**Statistics 6620**

**Va7892**

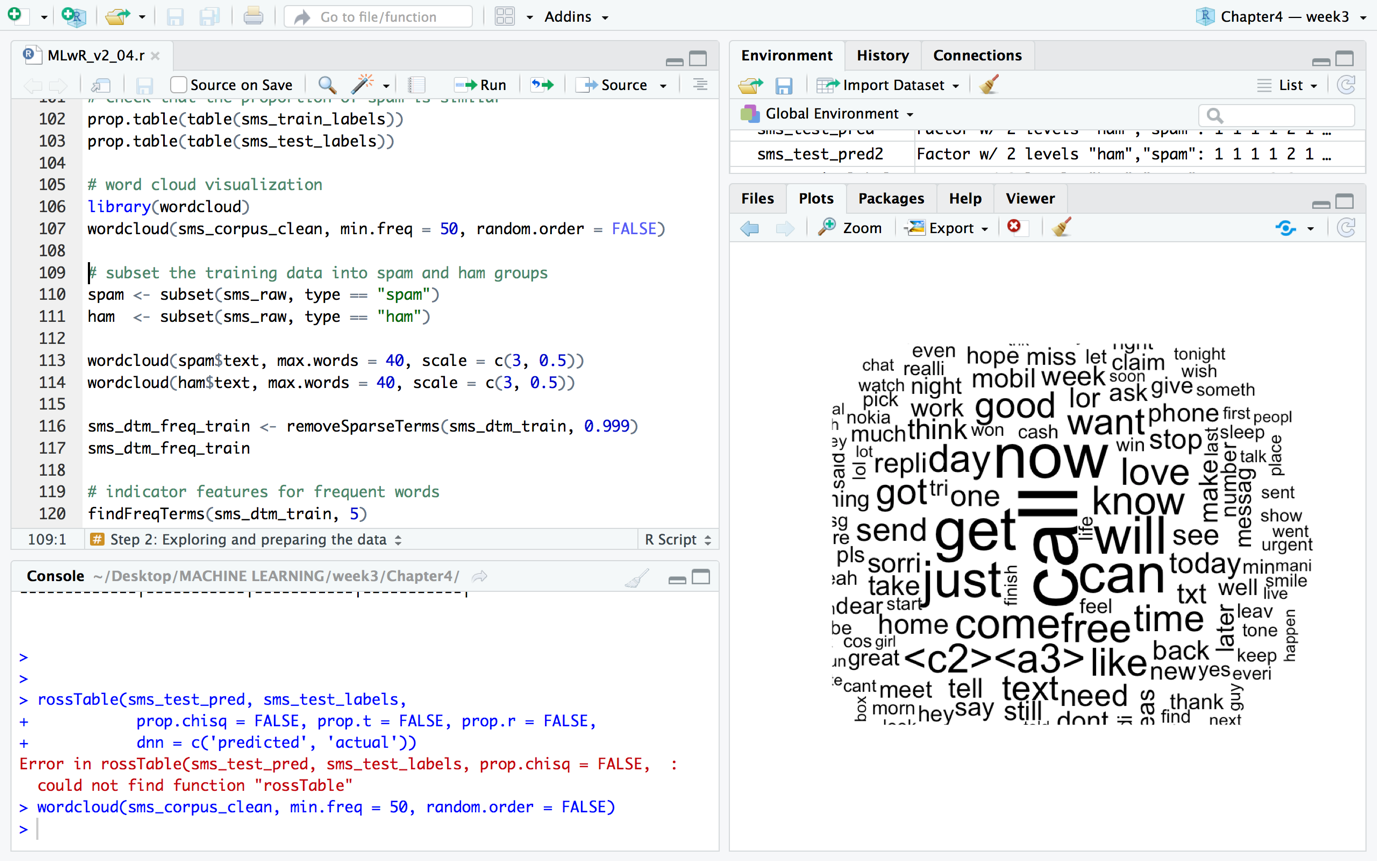
**Surabhi Asati**

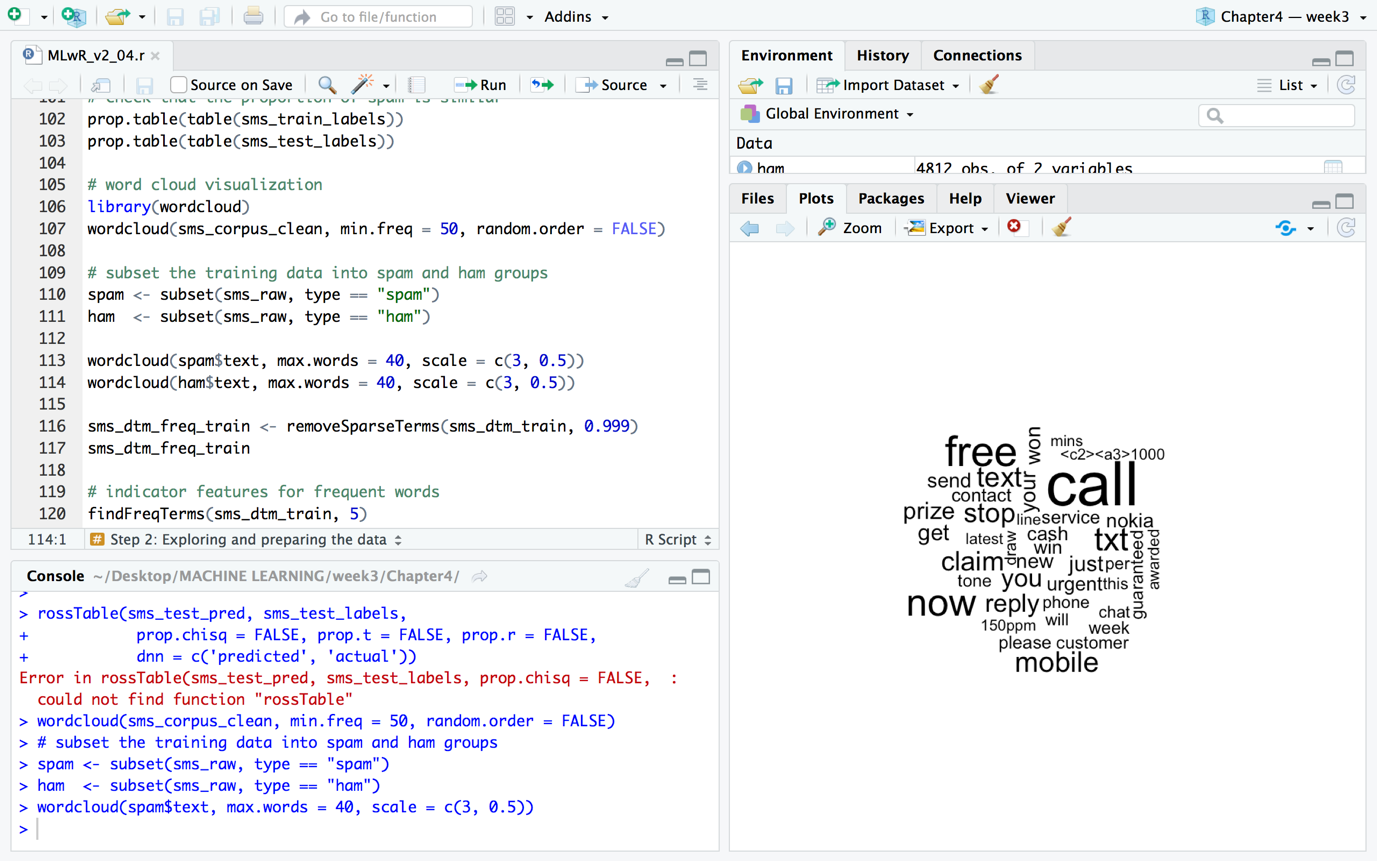
**Homework Assignment 