Project 3: Web APIs & NLP

USING REDDIT'S API AND PREDICT POST CONTENT

Pan Kah Fei DSIF-5
August 13, 2022

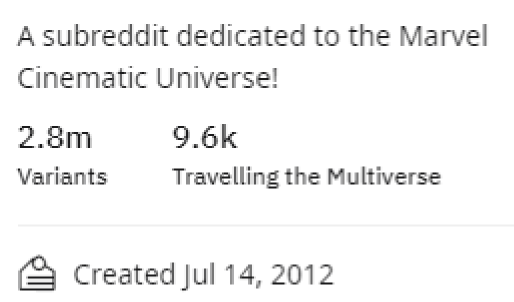
Agenda

- Problem Statement
- Methodology
- Result
- Conclusion and Recommendation

Problem Statement

Classify the subreddit posts using Natural Language Processing(NLP) and classication models.

1.r/marvelstudios



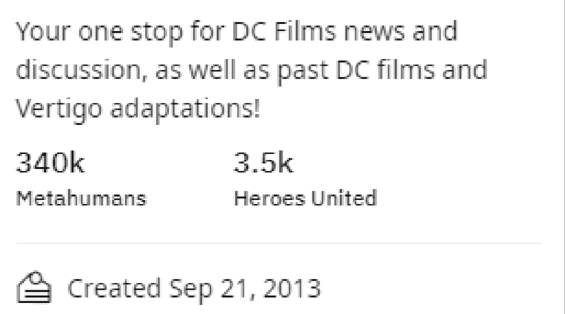
A subreddit dedicated to the Marvel Cinematic Universe!

2.8m Variants

9.6k Travelling the Multiverse

Created Jul 14, 2012

2.r/DC_Cinematic



Your one stop for DC Films news and discussion, as well as past DC films and Vertigo adaptations!

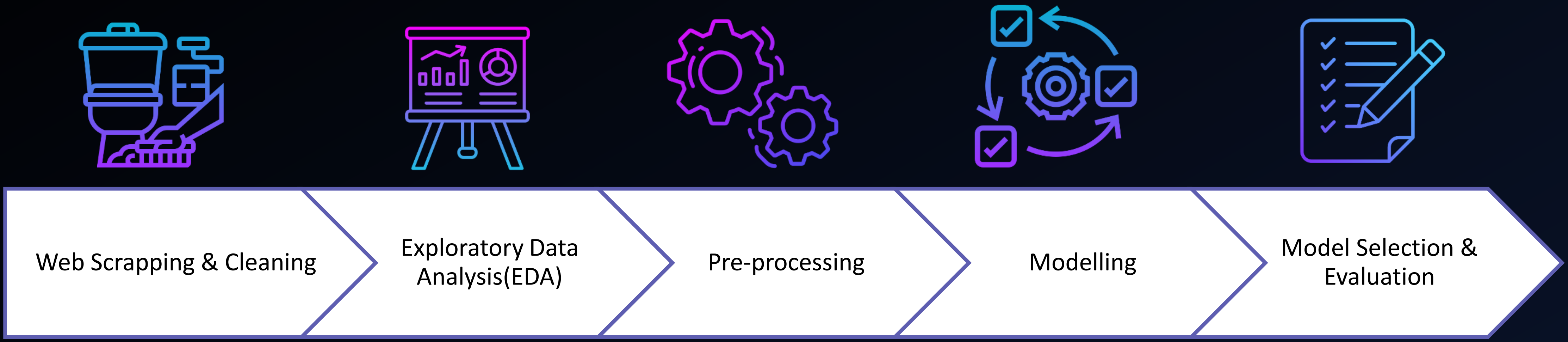
340k Metahumans

3.5k Heroes United

Created Sep 21, 2013

- The contents are similar (superhero, movie, meme etc)
- How well can they can be classified using NLP and SKLearn's classification model?

