# REDDIT POST CLASSIFICATION WITH NLP

Deepshika Sharma

# PROBLEM STATEMENT

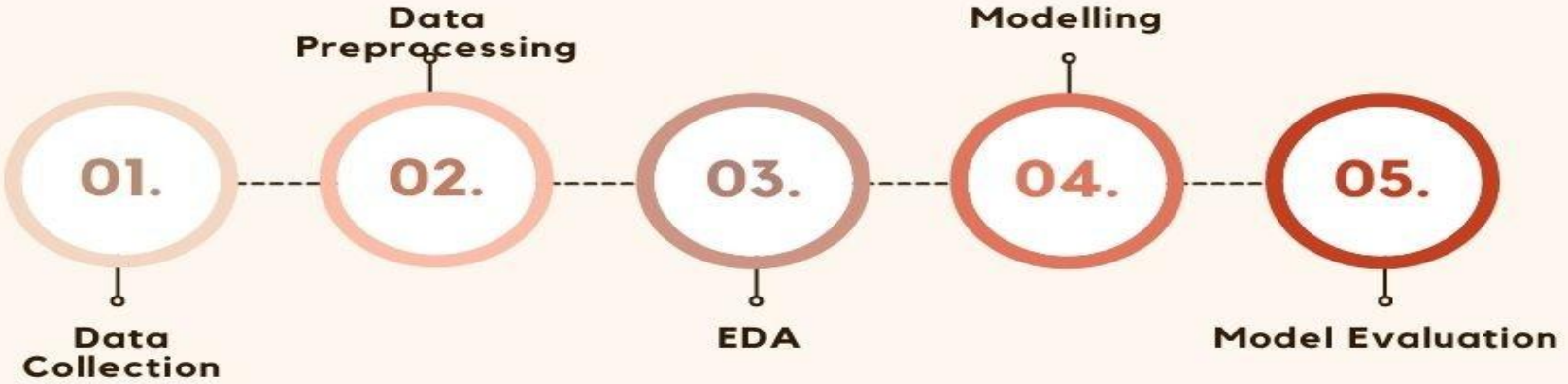
Company XYZ is planning to create a platform for streaming movies. The company wants to know which movies to stream or not depending on the box office posts.

The company would like to know if a particular post/comment is from the box office subreddit or not so that they can later analyze it further to understand which posts are positive about a movie and which are not.

# What do we know so far?

- Data is from reddit
- Subreddits are : movies and boxoffice

# PROCESS


01.
Data Collection
02.
Data Preprocessing
03.
EDA
04.
Modelling
05.
Model Evaluation

# PHASE 1: Data Collection

- Gather data from PushShiftAPI
- Has a limit of 100
- Use wrapper pmaw which will get requests every few seconds

# PHASE 2: Data Preprocessing

- Stop words : english, ha, wa, ive, im,..
- Lower case
- Remove extra whitespace
- Remove punctuations like commas, question marks
- Lemmatize the words: root words

# r/movies

“I mean, that could be a great comedy. It needs Bruce Campbell to work though.”

“I love that movie. One of my favorite horrors, and I would say it’s the second best zombie movie ever made. Only zombie movie that beats out The Godfather of the genre in my opinion is Train to Busan (2016).”

