Movie Review Sentiment Analysis

Siyu Chen

Xi’an Shiyou University

(None)

Overview

Problem Definition

Problem Description

Description

The Rotten Tomatoes movie review dataset is a corpus of movie reviews used for sentiment analysis.You are asked to label phrases on a scale of five values.

- The sentences have been shuffled from their original order. Each Sentence has been parsed into many phrases by the Stanford parser.
- Each Sentence has been parsed into many phrases by the Stanford parser.
- Each phrase has a PhraseId. Each sentence has a SentenceId.
- Phrases that are repeated (such as short/common words) are only included once in the data.

Read Data

- train.tsv :Contains the phrases and their associated sentiment labels. Provided a SentenceId can track which phrases belong to a single sentence.
- test.tsv : contains just phrases. Assign a sentiment label to each phrase.
- sampleSubmission.csv : A submission that meets the purpose.

The sentiment labels are:

- 1 : somewhat negative
- 2 : neutral
- 3 : somewhat positive
- 4 : positive
- 5 : negative

Analysing Data

Data Statistics

- Statistic the data in the training set.

Table 1: Statistic the data in the training set

	PhraseId	*SentenceId*	*Phrase*	*Sentiment*
0	1	1	A series of escapades demonstrating the adage ...	1
1	2	1	A series of escapades demonstrating the adage ...	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2
5	6	1	of escapades demonstrating the adage that	2

- Statistic the data in the test set.

Table 2: Statistic the data in the test set

	PhraseId	*SentenceId*	*Phrase*
0	156061	8545	An intermittently pleasing but mostly routine ...
1	156062	8545	An intermittently pleasing but mostly routine ...
2	156063	8545	An
3	156064	8545	intermittently pleasing but mostly routine effort
4	156065	8545	intermittently pleasing but mostly routine

Dataframe Analysis

- To better process the data, we need to do the following:
  - Count the total data of training set and test set.
    train shape: (156060, 4)
    test shape: (66292, 3)
  - Check to see if there is a free value.
  - Count the unique sentences in the training set and the test set.
    train: 8529
    test: 3310
  - Create a data set with only complete sentences.
    Add a label for the sentiment value.

Create Dataset

- Exploring data will give cleaner graphs that aren't biased toward longer sentences.

Table 3: A data set containing only full sentences

	PhraseId	*SentenceId*	*Phrase*	*Sentiment*	*sent_label*
0	1	1	A series of escapades demonstrating the adage ...	1	Somewhat Negative
63	64	2	This quiet , introspective and entertaining in...	4	Positive
81	82	3	Even fans of Ismail Merchant 's work , I suspe...	1	Somewhat Negative
116	117	4	A positively thrilling combination of ethnogra...	3	Somewhat Positive
156	157	5	Aggressive self-glorification and a manipulati...	1	Somewhat Negative

NLP Analysis

TF-IDF Algorithm

- Model of TF-IDF algorithm

$$TF-IDF(d,w) = TF(d,w) * IDF(w)$$

- $TF(d,w) \Leftrightarrow$ Frequency of occurrence of w in document d.
- $IDF(w) = log\frac{N}{N(w)}$
- N $\Leftrightarrow$ The total number of documents in a corpuss.
- N(w)$\Leftrightarrow$ How many documents does the w appear in.

Deal With Stop Words

- Process stop words to find the most valuable ones.
  - Add non-helpful stopwords to stopword list.
  - Scikit-learn provides a TfidfVectorizer class, which has the ability to remove common stop words (like a, the, and, or).
  - Create bag of word vectorizer for comparison in evaluation section.

Create TF-IDF Graphics

- Look at the graphs below
  - The words with the highest tf-idf score in negative sentiment class.

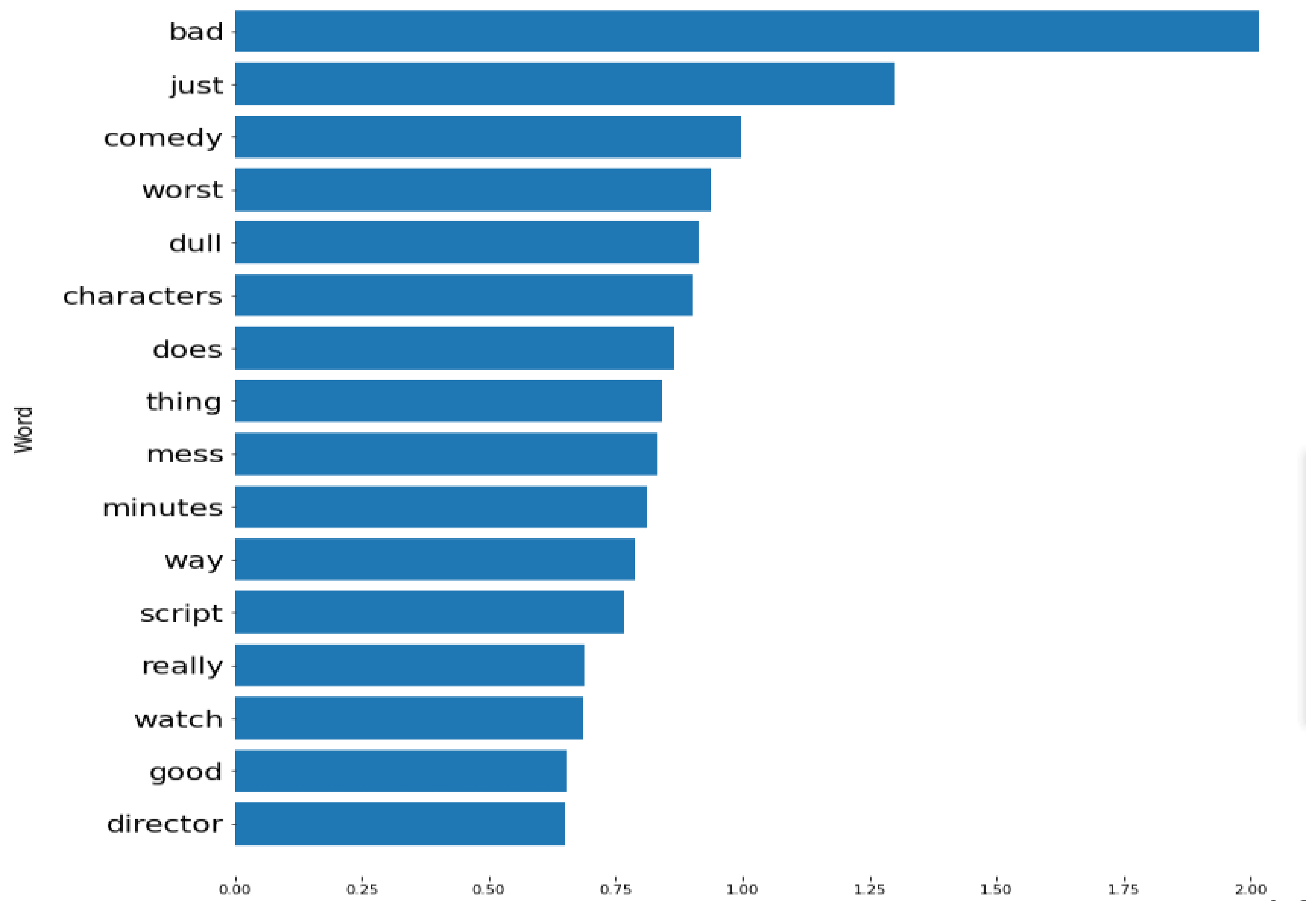

Figure 1: negative sentiment class

- The words with the highest tf-idf score in neutral sentiment class.

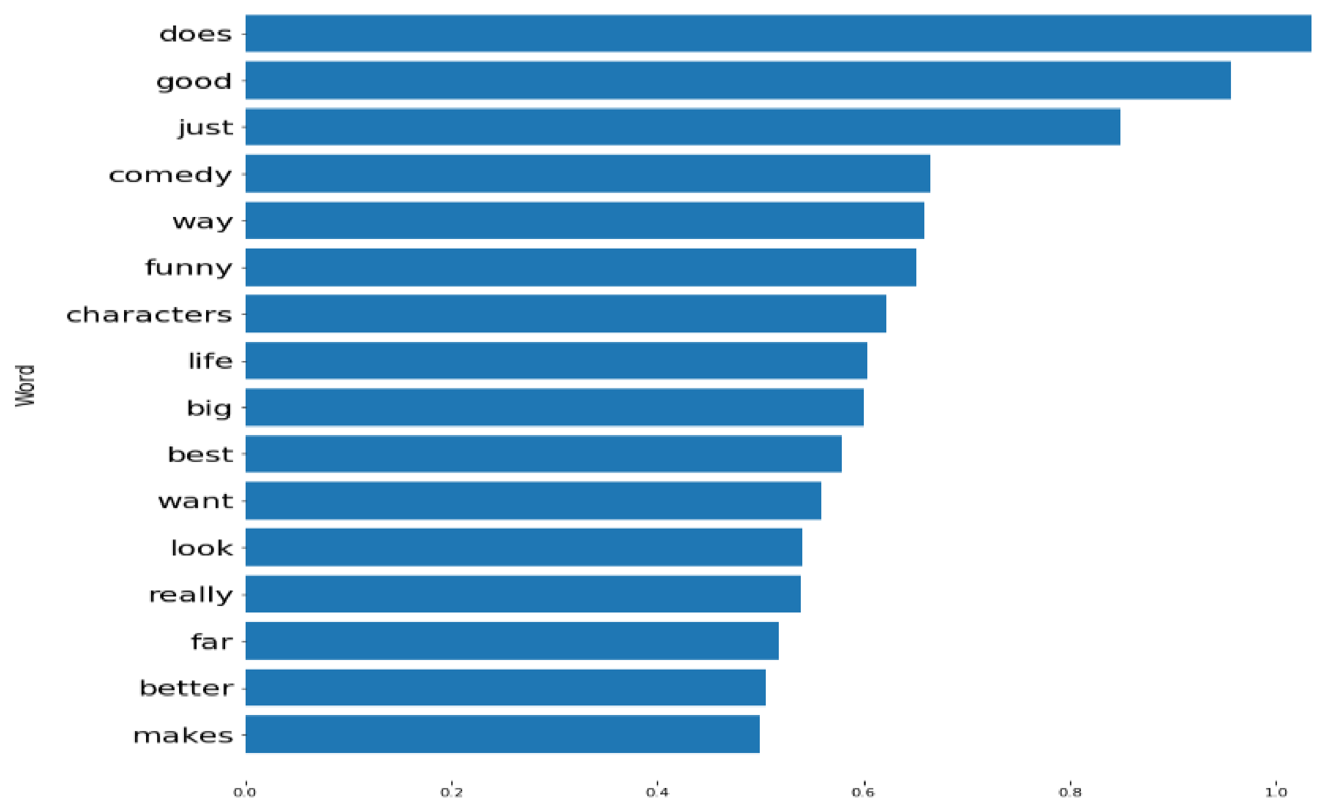

Figure 2: neutral sentiment class

- The words with the highest tf-idf score in positive sentiment class.

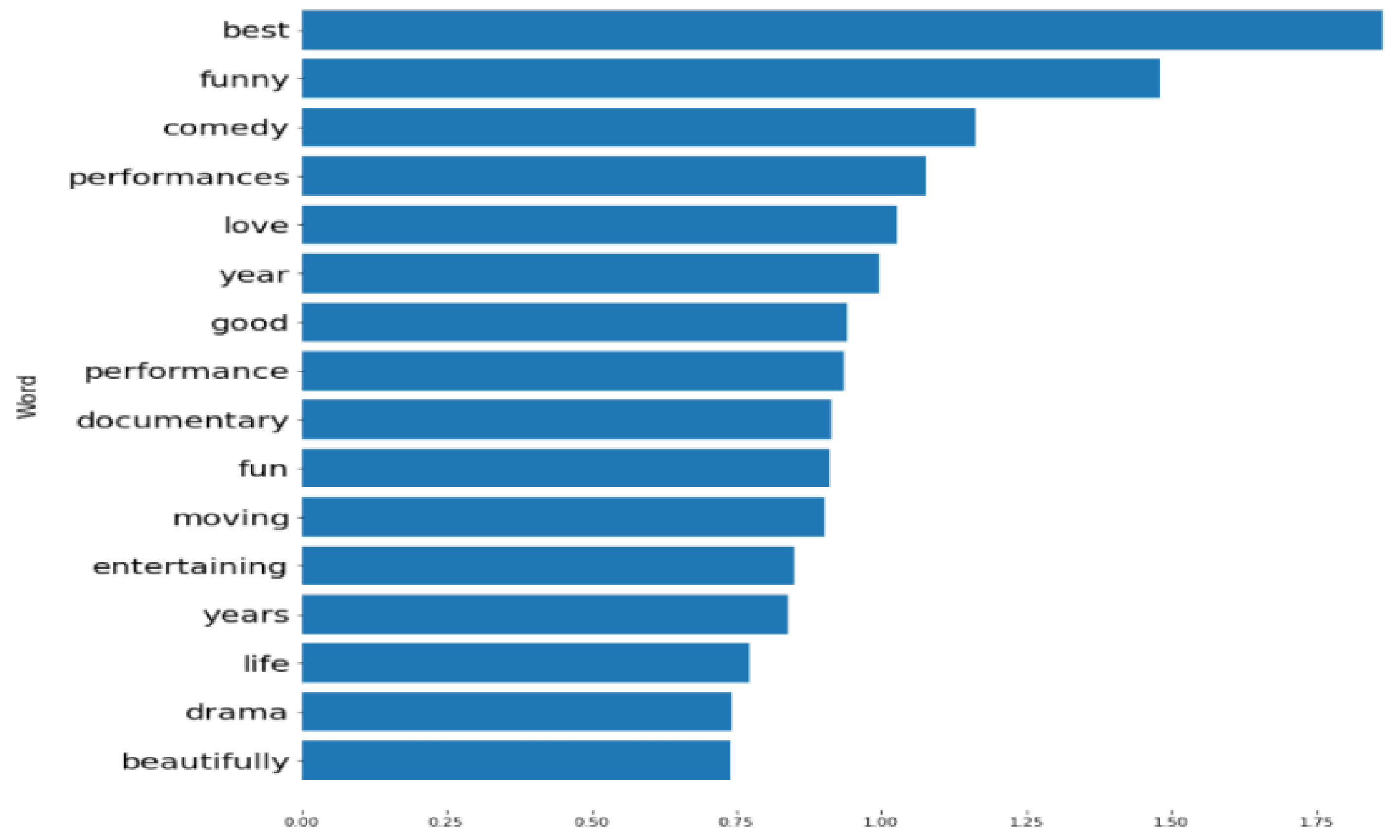

Figure 3: positive sentiment class

- The highest tf-idf score in somewhat negative sentiment class.
- The highest tf-idf score in somewhat positive sentiment class.

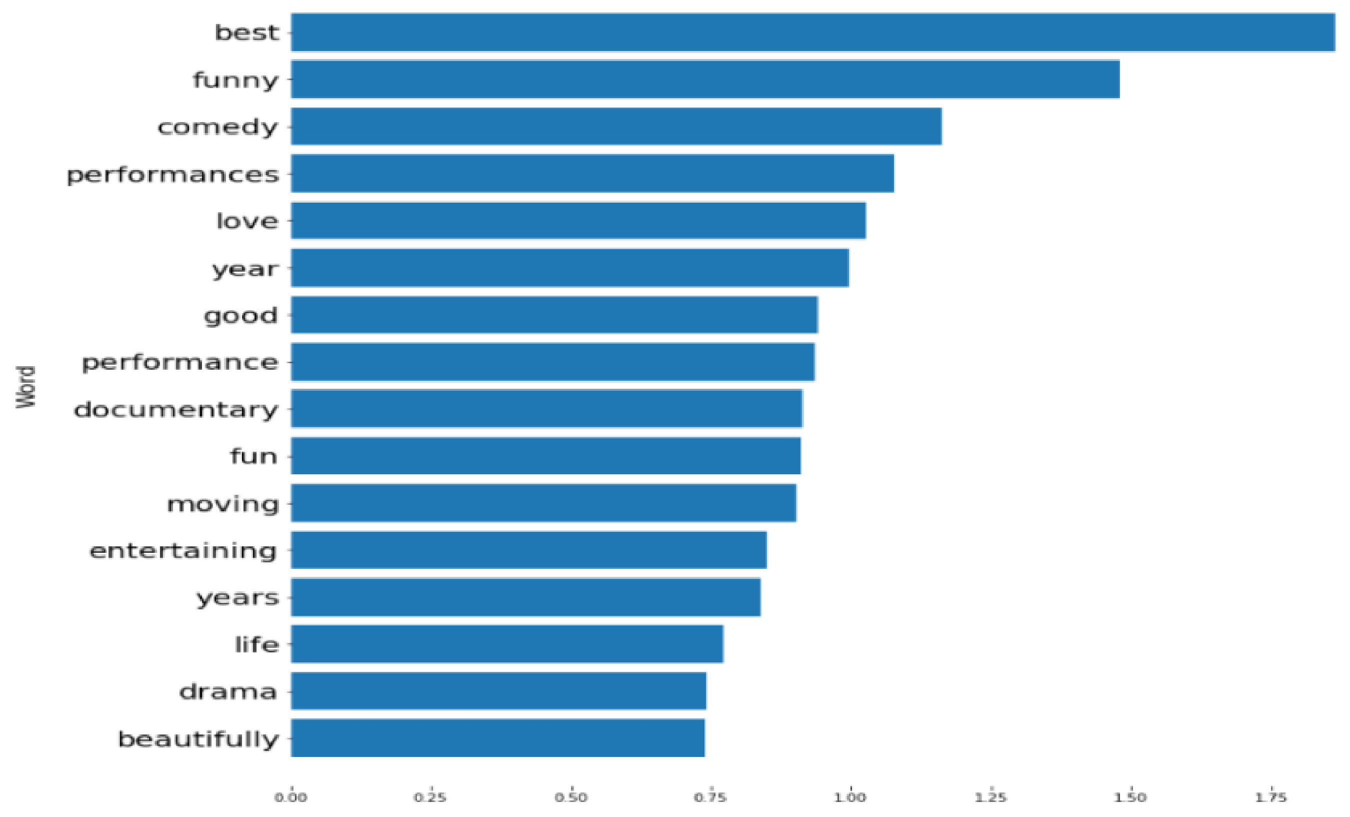

Figure 4: somewhat negative sentiment

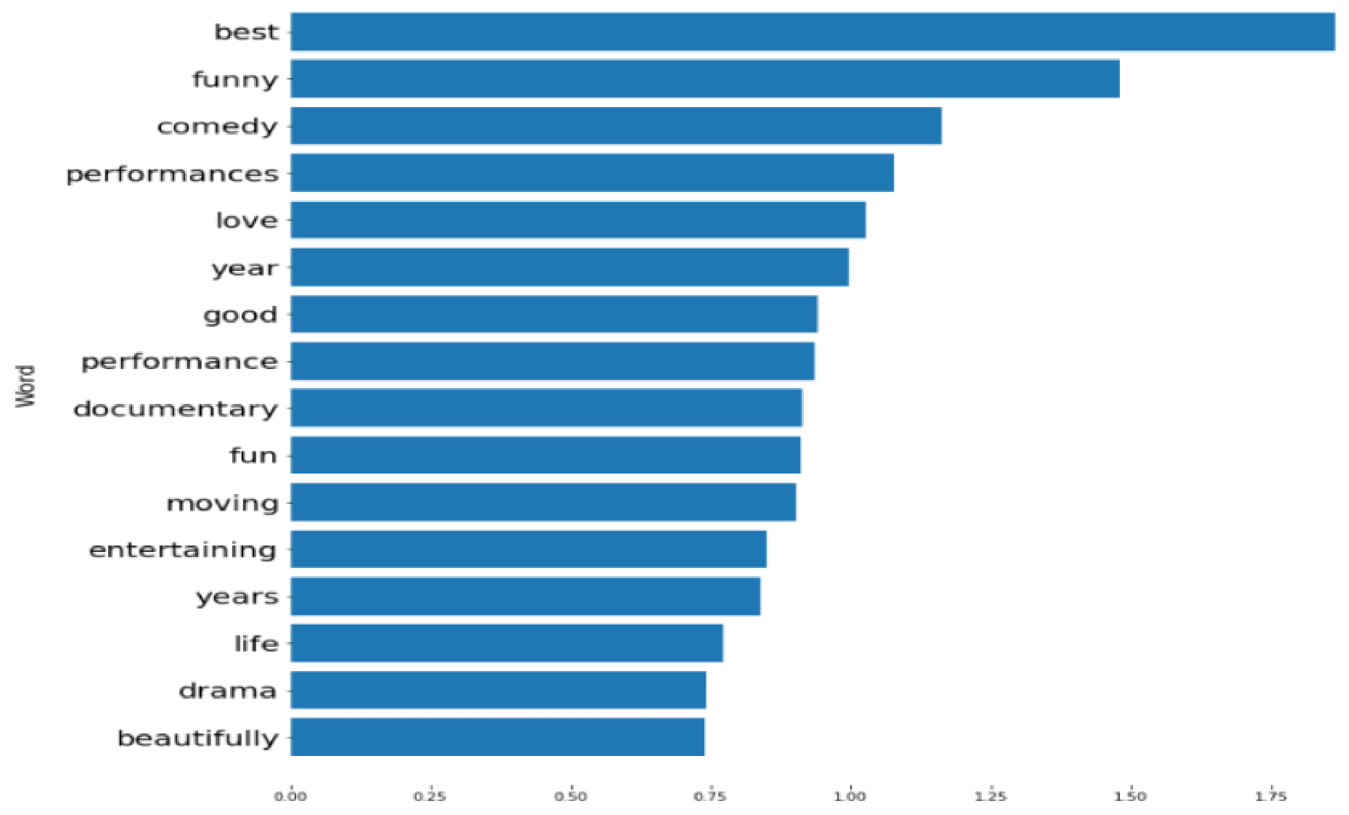

Figure 5: somewhat positive sentiment

Feature Engineering

Feature Extraction

- We can extract new features from existing data.
  - item_cnt_month:The monthly sales volume of each item in each store, sorted by month.
  - hist_min(max)_item_price:Figure out the maximum and minimum monthly sales of each item by month.
  - price_increace(decreace):How much each item's price changed from its (lowest/highest) historical price.
  - item_cnt_max,item_cnt_mean,item_cnt_std:Maximum, minimum, average, and median monthly sales of each item in each store.

Partition training set

- Training set
  - Use the first three months to generate features and implement functionality. The 3-27 blocks are used for training.
- Validation set
  - the five blocks 28-32 are used for verification.
    It is used to verify the accuracy of the model. The data is divided according to time because of its time characteristics.
- Test set
  - We want to predict for "date_block_num" 34 so our test set will be block 33 and our predictions should reflect block 34 values.
    In other words we use block 33 because we want to forecast values for block 34.

Mean Encoding

- Done after the train/validation split.
  - Find the average monthly sales volume by store number.
  - Find the average monthly sales volume by commodity number group.
  - Find the average monthly sales volume of each item in each store, grouped by store and item number.
  - Group by year, find the average annual sales.
  - Group by month, find the average monthly sales.
- Add meand encoding features to train set. Add meand encoding features to validation set.