Methodology



Web Scrapping & Cleaning

Exploratory Data Analysis(EDA)

Pre-processing

Modelling

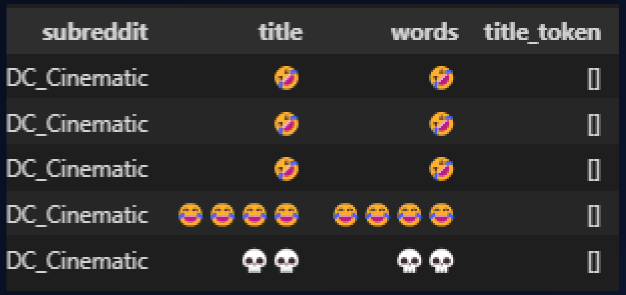
Model Selection & Evaluation

‘Pushshift API’ to extract posts from Reddit

- 2000 posts scrapped from each page

Cleaning

- Convert .Jason file into Pandas dataframe
- Remove blanks, duplicates, emojis
- Select only ‘title’ and ‘self-text’ as feature columns



subreddit	title	words	title_token
DC_Cinematic	🤣	🤣	[]
DC_Cinematic	🤣	🤣	[]
DC_Cinematic	🤣	🤣	[]
DC_Cinematic	😂😂😂😂	😂😂😂😂	[]
DC_Cinematic	💀💀	💀💀	[]

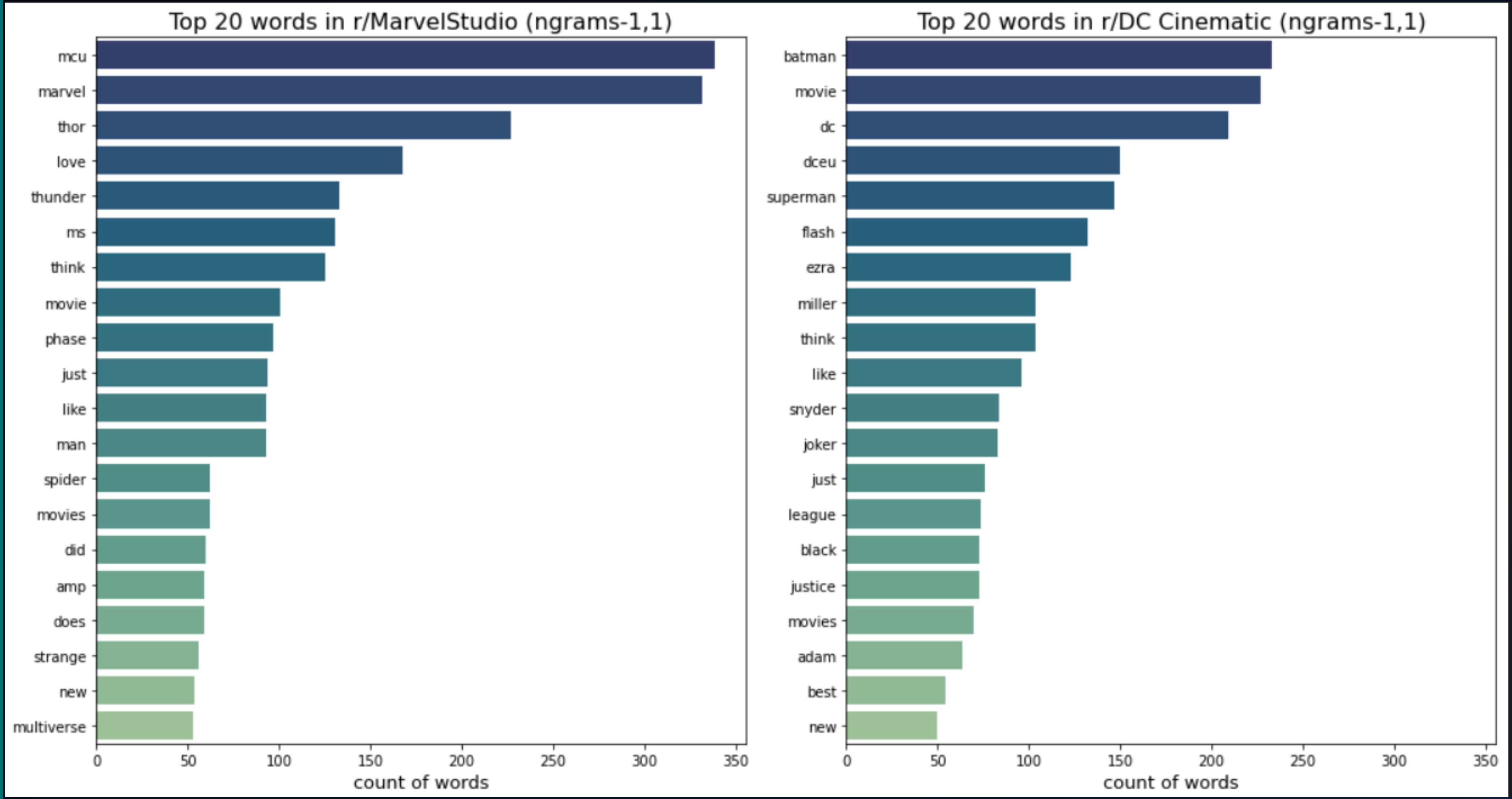
Methodology

Exploratory Data Analysis (EDA)

