

task-3-knn

August 6, 2024

```
[3]: #Importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import precision_score, recall_score, accuracy_score
```

```
[4]: #
df = pd.read_csv("breast_cancer.csv")
df.head()
```

```
[4]:
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	\
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	

	smoothness_mean	compactness_mean	concavity_mean	concave_points_mean	\
0	0.11840	0.27760	0.3001	0.14710	
1	0.08474	0.07864	0.0869	0.07017	
2	0.10960	0.15990	0.1974	0.12790	
3	0.14250	0.28390	0.2414	0.10520	
4	0.10030	0.13280	0.1980	0.10430	

...	radius_worst	texture_worst	perimeter_worst	area_worst	\
0	...	25.38	17.33	184.60	2019.0
1	...	24.99	23.41	158.80	1956.0
2	...	23.57	25.53	152.50	1709.0
3	...	14.91	26.50	98.87	567.7
4	...	22.54	16.67	152.20	1575.0

	smoothness_worst	compactness_worst	concavity_worst	concave_points_worst	\
0	0.1622	0.6656	0.7119	0.2654	

1	0.1238	0.1866	0.2416	0.1860
2	0.1444	0.4245	0.4504	0.2430
3	0.2098	0.8663	0.6869	0.2575
4	0.1374	0.2050	0.4000	0.1625

	symmetry_worst	fractal_dimension_worst
0	0.4601	0.11890
1	0.2750	0.08902
2	0.3613	0.08758
3	0.6638	0.17300
4	0.2364	0.07678

[5 rows x 32 columns]

```
[5]: #Dropping the ID Column
df.drop('id', axis = 1, inplace = True)
df.head()
```

```
[5]:  diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0         M        17.99        10.38        122.80        1001.0
1         M        20.57        17.77        132.90        1326.0
2         M        19.69        21.25        130.00        1203.0
3         M        11.42        20.38         77.58         386.1
4         M        20.29        14.34        135.10        1297.0
```

	smoothness_mean	compactness_mean	concavity_mean	concave_points_mean	\
0	0.11840	0.27760	0.3001	0.14710	
1	0.08474	0.07864	0.0869	0.07017	
2	0.10960	0.15990	0.1974	0.12790	
3	0.14250	0.28390	0.2414	0.10520	
4	0.10030	0.13280	0.1980	0.10430	

	symmetry_mean	...	radius_worst	texture_worst	perimeter_worst	\
0	0.2419	...	25.38	17.33	184.60	
1	0.1812	...	24.99	23.41	158.80	
2	0.2069	...	23.57	25.53	152.50	
3	0.2597	...	14.91	26.50	98.87	
4	0.1809	...	22.54	16.67	152.20	

	area_worst	smoothness_worst	compactness_worst	concavity_worst	\
0	2019.0	0.1622	0.6656	0.7119	
1	1956.0	0.1238	0.1866	0.2416	
2	1709.0	0.1444	0.4245	0.4504	
3	567.7	0.2098	0.8663	0.6869	
4	1575.0	0.1374	0.2050	0.4000	

	concave_points_worst	symmetry_worst	fractal_dimension_worst
--	----------------------	----------------	-------------------------

0	0.2654	0.4601	0.11890
1	0.1860	0.2750	0.08902
2	0.2430	0.3613	0.08758
3	0.2575	0.6638	0.17300
4	0.1625	0.2364	0.07678

[5 rows x 31 columns]

```
[8]: df.columns
```

```
[8]: Index(['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
          'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
          'concave_points_mean', 'symmetry_mean', 'fractal_dimension_mean',
          'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
          'compactness_se', 'concavity_se', 'concave_points_se', 'symmetry_se',
          'fractal_dimension_se', 'radius_worst', 'texture_worst',
          'perimeter_worst', 'area_worst', 'smoothness_worst',
          'compactness_worst', 'concavity_worst', 'concave_points_worst',
          'symmetry_worst', 'fractal_dimension_worst'],
          dtype='object')
```