3**

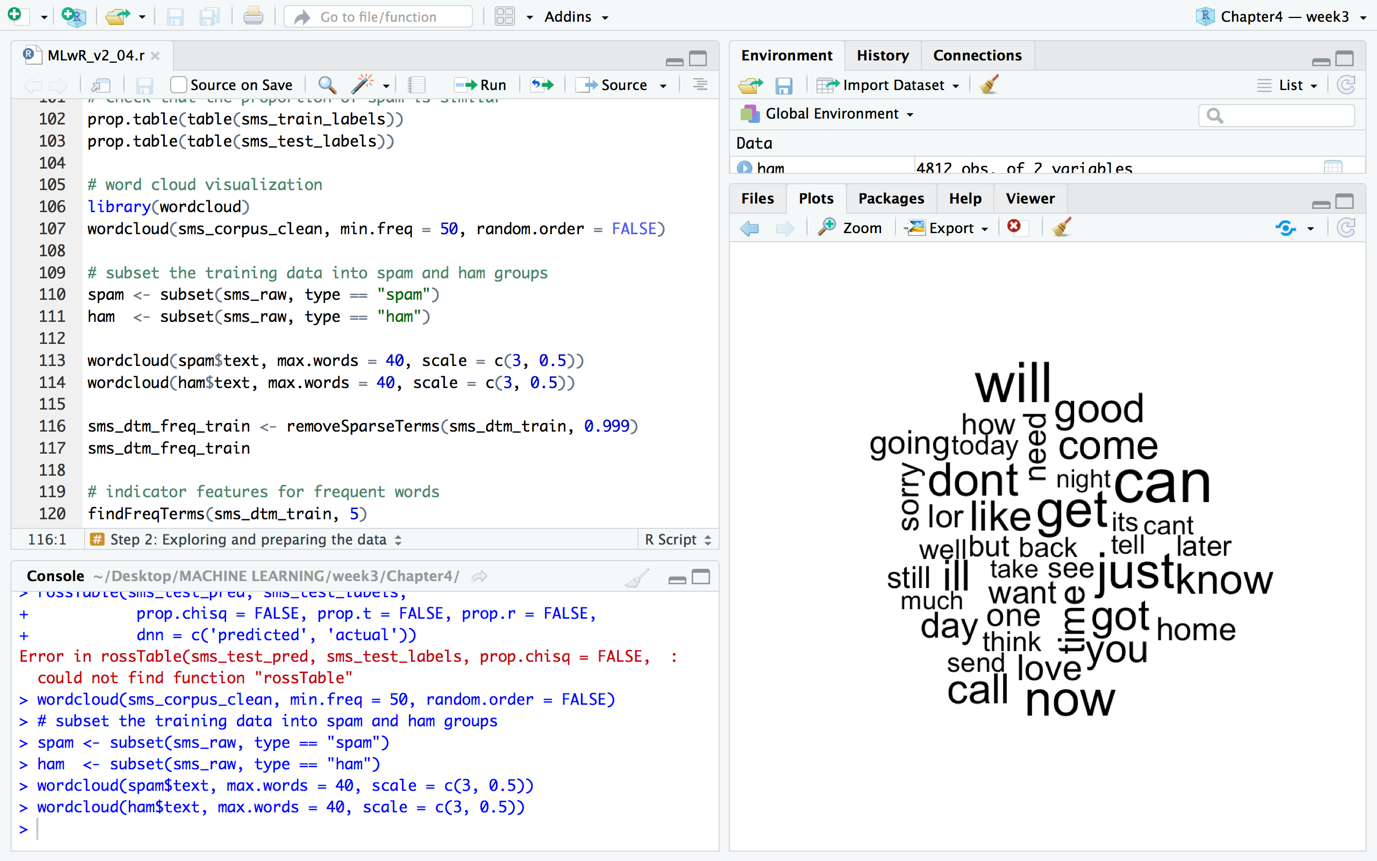
1. Perform the SMS spam filtering analysis. Produce a report explaining the data, the analysis, and the findings.

