Name : Prem Bansod

Roll no. 41310

Subject : LP2

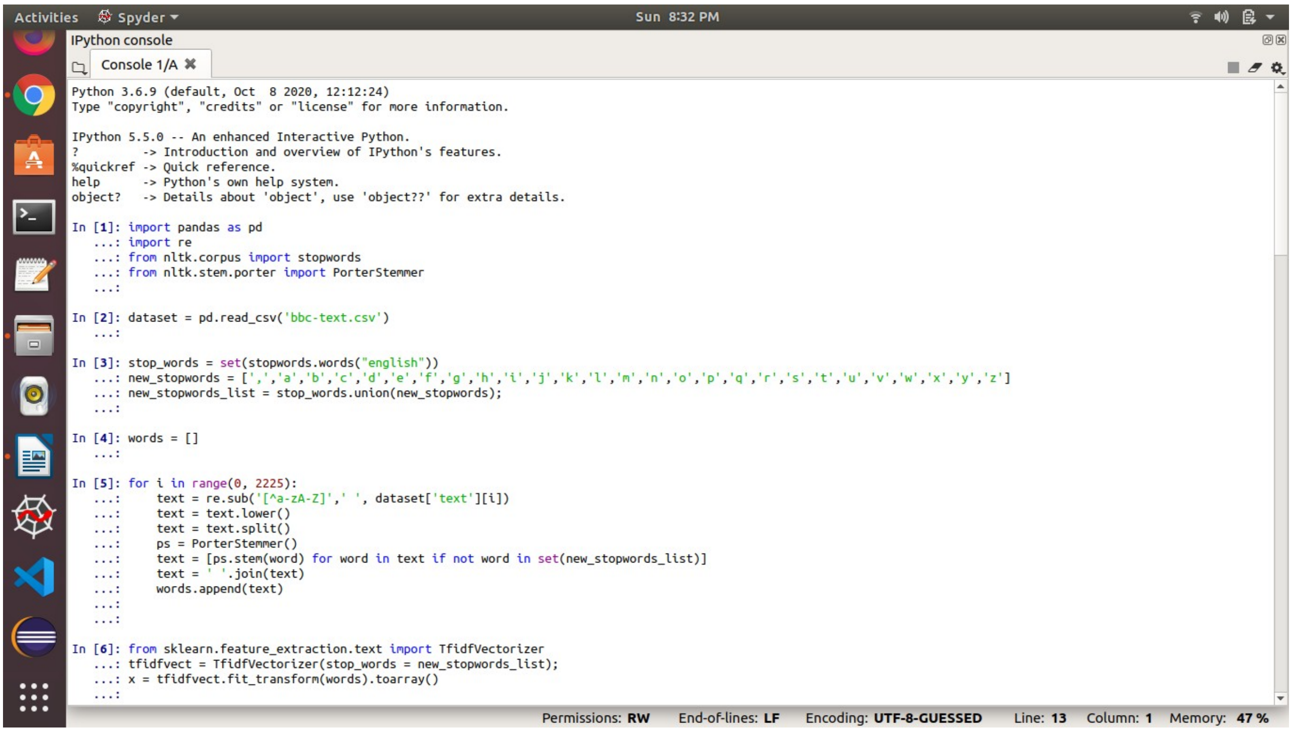
Assignment 4

Code:

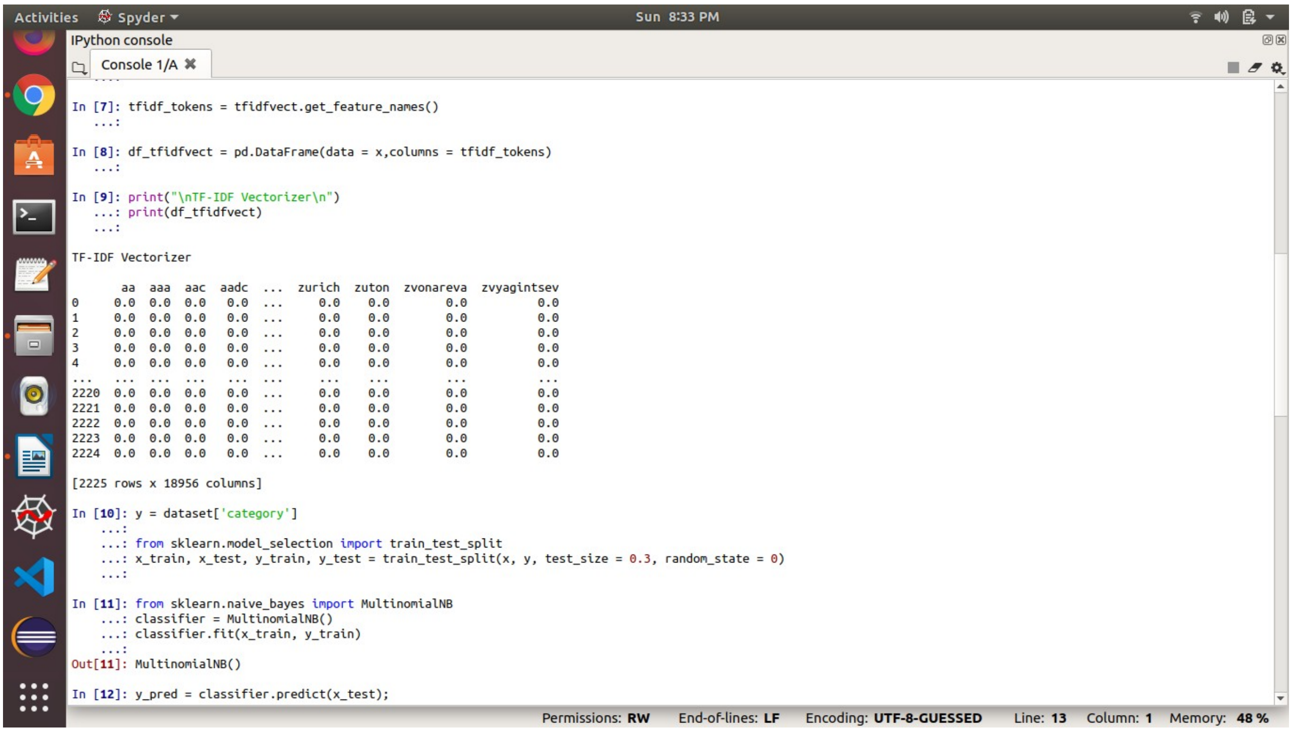
```
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer

dataset = pd.read_csv('bbc-text.csv')

stop_words = set(stopwords.words("english"))
new_stopwords = [',','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']
new_stopwords_list = stop_words.union(new_stopwords);

words = []

for i in range(0, 2225):
    text = re.sub('[^a-zA-Z]',' ', dataset['text'][i])
    text = text.lower()
    text = text.split()
    ps = PorterStemmer()