- Balanced Data:

  Class 0 = r/marvelstudios (50.02%) , Class 1 = r/DC_Cinematic(49.97%)

- Tokenize and Counvectorize with different N-grams.



Methodology

Pre- processing

Prepare data for Natural Language Processing (NLP) Model

Stop words removal

Use nltk ‘English’ stopwords library.

Tokenizing

Use Regular expression to parse string into multiple tokens

Lemmatizing

Reduce token words by removing suffixes but lighter touch than stemming

Methodology

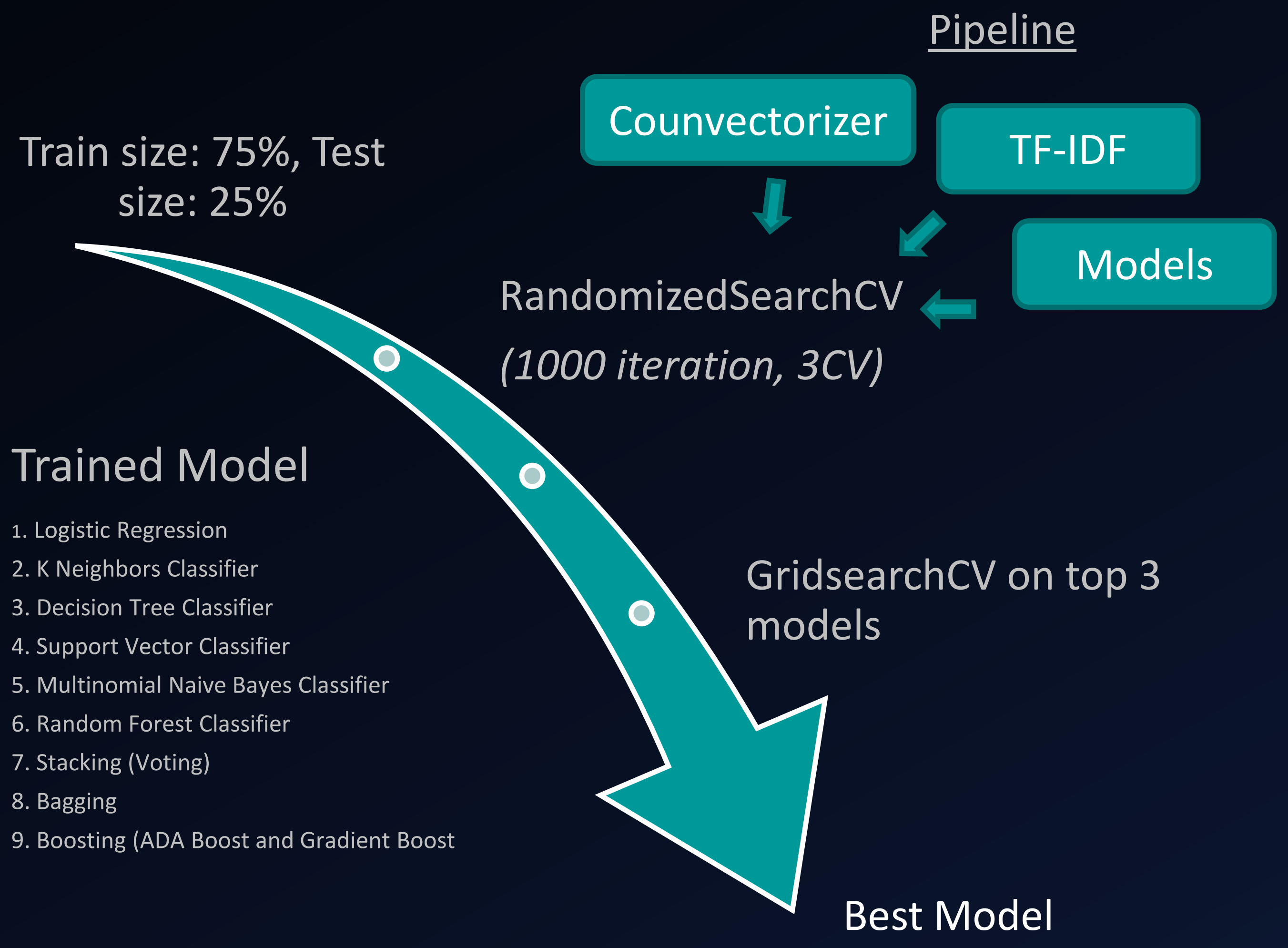
Modelling

Train test split on lemmatized dataset

Use *sklearn’s* RandomizedSearchCV to have an initial guess on hyperparameter range of each model

Pipeline groups of models with preprocessors

Use sklearn’s GridSearchCV to get the best hyperparameter



Trained Model

1. Logistic Regression
2. K Neighbors Classifier
3. Decision Tree Classifier
4. Support Vector Classifier
5. Multinomial Naive Bayes Classifier
6. Random Forest Classifier
7. Stacking (Voting)
8. Bagging
9. Boosting (ADA Boost and Gradient Boost

