padh.ai – AI for Education

Data X2: Project Report

Plaksha Tech Leaders Fellowship

Sunit Jain (sunit.jain@plaksha.org)

Pranav Suri (pranav.suri@plaksha.org)

Shruti Verma (shruti.verma@plaksha.org)

Bharatdeep Maan (bharatdeep.maan@plaksha.org)

Gursehaj Pal Singh Aneja (gursehaj.aneja@plaksha.org)

INDEX

Abstract

As artificial intelligence becomes an increasing part of our daily lives, it's no wonder that educational institutions are racing to catch up with the need to develop more talent to keep the engine of AI development running. However, not only is education being transformed as far as science, technology, engineering, and math (STEM) curricula, but the education industry as a whole is being transformed by AI. During the course of our DataX2 project lab, we decided to invest in implementing solutions driven by data to make online education experience closer to classrooms. Continuing from Project Drona (as developed in DataX1), we extend the possibilities to a multitude of data-driven solutions such that learning management systems can adopt them for a better online learning experience. First, we developed a recommender system that provides suggestions optimizing for content similarity rather than user engagement. Second, we built a system to auto-grade essays. Third, we developed an engine that predicts the type of answer a question requires. Further, we've assessed examination scores of a sample of students over a period of time to assess class progress from an instructor's point-of-view.

Keywords: artificial intelligence, edtech, education, data science

Part-I: Project Drona: Recommender for Educational Content

Online video hosting platforms such as YouTube are representative of large and sophisticated industrial recommendation systems in existence. Their system is built with two neural networks: one for candidate generation, and the other for ranking. The two-stage approach to provide recommendations from a very large corpus of videos is optimized for personalization of the content displayed to the user towards maximum engagement. Maximum engagement doesn't account for the relevancy of the videos with one another, which is rather critical when viewing video content for educational purposes. Hence, we decided to create a recommendation system that suggests related videos based on the content of the videos, rather than user activity.

To start, we needed data in a form that is representative of the content of the video; a close form to which is textual transcripts, which transcribe the vocal content to a large extent and any information lost is in the form of visual elements that aid the transcript. The data collection strategy is described further.

1. Leveraging YouTube's filtering options with search queries, we searched for 'topic + tutorial' (where 'topic' is the subject we are searching for) to get playlists for that topic.
2. To crawl the playlists from the obtained search results, we used Selenium and BeautifulSoup (to obtain all required links for crawling).
3. Next, we used *youtube-dl*: to get a list of all videos (ids) on those playlists and through those ids, transcripts and all related metadata of all the videos. A Python script was written to write youtube-dl queries in Python and store the data as a JSON dump.

After suitable data-processing, we took transcripts of each crawled video and vectorized them using TF-IDF word embeddings. We then used cosine similarity of each video's transcript's word embeddings with the embeddings of the other videos and then selected the top-five most similar video pairs, which were the recommended videos.

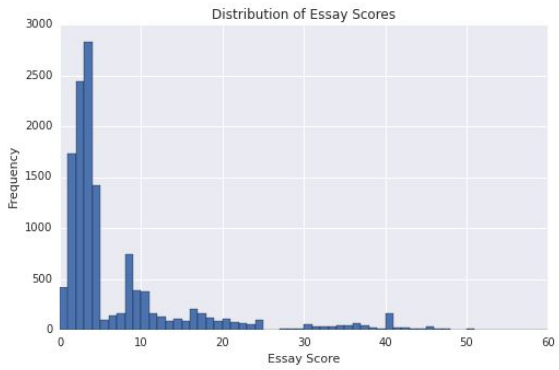
Sample recommendations (partial): for each video, a corresponding list contains the top-5 most similar videos (numbers) for that particular video.

