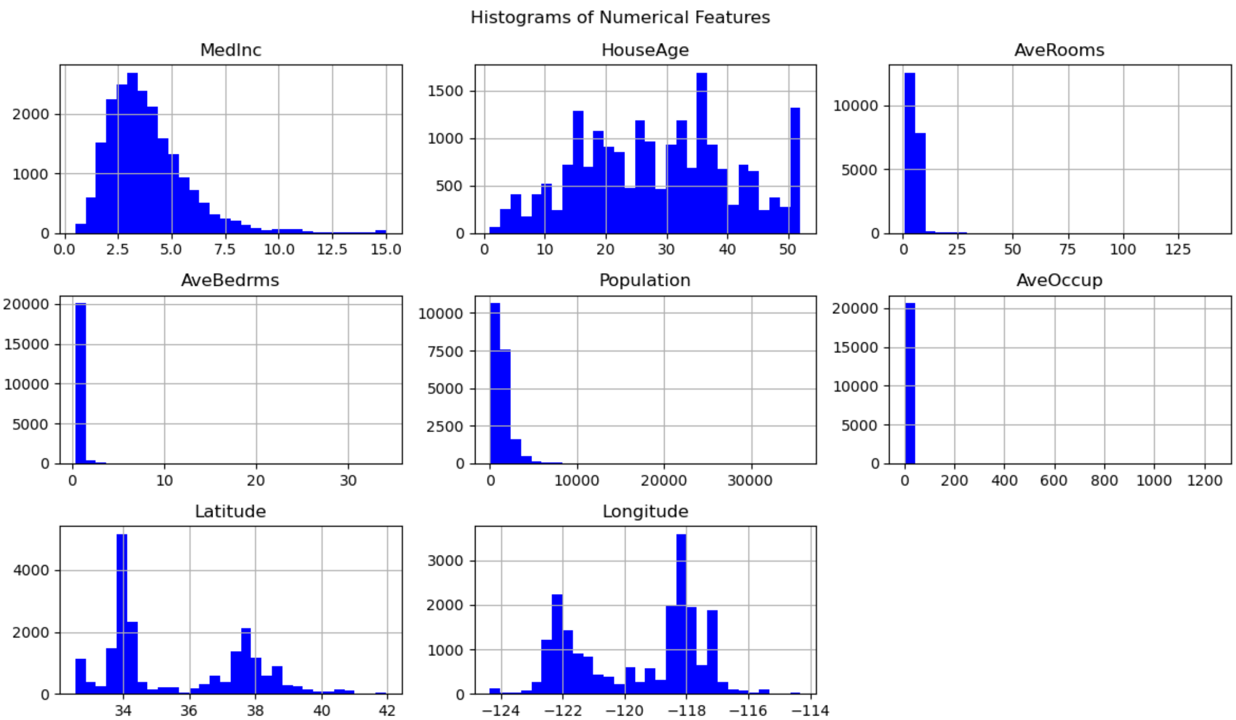
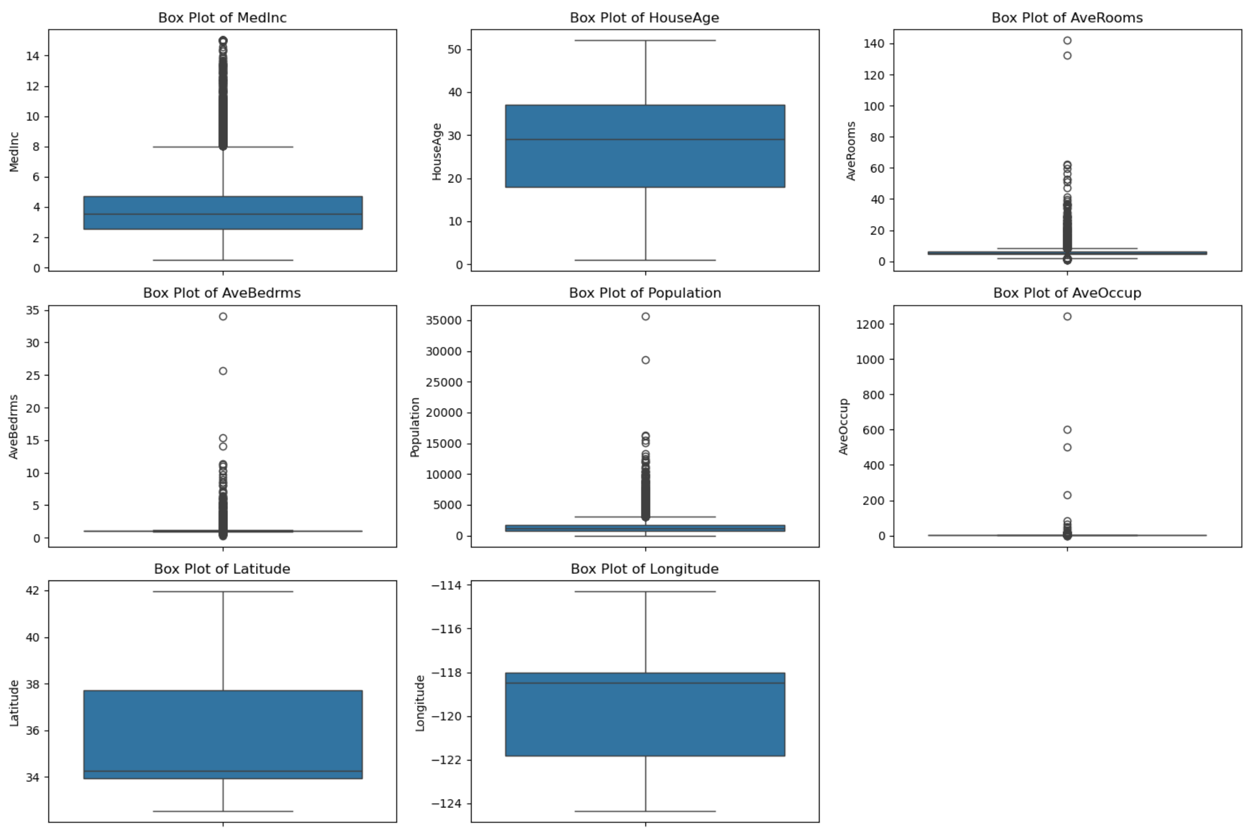
```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
from sklearn.datasets import fetch_california_housing
california_housing = fetch_california_housing()
data = pd.DataFrame(california_housing.data,
columns=california_housing.feature_names)
numerical_features = data.select_dtypes(include=[np.number]).columns
print(numerical_features)
data.hist(bins=30, figsize=(12, 7),color='blue')
plt.suptitle('Histograms of Numerical Features')
plt.tight_layout()
plt.show()
plt.figure(figsize=(15, 10))
for i, column in enumerate(data.columns, 1):
    plt.subplot(3, 3, i)
    sns.boxplot(y=data[column])
    plt.title(f'Box Plot of {column}')
plt.tight_layout()
plt.show()
print("Outliers Detection:\n")
outliers_summary = {}
for feature in numerical_features:
    Q1 = data[feature].quantile(0.25)
    Q3 = data[feature].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = data[(data[feature] < lower_bound) | (data[feature] >
upper_bound)]
    outliers_summary[feature] = len(outliers)
    print(f"\t{feature}: {len(outliers)} outliers\t")
```

```
Index(['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'Population',
'AveOccup',
       'Latitude', 'Longitude'],
      dtype='object')
```





