johnodonnell123 / **Flatiron_School_Projects** Public

Code Issues Pull requests Actions Projects Wiki Security

main

Flatiron_School_Projects / Mod_5_Project / **readme.md**

**johnodonnellCP** Update readme.md **History**

**1** contributor

82 lines (59 sloc) 3.51 KB

# Module 5 Final Project

## Introduction

In this project, news headlines from 2012 - 2018 are classified into different categories using Natural Language Processing.

## Data:

The dataset was sourced from Kaggle and has ~ 200k news headlines that are to be classified into 1 of 40 categories. There dataset comes in the form af a JSON file and contains the following fields:

1. category:
    - 40 unique entries
    - Examples: POLITICS, TECHNOLOGY, PARENTING, WELLNESS
2. headline
3. author

4. short_description

5. link

## Methodology:

1. Data Cleaning & Organization:

   - The different text fields are explored and common formatting problems are corrected
   - All the fields are combined into a single 'text' field

2. EDA:

   - Several areas are explored including:
     - Category Prevalence (showing class imbalance)
     - Author Prevalence (showing which authors write the most articles)
     - Word Frequency (across all categories and within, before and after removing stop words)

3. Preprocessing:

   - Need to represent the text in a way that models can understand
     - Vectorizing (strings --> tokens --> vectors)
   - There are many ways to represent the text
     - Count Vectorizaiton (binary / counts)
     - Term Frequency / Inverse Document Frequency
     - Word Embeddings (custom vs pre-built models such as GLoVE)
   - Stop Words
   - Stemming & Lemmatization
   - n-grams

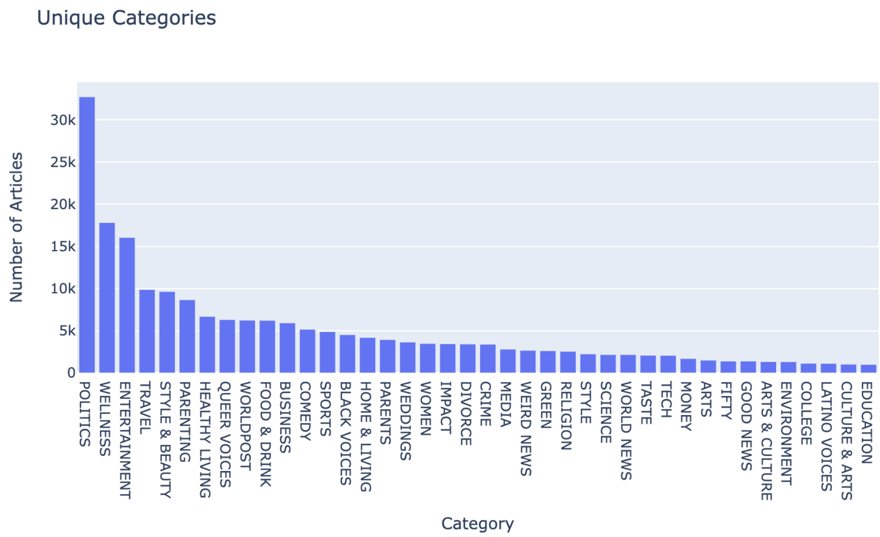
4. Modelling:

   - Many different models are built including Naive Bayes, Logistic Regression, Random Forest, and Neural Networks across these different pre-processing schemes.

## EDA Findings:

### Category Prevalence:

- There is a strong class imbalance, with Politics being the most prominent category




### Most common words in categories:

- After stopwords were removed, the most common words in each category are explored, and appear to make sense!

```
Category: CRIME
['us', '2015', 'unknown', '2016', 'police', '00', '2014', 'man', '2017', 'shooting']
Category: WEIRD NEWS
['us', '2016', '2015', '2017', 'lee', 'david', 'moye', 'moran', 'unknown', '2014']
Category: SPORTS
['us', '2015', '2016', 'unknown', '2014', '00', 'contributor', 'game', 'nfl', '0000']
Category: RELIGION
['us', 'contributor', '2015', '2014', '2016', 'unknown', 'antonia', 'blumberg', '2017', 'not']
Category: PARENTS
['us', 'contributor', '2017', '2014', '2015', '2016', 'kids', 'caroline', 'bologna', 'mom']
Category: HEALTHY LIVING
['contributor', 'us', '2014', '2015', '2017', '2016', 'health', 'author', 'life', 'not']
Category: WELLNESS
['us', 'contributor', '2013', '2012', 'unknown', 'author', 'life', 'not', 'health', '00']
Category: WEDDINGS
['us', 'wedding', 'unknown', '2013', '2012', 'contributor', 'weddings', 'marriage', '00', 'day']
```

