

NAME : Mansi Dwivedi
BRANCH : IT
UID : 2019140016
BATCH : B
COURSE : Data Analytics Lab
EXPERIMENT : 6

AIM : Separating spam from ham

PROBLEM STATEMENT :

Problem 1.1 – Loading the Dataset

Begin by loading the dataset emails.csv into a data frame called emails

- How many emails are in the dataset?
- How many of the emails are spam?
- Which word appears at the beginning of every email in the dataset?
Respond as a lower-case word with punctuation removed.
- Could a spam classifier potentially benefit from including the frequency of the word that appears in every email?
 - Yes -- the number of times the word appears might help us differentiate spam from ham.
 - No -- the word appears in every email so this variable would not help us differentiate spam from ham.
- The nchar() function counts the number of characters in a piece of text.
How many characters are in the longest email in the dataset (where longest is measured in terms of the maximum number of characters)?

Problem 2.1 -Preparing the Corpus

Follow the standard steps to build and pre-process the corpus:

- 1) Build a new corpus variable called corpus.
- 2) Using tm_map, convert the text to lowercase.
- 3) Using tm_map, remove all punctuation from the corpus.
- 4) Using tm_map, remove all English stop words from the corpus.

- 5) Using `tm_map`, stem the words in the corpus.
- 6) Build a document term matrix from the corpus, called `dtm`.
- 7) How many terms are in `dtm`? To obtain a more reasonable number of terms, limit `dtm` to contain terms appearing in at least 5% of documents, and store this result as `spdtm` (don't overwrite `dtm`, because we will use it in a later step of this homework).
- 8) How many terms are in `spdtm`?

Problem 3.1 – Building machine learning models

First, create a variable called "emailsSparse" from "spdtm" using the command `"emailsSparse = as.data.frame(as.matrix(spdtm))"` and ensure it has legal variable names with `"names(emailsSparse) = make.names(names(emailsSparse))"`. Then, copy the dependent variable from the original data frame called "emails" to "emailsSparse" using the command `"emailsSparse$spam = as.factor(emails$spam)"`.

Next, set the random seed to 123 and use the `sample.split` function to split `emailsSparse` 70/30 into a training set called "train" and a testing set called "test". Make sure to perform this step on `emailsSparse` instead of `emails`.

Using the training set, train the following two machine learning models. The models should predict the dependent variable "spam", using all other available variables as independent variables. Please be patient, as these models may take a few minutes to train.

- 1) A CART model called `spamCART`, using the default parameters to train the model (don't worry about adding `minbucket` or `cp` or specifying losses for false positives and false negatives).

Remember to add the argument `method="class"` since this is a binary classification problem.

- 2) A random forest model called `spamRF`, using the default parameters to train the model (don't worry about specifying `mtry` or `ntree` or `nodesize` or the losses for false positives and false negatives). Directly before training the random forest model, set the random seed to 123 (even though we've already done this earlier

in the problem, it's important to set the seed right before training the model so we all obtain the same results. Keep in mind though that on certain operating systems, your results might still be slightly different).

Similar to logistic regression, CART and random forest can be told to give you predicted probabilities for classification problems. CART does this by returning the proportion of observations in the bucket of interest that fall into the relevant category, and random forest does this by returning the proportion of the trees that predict the relevant category. The following commands can be used to obtain predicted probabilities for the two fitted models on the training

set:

```
predTrainCART = predict(spamCART)[,2]
```

```
predTrainRF = predict(spamRF, type="prob")[,2]
```

- 1) What is the training set accuracy of spamCART, using a threshold of 0.5 for predictions?
- 2) What is the training set accuracy of spamRF, using a threshold of 0.5 for predictions? (Remember that your answer might not match ours exactly, due to random behavior in the random forest algorithm on different operating systems.)
- 3) How many of the word stems "enron", "hou", "vinc", and "kaminski" appear in the CART tree?
- 4) What is the training set AUC of spamCART?
- 5) What is the training set AUC of spamRF?

CODE & OUTPUT:

<https://colab.research.google.com/drive/1xbOrNCYvgo2KVftGQH8-MkLfmgAB6SeO?usp=sharing>

CONCLUSION :

All the explanations and inferences have been precisely written in the Google Collab Notebook itself.