**Some ideas for the paper:**

**For the implementation section:**

The implementation includes the use of a countVectorizer to create Bag-of-Words with a maximum of 5000-most frequently used words. This is done to avoid dataset sparsity and thus remove noise. Using countVectorizer, text data is converted to numerical data which can further be mapped into output variables. A multi output classifier is used since each sentence has to be classified as one of the six output variables.

**For the part where we discuss different types of output/labels possible:**

Multi label classifier wrapper from sklearn library is sued to create a model to label the six output variables. The output variable distribution is as shown below:

Chart, waterfall chart

Description automatically generated

Hence, it can be noticed that there is class imbalance in all the training target variables.

**More for the pipeline part of things:**

Both text and tag are used for feature extraction. Once the features are obtained, they are passed through the Multinomial Naïve Bayesian Model to perform multi-output text classification. This process is shown in the diagram below:

Diagram

Description automatically generated

Credit to the picture shown below: <https://www.kaggle.com/anirbansen3027/jtcc-bag-of-words>

**More on evaluation metrics:**

To evaluate the performance of the implementation, we have used ROC – AUC metrics. ROC stands for Receiver Operating Characteristic. ROC is a graph showing the performance of the classification. AUC stands for Area under the Curve. AUC measures the entire two-dimensional area under the ROC curve. The ROC-AUC score indicates the degree of separability/distinctness or interminability/crossover between the predictions. Higher the ROC-AUC score, lower the crossover between the labels and thus better the classification.

Here is the link I used for the above information. You can use this in the reference section.

<https://towardsdatascience.com/roc-and-auc-how-to-evaluate-machine-learning-models-in-no-time-fb2304c83a7f>

Our model was able to achieve an accuracy of 81.38% and a mean AUC value of about 0.6319.

Further, probability predictions are done on the test data and dumped into a CSV submission file.

**Comparing our model to other implementations:**

A similar implementation shown in this link was able to achieve about 89.81% accuracy and a mean AUC value of about 0.94. (This is from the same link as the figure above, NOT SURE IF YOU WANT TO PUT THIS SINCE OUR ACCURACY ISN’T CLOSE TO WHAT THIS GUY WAS ABLE TO ACHIEVE. I’LL LEAVE IT TO YOU TO MAKE A DECISION).

**Our next plans for the project:**

Our plan for the next half of the project is to be able to implement the pipeline once again but with a different word representation and a different classification model. At the end of implementing another model, we want to be able to compare our results between the two pipelines.

**Please add the GitHub link to the code in the end.**

This is the link I referred to when developing my code:

<https://www.kaggle.com/anirbansen3027/jtcc-bag-of-words>