Machine Learning on Subreddit Posts

Esther Khor

Problem Statement.

Client

A start-up selling audio equipment online.

Context

- Create a chatbot that can help direct customers to the items closest to what they are looking for.
- Would also like to use the chatbot for potential upselling

Goal

Classify posts from two different subreddits based on their ‘title’ and ‘selftext’

Subreddits

Subreddit #1

r/Earbuds

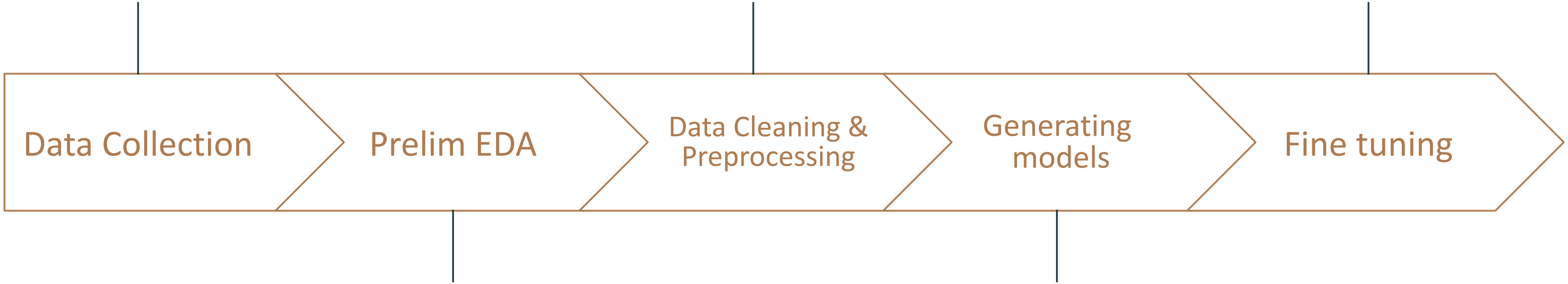
A community for discussion, reviews related to earbuds.

Subreddit #2

r/Headphones

A place for discussion, news, reviews and DIY projects related to portable audio, headphones, headphone amplifiers and DACs.

Process Flow



Data Collection

Web-scraping of posts done via Pushshift's Reddit API

Prelim EDA

- Distribution of type of posts
- Analyzing Title and Post Length

Data Cleaning & Preprocessing

- HTML links
- Non-alphanumeric characters
- Title + Body combination

Generating models

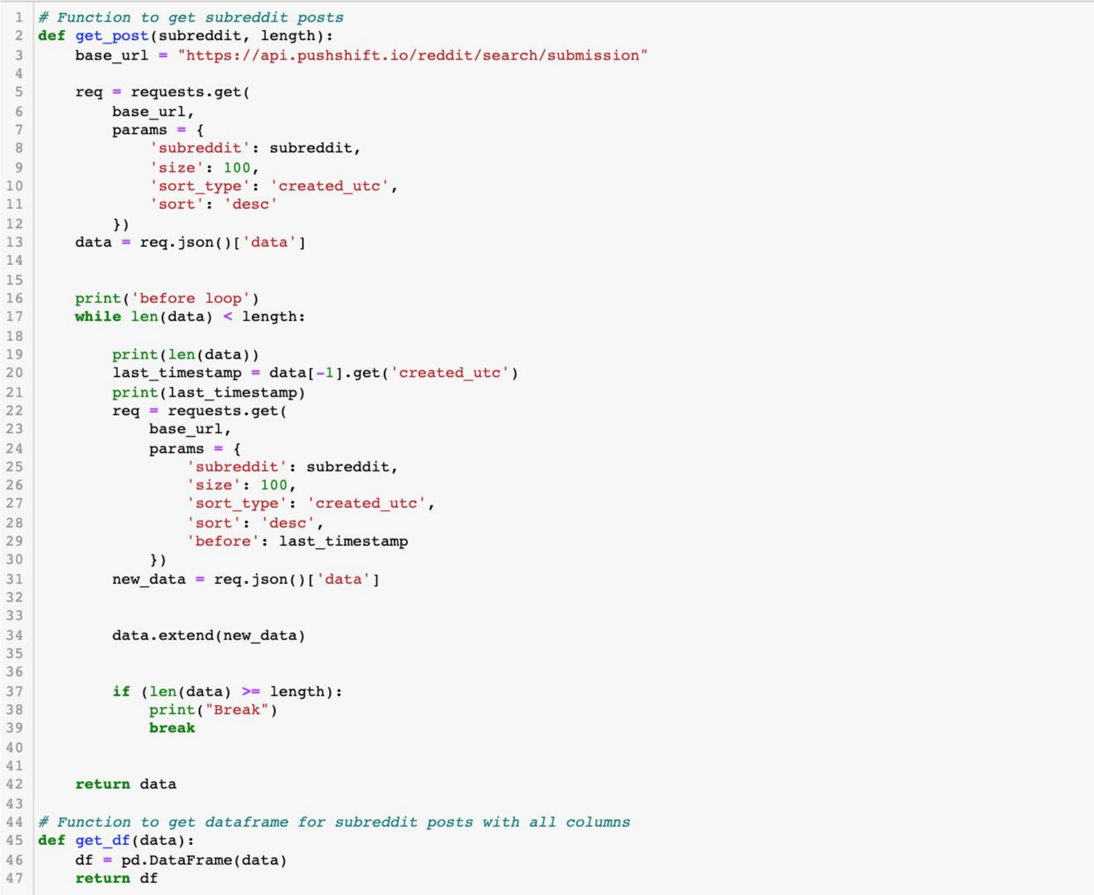
- Logistic Regression
- Random Forest
- Multinomial Naive Bayes
- Support Vector Machine (SVM) classification
- Boosting

