IST 557 Data Mining: Techniques and Applications

FINAL PROJECT REPORT

Arun Teja Muluka, PSU ID: avm6604@psu.edu | Kaggle Username: IST557_Fa20_A3

Project A. Home Depot Product Search Relevance

Introduction

The project deals about improving the search engine directing customers to right products using machine learning techniques. For this task, the competition has provided some of the data from their search engine.

- train.csv - the training set, contains products, searches, and relevance scores
- test.csv - the test set, contains products and searches. You must predict the relevance for these pairs.
- product_descriptions.csv - contains a text description of each product. You may join this table to the training or test set via the product_uid.
- attributes.csv - provides extended information about a subset of the products (typically representing detailed technical specifications). Not every product will have attributes.

Data fields:

- id - a unique Id field which represents a (search_term, product_uid) pair
- product_uid - an id for the products
- product_title - the product title
- product_description - the text description of the product (may contain HTML content)
- search_term - the search query
- relevance - the average of the relevance ratings for a given id
- name - an attribute name
- value - the attribute's value

Relevance is measure on a numerical scale from 1 to 3.

Method

The main features that we are going to closely look, and extract feature are search_term, product_title and product_description. Brand of the product is also considered for a small set of features.

Feature Extraction:

1. Jaccard Similarity – Product Title & Search_term
2. Jaccard Similarity – Product Description & Search_term
3. Intersection Count – Product Title & Search_term
4. Intersection Count – Product Description & Search_term
5. Length of the search_term
6. Fasttext (https://fasttext.cc/): used to get the Sentence embeddings for various features like search term, title and descriptions. And the cosine similarity is taken as a feature.
    a. Score for search_term and product description
    b. Score for search_term and product_title
7. Length of the title
8. Length of the description
9. Length of the brand
10. Entire search word appears in title (true/false)
11. Entire search word appears in description (true/false)
12. Last word of the search term appears in description (true/false)
13. Last word of search term appears in title (true/false)
14. TFIDF vector of search term (vector of size 20)
15. TFIDF vector of description (vector of size 50)
16. TFIDF vector of title (vector of size 20)
17. Universal Sentence Encoder (https://tfhub.dev/google/universal-sentence-encoder/4): used to get the Sentence embeddings for various features like search term, title and descriptions. And the cosine similarity is taken as a feature.
    a. Score for search_term and product description
    b. Score for search_term and product_title
18. InferSent sentence Embeddings: used to get the Sentence embeddings for various features like search term, title, and descriptions. And the cosine similarity is taken as a feature.
    a. Score for search_term and product description
    b. Score for search_term and product_title
19. Fuzzywuzzy: It uses Levenshtein Distance to calculate the differences between sequences.
    a. Score for search_term and product description
    b. Score for search_term and product_title

External Libraries used: numpy, pandas, xgboost, nltk, sklearn, sister, tensorflow, pytorch, fuzzywuzzy.

Total 135 feature were extracted from the available dataset. The above listed features are the very important features from the feature space. Other can be added as ratios and similarities of sentence embeddings. Some of the features extracting functions were adapted from Kaggle.com like stemmer and word segmentations etc. But most of the features were self-extracted. Spell Checking for search terms was added and the method was obtained from Kaggle.

- All the above features are used to predict the relevance of the search term. As the relevance is a continuous value from 1 to 3, we use various regressor model to predict the values.

Best Performance Model: XGBRegressor(learning_rate=0.1,max_depth=4,n_estimators=100)

Reason: Best train and validation score.

Results

Models Explored:

1. RandomForestRegressor with Bagging(n_estimators=50, max_depth=6, random_state=0)
2. XGBRegressor(learning_rate=0.1,max_depth=4,n_estimators=100)
3. GradientBoostingRegressor(random_state=0, n_estimators=100, subsample=1.0)
4. DecisionTreeRegressor(random_state=0, min_samples_split=2, min_samples_leaf=1)
5. RandomForestRegressor(n_estimators=100,n_jobs=-1,verbose=1)
6. Neural Network with 3 layers ([135,8,16,1]=>Layer Units, loss='mse',optimizer='rmsprop')

Train and Validation Errors:

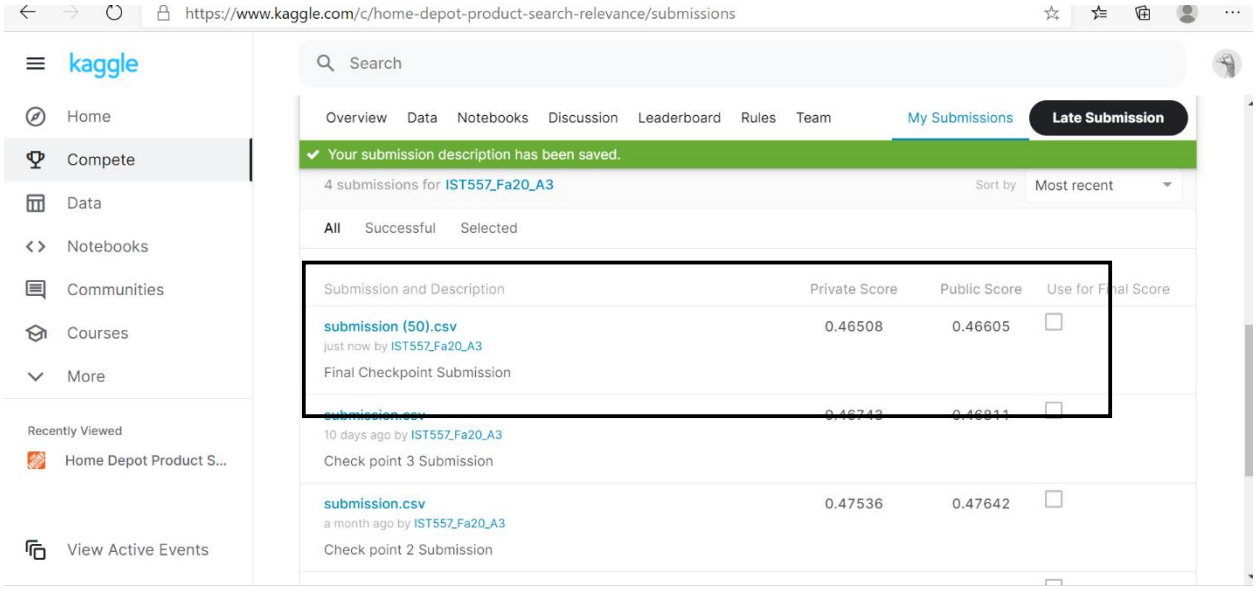
Model	Train Error	Validation Error
RandomForestRegressor with Bagging	0.46381	0.46566
XGBRegressor	0.44571	0.45503
GradientBoostingRegressor	0.45552	0.45881
DecisionTreeRegressor	0.01821	0.63185
RandomForestRegressor	0.17105	0.453463
Neural Network with 3 layers	0.54520	0.54382

Discussion:

- Random Forests are Quick in training, robust to outliers, robust to non-linear data, no scaling required but they lack in interpretability for ablation study and possibility of overfitting.
- XGBoost regressors Works better with complex data, no scaling required so perform better in most of the regression tasks.
- Neural Networks: Better for classification tasks. Apparently does not work very well in the regression tasks. In classification Neural networks have high potential of creating benchmark performances.
- Decision Tree Regressor: Does not require scaling and normalization of dataset. In the current dataset, the model treats the labels as categories which accounts for high validation error.

- From the results we can observe that some models have huge difference in train and validation error due to overfitting.

Screenshot: Public Score – 0.46605



Summary

- Initial phase of the project was observing and doing a exploratory data analysis. I got to learn various natural language processing techniques through this process.
- I learnt how to express natural language sentences in numerical format to apply machine learning techniques on them.
- I have implemented various regression models and through various parameter tuning techniques I have learnt how to choose best models from the available model.
- Search engine are vital part of internet and the project enabled me to learn how to analyze the engines data through machine learning methods and possibly apply them to improve the performance of the engines.
- Although this was an individual project, I have learnt how others approached and the methods they used through their presentations. I have adapted some of their methods to my method to improve the performance.
- My future plan would be participating in similar competitions to gain knowledge about other datasets and methods.