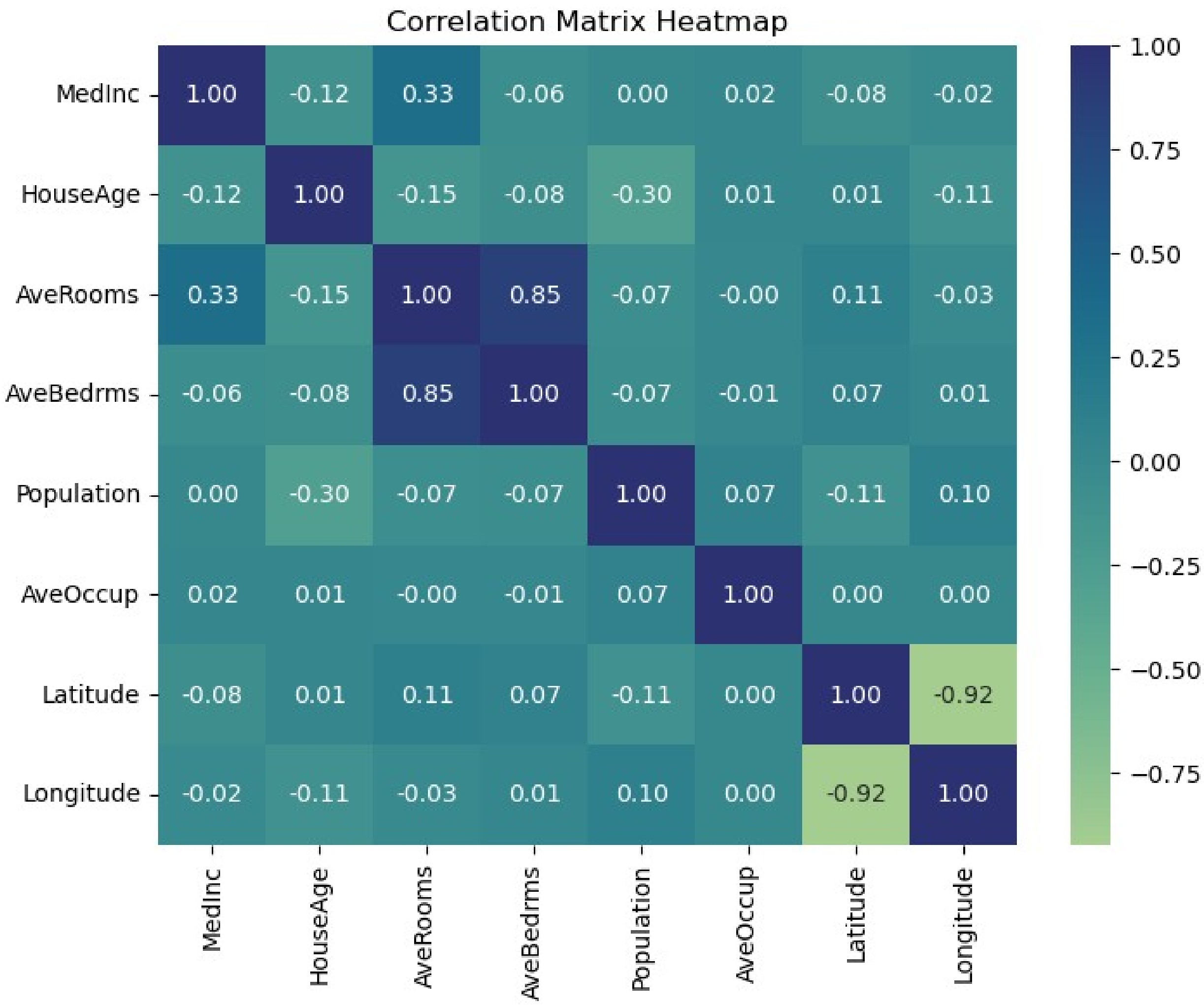

```
Outliers Detection:
```

```
      MedInc: 681 outliers
      HouseAge: 0 outliers
      AveRooms: 511 outliers
      AveBedrms: 1424 outliers
      Population: 1196 outliers
      AveOccup: 711 outliers
      Latitude: 0 outliers
      Longitude: 0 outliers
```