Fine tuning

Hyperparameter tuning of Vectorizers and Models

Data Collection

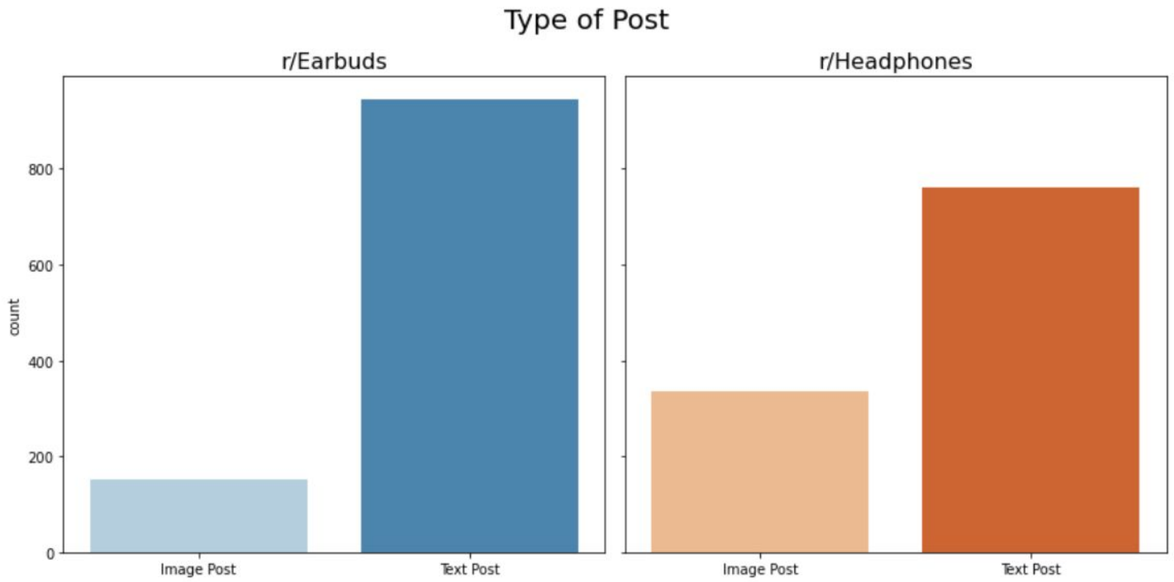
Web-scraping



```python
# Function to get subreddit posts
def get_post(subreddit, length):
    base_url = "https://api.pushshift.io/reddit/search/submission"

    req = requests.get(
        base_url,
        params = {
            'subreddit': subreddit,
            'size': 100,
            'sort_type': 'created_utc',
            'sort': 'desc'
        })
    data = req.json()['data']


    print('before loop')
    while len(data) < length:

        print(len(data))
        last_timestamp = data[-1].get('created_utc')
        print(last_timestamp)
        req = requests.get(
            base_url,
            params = {
                'subreddit': subreddit,
                'size': 100,
                'sort_type': 'created_utc',
                'sort': 'desc',
                'before': last_timestamp
            })
        new_data = req.json()['data']


        data.extend(new_data)


        if (len(data) >= length):
            print("Break")
            break


    return data

# Function to get dataframe for subreddit posts with all columns
def get_df(data):
    df = pd.DataFrame(data)
    return df
```

Preliminary EDA

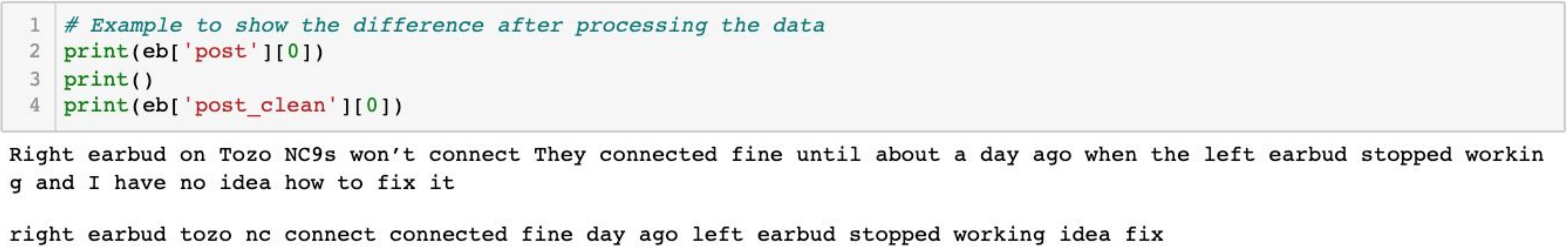
- Type of Post
- Title Lengths
- Post Lengths



