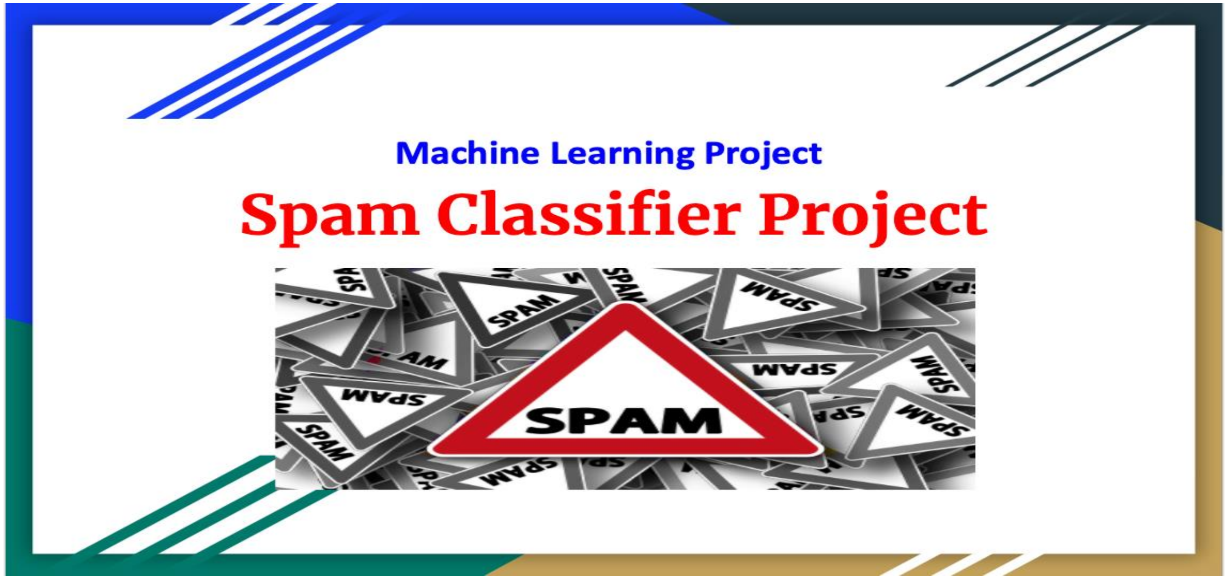
SPAM PREDICTION REPORT



Machine Learning Project

Spam Classifier Project

Submitted by:

