LDSSA

Hackathon #4
Text Classification

April 7th, 2024

Spacy Girls

Carmen Santana
Itai Soares
Gaspar Pereira
Inês Ribeiro

1. Problem description

Problem description

Classify song genre based on Lyrics

2. Workflow

2.1 Exploratory data analysis / Model Development

Exploratory Data Analysis

1. Slightly unbalanced data

Genre	
Rock	18993
Pop	11162
Hip Hop	8898

Train-test split with stratify

2. Hip Hop has more words, on average

Genre	
Hip Hop	509.433356
Pop	277.707311
Rock	207.418154

Feature added to model

3. Hip Hop uses more Adjectives

Genre	
Hip Hop	24.670038
Pop	13.236069
Rock	10.541357

Feature added to model

Features

Hip Hop has more swear words (average)

```
swear_words = [
    "fuck", "shit", "bitch", "asshole", "dick", "cunt", "bastard",
    "motherfucker", "cock", "piss", "twat", "ass", "damn", "hell",
    "bollocks", "arsehole", "wanker", "prick", "slut", "whore", "fucking", "mufuckas", "fuckin", "motherfuckin",
    "muthafuckin", "nigga", "niggas"]
```

Genre	
Hip Hop	7.405709
Pop	0.303530
Rock	0.235139

Baseline Model

```
F1 Score: 0.7585019748698412

Classification Report:
              precision    recall  f1-score   support

     Hip Hop       0.79      0.89      0.84      1593
         Pop       0.24      0.72      0.36       737
        Rock       0.96      0.67      0.79      5481

    accuracy                           0.72      7811
   macro avg       0.67      0.76      0.66      7811
weighted avg       0.86      0.72      0.76      7811
```

Preprocessing:

- Punct / stopword removal
- Vectorizer: Tfidf
- Features eng + StdScaler

Random Forest Classifier

Pop songs:
lowest f1-score

New Features

Pop has more romance words (average)

```
pop_specific_words = ['oh','love', 'kiss', 'baby', 'dance']
```

Genre	
Hip Hop	3.261295
Pop	4.035657
Rock	2.135313

New Features

Pop has a mode Positive sentiment!

using TextBlob

```
Sentiment Category Counts by Predicted Label (as percentages):
SentimentCategory  Negative   Neutral  Positive
PredictedLabel
Hip Hop            0.114691  0.595361  0.289948
Pop                0.080346  0.337454  0.582200
Rock               0.143103  0.429310  0.427586
```

Vectorizers

Doc2vec (100 feats)

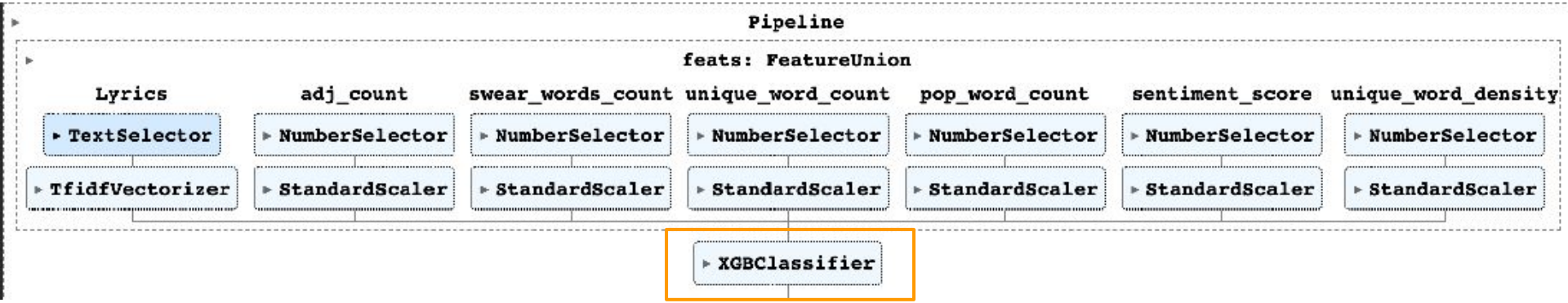
```
0     0.222456
1     0.264776
2     0.198353
3    -0.155944
4    -0.794881
        ...
95    0.301582
96   -0.132972
97    0.275249
98    0.554151
99    0.256146
Name: 0, Length: 100, dtype: float64
```

TfidfVectorizer

- ngram_range → 2
- min_df → 4

2.3 Results and discussion

Final Model


Pipeline
feats: FeatureUnion
Lyrics
adj_count
swear_words_count
unique_word_count
pop_word_count
sentiment_score
unique_word_density
TextSelector
NumberSelector
NumberSelector
NumberSelector
NumberSelector
NumberSelector
NumberSelector
TfidfVectorizer
StandardScaler
StandardScaler
StandardScaler
StandardScaler
StandardScaler
StandardScaler
XGBClassifier

Best Model Results

```
F1 Score: 0.7608700366287986

Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.90      0.86      1609
           1       0.90      0.73      0.80      4688
           2       0.44      0.65      0.53      1514

    accuracy                           0.75      7811
   macro avg       0.72      0.76      0.73      7811
weighted avg       0.79      0.75      0.76      7811
```

Test F1 score:

71,3 %

3. Future Work

Discussion

Our model *rocks* at identifying Rock.

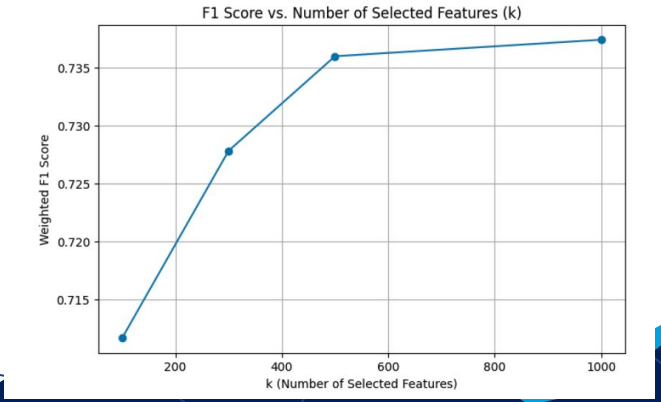
But it's confusing Pop with Rock quite often.

Future work

Feature Engineering
(e.g. number of singers in song)

Investigate Pop & Rock confusion

SelectKbest(500)



Do you wannabe a pop song?

But you rock!