Project 3: Reddit Web APIs & NLP Classification

For

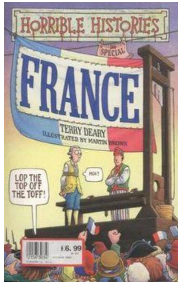
GENERAL ASSEMBLY

DSI-14
By Russell Quah

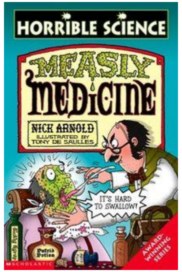
Problem statement

Problem statement

- As part of the marketing and research team for a series of Popular Science/ History books, my current goal is to increase the **science** book sales and online readership.
- Reddit has a good repository of questions that users ask on r/askscience.
- However, due to a server error the web scraping from r/askscience got mixed up with the data from r/askhistorians as well!

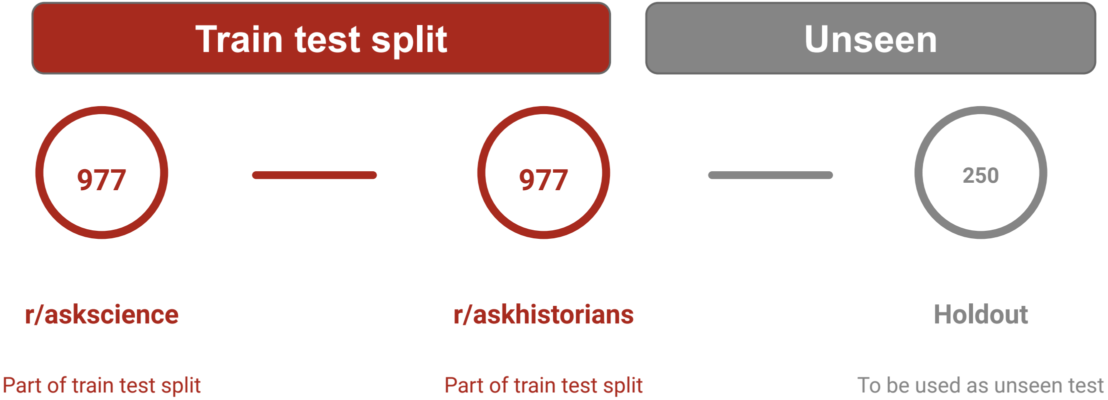


- 1 Naive Bayes Classifier
- 1 Support Vector Machine Classifier
- Evaluate the models based on:
    - accuracy (% predictions the model gets correct, both askscience and askhistorians)
    - precision (% predicted askscience when it is actually askscience)
    - sensitivity (% predicted askscience out of all correct predictions)
- choose the best performing model to test it on the holdout csv



Source: https://shop.scholastic.co.uk/horrible-science, https://shop.scholastic.co.uk/horrible-histories