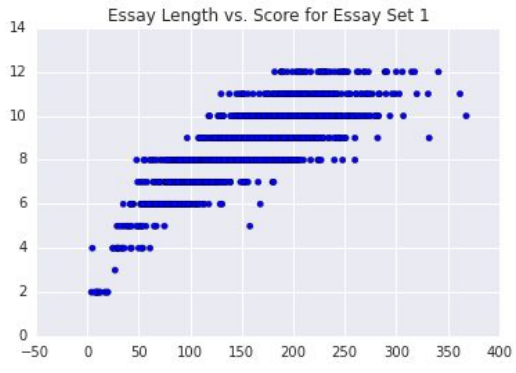
```
[[44, 61, 56, 76, 28], [52, 61, 56, 47, 72], [13, 4, 63, 41,
61], [61, 56, 72, 52, 47], [13, 41, 72, 63, 52], [52, 61, 56,
76, 47], [58, 56, 44, 61, 38], [5, 56, 52, 61, 76], [72, 52, 61,
41, 56], [16, 32, 61, 34, 52], [61, 52, 56, 41, 70], [13, 61,
56, 76, 5], [61, 15, 56, 52, 76], [72, 41, 61, 52, 56], [56, 52,
72, 61, 70], [28, 61, 52, 56, 5], [32, 9, 76, 61, 33], [19, 25,
33, 61, 52], [61, 56, 44, 52, 39], [17, 33, 25, 16, 61], [76,
52, 56, 61, 46], [26, 22, 52, 61, 72], ...
```

Part-II: Auto-Grader for Essays

To allocate funding and target educational support appropriately, we need to accurately measure how well students perform academically. To assess general writing ability, written essays are an invaluable tool. Unfortunately, grading of essays at scale is costly and resource-intensive. As a result, school systems turn to easier-to-grade alternatives, like multiple-choice tests. One solution to this problem is **automatic essay grading.** *Automatic essay grading uses prior data on written essays (i.e. grades assigned by human judges) to predict the grades of future essays.* This saves a significant amount of time and educational resources and helps make essays a more feasible part of standardized testing.

We downloaded our data from a Kaggle competition hosted by The Hewlett Foundation. For this project, *we looked at roughly 12,000 essays from 8 separate standardized tests*, each essay written by students between grades 7 and 10. These written assessments were responses to open-ended questions posed by an instructor (e.g. "When have you shown patience?"). Each of the essays was graded by a human on a numerical scale specific to the essay prompt.

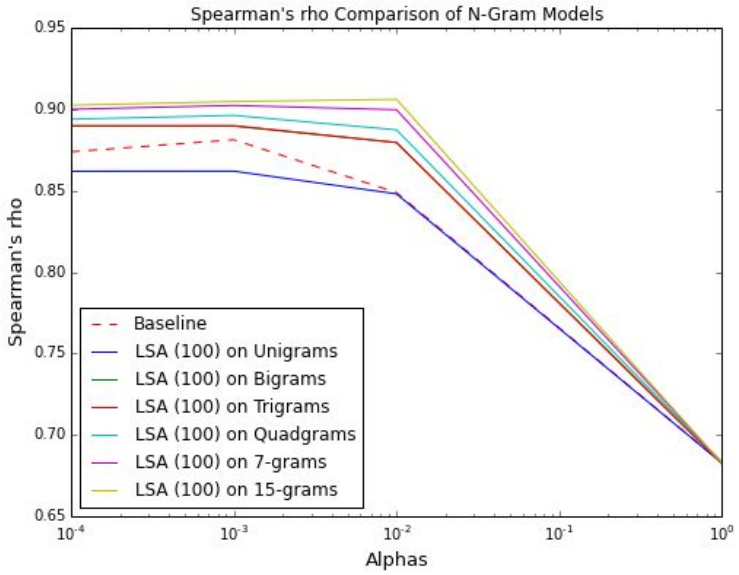




To break our data set into manageable components, we used two methods: one called *"text feature extraction", specifically TF-IDF, and another called n-grams*. These two approaches allowed us to extract characteristics of the essay and prompt text that we will use later for predictions. After breaking up our documents into TF-IDF weighted n-gram components, we applied *Latent Semantic Analysis to draw out different information about the relationship between words*. LSA at a high level allows us to capture underlying similarities between words that we wouldn't otherwise be able to capture using the n-gram approach alone.

