

Machine Learning
Assignment 9
Aakanksha Darekar
202200733 A1 09

Implement K-nearest neighbor algorithm using the Wine dataset.

Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Load the dataset
file_path = "wine_dataset.csv" # Update with your actual file path
df = pd.read_csv(file_path)

# Display first few rows
print(df.head())

# Split features and target variable
X = df.drop(columns=['quality']) # Features
y = df['quality'] # Target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Initialize KNN classifier
k = 5 # You can experiment with different k values
knn = KNeighborsClassifier(n_neighbors=k)

# Train the model
knn.fit(X_train, y_train)

# Make predictions
y_pred = knn.predict(X_test)

# Evaluate the model
```

```

accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print('Classification Report:')
print(classification_report(y_test, y_pred))
print('Confusion Matrix:')
print(confusion_matrix(y_test, y_pred))

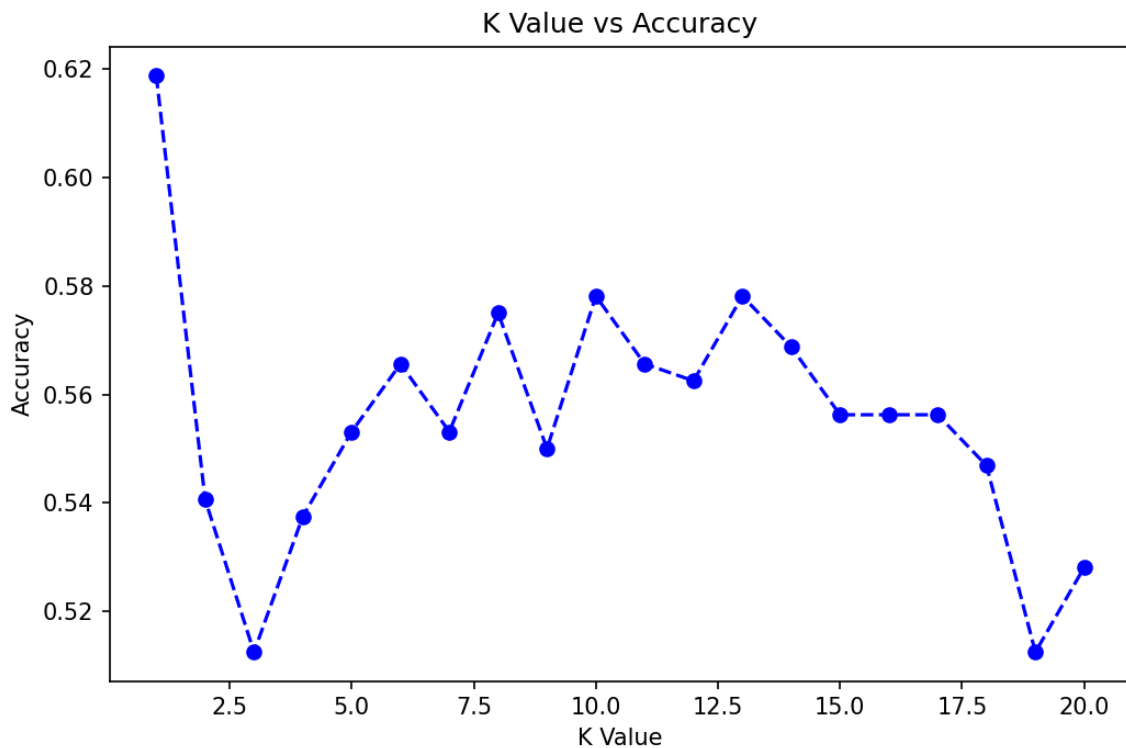
# Plot accuracy vs. k values
k_values = range(1, 21)
accuracy_scores = []

for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    y_pred_k = knn.predict(X_test)
    accuracy_scores.append(accuracy_score(y_test, y_pred_k))

plt.figure(figsize=(8, 5))
plt.plot(k_values, accuracy_scores, marker='o', linestyle='dashed', color='b')
plt.xlabel('K Value')
plt.ylabel('Accuracy')
plt.title('K Value vs Accuracy')
plt.show()

```

Output:



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	...	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	...	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	...	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	...	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	...	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	...	0.9978	3.51	0.56	9.4	5

[5 rows x 12 columns]
Accuracy: 0.55

	precision	recall	f1-score	support
3	0.00	0.00	0.00	1
4	0.33	0.10	0.15	10
5	0.60	0.68	0.64	130
6	0.52	0.56	0.54	132
7	0.52	0.33	0.41	42
8	0.00	0.00	0.00	5
accuracy			0.55	320
macro avg	0.33	0.28	0.29	320
weighted avg	0.54	0.55	0.54	320

Confusion Matrix:

```
[[ 0  0  0  1  0  0]
 [ 0  1  3  6  0  0]
 [ 0  2 88 40  0  0]
 [ 1  0 47 74 10  0]
 [ 0  0  7 21 14  0]
 [ 0  0  1  1  3  0]]
```

PS C:\Users\admin\Onedrive\Desktop\6SEM\ML>