Web Scraping



Train test split

Unseen

977

977

250

r/askscience

Part of train test split

r/askhistorians

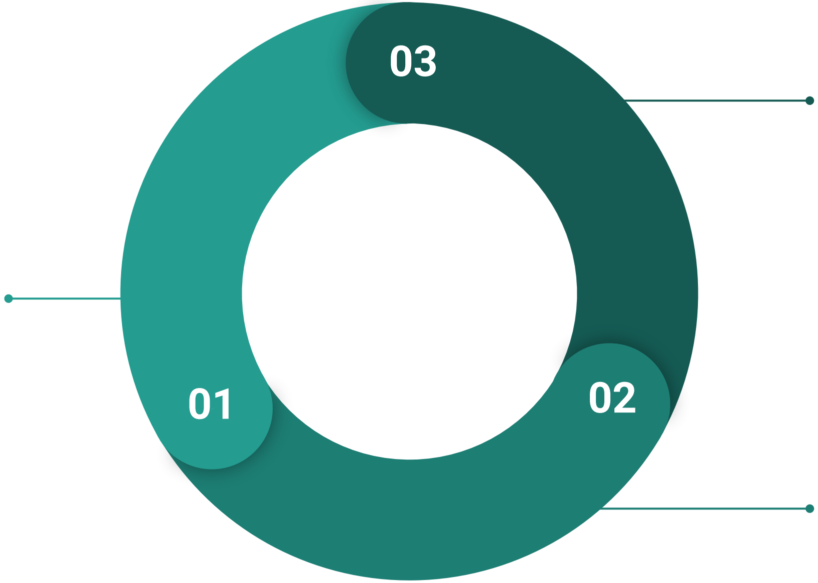
Part of train test split

Holdout

To be used as unseen test

Source: https://www.reddit.com/r/askscience, https://www.reddit.com/r/AskHistorians

Data Cleaning



Removal of title posts

remove any 'meta', 'moderator', 'ama series' or 'advertising posts'

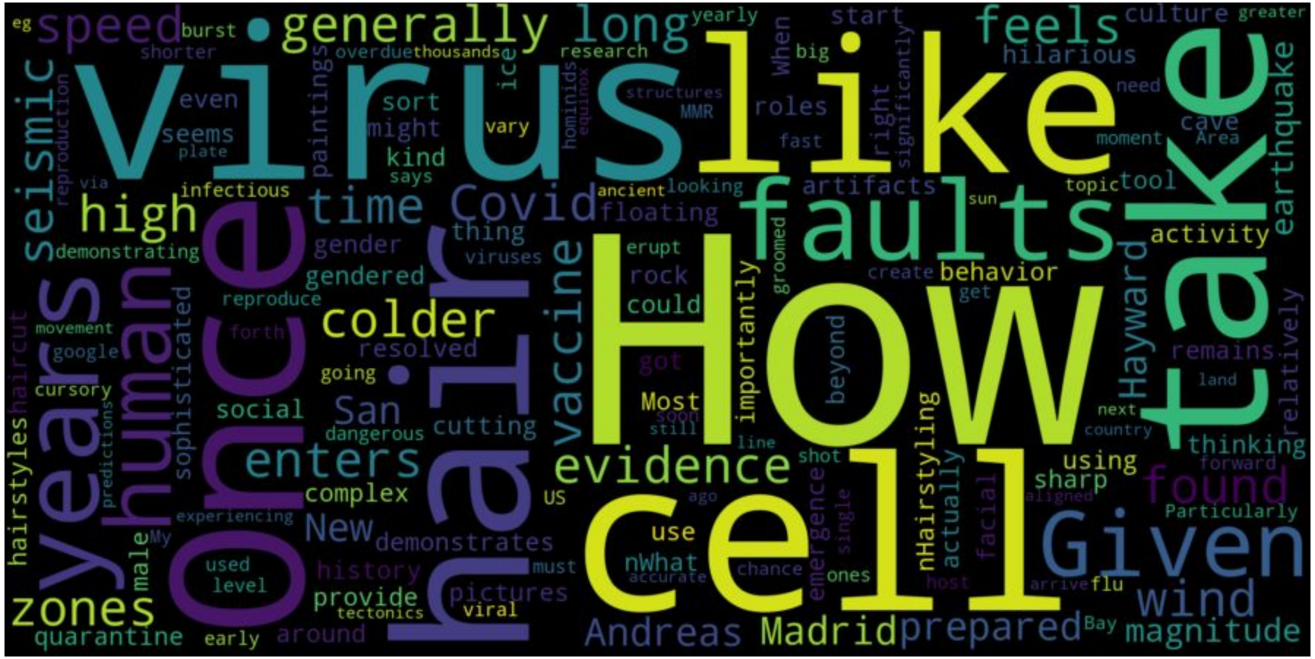
Concat ‘selftext’ and ‘title’