**Comparing the training and test datasets:**

Below are wordclouds to see if there is any difference in the commonly used words in ham and spam







**Analysis:**

Step 1 – collecting data

Step 2 – exploring and preparing the data

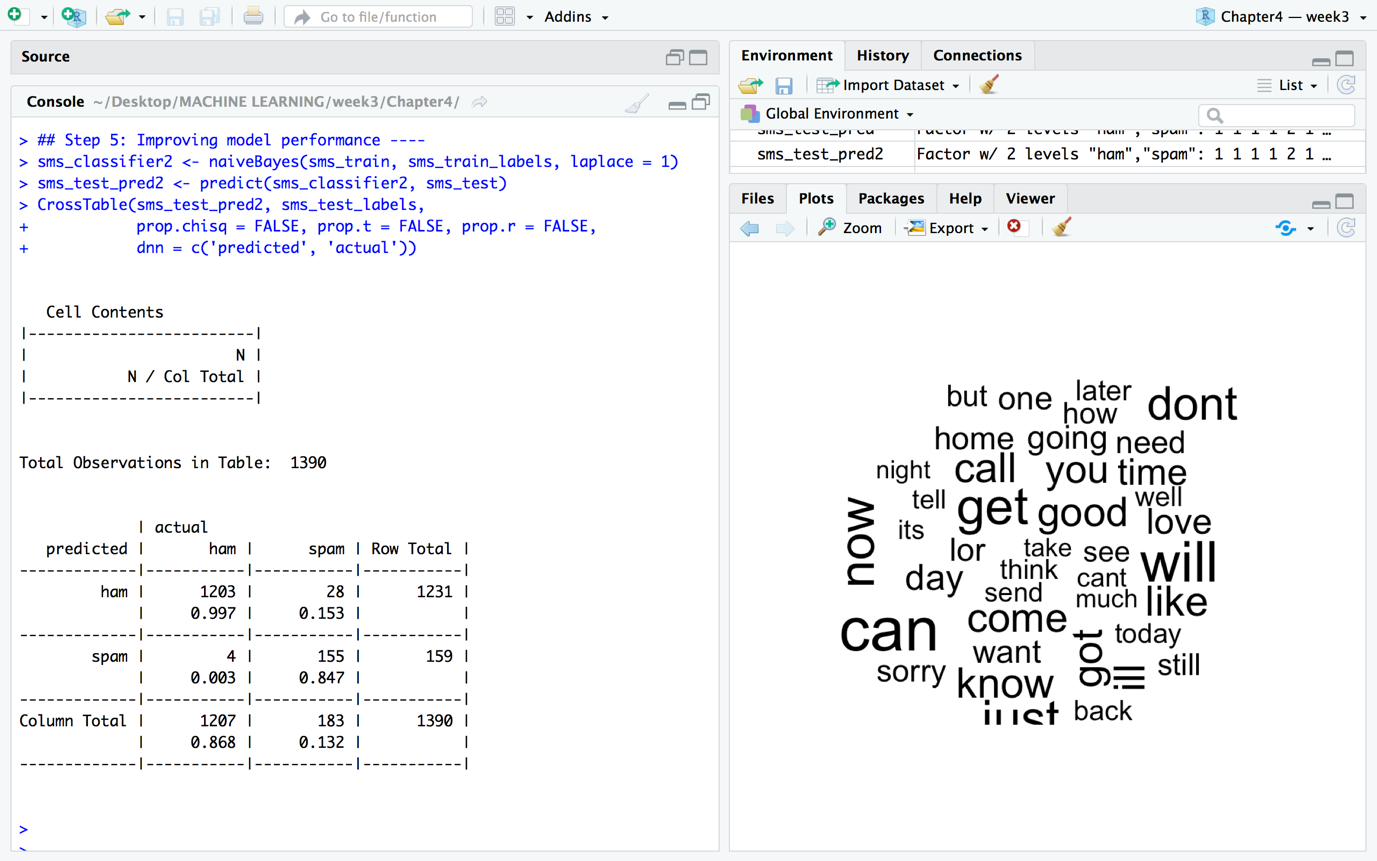
Step 3 – training a model on the data

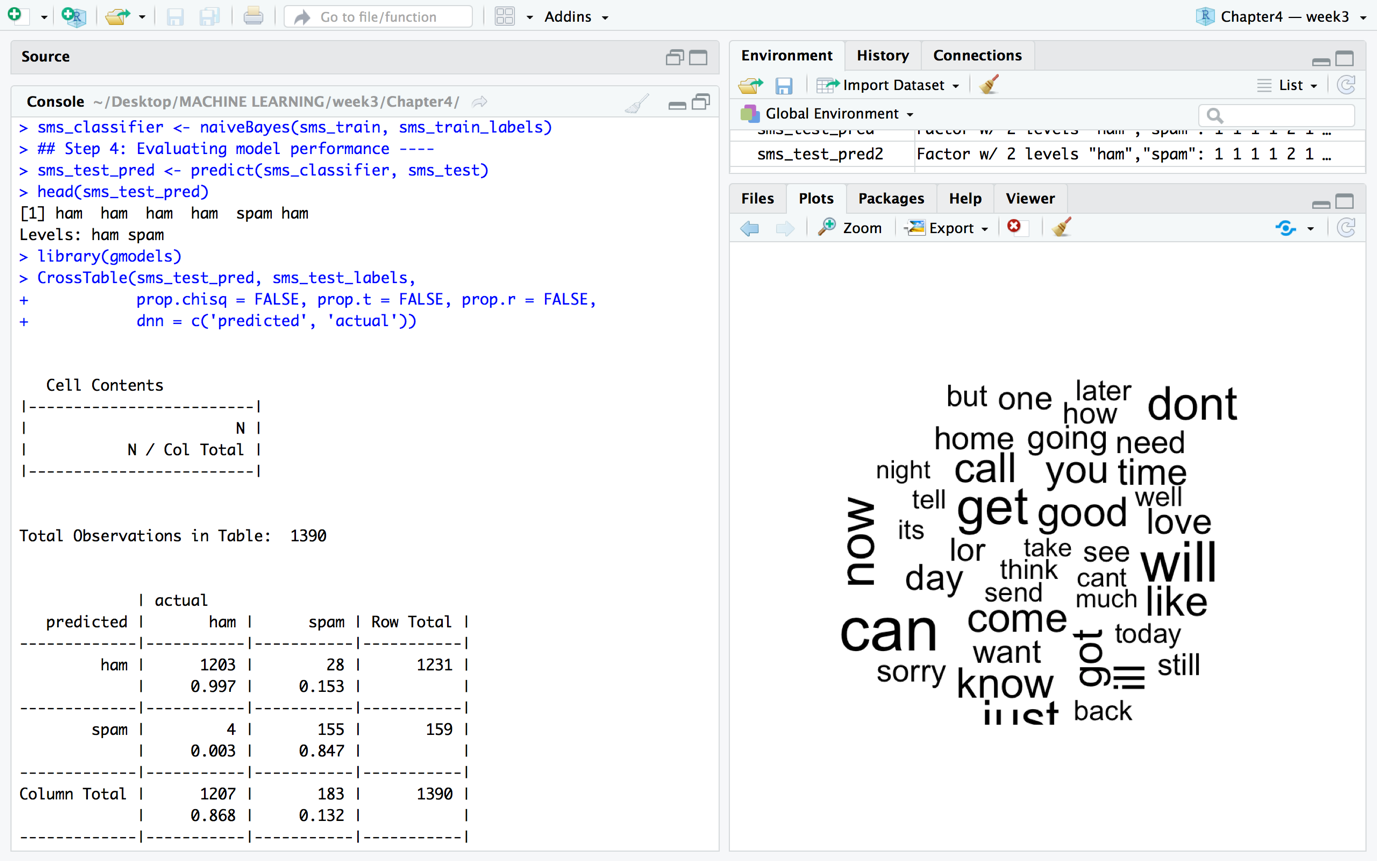
Step 4 – evaluating model performance

Step 5 – improving model performance

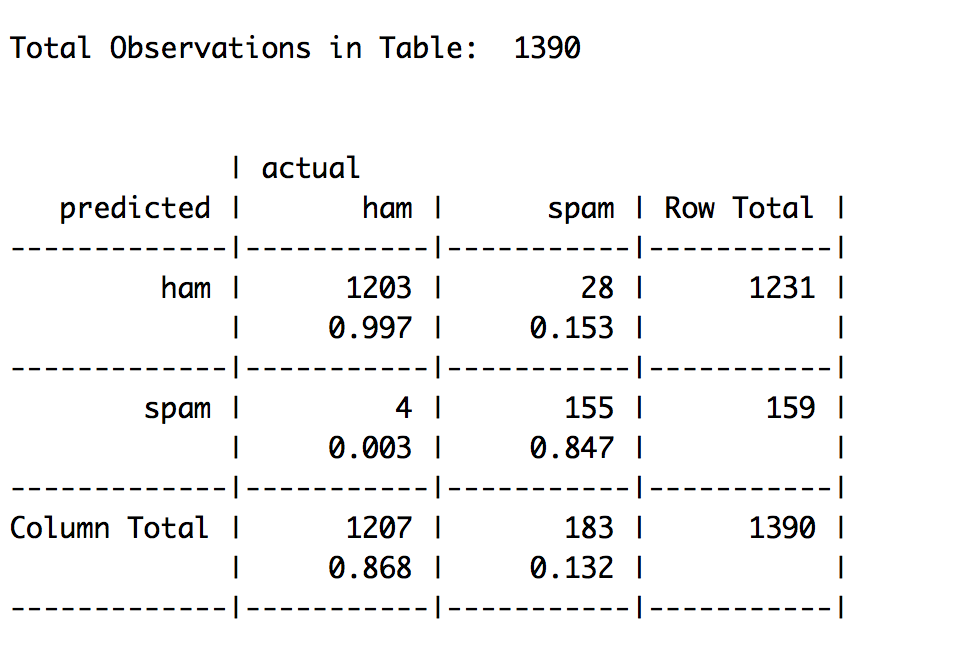
**Findings:**

* 1. The prediction that the algorithm produced are:





* 1. **The Accuracy of the predictions:**



* accuracy = TP + TN

TP+TN+FP+FN



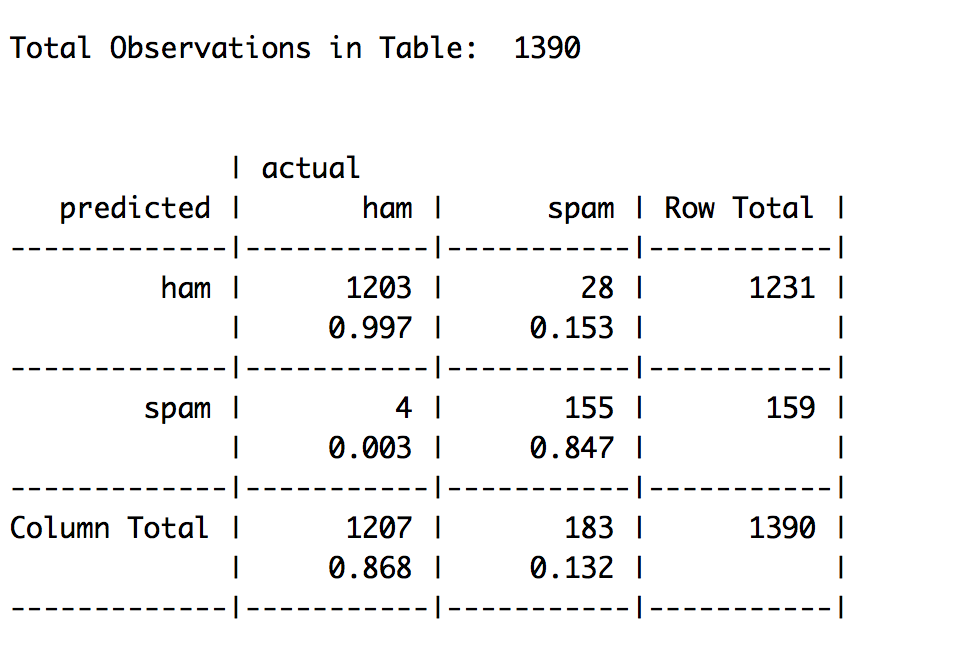
= (1203+155) / (1203+155+4+28)

= 1358/1390

= 97.7%

Out of 1390 messages, 32 were incorrectly classified. The model has an accuracy rate of 97.7% and an error rate of 2.3%. The results seem quite impressive.

* 1. **The confusion matrix.**
* True Positive (TP): Correctly classified as the class of interest - 1203
* True Negative (TN): Correctly classified as not the class of interest - 155
* False Positive (FP): Incorrectly classified as the class of interest - 4
* False Negative (FN): Incorrectly classified as not the class of interest - 28



1. Find an interesting dataset that is appropriate for applying the naive Bayes algorithm, try to load the data into R, and proceed to classify the data using naive Bayes. (You can find an example dataset anywhere you want. One suggestion is to try the first example form the [e1071](https://cran.r-project.org/web/packages/e1071/index.html) package naiveBayes function, see [Rdocumentation](https://www.rdocumentation.org/packages/e1071/versions/1.6-8). This example uses the HouseVotes84 data from the [mlbench](https://cran.r-project.org/web/packages/mlbench/index.html) package, see [RDocumentation](https://www.rdocumentation.org/packages/mlbench/versions/2.1-1). I do think everyone should try this example. Print out the dataset and see that it is full of Y and N. Also, note the NAs.)