Methodology

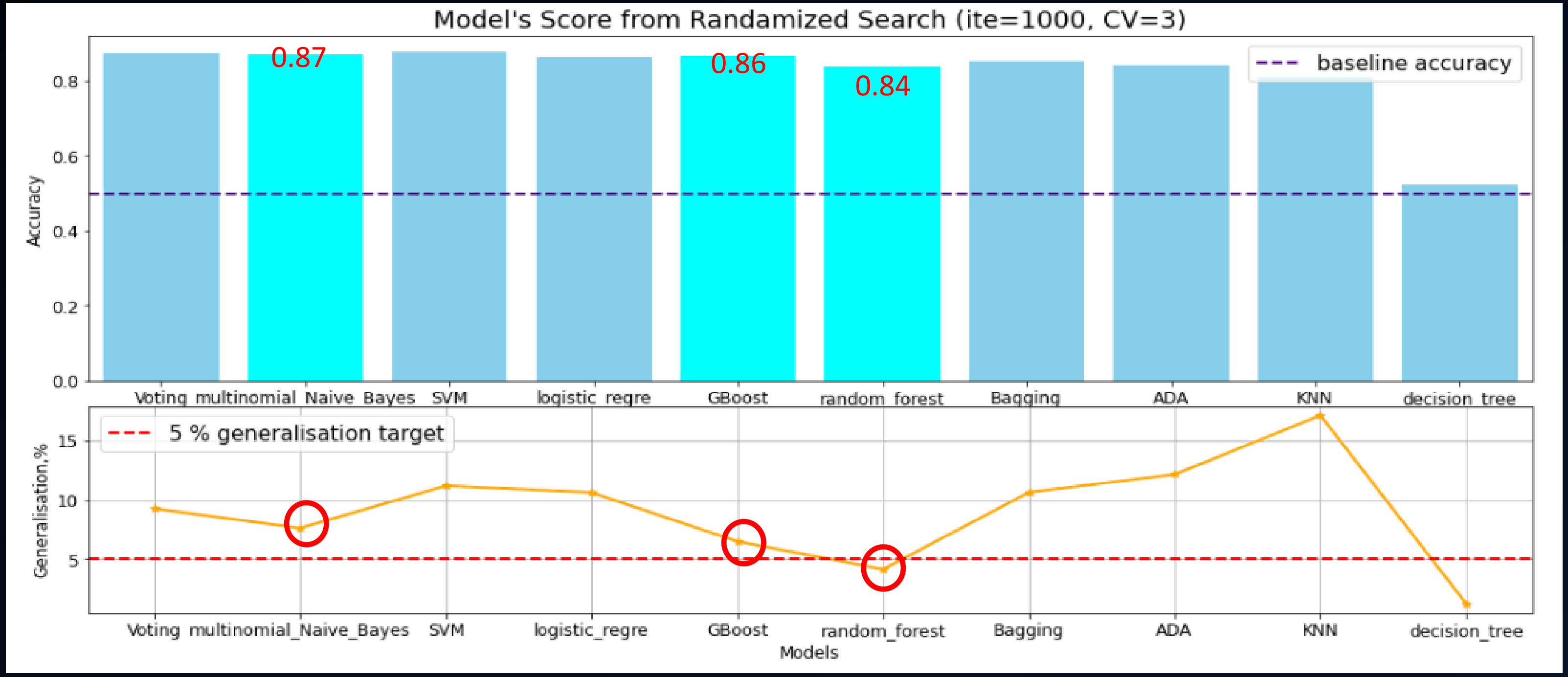
Model Selection

RandomizedSearch result is good enough to select the top 3 models

- Multinomial Naïve Bayes
- Gradient Boost(Decision Tree)
- Random Forest

Then follow by GridSearchCV to find the best performer

1st step: Randomized Search



2nd step: Grid Search

