News Article Categorization

Authors: Kyle Vosen, Chandler O’Neal,
Jordan Johnson

Overview

- **Original Data:** Contained 5 features, the target variable, and 200,853 news articles; these were then reduced to one feature, the concatenated headline and the description, the target variable, and 200,853 articles.
    - Target Variable: the categories of the articles.
    - Features: the variables that predicted the categories.

Business Problem

- **The Organization:** Huffington Post had found it necessary to add a model to their toolbelt that would help them to organize yet to be categorized articles.
- **Metric Used:** F-1 Score

- **Reasoning:**
    - F1 is useful with uneven class distribution - takes the weighted average of the precision and recall score.
    - It was not necessary to weigh false negatives or false positives more heavily than one another.
        - *False Negative* - incorrectly predicting the returned categorization to be inaccurate.
        - *False Positive* - incorrectly predicting the returned categorization to be accurate.

Data Used

- **Original Dataset**: News Category Dataset from The Huffington Post.
- **Post Cleaning:** 1 feature which was the combination of the headlines and descriptions from the articles, and the target variable (categories of the articles).
    - **Total News Articles:** 200,853.

Cleaning and Preprocessing

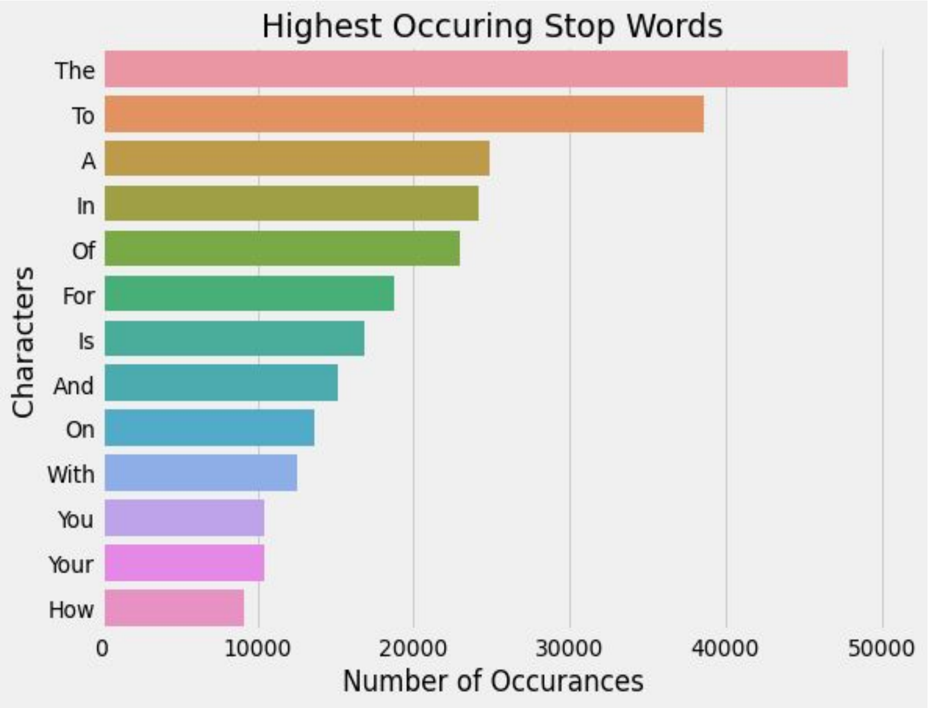
To Reduce Collinearity and Remove Unnecessary Columns

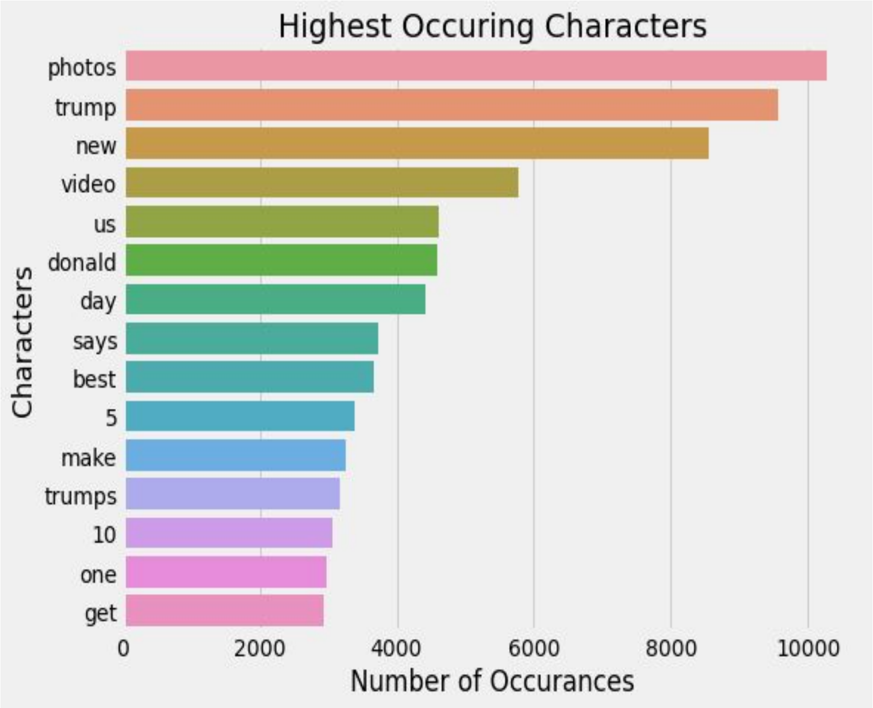
- We began with 41 different categories
- Grouped down to 12 categories
- Dropped authors, link and date as they were unnecessary for our predictions
- Combined headline and sh
- ort description