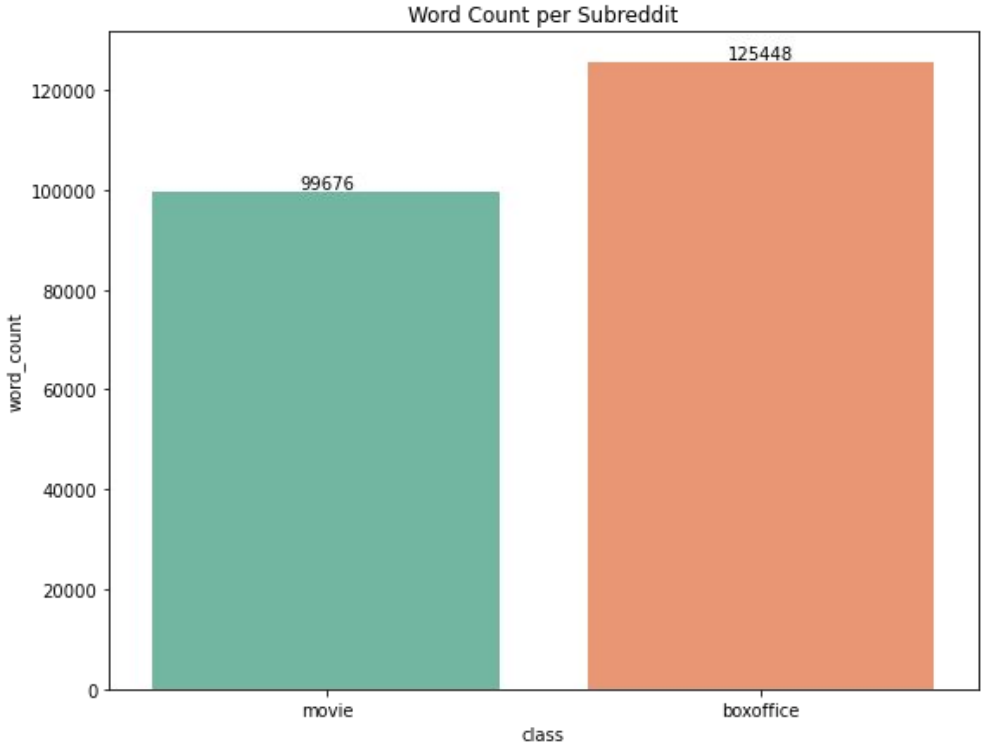
# r/boxoffice

'I’m with you. I absolutely expect Spiderverse to get a meaningful bump over the first movie.. but *triple*!? Yeah, not likely.'

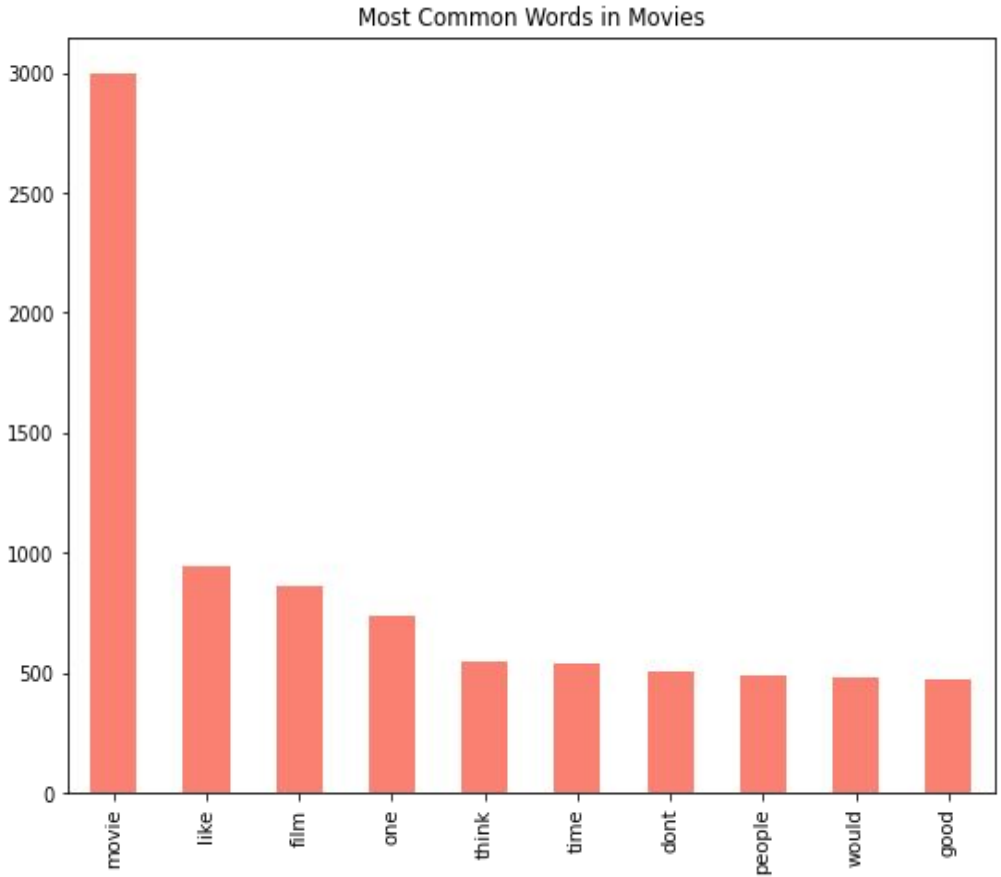
'Oh man you gotta watch The Wizard of Oz at least'