Filling missing values with blanks

Removal of text

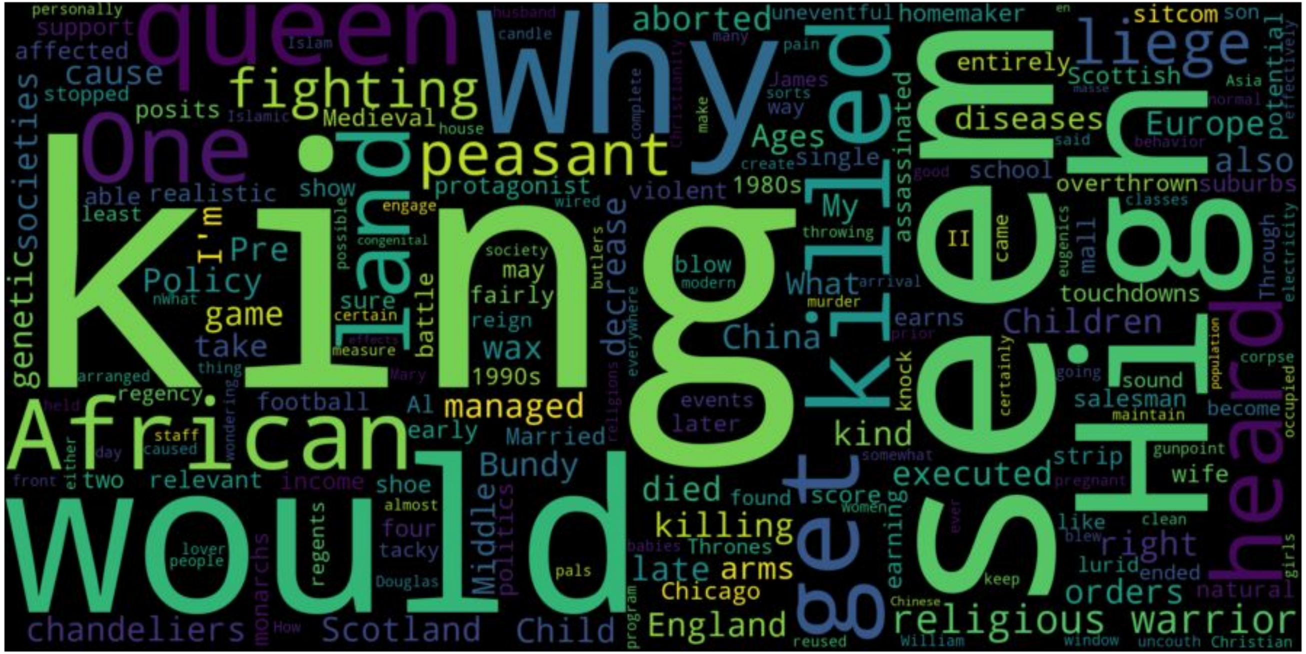
Any identifiable text like ‘AskScience’ or ‘AskHistorians’ in the title

