vectorization and model

May 31, 2020

[2]:
```
import pandas as pd
from  sklearn.model_selection import train_test_split
from  sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score
from sklearn.metrics import *

#Load dataset from CSV file
df=pd.read_csv('mal_corpus.
 ↪csv',names=['sentence','ambiguous_word','label','sense'],skiprows=1)
df_x=df["sentence"]
df_y=df['label']

#vectorization and data splitting
cv = TfidfVectorizer(input="content",encoding="utf-8",norm="l2")
x_train,x_test,y_train,y_test=train_test_split(df_x,df_y,test_size=0.
 ↪23,random_state=11)
xtrain_cv=cv.fit_transform(x_train)
xtest_cv=cv.transform(x_test)

clf = LinearSVC(penalty='l2', loss='squared_hinge', dual=True, tol=0.0001, C=1.
 ↪0, multi_class='ovr', fit_intercept=True, intercept_scaling=1,␣
 ↪class_weight=None, verbose=0, random_state=None, max_iter=1000)
y_train=y_train.astype('int')
y_test=y_test.astype('int')

#Model Training using training set data
clf.fit(xtrain_cv,y_train)

#Predicting the label of test set data
pred=clf.predict(xtest_cv)

#Calculating Accuracy
Accuracy=accuracy_score(y_test,pred)
Accuracy=Accuracy*100
Accuracy
```

```
[2]: 73.07692307692307
```

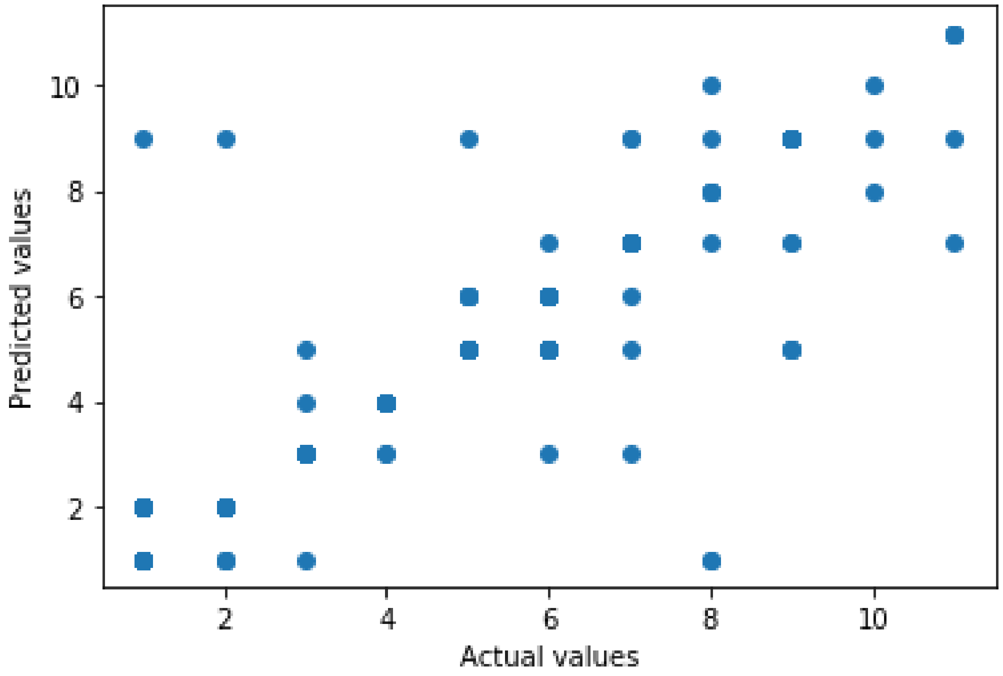
```
[ ]: #Plotting the model
```

```
[3]: from matplotlib import pyplot as plt
     plt.scatter(y_test, pred)

     plt.xlabel("Actual values")
     plt.ylabel("Predicted values")
     print("\n\nAccuracy Score : ")
     print(accuracy_score(y_test,pred ))
```

```


Accuracy Score :
0.7307692307692307
```



```
[ ]: #Printing Classification Report
```

```
[35]: print(classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           1       0.70      0.56      0.62        25
           2       0.66      0.83      0.73        23
           3       0.87      0.90      0.89        30
           4       0.95      0.91      0.93        22
           5       0.70      0.89      0.78        54
           6       0.65      0.39      0.49        28
           7       0.72      0.72      0.72        18
           8       0.80      0.44      0.57         9
           9       0.56      0.62      0.59        16
          10       0.50      0.33      0.40         3
          11       1.00      0.67      0.80         6

    accuracy                           0.73       234
   macro avg       0.74      0.66      0.68       234
weighted avg       0.73      0.73      0.72       234
```

[ ]:
```
#Confusion Matrix
```

[4]:
```
print(confusion_matrix(y_test,pred))
```

```
[[14 10  0  0  0  0  0  0  1  0  0]
 [ 3 19  0  0  0  0  0  0  1  0  0]
 [ 1  0 27  1  1  0  0  0  0  0  0]
 [ 0  0  2 20  0  0  0  0  0  0  0]
 [ 0  0  0  0 48  5  0  0  1  0  0]
 [ 0  0  1  0 15 11  1  0  0  0  0]
 [ 0  0  1  0  1  1 13  0  2  0  0]
 [ 2  0  0  0  0  0  1  4  1  1  0]
 [ 0  0  0  0  4  0  2  0 10  0  0]
 [ 0  0  0  0  0  0  0  1  1  1  0]
 [ 0  0  0  0  0  0  1  0  1  0  4]]
```

[ ]:
```

```