knn

December 9, 2023

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

[ ]:
```
data=pd.read_csv('dataset2.csv')
```

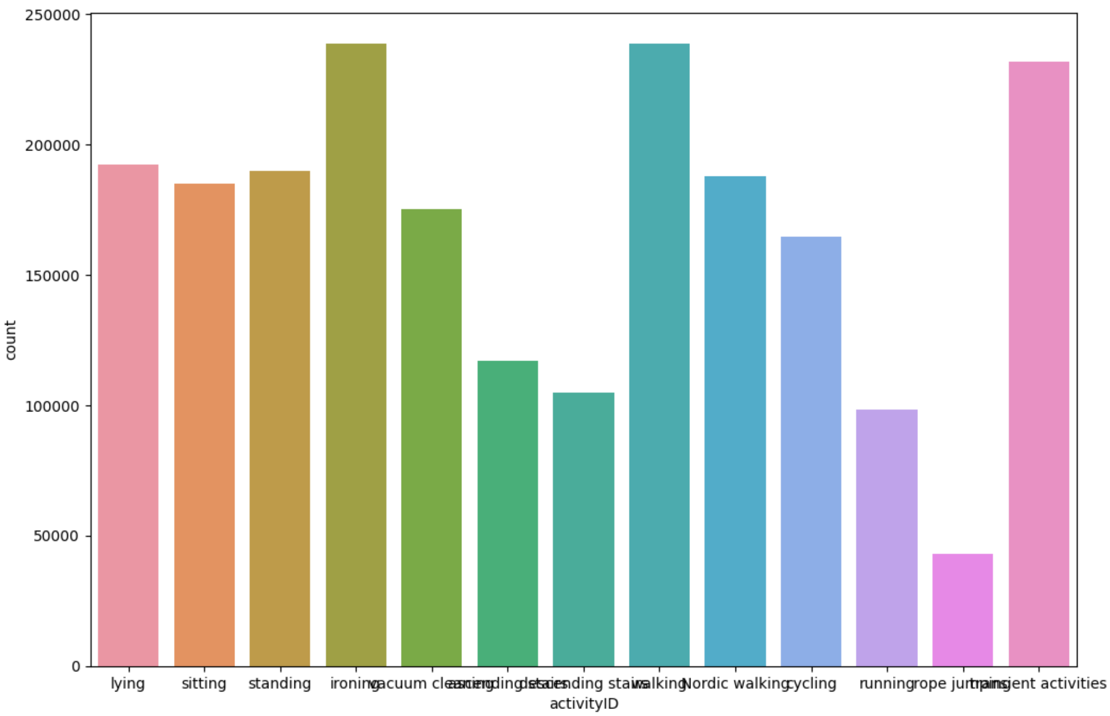
[ ]:
```
data=data.dropna()
```

[ ]:
```
data=data.drop(['PeopleId'],axis=1)
```

[ ]:
```
#filter data if activity id is transient activities
data_transient = data[data['activityID'] == 'transient activities']
#reduce number to 0.25 percent
data_transient = data_transient.sample(frac=0.25)
#add data_transient back to data
data = data[data['activityID'] != 'transient activities']
data = pd.concat([data,data_transient])
```

[ ]:
```
#plot activityID distributiom
plt.figure(figsize=(20,5))
sns.countplot(data['activityID'])
plt.show()
```

```
/Users/franklin/opt/anaconda3/lib/python3.9/site-
packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will
be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
```



[ ]:

```
#encode activityID
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
data['activityID']=le.fit_transform(data['activityID'])
```

[ ]:

```
#print(data.info())
data=data.sample(frac=0.1)
y=data["activityID"]
X=data.drop(["activityID"],axis=1)
#split the data into training, validation and testing sets
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.
 ↪2,stratify=y,random_state=42)
```

[ ]:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

accuracy_scores=[]
for k in range(4,11):
    knn=KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train,y_train)
    y_pred=knn.predict(X_test)
    accuracy=accuracy_score(y_test,y_pred)
    accuracy_scores.append(accuracy)
```