# EDA

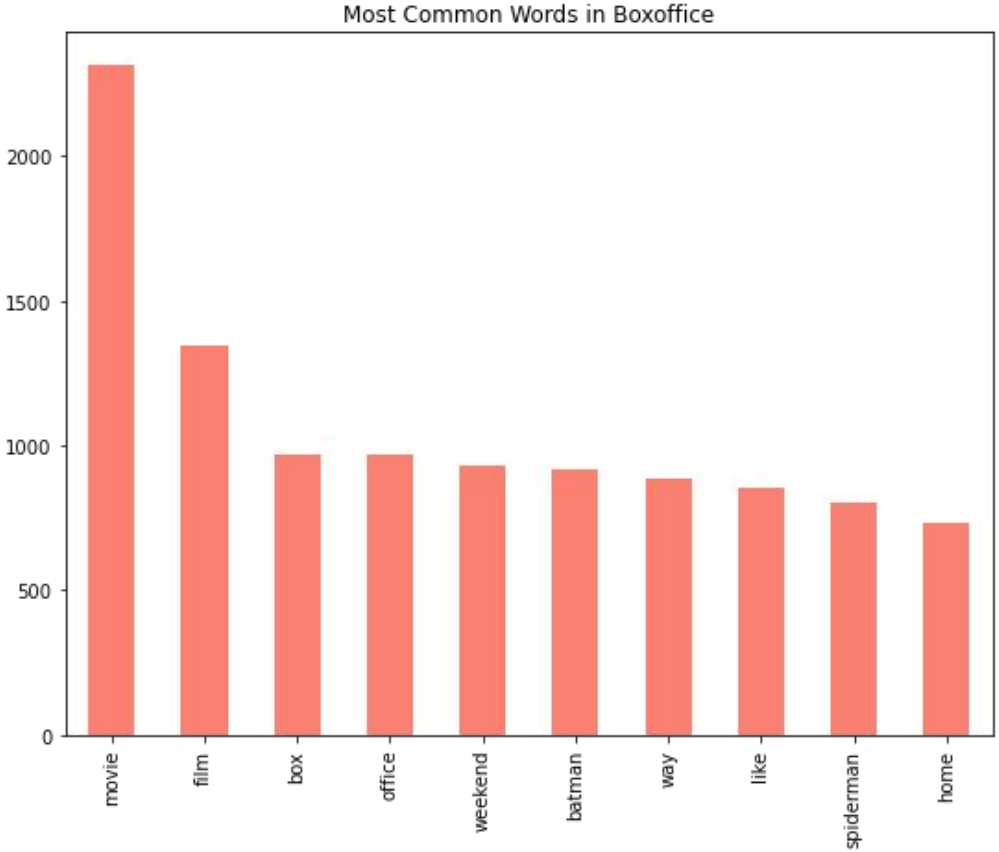
# Word Count


Word Count per Subreddit
125448
99676
120000
100000
80000
60000
40000
20000
0
word_count
movie
boxoffice
class

