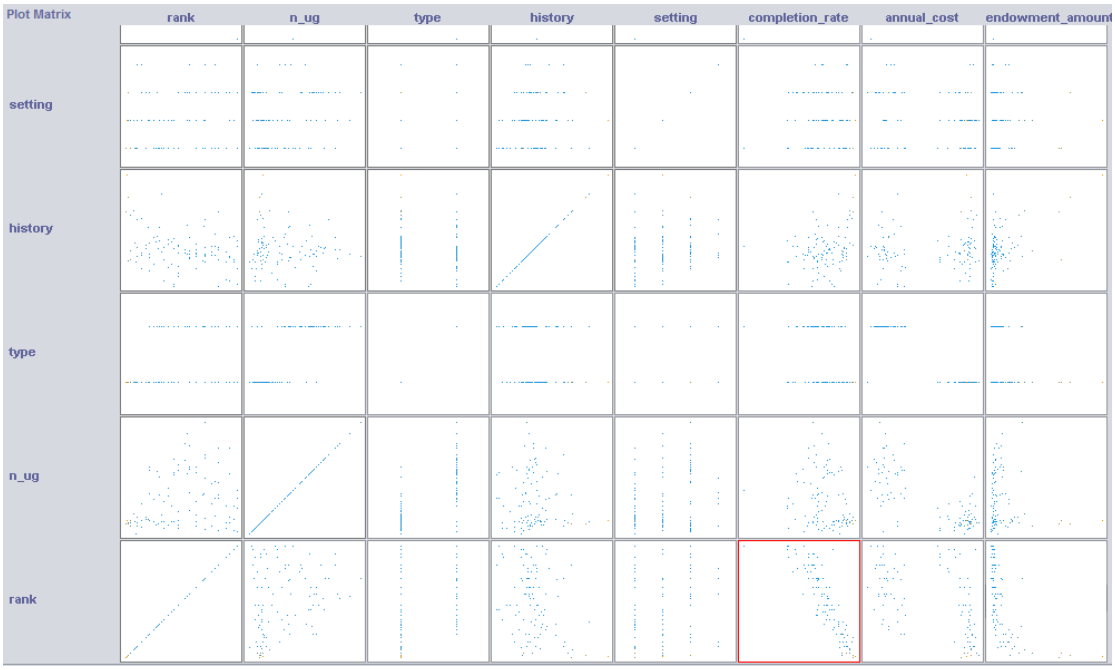
2. We generate an Adaboost model that achieves satisfying prediction result for endowment: Prediction values (see left plot next page) in both training set and testing set lie on a diagonal.
3. We generate a decision tree model (see right plot next page) that predict the endowment (depict the relationship between multiple variables and endowment). It is very intuitive to see that rank has been the determinant node in the first two layers of the tree. Followed by history and n_ug (#undergraduate students). The two variables that are originally acquired via API (completion rate and annual cost) do not really matter much.
4. We generate a plot matrix which displays relationships between any of the two variables. (see bottom plot next page) While most of the plots are not very illuminating, it is worth noticing the plot in red frame, which plots the rank against completion rate. It shows than the rank# and completion rate are of very observable negative relationship. Another two findings are the plot of type-n_ug and plot of type-annual cost. They show that the private universities have significantly smaller undergraduate community and higher academic cost.

```
rank < 8.5
|   rank < 2.5
|   |   history < 326 : 21700 (1/0)
|   |   history >= 326 : 35700 (1/0)
|   rank >= 2.5
|   |   history < 296.5
|   |   |   annual_cost < 63281.5
|   |   |   |   n_ug < 5779 : 13200 (1/0)
|   |   |   |   n_ug >= 5779 : 22400 (1/0)
|   |   |   annual_cost >= 63281.5
|   |   |   |   rank < 4 : 6100 (1/0)
|   |   |   |   rank >= 4 : 9850 (2/722500)
|   |   history >= 296.5 : 25400 (1/0)
rank >= 8.5
|   rank < 28.5
|   |   n_ug < 7496.5
|   |   |   rank < 23
|   |   |   |   rank < 20.5 : 3812.5 (8/2598593.75)
|   |   |   |   rank >= 20.5 : 6900 (1/0)
|   |   |   rank >= 23 : 1200 (2/10000)
|   |   n_ug >= 7496.5
|   |   |   completion_rate < 0.91 : 9600 (1/0)
|   |   |   completion_rate >= 0.91
|   |   |   |   completion_rate < 0.92 : 4200 (3/86666.67)
|   |   |   |   completion_rate >= 0.92 : 6860 (5/1234400)
|   rank >= 28.5
|   |   n_ug < 48283
|   |   |   setting = Suburban
|   |   |   |   rank < 35.5 : 1695.08 (6/524571.81)
|   |   |   |   rank >= 35.5 : 737.93 (20/324832.38)
|   |   |   setting = Urban
|   |   |   |   annual_cost < 41311 : 2828.57 (7/719183.67)
|   |   |   |   annual_cost >= 41311 : 1211.18 (13/682081.34)
|   |   |   setting = City
|   |   |   |   completion_rate < 0.85 : 879.14 (16/290107.78)
|   |   |   |   completion_rate >= 0.85 : 1633.33 (6/508888.89)
|   |   |   setting = Rural
|   |   |   |   annual_cost < 26926.5 : 835.8 (1/0)
|   |   |   |   annual_cost >= 26926.5 : 402.2 (2/1900.96)
|   |   n_ug >= 48283 : 9900 (1/0)
```



(Visualization for finding 2 and 3)

(Visualization for finding 4)



References:

1. US News National University Rankings: https://www.usnews.com/best-colleges/rankings/national-universities
2. College Scorecard API documentation: https://collegescorecard.ed.gov/data/documentation/

Acknowledgement:

The project's coding part has a little overlap with my final project that is going to be submitted (but not yet) for SI507 this semester: Small part of both projects' scraping code actually do the same job by scraping the similar data from US News website. The cache system of both projects are also the same. However, I still make promise about the full originality of my work in this project.