Data Cleaning & Preprocessing

- Remove unnecessary columns
- Create and fill new column: ‘post’
  - Empty ‘selftext’ vs
  - Filled ‘selftext’

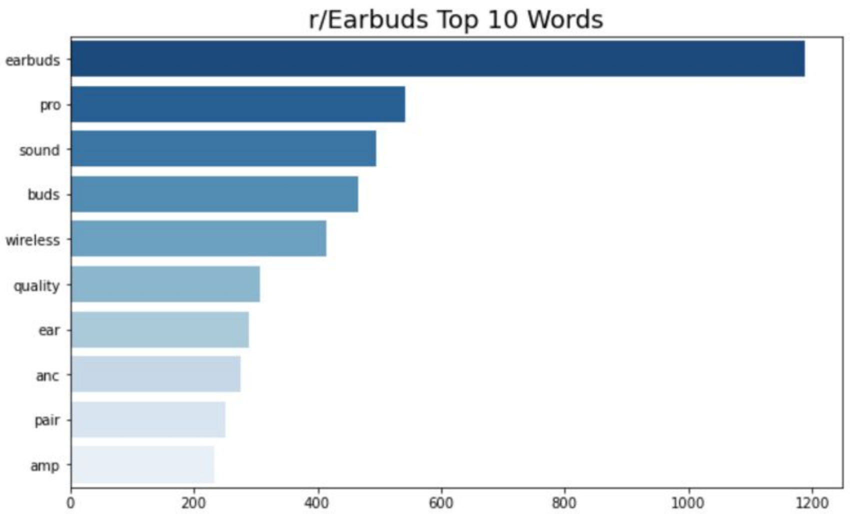
- Remove HTML , non-alphanumeric
- Lowercase all words
- Split into individual tokens
- Lemmatize
- Combine data frames and map

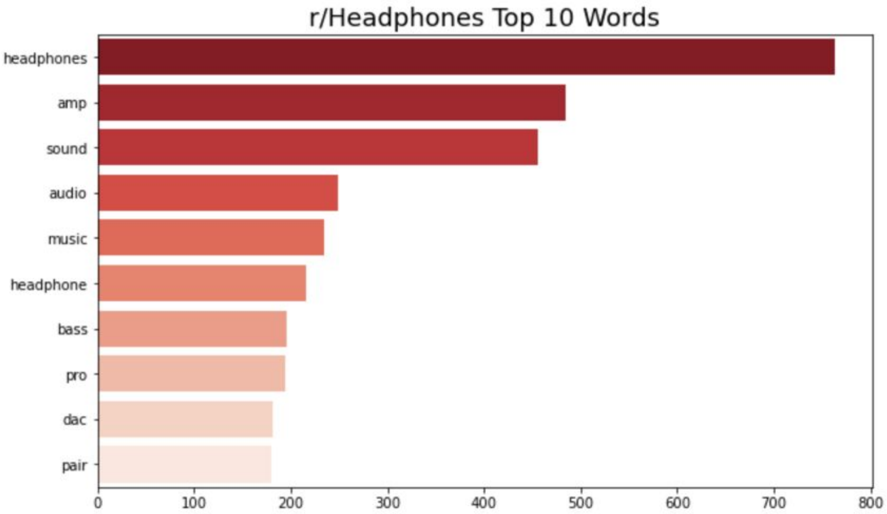


```python
# Example to show the difference after processing the data
print(eb['post'][0])
print()
print(eb['post_clean'][0])
```

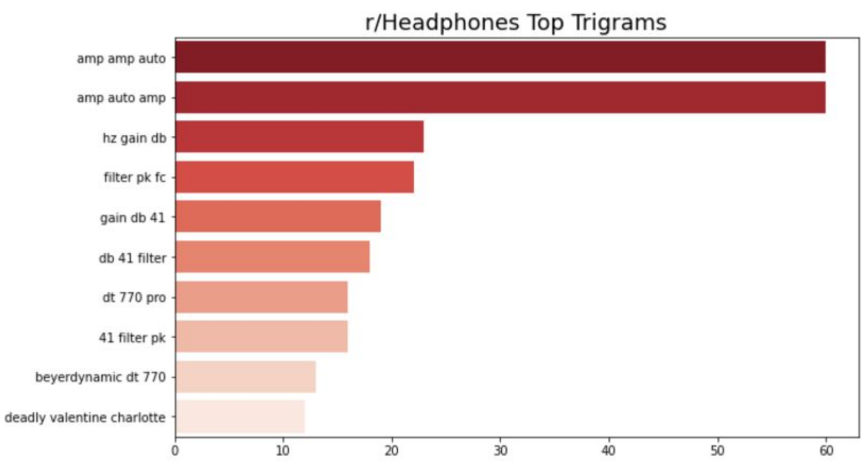
```
Right earbud on Tozo NC9s won’t connect They connected fine until about a day ago when the left earbud stopped workin
g and I have no idea how to fix it

right earbud tozo nc connect connected fine day ago left earbud stopped working idea fix
```