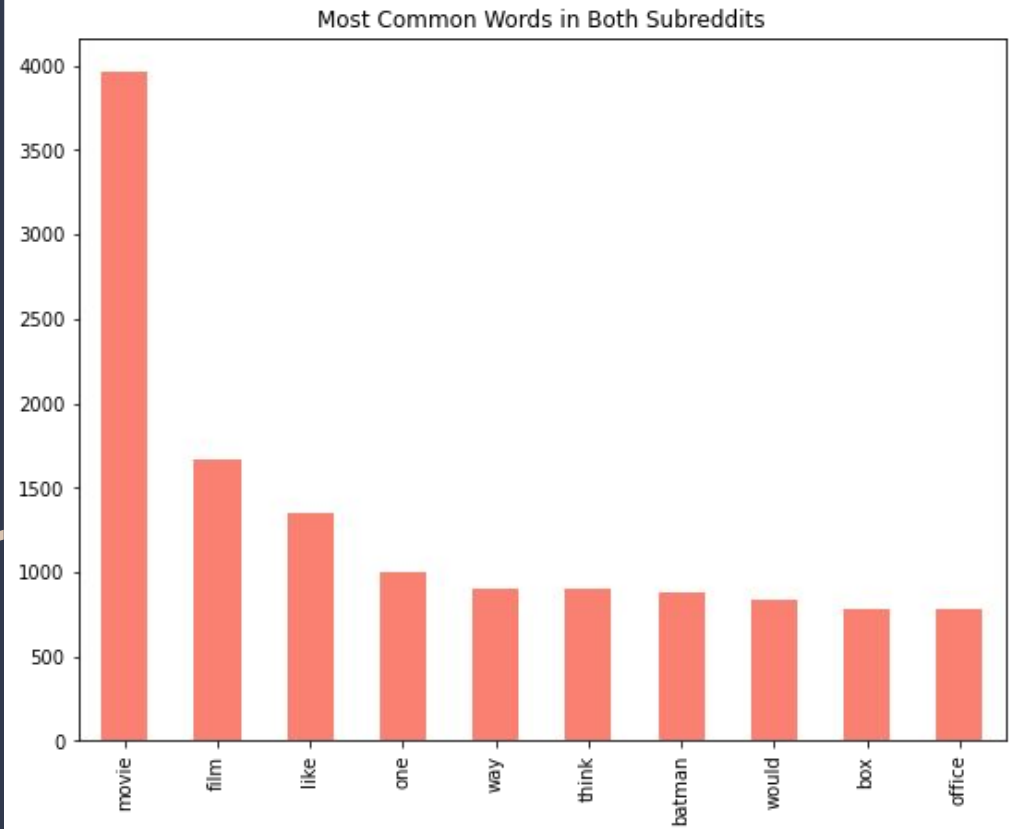
# r/movie


Most Common Words in Movies
3000
2500
2000
1500
1000
500
0
movie
like
film
one
think
time
dont
people
would
good

# r/boxoffice


Most Common Words in Boxoffice
2000
1500
1000
500
0
movie
film
box
office
weekend
batman
way
like
spiderman
home

# MOST COMMON WORDS



# MODELING

# Multinomial Naive Bayes

| train_score_c | test_score_c | train_score_t | test_score_t |
|---|---|---|---|
| 0.85 | 0.76 | 0.87 | 0.76 |
| 0.77 | 0.75 | 0.78 | 0.75 |
| 0.80 | 0.76 | 0.81 | 0.76 |
| 0.70 | 0.69 | 0.71 | 0.70 |

# Logistic Regression

| train_score_c | test_score_c | train_score_t | test_score_t |
|---|---|---|---|
| 0.915 | 0.768 | 0.860 | 0.769 |
| 0.834 | 0.765 | 0.860 | 0.769 |
| 0.807 | 0.761 | 0.825 | 0.765 |
| 0.816 | 0.760 | 0.834 | 0.762 |

# Random Forest Classifier

| train_score_c | test_score_c | train_score_t | test_score_t |
|---|---|---|---|
| 0.995 | 0.738 | 0.995 | 0.743 |
| 0.990 | 0.768 | 0.994 | 0.766 |
| 0.89 | 0.75 | 0.9 | 0.76 |
| | | | |

# Decision Trees Classifier

| train_score_c | test_score_c | train_score_t | test_score_t |
|---|---|---|---|
| 1.00 | 0.68 | 1.00 | 0.69 |
| 0.91 | 0.64 | 0.96 | 0.64 |
| 0.73 | 0.69 | 0.76 | 0.69 |
| 0.73 | 0.69 | 0.79 | 0.71 |

# Support Vector Classification

| train_score_c | test_score_c | train_score_t | test_score_t |
|---|---|---|---|
| 0.90 | 0.77 | 0.97 | 0.77 |
| 0.90 | 0.77 | 0.97 | 0.77 |
| 0.86 | 0.76 | 0.99 | 0.77 |
| | | | |

# CONCLUSION & RECOMMENDATION

- Most of the models had a similar score of around 0.76/0.77 except the Decision Tree which scored 0.71
- The overlap between the subreddits makes it a bit hard for the model to figure out which post is which
- More stop words can be used to see if there will be a difference in the scores
- More hyperparameter tuning can be carried out to look for any changes in the scores especially for Decision Tree
- For further analysis , maybe a sentiment analysis can be carried out on the data to see whether it is negative, neutral or positive with  a sentiment analyzing tool  like Vader