### Custom embeddings appear strong:

- Custom word embeddings are created using the articles, and the results (while not perfect) appear to capture some of the semantic meanings

```
senator ['sen', 'senate', 'senators', 'rep', 'sessions', 'ted', 'rnc', 'representative', 'rubio', 'sanders']
son ['sons', 'sister', 'teen', 'sisters', 'siblings', 'screaming', 'seemed', 'scream', 'separated', 'toddler']
daughter ['daughters', 'dad', 'child', 'dear', 'crying', 'father', 'grandmother', 'decided', 'fatherhood', 'daddy']
business ['ceo', 'companies', 'consumer', 'build', 'corporate', 'consulting', 'connections', 'company', 'clear', 'changing']
healthy ['healthier', 'foods', 'healthily', 'intensity', 'habits', 'huffposthealthyliving', 'hormones', 'gluten', 'ingredients', 'grain']
technology ['tech', 'software', 'startup', 'startups', 'technologies', 'smartphones', 'tools', 'smartphone', 'resource', 'se']
```

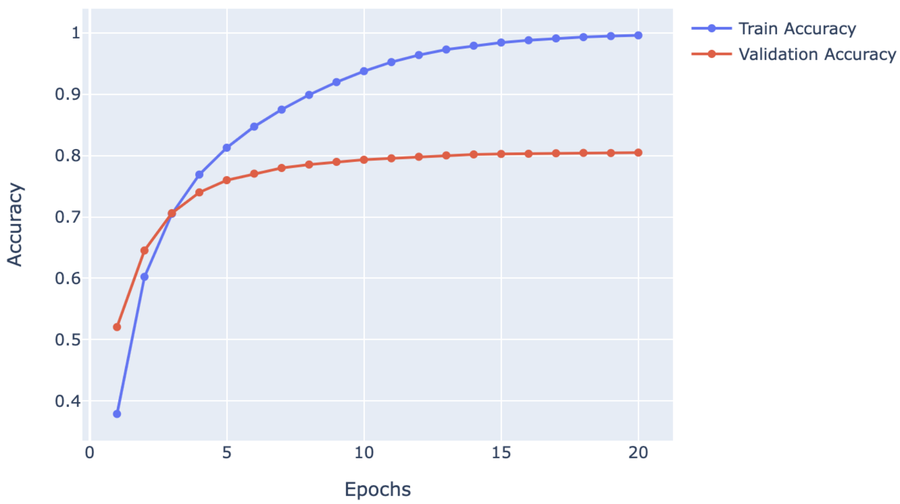
## Model Results

- The two best models both used tri-grams without any stemming, lemmatization, or word embeddings. The best model was a neural network with an accuracy score of ~ 80%, and the second best was a Naive Bayes classifier with ~78% accuracy. The

NN has stronger performance, but is not as directly interpretable as the NB classifier!

- Note the gap in accuracy between the validations and test set, there is clear evidence of overfitting here, and some regularization should be attempted.

