ml-classification-algorithms

April 11, 2025

[5]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

[6]:
```
import pandas as pd


data = {
    'Age': [45, 50,35,60,40,55,25,48],
    'Glucose': [180,155,100,200,130,190,85,170],
    'BMI': [28.5,30.1,24.3,32.0,27.5,31.4,22.1,29.7],
    'BloodPressure': [80, 75,68,85,70,78,65,82],
    'Diabetes': [1, 1, 0, 1, 0, 1, 0, 1]
}
df = pd.DataFrame(data)


print(df)
```

```
   Age  Glucose   BMI  BloodPressure  Diabetes
0   45      180  28.5             80         1
1   50      155  30.1             75         1
2   35      100  24.3             68         0
3   60      200  32.0             85         1
4   40      130  27.5             70         0
5   55      190  31.4             78         1
6   25       85  22.1             65         0
7   48      170  29.7             82         1
```

[7]:
```
1.#LOGISTIC REGRESSION
X = df[['Age', 'Glucose', 'BMI', 'BloodPressure']]  # input features
y = df['Diabetes']
```

[8]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
↪random_state=42)
```

[9]:
```
model = LogisticRegression()
```

[10]:
```
model.fit(X_train, y_train)
```

[10]:
```
LogisticRegression()
```

[11]:
```
y_pred = model.predict(X_test)
```

[12]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

[13]:
```
print("Predictions:", y_pred)
```

```
Predictions: [1 1 1]
```

[14]:
```
#2.KNN
from sklearn.neighbors import KNeighborsClassifier
```

[15]:
```
X = df[['Age', 'Glucose', 'BMI', 'BloodPressure']]  # input features
y = df['Diabetes']
```

[16]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
↪random_state=42)
```

[18]:
```
model = KNeighborsClassifier(n_neighbors=3)
```

[19]:
```
model.fit(X_train, y_train)
```

[19]:
```
KNeighborsClassifier(n_neighbors=3)
```

[21]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

[22]:
```
print("Predictions:", y_pred)
```

```
Predictions: [1 1 1]
```

[64]:
```
3.#SVM
from sklearn.svm import SVC
```

[65]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
↪random_state=42)
```

[66]:
```
model = SVC()
```

[67]:
```
model.fit(X_train, y_train)
```

[67]:
```
SVC()
```

[68]:
```
y_pred = model.predict(X_test)
```

[69]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

[70]:
```
print("Predictions:", y_pred)
```

```
Predictions: [1 1 1]
```

[71]:
```
4.#DECISION TREE
from sklearn.tree import DecisionTreeClassifier
```

[52]:
```
X = df[['Age', 'Glucose', 'BMI', 'BloodPressure']]
y = df['Diabetes']
```

[53]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
↪random_state=42)
```

[54]:
```
model = DecisionTreeClassifier()
```

[55]:
```
model.fit(X_train, y_train)
```

[55]:
```
DecisionTreeClassifier()
```

[56]:
```
y_pred = model.predict(X_test)
```

[57]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

[58]:
```
print("Predictions:", y_pred)
```

```
Predictions: [1 1 1]
```

[59]:
```
#5.RANDOM FOREST
from sklearn.ensemble import RandomForestClassifier
```

[60]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
↪random_state=42)
```

[61]:
```
model = RandomForestClassifier(n_estimators=100, random_state=42)
```

[62]:
```
model.fit(X_train, y_train)
```

[62]:
```
RandomForestClassifier(random_state=42)
```

[63]:
```
y_pred = model.predict(X_test)
```

[49]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

[50]:
```
print("Predictions:", y_pred)
```

```
Predictions: [1 1 1]
```

[ ]:
```

```