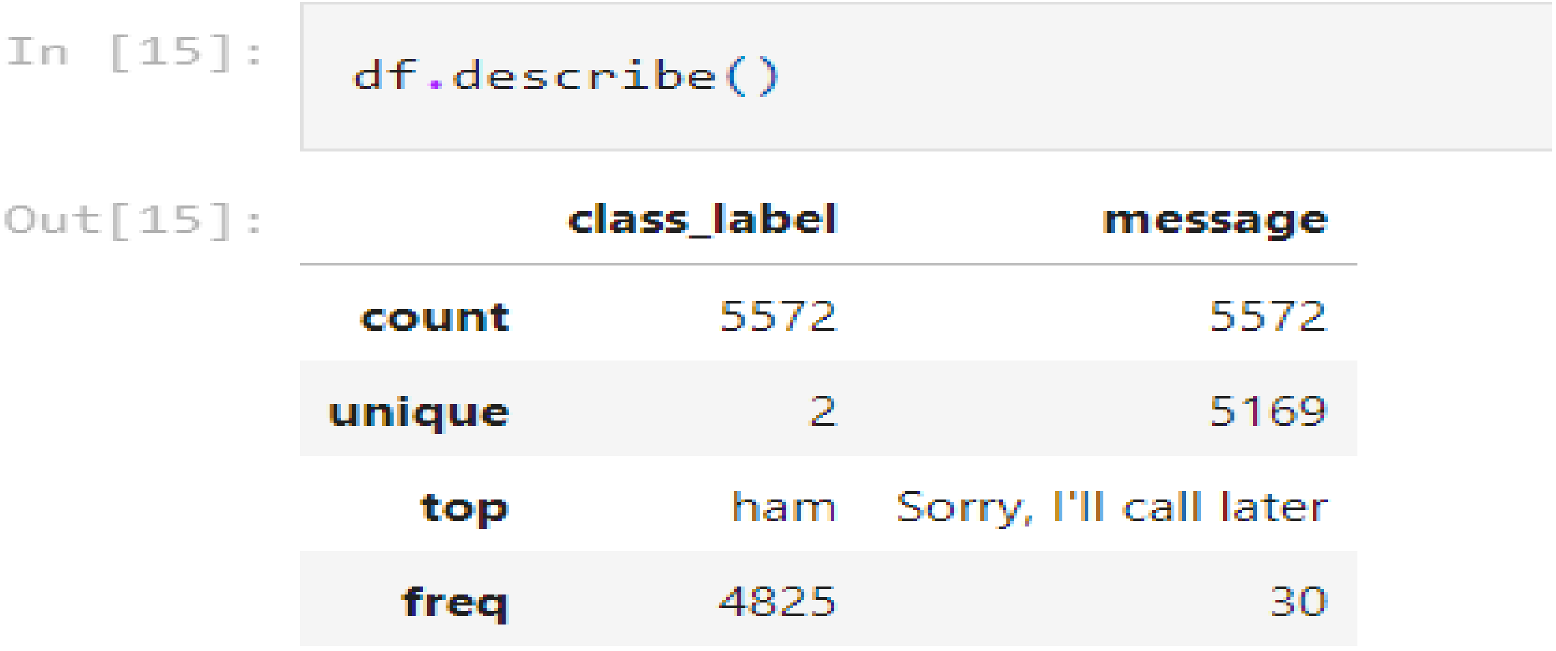
NEETAL TIWARI

Business Problem Framing

Most of us consider spam emails as one which is annoying and repetitively used for purpose of advertisement and brand promotion. We keep on blocking such email-ids but it is of no use as spam emails are still prevalent. Some major categories of spam emails that are causing great risk to security, such as fraudulent e-mails, identify theft, hacking, viruses, and malware. In order to deal with spam emails, we need to build a robust real-time email spam classifier that can efficiently and correctly flag the incoming mail spam, if it is a spam message or looks like a spam message. The latter will further help to build an Anti-Spam Filter.

Google and other email services are providing utility for flagging email spam but are still in the infancy stage and need regular feedback from the end-user. Also, popular email services such as Gmail, Yandex, yahoo mail, etc provide basic services as free to the end-user and that of course comes with EULA. There is a great scope in building email spam classifiers, as the private companies run their own email servers and want them to be more secure because of the confidential data, in such cases email spam classifier solutions can be provided to such companies.

