```
from tensorflow.keras import losses, models, optimizers
from tensorflow.keras.models import Sequential
from tensorflow.keras.wrappers.scikit_learn import KerasRegressor
from tensorflow.keras.activations import relu, softmax
from tensorflow.keras.callbacks import (EarlyStopping, LearningRateScheduler,
                            ModelCheckpoint, TensorBoard, ReduceLROnPlateau)
from tensorflow.keras.layers import (Convolution2D, Dense, Dropout, GlobalAveragePooling2D,
                             GlobalMaxPool2D, Input, MaxPool2D, concatenate, Activation,
                             MaxPooling2D,Flatten,BatchNormalization, Conv2D,AveragePooling2D)
from tensorflow.keras.utils import Sequence, to_categorical
from sklearn.datasets import load_iris
from sklearn.datasets import make_moons
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_graphviz
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix
from sklearn.pipeline import Pipeline
from sklearn import preprocessing
from sklearn import utils
from matplotlib import pyplot as plt

import numpy as np
import pandas as pd
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
```

Problem 1.1 – Loading the Dataset

```
data = pd.read_csv("emails.csv")
```

```
data.head()
```

	text	spam
0	Subject: naturally irresistible your corporate...	1
1	Subject: the stock trading gunslinger fanny i...	1
2	Subject: unbelievable new homes made easy im ...	1
3	Subject: 4 color printing special request add...	1
4	Subject: do not have money , get software cds ...	1

```
is_null = pd.isnull(data).sum()
print('The number of empty values by column:')
print(is_null)
```

```
The number of empty values by column:
text    0
spam    0
dtype: int64
```

clean dataset; can go onto the second step of preparing the corpus

```
data.shape
```

```
(5728, 2)
```

```
data.nunique()
```

```
text    5695
spam       2
dtype: int64
```

```
data['spam'].value_counts(dropna=False)
```

```
0    4360
1    1368
Name: spam, dtype: int64
```

```
len(data['text'].max())
```

```
1032
```

Answers to the questions of Problem 1.1

Q) How many emails are in the dataset?

Ans) 5728

Q) How many of the emails are spam?

Ans) 1368

Q) Which word appears at the beginning of every email in the dataset?

Ans) subject

Q) Could a spam classifier potentially benefit from including the frequency of the word that appears in every email?

Ans) No -- the word appears in every email so this variable would not help us differentiate spam from ham.

Q) How many characters are in the longest email in the dataset (where longest is measured in terms of the maximum number of characters)?

Ans) 1032

Problem 2.1 -Preparing the Corpus

```
#2 convert the text to lowercase
data['text'] = data['text'].str.lower()
```

```
data.head()
```

	text	spam
0	subject: naturally irresistible your corporate...	1
1	subject: the stock trading gunslinger fanny i...	1
2	subject: unbelievable new homes made easy im ...	1
3	subject: 4 color printing special request add...	1
4	subject: do not have money , get software cds ...	1

```
#3 removing all punctuation from dataset
data["text"] = data['text'].str.replace('[^\w\s]','')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: The default value of regex will change from True to False
```

```
data.head()
```

	text	spam
0	subject naturally irresistible your corporate ...	1

```
##Let us remove the 'subject' from each of the texts; its absoloutely useless
data['text'] = data["text"].str.replace("subject","")
```

```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords


def remove_stopwords(text):
    new_list = []
    words = word_tokenize(text)
    stopwrds = stopwords.words('english')
    for word in words:
        if word not in stopwrds:
            new_list.append(word)
    return ' '.join(new_list)


#4 removing all English stopwords from the dataset
data['text'] = data['text'].apply(remove_stopwords)


data.head()
```

	text	spam
0	naturally irresistible corporate identity It r...	1
1	stock trading gunslinger fanny merrill muzo co...	1
2	unbelievable new homes made easy im wanting sh...	1
3	4 color printing special request additional in...	1
4	money get software cds software compatibility ...	1

