MOVIE SUGGESTIONS VS TV SUGGESTIONS

AGENDA

- Introduction
- Problem Statement
- Data Collection
- Modeling
- Confusion Matrix
- Conclusion and Recommendation

INTRO


STRANGER THINGS 2
A NETFLIX ORIGINAL SERIES


jtbc 금토드라마
이태원 클라쓰
ITAEWON CLASS
2020.1.31 금 PM 10:50 첫방송
박서준 / 김다미


ANSEL ELGORT
KEVIN SPACEY
LILY JAMES
EIZA GONZALEZ
with JON HAMM
and JAMIE FOXX
Written and Directed by
EDGAR WRIGHT
BABY DRIVER
ALL YOU NEED IS ONE KILLER TRACK.
SONY


1917
TIME IS THE ENEMY

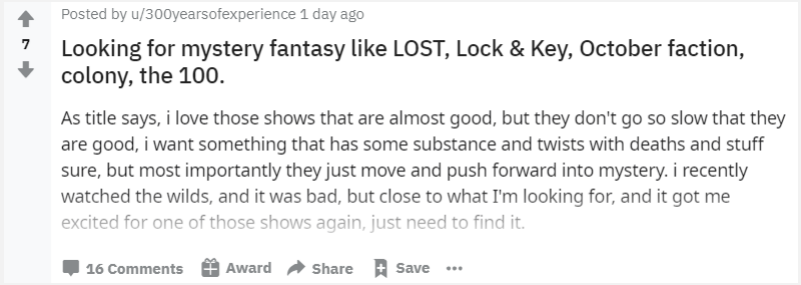
PROBLEM STATEMENT

As a part of data science team at film and television production company, I was tasked with creating a model that will be able to take a reddit post and classify which posts were talking about movies and which posts were talking about TV show to find current trends so the company can decide to make movies or television show according to consumer demand.