    text = [ps.stem(word) for word in text if not word in set(new_stopwords_list)]
    text = ' '.join(text)
    words.append(text)

from sklearn.feature_extraction.text import TfidfVectorizer
tfidfvect = TfidfVectorizer(stop_words = new_stopwords_list);
x = tfidfvect.fit_transform(words).toarray()

tfidf_tokens = tfidfvect.get_feature_names()

df_tfidfvect = pd.DataFrame(data = x,columns = tfidf_tokens)

print("\nTF-IDF Vectorizer\n")
print(df_tfidfvect)


y = dataset['category']

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 0)

from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(x_train, y_train)
```

```
y_pred = classifier.predict(x_test);

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)

from sklearn.metrics import accuracy_score
a = accuracy_score(y_test,y_pred)
print("The accuracy of this model is: ", a*100)


from sklearn.metrics import precision_score,recall_score,f1_score

print('precision:',precision_score(y_test,y_pred,average="macro"))
print('recall:',recall_score(y_test,y_pred,average="macro"))
print('fscore:',f1_score(y_test,y_pred,average="macro"))
```

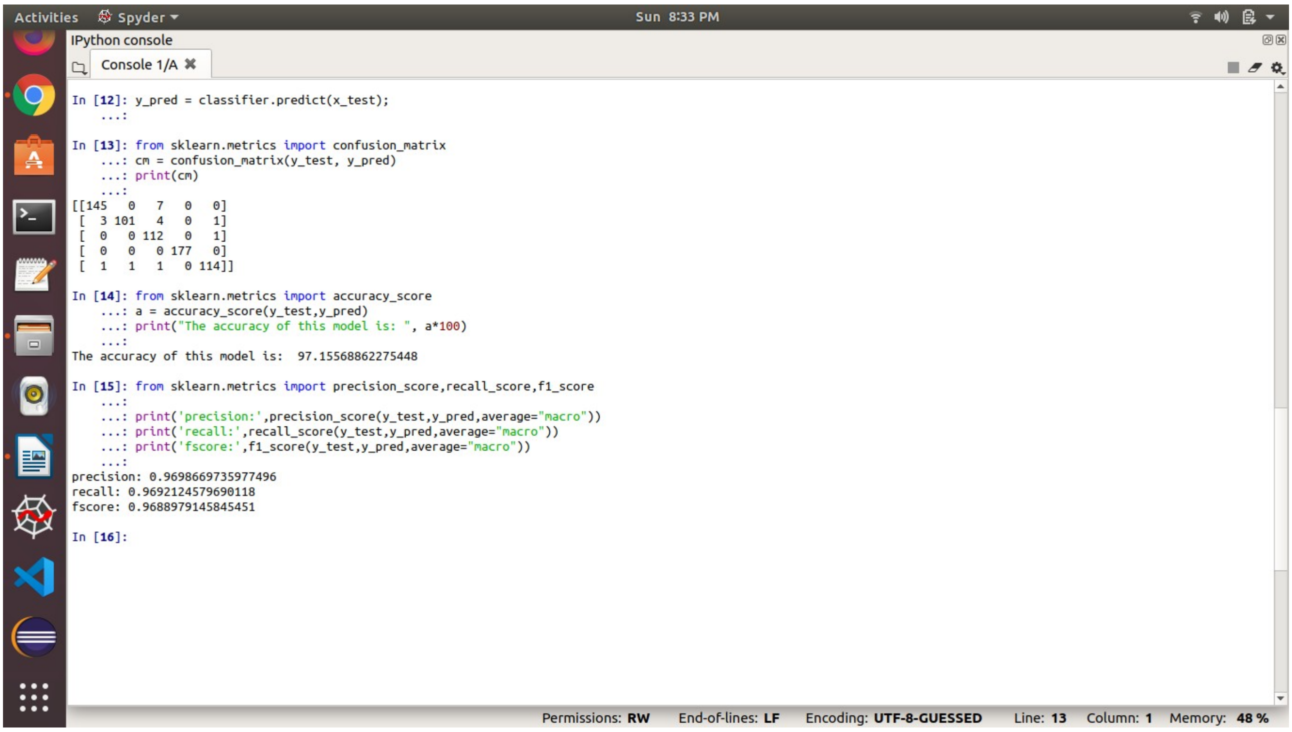
Output :



```
Python 3.6.9 (default, Oct  8 2020, 12:12:24)
Type "copyright", "credits" or "license" for more information.