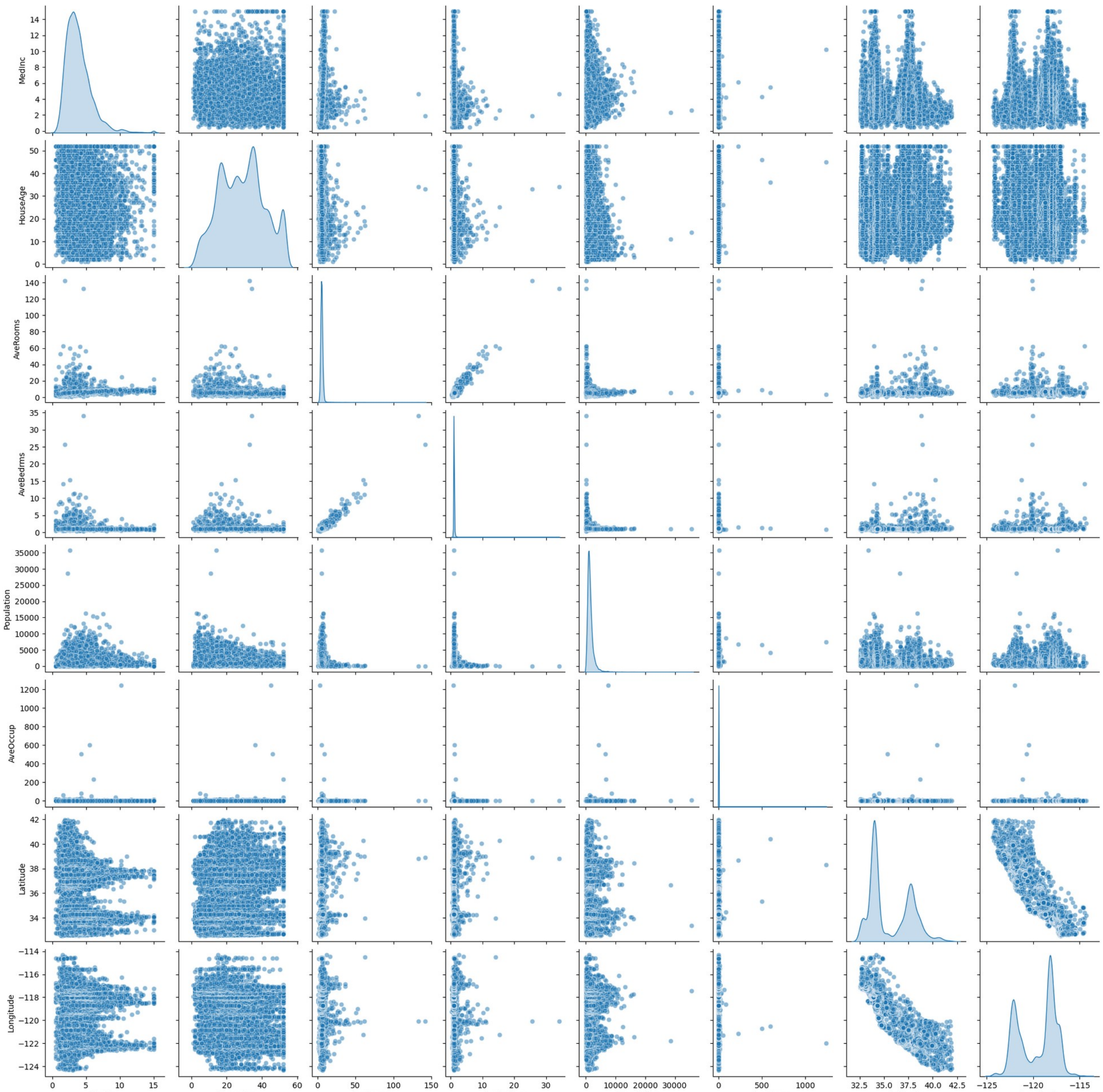
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.datasets import fetch_california_housing
california_housing = fetch_california_housing()
data = pd.DataFrame(california_housing.data,
columns=california_housing.feature_names)
corr_matrix = data.corr()
print(corr_matrix)
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap= 'crest', fmt='.2f')
plt.title('Correlation Matrix Heatmap')
plt.show()
plt.figure()
sns.pairplot(data, kind='scatter',diag_kind='kde', plot_kws={'alpha':
0.5})
plt.show()
```

```
              MedInc  HouseAge  AveRooms  AveBedrms  Population
AveOccup  \
MedInc      1.000000 -0.119034  0.326895  -0.062040    0.004834
0.018766
HouseAge   -0.119034  1.000000 -0.153277  -0.077747   -0.296244
0.013191
AveRooms    0.326895 -0.153277  1.000000   0.847621   -0.072213 -
0.004852
AveBedrms  -0.062040 -0.077747  0.847621   1.000000   -0.066197 -
0.006181
Population  0.004834 -0.296244 -0.072213  -0.066197    1.000000
0.069863
AveOccup    0.018766  0.013191 -0.004852  -0.006181    0.069863
1.000000
Latitude   -0.079809  0.011173  0.106389   0.069721   -0.108785
0.002366
Longitude  -0.015176 -0.108197 -0.027540   0.013344    0.099773
0.002476

            Latitude  Longitude
MedInc     -0.079809  -0.015176
HouseAge    0.011173  -0.108197
```