To Increase Model Understanding:

- Lowercasing words
- Removing stopwords
- Removing punctuation
- Removing special characters

Reasoning Behind Removing Stop Words

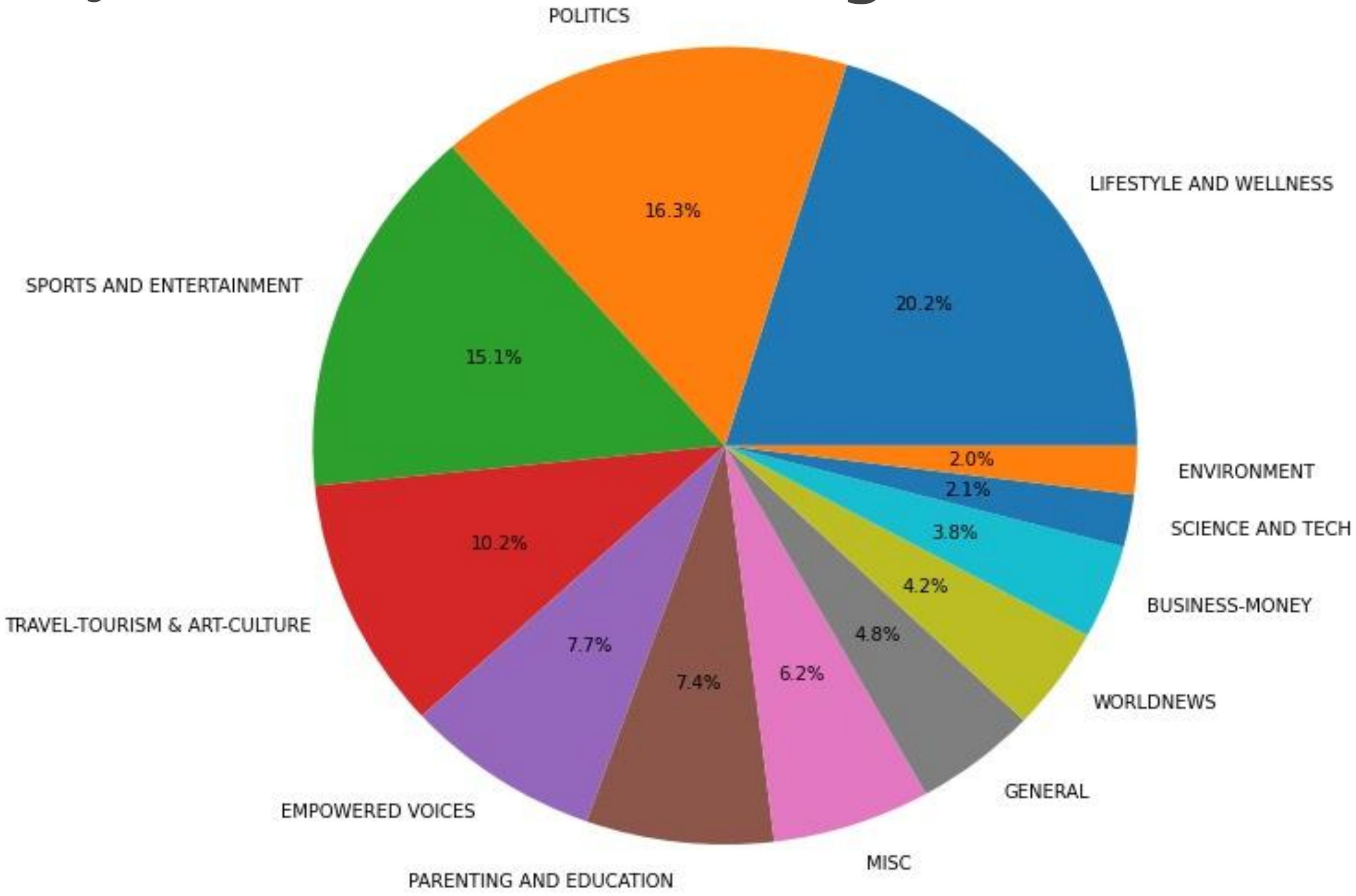




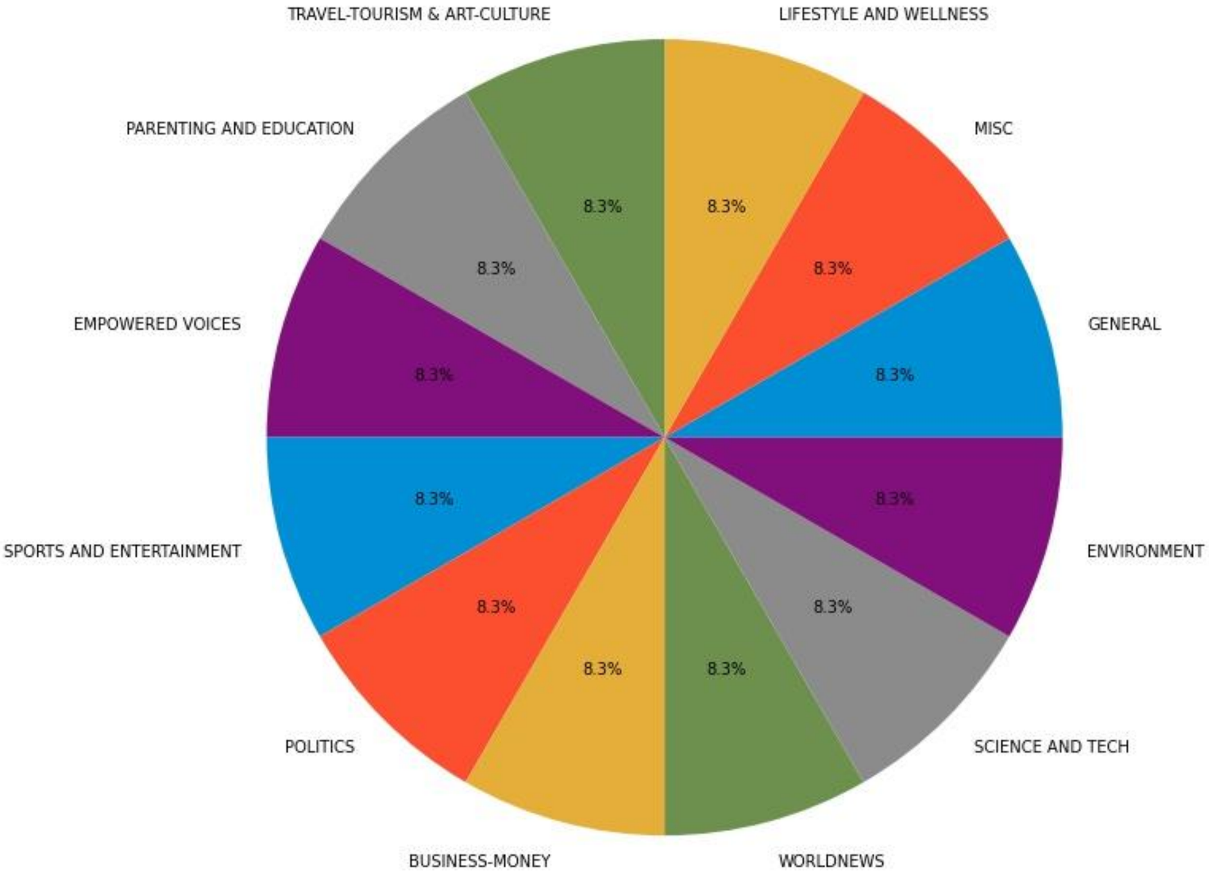
Methods used

- **Feature Engineering**
  - Tokenization - Breaking all sentences into individual words.
  - Lemmatization - Taking the root of the word and getting rid of end: ex(-ing)
  - Vectorization - Turning words into numerical values
  - SMOTE - Filling unbalanced data with randomly selected data from the dataset.
- **Modeling**
  - ***Final Model***: Logistic Regression
  - Decision Tree Classifier
  - Naive Bayes
  - Pipeline Model
  - Random Forest

Category Balance Pre-Smoting



Category Balance Post-Smoting



Model Results

- Our best model was a Logistic Regression using 1000 max iterations and a random state to ensure reproducibility.
- The model resulted with an F-1 score of 66.9% on the train data.
- F-1 score of 62% on the test data

Conclusions

- **Prediction Outcome:** The model was able to predict the category of each article description with 62 % accuracy with a slight overfit on the training data - 66 % F1-score.
- **Usefulness:** While the model accuracy was not substantially high, it would prove to be a meaningful model in predicting news article categories for Huffington Post.

Next Steps

- **Features:**
  - To add a new feature that would account for the percentage of the different parts of speech in each description to improve model accuracy.
  - Create a model that takes in user input and outputs the category
- **Reduce Overfit:** To reduce model bias and overfitting by using a Grid Search CV to establish the best hyperparameters.
- **Host on Web Domain:** To host the final logistic regression model on a website domain.

Contact

Kyle Vosen:

Email: kylevosen1999@gmail.com

Github: krvosen

Chandler O'Neal:

Email: jchandleroneal@gmail.com

Github: jchandleroneal

Jordan Johnson:

Email: johnsonjordan556677@gmail.com

Github: Jorno1