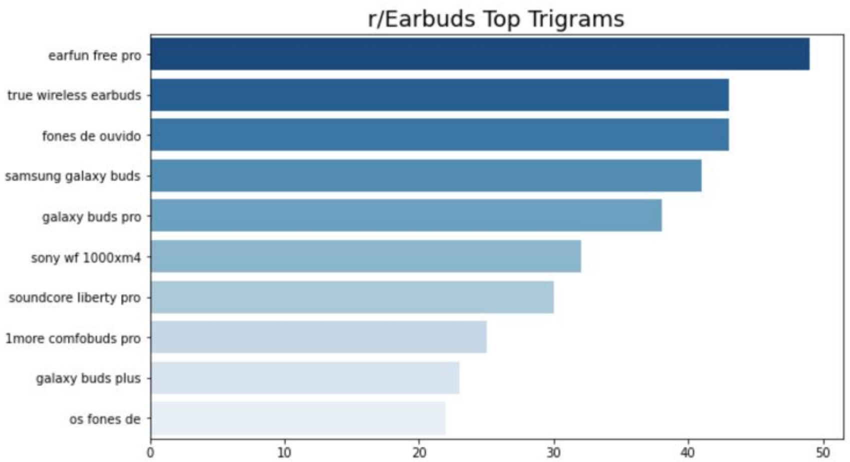
Analysis of n-grams

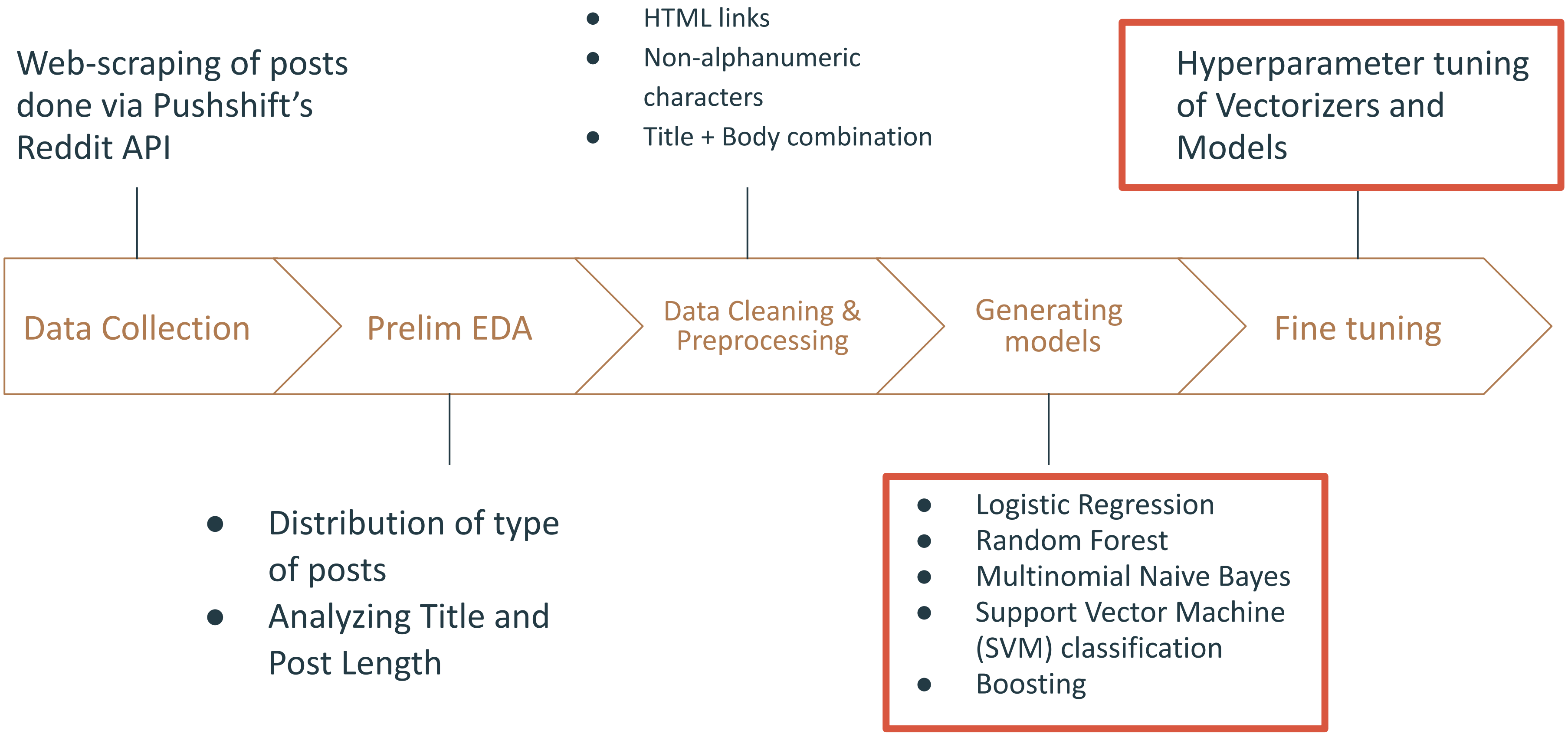




Analysis of n-grams







Data Collection

Web-scraping of posts done via Pushshift's Reddit API

Prelim EDA

- Distribution of type of posts
- Analyzing Title and Post Length

Data Cleaning & Preprocessing

- HTML links
- Non-alphanumeric characters
- Title + Body combination

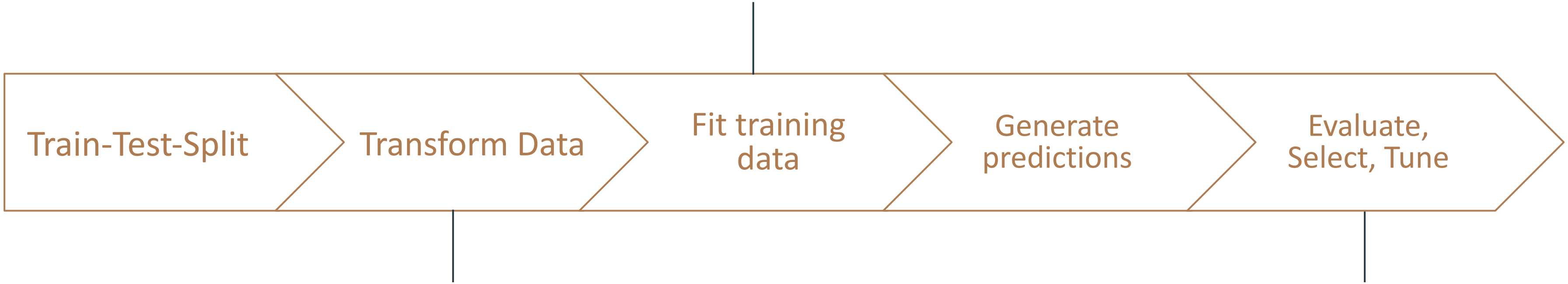
Generating models

- Logistic Regression
- Random Forest
- Multinomial Naive Bayes