Exploratory Data Analysis

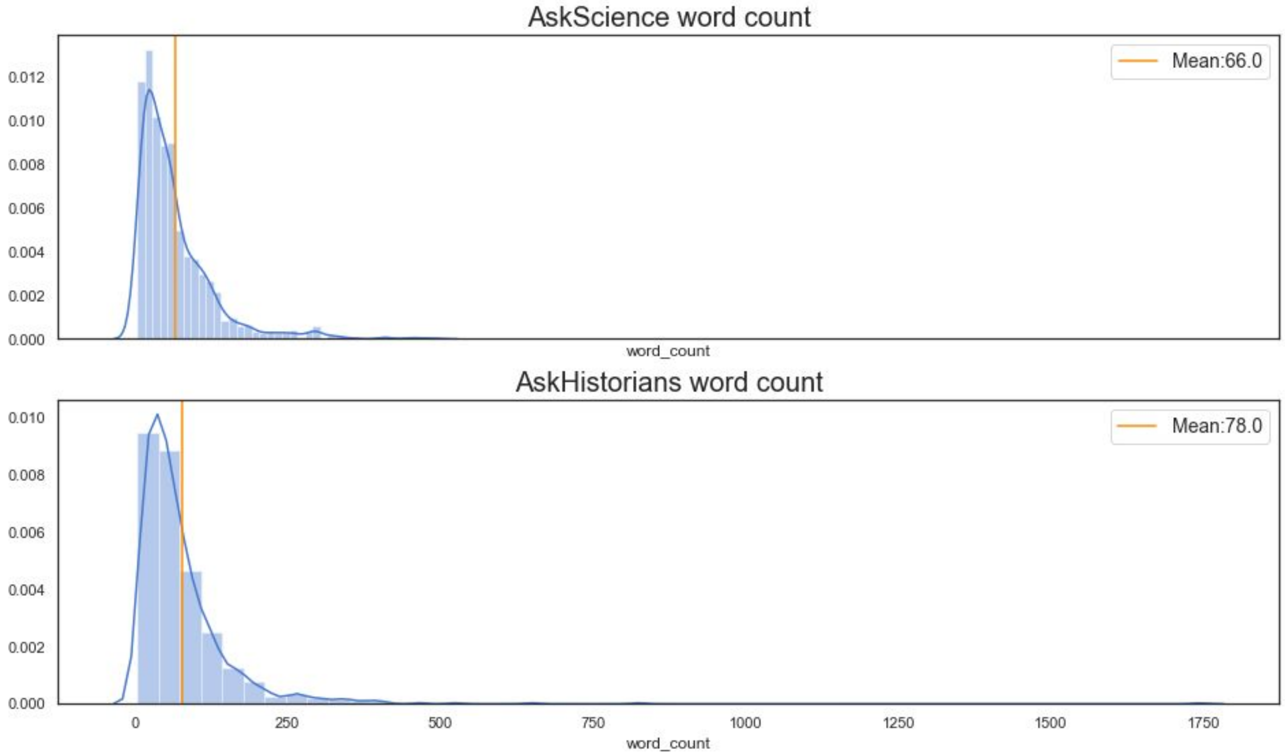

AskScience Wordcloud

Exploratory Data Analysis

AskHistorians Wordcloud



Exploratory Data Analysis



AskScience word count

AskHistorians word count

Pre-processing

