student-pass

June 27, 2024

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import model_selection as train_test_split
from sklearn import neighbors as KNeighborsClassifier
from sklearn import metrics as accuracy_score
```

[ ]:
```
import pandas as pd  # Import pandas and give it the alias 'pd'

data_bmi = pd.read_csv("/content/drive/MyDrive/body _mass/student_data.csv")
```

[ ]:
```
x=data_bmi[['Hours_Study','Hours_Sleep']]
y=data_bmi['Pass']
```

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split  # Import the class␣
 ↪disrectly
from sklearn.neighbors import KNeighborsClassifier  # Import the class directly
from sklearn.metrics import accuracy_score

# ... (rest of your code)

k = 5
knn = KNeighborsClassifier(n_neighbors=k)  # Now you're using the class␣
 ↪correctly
knn.fit(x, y)
```

[ ]:
```
KNeighborsClassifier()
```

[ ]:
```
new_data = np.array([[2,9]])
prediction = knn.predict(new_data)



if prediction==0:
```

```
    print("fail")
else: # Fixed indentation here
    print("pass")
```

```
fail
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but KNeighborsClassifier was fitted with feature
names
  warnings.warn(
```

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[29]:
```
import numpy as np
from sklearn.linear_model import LinearRegression

# ... (Load your data here) ...

# Create and train the LinearRegression model
model = LinearRegression() # Create the model object
model.fit(x, y)  # Train the model (replace x and y with your training data)

# Get user input
hours_study = float(input("Enter the number of hours studied: "))
hours_sleep = float(input("Enter the number of hours slept: "))

# Create input array for prediction
new_data = np.array([[hours_study, hours_sleep]])

# Make prediction
prediction = model.predict(new_data)

# Interpret and print the prediction
if prediction >= 0.5:
    print("Prediction: Pass")
else:
    print("Prediction: Fail")
```

```
Enter the number of hours studied: 8
Enter the number of hours slept: 6
Prediction: Pass
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

[30]:
```
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Assuming 'x' and 'y' are your features and target variable respectively
# If not, load your data here and define x and y

# Split 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)

# Create and train the LogisticRegression model
model = LogisticRegression()
model.fit(x_train, y_train)

# Make predictions on the test set
y_pred = model.predict(x_test)

# Calculate and print the accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Get user input
hours_study = float(input("Enter the number of hours studied: "))
hours_sleep = float(input("Enter the number of hours slept: "))

# Create input array for prediction
new_data = np.array([[hours_study, hours_sleep]])

# Make prediction
prediction = model.predict(new_data)

# Interpret and print the prediction
if prediction[0] == 1:  # Access the prediction result from the array
    print("Prediction: Pass")
else:
    print("Prediction: Fail")
```

```
Accuracy: 0.5
Enter the number of hours studied: 2
Enter the number of hours slept: 9
Prediction: Fail
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but LogisticRegression was fitted with feature
names
  warnings.warn(
```

[31]:
```
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Assuming 'x' and 'y' are your features and target variable respectively
# If not, load your data here and define x and y

# Split 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)

# Create and train the Decision Tree Classifier model
model = DecisionTreeClassifier()
model.fit(x_train, y_train)

# Make predictions on the test set
y_pred = model.predict(x_test)

# Calculate and print the accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Get user input
hours_study = float(input("Enter the number of hours studied: "))
hours_sleep = float(input("Enter the number of hours slept: "))

# Create input array for prediction
new_data = np.array([[hours_study, hours_sleep]])

# Make prediction
prediction = model.predict(new_data)

# Interpret and print the prediction
if prediction[0] == 1:
    print("Prediction: Pass")
else:
    print("Prediction: Fail")
```

```
Accuracy: 1.0
Enter the number of hours studied: 8
Enter the number of hours slept: 6
Prediction: Pass
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but DecisionTreeClassifier was fitted with feature
names
  warnings.warn(
```

[28]:
```
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Assuming 'x' and 'y' are your features and target variable respectively
# If not, load your data here and define x and y

# Split 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)

# Create and train the Random Forest Classifier model
model = RandomForestClassifier(n_estimators=100)  # You can adjust n_estimators␣
 ↪(number of trees)
model.fit(x_train, y_train)

# Make predictions on the test set
y_pred = model.predict(x_test)

# Calculate and print the accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Get user input
hours_study = float(input("Enter the number of hours studied: "))
hours_sleep = float(input("Enter the number of hours slept: "))

# Create input array for prediction
new_data = np.array([[hours_study, hours_sleep]])

# Make prediction
prediction = model.predict(new_data)

# Interpret and print the prediction
if prediction[0] == 1:
    print("Prediction: Pass")
else:
    print("Prediction: Fail")
```

```
Accuracy: 0.5
Enter the number of hours studied: 2
Enter the number of hours slept: 9
Prediction: Fail
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but RandomForestClassifier was fitted with feature
names
```

```
warnings.warn(
```