Differentiating Language about Movie Facts from Movie Descriptions using Natural Language Processing

Anna Funsten

Can we create a machine learning model that can accurately differentiate language about movie facts from movie descriptions?

r/MovieDetails

97

Posted by u/ofek162 2 days ago

? Trivia In The Godfather (1972), Actor Al Martino has used connections from real life organized crime boss Russell Bufalino in order to secure the part of Johnny Fontane. The same thing is shown in the film when Johnny Fontane asks Don Corleone to help him get a movie role.

https://preview.redd.it/gay6jm733ew61.png?width=445&format=png&auto=webp&s=96c12add6973fc4c13b86a57f4243ccbc6ff4f0a

14 Comments Share Save ...

r/moviefinder

1

Posted by u/Hammie_Da_Wizard 2 days ago

Pls help me remember my childhood

I remember an animated movie about forest animals. The main character isn't very liked so he tries different ways to make friends. I remember a scene where all the animals are dancing on a lake and the main character uses a plan that ends up scaring them away. I know this one's tough but pls help me.

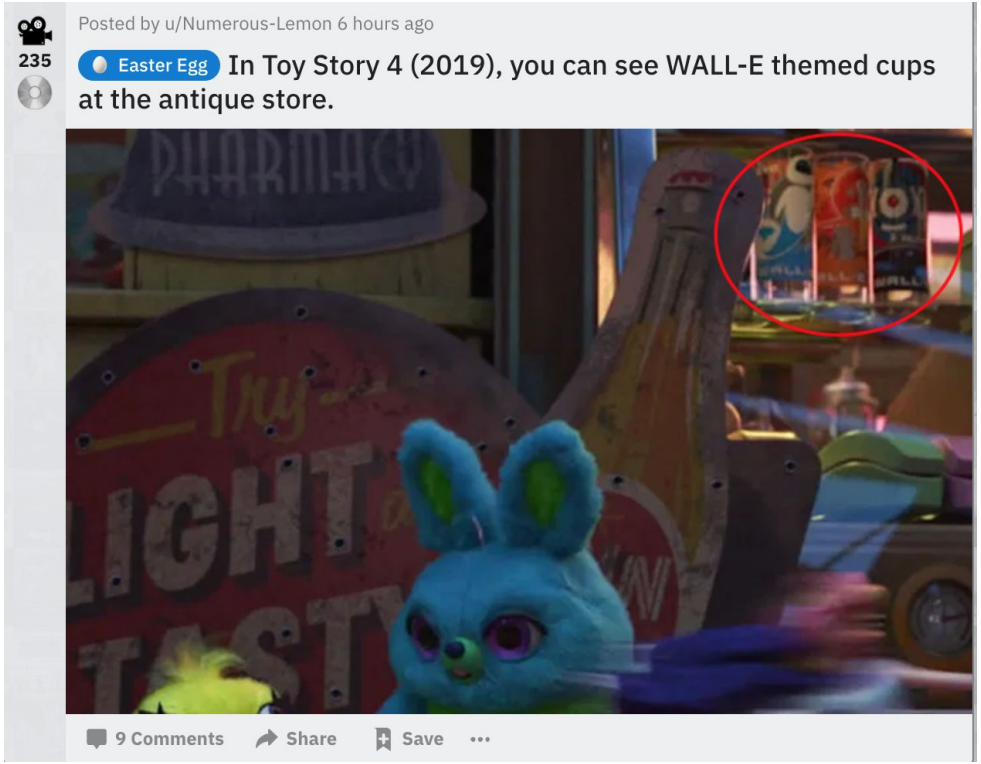
1 Comment Share Save •••

Data Collection

1) Used Pushshift’s API to collect submissions
2) Obtain 1000 post each from each subreddit
3) Made two dataframe for each subreddit 1000 post each
4) Focused on selftext and title sections as columns

Data Cleaning

1) Dropped spam posts
2) combined selftext and title sections into one column
3) Concated my 2 DataFrames into 1 DataFrame
4) Final DataFrame: 1982 rows and no nan values



Modeling

X = text from posts (title + selftext)

y = subreddit (0 = Movie Details, 1 = Movie Finder)

X_train, X_test, y_train, y_test = train_test_split(X, y, stratify = y, random_state = 22)

