Question quality assessment

Github Link

https://github.com/HooKim/question_quality_analysis

Summary

I predicted the ranks of the quality of the questions based on a quality metric for questions using unsupervised machine learning model, Kmeans algorithm. This quality metric has to be similar with the expert's judgements. The goal here was to maximize the agreement between my quality metric and one from the experts.

Introduction

There are many online education platforms. These platforms are to provide many questions for learning students. However, how well these questions evaluate the students? These are really tough question because quality is not something we can compare and say that one is better than the other.

In this task, I needed to design a quality metric for questions. To this end, we compare the ranks we derived from the quality metric with the several pairwise comparisons from the 5 experts. Our objectives is to maximize this agreement rate. This is made by comparing a ranking list from the quality metric I designed with the ones from the experts.

Datasets

1. Main data to be trained with. In this data we can know the interaction between the students and the questions.
2. Metadata for Questions. In this data we can see the subjects which the questions belong to.
3. Metadata for Students. In this data we can check their date of the birth or gender.
4. Metadata for Subjects. This data is to figure out the subcategories of each subjects. They might belong to other subjects or include the others.

5. Metadata for Answers. It tells us the interaction itself. When students submitted or how confident they were.
6. Pairwise comparison data from the experts. The experts compare two questions and picked the better quality one. This data is later used to evaluate the quality metric I designed.

Features

1. Correct Rate for each questions. I could get this from the main data by using `groupby` methods.
2. Average Confidence students had for each questions. I joined the main data with the metadata of Answers. There were many missing values for the Confidence column. About 75% of the data were not in place. So I filled the missing values with the mean value of the remaining confidence data.
3. Average Age of students for each questions. I joined the main data with the metadata of Students. Likewise 50% of values in DateOfBirth column was missing. so I filled the values with the mean of the remaining ones. And I could get the age of students using DateAnswered.
4. How dispersed the topic was for each questions. Therefore I checked the metadata for Subjects. In there I could structure the tree of inclusion relationship between the subjects. I counted the number of leaf nodes. I think It represents how diverse the topic is for the question. I used this feature based on the Golden Rule by Craig B., which said that high-quality question should be clear and single skill.

Model

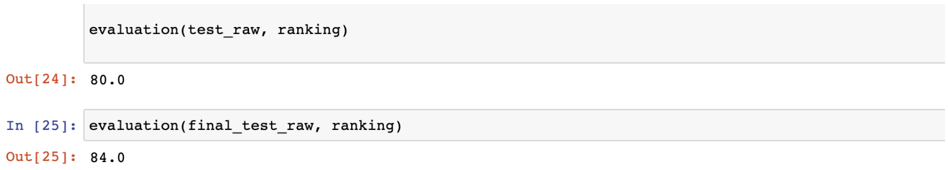
I used the Kmeans algorithm to design the model for the quality metric. Because we didn’t have the proper labels for the question whether it is good quality or not. In this case it is better to use unsupervised learning.

However, I need some index which I can refer when assigning rank to questions. So in addition to `KMeans` from `Scikitlearn` module, I imported methods called `silhouette_samples`. silhouette score indicates how the data point represents the group it is assigned to. so I decided to use this one as index for ranks.

This score value is within range from 0 to 1. and I had two categories one that is in a poor quality and one in a good one. For assigning ranks, I wanted this score to be in the continuous form. So I converted the scores from the one group in negative form. and the scores from the other groups remained still.

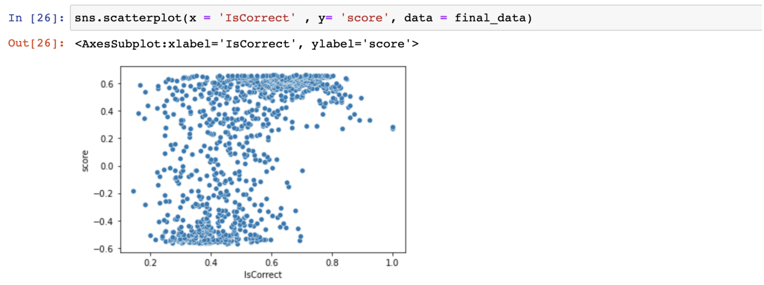
Results

I tested with with public one and private one respectively. The tricky part here was that I was not sure about which group represents the good quality group because this task is inherently unsupervised learning task. So I compared with two opposite cases : assume group 0 is a good quality and vice versa. And chose the case with the larger agreement rate. the other case were 20 and 16 respectively!

