Desh Iyer | 500081889 | Experiment - 9

April 9, 2023

1 Experiment 9 - Implement the K-Nearest Neighbours Algorithm on the Iris dataset

1.1 Import libraries and data

[ ]:
```
# Import libraries
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

# Load dataset
iris = load_iris()
X = iris.data
y = iris.target
```

1.2 Train-test Split

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

1.3 Fitting data to a KNN Model

1.3.1 Determining the optimal K value through the elbow method

[ ]:
```
# Determine the best value for K using the elbow method
k_range = range(1, 21)
scores = []
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    scores.append(knn.score(X_test, y_test))

best_k = k_range[scores.index(max(scores))]
```

[ ]:
```
print(f'Scores: {scores}\nOptimal K value: {best_k}')
```

```
Scores: [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
Optimal K value: 1
```

1.3.2 Creating the model

[ ]:
```
knn = KNeighborsClassifier(n_neighbors=best_k)
```

1.3.3 Fitting the data

[ ]:
```
knn.fit(X_train, y_train)
```

[ ]:
```
KNeighborsClassifier(n_neighbors=1)
```

1.4 Make predictions on the testing data

[ ]:
```
y_pred = knn.predict(X_test)
```

1.5 Printing wrong and right predictions

[ ]:
```
# Print correct and wrong predictions
for i in range(len(y_pred)):
    if y_pred[i] == y_test[i]:
        print(f"Correct prediction: actual={y_test[i]}, predicted={y_pred[i]}")
    else:
        print(f"Wrong prediction: actual={y_test[i]}, predicted={y_pred[i]}")

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

```
Correct prediction: actual=1, predicted=1
Correct prediction: actual=0, predicted=0
Correct prediction: actual=2, predicted=2
Correct prediction: actual=1, predicted=1
Correct prediction: actual=1, predicted=1
Correct prediction: actual=0, predicted=0
Correct prediction: actual=1, predicted=1
Correct prediction: actual=2, predicted=2
Correct prediction: actual=1, predicted=1
Correct prediction: actual=1, predicted=1
Correct prediction: actual=2, predicted=2
Correct prediction: actual=0, predicted=0
Correct prediction: actual=0, predicted=0
Correct prediction: actual=0, predicted=0
Correct prediction: actual=0, predicted=0
```

```
Correct prediction: actual=1, predicted=1
Correct prediction: actual=2, predicted=2
Correct prediction: actual=1, predicted=1
Correct prediction: actual=1, predicted=1
Correct prediction: actual=2, predicted=2
Correct prediction: actual=0, predicted=0
Correct prediction: actual=2, predicted=2
Correct prediction: actual=0, predicted=0
Correct prediction: actual=2, predicted=2
Correct prediction: actual=2, predicted=2
Correct prediction: actual=2, predicted=2
Correct prediction: actual=2, predicted=2
Correct prediction: actual=2, predicted=2
Correct prediction: actual=0, predicted=0
Correct prediction: actual=0, predicted=0
Correct prediction: actual=0, predicted=0
Correct prediction: actual=0, predicted=0
Correct prediction: actual=1, predicted=1
Correct prediction: actual=0, predicted=0
Correct prediction: actual=0, predicted=0
Correct prediction: actual=2, predicted=2
Correct prediction: actual=1, predicted=1
Correct prediction: actual=0, predicted=0
Correct prediction: actual=0, predicted=0
Correct prediction: actual=0, predicted=0
Correct prediction: actual=2, predicted=2
Correct prediction: actual=1, predicted=1
Correct prediction: actual=1, predicted=1
Correct prediction: actual=0, predicted=0
Correct prediction: actual=0, predicted=0
Accuracy: 1.0
```

1.6 Conclusion

The iris dataset has relatively few features and classes, with a clear separation between them. This means that the decision boundaries for a KNN classifier will be relatively simple, leading to high accuracy. A more complex dataset with overlapping classes and more ambiguous feature relationships would pose a greater challenge for a KNN classifier.