Final Project Report

Collaborators

1. Saran Chowdary Malakalapalli
2. Christian Brough
3. Lakshmi Madhira
4. Poornima Sharm
5. Chandana Mallu

There were two main tasks that our group had when we started our final project. The first was to create a machine learning model that would be able to classify a comment as a certain class that I will talk about later.

The second task had two different options, we could either choose to find the sentiment for the comment or to see if the comment was either a suggestion or opinion. We chose to classify if the comment was a suggestion, opinion, other, or both. I will also talk more about those labels later on in the report.

For our first task, we had to separate the different comments into separate types of classifications. The classifications that we used were transportation, data governance, connectivity, cybersecurity, public safety, environment, public utilities, agriculture, Smart tech (iot), community, education, and other. We felt like including all of these would allow the model to have a better ability to assign a certain classification to any of the comments. The annotation section of the final project was the hardest and took the longest amount of time because that is the part of the project that we could not just write code to predict what the correct classification would be.

When we created the model for task one, we wanted first to establish a baseline level of accuracy that we could compare everything else to. A training and testing split of 80%/20% was used for all of our models. The first model that we used was a simple logistic regression which had an accuracy score of 0.025, precision of 0.0195, recall of 0.1278, and a macro f1 of 0.0324. We were able to use this as a baseline where we can see if we are progressing with the models or regressing.

The next model that we tried was a OneVsRestClassifier Linearsvc with a term frequency-inverse document frequency with an ngram range of (1, 3). This model gave us a result where accuracy was 0.075, precision 0.0752, recall 0.2972, and the macro f1 was 0.1136.

Next we tried the OneVsRestClassifier Linearsvc with term frequency-inverse document frequency but without the ngram range this time. This actually proved to be a decent amount more accurate than the same model with a ngram range. This model produced an accuracy of 0.12, a precision of 0.1352, a recall of 0.3779, and a macro f1 of 0.1849.

Our final two models were almost identical but one was just slightly better performance than the other. The first of these two was a Linear SVC which had countvectorizer and a ngram range of (1, 2). This was letting us be able to use bigrams and unigrams to help train the model. This model was able to have a final accuracy of 0.145, a precision of 0.146, a recall of 0.389, and a macro f1 of 0.1991.

The final model that performed slightly more accurately than the previous one was also a Linear SVC that had countvectorizer, but this time only a ngram range of (1, 1). We wanted to see how only being able to use unigrams would help us in the modeling of this problem. Changing the ngram range allowed a very very slight increase in precision and recall. This amount is small enough though where it could change with the addition of more data. The accuracy of this model was 0.145, the precision was 0.1467, the recall was 0.3897, and the macro f1 was 0.1991.

This model also brought up an interesting problem with the data and using machine learning with this data. We are working with such a small dataset of only 800 training comments and there are a total of 12 different classifications. This could introduce issues when rare classifications are not getting predicted correctly because there just is not enough information about them to classify. We believe that this model could become much more accurate with more data. The second task of the final project was to predict if the comment was an opinion, suggestion, both, or neither. Here we used slightly different models, with the first one being a logistic regression that utilized term frequency-inverse document frequency. This model had a validation of 0.3087, a precision of 0.3111, a recall of 0.3134, and a macro f1 of 0.309.

The next model that we tried was a Linear SVC that also had a term frequency-inverse document frequency. This model was able to attain a validation of 0.3087, a precision of 0.3139, a recall of 0.3163, and a macro f1 of 0.3089. This model gave us a slightly higher precision, but it also gave us a slightly lower recall and macro f1.

The following model was a Linear SVC with countvectorizer to get features for the model. This model had a validation of 0.3517, a precision of 0.3864, a recall of 0.3928, and a macro f1 of 0.3846. We then compared what would happen if we did the exact same thing by including the countvectorizer, but just changed the model to a logistic regression. The logistic regression model with countvectorizer had a validation of 0.357, a precision of 0.3711, a recall of 0.38, and a macro f1 of 0.3723.

The final group of models that we tried for our task two were a logistic regression with ngram range of (1, 2) and a Linear SVC with ngram range of (1, 2). For the logistic regression, the model had a validation of 0.3665, a precision of 0.3771, a recall of 0.38, and a macro f1 of 0.3723. For the Linear SVC, the model achieved a validation of 0.3665, a precision of 0.3335, a recall of 0.3408, and a macro f1 of 0.3261.

The final model that we would present for our task two would be the logistic regression with a ngram range of (1, 2) because, while it had the same validation rate as the Linear SVC which included the same ngram range, the logistic regression had a higher macro f1 score which means that we are predicting a comment correctly more often in the logistic regression. We wanted to use the f1 score at the end in addition to validation to choose which model should be the final model because we are trying to achieve a high true positive rate and a high true negative rate.