Data Science Certificate Diploma
DS650 Data Analytics

Homework 3 (20 points)

You are expected to deliver a PDF document (this document) and one Jupyter notebook with your Python code.

For the following questions, we will be working with an SMS Spam dataset provided as a .csv file. The dataset contains SMS messages labeled as either "spam" or "ham" (not spam). Our goal is to analyze, preprocess, and build a machine learning model to classify messages as spam or not. All questions must be answered using Google Colab.

To obtain reproducible results, call the following command in the beginning of your code:

```
np.random.seed(123)
```

a. Read the data into a data frame. Report in the table below the total number of spam and ham messages in the dataset. What do you observe? Which evaluation metric might be more useful (e.g. accuracy)? (2 points)

Label	**Count**
Ham	4825
Spam	747

Answer: Due to the imbalance of the dataset, accuracy may not be the best evaluative metric, since we want to minimize false positives and catch as many spam messages as possible the F1-score is the most useful option as it combines the precision and recall evaluation.

b.
We will convert the messages into tokens with different ways. Firstly, use the `LemmaTokenizer` class (that tokenizes the text, lemmatizes each word, and returns the cleaned tokens), the `StemTokenizer` class and the `get_wordnet_pos` function(maps POS tags to WordNet tags for lemmatization) from the count `vectorizer.ipynb` file from Activity Week 5 Folder. Secondly, test these two tokenizers on the sample message (displaying in the box below the tokens created for each method):

sample_msg = ["Congratulations you have won a 1000 dollars gift card Click here to claim your prize"]

(2 points)
Answer:

Lemma Tokenizer

```
sample_msg = ['Congratulations you have won a 1000 dollars gift card Click here to claim your prize']
tokenizertest = LemmaTokenizer()
tokenizertest(sample_msg[0])
```

```
['Congratulations',
 'you',
 'have',
 'win',
 'a',
 '1000',
 'dollar',
 'gift',
 'card',
 'Click',
 'here',
 'to',
 'claim',
 'your',
 'prize']
```

Stem Tokenizer

```
sample_msg = ['Congratulations you have won a 1000 dollars gift card Click here to claim your prize']
tokenizertest = StemTokenizer()
tokenizertest(sample_msg[0])
```

```
['congratul',
 'you',
 'have',
 'won',
 'a',
 '1000',
 'dollar',
 'gift',
 'card',
 'click',
 'here',
 'to',
 'claim',
 'your',
 'prize']
```

c. Let’s return to on our data! Create a binary label column called b_labels where 'ham' is mapped to 0 and 'spam' is mapped to 1. Split the dataset into training and testing sets (80% train, 20% test). Then convert the text data in the train and test set into numerical vectors using 5 different techniques:

- CountVectorizer with default arguments
- CountVectorizer with lemmatization
- CountVectorizer with stemming
- TF-IDF vectorizer with default arguments
- TF-IDF vectorizer with a maximum of 2000 features

Report and compare the shape of the resulting matrices for the training set.

(# of rows, # of columns)

(3 points)

Vector Model	**Matrix Shape**
TFIDF	(4457 , 7735)
TFIDF with 2000 features	(4457,2000)
Count Vectorizer	(4457,7735)
Count Vectorizer with Lemmatization	(4457,7520)
Count Vectorizer with Stemming	(4457 , 7228)

Answer:

Count Vectorizer and TF-IDF give the highest number of features which is 7735, while Count Vectorizer with Lemmatization and stemming reduces the vocabulary size especially with the stemming as it aggressively trims words to their root forms. The TF-IDF with 2000 features is restricted to 2000 in order to reduce dimensionality and increase computational efficiency.

c. Train a Logistic Regression classifier using with all the above vectorizers as input features. Instead of simply training your models on the entire training set and evaluating on the test set, we will use 5-fold cross validation which is a methodology introduced in Lecture 2 for model evaluation and selection. For each vectorizer use 5-fold cross validation with grid

search for selecting the hyperparameter C with options [1, 0.1, 0.01] and F1-score as the metric. Report the cross validation F1-score of these models.