```
AveRooms    0.106389  -0.027540
AveBedrms   0.069721   0.013344
Population -0.108785   0.099773
AveOccup    0.002366   0.002476
Latitude    1.000000  -0.924664
Longitude  -0.924664   1.000000
```

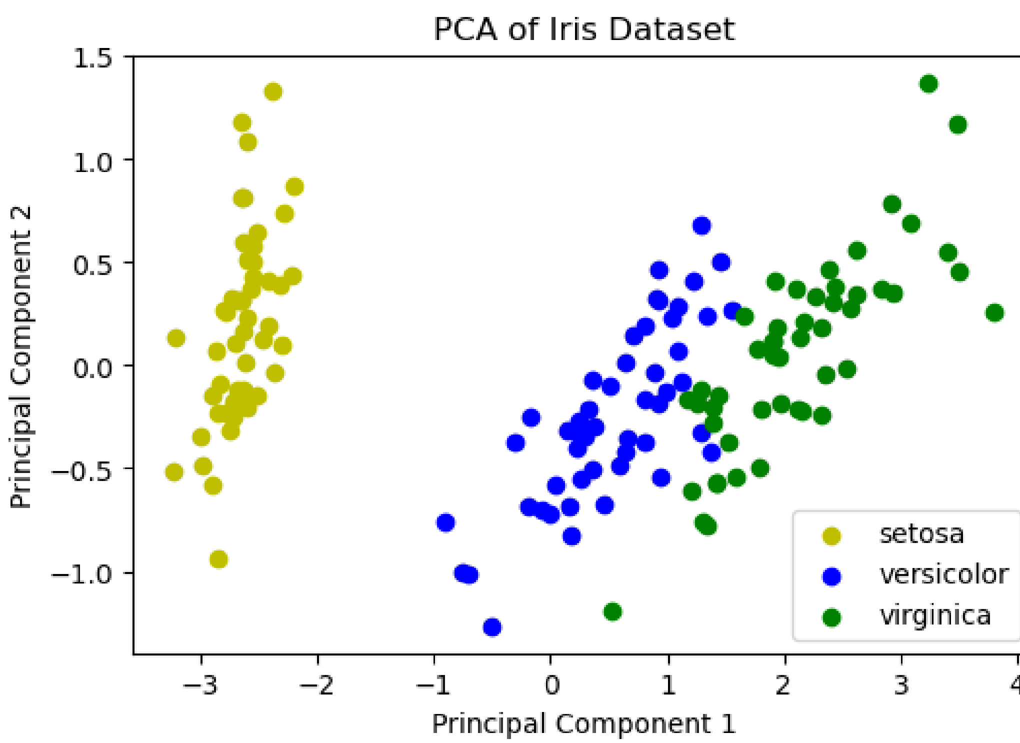



```
<Figure size 640x480 with 0 Axes>
```




```python
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
iris = load_iris()
X = iris.data
y = iris.target
label_names = iris.target_names
df = pd.DataFrame(X, columns=iris.feature_names)
pca = PCA(n_components=2)
principal_components = pca.fit_transform(X)
```

```
df_pca = pd.DataFrame(data=principal_components,
columns=['Principal Component 1', 'Principal Component 2'])
df_pca['Target'] = y
plt.figure(figsize=(6, 4))
colors = ['y', 'b', 'g']
for i, label in enumerate(np.unique(y)):
    plt.scatter(df_pca[df_pca['Target'] == label]['Principal Component
1'],
df_pca[df_pca['Target'] == label]['Principal Component 2'],
label=label_names[label],
color=colors[i])
plt.title('PCA of Iris Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend()
plt.savefig('pcaofirisdataset.png')
plt.show()
```




```
import csv
a = []
with open('/home/ewitise/Downloads/ENJOYSPORT.csv', 'r') as csvfile:
    for row in csv.reader(csvfile):
        a.append(row)
print(a)
print("\n The total number of training instances are : ",len(a))
num_attribute = len(a[0])-1
```

```
print("\n The initial hypothesis is : ")
hypothesis = ['0']*num_attribute
print(hypothesis)
for i in range(0, len(a)):
    if a[i][num_attribute] == 'yes':
       for j in range(0, num_attribute):
            if hypothesis[j] == '0' or hypothesis[j] == a[i][j]:
                hypothesis[j] = a[i][j]
            else:
              hypothesis[j] = '?'
    print("\n The hypothesis for the training instance {} is :\
n" .format(i+1),hypothesis)
print("\n The Maximally specific hypothesis for the training instance
is ")
print(hypothesis)
```

```
[['Sky', 'AirTemp', 'Humidity', 'Wind', 'Water', 'Forecast',
'EnjoySport'], ['Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same',
'1'], ['Sunny', 'Warm', 'High', 'Strong', 'Warm', 'Same', '1'],
['Rainy', 'Cold', 'High', 'Strong', 'Warm', 'Change', '0'], ['Sunny',
'Warm', 'High', 'Strong', 'Cool', 'Change', '1']]

 The total number of training instances are :  5

 The initial hypothesis is :
['0', '0', '0', '0', '0', '0']

 The hypothesis for the training instance 1 is :
 ['0', '0', '0', '0', '0', '0']

 The hypothesis for the training instance 2 is :
 ['0', '0', '0', '0', '0', '0']

 The hypothesis for the training instance 3 is :
 ['0', '0', '0', '0', '0', '0']

 The hypothesis for the training instance 4 is :
 ['0', '0', '0', '0', '0', '0']

 The hypothesis for the training instance 5 is :
 ['0', '0', '0', '0', '0', '0']

 The Maximally specific hypothesis for the training instance is
['0', '0', '0', '0', '0', '0']
```

```
import numpy as np
import matplotlib.pyplot as plt
from collections import Counter
data = np.random.rand(100)
labels = ["Class1" if x <= 0.5 else "Class2" for x in data[:50]]
```