```
[12]: #Applying Min-Max scaling to the selected numerical columns
num_columns = ['radius_mean', 'texture_mean', 'perimeter_mean',
               'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
               'concave_points_mean', 'symmetry_mean', 'fractal_dimension_mean',
               'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
               'compactness_se', 'concavity_se', 'concave_points_se', 'symmetry_se',
               'fractal_dimension_se', 'radius_worst', 'texture_worst',
               'perimeter_worst', 'area_worst', 'smoothness_worst',
               'compactness_worst', 'concavity_worst', 'concave_points_worst',
               'symmetry_worst', 'fractal_dimension_worst']
df[num_columns] = (df[num_columns] - df[num_columns].min()) / (df[num_columns].
    ↪max() - df[num_columns].min())
df.head()
```

```
[12]:
```

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	\
0	M	0.521037	0.022658	0.545989	0.363733	
1	M	0.643144	0.272574	0.615783	0.501591	
2	M	0.601496	0.390260	0.595743	0.449417	
3	M	0.210090	0.360839	0.233501	0.102906	
4	M	0.629893	0.156578	0.630986	0.489290	

	smoothness_mean	compactness_mean	concavity_mean	concave_points_mean	\
0	0.593753	0.792037	0.703140	0.731113	
1	0.289880	0.181768	0.203608	0.348757	
2	0.514309	0.431017	0.462512	0.635686	
3	0.811321	0.811361	0.565604	0.522863	

4	0.430351	0.347893	0.463918	0.518390
---	----------	----------	----------	----------

	symmetry_mean	...	radius_worst	texture_worst	perimeter_worst	\
0	0.686364	...	0.620776	0.141525	0.668310	
1	0.379798	...	0.606901	0.303571	0.539818	
2	0.509596	...	0.556386	0.360075	0.508442	
3	0.776263	...	0.248310	0.385928	0.241347	
4	0.378283	...	0.519744	0.123934	0.506948	

	area_worst	smoothness_worst	compactness_worst	concavity_worst	\
0	0.450698	0.601136	0.619292	0.568610	
1	0.435214	0.347553	0.154563	0.192971	
2	0.374508	0.483590	0.385375	0.359744	
3	0.094008	0.915472	0.814012	0.548642	
4	0.341575	0.437364	0.172415	0.319489	

	concave_points_worst	symmetry_worst	fractal_dimension_worst
0	0.912027	0.598462	0.418864
1	0.639175	0.233590	0.222878
2	0.835052	0.403706	0.213433
3	0.884880	1.000000	0.773711
4	0.558419	0.157500	0.142595

[5 rows x 31 columns]

```
[13]: #Initializing the label encoder
label_encoder = LabelEncoder()
df['diagnosis'] = label_encoder.fit_transform(df['diagnosis'])
df.head()
```

[13]:	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	\
0	1	0.521037	0.022658	0.545989	0.363733	
1	1	0.643144	0.272574	0.615783	0.501591	
2	1	0.601496	0.390260	0.595743	0.449417	
3	1	0.210090	0.360839	0.233501	0.102906	
4	1	0.629893	0.156578	0.630986	0.489290	

	smoothness_mean	compactness_mean	concavity_mean	concave_points_mean	\
0	0.593753	0.792037	0.703140	0.731113	
1	0.289880	0.181768	0.203608	0.348757	
2	0.514309	0.431017	0.462512	0.635686	
3	0.811321	0.811361	0.565604	0.522863	
4	0.430351	0.347893	0.463918	0.518390	

	symmetry_mean	...	radius_worst	texture_worst	perimeter_worst	\
0	0.686364	...	0.620776	0.141525	0.668310	
1	0.379798	...	0.606901	0.303571	0.539818	

2	0.509596	...	0.556386	0.360075	0.508442
3	0.776263	...	0.248310	0.385928	0.241347
4	0.378283	...	0.519744	0.123934	0.506948

	area_worst	smoothness_worst	compactness_worst	concavity_worst	\
0	0.450698	0.601136	0.619292	0.568610	
1	0.435214	0.347553	0.154563	0.192971	
2	0.374508	0.483590	0.385375	0.359744	
3	0.094008	0.915472	0.814012	0.548642	
4	0.341575	0.437364	0.172415	0.319489	

	concave_points_worst	symmetry_worst	fractal_dimension_worst
0	0.912027	0.598462	0.418864
1	0.639175	0.233590	0.222878
2	0.835052	0.403706	0.213433
3	0.884880	1.000000	0.773711
4	0.558419	0.157500	0.142595

[5 rows x 31 columns]