Mathematical/ Analytical Modelling of the Problem



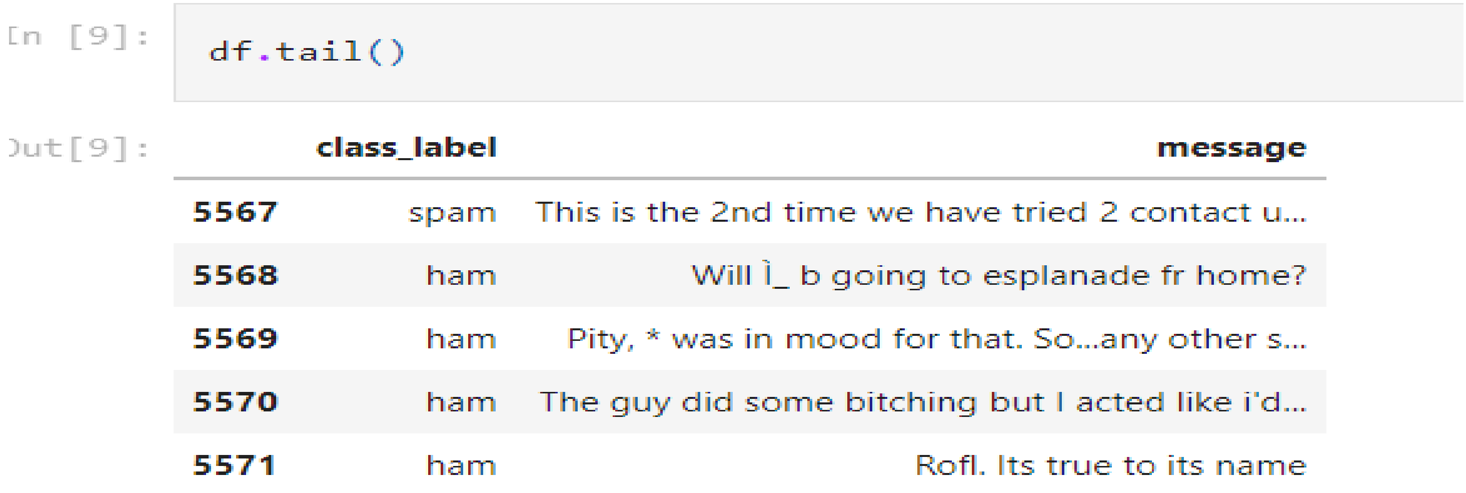
```
In [15]: df.describe()
```

Out[15]:

	class_label	message
count	5572	5572
unique	2	5169
top	ham	Sorry, I'll call later
freq	4825	30

There are 2 unique values in class_label which are ham and spam. There are 5169 unique values in massage.

Data Sources and their formats



In [9]:

```
df.tail()
```

Out[9]:

	class_label	message
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will Ì_ b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

The dataset contains two columns.