```
def euclidean_distance(x1, x2):
  return abs(x1 - x2)
def knn_classifier(train_data, train_labels, test_point, k):
  distances = [(euclidean_distance(test_point, train_data[i]),
train_labels[i])
                                                                 for i
in range(len(train_data))]
  distances.sort(key=lambda x: x[0])
  k_nearest_neighbors = distances[:k]
  k_nearest_labels = [label for _, label in k_nearest_neighbors]
  return Counter(k_nearest_labels).most_common(1)[0][0]
train_data = data[:50]
train_labels = labels
test_data = data[50:]
k_values = [1, 2, 3, 4, 5, 20, 30]
print("--- k-Nearest Neighbors Classification ---")
print("Training dataset: First 50 points labeled based on the rule (x
<= 0.5 -> Class1, x > 0.5 -> Class2)")
print("Testing dataset: Remaining 50 points to be classified\n")
results = {}
for k in k_values:
   print(f"Results for k = {k}:")
   classified_labels = [knn_classifier(train_data, train_labels,
test_point, k) for test_point in test_data]
   results[k] = classified_labels
   for i, label in enumerate(classified_labels, start=51):
        print(f"Point x{i} (value: {test_data[i - 51]:.4f}) is
classified as {label}")
   print("\n")
print("Classification complete.\n")

num_k = len(k_values)
rows = (num_k + 2) // 3
cols = 3
plt.figure(figsize=(15, 5 * rows))
for idx, k in enumerate(k_values):
   classified_labels = results[k]
   class1_points = [test_data[i] for i in range(len(test_data)) if
classified_labels[i] == "Class1"]
   class2_points = [test_data[i] for i in range(len(test_data)) if
classified_labels[i] == "Class2"]

   plt.subplot(rows, cols, idx + 1)
   plt.scatter(train_data, [0] * len(train_data), c=["blue" if label
== "Class1" else "red" for label in train_labels],
          label="Training Data", marker="o")
   plt.scatter(class1_points, [1] * len(class1_points), c="blue",
label="Class1 (Test)", marker="x")
   plt.scatter(class2_points, [1] * len(class2_points), c="red",
```

```
label="Class2 (Test)", marker="x")
   plt.title(f"k-NN Results for k = {k}")
   plt.xlabel("Data Points")
   plt.ylabel("Classification Level")
   plt.legend()
   plt.grid(True)

plt.tight_layout()
plt.savefig('knn_classification.png')
plt.show()
```

```
--- k-Nearest Neighbors Classification ---
Training dataset: First 50 points labeled based on the rule (x <= 0.5
-> Class1, x > 0.5 -> Class2)
Testing dataset: Remaining 50 points to be classified

Results for k = 1:
Point x51 (value: 0.3537) is classified as Class1
Point x52 (value: 0.3931) is classified as Class1
Point x53 (value: 0.0939) is classified as Class1
Point x54 (value: 0.9610) is classified as Class2
Point x55 (value: 0.8390) is classified as Class2
Point x56 (value: 0.7921) is classified as Class2
Point x57 (value: 0.7260) is classified as Class2
Point x58 (value: 0.2140) is classified as Class1
Point x59 (value: 0.4824) is classified as Class1
Point x60 (value: 0.8411) is classified as Class2
Point x61 (value: 0.6597) is classified as Class2
Point x62 (value: 0.3090) is classified as Class1
Point x63 (value: 0.2481) is classified as Class1
Point x64 (value: 0.2470) is classified as Class1
Point x65 (value: 0.9320) is classified as Class2
Point x66 (value: 0.9717) is classified as Class2
Point x67 (value: 0.8989) is classified as Class2
Point x68 (value: 0.4311) is classified as Class1
Point x69 (value: 0.0788) is classified as Class1
Point x70 (value: 0.0354) is classified as Class1
Point x71 (value: 0.8361) is classified as Class2
Point x72 (value: 0.4777) is classified as Class1
Point x73 (value: 0.2504) is classified as Class1
Point x74 (value: 0.6567) is classified as Class2
Point x75 (value: 0.2831) is classified as Class1
Point x76 (value: 0.9361) is classified as Class2
Point x77 (value: 0.6256) is classified as Class2
Point x78 (value: 0.4972) is classified as Class1
Point x79 (value: 0.3888) is classified as Class1
Point x80 (value: 0.4373) is classified as Class1
Point x81 (value: 0.4775) is classified as Class1
Point x82 (value: 0.2877) is classified as Class1
Point x83 (value: 0.8856) is classified as Class2
```

```
Point x84 (value: 0.2611) is classified as Class1
Point x85 (value: 0.3077) is classified as Class1
Point x86 (value: 0.9066) is classified as Class2
Point x87 (value: 0.9856) is classified as Class2
Point x88 (value: 0.3278) is classified as Class1
Point x89 (value: 0.5034) is classified as Class2
Point x90 (value: 0.5070) is classified as Class2
Point x91 (value: 0.9881) is classified as Class2
Point x92 (value: 0.8672) is classified as Class2
Point x93 (value: 0.0983) is classified as Class1
Point x94 (value: 0.9252) is classified as Class2
Point x95 (value: 0.8664) is classified as Class2
Point x96 (value: 0.9819) is classified as Class2
Point x97 (value: 0.5803) is classified as Class2
Point x98 (value: 0.6352) is classified as Class2
Point x99 (value: 0.9272) is classified as Class2
Point x100 (value: 0.0044) is classified as Class1


Results for k = 2:
Point x51 (value: 0.3537) is classified as Class1
Point x52 (value: 0.3931) is classified as Class1
Point x53 (value: 0.0939) is classified as Class1
Point x54 (value: 0.9610) is classified as Class2
Point x55 (value: 0.8390) is classified as Class2
Point x56 (value: 0.7921) is classified as Class2
Point x57 (value: 0.7260) is classified as Class2
Point x58 (value: 0.2140) is classified as Class1
Point x59 (value: 0.4824) is classified as Class1
Point x60 (value: 0.8411) is classified as Class2
Point x61 (value: 0.6597) is classified as Class2
Point x62 (value: 0.3090) is classified as Class1
Point x63 (value: 0.2481) is classified as Class1
Point x64 (value: 0.2470) is classified as Class1
Point x65 (value: 0.9320) is classified as Class2
Point x66 (value: 0.9717) is classified as Class2
Point x67 (value: 0.8989) is classified as Class2
Point x68 (value: 0.4311) is classified as Class1
Point x69 (value: 0.0788) is classified as Class1
Point x70 (value: 0.0354) is classified as Class1
Point x71 (value: 0.8361) is classified as Class2
Point x72 (value: 0.4777) is classified as Class1
Point x73 (value: 0.2504) is classified as Class1
Point x74 (value: 0.6567) is classified as Class2
Point x75 (value: 0.2831) is classified as Class1
Point x76 (value: 0.9361) is classified as Class2
Point x77 (value: 0.6256) is classified as Class2
Point x78 (value: 0.4972) is classified as Class1
Point x79 (value: 0.3888) is classified as Class1
```

