**PROJECT PLAN 373**

**1. Students’ name and Purdue e-mail. The project is to be done in groups of 3 students.**

Aditya Kotalwar - [akotalwa@purdue.edu](mailto:akotalwa@purdue.edu)

Alisha Lakkad - [alakkad@purdue.edu](mailto:alakkad@purdue.edu)

Rushank Jhanjaria - [rjhanjar@purdue.edu](mailto:rjhanjar@purdue.edu)

**2. [Up to 3 lines] Definition of the problem, possibly relevant to your interests.**

Often, when an influencer posts their opinion or announcement on twitter on a particular topic, there is a wave of both positive and negative comments that follow this tweet. We want to make a model that does sentiment analysis to give the average public opinion following these tweets and labels the sentiment as positive, negative or neutral.

**3. [Up to 3 lines] Description of the dataset (or datasets) to be used. Datasets should be already publicly available, since there is not enough time for you to collect data. For possible datasets, see the course webpage.**

Sentiment140 dataset has 1.6 million tweets and has been extracted using the twitter API. The tweets have been annotated (0 = negative, 4 = positive, 2 = neutral). We will convert all the negative values to -1, neutrals to 0 and positives to +1. The dataset contains 6 fields. We will transform 1500 random records of this data to contain the text of the tweets in 1 column and its corresponding sentiment value (-1, 0 or 1) in the second column.

**4.** **URL where the above dataset(s) is(are) available.**

<https://www.kaggle.com/kazanova/sentiment140>

**5. [Up to 5 lines] Which machine learning algorithm(s) is(are) going to be used? (e.g., SVM, classification trees, etc.) You are allowed to either implement this from scratch or use third-party code, e.g., scikit-learn.**

As our problem, sentiment analysis, is linearly separable we plan on using  **linear SVM classification** for sentiment analysis on a subset of the dataset. This would require gathering perfect data for testing and training, vectorizing the data, and creating a Linear SVM Model to train and then predict. In addition, we plan on using the **Multinomial Naive Bayes Classifier** for which we will split the algorithm into two essential parts, the training and classifying. In addition, we will be using **KNN Classification** for text classification into positive, negative, and neutral.

**6. [Up to 5 lines] Which cross-validation technique(s) is(are) going to be used? (e.g., training/validation/testing, k-fold cross-validation, bootstrapping). You MUST implement this from scratch.**

K fold cross validation will be used. This is because with K fold we don't need to reduce the training data. Not reducing training data is preferred because reducing the training data means we risk losing important patterns/ trends in data set, which in turn increases errors induced by bias. Since the dataset being used is pretty big the k value we would utilize would be 3 as making the fold value larger would cause bias error to reduce but would cause the variance error to increase. Making folds smaller would do the opposite of increasing the folds

**7. [Up to 10 lines] Which hyperparameter(s) is(are) going to be tuned. You MUST implement this from scratch.**

Since we are using SVMs, we will have to tune the hyperparameter **C** for SVMs. The value of C will have to be pre determined and set before the learning process begins and different values of C need to be tested since a C value which is too high will cause the margin to be too little which might cause overfitting, and in the other cause, if the C value is too low then the margin would be big cause there to be underfitting.

For multinomial naive bayes, the **alpha parameter** is the hyperparameter.

For KNN, the **K** is known as a hyperparameter.

One way to tune the hyperparameter is the use **nested cross validation**. This will be implemented by using bootstrapping inside two-fold cross validation. Three values will be used for the hyperparameter, 0.1, 1 and 10. B = 30 bootstraps will be used. For each outer fold, k-fold cross validation makes a training and testing set. For each fold, bootstrapping on the training set will be done to choose the hyperparameter with the lowest error. This hyperparameter will then be trained in the train set and then tested on the test set.

**8. [Up to 15 lines] Which experimental results will you show? (e.g., plots of number of samples versus accuracy (you can use different subsets of the same dataset), regularization parameter versus accuracy, ROC curves, plots of different datasets, etc.) You MUST implement this from scratch.**

One experimental result we will show is a plot of value of hyperparameter against accuracy. In SVMs, where C is the hyperparameter, we will plot different values of C against the model accuracy and find a C value that better fits the training data. In Multinomial Naive Bayes Classifier, since alpha is the hyperparameter we will plot different values of alpha against the accuracy. And in KNN, since K is the hyperparameter we will plot different values of K against the accuracy.

While varying the hyper-parameter C we will get different values for specificity and sensitivity and we would like to plot the ROC (Receiver Operating Characteristic) curve for the same.

In our second experiment, we will plot the number of samples vs accuracy for Naive Bayes, SVM Classifier, and KNN as another experimental result.

**9. Which programming language are you going to use? (Only MATLAB, C++, Java and Python are allowed.)**

Python

**Advice: Do not spend too much time on things such as "understanding the data", "memory problems because your data is too big", etc. Only if you are already familiar with computer vision, brain data, natural language processing, big data, parallelism, etc. then you can make use of those things, but this will not imply that you will get a higher grade just based on that fact. In general, I would recommend using easy-to-understand datasets, and smaller subsets of the data, for instance.**