After applying LSA and n-grams, we extracted a couple of features about the essays, namely the number of words and essay type. We then used these features as predictors in our model, a *linear LASSO mode*l. We decided to use LASSO as opposed to other regularization techniques because of its *dimensionality reduction characteristic*. When we use an n-gram model over our essay corpus, we have over 50,000 predictors. *LASSO allows us to eliminate many of those predictors that are not predictive of the essay score.*

- **Cross-validation of tuning parameter:** We used 4-fold validation to find the optimal tuning parameter for our LASSO model. We then did tuning across models without using LSA (i.e. just using the n-gram model) and then compared that to models that employed LSA. Finally, tuned across various sized n-grams to find our optimal model, which ended up using n-grams of size 15 (roughly the size of the average sentence).



- **Our accuracy metric:** Spearman's Rho Correlation – to evaluate our models, we used Spearman's rank correlation, which is a Pearson correlation between the sorted values of two variables; it can measure non-linear relationships.
- **Results:** Ultimately, we found that a 15-gram linear LASSO with LSA applied (100 components) produced the best model, yielding an accuracy of about 0.91.

We learned two main things in the model building phase: *Non-reduced unigram models perform better when LSA is not employed.* Also, LSA gives us increased performance in models with bigrams and above. This is probably because the information is encoded in bigram models in ways that it is not for unigram models (e.g. it picks up on the difference between "not good" and "good").

Part-III: Predicting Answer-Type for a Given Question

Understanding the questions posed by instructors and students alike plays an important role in the development of educational technology applications. Here, we applied NLP to one piece of this real-world problem by building a model to predict the type of answer (e.g. entity, description, number, etc.) a question elicits. The tasks were performed in the following order:

1. Perform preprocessing, normalization, and exploratory analysis on a question dataset
2. Identify salient linguistic features of natural language questions, and experiment with different feature sets and models to predict the answer type.
3. Use powerful pre-trained language models to create dense sentence representations and apply deep learning models to text classification.

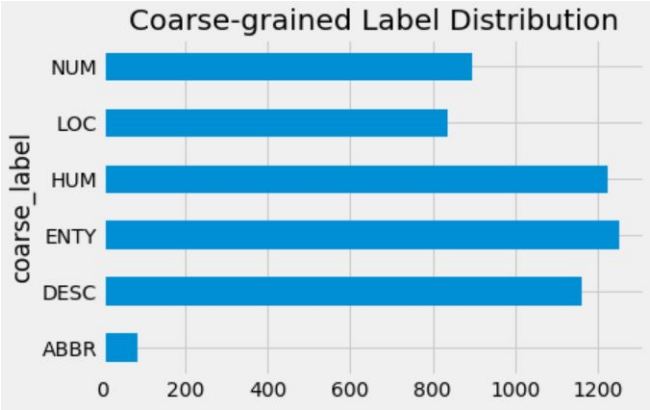
We implemented it using NLP and ML packages like SpaCy, Scikit Learn, and Tensorflow.

Exploratory Data Analysis

https://cogcomp.seas.upenn.edu/Data/QA/QC/"https://cogcomp.seas.upenn.edu/Data/QA/QC/ – First, we downloaded the train and test data from Xin Li, Dan Roth, Learning Question Classifiers. COLING'02, Aug. 2002.

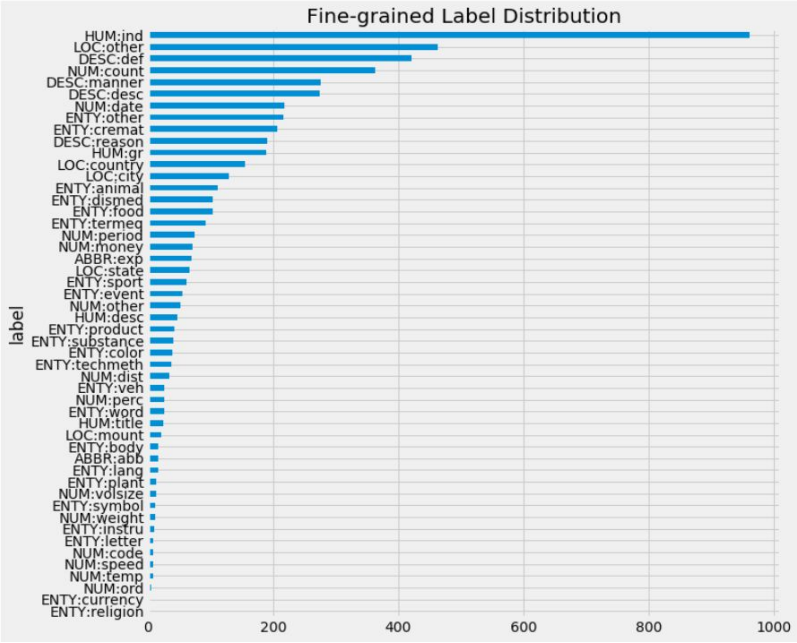
For Exploratory Data Analysis, we load train and test data into Pandas DataFrames, process question text with a pre-trained SpaCy English model, followed by observing the labels and question text in an effort to inform the next steps in the ML pipeline (namely, feature extraction and text classification).