```
Point x80 (value: 0.4373) is classified as Class1
Point x81 (value: 0.4775) is classified as Class1
Point x82 (value: 0.2877) is classified as Class1
Point x83 (value: 0.8856) is classified as Class2
Point x84 (value: 0.2611) is classified as Class1
Point x85 (value: 0.3077) is classified as Class1
Point x86 (value: 0.9066) is classified as Class2
Point x87 (value: 0.9856) is classified as Class2
Point x88 (value: 0.3278) is classified as Class1
Point x89 (value: 0.5034) is classified as Class2
Point x90 (value: 0.5070) is classified as Class2
Point x91 (value: 0.9881) is classified as Class2
Point x92 (value: 0.8672) is classified as Class2
Point x93 (value: 0.0983) is classified as Class1
Point x94 (value: 0.9252) is classified as Class2
Point x95 (value: 0.8664) is classified as Class2
Point x96 (value: 0.9819) is classified as Class2
Point x97 (value: 0.5803) is classified as Class2
Point x98 (value: 0.6352) is classified as Class2
Point x99 (value: 0.9272) is classified as Class2
Point x100 (value: 0.0044) is classified as Class1


Results for k = 3:
Point x51 (value: 0.3537) is classified as Class1
Point x52 (value: 0.3931) is classified as Class1
Point x53 (value: 0.0939) is classified as Class1
Point x54 (value: 0.9610) is classified as Class2
Point x55 (value: 0.8390) is classified as Class2
Point x56 (value: 0.7921) is classified as Class2
Point x57 (value: 0.7260) is classified as Class2
Point x58 (value: 0.2140) is classified as Class1
Point x59 (value: 0.4824) is classified as Class1
Point x60 (value: 0.8411) is classified as Class2
Point x61 (value: 0.6597) is classified as Class2
Point x62 (value: 0.3090) is classified as Class1
Point x63 (value: 0.2481) is classified as Class1
Point x64 (value: 0.2470) is classified as Class1
Point x65 (value: 0.9320) is classified as Class2
Point x66 (value: 0.9717) is classified as Class2
Point x67 (value: 0.8989) is classified as Class2
Point x68 (value: 0.4311) is classified as Class1
Point x69 (value: 0.0788) is classified as Class1
Point x70 (value: 0.0354) is classified as Class1
Point x71 (value: 0.8361) is classified as Class2
Point x72 (value: 0.4777) is classified as Class1
Point x73 (value: 0.2504) is classified as Class1
Point x74 (value: 0.6567) is classified as Class2
Point x75 (value: 0.2831) is classified as Class1
```

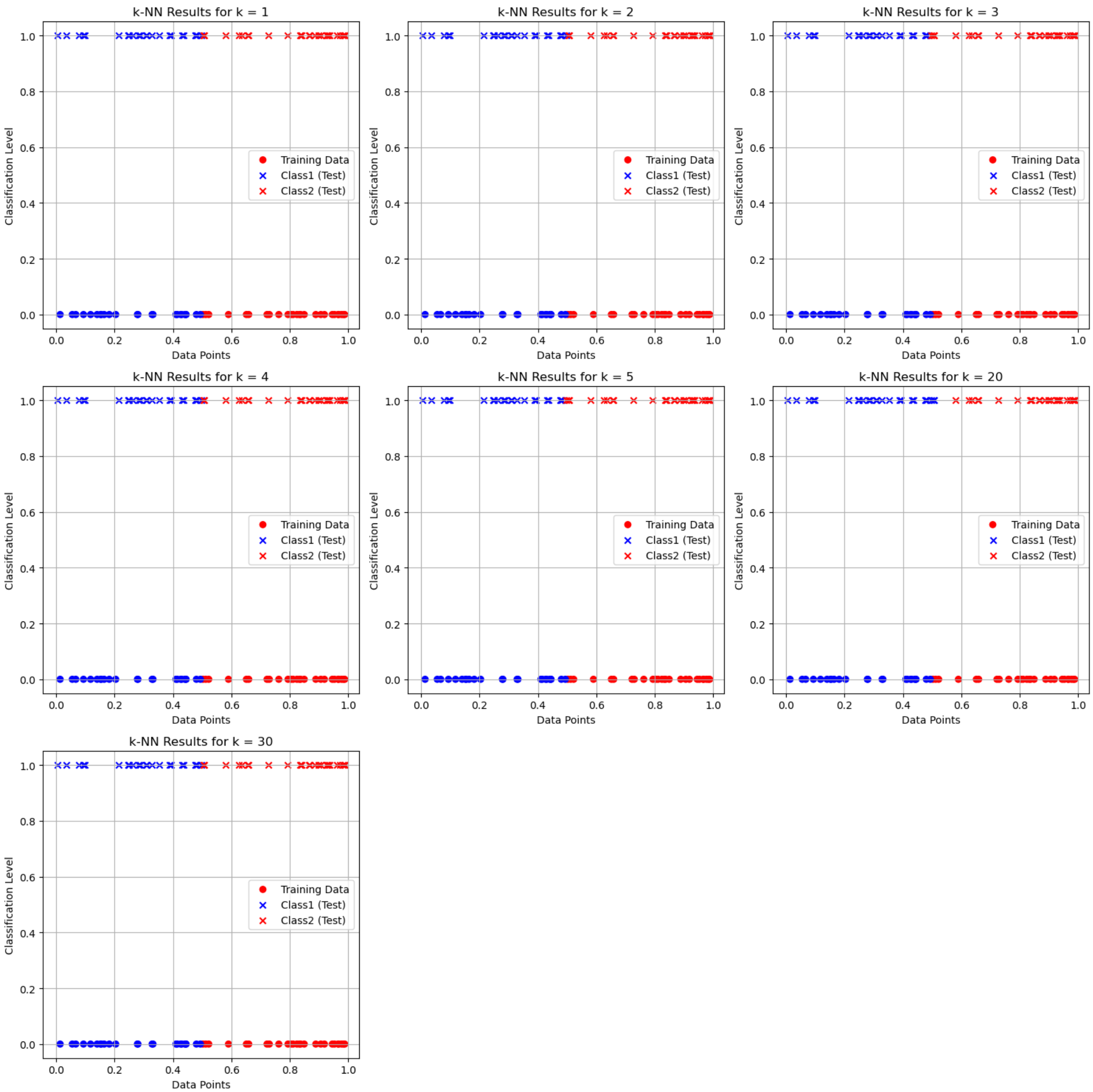
```
Point x76 (value: 0.9361) is classified as Class2
Point x77 (value: 0.6256) is classified as Class2
Point x78 (value: 0.4972) is classified as Class2
Point x79 (value: 0.3888) is classified as Class1
Point x80 (value: 0.4373) is classified as Class1
Point x81 (value: 0.4775) is classified as Class1
Point x82 (value: 0.2877) is classified as Class1
Point x83 (value: 0.8856) is classified as Class2
Point x84 (value: 0.2611) is classified as Class1
Point x85 (value: 0.3077) is classified as Class1
Point x86 (value: 0.9066) is classified as Class2
Point x87 (value: 0.9856) is classified as Class2
Point x88 (value: 0.3278) is classified as Class1
Point x89 (value: 0.5034) is classified as Class2
Point x90 (value: 0.5070) is classified as Class2
Point x91 (value: 0.9881) is classified as Class2
Point x92 (value: 0.8672) is classified as Class2
Point x93 (value: 0.0983) is classified as Class1
Point x94 (value: 0.9252) is classified as Class2
Point x95 (value: 0.8664) is classified as Class2
Point x96 (value: 0.9819) is classified as Class2
Point x97 (value: 0.5803) is classified as Class2
Point x98 (value: 0.6352) is classified as Class2
Point x99 (value: 0.9272) is classified as Class2
Point x100 (value: 0.0044) is classified as Class1