```
[16]: #Extracting the features by removing 'diagnosis' column and assigning it to the
      ↪target variable
X = df.drop('diagnosis', axis = 1)
y = df['diagnosis']
print(x)
```

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	\
0	0.521037	0.022658	0.545989	0.363733	0.593753	
1	0.643144	0.272574	0.615783	0.501591	0.289880	
2	0.601496	0.390260	0.595743	0.449417	0.514309	
3	0.210090	0.360839	0.233501	0.102906	0.811321	
4	0.629893	0.156578	0.630986	0.489290	0.430351	
..	
564	0.690000	0.428813	0.678668	0.566490	0.526948	
565	0.622320	0.626987	0.604036	0.474019	0.407782	
566	0.455251	0.621238	0.445788	0.303118	0.288165	
567	0.644564	0.663510	0.665538	0.475716	0.588336	
568	0.036869	0.501522	0.028540	0.015907	0.000000	

	compactness_mean	concavity_mean	concave_points_mean	symmetry_mean	\
0	0.792037	0.703140	0.731113	0.686364	
1	0.181768	0.203608	0.348757	0.379798	
2	0.431017	0.462512	0.635686	0.509596	
3	0.811361	0.565604	0.522863	0.776263	
4	0.347893	0.463918	0.518390	0.378283	
..	
564	0.296055	0.571462	0.690358	0.336364	

565	0.257714	0.337395	0.486630	0.349495
566	0.254340	0.216753	0.263519	0.267677
567	0.790197	0.823336	0.755467	0.675253
568	0.074351	0.000000	0.000000	0.266162

	fractal_dimension_mean	...	radius_worst	texture_worst	\
0	0.605518	...	0.620776	0.141525	
1	0.141323	...	0.606901	0.303571	
2	0.211247	...	0.556386	0.360075	
3	1.000000	...	0.248310	0.385928	
4	0.186816	...	0.519744	0.123934	
..	
564	0.132056	...	0.623266	0.383262	
565	0.113100	...	0.560655	0.699094	
566	0.137321	...	0.393099	0.589019	
567	0.425442	...	0.633582	0.730277	
568	0.187026	...	0.054287	0.489072	

	perimeter_worst	area_worst	smoothness_worst	compactness_worst	\
0	0.668310	0.450698	0.601136	0.619292	
1	0.539818	0.435214	0.347553	0.154563	
2	0.508442	0.374508	0.483590	0.385375	
3	0.241347	0.094008	0.915472	0.814012	
4	0.506948	0.341575	0.437364	0.172415	
..	
564	0.576174	0.452664	0.461137	0.178527	
565	0.520892	0.379915	0.300007	0.159997	
566	0.379949	0.230731	0.282177	0.273705	
567	0.668310	0.402035	0.619626	0.815758	
568	0.043578	0.020497	0.124084	0.036043	

	concavity_worst	concave_points_worst	symmetry_worst	\
0	0.568610	0.912027	0.598462	
1	0.192971	0.639175	0.233590	
2	0.359744	0.835052	0.403706	
3	0.548642	0.884880	1.000000	
4	0.319489	0.558419	0.157500	
..	
564	0.328035	0.761512	0.097575	
565	0.256789	0.559450	0.198502	
566	0.271805	0.487285	0.128721	
567	0.749760	0.910653	0.497142	
568	0.000000	0.000000	0.257441	

	fractal_dimension_worst
0	0.418864
1	0.222878
2	0.213433

3	0.773711
4	0.142595
..	...
564	0.105667
565	0.074315
566	0.151909
567	0.452315
568	0.100682

[569 rows x 30 columns]

```
[17]: #Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
↳ random_state = 42)
```

```
[19]: #Training the KNN Model
k = 5
knn = KNeighborsClassifier(n_neighbors = k)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

precision = precision_score(y_test, y_pred)
print("Precision:", precision)

recall = recall_score(y_test, y_pred)
print("Recall:", recall)
```

Accuracy: 0.9649122807017544
Precision: 0.9534883720930233
Recall: 0.9534883720930233

```
[ ]:
```