Truth is Stranger than Fiction

Revathi Satkuna

Problem Statement

Truth is stranger than fiction. The ENG105MISC class, Creative Writing in Tech, at the local community college wanted to visualize the intersection in English writing and technology. The professor wanted to demonstrate how language can be generated by AI and in tune how AI can classify different types of text. To help with their learning, you wanted to develop a Classification model to show the students how your computer, with the help of a coding language like Python, can predict where a text can come from. To create the model, and the type of model in question is either going to be a RandomForestClassifier or LogisticRegression, you use the r/creativewriting and r/talesfromtechsupport subreddits to classify whether a phrase would belong in the Creative Writing subreddit or in the Technical Support subreddit.

Cleanup/EDA

- I first used Pushshift's API to collect posts from the r/creativewriting and the r/talesfromtechsupport subreddits. I then cleaned the data and then perform a binary classification by using Natural Language Processing to train a classifier on which subreddit a given post came from.
- I cleaned the data by selecting relevant features from the raw data and removing null values and creating new relevant columns via column transformations.
- The most relevant columns were the 'selftext' and the 'title' columns, and I combined those to create a 'fulltext' column.
- I also assigned a binary value 0 or 1 to the creative writing subreddit posts and the tech support subreddit post in a new column called 'subreddit' so that we can use for Classification modeling later.

NLP

- To prepare the cleaned data for Natural Language Processing, I first used a tokenizer on the fulltext column and then lemmatized and stemmatized it to better prepare it for sentiment analysis and NLP.
- I then wanted to perform either a CountVectorizer or a TfidfVectorizer to see which one had a better score.
- I then compared the best scores between a Pipeline with a TfidfVectorizer running into a Multinomial Naive Bayes and then setting up a BayesSearchCV, and then doing the same but with a CountVectorizer instead.
- TfidfVectorizer had a higher best score, so I used its best parameters it gave for max features.

Modeling

- I used the 'stemmatized_lemmatized_tokenized_fulltext' column as our feature and the 'subreddit' column as our target.
- I then created and compared two models, a Random Forest Classifier and a Logistic Regression Classifier. From there, I compared the baseline score to the accuracy scores of the models, the higher the accuracy score the more successful you have trained your computer to predict whether a phrase corresponds to the Creative Writing and Technical Support subreddits. Since the Logistic Regression model had the highest score, we used it for our models.

Modeling

- I visualized the models by displaying the top positive and negative correlations of the features, corresponding to the Technical Support and Creative Writing subreddits, respectively.
- I also plotted the confusion matrix and its metrics to show how effective the model is at predicting whether a set of words will belong into either subreddit.

Selecting Classification Model

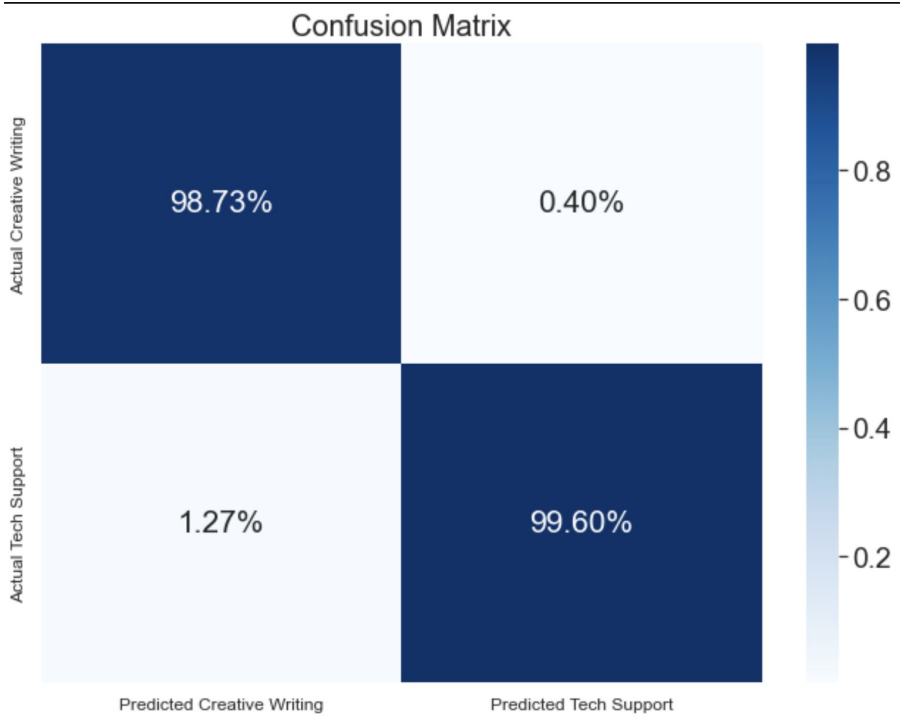
- The Logistic Regression model had an accuracy of 99.4% for the training data and 99.1% for the testing data, and the Random Forest Classifier had an accuracy of 99.8% for the training data and 98.6% for the testing data. Since the testing data scored higher in our Logistic Regression model, I went forward to do the analysis on the Logistic Regression model.