Coarse-Grained Label Distribution



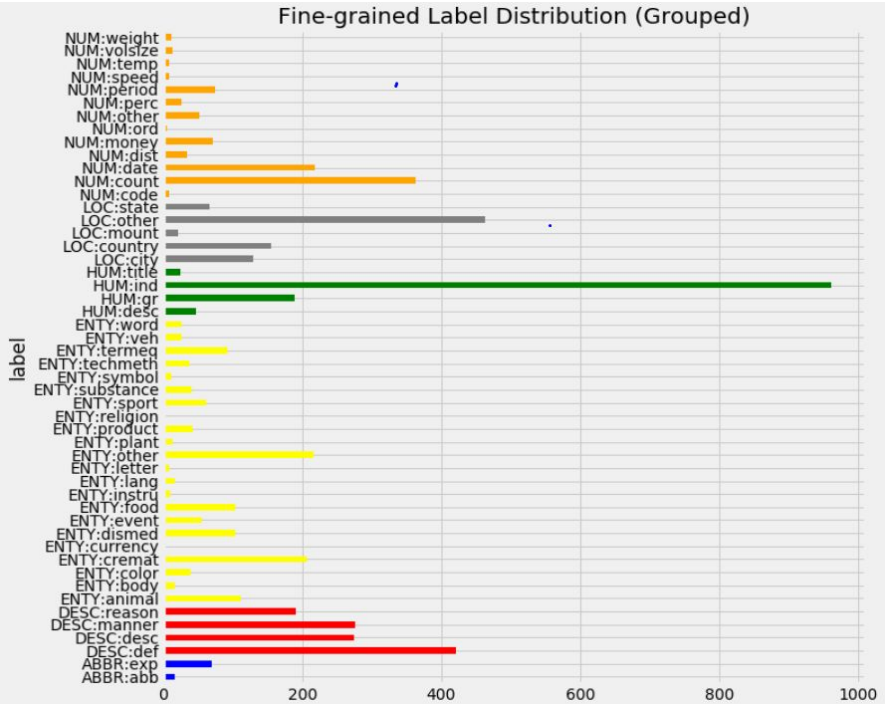
The coarse-grained labels are similarly-distributed, with one exception, ABBR.

Fined Grained Label Distribution



Unlike the coarse-grained label distribution, the fine-grained label distribution is quite skewed, with "HUM:ind" (human-individual) far more frequently than any other fine-grained label.

Fined Grained Label Distribution (Grouped)



Even though HUM:ind is far more frequent than the rest, its coarse-grained class has a small number of fine-grained labels when compared to the rest of the coarse-grained labels (ENTY, for example, is spread across 22 fine-grained labels vs 3 for HUM).

Explore the Questions

Some of the question texts by sampling a few questions per coarse-grained label

- Who sang about Desmond and Molly Jones?
- Who won the first general election for President held in Malawi in May 1994?
- Which French leader sold Louisiana to the United States?
- Who accompanied Space Ghost on his missions?

Some things to notice about HUM, many of the HUM questions start with 'Who'. This suggests that the specific question word in the sentence may offer some indication of its label.