(5 points)

Vector Model	**CV F1-score Logistic Regression**
TFIDF	0.8123
TFIDF with 2000 features	0.8621
Count Vectorizer	0.9292
Count Vectorizer with Lemmatization	0.9350
Count Vectorizer with Stemming	0.9394

d. Which preprocessing technique works best? What is the optimal value for C? (1 point)

Answer:It seems the count vectorizer with stemming works the best with an F1 score of 0.9394, being the highest in comparison to the other vector models. it seems that the optimal value for C is 1

e. Choosing the TF-IDF with 2000 features and Count Vectorizer with Stemming from above use 5-fold cross validation with a random forest classifier using F1 score as a metric with the following hyperparameters for tunning:

'n_estimators': [50, 100, 200],
'max_depth': [None, 10, 20],
'min_samples_split': [2, 5, 10]

Report the cross validation F1 score and the best hyperparameters.

(2 points)

Vector Model	**CV F1-score Random Forest**

TFIDF with 2000 features	0.9292
Count Vectorizer with Stemming	0.8911

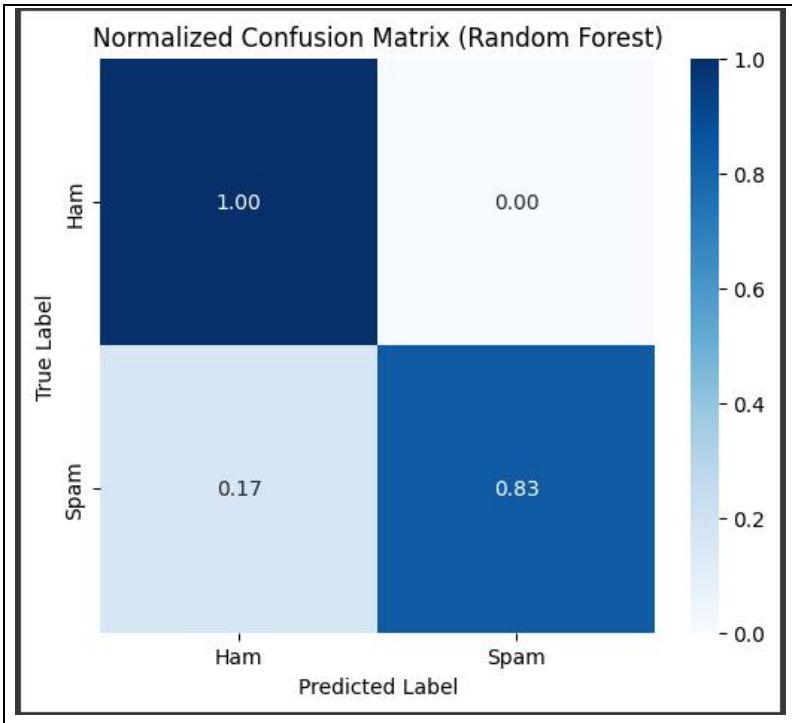
Answer: TF-IDF 2000 performed better as it is pretty efficient at finding meaningful words while count vectorizer with stemming loses some nuance due to inefficient stemming.

f. Select the best model among random forest and logistic regression and evaluate the model on the test set and report the accuracy, precision, recall, and F1-score. Plot in the following box the normalized confusion matrix based on the test data.

(4 points)

Metric	**Best Model Test Set**
Accuracy	0.9776
Precision	1.0000
Recall	0.8333
F1-score	0.9091

Plot here



g. Examine false positives (ham classified as spam) and false negatives (spam classified as ham) from the test set. Show those misclassified cases in the box below and elaborate on what might have confused the algorithm.

(3 points)

Answer:

There was no misclassification for the ham classified as spam, but there were spams that were misclassified as ham. I chose to represent the first 5 cases

“Hi I'm sue. I am 20 years old and work as a lapdancer. I love sex. Text me live - I'm i my bedroom now. text SUE to 89555. By TextOperator G2 1DA 150ppmsg 18+.

Loans for any purpose even if you have Bad Credit! Tenants Welcome. Call NoWorriesLoans.com on 08717111821.

ringtoneking 84484.

You have an important customer service announcement from PREMIER.

08714712388 between 10am-7pm Cost 10p”

It could be due to the natural and personal attitude as in the first message or the lack of Typical spam words. The message could also be too short to be detected as spam so it automatically goes to ham.