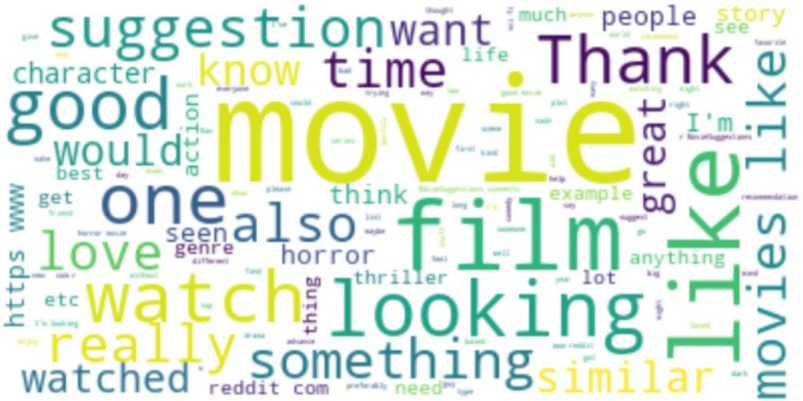
DATA COLLECTION

- r/MovieSuggestions
- 860 posts

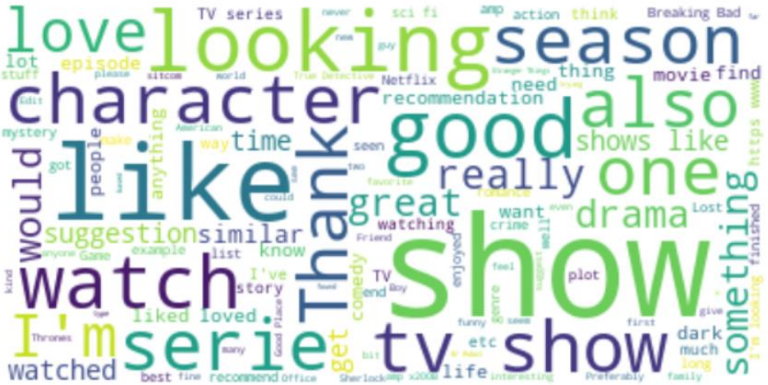
- r/TelevisionSuggestions
- 996 posts



Movie Suggestions



Television Suggestions

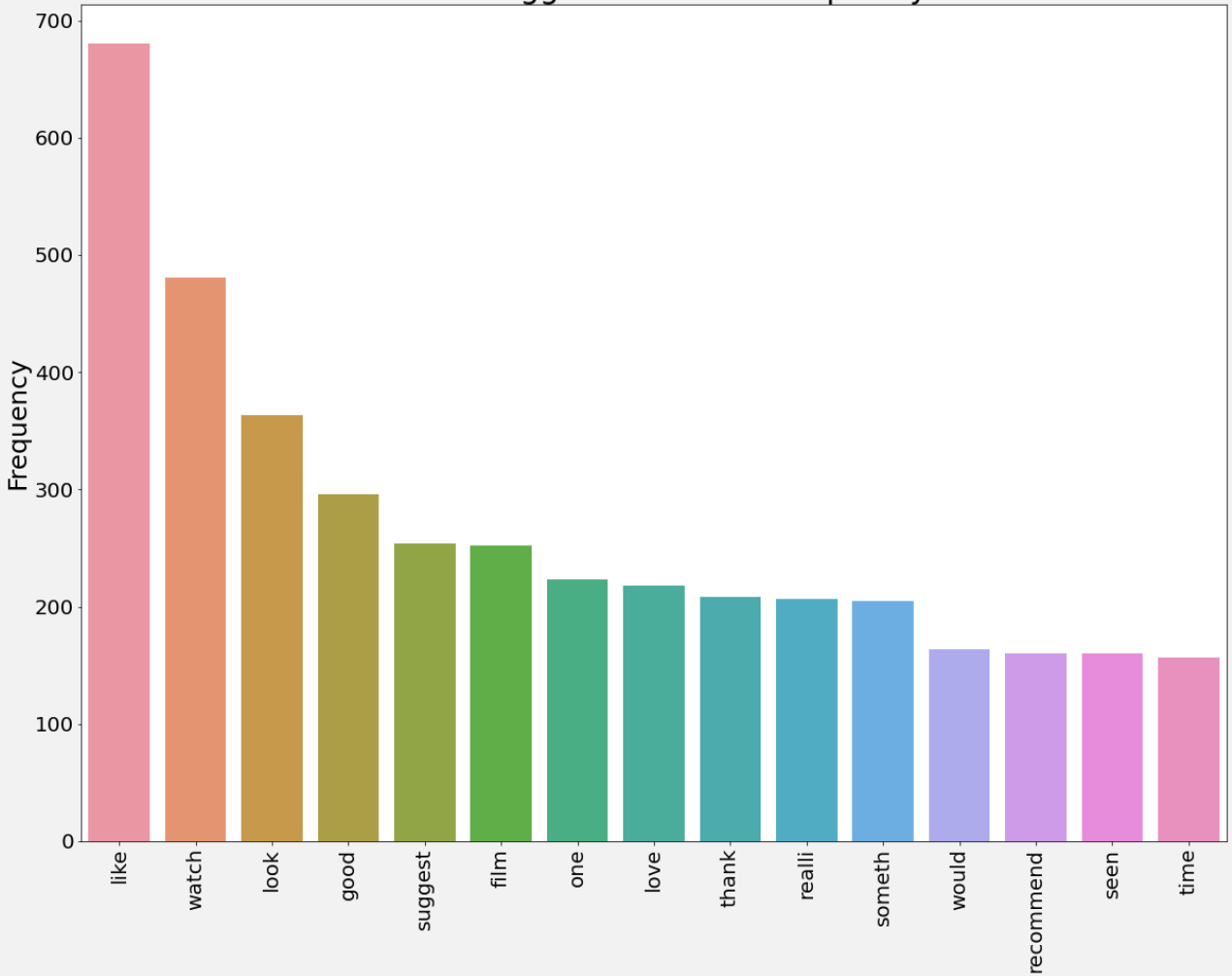


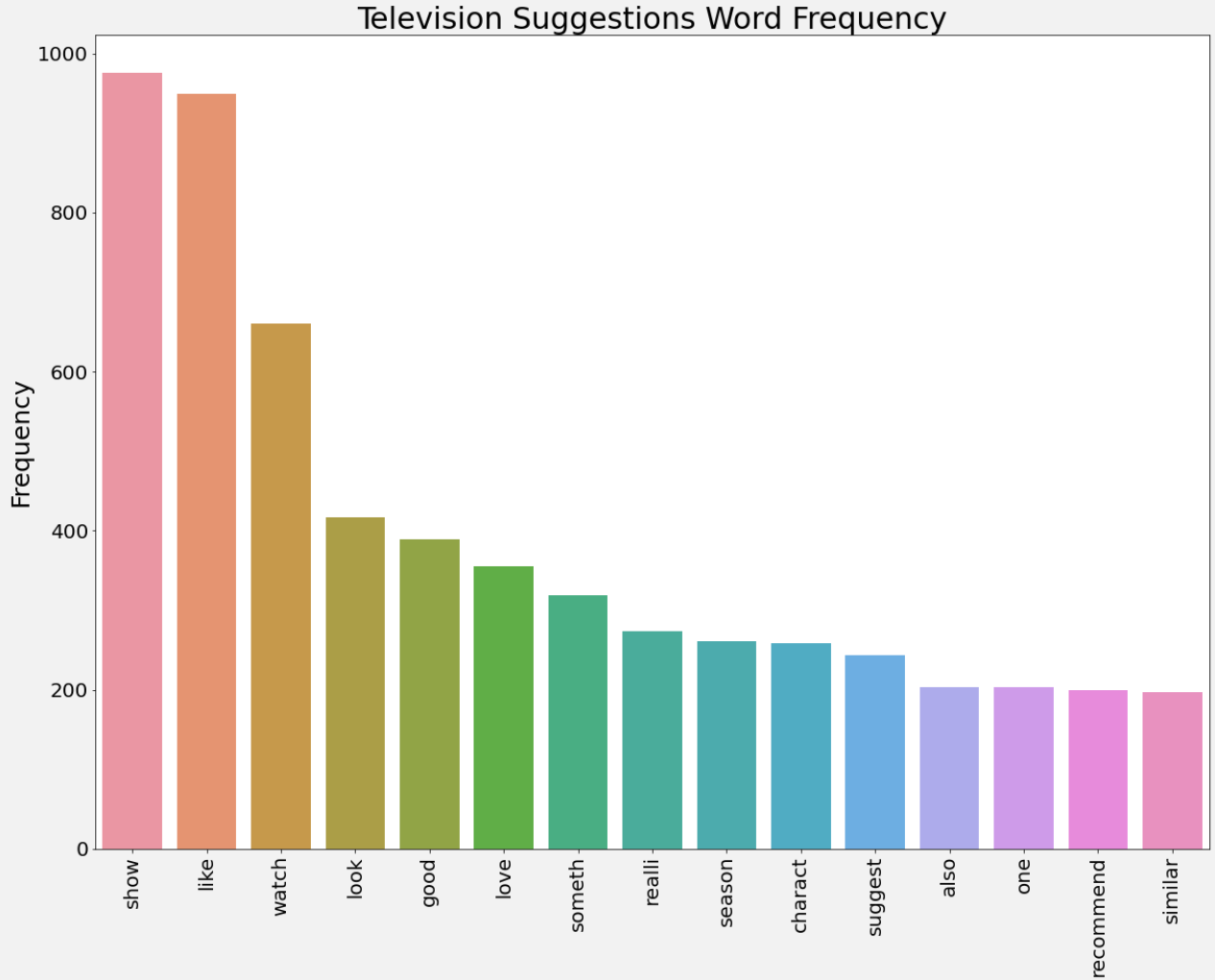
DATA CLEANING

- Remove HTML
- Remove non-letters
- Remove stop words
- Stemming word

```
In [530]: clean_text(final_df['text'][0])

Out[530]: 'top februari upi taken comment indic great januari ment ad round thread end januari vote count next month climb climb onto top
          favourit harakiri bruge minari place beyond pine promis young woman solari fall fell top amadeu art self defens brigsbi bear bl
          ood tumbbad wind river name gain biggest gain la hain john wick readi top movement within top rank parasit upon time hollywood
          hereditari midsommar quiet place knive lighthous isl dog joker game night thought top fell climb anyth els'
```


Movie Suggestions Word Frequency
Frequency
0
100
200
300
400
500
600
700
like
watch
look
good
suggest
film
one
love
thank
realli
someth
would
recommend
seen
time


Television Suggestions Word Frequency
Frequency
1000
800
600
400
200
0
show
like
watch
look
good
love
someth
realli
season
charact
suggest
also
one
recommend
similar

Movie Suggestions

Television Suggestions

BASELINE MODEL

```
In [219]: X = final_df['text']
          y = final_df['y']
          X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 42, stratify = y)
```

```
In [223]: # Baseline accuracy
          y_train.value_counts(normalize
```

```
Out[223]: 0    0.536638
          1    0.463362
          Name: y, dtype: float64
```