- Support Vector Machine (SVM) classification
- Boosting

Fine tuning

Hyperparameter tuning of Vectorizers and Models

- Logistic Regression
- Random Forest
- Multinomial Naive Bayes
- Support Vector Machine (SVM) classification
- Boosting



Train-Test-Split

Transform Data

Fit training data

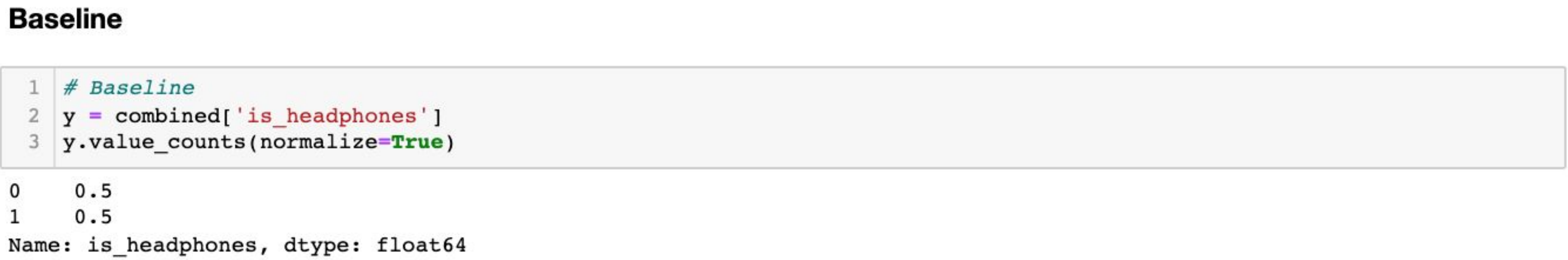
Generate predictions

Evaluate, Select, Tune

- Count Vectorizer
- TFIDF Vectorizer

- Evaluation metrics
- Select best model
- Tune hyperparameters (vectors + models)

