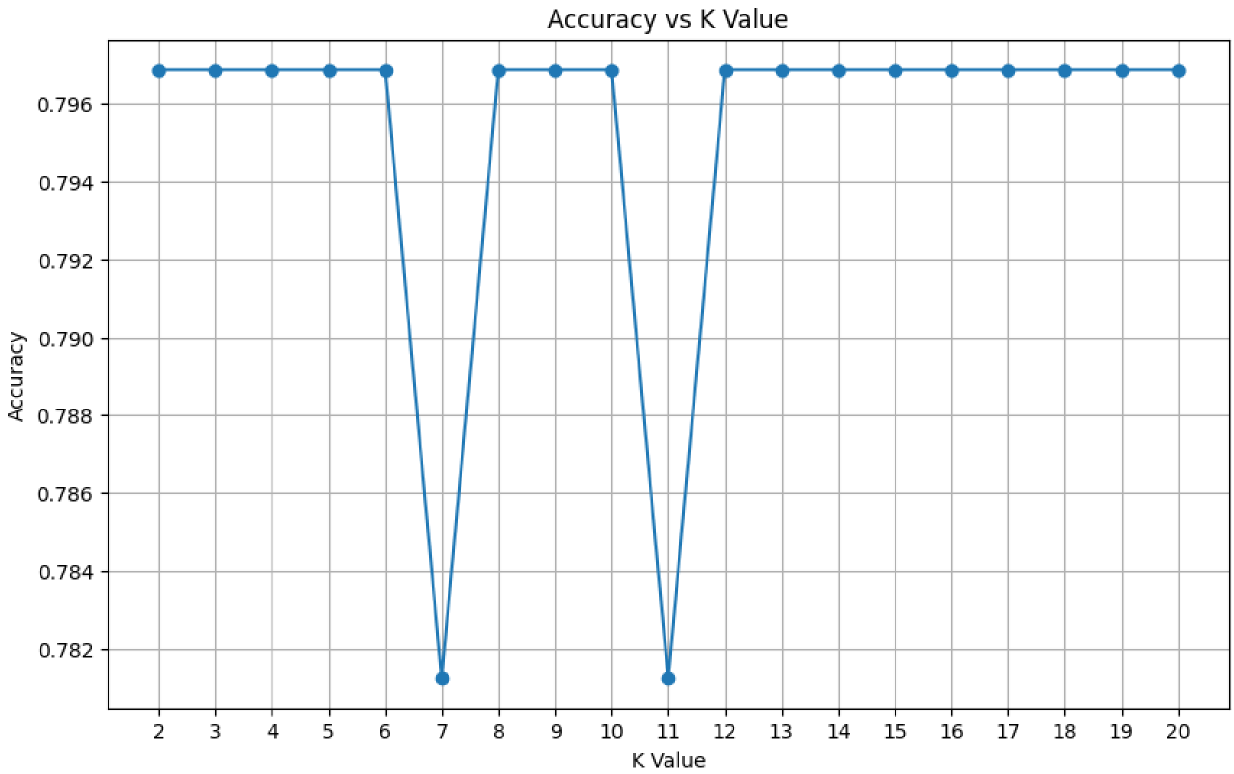
```
 import pandas as pd
 from sklearn.model_selection import train_test_split
 from sklearn.neighbors import KNeighborsClassifier
 from sklearn.metrics import accuracy_score
 import matplotlib.pyplot as plt
 data=pd.read_csv("/content/breast_cancer_survival.csv")
 data = data.dropna()
 X = data[['Age', 'Gender', 'Protein1', 'Protein2', 'Protein3',
'Protein4','Tumour_Stage']]
 y = data['Patient_Status']
 # Convert categorical variables to one-hot encoding
 X_encoded = pd.get_dummies(X)
 # Split the data into training and testing sets
 X_train, X_test, y_train, y_test = train_test_split(X_encoded, y,
test_size=0.2, random_state=42)
 # Train KNN for different K values and find accuracies
 accuracies = []
 for k in range(2, 21):
     knn = KNeighborsClassifier(n_neighbors=k)
     knn.fit(X_train, y_train)
     y_pred = knn.predict(X_test)
     accuracy = accuracy_score(y_test, y_pred)
     accuracies.append(accuracy)
 print(f"Accuracy:{accuracy}")
 # Plot the graph

 plt.figure(figsize=(10, 6))
 plt.plot(range(2, 21), accuracies, marker='o', linestyle='-')

 plt.title('Accuracy vs K Value')
 plt.xlabel('K Value')
 plt.ylabel('Accuracy')
 plt.xticks(range(2, 21))
 plt.grid(True)
 plt.show()
```

```
Accuracy:0.796875
```



```
 import pandas as pd
 from sklearn.model_selection import train_test_split
 from sklearn.tree import DecisionTreeClassifier
 from sklearn.metrics import accuracy_score, precision_score,
recall_score,f1_score
 # Load the dataset
 data = pd.read_csv("/content/breast_cancer_survival.csv")
 # Drop rows with NaN values
 data = data.dropna()
 # Extract features and target variable
 X = data[['Age', 'Gender', 'Protein1', 'Protein2', 'Protein3',
'Protein4','Tumour_Stage']]

 y = data['Patient_Status']
  # Convert categorical variables to one-hot encoding
 X_encoded = pd.get_dummies(X)
 # Split the data into training and testing sets
 X_train, X_test, y_train, y_test = train_test_split(X_encoded, y,
test_size=0.2, random_state=42)
 # Train Decision Tree classifier
 decision_tree = DecisionTreeClassifier()
 decision_tree.fit(X_train, y_train)
 # Predict on the test set
 y_pred = decision_tree.predict(X_test)
  # Calculate performance metrics
 accuracy = accuracy_score(y_test, y_pred)
```

```python
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')
# Print the metrics
print("Decision Tree Classifier:")
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
```

```
Decision Tree Classifier:
Accuracy: 0.734375
Precision: 0.7102430555555554
Recall: 0.734375
F1 Score: 0.7208462732919254
```