TEXT FEATURE EXTRACTION

```
In [225]: cvec = CountVectorizer()
          cvec.fit(X_train)
          X_train = cvec.transform(X_train)
          X_test = cvec.transform(X_test)
```

```
In [221]: X_train.shape
Out[221]: (1392,)

In [222]: X_test.shape
Out[222]: (464,)
```

```
In [337]: X_train.shape
Out[337]: (1392, 6494)

In [338]: X_test.shape
Out[338]: (464, 6494)
```

MODELING

Model	Train Accuracy	Validation Accuracy	Test Accuracy
Logistic Regression	99%	82%	82%
Multinomial Naïve Bayes	96%	85%	83%
Support Vector Machine	96%	83%	85%

GRID SEARCH

Model	Train Accuracy	Best Score	Test Accuracy
Logistic Regression	96%	83%	83%
Multinomial Naïve Bayes	94%	85%	85%
Support Vector Machine	95%	85%	83%

BEST PREDICTION MODEL

In [339]:
```
X = final_df['text']
y = final_df['y']
X_train, X_test, y_train, y_test = train_test_split(X, y,random_state = 42, stratify = y)
```

In [340]:
```
cvec = CountVectorizer(max_df = 0.9, max_features = 3000, ngram_range = (1,2))
cvec.fit(X_train, y_train)
X_train = cvec.transform(X_train)
X_test = cvec.transform(X_test)
```

In [341]:
```
nb = MultinomialNB(alpha = 1)
nb.fit(X_train, y_train)
print(f'Train Accuracy Rate = {nb.score(X_train, y_train)} \nTest Accuracy Rate  = {nb.score(X_test, y_test)}')
```

```
Train Accuracy Rate = 0.9425287356321839
Test Accuracy Rate  = 0.8577586206896551
```

In [337]:
```
X_train.shape
```

Out[337]: (1392, 6494)

In [338]:
```
X_test.shape
```

Out[338]: (464, 6494)

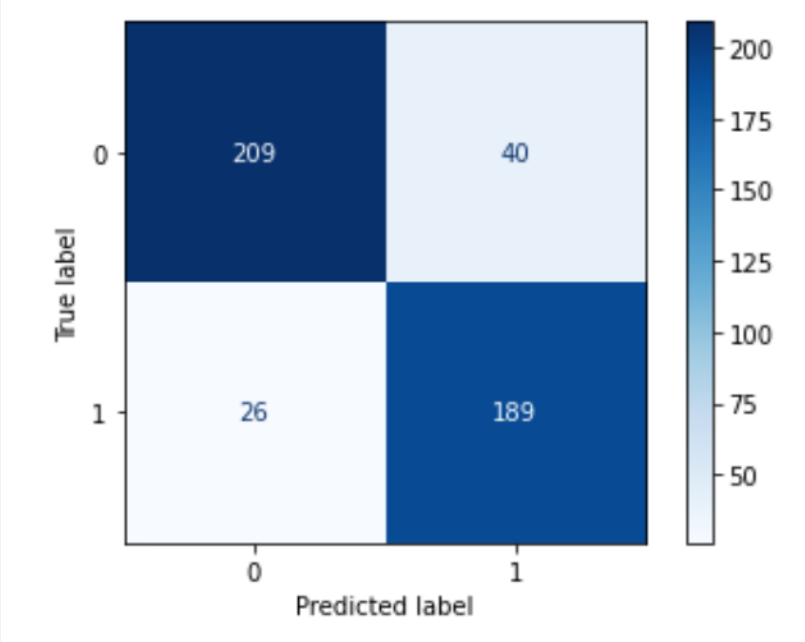
In [342]:
```
X_train.shape
```

Out[342]: (1392, 3000)

In [343]:
```
X_test.shape
```

Out[343]: (464, 3000)

CONFUSION MATRIX



True Positive = 189
True Negative = 209
False Positive: = 40
False Negative = 26

Accuracy = 86%
Misclassification Rate = 14%
Sensitivity = 88%
Specificity = 84%
Precision = 83%

CONCLUSION AND RECOMMENDATION

- Based on train and test accuracy rate multinomial naïve bayes is the best model among the three models.
- This model has accuracy rate at 86%
- Sensitivity or true positive rate at 88%
- Specificity or true negative rate at 84%
- Try another model like Random forest, Adaboost classifier
- Try another feature extraction like TF-IDF or Hashing vectorizer.