HW1 Big Data Platform

Dataset

Our chosen dataset is World University Rankings.
It gathers University Rankings and their affecting criteria's from 3 popular rankings over the years. These rankings are:

- CWUR (Center of World University Rankings)
- Times Higher Education World University Rankings
- Shanghai Ranking (The Center of World University Rankings).

We chose it cause we though we could detect high education trends across the world and to find correlations between multiple criteria that define the intuitions quality.

Database Schema

We decided to arrange the data in 3 tables:

1. **"schools_combined_worlds_ranks"**:
   Stores the CWUR, Shanghai and Times rankings and total scores of the universities over the years, partitioned by country.
   a. Partition Column: country'. To filter and group by it. To collect per country stats. And to be able to effectively compare country 's schools.
   b. Clustering Columns: 'year' – to be able aggregate by it, and limit result to given range of years.
   'university_name' – to make the PK unique.

2. **"University_Criteria":**
   Gathers all the supporting criteria's (scores, ranks, statistics) the different rankings use to rank a university in a single table. To be able to compare them easily.
   a. Partition Column: 'university_name'. Assuming that this table will be accessed given a university, to support rankings.
   b. Clustering Column: 'year'. Order by it and make PK unique.

Data ingestion

In the data ingestion process outlined in the "ingest to tables.py" file, we used three different data sets that hold all the ranking and criteria data per each rank. A significant challenge arose from the inconsistency in university names across datasets. For instance, 'The University of Texas Southwestern Medical Center at Dallas' was referred to as 'The University of Texas Southwestern Medical Center at Dallas' in the Shanghai ranking but as 'University of Texas Southwestern Medical Center' in the CWUR ranking. This discrepancy complicated the merging process between the datasets for our two Cassandra tables. To address this issue, our code employed several methods to identify similarities between names (strings):

1. Detecting if one name is a substring of another.

2. Utilizing a Python string matching library based on Levenshtein distance, which proved effective in resolving most merging issues.
3. Implementing hardcoded solutions for a few universities whose names were not detected by the aforementioned methods.

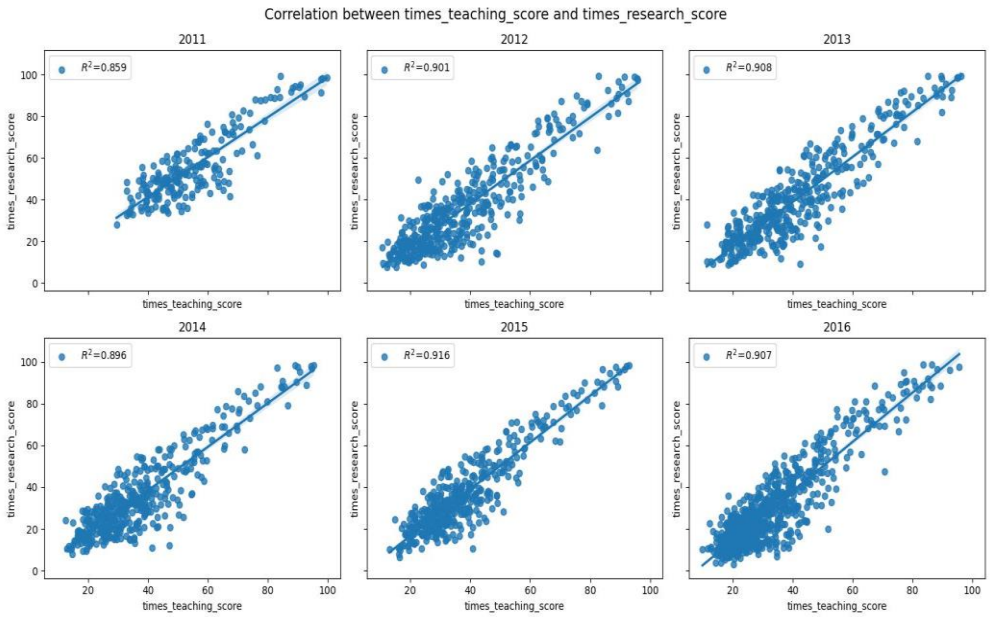
Queries

1. Our queries are designed to be generic, accommodating multiple arguments such as schools, countries, rank, and criteria provided by the user. The rationale behind these queries is outlined in the 'Queries Descriptions' file.
2. The generic queries can be found in python files: university_criteria_queries.py, University_World_Rankings_queries.py.

Key Insights derived from data.