Null Model

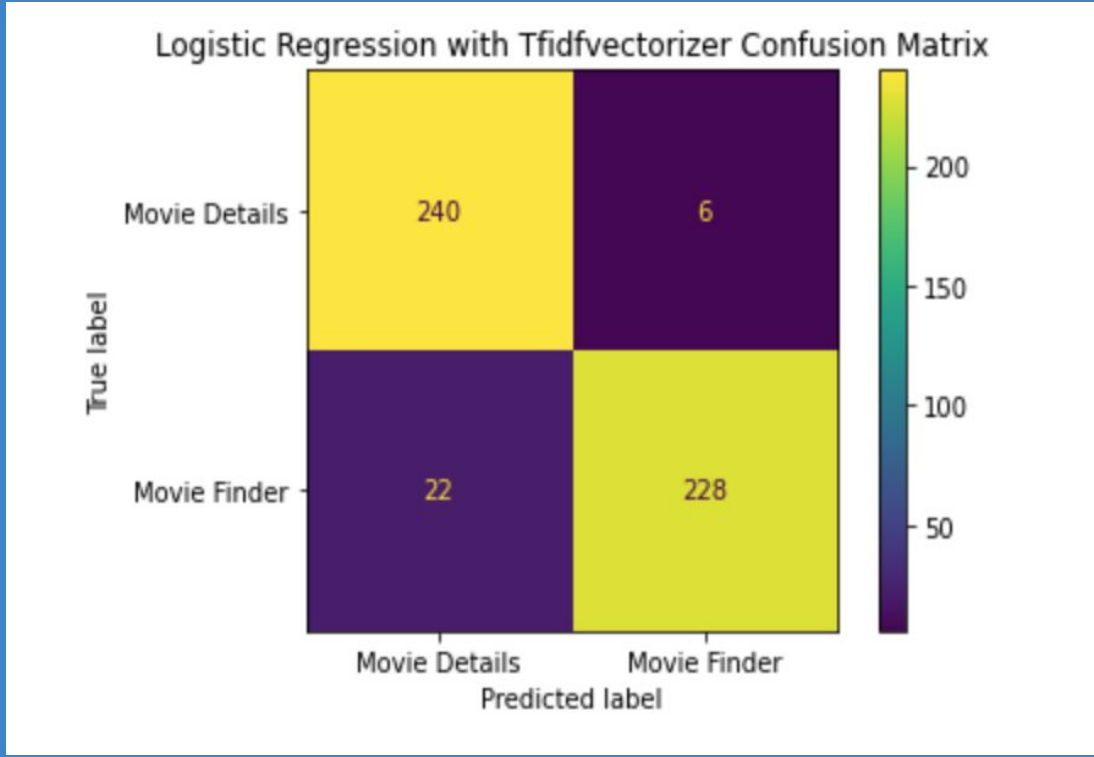
Metric: Accuracy

50.5% chance of guessing Movie Finder category

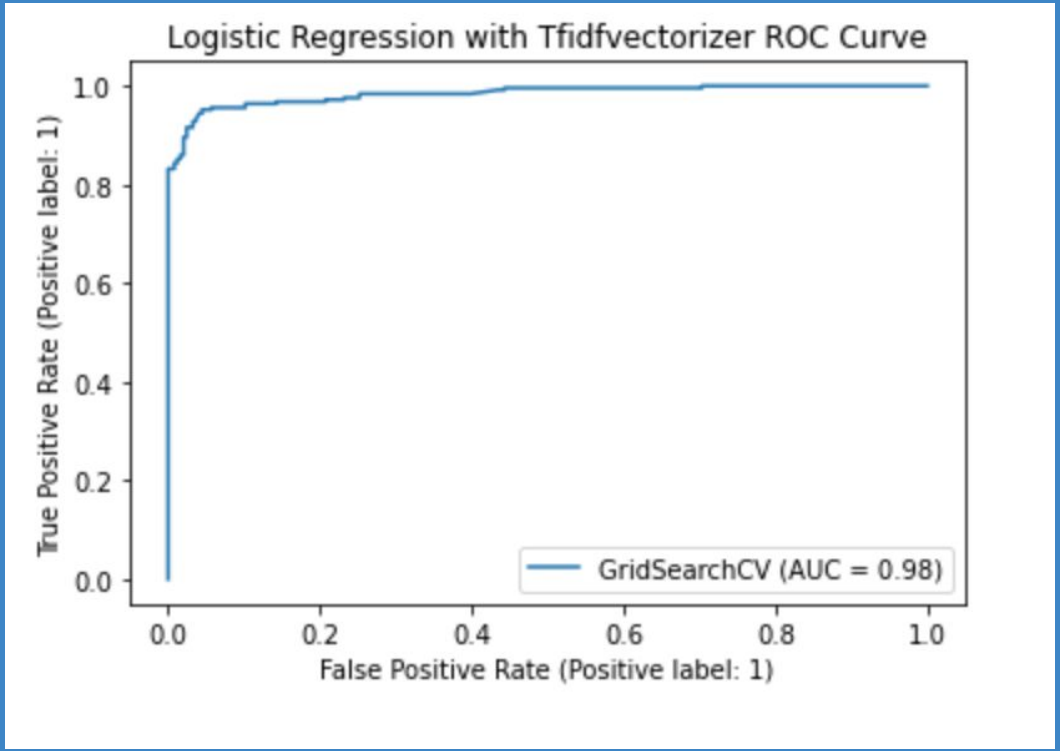
Model	Train Score	Test Score
Logistic Regression with CountVectorizer	0.989	0.941
Logistic Regression with TfidfVectorizer	**0.965**	**0.944**
Naive Bayes with CountVectorizer	0.921	0.919
Naive Bayes with TfidfVectorizer	0.933	0.911
Decision Tree with CountVectorizer	0.954	0.907
Decision Tree with TfidfVectorizer	0.976	0.903
Random Forest with TfidfVectorizer	0.956	0.931
KNeighbors Classifier with TfidfVectorizer	0.883	0.877

Best Model: Logistic Regression with TfidfVectorizer

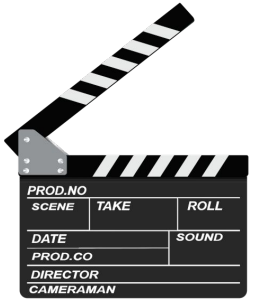
C: 0.001
Max Features: 800
Ngram range: (1, 3)
Accuracy: .944



True Positive Rate: 0.912
Type I Error: 0.024
