practical-02-MIDS

March 27, 2025

[ ]:

**import pandas as pd import numpy as np import seaborn as sns**

**import matplotlib.pyplot as plt**

**from sklearn.model\_selection import** train\_test\_split

**from sklearn.preprocessing import** LabelEncoder, StandardScaler

**from sklearn.neighbors import** KNeighborsClassifier

**from sklearn.metrics import** accuracy\_score, confusion\_matrix,␣

𝗌classification\_report

[2]:

df = sns.load\_dataset('titanic')

[8]:

df = df[['survived', 'pclass', 'sex', 'age', 'fare', 'embarked']]

[9]:

df = df.assign( age=df['age'].fillna(df['age'].median()),

embarked=df['embarked'].fillna(df['embarked'].mode()[0])

)

[10]:

encoder = LabelEncoder()

df['sex'] = encoder.fit\_transform(df['sex']) df['embarked'] = encoder.fit\_transform(df['embarked'])

[11]:

X = df.drop(columns=['survived']) y = df['survived']

[12]:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,␣

𝗌random\_state=42)

[13]:

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train) X\_test = scaler.transform(X\_test)

[14]:

*# Train KNN model*

knn = KNeighborsClassifier(n\_neighbors=5) *# Using 5 neighbors*

knn.fit(X\_train, y\_train)

[14]: KNeighborsClassifier()

[17]:

*# Predictions*

y\_pred = knn.predict(X\_test)

[16]:

*# Evaluate model performance*

accuracy = accuracy\_score(y\_test, y\_pred) print(f'Accuracy: **{**accuracy**:**.2f**}**')

print('**\n**Confusion Matrix:') print(confusion\_matrix(y\_test, y\_pred))

print('**\n**Classification Report:') print(classification\_report(y\_test, y\_pred))

Accuracy: 0.78

Confusion Matrix: [[84 21]

[19 55]]

Classification Report:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | precision | recall | f1-score | support |
| 0 | 0.82 | 0.80 | 0.81 | 105 |
| 1 | 0.72 | 0.74 | 0.73 | 74 |
| accuracy |  |  | 0.78 | 179 |
| macro avg | 0.77 | 0.77 | 0.77 | 179 |
| weighted avg | 0.78 | 0.78 | 0.78 | 179 |
| [ ]: |  |  |  |  |  |