University Ranking Prediction
By Kevin Shen

Goal
Given a dataset of the Times Higher Education World University Rankings and scraping extra Wikipedia data, can a model be trained to predict the ranking of a university?

Data Analysis and Preprocessing
I chose to frame this as a regression problem by trying to predict the score of the university (minimize the total error between my predicted scores and the actual scores).

I chose to use only the Times Higher Education World University Rankings. Upon looking at the data (both visually and through Python), I realized there were two major challenges:

1. The dataset contained rankings spanning different years (2011-2016). This is a problem because one of the requirements required for Wikipedia data to be integrated. I chose to only work with 2016 data as most of the university's Wikipedia pages are up to date until 2018 or 2019. Therefore, 2016 gives me the most recent data in case I wanted to scrape certain values from Wikipedia (such as size of endowment for example). Another problem is that in earlier years they only ranked the top 400 universities, while in 2016 rankings went to the top 800.

1. After the top 200 rankings, the dataset grouped the universities into buckets (201-250 for example), and did not give the exact score. Because of this, I chose to take a safer approach and use just the top 200 universities.

My simplified approach resulted in less data (only top 200 universities in 2016), however, I believe this is fine. My goal in the limited time I have is to create a safe workable baseline. From there on, in there future, I can expand the scope of the problem, have a better idea of other methods, or perhaps integrate more data.

Further exploratory data analysis did not reveal anything serious about the data.

There was not much heavy data pre-processing to be done. My pre-processing tasks included:
- Converting string numerical values to float, and categorical values to categories.
- Replacing commas in numbers (26,788 is turned into 26788).
- Replacing NaN values with the column's mean (I used this approach in previous Kaggle competitions).
- Convert percentages to decimal/float (27% is turned into 0.27).

Model and Training

I initially shuffled and divided the dataset into a random train and test set with 80-20 split. I chose not to create a validation set due to the limited data I have, and the fact that I won't be comparing many different models.

I initially did not include any external Wikipedia data.

The model I chose was LightGBM (LGBM), because in similar situations in the past I have used it. In addition, because I am working with a regression problem on tabular data, LGBM should work well as a baseline. The parameters for the LGBM model I chose were the default ones given in a quickstart guide.

The model initially gave an MSE of 4.11.

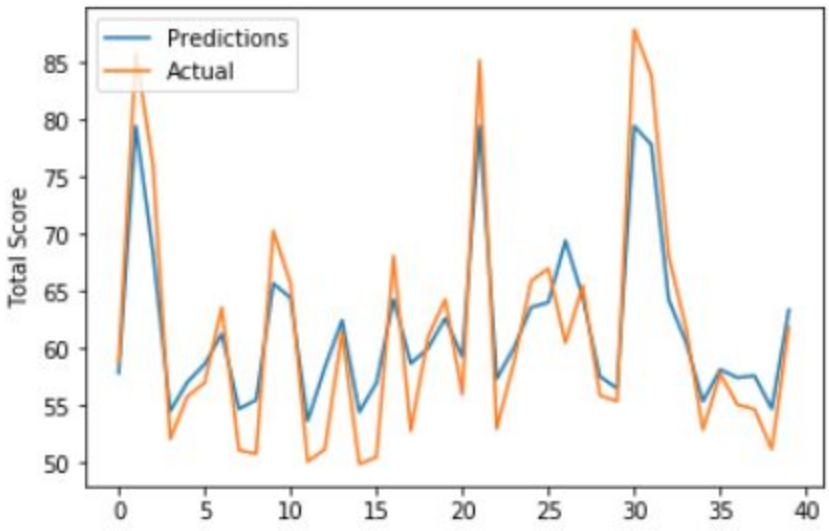

Figure 1: The difference in predictions vs actual scores on the 40 test samples.

Visually, the LGBM model was able to do pretty well.

Adding in Wikipedia Data

I found this part of the challenge to be the hardest due to a few reasons:

1. Reliability of Wikipedia pages: Many of the universities' Wikipedia pages contain a lot of information that I'm not sure are useful. For example, I'm uncertain as to whether student associations, or how the campus looks like impacts the final score. This is because the "Times Higher Education World University Rankings" only look at specific things such as research, teaching, and international diversity. If we find other features that are not at least somewhat related to impact their criteria, it is unlikely to be helpful.

2. Consistency of Wikipedia pages: Not all of the university's information are structured similarly. For example, one of the most useful section I found was the info table on the right. However the items in the tables differed amongst university pages. A value that I wanted to look into further was the size of the endowment fund. However as I was taking a quick browse, I found not all universities list their endowment fund size, especially those on the lower end of the rankings. In addition, not all endowment funds are listed in the same currency unit and would require conversion (which takes into account currency fluctuations over time based on when the pages were last edited).

