Sentiment Analysis on Amazon Product Review

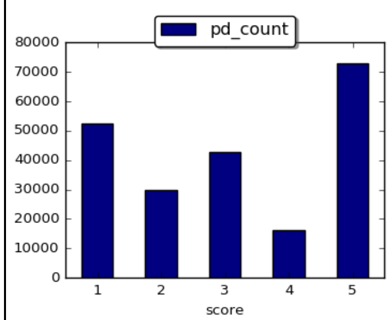
Rex Yung and Sergio Rozada Doval

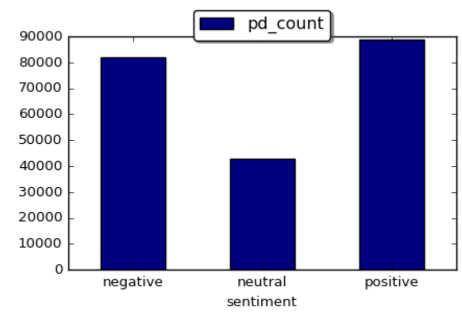
Description and Motivation of the Problem

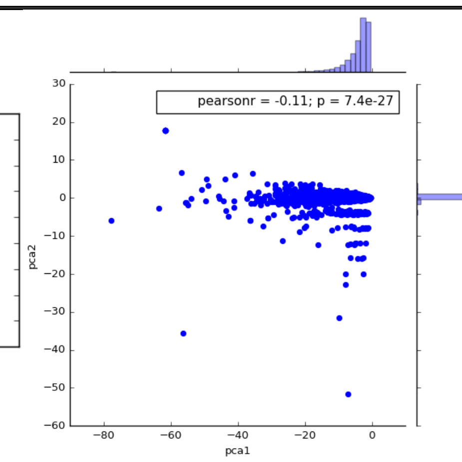
- Perform a score and sentiment analysis in a dataset consists of fine foods product reviews on Amazon.
- Aim to find a relation between the vocabulary used in the review and the degree of acceptance of the product.
- Inspect the effect of preprocessing strategies and training set size on the performance of machine learning algorithms in terms of metrics and running time

Initial Analysis and Basic Statistics

- Dataset: Fine Foods product reviews on Amazon from Kaggle
- The data set has around 570,000 rows. 2 feature columns are used: review summary and the actual review text.
- 2 label columns: Original score column – from 1 to 5, and sentiment column – positive (Score 4-5), neutral (Score 3), negative (Score 1-2)
- Original dataset has imbalanced number of rows for each class, we obtained a dataset with around 210,00 rows after undersampling some rows of rating 4 – 5 to balance the dataset.
- PCA analysis is performed to try to find any patterns in the data







Data Processing and Machine Learning Pipeline

Data (Text) Processing

- Using the pyspark.ml.feature package for processing text data
- Step 1: Tokenizer
- Step 2: Stop Words Remover
- Step 3: Feature Vectorization (TF-IDF) – transforming text data into numerical features
  - Step 3.1: Term Frequency (TF) Hashing
  - Step 3.2: Inverse Document Frequency (IDF) matrix

Machine Learning Pipeline

- 3 algorithms to train the model
  - Random Forest
  - Naïve Bayes
  - Logistic Regression
- Parameter Tuning (Grid Search)
- Used 20% of training set as validation set
- Validate models by accuracy as evaluation metric

Data Processing

Tokenizer → Stop Word Remover → TF-IDF

Parameters	Grid Search Values
# Features	[10, 100]
Min Doc Freq	[1, 10]

→ Feature Vectors

Model Training

Parameter Tuning

RF Parameters	Grid Search Values
Number of Trees	[10, 100]
Maximum Depth	[1, 10]

NB Parameters	Grid Search Values
Smoothing Constant	[0, 0.5, 1]

LR Parameters	Grid Search Values
Elastic Net Constant	[0, 0.5, 1]

→ Tuned Models

Model Evaluation

- Accuracy
- Precision
- Recall

Experimental Results

1. Score model trained with full dataset
2. Sentiment model trained with full dataset
3. Sentiment model trained with 1% and 10% size of the original data
4. Sentiment model trained with review summary

Score Model (Full Dataset)

Models	Train Accuracy	Test Accuracy	Running Time
Random Forest	0.4455	0.4234	58 min 41 sec
Naïve Bayes	0.5694	0.5419	7 min 56 sec
Logistic Regression	0.6875	0.6257	14 min 6 sec

Sentiment Model (Full Dataset)

Models	Train Accuracy	Test Accuracy	Running Time
Random Forest	0.6099	0.5990	64 min 52 sec
Naïve Bayes	0.6941	0.6819	8 min 44 sec
Logistic Regression	0.7738	0.7403	13 min 23 sec

Sentiment Model (1% of Original Data Size)

Models	Train Accuracy	Test Accuracy	Running Time
Logistic Regression	0.7410	0.5906	54.3 sec

Sentiment Model (10% of Original Data Size)

Models	Train Accuracy	Test Accuracy	Running Time
Logistic Regression	0.6762	0.6468	1 min 50 sec

Sentiment Model with Review Summary

Models	Train Accuracy	Test Accuracy	Running Time
Logistic Regression	0.6859	0.6588	1 min 44 sec

Analysis and Critical Evaluation of Results

- Surprisingly, random forest, the more complex model, performs the worst among the three models we trained. Random Forest also took the longest to train.
- In contrast, logistic regression, a relatively simpler model, performs the best and have the second best running time (13 mins), which is close to the fastest (8 min).
- The models performed better at predicting sentiment than predicting the rating.
- With the downsampled dataset (1% of original), we can see that there is a big difference between training set accuracy and testing accuracy, illustrating that there might be not sufficient data for the model to predict data it has never seen.
- While with the second downsampled dataset (10% of original), it provides results that are similar to the ones trained with full dataset, implying that we might only need 10% of the dataset to train a sufficiently accurate model with efficiency, as it only needs 2 minutes to train, instead of 13 minutes when training with full dataset.
- We observed not much of a difference in running time between the downsampled datasets (1% and 10%). It implies that distributed systems like Spark are more suitable for bigger datasets, as there is a minimum computation cost to pay, which should be the communication cost between clusters.
- It is also observed that when the model is trained with the review summary, it created a decent result compared with models trained with the actual review. The running time is also greatly reduced. Therefore it might be a good alternative if computation cost is more prioritized than prediction performance.