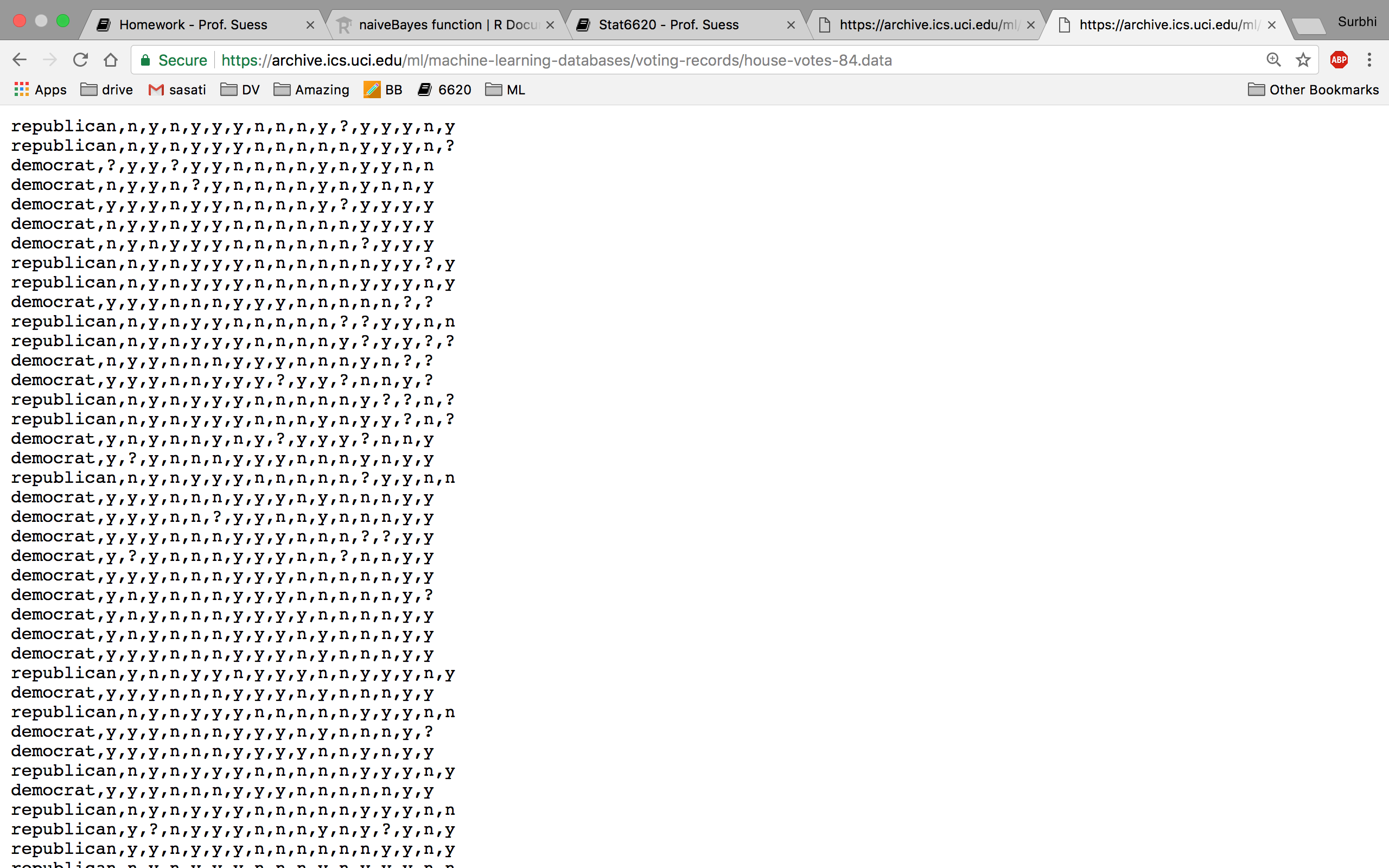
**Dataset:**

Title: 1984 United States Congressional Voting Records Database

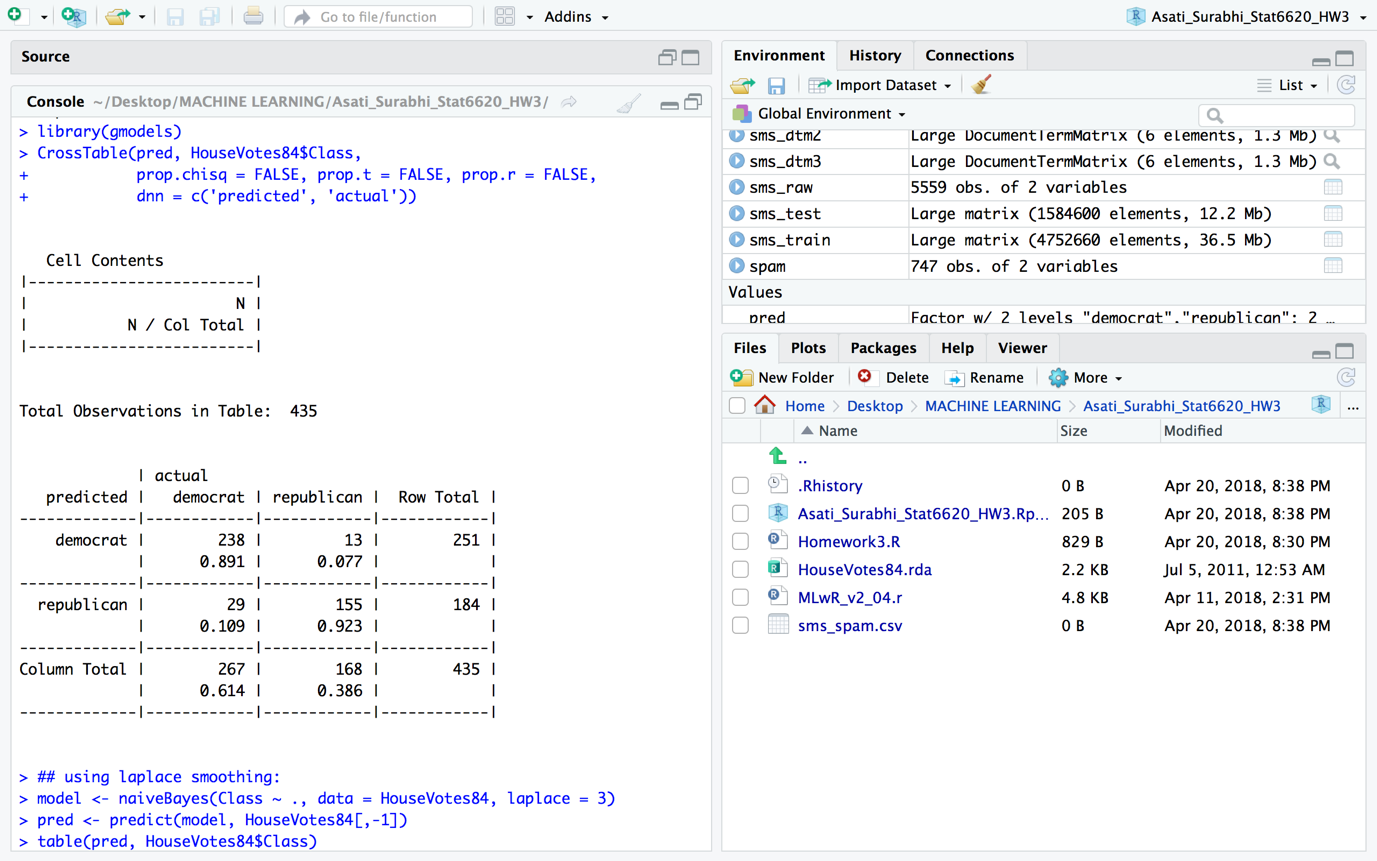
This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. The CQA lists nine different types of votes: voted for, paired for, and announced for (these three simplified to yea), voted against, paired against, and announced against (these three simplified to nay), voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known(these three simplified to an unknown disposition).

