Practical Data Science Assignment 3 Semester 2, 2021

IDENTIFIERS

- Name: Kai Run Leong
- Student ID: S3862092

OVERVIEW

The submission contains four python script files and three output files from the scripts. The objective of this assignment is to predict the users' ratings by using high quality algorithms from a dataset that contains information about customer reviews on beers that they have tried. Additional features such as the alcohol volume and the text review were also merged into the train dataset during this assignment to perform collaborative filtering through exploratory data analysis. Among the four submitted python script, “A3-4.py” contains code that uses Natural Language Processing to predict a user rating from the test set (advance algorithm).

MODEL DESCRIPTION

The algorithms that were used in this assignment were ***KNNBaseline***, ***BaselineOnly***, ***SVD,*** and the NLP algorithm which consisted of the ***TfidVectorizer*** and **GradientBoostingRegressor** from the scikit-learn library. The models were trained using the dataset from the “train.tsv” file and only two features were utilized from the “features.tsv” file for collaborative filtering and natural language processing. The features were alcohol volume by beer (ABV) and the text review. However, the text review was only used on the last algorithm while the first three algorithms made used of the ABV feature.

Starting with the model parameter tuning for the KNNBaseline algorithm, k was assigned 458 neighbours and the similarity measure configuration was set to cosine similarity and user based. The parameter k was set to 458 because it was found to be the most optimal value in reducing the mean absolute error based on the similarity measure and the elbow method. Furthermore, based on its current parameter settings the model for the KNNBaseline algorithm managed to score a mean absolute error of 0.4405, which was roughly 0.60% better than the baseline performance that was done during the experiment (refer to figure 1).

As for the BaselineOnly model, the method in the parameter configuration was set to alternating least square, and had an “n_epochs” of 15. With this model configuration for this particular algorithm, the model was able to produce a mean absolute error score of 0.4417 which had an improvement of roughly 0.47% from the baseline testing performance (refer to figure 1). The last prediction algorithm that was also from the surprise library was SVD. The model parameter settings had a n value of 10 and an “n_epochs” of 30. With this parameter the SVD model was able to achieve an even mean absolute error score of 0.4397 which was the lowest among the four algorithms that were used.

Lastly, the last algorithm that was used was an NLP based algorithm which made use of the ensemble and text feature extraction packages from the scikit learn library. Before the text was used for training, the text was cleaned twice and the parameter settings for both the ***TfidVectorizer*** and ***GradientBoostingRegressor*** model was set to default. Although a benchmark was not set for the NLP algorithm during the experiment, it still scored an interesting mean absolute error of 0.4622 which was still much better than the ***NormalPredictor*** model which scored an mean absolute error of 0.77 (Week 12 lectorial demonstration).

FEATURE ENGINEERING (optional)

No feature engineering was performed during this assignment, however the text feature had two levels of cleaning before fitting the train data into the model.

EXPERIMENTS

In this assignment, since a total of at least 3 algorithms were required and having one of the it to be advanced, it was decided that four algorithms would be made and one out of the four algorithms would be the advance algorithm. Starting with the experiment to find the top three simple algorithms, a benchmark was first made to ensure that other models could be tested against the baseline performance of the benchmark scores. If the mean absolute error of a particular model that is currently being experimented on scores a lower mean absolute error as compared to other models that are using the same prediction algorithm, then it would be a clear indication that the current model is not suitable for user rating prediction and that it requires further parameter refinement.

Out[9]:

Algorithm	test_mae	fit_time	test_time
BaselineOnly	0.446471	1.250937	1.932984
KNNBaseline	0.447377	9.672734	55.638303
SVD	0.451608	34.464788	2.400656
KNNBasic	0.458706	7.812342	45.083593
KNNWithMeans	0.462572	7.800552	46.836052

Figure 2: Benchmark for algorithm performance

Figure 2 contains the benchmark result that model must beat in order to be qualified as suitable model for prediction. While constructing the benchmark scores, it was decided that the training dataset that was to be used to create a baseline performance that models must beat had to be simple in order to create a fair and controlled test. Thus, only the features "ReviewerID", "BeerID" and "Label" were used after cleaning the training dataset. With the benchmark scores already set up for models to beat, the experiments on finding a suitable model for each of the algorithms can be performed.

For the first three algorithm which were ***KNNBaseline***, ***BaselineOnly***, & ***SVD,*** before its most optimal parameter was found (optimal parameters can be found in the model description section and in the code), most of the experiments involved looping multiple times to adjust

the values of either the k-neighbours, n-factor ,or n_epochs. The parameters that were able to produce the most optimal mean absolute error was usually chosen. However, the result usually appeared to improve only slightly, thus in order to improve the mean absolute error score, some features from the “features.tsv” file was chosen to filter out unwanted rows of data.

After performing a multitude of test and exploration data analysis on the different features that would help to lower the mean absolute error, it was found that the filtering out unwanted data from the alcohol volume (ABV) feature was the most effective because the mean absolute error score managed to improve. For example, during the exploratory data analysis process it was found that alcohol levels above 20 had close to zero ratings and after it was filtered out of the training data set, the mean absolute error had a slight improvement (refer to figure 2). Thus, it showed how important it was to ensure that unwanted data is removed to have a more accurate prediction.

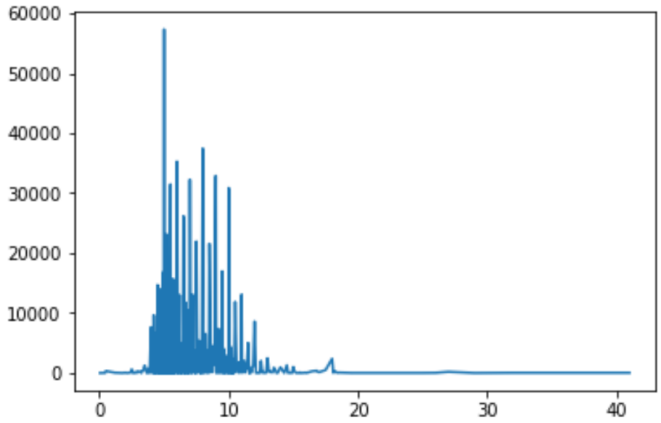

Figure 2: ABV Count (x-axis = ABV, y-axis = count)

Lastly, for the NLP algorithm, since the text was the most important and the only feature that was used, a separate experiment to see whether uncleanse data was more useful than filtering out information such as stop words and numbers. However, it turned out that cleaning out stop words, number and punctuation helped to further improved the score. Thus, allowing the algorithm to score a decent mean absolute error of 0.4622.

In conclusion, during the whole assignment it involved only refinement of the model parameters, filtering unwanted data and ensuring that not only did the refined models beat the benchmark score, but that it also could improve beyond that. Even though more could have been done, with the limited time allocated this was the best score that could be achieved thus far.

LIBRARIES USED

certifi==2021.10.8
joblib==1.1.0
numpy==1.21.3
pandas==1.3.4
python-dateutil==2.8.2
pytz==2021.3

scikit-learn==1.0.1
scikit-surprise==1.1.1
scipy==1.7.1
six==1.16.0
surprise==0.1
threadpoolctl==3.0.0

REFERENCES (APA 7th Edition)

- Unfold Data Science. (2020, May 1). *Hotel Reviews Sentiment Analysis In python|NLP Sentiment analysis in Python* [Video]. YouTube. https://www.youtube.com/watch?v=HpKMc78OYts&t=500s

Methods such as text_clean_1(text) and text_clean_2(text) in A3-4.py were taken from this author