13Z311-EXERCISE 4-TEXT PROCESSING

October 17, 2018

1 EXERCISE 4

1.1 TEXT PROCESSING

```
In [33]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import seaborn as sns

In [34]: data = pd.read_csv("spam.csv",encoding='latin-1')

In [35]: data.head()
```

Out[35]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

```
In [36]: data = data.drop(["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"], axis=1)
         data = data.rename(columns={"v1":"label", "v2":"text"})

In [37]: data.tail()
```

Out[37]:

	label	text
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

```
In [38]: data.label.value_counts()
```

```
Out[38]: ham     4825
         spam     747
         Name: label, dtype: int64
```

```
In [39]: data['label_num'] = data.label.map({'ham':0, 'spam':1})
```

```
In [40]: data.head()
```

```
Out[40]:   label                                               text  label_num
         0   ham  Go until jurong point, crazy.. Available only ...          0
         1   ham                      Ok lar... Joking wif u oni...          0
         2  spam  Free entry in 2 a wkly comp to win FA Cup fina...          1
         3   ham  U dun say so early hor... U c already then say...          0
         4   ham  Nah I don't think he goes to usf, he lives aro...          0
```

2 DATASET SPLIT

```
In [41]: from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test = train_test_split(data["text"],data["label"], test_size
```

```
In [42]: print(X_train.shape)
         print(X_test.shape)
         print(y_train.shape)
         print(y_test.shape)
```

```
(4457,)
(1115,)
(4457,)
(1115,)
```

```
In [43]: from sklearn.feature_extraction.text import CountVectorizer
         vect = CountVectorizer()
         vect.fit(X_train)
         X_train_df = vect.transform(X_train)
         X_test_df = vect.transform(X_test)
```

3 Multinomial Naive Bayes

```
In [44]: prediction = dict()
         from sklearn.naive_bayes import MultinomialNB
         model = MultinomialNB()
         model.fit(X_train_df,y_train)
```

```
Out[44]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

```
In [45]: prediction["Multinomial"] = model.predict(X_test_df)
```

```
In [46]: from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

In [47]: accuracy_score(y_test,prediction["Multinomial"])

Out[47]: 0.9883408071748879
```

4 KNN

```
In [48]: from sklearn.neighbors import KNeighborsClassifier
         model = KNeighborsClassifier(n_neighbors=5)
         model.fit(X_train_df,y_train)

Out[48]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=1, n_neighbors=5, p=2,
                    weights='uniform')

In [49]: prediction["knn"] = model.predict(X_test_df)

In [50]: accuracy_score(y_test,prediction["knn"])

Out[50]: 0.9121076233183857

In [ ]:
```