Sentiment Analysis of the Roman Urdu Language

Steven Dye

May 12, 2020

Scope of Project

Sentiment Analysis is a critical tool for any technology company. It allows them to better understand the opinions their customers have about them and their products. While large strides have been made in this field, this progress is language dependent with more popular, more established languages receiving the majority of the resources. This creates a need for similar analysis in less popular languages, as the views of customers who speak these languages are often missed or ignored. It is in the best interest of companies to know the views of these customers, as it leads to better insight into that particular marketplace. This project aims to accurately predict the sentiment of the text of an under-resourced language, with a special focus on negative sentiment.

Data and Data Preparation

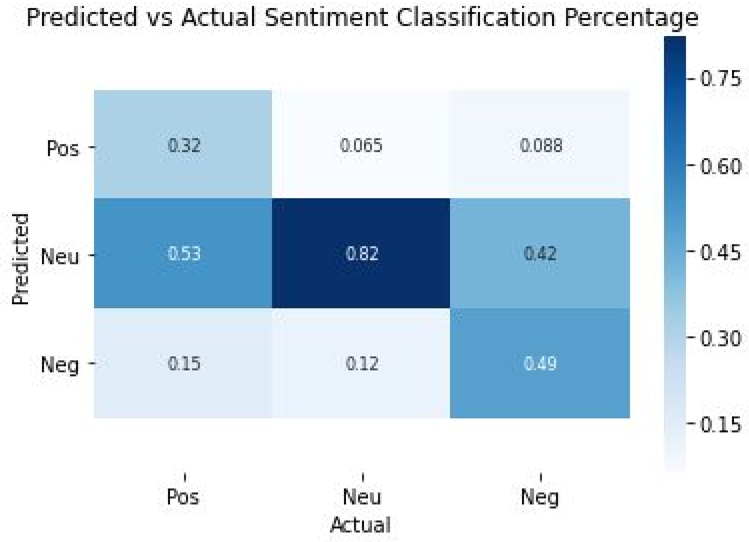
The data used in this project was the Roman Urdu data set found here: https://archive.ics.uci.edu/ml/datasets/Roman+Urdu+Data+Set. It is a collection of data from a variety of sources with a sentiment label attached to it. The data was cleaned and the text was vectorized with the Tfidf vectorizer. The sentiment imbalance was leveled out by creating synthetic data using the SMOTE method. The Pipeline function from the imbalanced learn library. The final data output resulted in two different outputs: one with a hard limit on vectorized features and one without a limit. The unlimited dataset contained 26,603 features, while the limited dataset contained the 20,000 features that had the most counts. The motivation for this was to remove misspelled words that would only show up a handful of times. Limiting the dataset this way reduces the size of the datafile by 24.8%, while keeping 95.2% of the data. While the final result in this analysis uses the unlimited dataset, it is only done so due to the scarcity of the data. In larger projects, in order to reduce computation costs, reducing the datafile size as described above is recommended.

Modeling

Of the five models that were considered (naive bayes, logistic regression, SVM, random forest, and XGBoost), XGBoost has shown to have the most consistent performance. When considering larger projects, naive bayes could be reasonably considered as it takes a significantly shorter amount of time to train it in comparison to the other models, at the cost of consistency. The XGBoost model was created with python’s sklearn library. In tuning the hyperparameters, the max depth was set to 7 and the minimum child weight was set to 0.5.

Results

The final training accuracy score is 0.6721, while the final testing accuracy score is 0.5877. This implies that the model is overfitted. The mean of the cross validation values scored on accuracy is 0.5865, and is within 0.16 standard deviations of the testing score. A breakdown of the distribution of predictions is provided on the confusion matrix below.



Data Limitations

Because Roman Urdu is an under-resourced language, many popular techniques used in sentiment analysis are not available. The most common technique is to remove stop words from the analysis. While a list of stop words is not readily available for the Roman Urdu language, a list has been found at this website: https://github.com/haseebelahi/roman-urdu-stopwords/blob/master/stopwords.txt. Note, it is impossible to know if this list is correct or complete without domain knowledge in the Roman Urdu language. Another common technique is to stem or lemmat words to a root word in order to reduce dimensionality of the data. While in theory this can be done by hand, domain knowledge of the language is required to do so. Correcting spelling mistakes in the analyzed texts also produces the same benefits, and faces the same challenges.