Because of these two reasons, the majority of the content on Wikipedia pages are not useful for a baseline model. I ended up integrating only one feature, the type of the university - whether it was public/state funded, or private/autonomous. When I added this feature into the pre-existing dataset, only three universities did not have a type. I made the decision to leave them as NaN, as they did not represent a large portion of the data, and LGBM automatically can handle NaN values.

I was able to extract data from Wikipedia using the Python wikipedia api (to search for the exact name of a university's page), and BeautifulSoup (to web scrape the table info from page).

After re-running the model on the new dataset (with the same model settings and split), the MSE went down to 4.06.

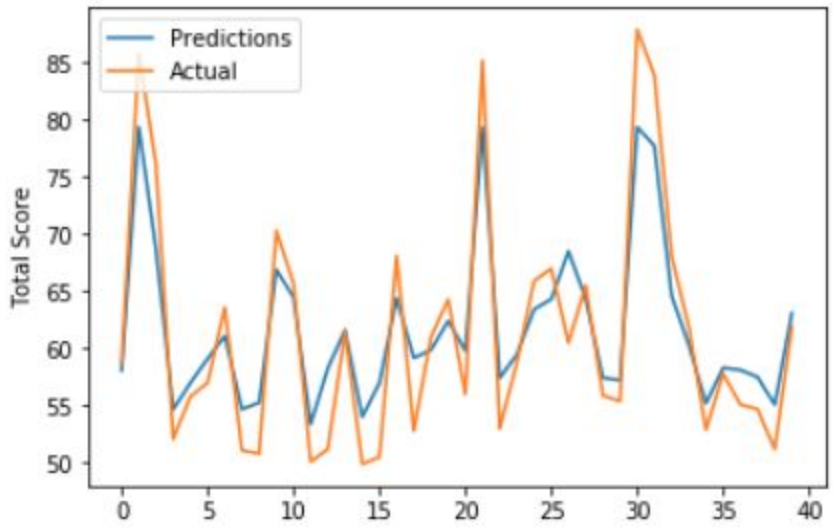

Figure 2: The difference in predictions vs actual scores on the 40 test samples on the new dataset.

While adding in the "type" feature resulted in a decrease in the error, it wasn't very significant. This is further confirmed after running a SHAP feature importance explainer.

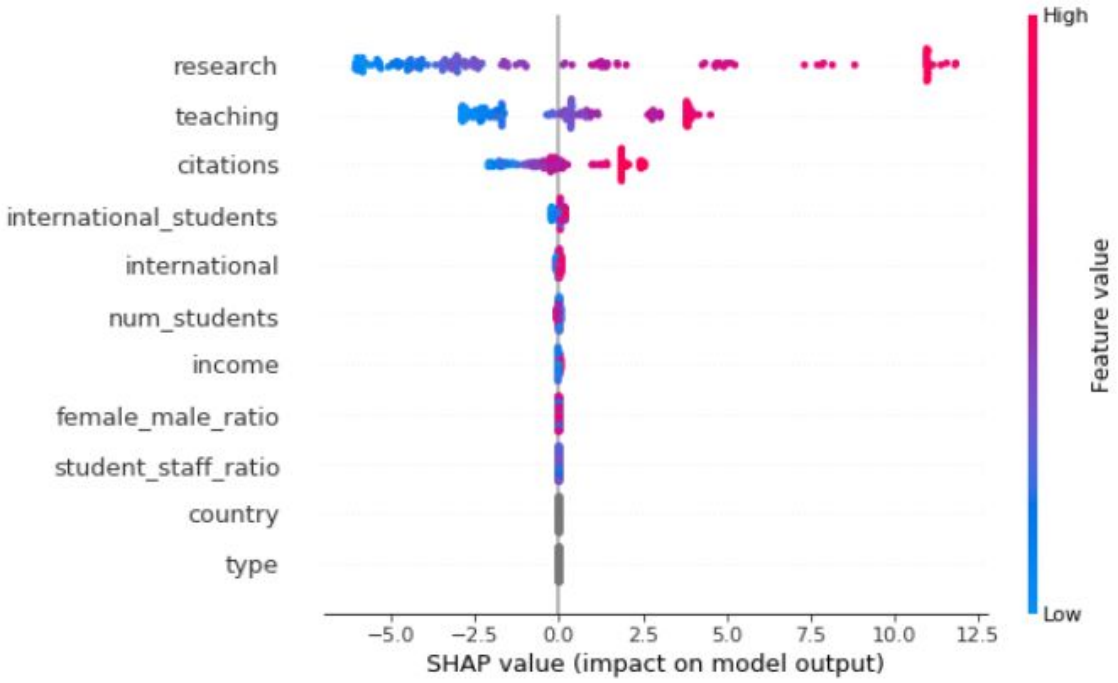

Figure 3: SHAP values for the LGBM model

As can be seen, the type of university doesn't have a large impact on model performance.

