Sentiment Analysis of IMDB dataset

by
Muniba Shaikh
CS 777 – Big Data Analytics

Data Set Description:

- IMDB dataset is obtained from Kaggle URL: https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews?select=IMDB+Dataset.csv
- It has 50K highly polar movie reviews.
- Each row in the data has a review and a sentiment (positive and negative) about a movie.

Classification Problem:

- My goal is to build a classifier that can predict whether a review about a movie is positive or negative based on the review text only.

Features:

No	Features	Feature Values and Description	Feature Type
1.	**review**	Text review about a movie	Text
2.	**sentiment**	Positive, negative	Categorical
3.	**label**	1: if the review is positive 0: if the review is negative	Class Label

```
+--------------------+---------+-----+
|              review|sentiment|label|
+--------------------+---------+-----+
|One of the other ...| positive|    1|
|A wonderful littl...| positive|    1|
|I thought this wa...| positive|    1|
|Basically there's...| negative|    0|
|Petter Mattei's "...| positive|    1|
|Probably my all-t...| positive|    1|
|I sure would like...| positive|    1|
|This show was an ...| negative|    0|
|Encouraged by the...| negative|    0|
|If you like origi...| positive|    1|
+--------------------+---------+-----+
```

Split the dataset into training and test set:

- IMDB dataset is split into train and test set using 70/30 split size.

Training Data

label	count
1	17450
0	17543

Test Data

label	count
1	7550
0	7457

Data Preprocessing:

is done on the training set using pipeline functionality of MlLib Spark's machine learning (ML) library

Tokenization:

- All non-letter characters are removed from the review text
- Review text is converted to lower case.
- Review is tokenized into individual words.

Removing stop words:

- Stop words such as *a, the, is, are, etc.* are removed from the review text because they appear frequently and don't carry as much meaning.
- Some stop words such as *"br", 'm', 've', 're', 'll', 'd'* , are determined through eye-balling and removed from the review text

```
|[turkish, bath, sequence, film, noir, located, new, york, 50, must, hint, something, something, curiously, previous, comments, one,
pointed, seems, essential, understanding, movie, turkish, baths, sequence, back, street, night, entrance, sleazy, sauna, scalise, wrapped,
sheet, getting, thighs, massaged, steve, masseur, young, rough, boxer, beefcake, type, another, guy, bodyguard, finishes, dressing, dixon,
obviously, hates, sees, gets, rough, right, away, know, reputation, roughing, suspects, good, cop, getting, control, easy, hates, much,
hates, part, inherited, father, dark, side, lead, right, end, sidewalk, gutter, dark, side, lurked, within, closet, remember, whenever,
dixon, meets, scalise, 3, times, guy, lying, bed, men, around, company, irony, girls, poster, pinned, wall, near, bed, scalise, acts,
```

Bag of Words

A vocabulary of words is extracted from reviews collections and the top 5000 words ordered by their term frequency across the corpus are selected using CountVectorizerModel.

Top 10 vocabulary words
movie
film
one
like
good
time
even
story
really
see

Feature vectors

Vocabulary of 5000 words are then used to vectorize the review text into feature vectors using CountVectorizerModel.

IDF:

Then IDF Model is applied to feature vectors to down-weight features which appear frequently in the reviews.

```
+---------+-----+--------------------+
|sentiment|label|            features|
+---------+-----+--------------------+
| positive|    1|(5000,[0,1,2,3,4,...|
| negative|    0|(5000,[0,1,2,3,6,...|
| negative|    0|(5000,[0,3,4,7,11...|
| negative|    0|(5000,[1,3,5,6,17...|
| negative|    0|(5000,[1,2,9,11,1...|
| negative|    0|(5000,[0,1,3,7,15...|
| positive|    1|(5000,[0,1,3,7,8,...|
| positive|    1|(5000,[1,2,3,5,7,...|
| negative|    0|(5000,[0,8,16,18,...|
| negative|    0|(5000,[1,3,10,25,...|
+---------+-----+--------------------+
```

Chi-Squared feature selection:

ChiSqSelector uses the Chi-Squared test of independence to decide which features to choose. It operates on labeled data with categorical features. Top 500 features are determined by Chi-Squared feature selection method to be used in the classification of reviews. Now the data is ready for the applying classifiers and predicting sentiment labels for reviews.

Classifiers used on the dataset:

- *SVM*
- *Logistic Regression*

Support Vector Machine

Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	6467	828
Actual Positive	990	6722

Computation Time

Operation	Time (s)
Model Training	67.907753
Testing Model	0.26406
Performance Metrics	24.151414
Total Time	92.323227

Performance Metrics

Analytics	Value
Accuracy	0.878857
Precision (Class 0)	0.867238
Recall (Class 0)	0.886498
Precision (Class 1)	0.890331
Recall (Class 1)	0.871629
F1-Measure	0.880881

Logistic Regression

Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	6491	844
Actual Positive	966	6706

Computation Time

Operation	Time (s)
Model Training	75.9268169
Testing Model	0.364875
Performance Metrics	24.945547
Total Time	101.237239

Performance Metrics

Analytics	Value
Accuracy	0.8793896
Precision (Class 0)	0.870457
Recall (Class 0)	0.884935
Precision (Class 1)	0.888212
Recall (Class 1)	0.874088
F1-Measure	0.881093

Comparison of performance measures between classifiers

Model	Recall (Class 0)	Recall (Class 1)	Accuracy%	Computation Time(s)
SVM	0.886498	0.871629	87.8	92.323227
Logistic Regression	0.884935	0.874088	87.9	101.237239

- Accuracy of prediction and TPR is the same for both the classifiers. However, SVM is faster than Logistic Regression.
- Since the dataset is balanced meaning it has almost equal number of positive and negative reviews to train the model, and its equally important to predict positive as well as negative reviews accurately. Therefore, the best metric to measure performance of the classifiers is accuracy.

Predicting sentiment on new data

```
Prediction using Logistic Regression model:
+-----------------------------------------------------------+----------+
|review                                                     |prediction|
+-----------------------------------------------------------+----------+
|This movie was horrible, plot was boring, acting was okay.|0.0       |
|The film really sucked. I want my money back               |0.0       |
|What a beautiful movie. Great plot, great acting.          |1.0       |
|Harry Potter was a good movie.                             |1.0       |
+-----------------------------------------------------------+----------+
```

```
Prediction using SVM model:
+-----------------------------------------------------------+----------+
|review                                                     |prediction|
+-----------------------------------------------------------+----------+
|This movie was horrible, plot was boring, acting was okay.|0.0       |
|The film really sucked. I want my money back               |0.0       |
|What a beautiful movie. Great plot, great acting.          |1.0       |
|Harry Potter was a good movie.                             |1.0       |
+-----------------------------------------------------------+----------+
```