MACHINE LEARNING LABORATORY

1818128_JAYASREE T

☆ EXP 5 :Write a program to classify data using the K-Nearest Neighbors algorithm in Python using breast cancer dataset from the sklearn.datasets module

Importing the Dataset

```
[7] import numpy as np
    import pandas as pd
    from sklearn.datasets import load_breast_cancer
    data = load_breast_cancer()
    X=data['data']
    y=data['target']
```

```
[10] print(data.keys())
```

1818128_JAYASREE T_EXP 5

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[10] print(data.keys())
```

```
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'filename'])
```

```
[11] df = pd.DataFrame(data.data, columns=data.feature_names)
     df['target'] = data.target
     df.head()
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	radius error	texture error	perimeter error	area error	smoo
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	1.0950	0.9053	8.589	153.40	0.
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	0.4956	1.1560	3.445	27.23	0.
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.

Splitting Dataset into training and testing sets

```
[9] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
```

1818128_JAYASREE T_EXP 5

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Splitting Dataset into training 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=0.25, random_state=42)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Accuracy of the model

[12]
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV
parameters = {'n_neighbors':[3,5,8,10], 'p':[1, 2],'algorithm':('auto', 'ball_tree', 'kd_tree', 'brute'),'leaf_size':[20,3
knn=KNeighborsClassifier()
model= GridSearchCV(knn, parameters)
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
from sklearn.metrics import accuracy_score
print("Accuracy: ",accuracy_score(y_test,y_pred))
```

```
Accuracy:  0.972027972027972
```