```
print(accuracy_scores)
```

```
[0.9569267662792843, 0.9531682346430548, 0.9485334809075816, 0.9458356391809629,
0.9421923999262128, 0.9400710201069913, 0.9372117690463014]
```

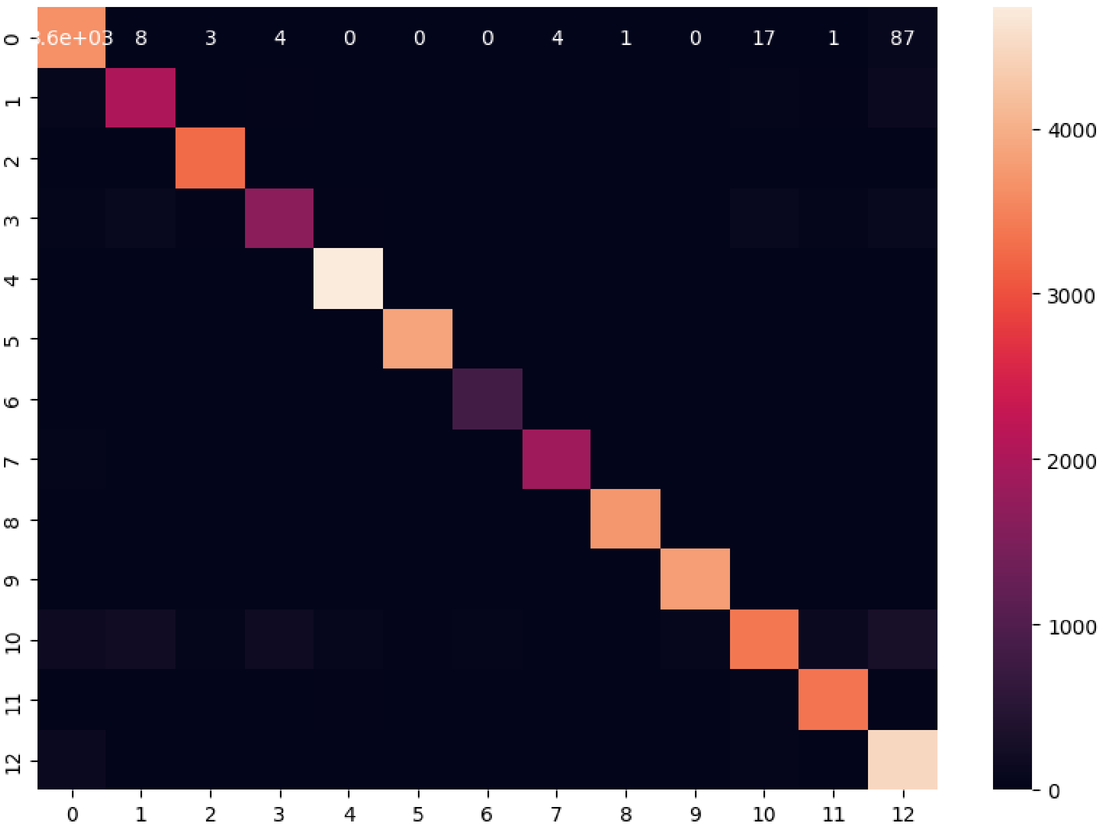
[ ]:

```
import pickle
with open('knn_model.pkl', 'rb') as f:
    knn = pickle.load(f)

y_pred_knn=knn.predict(X_test)
print(accuracy_score(y_test,y_pred_knn))
```

[ ]:

```
import matplotlib.pyplot as plt
import seaborn as sn
from sklearn.metrics import confusion_matrix,classification_report

plt.figure(figsize = (10,7))
sn.heatmap(confusion_matrix(y_test,y_pred), annot=True)

print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.88      0.97      0.92      3766
           1       0.85      0.87      0.86      2336
           2       0.97      0.99      0.98      3291
           3       0.88      0.79      0.84      2116
           4       0.97      0.99      0.98      4767
           5       0.99      1.00      1.00      3872
           6       0.94      0.98      0.96       852
           7       0.99      0.95      0.97      1965
           8       0.99      1.00      0.99      3735
           9       0.98      1.00      0.99      3815
          10       0.92      0.73      0.81      4630
          11       0.94      0.96      0.95      3490
          12       0.87      0.95      0.91      4733

    accuracy                           0.94     43368
   macro avg       0.94      0.94      0.93     43368
weighted avg       0.94      0.94      0.94     43368
```


.6e+03
8
3
4
0
0
0
4
1
0
17
1
87
0
1
2
3
4
5
6
7
8
9
10
11
12
0
1
2
3
4
5
6
7
8
9
10
11
12
4000
3000
2000
1000
0

```
[ ]:
```