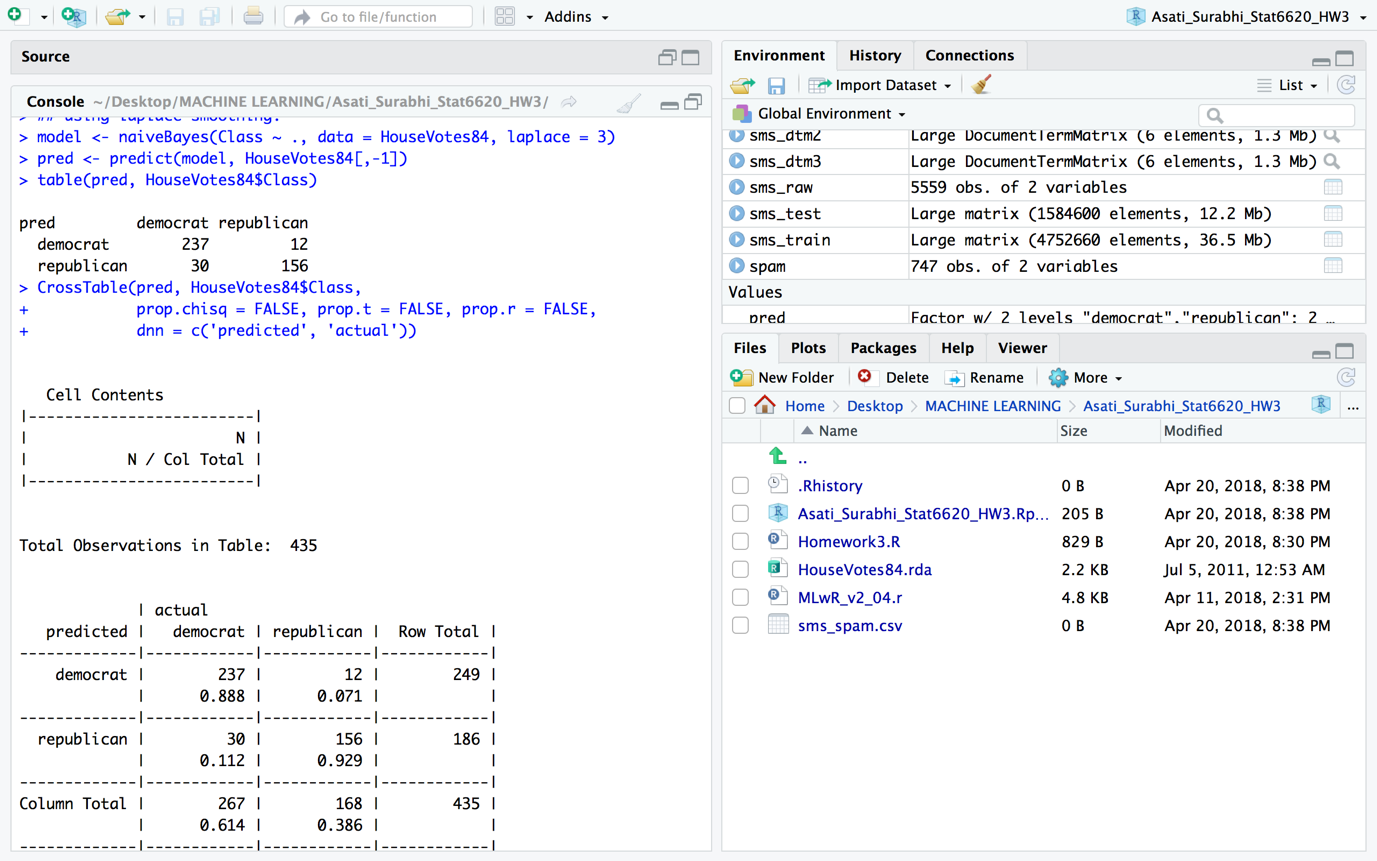
Number of Instances: 435 (267 democrats, 168 republicans)