Sample Questions for the sample “ABBR” (Abbreviations)

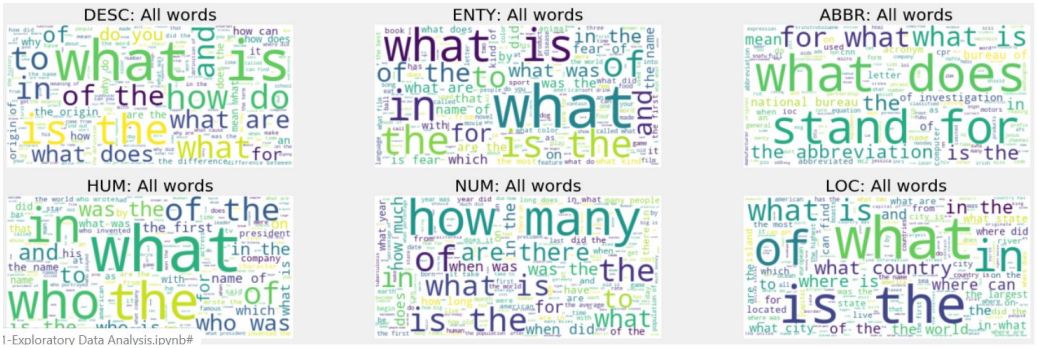
- What is DSL?
- What does the abbreviation IOC stand for?
- What is the full form of .com?
- What does e.g. stand for?
- What does pH stand for?
- What does LOL mean?

Some things to notice about ABBR these questions mostly tend to follow: *"What is [A-Z]+?", "What does __ stand for?"*

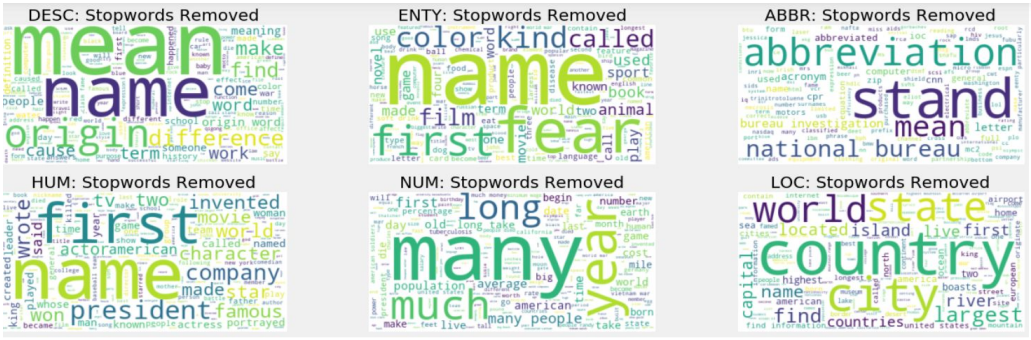
Word Frequency Cloud

We used the Wordcloud package to visualize the word frequencies for the different coarse-grained labels. By default, Wordcloud removes words that are on its "stopwords" list.

Word Cloud For All The Coarse Group Words

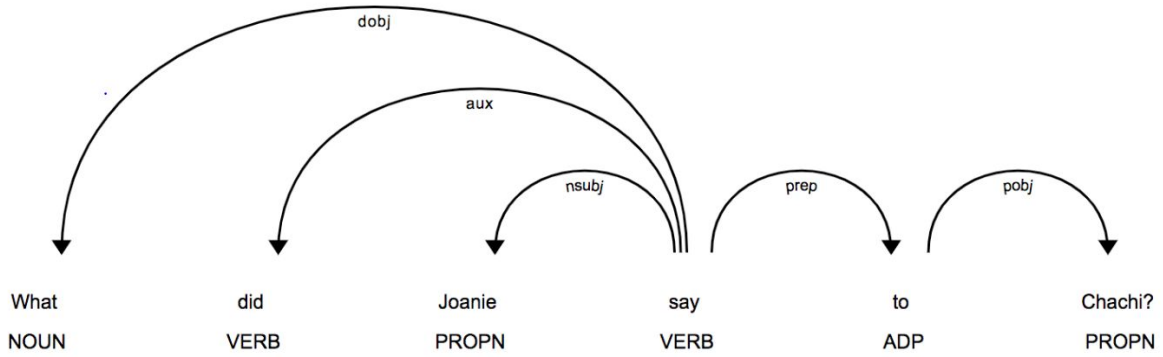


Word Cloud For All Removed The Coarse Group Words



Feature Extraction and Classification

We have broken down these features into "Syntactic Features" (word n-grams, part-of-speech n-grams, dependency triplets) and "Semantic Features" (brown-cluster n-grams, named-entity types, primary np-head hypernyms).