Baseline score

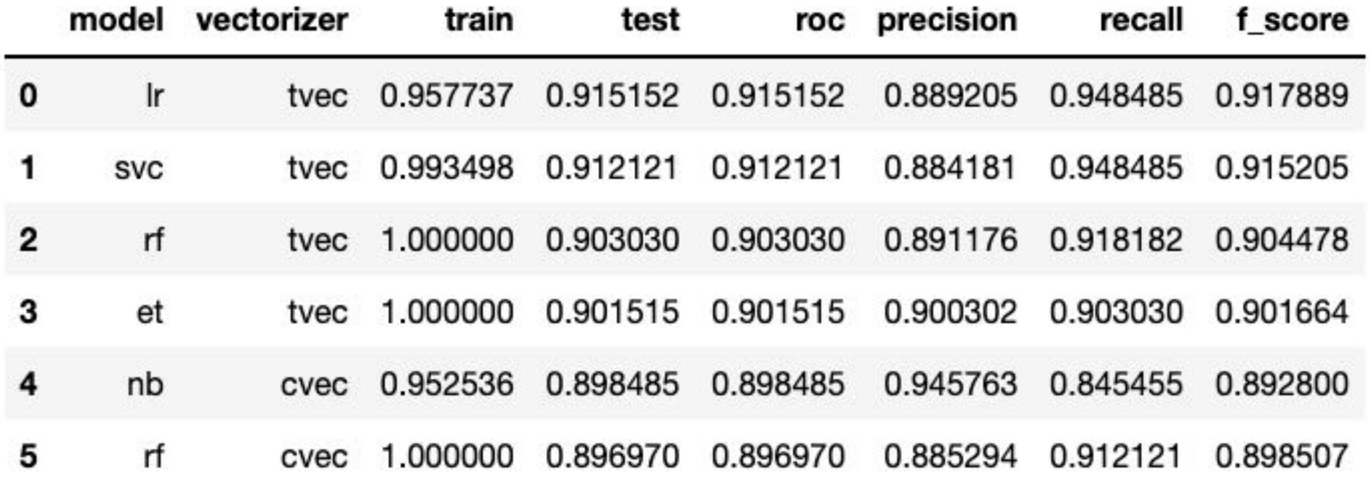


Baseline

```python
# Baseline
y = combined['is_headphones']
y.value_counts(normalize=True)
```

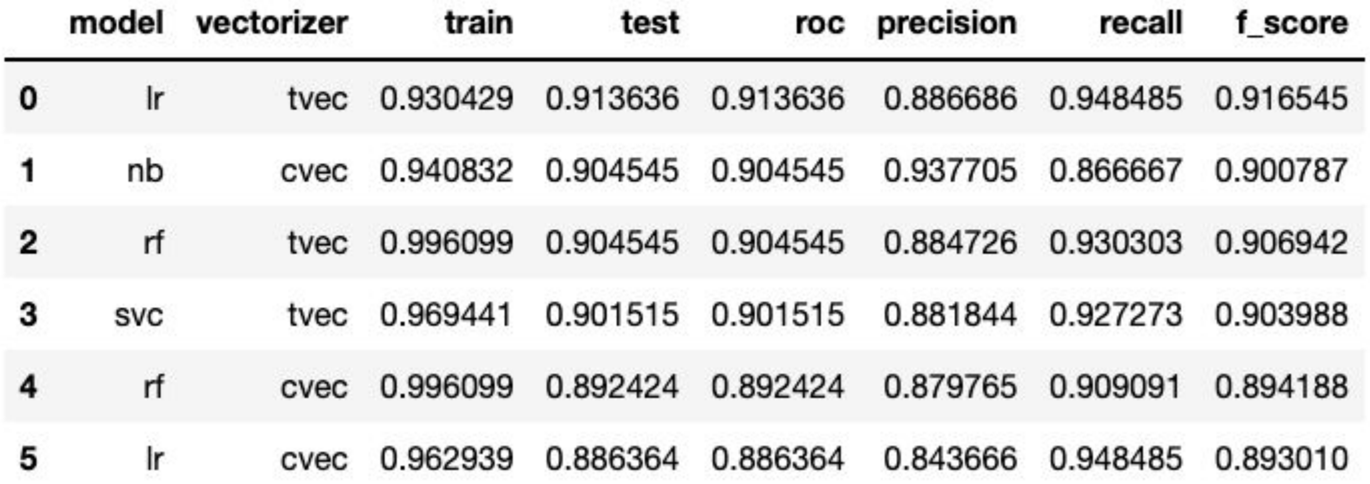
```
0    0.5
1    0.5
Name: is_headphones, dtype: float64
```