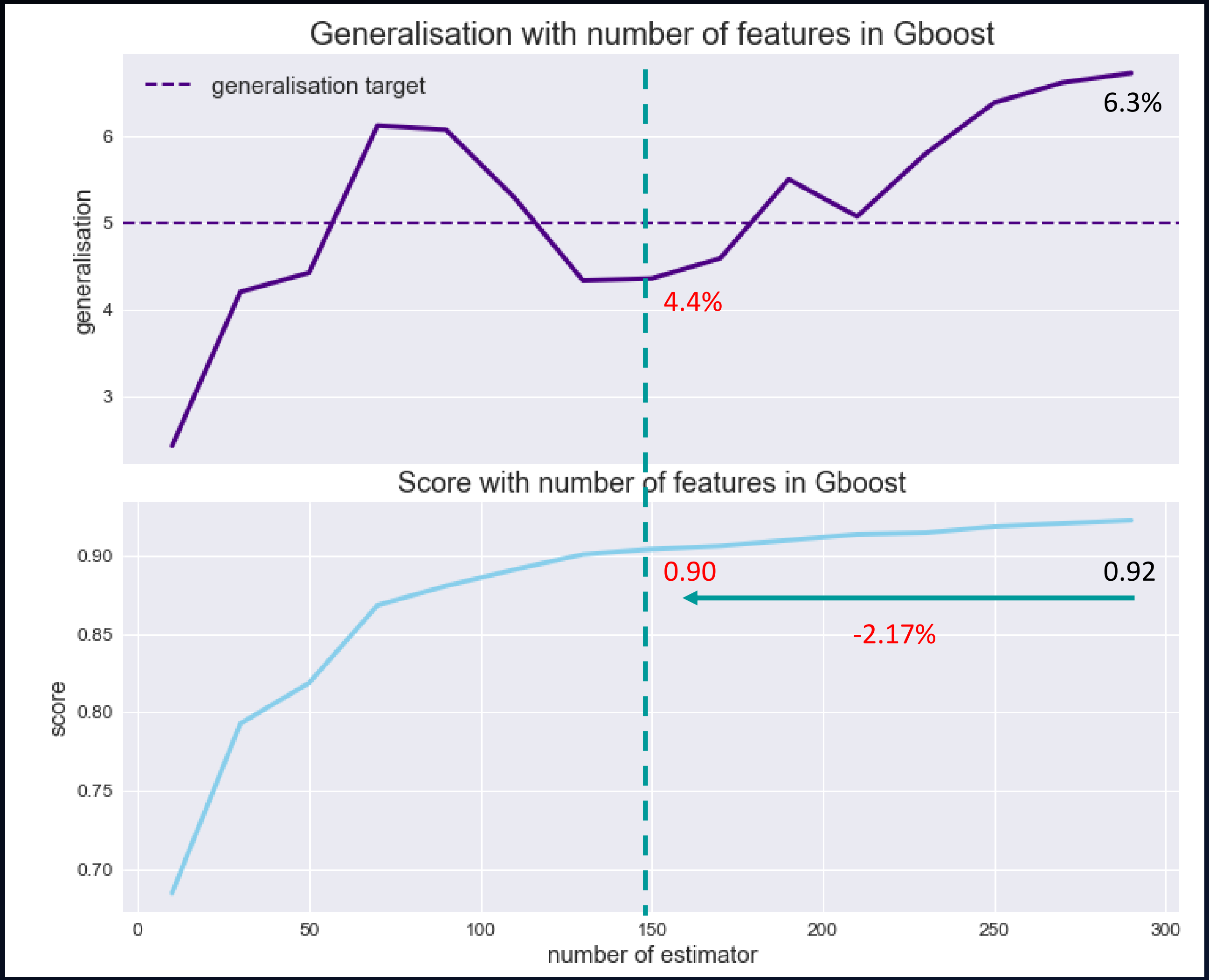
Model	Pre-processor	Train Score	Test Score	Test Score Improvement	Generalisation	Test ROC AUC	Actual Runtime (min)	Runtime (Full grid search, min)
MNB	Count Vectorizer	0.94	0.87	0%	7.4%	0.96	0.51	4555
RF	Count Vectorizer	0.89	0.85	1.2%	4.53%	0.95	4.45	6220877
GBOOST	Count Vectorizer	0.92	0.87	1.2%	6.3%	0.95	5.5	6121