10 Fold Cross-Validation Using Multinomial Logistic Regression

We have performed 10-fold cross-validation, using the weighted F1-score, on multiple combinations of feature sets. We have used a Maximum Entropy (Multinomial Logistic Regression) model as the classifier. We have taken the 10th fold of our most recent model, and the features which the model found most impactful in the training data. The higher the coefficient value for a feature, the more impact it has on the final prediction.

Part-IV: Student Performance Analysis

[Dataset] M. Vahdat, L. Oneto, D. Anguita, M. Funk, M. Rauterberg.: A learning analytics approach to correlate the academic achievements of students with interaction data from an educational simulator. In: G. Conole et al. (eds.): EC-TEL 2015, LNCS 9307, pp. 352-366. Springer (2015). DOI: 10.1007/978-3-319-24258-3 26

The experiments have been carried out with a group of 115 students of first-year, undergraduate Engineering major of the University of Genoa. The data set contains the students' time series of activities during six sessions of laboratory sessions of the course of digital electronics. There are 6 folders containing the students' data per session. Each 'Session' folder contains up to 99 CSV files each dedicated to a specific student log during that session. The number of files in each folder changes due to the number of students present in each session. Each file contains 13 features.

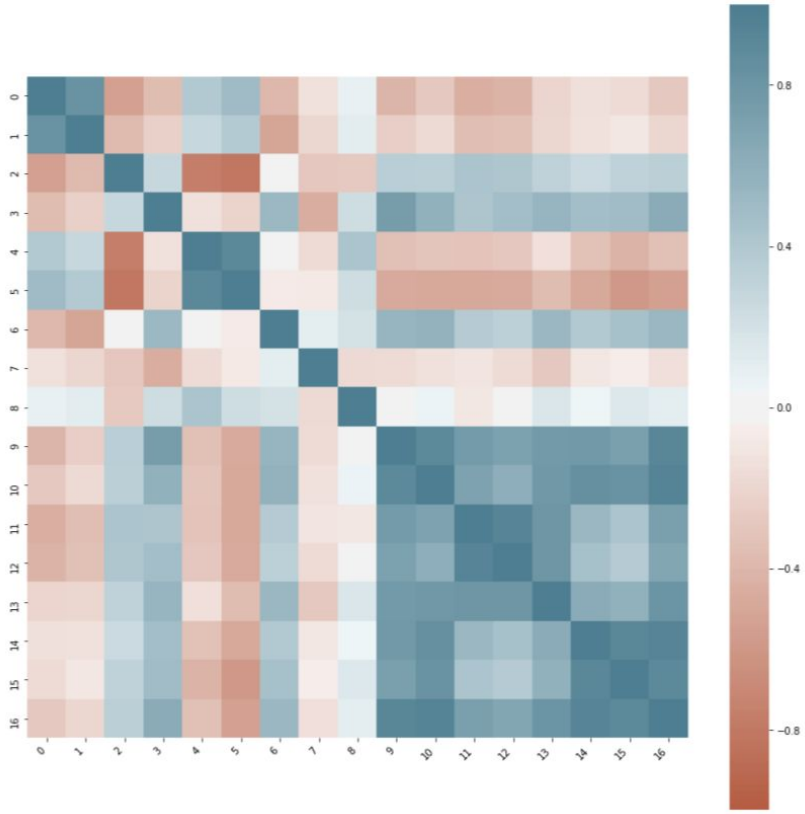
Students Performance Analysis is an intelligent way to assess a student's performance during the term of the course or degree. We are prototyping a way to measure performance through the results in final and intermediate exams using Machine Learning techniques. Using the dataset, we gather insights on the important features involved in the final scores and grades for a course.

Truncated view of data frame.