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize


def perform_stemming(text):
    stemmer = PorterStemmer()
    new_list = []
    words = word_tokenize(text)
    for word in words:
        new_list.append(stemmer.stem(word))

    return " ".join(new_list)


#5 stem the words in the dataset
data['text'] = data['text'].apply(perform_stemming)


data.head()
```

```
from sklearn.feature_extraction.text import CountVectorizer


vec = CountVectorizer()
X = vec.fit_transform(data['text'])


#6  Build a document term matrix from the dataset, called dtm.
dtm = pd.DataFrame(X.toarray(), columns=vec.get_feature_names())
dtm.head()
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprec
  warnings.warn(msg, category=FutureWarning)
```

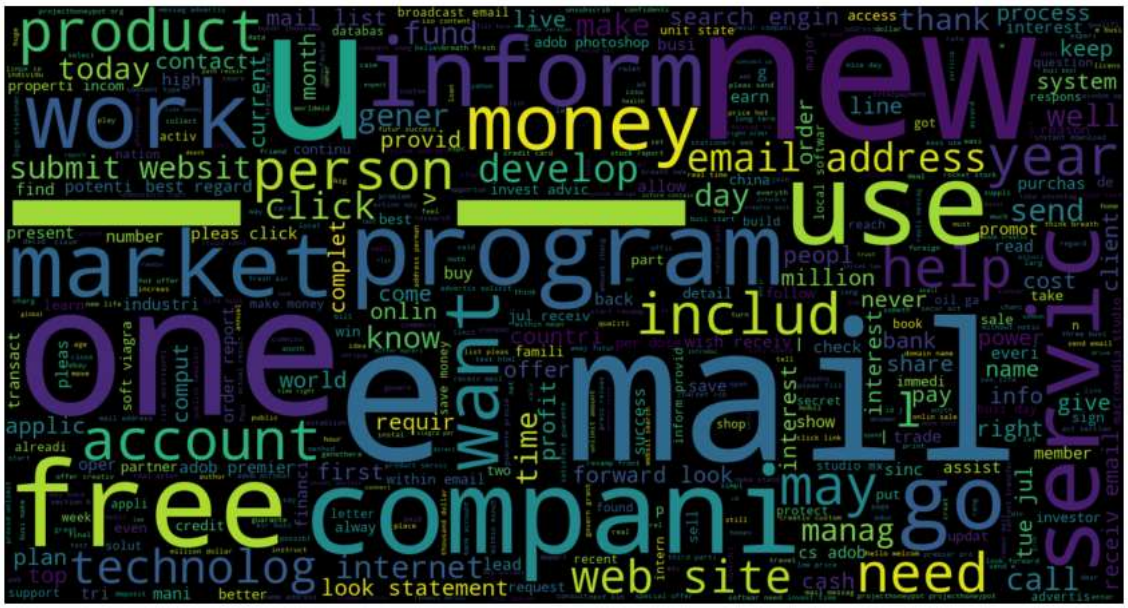
	00	000	0000	000000	00000000	0000000000	000000000003619	000000000003991	000000000003997	000000000005168	...	zwzi
0	0	0	0	0	0	0	0	0	0	0	...	
1	0	0	0	0	0	0	0	0	0	0	...	
2	0	0	0	0	0	0	0	0	0	0	...	
3	0	0	0	0	0	0	0	0	0	0	...	
4	0	0	0	0	0	0	0	0	0	0	...	

5 rows × 29251 columns

Q) How many terms are in dtm?

Ans) 29251

```
#6  Build a document term matrix from the dataset, called dtm.
from wordcloud import WordCloud
text = " ".join(data[data['spam'] == 1]['text'])
plt.figure(figsize = (15, 10))
wordcloud = WordCloud(max_words=500, height= 800, width = 1500,  background_color="black", colormap= 'viridis').generate(text)
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```



Problem 3.1 – Building machine learning models

[ ] ↳ *11 cells hidden*