Baseline

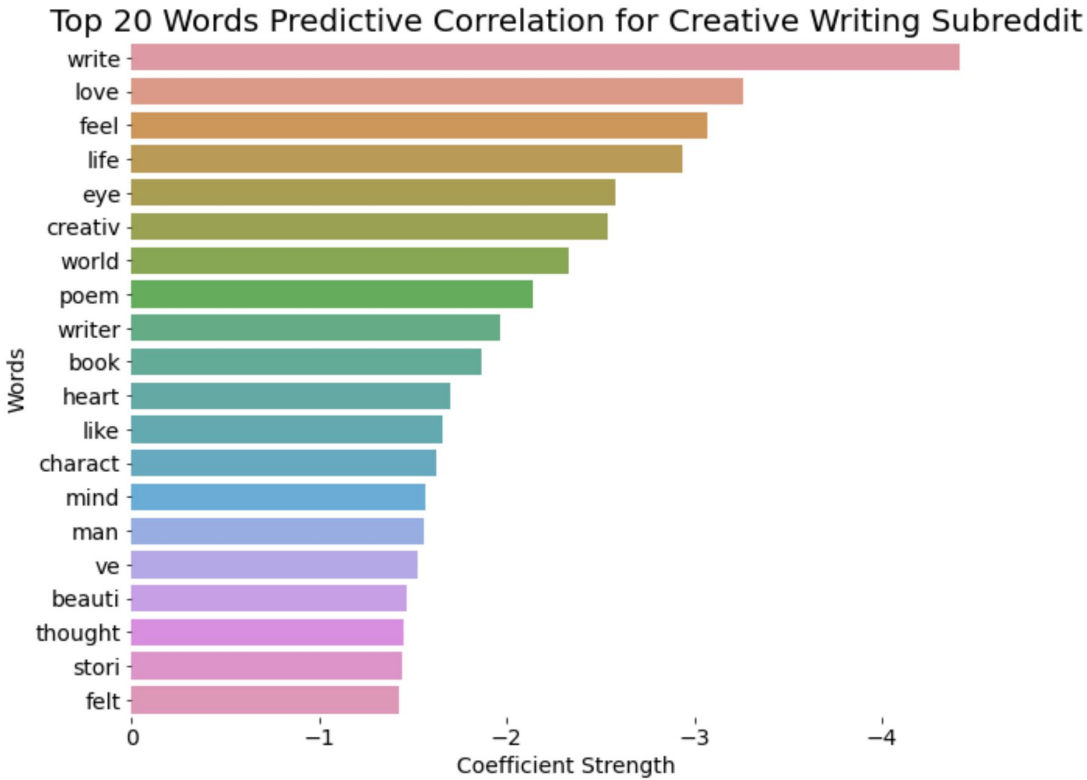
- For calculating for whether the text post will be predicted to come from the creative writing subreddit or the tech support subreddit, if you predict that all the posts came from the creative writing subreddit, you would be correct about 58.2 % of the time.
- This is because we had more posts from the creative writing subreddit. For the tech support subreddit, if you predicted that all the posts came from it, you would be right about 41.7% of the time.
- Since the true positive of this classification matrix would correspond to the tech support reddit, the **baseline accuracy** would be 41.7%

Confusion Matrix

- After analyzing the confusion matrix generated with the Logistic Regression model, the accuracy was 99.1%. The sensitivity was 99.6%. The misclassification rate was 0.9%. The specificity is 98.7%. The precision is 98.2%.



Top 20 Words Creative Writing Subreddit



Top 20 Words Tech Support Subreddit