	Student ID	ES 1.1 (2 points)	ES 1.2 (3 points)	ES 2.1 (2 points)	ES 2.2 (3 points)	ES 3.1 (1 points)	ES 3.2 (2 points)	ES 3.3 (2 points)	ES 3.4 (2 points)	ES 3.5 (3 points)	ES 4.1 (15 points)
0	3	2.0	3	1.0	2.0	1	2	2	2.0	3.0	15.0
1	6	2.0	3	2.0	3.0	1	2	2	0.0	3.0	15.0
2	7	2.0	3	1.0	1.5	1	2	0	0.0	3.0	5.0
3	10	2.0	3	2.0	1.5	1	2	0	2.0	3.0	11.0
4	13	2.0	3	2.0	1.5	1	2	2	2.0	3.0	14.5

There are 16 data-points for each student enrolled in the course. The final column has the total_marks.

Correlation Matrix



Here are a few insights which we found:

- There is a high correlation between the last 7 assessments and the final grade. Apart from those, 3rd, 4th and 7th assessment also correlated highly with the final grade. One of the reasons for that could be that the assessments towards the end carried a higher weightage.
- Assessments 3, 4 and 7 could be relatively more difficult as compared to the rest of the assessments because it had a relatively lower mean in the student scores.

Data Description: mean of 3rd (76%), 4th (63%), and 7th (71%) assessments being lower than rest of the assessments (>90%)

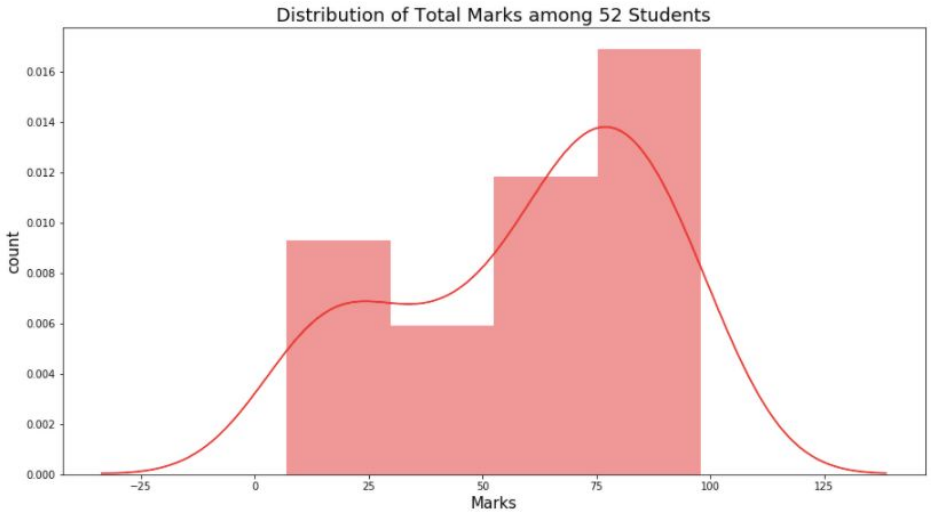
	Student ID	ES 1.1 (2 points)	ES 1.2 (3 points)	ES 2.1 (2 points)	ES 2.2 (3 points)	ES 3.1 (1 points)	ES 3.2 (2 points)	ES 3.3 (2 points)
count	52.000000	52.000000	52.000000	52.000000	52.000000	52.000000	52.000000	52.000000
mean	55.538462	1.884615	2.865385	1.432692	1.259615	0.942308	1.903846	1.423077
std	30.733811	0.351682	0.595040	0.569045	1.077808	0.235435	0.408710	0.914934
min	3.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	28.750000	2.000000	3.000000	1.000000	0.000000	1.000000	2.000000	0.000000
50%	57.500000	2.000000	3.000000	1.500000	1.500000	1.000000	2.000000	2.000000
75%	83.500000	2.000000	3.000000	2.000000	2.000000	1.000000	2.000000	2.000000
max	106.000000	2.000000	3.000000	2.000000	3.000000	1.000000	2.000000	2.000000

The total marks of 52 students followed the following distribution which followed the session-wise distribution of marks for 115 students across 5 sessions:

