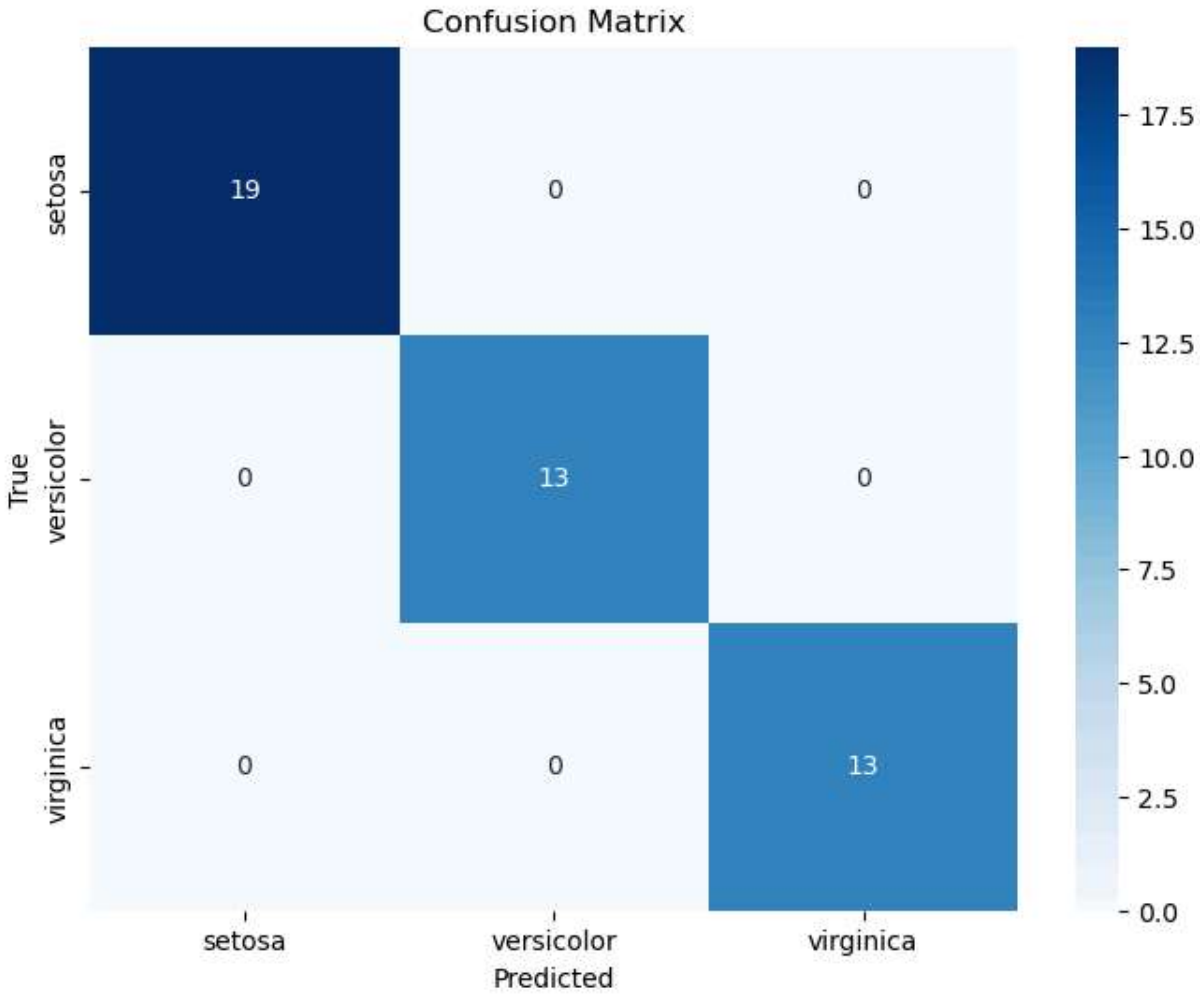
iris classification project

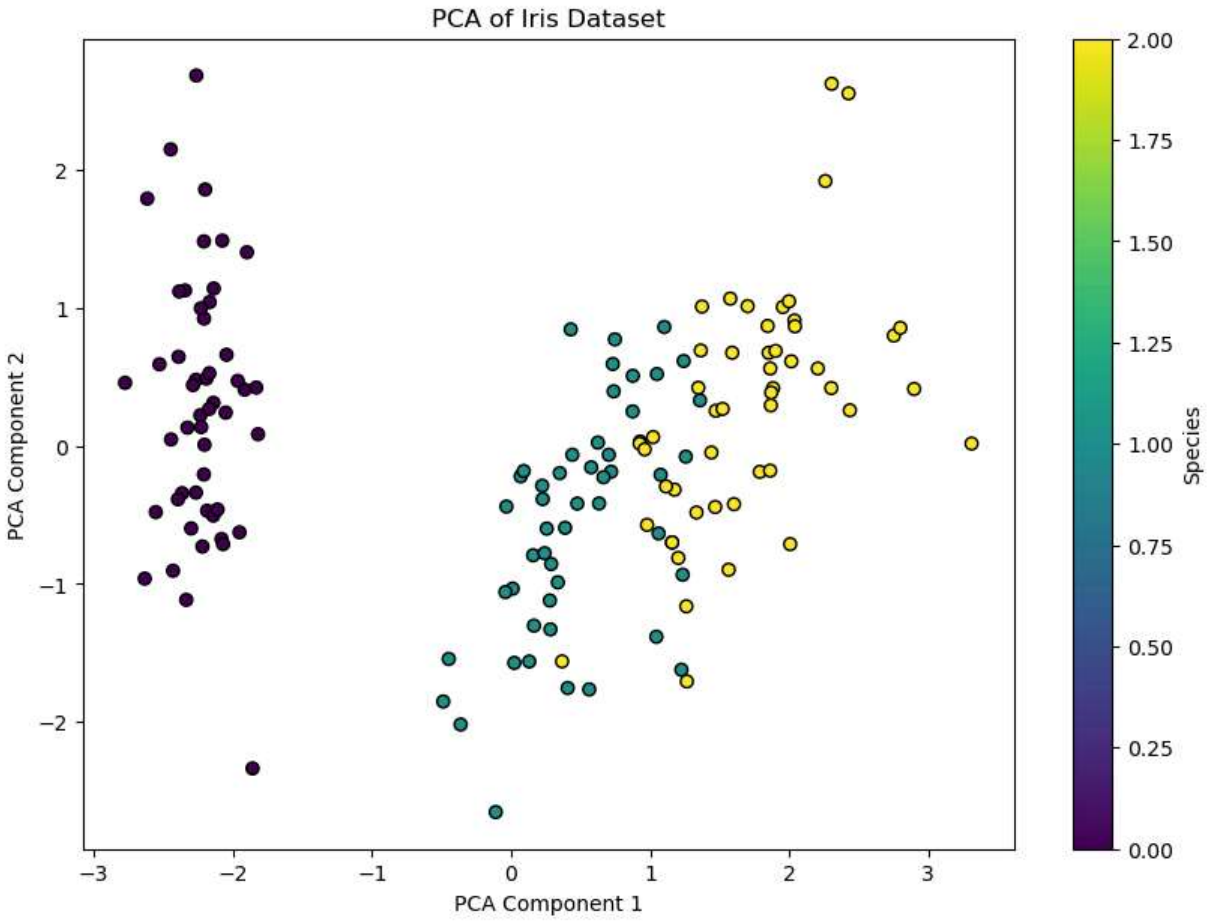
In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

# Load Iris dataset
iris = load_iris()
X = iris.data
y = iris.target
feature_names = iris.feature_names
target_names = iris.target_names

# Create DataFrame
df = pd.DataFrame(X, columns=feature_names)
df['species'] = y

# Preprocess Data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split Data
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, ran

# Train Model
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)

# Evaluate Model
y_pred = model.predict(X_test)
print("Classification Report:")
print(classification_report(y_test, y_pred, target_names=target_names))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=target_names, ytickl
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

# PCA for Visualization
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

plt.figure(figsize=(10, 7))
```

```
scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y, cmap='viridis', edgecolor='k')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.title('PCA of Iris Dataset')
plt.colorbar(scatter, label='Species')
plt.show()
```

```
Classification Report:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        19
  versicolor       1.00      1.00      1.00        13
   virginica       1.00      1.00      1.00        13

    accuracy                           1.00        45
   macro avg       1.00      1.00      1.00        45
weighted avg       1.00      1.00      1.00        45
```





In [ ]:

wine quality classification

In [1]:
```
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 StandardScaler
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix

# Load Wine Quality dataset
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/wineq
df = pd.read_csv(url, sep=';')

# Preview the dataset
print("First 5 rows of the dataset:")
print(df.head(10))

# Binary Classification: Good (quality >= 6) or Bad (quality < 6)
df['quality'] = df['quality'].apply(lambda x: 1 if x >= 6 else 0)

# Features and Target
X = df.drop('quality', axis=1)
```

```
y = df['quality']

# Preprocess Data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split Data
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, ran

# Train Model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Evaluate Model
y_pred = model.predict(X_test)
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Bad', 'Good'], yti
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