Type II Error: 0.088



Top Features for Predictions



Movie Finder
(largest coefficients)

1) remember
2) movie
3) help
4) watch
5) looking

Movie Details
(lowest coefficients)

1) 2021
2) 2019
3) director
4) reference
5) 2020

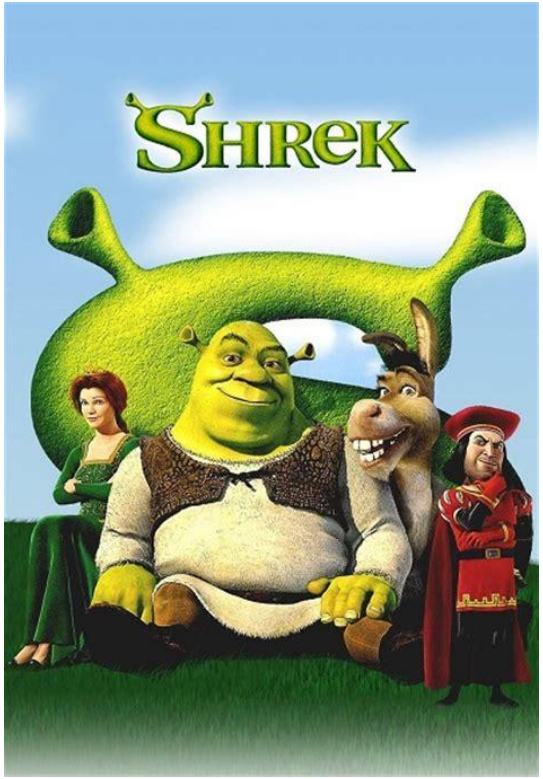
Testing the Model with New Data!

Input: 'There Will Be Blood Disrupted Shooting for No Country for Old Men'
Output: Movie Details!!

Input: '"Does anyone know the movie with the green ogre"'
Output: Movie Finder!!

Recommendation & Conclusion

1) Logistic Regression best model
    a) Accuracy score of 0.944
    b) High true positive rate
    c) Low type I and type II error
2) Able to work with real examples outside of model data!
3) Limitations:
    a) Unable to use lemmatized words
    b) Movie date information in model




The End