The last thing I tried was cross validation using K-Fold. This was done because I wanted to have diversity in the train-test split, and make sure the model worked on different test sets. I chose a 5-fold CV process, training one LGBM model per fold. At the end the score predictions of the 5 models were average to give a final MSE of 4.38. I believe this is the most accurate representation of the baseline model's performance.

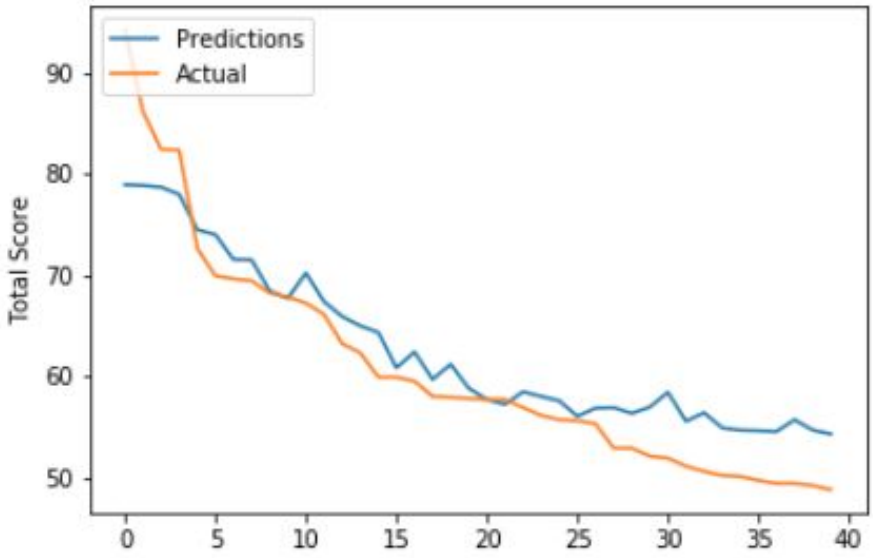

Figure 4: 5-fold result. The downward sloping is because sklearn's K-Fold automatically sorts the indexes. The data during the fold selection was shuffled.

Next Steps

As initially hypothesized, successfully creating a baseline model gave me a better idea on what to work on next in the future or look into further.

1. Ranking buckets: Another way to frame the problem would be as classification on the ranking buckets (16 buckets of 50: from 1-50, all the way to 751-800, or 4 buckets of 200: from 1-200, all the way to 601-800). This way, all 800 universities can be included. The dataset is still in order of rankings even after 200th place. This approach sacrifices the precision of knowing exactly where a university is but can tell you broadly what "tier" a university belongs to. We can combine the classification and regression approach, so only if a university appears in top 200 then we run regression to see exactly where it stands.

2. Weighing criterias: According to “Times Higher Education World University Rankings”, the universities are ranked according to the following criterias industry income (2.5% weight), international diversity (5% weight), teaching (30% weight), research (30% weight), citations (32.5% weight). Although LGBM automatically scales the weighting of categories, I believe this is important for other models to take into account. At the least this weighting requires a bit of investigation.

3. Alumni as a feature: Another possible Wikipedia feature that could be important is notable alumni. I suspect high ranking universities produce influential people. So it would be good to find exactly what notable alumni a university has and match their exact influence (or how well known they are perhaps through Forbes rankings).

4. Ensemble of different rankings: By integrating the other two rankings dataset, we can train individual models on each dataset and combine their results at the end. The three rankings rank universities slightly differently and this can be a good start for diversity.

Conclusion

In the span of a few hours, a baseline regression LGBM model was deployed and was able to predict the score of a university in the top 200 of “Times Higher Education World University Rankings”. I opted for a safer approach by using less data and being more selective about the features being used. External Wikipedia data proved to be difficult to integrate, but the type of university was chosen at the end as the one external feature. I believe I could have integrated much more Wikipedia data (sentence embeddings of sections, endowment size, etc.), but I’m not sure it would have been helpful and adding in features just for the sake of adding features isn’t always a good idea.