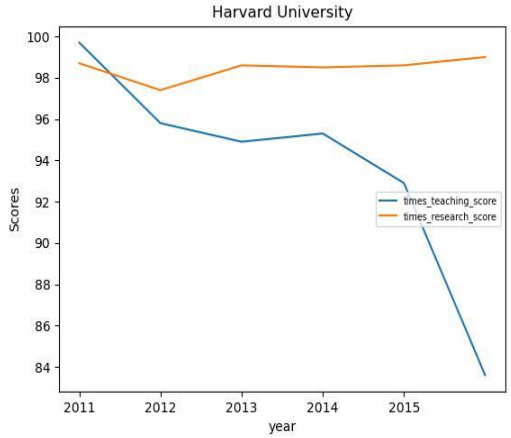
Relationship between teaching quality and research quality

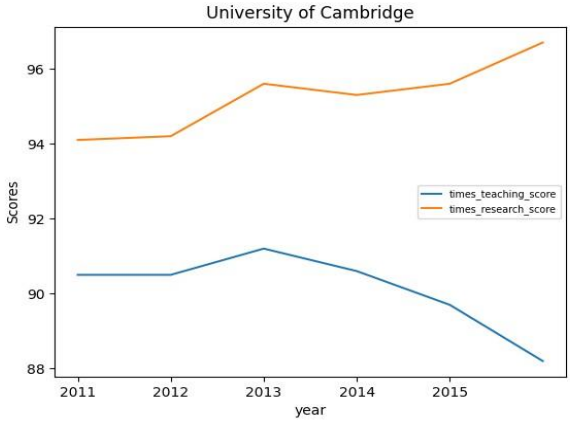
We found that there is a strong correlation between teaching and research scores. The finding may suggest that, indeed, better teaching leads to the growth of better researchers. Another



In the other hand, our findings across several sampled universities indicate a contrasting trend between teaching and research scores. This observation suggests a potential trade-off between the emphasis placed on teaching and research efforts within institutions. Institutions may prioritize one over the other, leading to inverse trends in their respective scores.

In Harvard and Cambridge Universities and for example we could observe the suggestion:

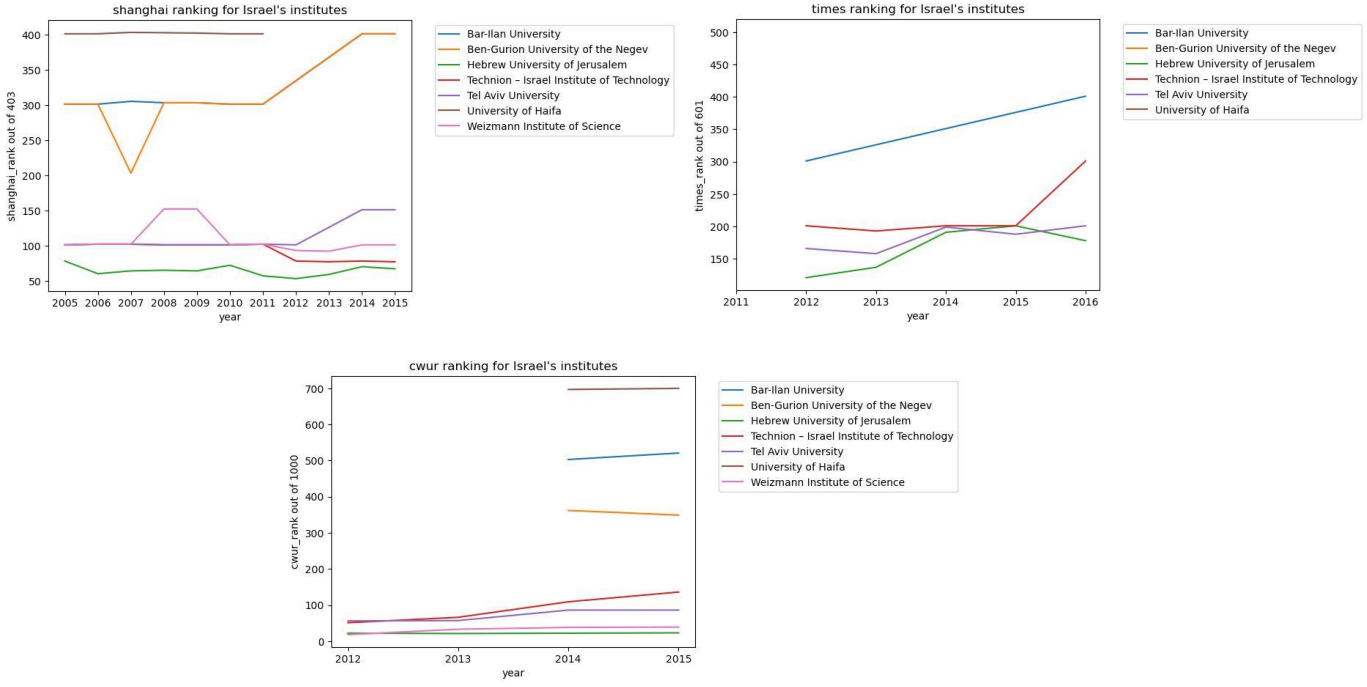




Israel ranking.

In general, Israeli institutions do not appear to be ranked highly in comparison to those around the world. There is, however, a consistent trend among them in which Hebrew University consistently ranks at the top.

Israel ranking charts:



Israel total ranked schools

israel

	year	total_schools_ranked_in_shanghai	total_schools_ranked_in_cwur	total_schools_ranked_in_times	total_ranked_universities
0	2012	4	4	4	5
1	2013	3	4	3	4
2	2014	6	7	3	7
3	2015	6	7	3	7