# PCA for Visualization
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

plt.figure(figsize=(10, 7))
scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y, cmap='viridis', edgecolor='k')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.title('PCA of Wine Quality Dataset')
plt.colorbar(scatter, label='Quality')
plt.show()
```

First 5 rows of the dataset:

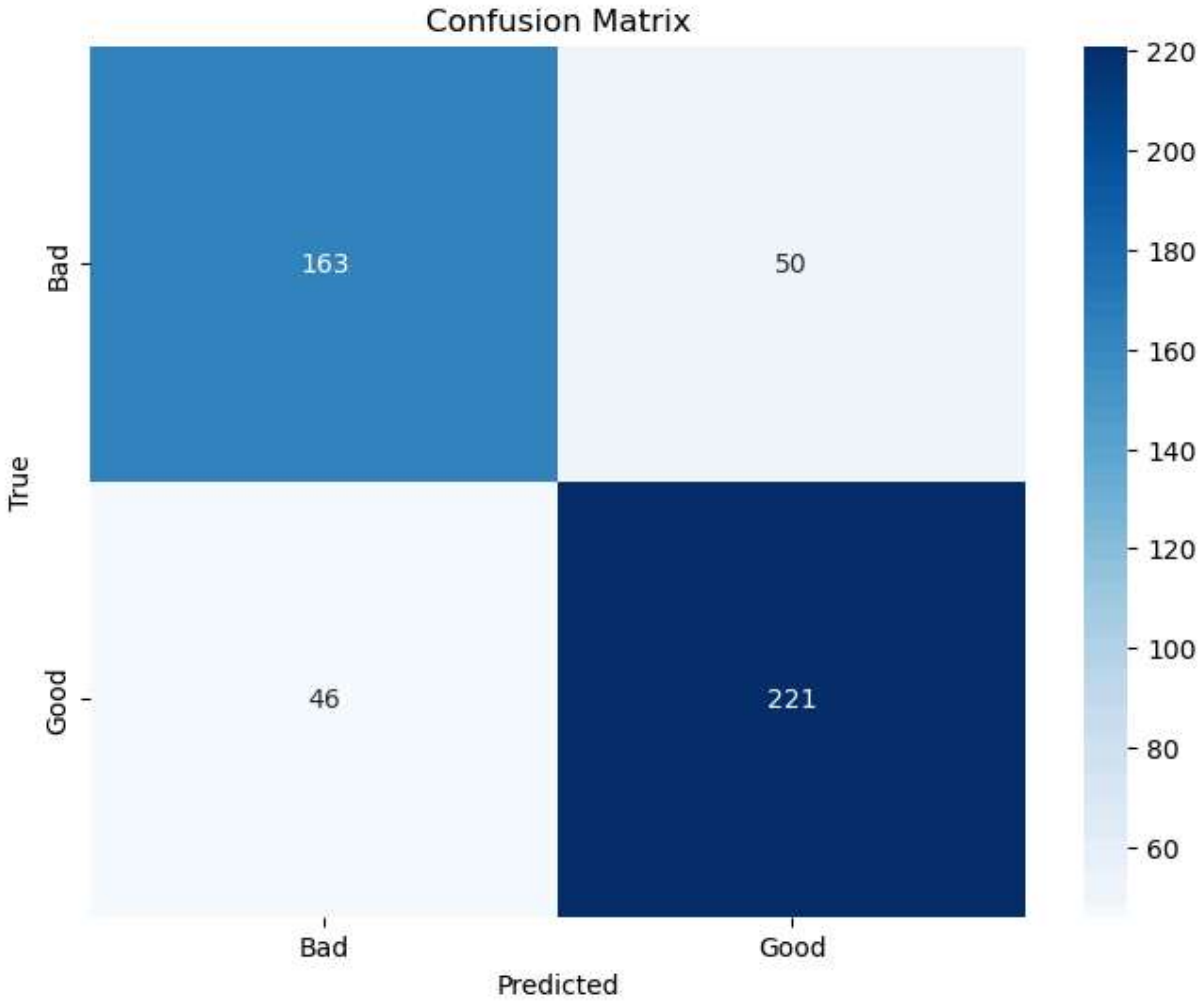
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides
0	7.4	0.70	0.00	1.9	0.076
1	7.8	0.88	0.00	2.6	0.098
2	7.8	0.76	0.04	2.3	0.092
3	11.2	0.28	0.56	1.9	0.075
4	7.4	0.70	0.00	1.9	0.076
5	7.4	0.66	0.00	1.8	0.075
6	7.9	0.60	0.06	1.6	0.069
7	7.3	0.65	0.00	1.2	0.065
8	7.8	0.58	0.02	2.0	0.073
9	7.5	0.50	0.36	6.1	0.071

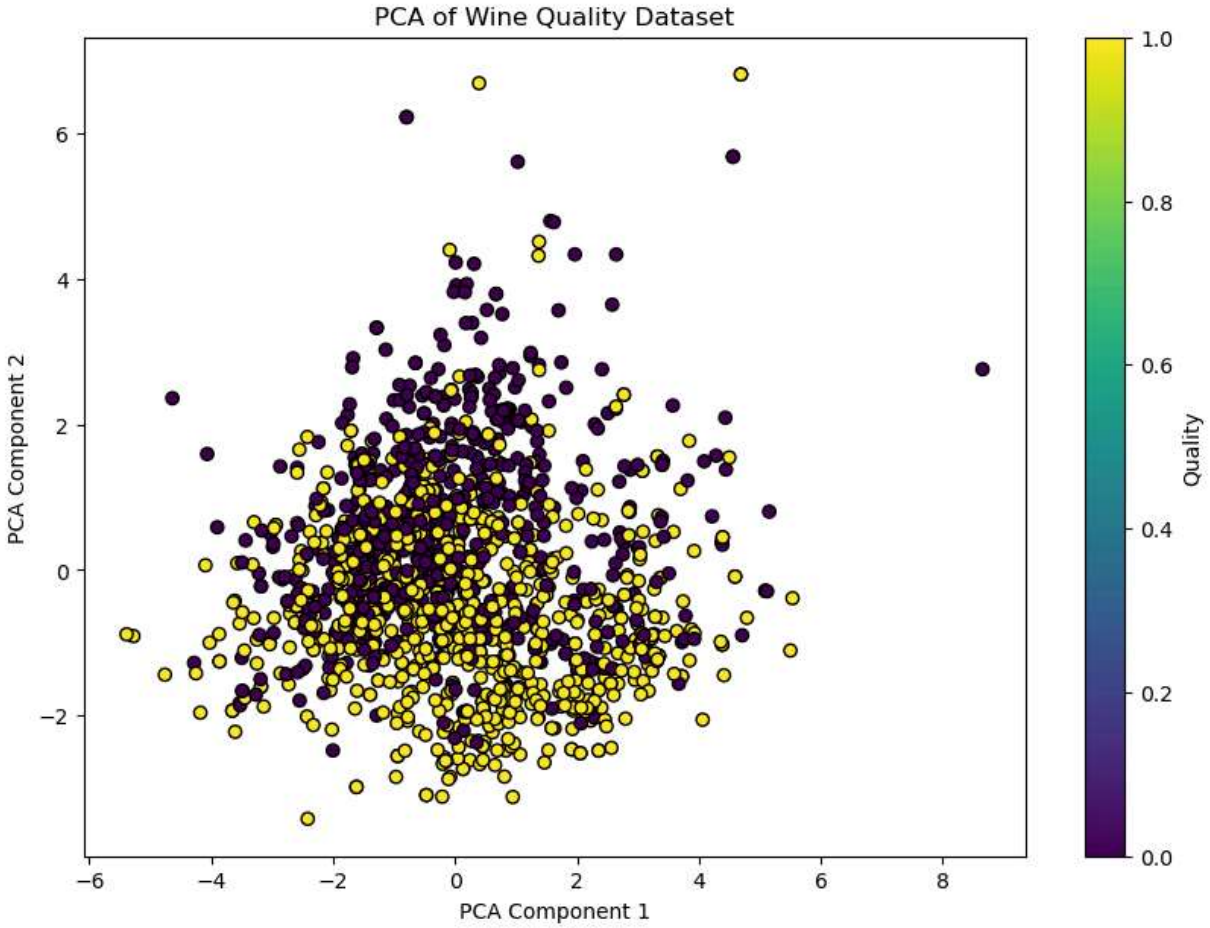
	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates
0	11.0	34.0	0.9978	3.51	0.56
1	25.0	67.0	0.9968	3.20	0.68
2	15.0	54.0	0.9970	3.26	0.65
3	17.0	60.0	0.9980	3.16	0.58
4	11.0	34.0	0.9978	3.51	0.56
5	13.0	40.0	0.9978	3.51	0.56
6	15.0	59.0	0.9964	3.30	0.46
7	15.0	21.0	0.9946	3.39	0.47
8	9.0	18.0	0.9968	3.36	0.57
9	17.0	102.0	0.9978	3.35	0.80

	alcohol	quality
0	9.4	5
1	9.8	5
2	9.8	5
3	9.8	6
4	9.4	5
5	9.4	5
6	9.4	5
7	10.0	7
8	9.5	7
9	10.5	5

Classification Report:

	precision	recall	f1-score	support
0	0.78	0.77	0.77	213
1	0.82	0.83	0.82	267
accuracy			0.80	480
macro avg	0.80	0.80	0.80	480
weighted avg	0.80	0.80	0.80	480


Confusion Matrix
Bad
Good
True
163
50
46
221
Bad
Good
Predicted
220
200
180
160
140
120
100
80
60


PCA of Wine Quality Dataset
PCA Component 2
PCA Component 1
Quality
6
4
2
0
−2
−6
−4
−2
0
2
4
6
8
1.0
0.8
0.6
0.4
0.2
0.0

In [ ]: