Recommender system with Sentiment analysis

...

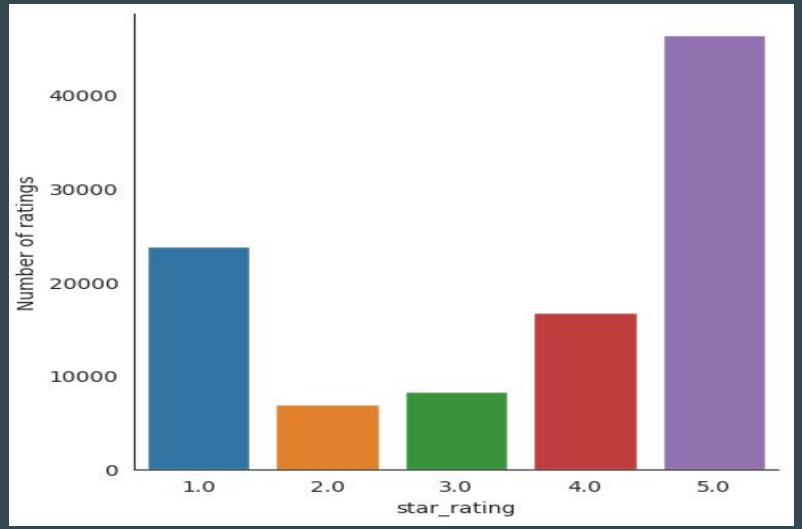
Frederico Lopes (up201904580)
José Miguel Maçães (up201806622)
Gabriel Martins (up201906072)

Main objective

- Create a recommender system
- Create 2 NLP models to do sentiment analysis
- Weight the sentiment into the recommender system
- Compare the initial recommender systems with the ones who also weight the sentiment of the review

Dataset - Amazon US reviews

- Number of total reviews: 101837
- Number of unique customers: 94099
- Number of unique products: 2995
- Customer with more reviews: 19



Recommender System

Data preparation

- Delete unnecessary columns
- Apply a threshold on the number of reviews per customer and product (minimum 3 reviews)
- 1063 reviews
- Number of unique customers: 161
- Number of unique products: 69

Recommender System

Model

- User-based collaborative filtering
- Train/test split originally 70/30, then eliminating from test customers who aren't in training (to solve cold start problem)
- Steps:
  - Applying adjusted cosine similarity
  - Obtaining similarity matrices
  - Predicting ratings

Recommender System

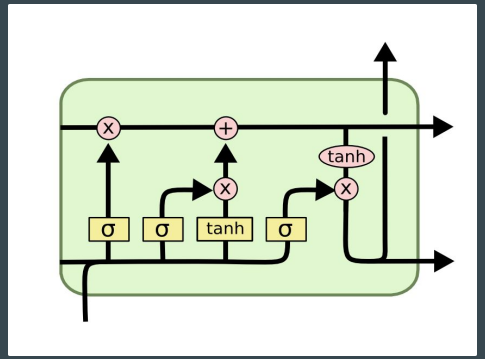
Evaluation

- RMSE: 2.86
- Precision@k
  - Precision@5: 0.042
  - Precision@10: 0.028

Sentiment analysis with LSTM (1/2)

Data preparation

- Delete unnecessary columns
- Changes labels from 1-5 stars to 0 or 1 (positive or negative)
- Create corpus with all reviews
- Remove punctuation, apply lowercase, remove stop words and apply stemming
- Create a list with all the vocabulary, sort by repetition and create a number for each word
- Pad the reviews
- Create training and testing sets



Sentiment analysis with LSTM (2/2)

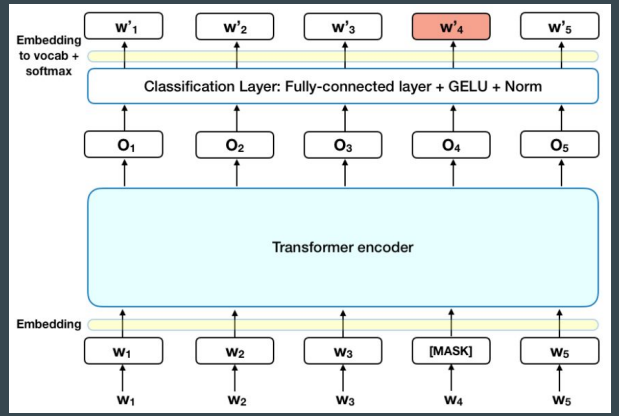
Model

- Define model with fully connected layers, LSTM layers and uses Sigmoid as activation for output layer
- Learning rate of 0.001, BCELoss and Adam optimizer used
- 90% accuracy in training data achieved in 5 epochs
- Able to predict with accuracy most inputs you give as normal text.
- 86% accuracy in test data after 20 epochs
- Accuracy almost doesn't increase after the first 5 epochs
- Model may not be complex enough to achieve a much higher accuracy

Sentiment analysis using BERT (1/2)

Data preparation

- Delete unnecessary columns
- Changes labels from 1-5 stars to 0 or 1 (positive or negative)
- Create corpus with all reviews
- Import pre-trained tokenizer that performs similar steps to the ones mentioned before
- Create custom dataset to load data



Sentiment analysis using BERT (2/2)

Model

- Import Bert pre-trained model
- Train model in our dataset
- Tried with stars prediction and sentiment analysis prediction
- Cross entropy loss, 1e-5 learning rate and Adam optimizer
- Similar accuracy for less epochs and less data
- Takes much longer to train
- 89% accuracy after only 1 epoch and 5000 lines of data (from the 101837)
- More data would probably result in more accuracy

Recommender system with Sentiment analysis

- Load the data containing product reviews and the recommendation and sentiment classification trained models.
- With the formula (W1*predicted rating of recommended product + W2*normalized sentiment score on scale of 1–5 of recommended product) we generate a new product racking score.
- Because we believe people put more thought when writing a review we gave W1 a value of 1 and W2 2.

Results

Only Recommender System

- RMSE: 2.86
- Precision@k
  - Precision@5: 0.042
  - Precision@10: 0.028

Recommender System with Sentiment Analysis

- Precision@k for lstm
  - Preicion@10: 0.098
- Precision@k for bert
  - Precision@10: 0.1

Discussion

- Difficult dataset with sparse connections
- Sentiment analysis still got a good result
- Recommender system did improve with the sentiment analysis but could be better
- Next steps would probably be change the recommender system to something more complex to achieve a higher accuracy