Class_label- containing the class labels ham/spam for each text message

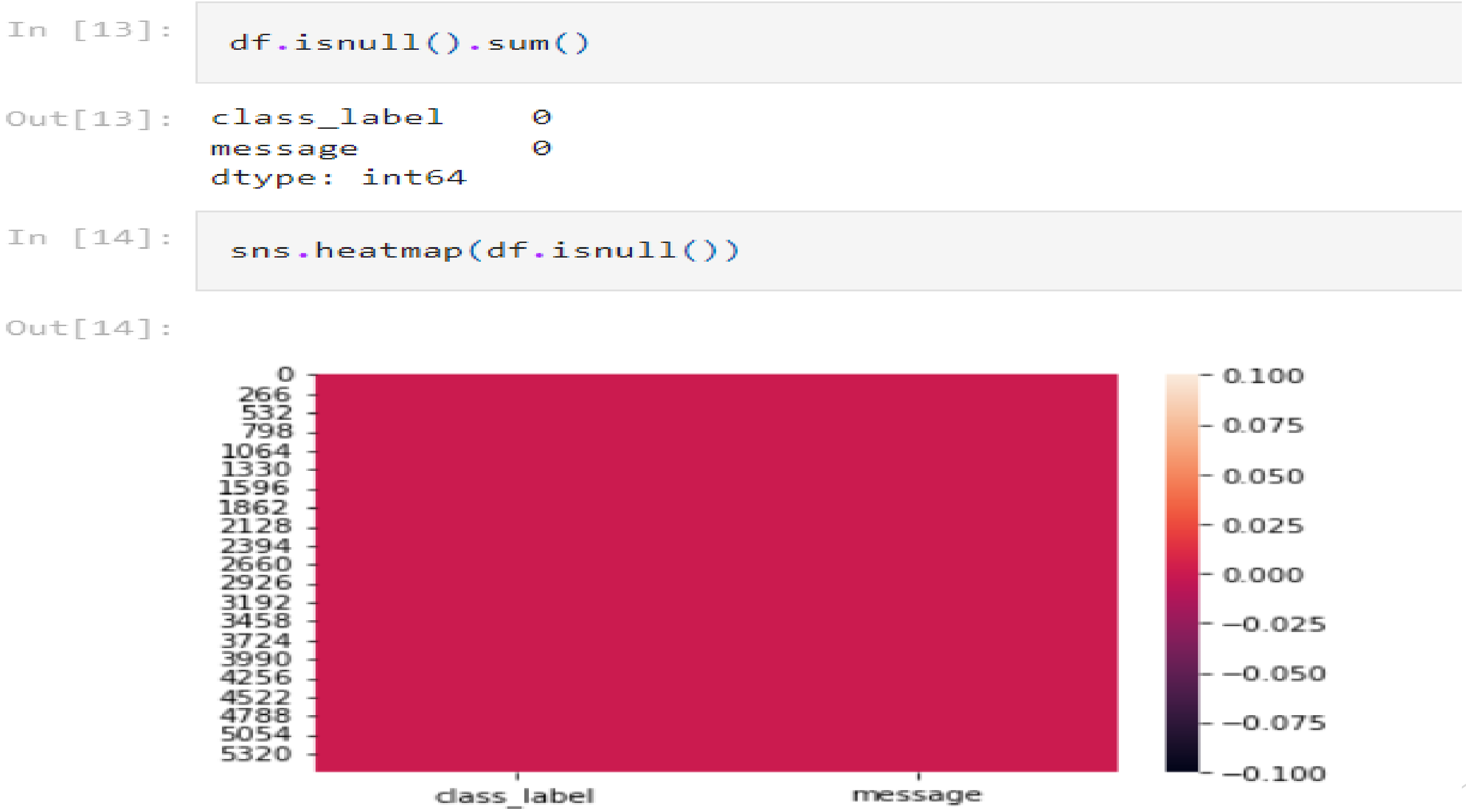
HAM-Good Mails

SPAM-Bad Mails

Message- containing the text messages themselves

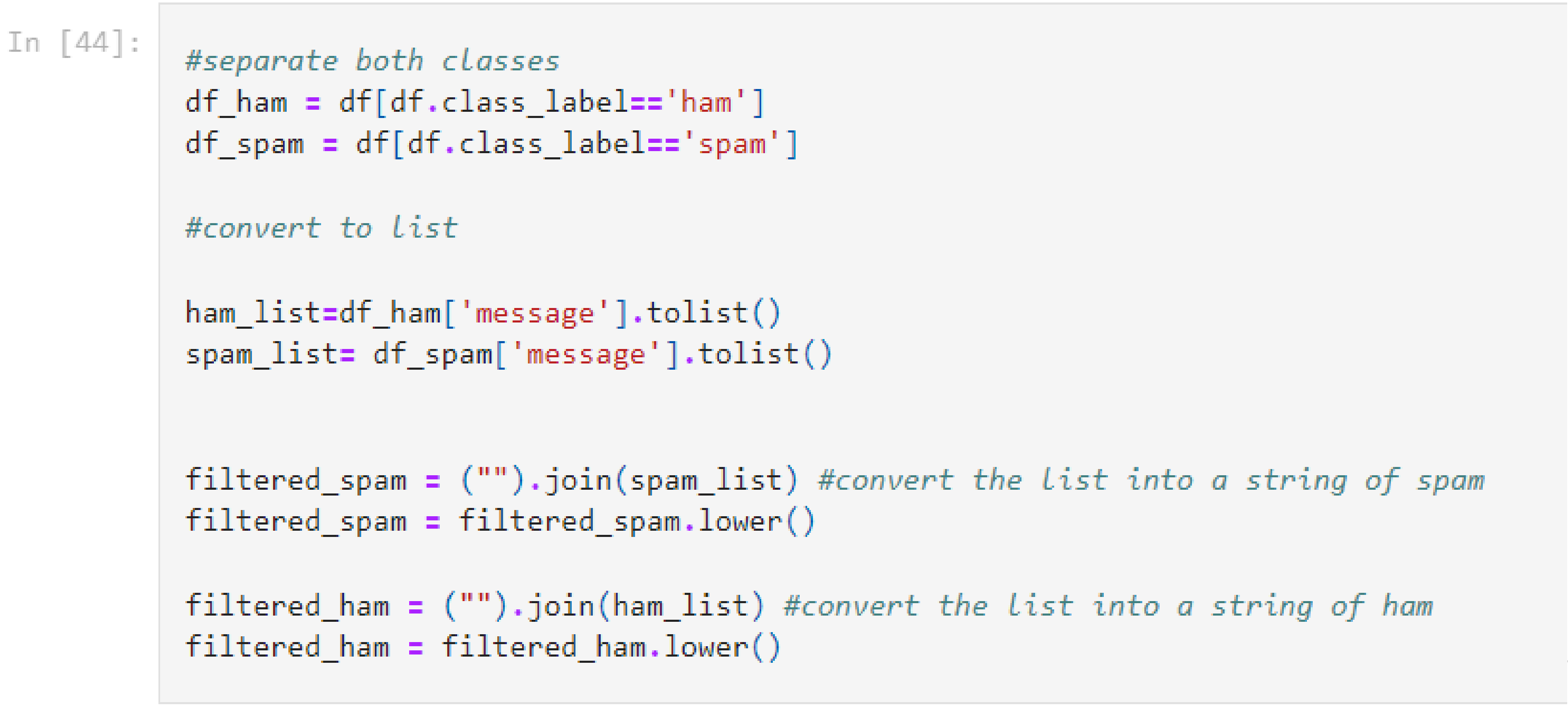
Data Pre-processing Done

