PR1: Movie Review Classification Report

Submission by: Surbhi Jain **Student Id:** 011428040
Rank: 1st **Accuracy:** 86.26%

Abstract

In current project, reviews for movies and corresponding sentiment labels were provided. The task is to train a KNN model to be able to predict the sentiment for an unseen test data. The given problem can be categorized under supervised classification setting with instance based learning. In current study, training data was used to train a KNN model and it was used to predict the sentiments for unseen data. After performing significant amount of data-mining and data analysis, Skip-gram word level and sentence embeddings, N-grams (1 to 4 grams), Term Frequency-Inverse Document Frequency (TF-IDF) were used as features to train the model. Several experiments were evaluated locally using 3-fold cross-validation, before making the submission. The accuracy of the best model on Test-data and Validation-data is found to be 86.26% and 90% respectively, which makes it the top submission on the Leaderboard.

Description

One of the most important aspect of model building is data mining and feature engineering. Following steps were performed in the sequential order to build the model:

1. Data Cleaning:
   Labels for each sample were converted to integer format. Regarding the reviews in text format, regex was used to remove html tags and special characters. Furthermore, stop-words were removed for the embedding based features. Stop-words list was obtained using "NLTK" library. Given the dimensionality of the data, stemming was also performed using Porter Stemmer.
2. Feature Extraction:
   Multiple features were extracted and evaluated for this project. The following features were extracted: (i) Word frequency, (ii) Term Frequency Inverse Document Frequency (TF-IDF), (iii) N-gram features from unigram to four grams were extracted, (iv) 300-dimensional Skip-gram word-embeddings and sentence level embeddings were used as extended features to carry the semantic information in the model. These embeddings were trained on the training data and also retrieved from pre-trained Google's News corpora.
3. Feature Engineering and Selection:
   Several experiments were performed to find the best feature combination. TF-IDF features were obtained with words present in less than 0.1% documents and those present in more than 98% documents were removed. After performing several experiments, one to four n-grams were chosen as best n-gram features. Furthermore, 7000 most relevant features were selected out of all the n-gram and TF-IDF features (~200,000 features) in addition to 300-dimensional word-embeddings. So, in total 7300 features were selected and used in the best model configuration.
4. Model Selection:
   KNN model comes up with several configurations and types of technique. In current study, brute force, KD-Tree and Ball-Tree algorithms were tested, however with right tuning, a basic L2-norm based similarity measure itself gave great performance and speed gains.

Tools and Libraries used: (1) TensorFlow (2) SkLearn (3) Gensim (4) Scipy (5) Numpy.

Experiments and Results

For fast iteration and effective turnout time, experiments were setup in the k-fold cross-validation setting, wherein given training data was split into 80-20% ratio, with former used for training and later for validation. The following experiments were performed and the results are posted on validation data:

Features	Main Parameters	Results
Word Frequency (WF)	K = 100	61%
TF-IDF	K = 100	70%
TF-IDF + Word Embedding (WE)	K = 100, D = 300	79%
TF-IDF + Word Embedding (WE) + Feature Selection (FS)	K = 100, D = 300, F= 7000	82%
TF-IDF + (WE) + FS + N-gram	K = 500, D = 300, F= 7000 + N: 1-3	85%
TF-IDF + (WE) + FS + More N-gram	K = 500, D = 300, F= 7000 + N: 1-4	87%
TF-IDF + (WE - No stop words) + FS + More N-gram	K = 250, D = 300, F= 7000 + N: 1-4	89%

K: Number of nearest neighbors, D: Embedding Size, F: Selected Features, N: N-grams

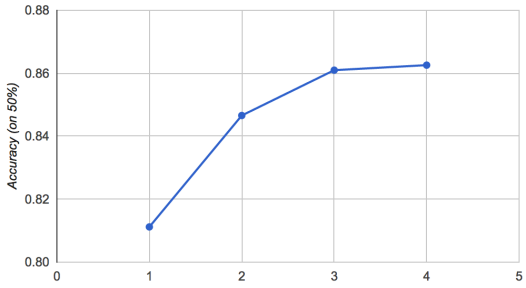
The model performed well on the test data as well and the best setup acquired 1st rank in the leaderboard. Following are the submission details:

Leaderboard:

Rank	Accuracy (on 50%)	User ID	Submission Count
1	0.8626	11428040	4
2	0.8594	11815440	11
3	0.8583	11430939	39

Accuracy (on 50%)	File	Submission Time
0.8626	test_predictions_submission.dat	Oct 10, 3:08 AM
0.8610	test.dat	Oct 9, 11:46 PM
0.8466	test.dat	Oct 9, 1:01 AM
0.8112	test.dat	Oct 9, 12:59 AM

Personal submissions:



Conclusions, Findings and Learnings

KNN is an extremely powerful instance based learning and it can be used for text classification and sentiment prediction problems. Word-embeddings combined with TF-IDF with right feature selection can lead to significant performance improvements. Data Mining is the key. With this project, I got good hands on how to work with textual data and how to effectively use them in applied setting.