Number of Attributes: 16 + class name = 17 (all Boolean valued)

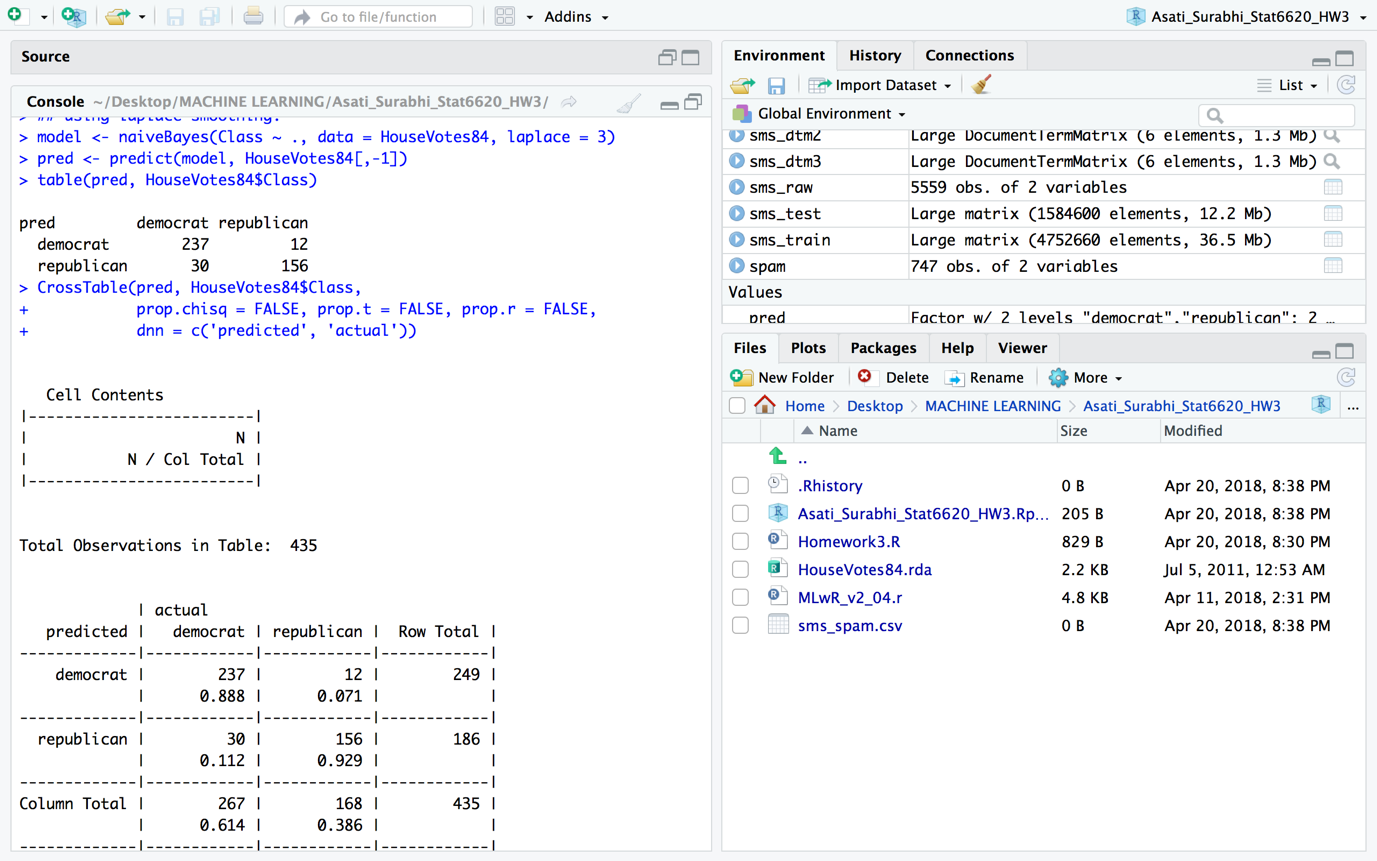


Results:





Confusion Matrix:



Case: Predicted to be Democrat

* Accuracy = TP + TN

TP+TN+FP+FN



= (237+156) / (237+156+12+30)

= 393/435

= 90.34%

Out of 435 votes, 42 were incorrectly classified. The model has an accuracy rate of 90.34% and an error rate of 9.66%. There is a scope of improvement.

1. Find a Google Sheets Add-ons app or R package that can perform Sentiment Analysis. See if you can figure out what the algorithm that is used. This problem is to look in the Google Sheets Add-ons and not in the google Chrome AppStore.)

<https://aylien.com/text-analysis-addon/>