1) Identifying Missing values.



```
In [13]: df.isnull().sum()

Out[13]: class_label    0
         message        0
         dtype: int64

In [14]: sns.heatmap(df.isnull())

Out[14]:
```

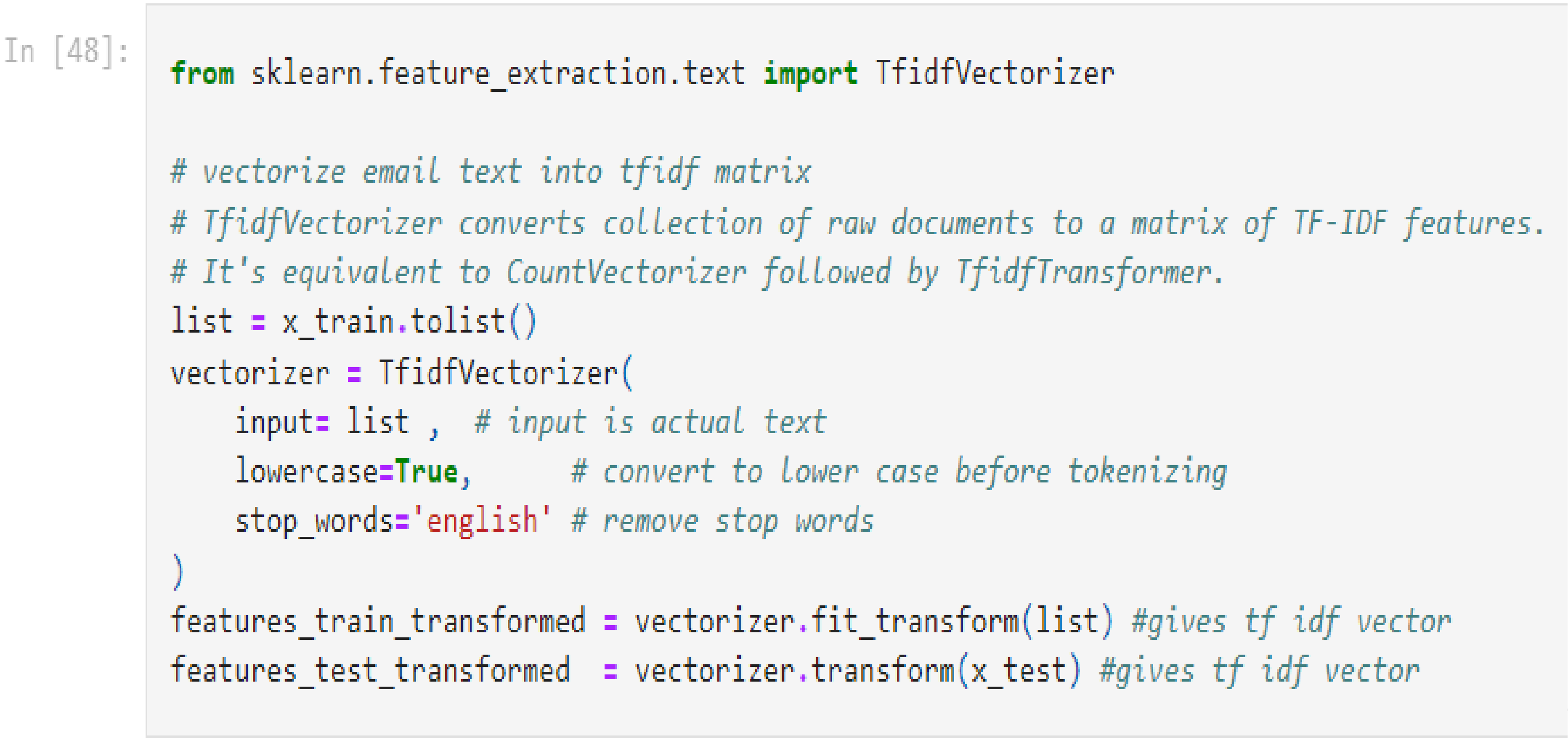
2) Converting all text to lower case.