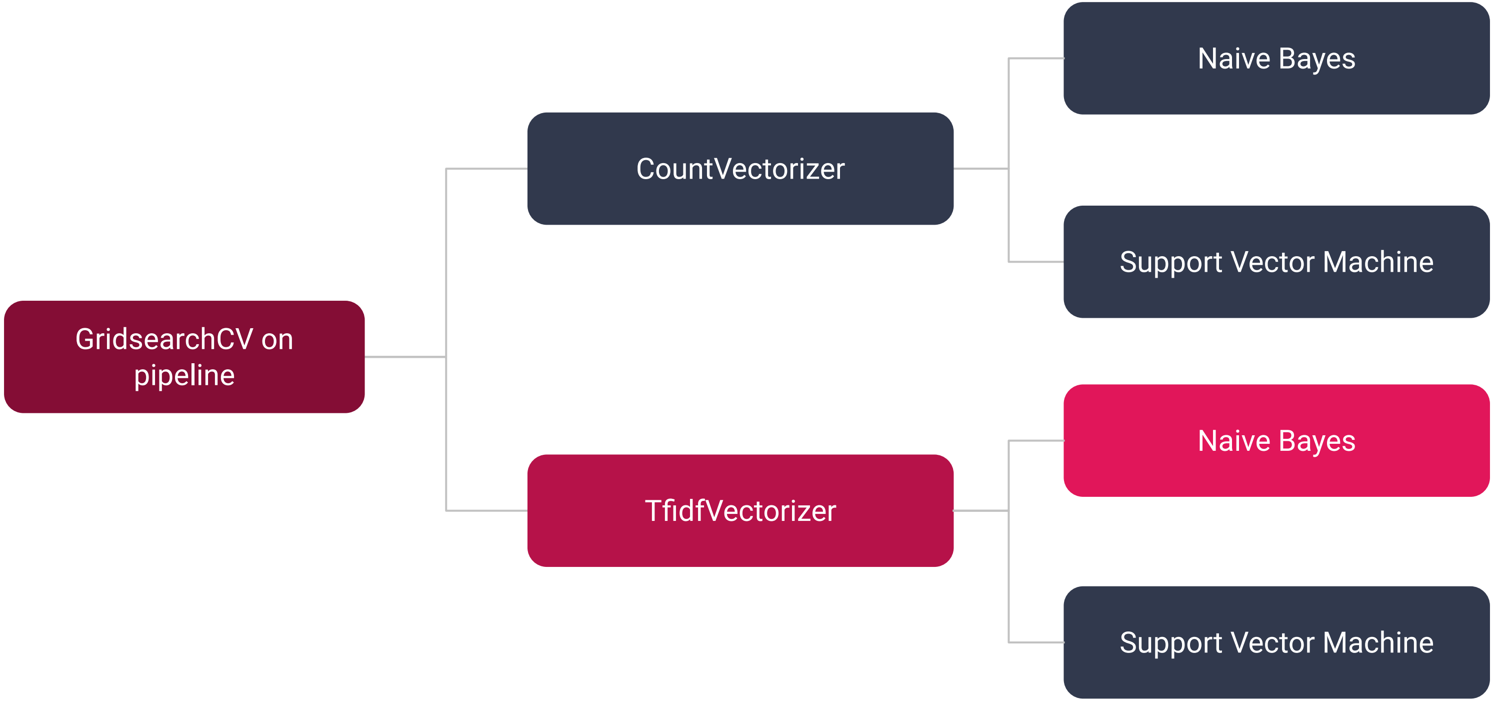
Train test split

- 1453 rows for X_train
- 485 rows for X_test
- 250 holdout

Pre-processing

- Used lemmatizer instead of stemming
- Stopwords + ‘askscience’ + ‘askhistorians’
- Removal of html
- Lowercase
- Removal of non letters