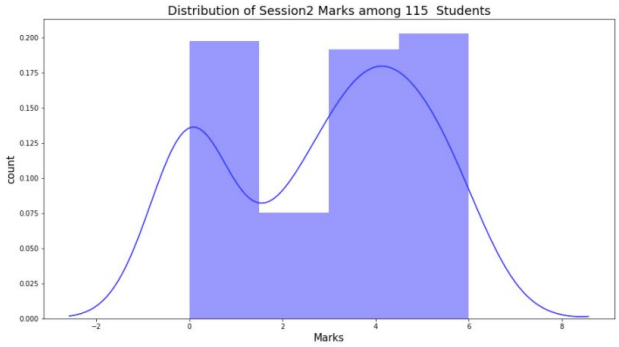
Session-wise marks for students

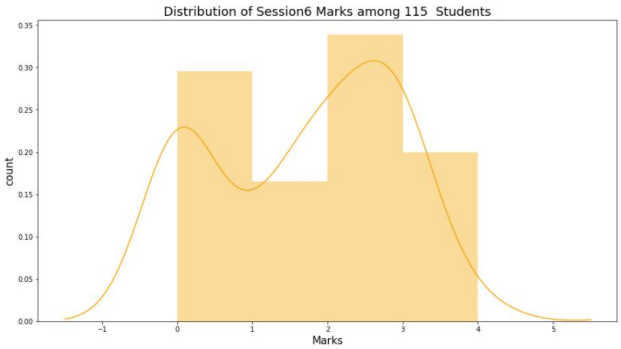
	Student Id	Session 2	Session 3	Session 4	Session 5	Session 6
0	1	5.0	0.0	4.5	4.0	2.25
1	2	4.0	3.5	4.5	4.0	1.00
2	3	3.5	3.5	4.5	4.0	0.00
3	4	6.0	4.0	5.0	3.5	2.75
4	5	5.0	4.0	5.0	4.0	2.75

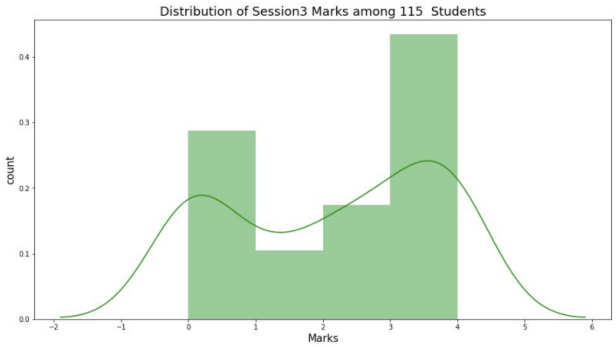
Distribution of Total Marks



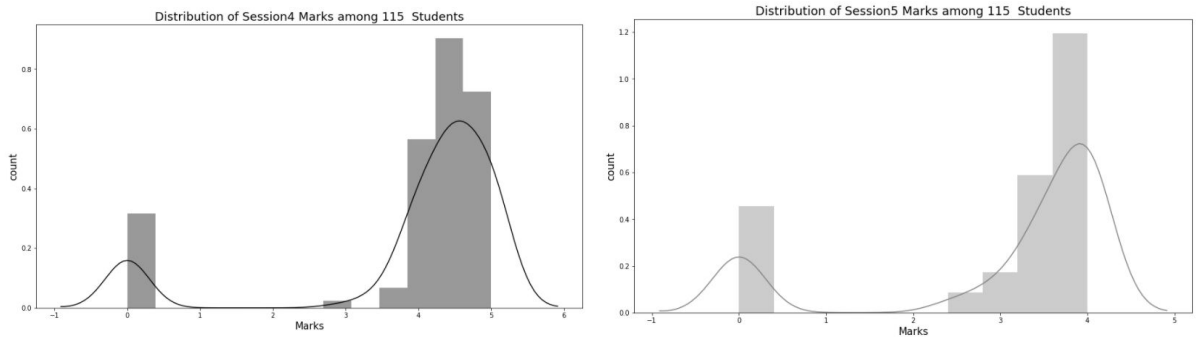
Session 2, Session 3 and Session 6 provided a great overlap between the distribution type.







Session 4 and Session 5 didn't represent the same distribution as the total marks distribution. These contained more binary grading as many students scored 0 in these sessions.



These insights we could obtain from a dataset that was constructed for a study, not in our control, show a lot of promise in measuring classroom performance, student learning, tutoring effectiveness – overall, explaining the gap between what the instructor taught and what the student perceived. Moreover, whenever a student stops making progress, it gives a good indicator of a point where the student possibly needs attention.