In [44]:

```python
#separate both classes
df_ham = df[df.class_label=='ham']
df_spam = df[df.class_label=='spam']

#convert to list

ham_list=df_ham['message'].tolist()
spam_list= df_spam['message'].tolist()


filtered_spam = ("").join(spam_list) #convert the list into a string of spam
filtered_spam = filtered_spam.lower()

filtered_ham = ("").join(ham_list) #convert the list into a string of ham
filtered_ham = filtered_ham.lower()
```

3) Used TfidfVectorizer method

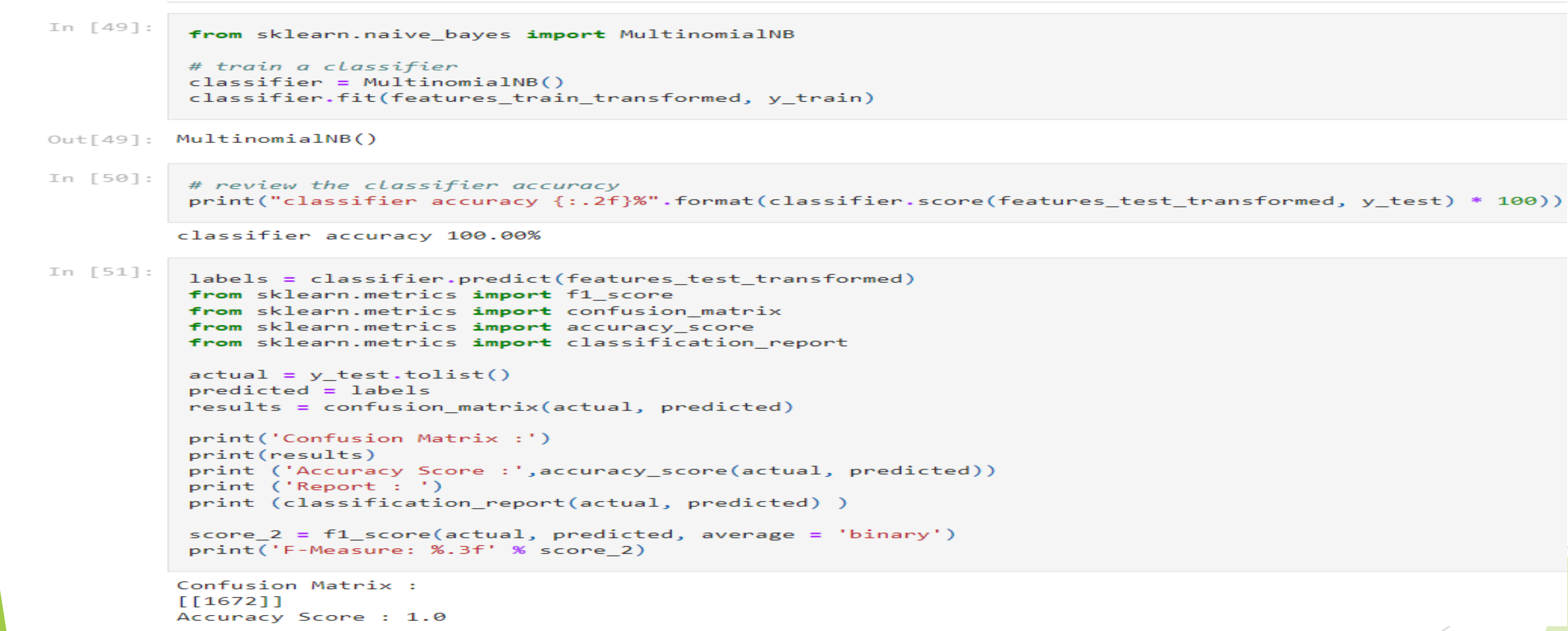


In [48]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

# vectorize email text into tfidf matrix
# TfidfVectorizer converts collection of raw documents to a matrix of TF-IDF features.
# It's equivalent to CountVectorizer followed by TfidfTransformer.
list = x_train.tolist()