Preliminary Model Evaluation



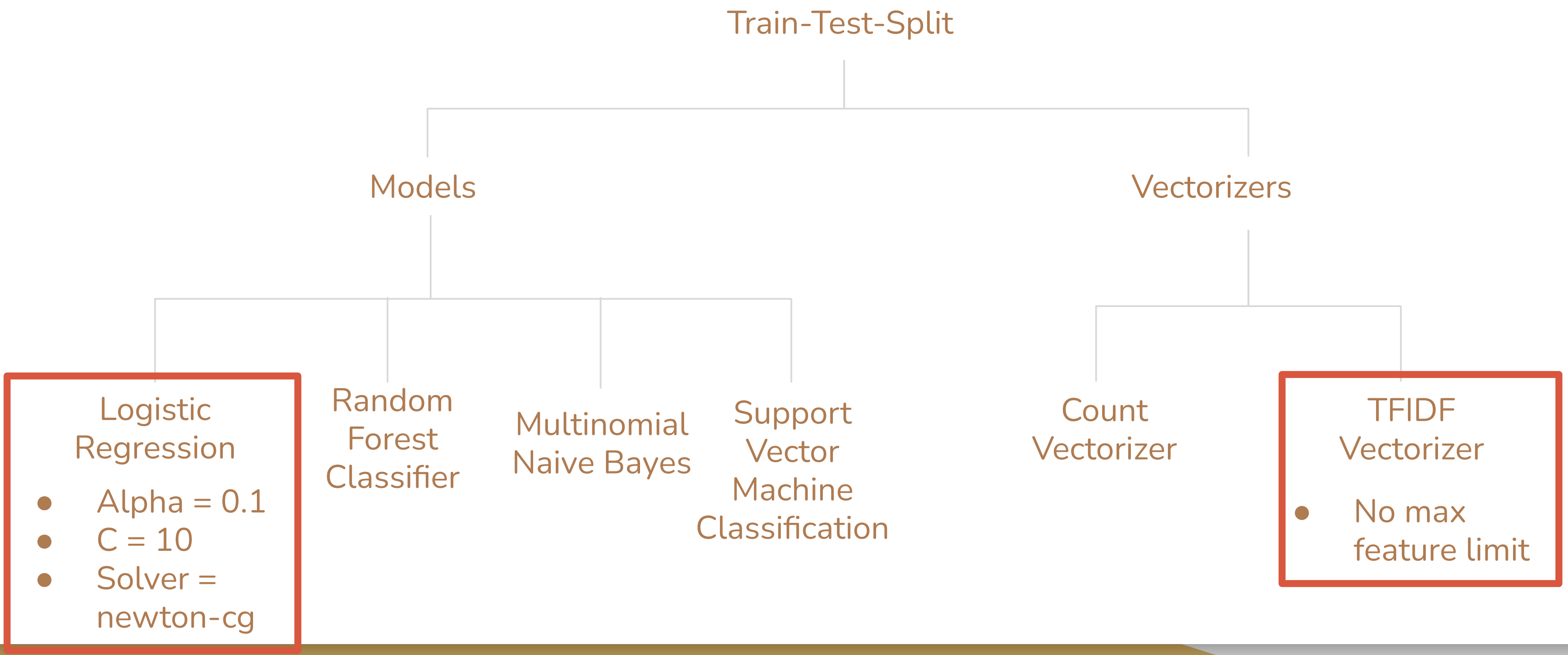
	model	vectorizer	train	test	roc	precision	recall	f_score
0	lr	tvec	0.957737	0.915152	0.915152	0.889205	0.948485	0.917889
1	svc	tvec	0.993498	0.912121	0.912121	0.884181	0.948485	0.915205
2	rf	tvec	1.000000	0.903030	0.903030	0.891176	0.918182	0.904478
3	et	tvec	1.000000	0.901515	0.901515	0.900302	0.903030	0.901664
4	nb	cvec	0.952536	0.898485	0.898485	0.945763	0.845455	0.892800
5	rf	cvec	1.000000	0.896970	0.896970	0.885294	0.912121	0.898507

Final Model Evaluation



	model	vectorizer	train	test	roc	precision	recall	f_score
0	lr	tvec	0.930429	0.913636	0.913636	0.886686	0.948485	0.916545
1	nb	cvec	0.940832	0.904545	0.904545	0.937705	0.866667	0.900787
2	rf	tvec	0.996099	0.904545	0.904545	0.884726	0.930303	0.906942
3	svc	tvec	0.969441	0.901515	0.901515	0.881844	0.927273	0.903988
4	rf	cvec	0.996099	0.892424	0.892424	0.879765	0.909091	0.894188
5	lr	cvec	0.962939	0.886364	0.886364	0.843666	0.948485	0.893010