Build Model

Linear Regression and Random Forest

- Linear Regression
  - Train rmse: 0.7347132326333324
    Validation rmse: 0.7755311093532987
- Random Forest
  - Train rmse: 0.6985868322226099
    Validation rmse: 0.776123635046122

XGBoost

- Train rmse: 0.697475453300762

  Validation rmse: 0.798117433161014
- XGBoost feature importance.

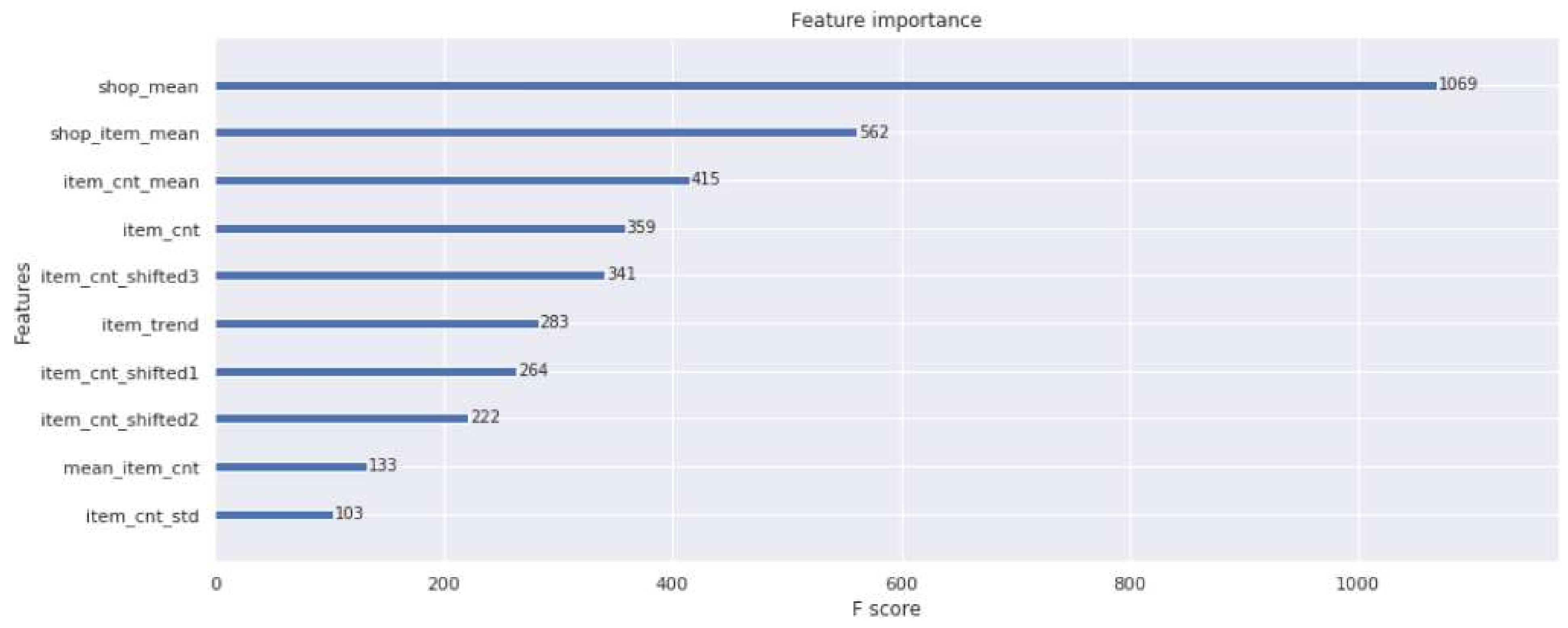

Figure 6: Feature importance

Create New Datasets

Table 4: Predictions from first level models

	random_forest	linear_regression	xgboost
0	0.98	0.85	0.44
1	0.06	0.06	0.10
2	0.85	1.79	0.50
3	0.00	0.06	0.10
4	0.06	0.06	0.10

Ensembling

- To combine the 1st level model predictions,to use a simple linear regression.
  This is the model that will combine the other ones to hopefully make an overall better prediction.
- Make predictions on test set using the 1st level models predictions as features.