vectorizer = TfidfVectorizer(
    input= list ,  # input is actual text
    lowercase=True,      # convert to lower case before tokenizing
    stop_words='english' # remove stop words
)
features_train_transformed = vectorizer.fit_transform(list) #gives tf idf vector
features_test_transformed  = vectorizer.transform(x_test) #gives tf idf vector
```

Model/s Development and Evaluation

Models used: **Naive Bayes.** Email spam classification done using traditional machine learning techniques comprise Naive Bayes, due to not having sufficient hardware resources, takes less time to train. Also, not opting for neural algorithms due to less data and computing resources.

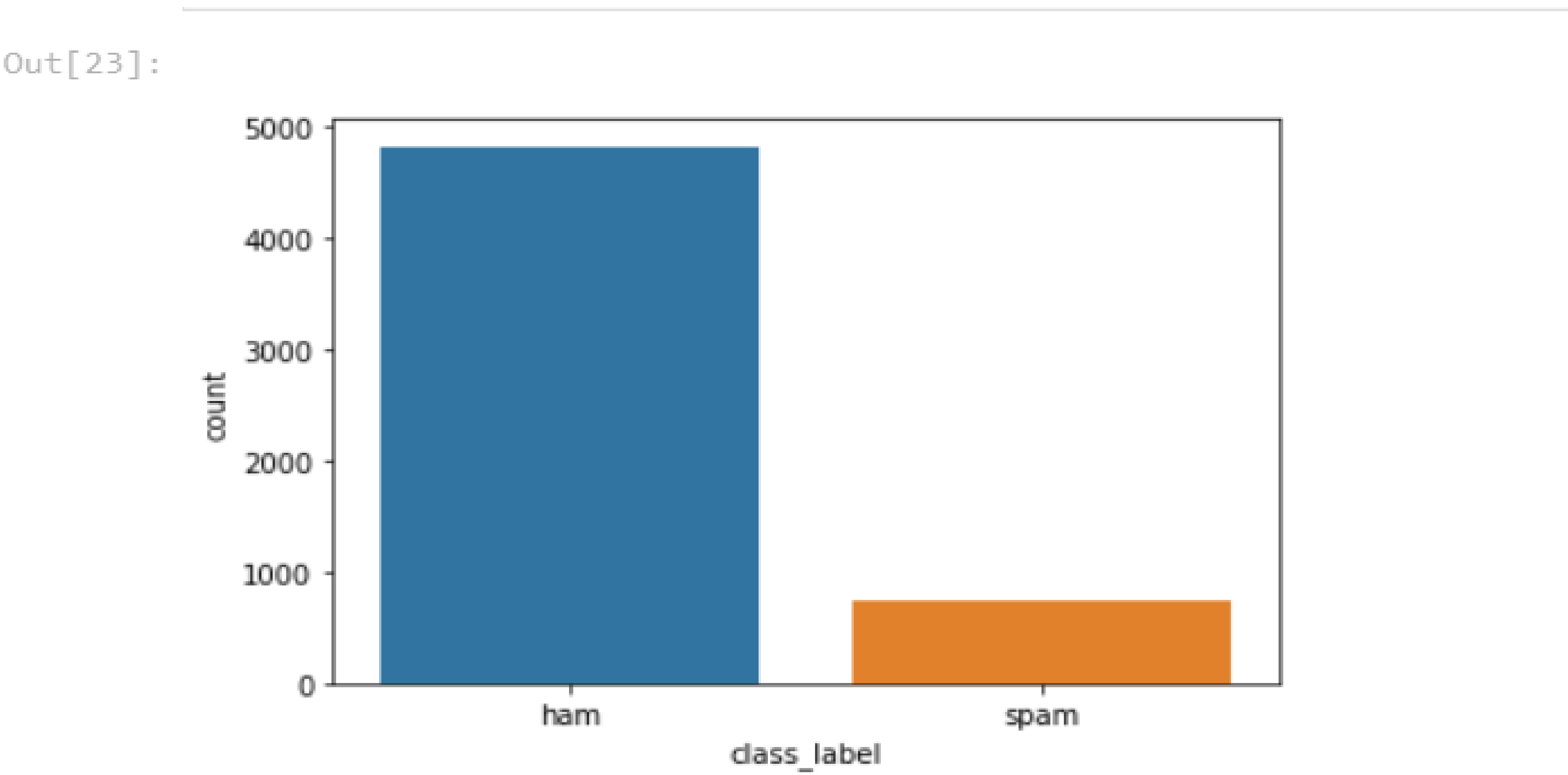