Final Model Selected



Train-Test-Split

Models

Logistic Regression

- Alpha = 0.1
- C = 10
- Solver = newton-cg

Random Forest Classifier

Multinomial Naive Bayes

Support Vector Machine Classification

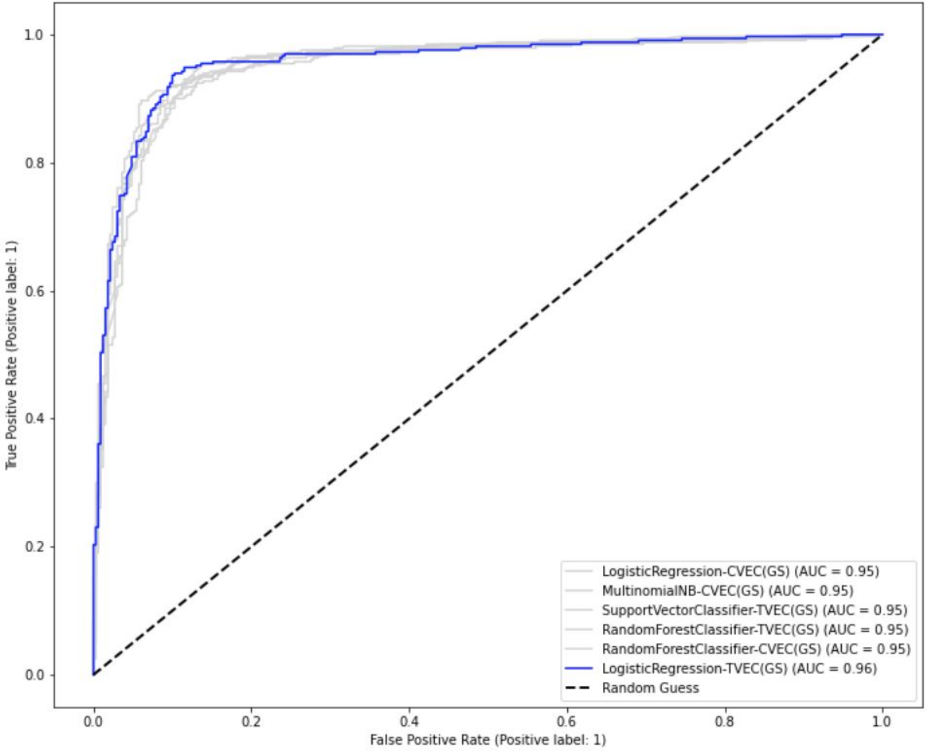
Vectorizers

Count Vectorizer

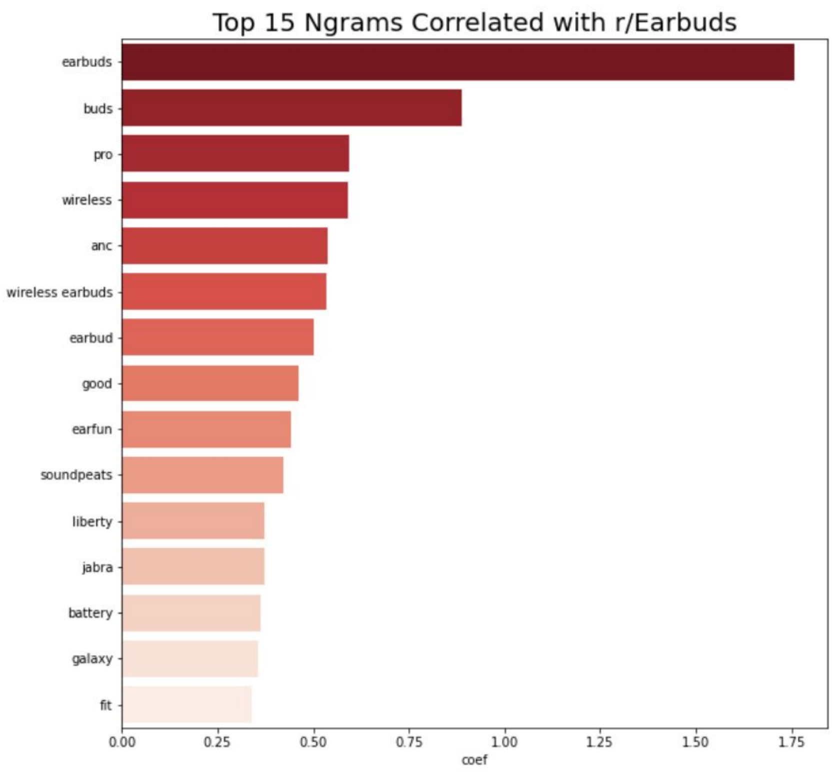
TFIDF Vectorizer

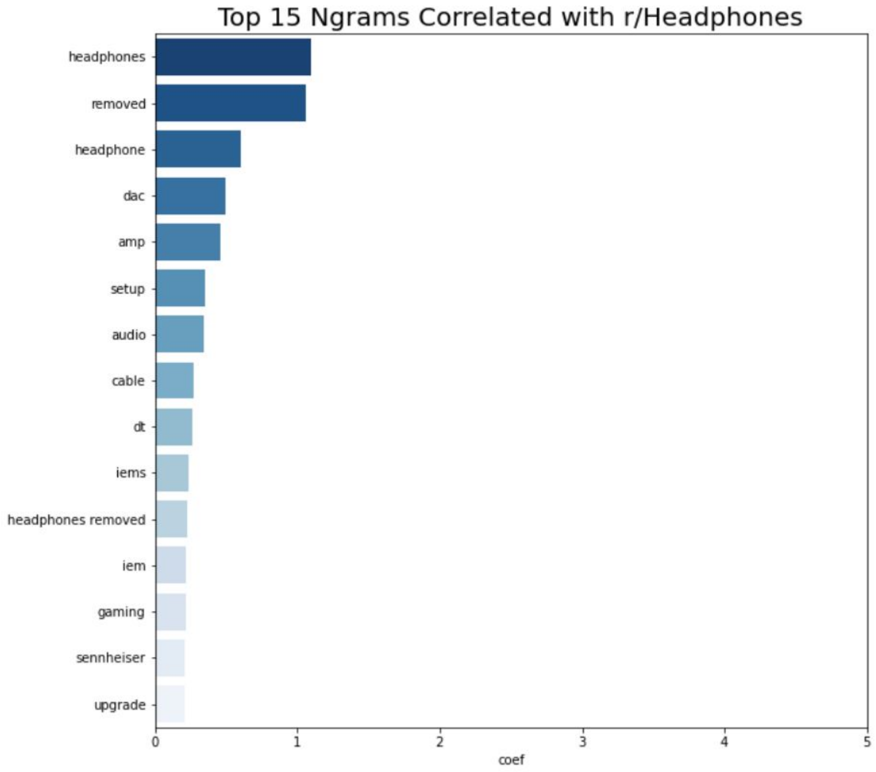
- No max feature limit

AUC-ROC Curve



Coefficient Visualisation





Limitations & Recommendations

- False positive/negatives
- Nature of dataset

- Other applications (Reddit moderators)
- Larger & more reliable dataset
- Include more stop-words