IPython 5.5.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: import pandas as pd
   ...: import re
   ...: from nltk.corpus import stopwords
   ...: from nltk.stem.porter import PorterStemmer
   ...:

In [2]: dataset = pd.read_csv('bbc-text.csv')
   ...:

In [3]: stop_words = set(stopwords.words("english"))
   ...: new_stopwords = [',','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']
   ...: new_stopwords_list = stop_words.union(new_stopwords);
   ...:

In [4]: words = []
   ...:

In [5]: for i in range(0, 2225):
   ...:     text = re.sub('[^a-zA-Z]',' ', dataset['text'][i])
   ...:     text = text.lower()
   ...:     text = text.split()
   ...:     ps = PorterStemmer()
   ...:     text = [ps.stem(word) for word in text if not word in set(new_stopwords_list)]
   ...:     text = ' '.join(text)
   ...:     words.append(text)
   ...:
   ...:

In [6]: from sklearn.feature_extraction.text import TfidfVectorizer
   ...: tfidfvect = TfidfVectorizer(stop_words = new_stopwords_list);
   ...: x = tfidfvect.fit_transform(words).toarray()
   ...:
```



```
In [7]: tfidf_tokens = tfidfvect.get_feature_names()
   ...:

In [8]: df_tfidfvect = pd.DataFrame(data = x,columns = tfidf_tokens)
   ...:

In [9]: print("\nTF-IDF Vectorizer\n")
   ...: print(df_tfidfvect)
   ...:

TF-IDF Vectorizer

       aa  aaa  aac  aadc  ...  zurich  zuton  zvonareva  zvyagintsev
0     0.0  0.0  0.0   0.0  ...     0.0    0.0        0.0          0.0
1     0.0  0.0  0.0   0.0  ...     0.0    0.0        0.0          0.0
2     0.0  0.0  0.0   0.0  ...     0.0    0.0        0.0          0.0
3     0.0  0.0  0.0   0.0  ...     0.0    0.0        0.0          0.0
4     0.0  0.0  0.0   0.0  ...     0.0    0.0        0.0          0.0
...   ...  ...  ...   ...  ...     ...    ...        ...          ...
2220  0.0  0.0  0.0   0.0  ...     0.0    0.0        0.0          0.0
2221  0.0  0.0  0.0   0.0  ...     0.0    0.0        0.0          0.0
2222  0.0  0.0  0.0   0.0  ...     0.0    0.0        0.0          0.0
2223  0.0  0.0  0.0   0.0  ...     0.0    0.0        0.0          0.0
2224  0.0  0.0  0.0   0.0  ...     0.0    0.0        0.0          0.0

[2225 rows x 18956 columns]

In [10]: y = dataset['category']
    ...:
    ...: from sklearn.model_selection import train_test_split
    ...: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 0)
    ...:

In [11]: from sklearn.naive_bayes import MultinomialNB
    ...: classifier = MultinomialNB()
    ...: classifier.fit(x_train, y_train)
    ...:
Out[11]: MultinomialNB()

In [12]: y_pred = classifier.predict(x_test);
```



```
In [12]: y_pred = classifier.predict(x_test);
    ...:

In [13]: from sklearn.metrics import confusion_matrix
    ...: cm = confusion_matrix(y_test, y_pred)
    ...: print(cm)
    ...:
[[145   0   7   0   0]
 [  3 101   4   0   1]
 [  0   0 112   0   1]
 [  0   0   0 177   0]
 [  1   1   1   0 114]]

In [14]: from sklearn.metrics import accuracy_score
    ...: a = accuracy_score(y_test,y_pred)
    ...: print("The accuracy of this model is: ", a*100)
    ...:
The accuracy of this model is:  97.15568862275448

In [15]: from sklearn.metrics import precision_score,recall_score,f1_score
    ...:
    ...: print('precision:',precision_score(y_test,y_pred,average="macro"))
    ...: print('recall:',recall_score(y_test,y_pred,average="macro"))
    ...: print('fscore:',f1_score(y_test,y_pred,average="macro"))
    ...:
precision: 0.9698669735977496
recall: 0.9692124579690118
fscore: 0.9688979145845451

In [16]:
```