```
In [49]:  from sklearn.naive_bayes import MultinomialNB

          # train a classifier
          classifier = MultinomialNB()
          classifier.fit(features_train_transformed, y_train)

Out[49]:  MultinomialNB()

In [50]:  # review the classifier accuracy
          print("classifier accuracy {:.2f}%".format(classifier.score(features_test_transformed, y_test) * 100))

          classifier accuracy 100.00%

In [51]:  labels = classifier.predict(features_test_transformed)
          from sklearn.metrics import f1_score
          from sklearn.metrics import confusion_matrix
          from sklearn.metrics import accuracy_score
          from sklearn.metrics import classification_report

          actual = y_test.tolist()
          predicted = labels
          results = confusion_matrix(actual, predicted)

          print('Confusion Matrix :')
          print(results)
          print ('Accuracy Score :',accuracy_score(actual, predicted))
          print ('Report : ')
          print (classification_report(actual, predicted) )

          score_2 = f1_score(actual, predicted, average = 'binary')
          print('F-Measure: %.3f' % score_2)

          Confusion Matrix :
          [[1672]]
          Accuracy Score : 1.0
```

Accuracy score for SPAM project is 1.0

Visualizations

The bar graph given below depicts the percentage of Ham and Spam emails in the given dataset. The blue bar represents the count of ham emails and the orange bar shows the count of spam emails in the dataset.



```
Out[23]:
```

CONCLUSION

In this study, we reviewed machine learning approaches and their application to the field of spam filtering. A review of the state-of-the-art algorithms been applied for classification of messages as either spam or ham is provided. The attempts made by different researchers to solving the problem of spam through the use of machine learning classifiers was discussed. The evolution of spam messages over the years to evade filters was examined. The basic architecture of email spam filter and the processes involved in filtering spam emails were looked into. The paper surveyed some of the publicly available datasets and performance metrics that can be used to measure the effectiveness of any spam filter. The challenges of the machine learning algorithms in efficiently handling the menace of spam were pointed out and comparative studies of the machine learning technics available in literature was done. We also revealed some open research problems associated with spam filters. In general, the figure and volume of literature we reviewed shows that significant progress have been made and will still be made in this field. Having discussed the open problems in spam filtering, further research to enhance the effectiveness of spam filters need to be done. This will make the development of spam filters to continue to be an active research field for academician and industry practitioners researching machine learning techniques for effective spam filtering. Our hope is that research students will use this paper as a spring board for doing qualitative research in spam filtering using machine learning, deep leaning and deep adversarial learning algorithms.