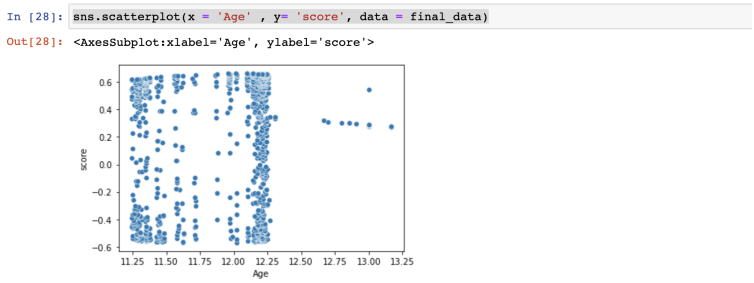


```
         evaluation(test_raw, ranking)

Out[24]: 80.0

In [25]: evaluation(final_test_raw, ranking)
Out[25]: 84.0
```

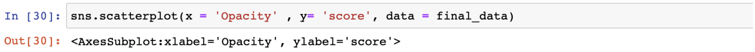
It seems Correct rate, Age and Opacity have nothing to do with the score



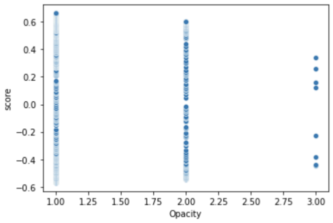
```
In [26]: sns.scatterplot(x = 'IsCorrect' , y= 'score', data = final_data)
Out[26]: <AxesSubplot:xlabel='IsCorrect', ylabel='score'>
```



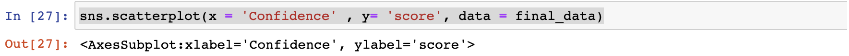
```
In [28]: sns.scatterplot(x = 'Age' , y= 'score', data = final_data)
Out[28]: <AxesSubplot:xlabel='Age', ylabel='score'>
```



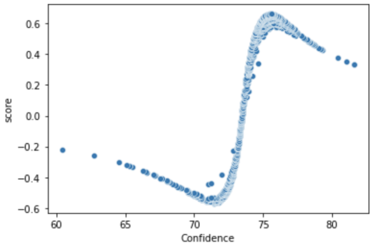
```
In [30]: sns.scatterplot(x = 'Opacity' , y= 'score', data = final_data)
Out[30]: <AxesSubplot:xlabel='Opacity', ylabel='score'>
```



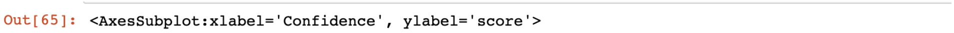
However I saw some meaningful pattern from Confidence feature.



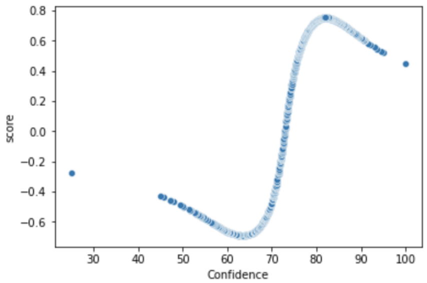
```
In [27]: sns.scatterplot(x = 'Confidence' , y= 'score', data = final_data)
Out[27]: <AxesSubplot:xlabel='Confidence', ylabel='score'>
```



At first I thought It was from the way I dealt with the missing values. So I dropped the all missing values and tested again. And still similar shape.



```
Out[65]: <AxesSubplot:xlabel='Confidence', ylabel='score'>
```



Discussion

Confidence is not proportional to the score. Rather it forms S-shaped which contains steep-slope zone where small increase in confidence leads to the big increase in score. Before reaching the zone Confidence decreases the quality. The beginning point of the zone is about 65 and it primes at 80. upon exiting the zone, Confidence decreases the score again.

Conclusion

It seems the confidence students take when encountering the question has impact on the quality. Therefore we need more data and study on this. What influences the confidence. if we can get to know this It will be easier to design the test which encourages student’s confidence properly.

References

- Results and Insights from Diagnostic Questions: The NeurIPS 2020 Education Challenge