Methodology

Model Selection

Further study on Gboost to narrow down generalization gap:

- Shrinkage/Weighted Updates
  - learning rate
  - number of estimator
- Random Sampling/Stochastic Boosting
  - subsample



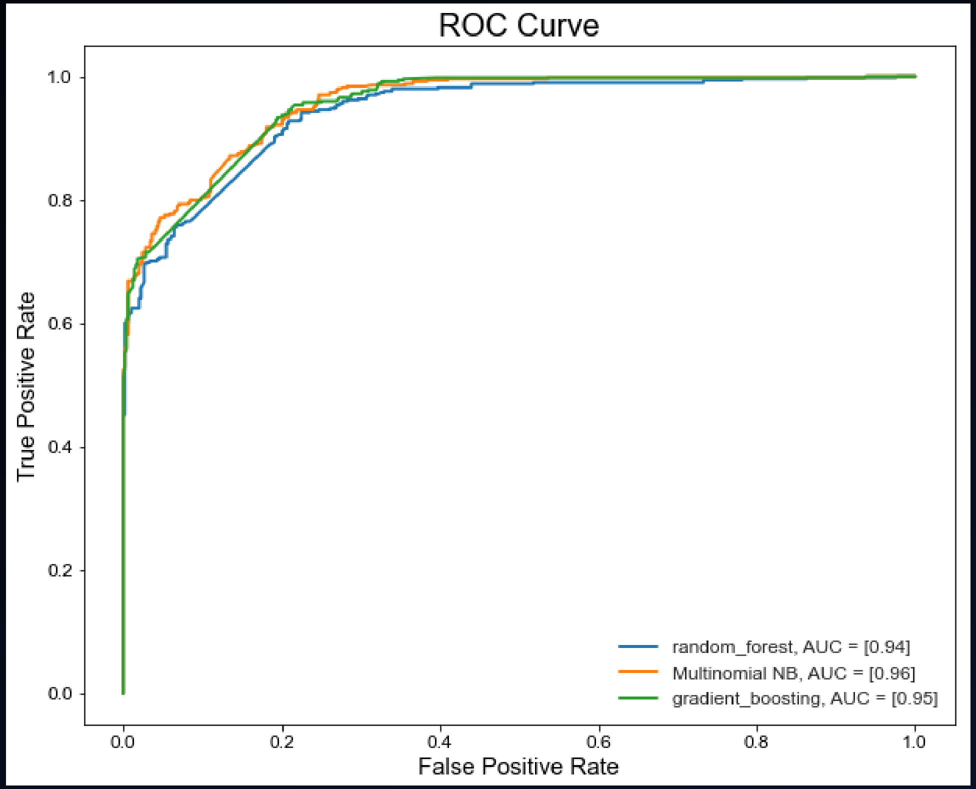
150 number of estimator is chosen for gradient boost model

Methodology

Evaluation

The generalization gap must <5% as requested for this project

All beat baseline accuracy of 50%



Model	Pre-processor	Train Accuracy Score	Test Accuracy Score	Generalisation	Test ROC AUC	Test F1 score(class 0)	Test F1 score(class 1)
MNB	Count Vectorizer	0.92	0.86	7.4%	0.96	0.87	0.86
RF	Count Vectorizer	0.88	0.84	4.53%	0.94	0.85	0.83
GBOOST	Count Vectorizer	0.90	0.87	4.36%	0.95	0.86	0.88

Result

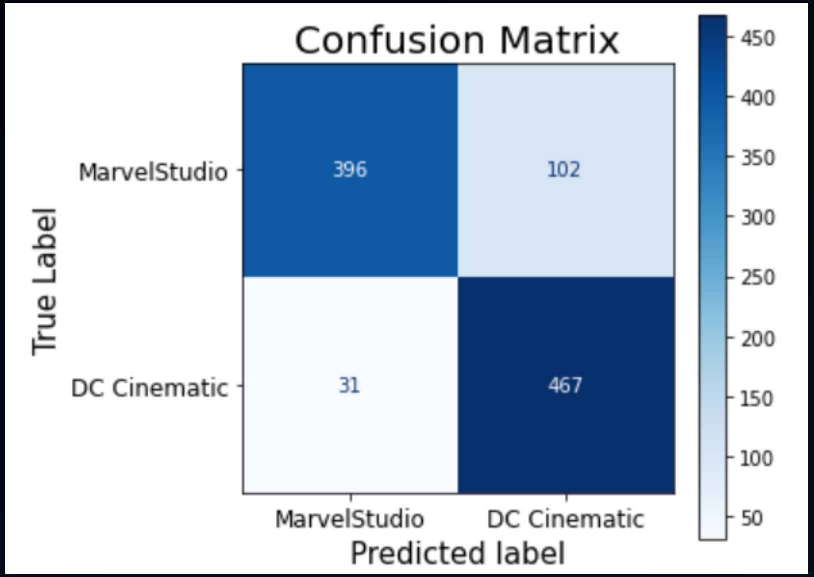
Winner - Gradient Boost

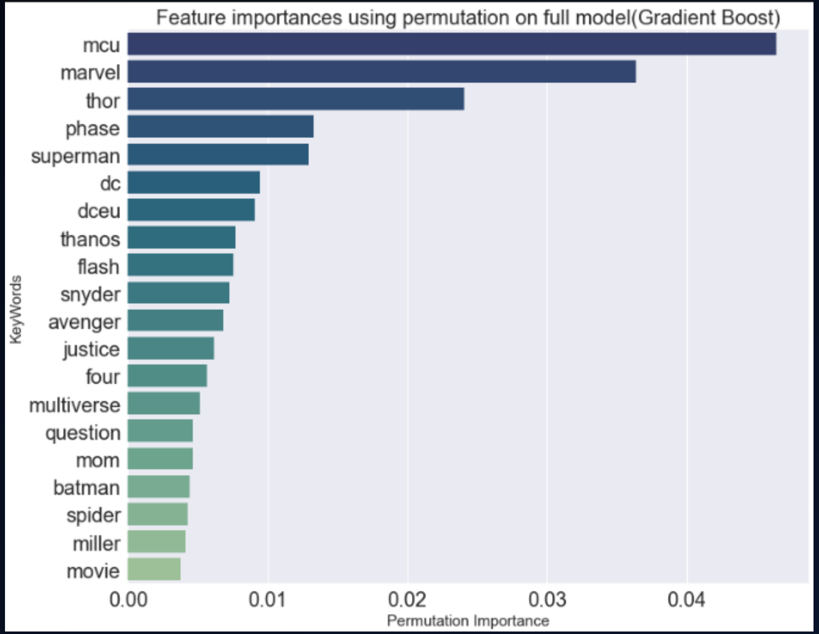
CountVectoriser

- max_df=0.85
- Max_feature=1500
- ngram_range = (1,1)

Parameter

- Learning_rate=0.12
- Max_dept = 4,
- Max_features = 12
- Min_sample_leaf =2,
- N_estimator=150,
- Min_sample_split=10





Confusion matrix

F1 score 88%

Accuracy score 87%

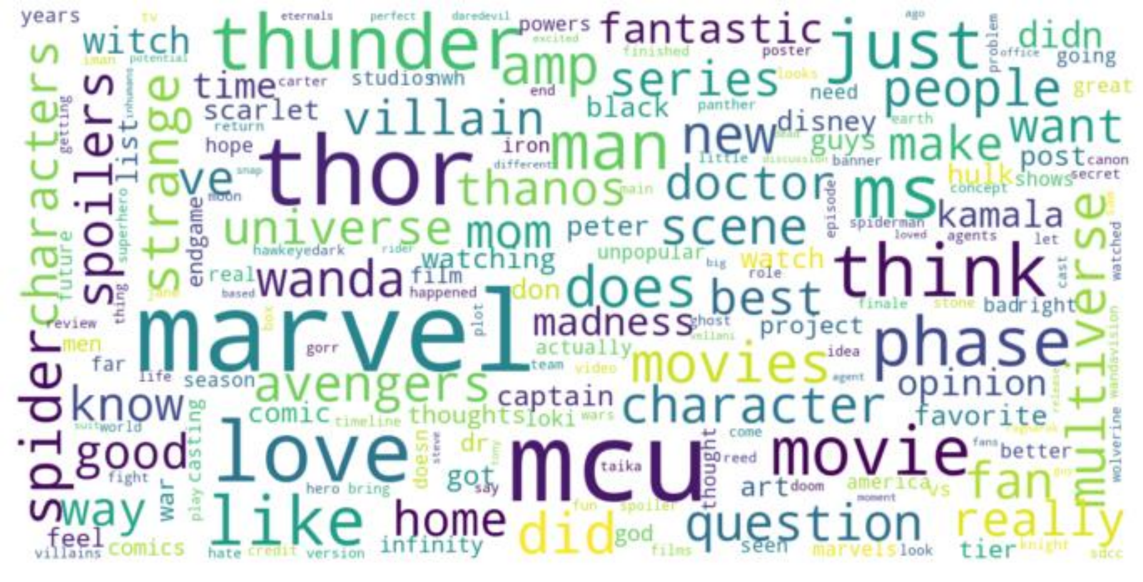
Best performance with threshold of 0.5

Permutation Importance

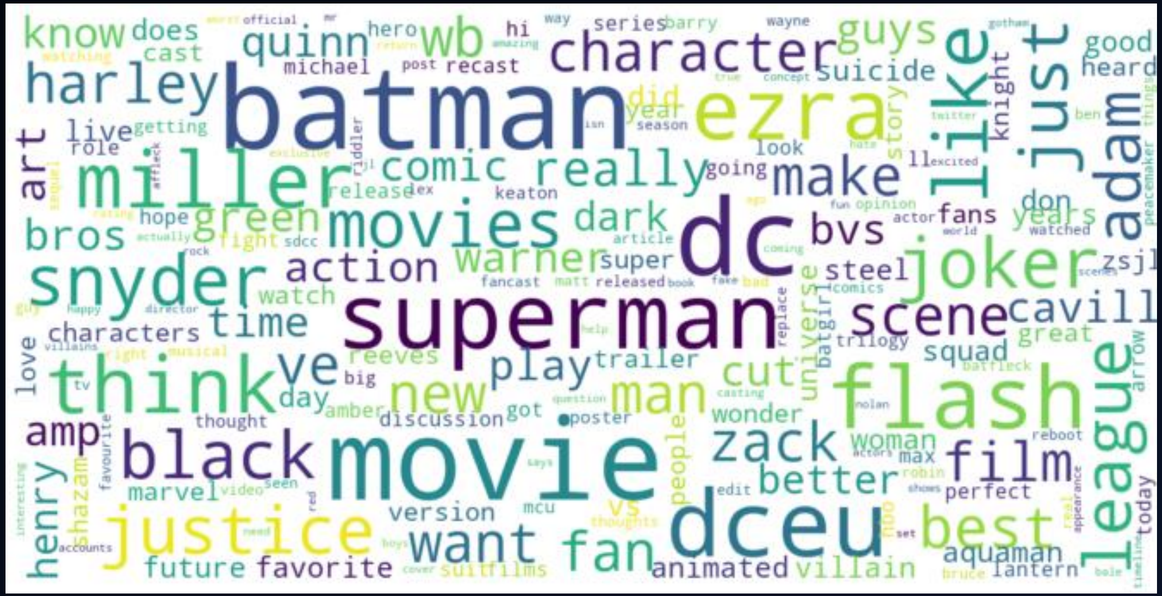
Model-agnostic global explanation method

Top 20 words with high predictive power

Word Cloud Visualization on key terms for each sub:



r/marvelstudios



r/DC_Cinematic

Conclusion & Recommendation

- Highest-performing model, Gradient Boost is doing well in predicting the subreddit posts with accuracy of 87%
- Every model has its pros and cons, we must gauge the trade off between the computation time and target scores.
- The method of using randomizedsearchCV to narrow down the hyperparameter range followed by full GridsearchCV able to save whole computing time by enormous amount (1112% shorter run time for Gradient boost model).
- Advance model like CatBoost and XGBoost can be tested out in the future for NLP project

Thank you

Q & A