Results for k = 4:
Point x51 (value: 0.3537) is classified as Class1
Point x52 (value: 0.3931) is classified as Class1
Point x53 (value: 0.0939) is classified as Class1
Point x54 (value: 0.9610) is classified as Class2
Point x55 (value: 0.8390) is classified as Class2
Point x56 (value: 0.7921) is classified as Class2
Point x57 (value: 0.7260) is classified as Class2
Point x58 (value: 0.2140) is classified as Class1
Point x59 (value: 0.4824) is classified as Class1
Point x60 (value: 0.8411) is classified as Class2
Point x61 (value: 0.6597) is classified as Class2
Point x62 (value: 0.3090) is classified as Class1
Point x63 (value: 0.2481) is classified as Class1
Point x64 (value: 0.2470) is classified as Class1
Point x65 (value: 0.9320) is classified as Class2
Point x66 (value: 0.9717) is classified as Class2
Point x67 (value: 0.8989) is classified as Class2
Point x68 (value: 0.4311) is classified as Class1
Point x69 (value: 0.0788) is classified as Class1
Point x70 (value: 0.0354) is classified as Class1
Point x71 (value: 0.8361) is classified as Class2
```

```
Point x72 (value: 0.4777) is classified as Class1
Point x73 (value: 0.2504) is classified as Class1
Point x74 (value: 0.6567) is classified as Class2
Point x75 (value: 0.2831) is classified as Class1
Point x76 (value: 0.9361) is classified as Class2
Point x77 (value: 0.6256) is classified as Class2
Point x78 (value: 0.4972) is classified as Class1
Point x79 (value: 0.3888) is classified as Class1
Point x80 (value: 0.4373) is classified as Class1
Point x81 (value: 0.4775) is classified as Class1
Point x82 (value: 0.2877) is classified as Class1
Point x83 (value: 0.8856) is classified as Class2
Point x84 (value: 0.2611) is classified as Class1
Point x85 (value: 0.3077) is classified as Class1
Point x86 (value: 0.9066) is classified as Class2
Point x87 (value: 0.9856) is classified as Class2
Point x88 (value: 0.3278) is classified as Class1
Point x89 (value: 0.5034) is classified as Class2
Point x90 (value: 0.5070) is classified as Class2
Point x91 (value: 0.9881) is classified as Class2
Point x92 (value: 0.8672) is classified as Class2
Point x93 (value: 0.0983) is classified as Class1
Point x94 (value: 0.9252) is classified as Class2
Point x95 (value: 0.8664) is classified as Class2
Point x96 (value: 0.9819) is classified as Class2
Point x97 (value: 0.5803) is classified as Class2
Point x98 (value: 0.6352) is classified as Class2
Point x99 (value: 0.9272) is classified as Class2
Point x100 (value: 0.0044) is classified as Class1