## Neural Network Results:




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.74 | 0.80 | 0.77 | 315 |
| 1 | 0.85 | 0.72 | 0.78 | 283 |
| 2 | 0.76 | 0.59 | 0.66 | 909 |
| 3 | 0.72 | 0.68 | 0.70 | 1242 |
| 4 | 0.62 | 0.56 | 0.59 | 227 |
| 5 | 0.74 | 0.72 | 0.73 | 1037 |
| 6 | 0.63 | 0.66 | 0.65 | 676 |
| 7 | 0.72 | 0.69 | 0.70 | 189 |
| 8 | 0.87 | 0.85 | 0.86 | 637 |
| 9 | 0.65 | 0.62 | 0.63 | 206 |
| 10 | 0.84 | 0.84 | 0.84 | 3205 |
| 11 | 0.75 | 0.62 | 0.68 | 283 |
| 12 | 0.66 | 0.61 | 0.64 | 290 |
| 13 | 0.89 | 0.93 | 0.91 | 1239 |
| 14 | 0.71 | 0.53 | 0.61 | 275 |
| 15 | 0.59 | 0.64 | 0.61 | 498 |
| 16 | 0.68 | 0.81 | 0.74 | 1327 |
| 17 | 0.90 | 0.90 | 0.90 | 848 |
| 18 | 0.57 | 0.50 | 0.53 | 661 |
| 19 | 0.91 | 0.55 | 0.69 | 262 |
| 20 | 0.74 | 0.61 | 0.67 | 563 |
| 21 | 0.73 | 0.76 | 0.75 | 334 |
| 22 | 0.83 | 0.85 | 0.84 | 1716 |
| 23 | 0.75 | 0.78 | 0.76 | 790 |
| 24 | 0.85 | 0.89 | 0.87 | 6430 |
| 25 | 0.89 | 0.84 | 0.86 | 1270 |
| 26 | 0.75 | 0.64 | 0.69 | 521 |
| 27 | 0.79 | 0.66 | 0.72 | 440 |
| 28 | 0.84 | 0.85 | 0.84 | 986 |
| 29 | 0.83 | 0.80 | 0.81 | 468 |
| 30 | 0.93 | 0.93 | 0.93 | 1963 |
| 31 | 0.86 | 0.80 | 0.83 | 414 |
| 32 | 0.73 | 0.70 | 0.72 | 417 |
| 33 | 0.87 | 0.90 | 0.88 | 1902 |
| 34 | 0.89 | 0.85 | 0.87 | 743 |
| 35 | 0.56 | 0.66 | 0.61 | 557 |
| 36 | 0.87 | 0.91 | 0.89 | 3629 |
| 37 | 0.63 | 0.60 | 0.61 | 694 |
| 38 | 0.61 | 0.59 | 0.60 | 441 |
| 39 | 0.73 | 0.73 | 0.73 | 1281 |
| accuracy | | | 0.80 | 40168 |
| macro avg | 0.76 | 0.73 | 0.74 | 40168 |
| weighted avg | 0.80 | 0.80 | 0.80 | 40168 |

## Naive Bayes Results:

```
              precision    recall  f1-score   support

           0       0.83      0.68      0.74       315
           1       0.86      0.63      0.73       283
           2       0.72      0.62      0.67       909
           3       0.69      0.69      0.69      1242
           4       0.76      0.52      0.62       227
           5       0.67      0.71      0.69      1037
           6       0.56      0.73      0.63       676
           7       0.82      0.61      0.70       189
           8       0.82      0.85      0.84       637
           9       0.70      0.51      0.59       206
          10       0.79      0.85      0.82      3205
          11       0.85      0.43      0.57       283
          12       0.87      0.68      0.76       290
          13       0.85      0.90      0.87      1239
          14       0.81      0.48      0.60       275
          15       0.58      0.62      0.60       498
          16       0.69      0.67      0.68      1327
          17       0.89      0.89      0.89       848
          18       0.54      0.57      0.55       661
          19       0.91      0.50      0.65       262
          20       0.76      0.58      0.66       563
          21       0.80      0.68      0.73       334
          22       0.71      0.82      0.76      1716
          23       0.74      0.67      0.70       790
          24       0.84      0.87      0.86      6430
          25       0.83      0.83      0.83      1270
          26       0.76      0.64      0.70       521
          27       0.80      0.64      0.71       440
          28       0.81      0.85      0.83       986
          29       0.87      0.66      0.75       468
          30       0.88      0.94      0.91      1963
          31       0.88      0.65      0.75       414
          32       0.78      0.60      0.68       417
          33       0.81      0.91      0.86      1902
          34       0.87      0.81      0.84       743
          35       0.60      0.55      0.58       557
          36       0.82      0.89      0.85      3629
          37       0.61      0.49      0.55       694
          38       0.61      0.43      0.50       441
          39       0.69      0.74      0.72      1281

    accuracy                           0.78     40168
   macro avg       0.77      0.69      0.72     40168
weighted avg       0.78      0.78      0.78     40168
```

## Forward:

- Modelling:
  - The gap between training and validation accuracy in the NN points towards overfitting, and some attempts at regularization can be made.
  - RNNs and LSTM can also be built to try to improve accuracy
- Reframing the problem:
  - Instead of treating these labels and categories, we could treat them as tags, assigning 2-3 to each article
  - This will likely increase the accuracy of the predictions, and also provide more value to the user audience