Classification



GridsearchCV on pipeline

CountVectorizer

Naive Bayes

Support Vector Machine

TfidfVectorizer

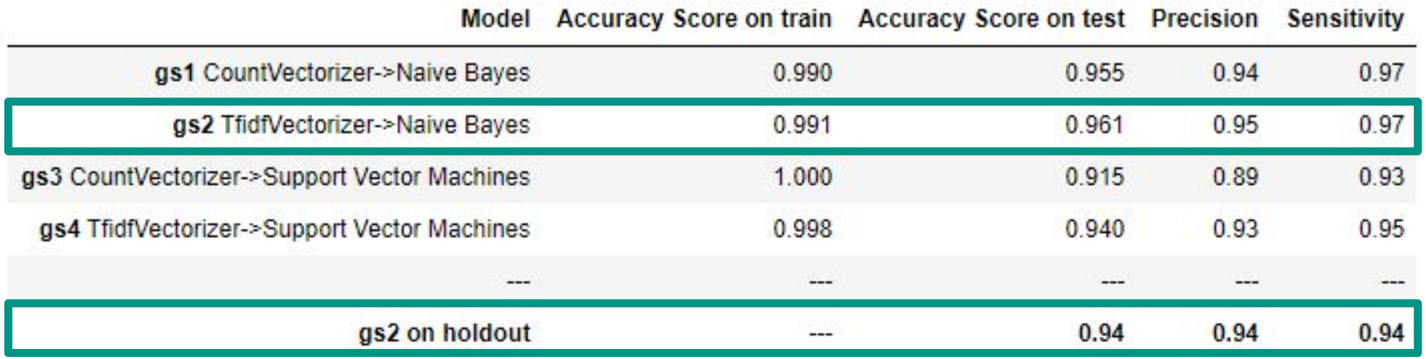
Naive Bayes

Support Vector Machine

Best model

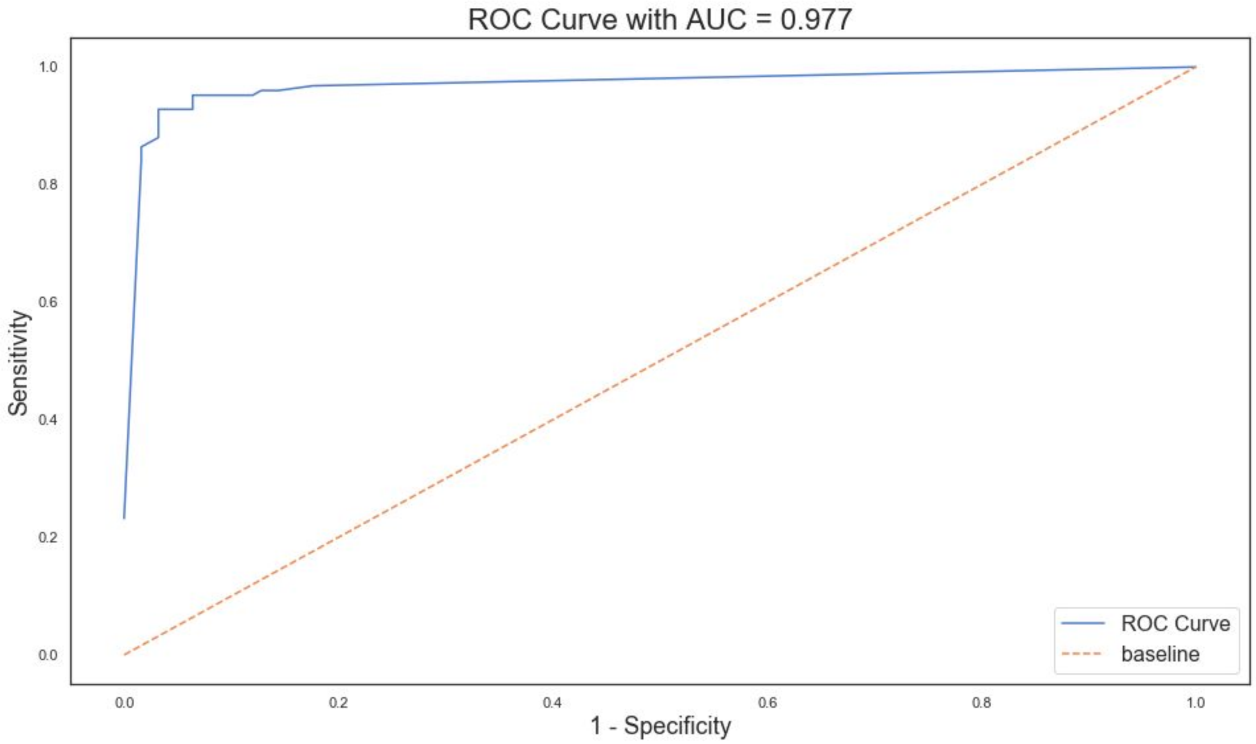
- 'nb__alpha': 1,
- 'tfid__max_df': 0.85,
- 'tfid__max_features': 4000,
- 'tfid__min_df': 2,
- 'tfid__ngram_range': (1, 1)

Evaluation



Model	Accuracy Score on train	Accuracy Score on test	Precision	Sensitivity
gs1 CountVectorizer->Naive Bayes	0.990	0.955	0.94	0.97
gs2 TfidfVectorizer->Naive Bayes	0.991	0.961	0.95	0.97
gs3 CountVectorizer->Support Vector Machines	1.000	0.915	0.89	0.93
gs4 TfidfVectorizer->Support Vector Machines	0.998	0.940	0.93	0.95
---	---	---	---	---
gs2 on holdout	---	**0.94**	**0.94**	**0.94**

Evaluation


ROC Curve with AUC = 0.977
Sensitivity
1 - Specificity
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC Curve
baseline

Conclusion

- TfidfVectorizer into Naive Bayes scored **0.94** across all metrics on the holdout dataset

- We are able to correctly identify askscience and askhistorians posts in order to improve book sales and online viewership.

- Further improvements can be made by:
  - Increasing sample size to reduce overfitting
  - Using other models like RFC, log etc.
  - Ensembling models will help reduce overfitting
  - Testing the model on more similar datasets

Thank you

Q&A

Image sources are taken from respective subreddits and scholastic. No copyright infringement intended