Ensemble Diagram

- Look at the line chart below
  - Here is an image to help the understanding

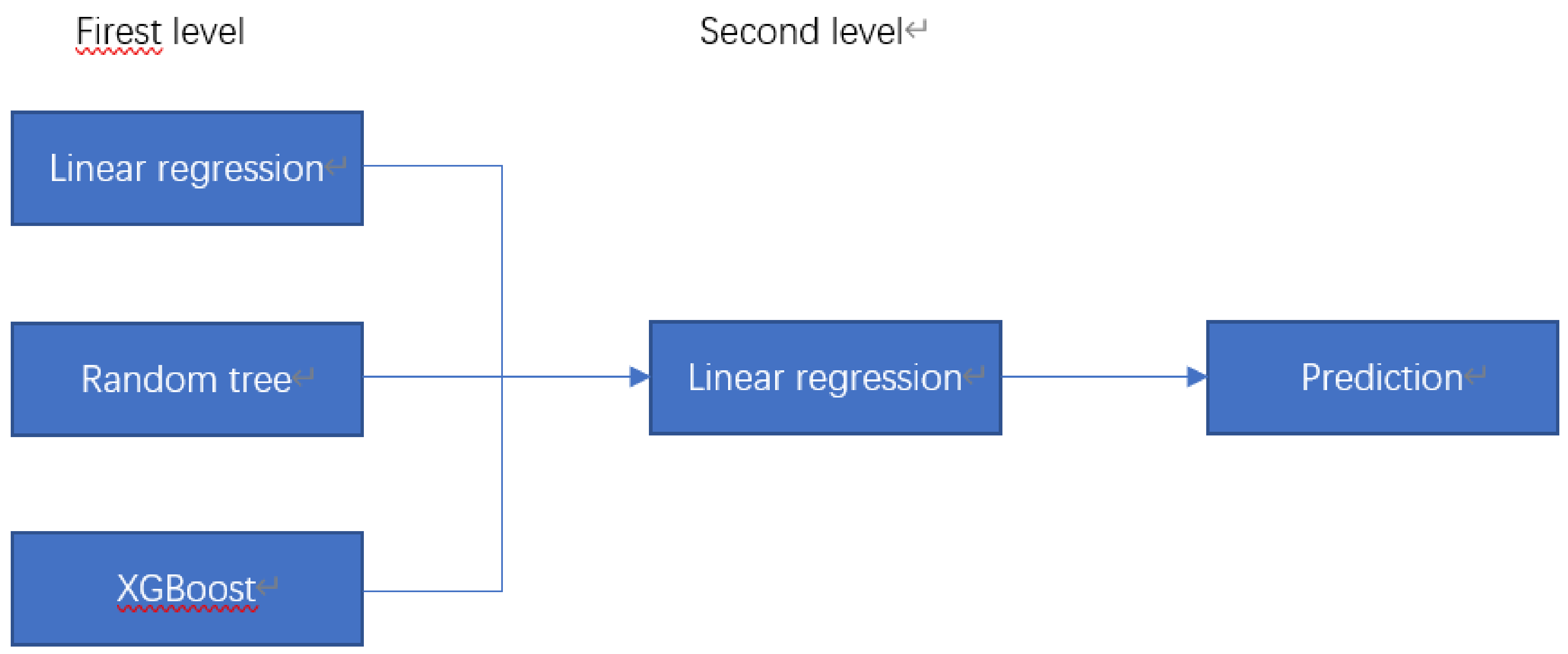

Figure 7: Ensemble diagram

Output Dataframe

Table 5: Output dataframe

	ID	item_cnt_month
0	0	0.85
1	1	0.08
2	2	1.29
3	3	0.06
4	4	0.08
5	5	0.96
6	6	1.25
7	7	0.21
8	8	1.99
9	9	0.06

Contact Information

Thank you for listening!

Siyu Chen

Xi'an Shiyou University

✉ 785987165@QQ.COM