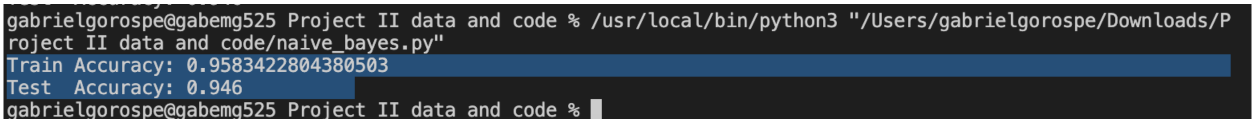
Gabriel Gorospe
ggorospe@ucsc.edu
SID: 1696580

Project II Report

- **Command to compile:**

    python3 naive_bayes.py

- **Screenshot of output of program:**



```
gabrielgorospe@gabemg525 Project II data and code % /usr/local/bin/python3 "/Users/gabrielgorospe/Downloads/P
roject II data and code/naive_bayes.py"
Train Accuracy: 0.9583422804380503
Test  Accuracy: 0.946
gabrielgorospe@gabemg525 Project II data and code %
```

- **Description/How it works**

For this assignment I implemented a learning algorithm following the functionality of a Naive Bayes Classifier (assuming each element, in this case a word in the email, is conditionally independent) which would be used to predict whether an email was “spam” or “ham.”

Using a dictionary defined in the self of the Naive Bayes class of the python code, each word in a given set of training data files representing the emails was assigned a probability of being classified as “spam” or “ham” in the fit function of the Naive Bayes class. It was necessary to implement a list of words in each email without duplicates as we wanted to see the probability that an email was spam or ham based on whether the word appeared in the email or not and disregarding the amount of appearances.

The predict function of the Naive Bayes class takes a set of testing data files representing emails different from those in the training data to test the algorithm on. The probability of each word being ham or spam is taken from the dictionary and affects the “balance” variable which will predict ham if balance>=0 and spam if <0.

Testing the accuracy of the algorithm, there was a 95.8 percent prediction accuracy for the training data and 94.6 percent for the testing data.