Results for k = 5:
Point x51 (value: 0.3537) is classified as Class1
Point x52 (value: 0.3931) is classified as Class1
Point x53 (value: 0.0939) is classified as Class1
Point x54 (value: 0.9610) is classified as Class2
Point x55 (value: 0.8390) is classified as Class2
Point x56 (value: 0.7921) is classified as Class2
Point x57 (value: 0.7260) is classified as Class2
Point x58 (value: 0.2140) is classified as Class1
Point x59 (value: 0.4824) is classified as Class1
Point x60 (value: 0.8411) is classified as Class2
Point x61 (value: 0.6597) is classified as Class2
Point x62 (value: 0.3090) is classified as Class1
Point x63 (value: 0.2481) is classified as Class1
Point x64 (value: 0.2470) is classified as Class1
Point x65 (value: 0.9320) is classified as Class2
Point x66 (value: 0.9717) is classified as Class2
Point x67 (value: 0.8989) is classified as Class2
```

```
Point x68 (value: 0.4311) is classified as Class1
Point x69 (value: 0.0788) is classified as Class1
Point x70 (value: 0.0354) is classified as Class1
Point x71 (value: 0.8361) is classified as Class2
Point x72 (value: 0.4777) is classified as Class1
Point x73 (value: 0.2504) is classified as Class1
Point x74 (value: 0.6567) is classified as Class2
Point x75 (value: 0.2831) is classified as Class1
Point x76 (value: 0.9361) is classified as Class2
Point x77 (value: 0.6256) is classified as Class2
Point x78 (value: 0.4972) is classified as Class2
Point x79 (value: 0.3888) is classified as Class1
Point x80 (value: 0.4373) is classified as Class1
Point x81 (value: 0.4775) is classified as Class1
Point x82 (value: 0.2877) is classified as Class1
Point x83 (value: 0.8856) is classified as Class2
Point x84 (value: 0.2611) is classified as Class1
Point x85 (value: 0.3077) is classified as Class1
Point x86 (value: 0.9066) is classified as Class2
Point x87 (value: 0.9856) is classified as Class2
Point x88 (value: 0.3278) is classified as Class1
Point x89 (value: 0.5034) is classified as Class2
Point x90 (value: 0.5070) is classified as Class2
Point x91 (value: 0.9881) is classified as Class2
Point x92 (value: 0.8672) is classified as Class2
Point x93 (value: 0.0983) is classified as Class1
Point x94 (value: 0.9252) is classified as Class2
Point x95 (value: 0.8664) is classified as Class2
Point x96 (value: 0.9819) is classified as Class2
Point x97 (value: 0.5803) is classified as Class2
Point x98 (value: 0.6352) is classified as Class2
Point x99 (value: 0.9272) is classified as Class2
Point x100 (value: 0.0044) is classified as Class1


Results for k = 20:
Point x51 (value: 0.3537) is classified as Class1
Point x52 (value: 0.3931) is classified as Class1
Point x53 (value: 0.0939) is classified as Class1
Point x54 (value: 0.9610) is classified as Class2
Point x55 (value: 0.8390) is classified as Class2
Point x56 (value: 0.7921) is classified as Class2
Point x57 (value: 0.7260) is classified as Class2
Point x58 (value: 0.2140) is classified as Class1
Point x59 (value: 0.4824) is classified as Class1
Point x60 (value: 0.8411) is classified as Class2
Point x61 (value: 0.6597) is classified as Class2
Point x62 (value: 0.3090) is classified as Class1
Point x63 (value: 0.2481) is classified as Class1
```

```
Point x64 (value: 0.2470) is classified as Class1
Point x65 (value: 0.9320) is classified as Class2
Point x66 (value: 0.9717) is classified as Class2
Point x67 (value: 0.8989) is classified as Class2
Point x68 (value: 0.4311) is classified as Class1
Point x69 (value: 0.0788) is classified as Class1
Point x70 (value: 0.0354) is classified as Class1
Point x71 (value: 0.8361) is classified as Class2
Point x72 (value: 0.4777) is classified as Class1
Point x73 (value: 0.2504) is classified as Class1
Point x74 (value: 0.6567) is classified as Class2
Point x75 (value: 0.2831) is classified as Class1
Point x76 (value: 0.9361) is classified as Class2
Point x77 (value: 0.6256) is classified as Class2
Point x78 (value: 0.4972) is classified as Class1
Point x79 (value: 0.3888) is classified as Class1
Point x80 (value: 0.4373) is classified as Class1
Point x81 (value: 0.4775) is classified as Class1
Point x82 (value: 0.2877) is classified as Class1
Point x83 (value: 0.8856) is classified as Class2
Point x84 (value: 0.2611) is classified as Class1
Point x85 (value: 0.3077) is classified as Class1
Point x86 (value: 0.9066) is classified as Class2
Point x87 (value: 0.9856) is classified as Class2
Point x88 (value: 0.3278) is classified as Class1
Point x89 (value: 0.5034) is classified as Class1
Point x90 (value: 0.5070) is classified as Class1
Point x91 (value: 0.9881) is classified as Class2
Point x92 (value: 0.8672) is classified as Class2
Point x93 (value: 0.0983) is classified as Class1
Point x94 (value: 0.9252) is classified as Class2
Point x95 (value: 0.8664) is classified as Class2
Point x96 (value: 0.9819) is classified as Class2
Point x97 (value: 0.5803) is classified as Class2
Point x98 (value: 0.6352) is classified as Class2
Point x99 (value: 0.9272) is classified as Class2
Point x100 (value: 0.0044) is classified as Class1


Results for k = 30:
Point x51 (value: 0.3537) is classified as Class1
Point x52 (value: 0.3931) is classified as Class1
Point x53 (value: 0.0939) is classified as Class1
Point x54 (value: 0.9610) is classified as Class2
Point x55 (value: 0.8390) is classified as Class2
Point x56 (value: 0.7921) is classified as Class2
Point x57 (value: 0.7260) is classified as Class2
Point x58 (value: 0.2140) is classified as Class1
Point x59 (value: 0.4824) is classified as Class1
```

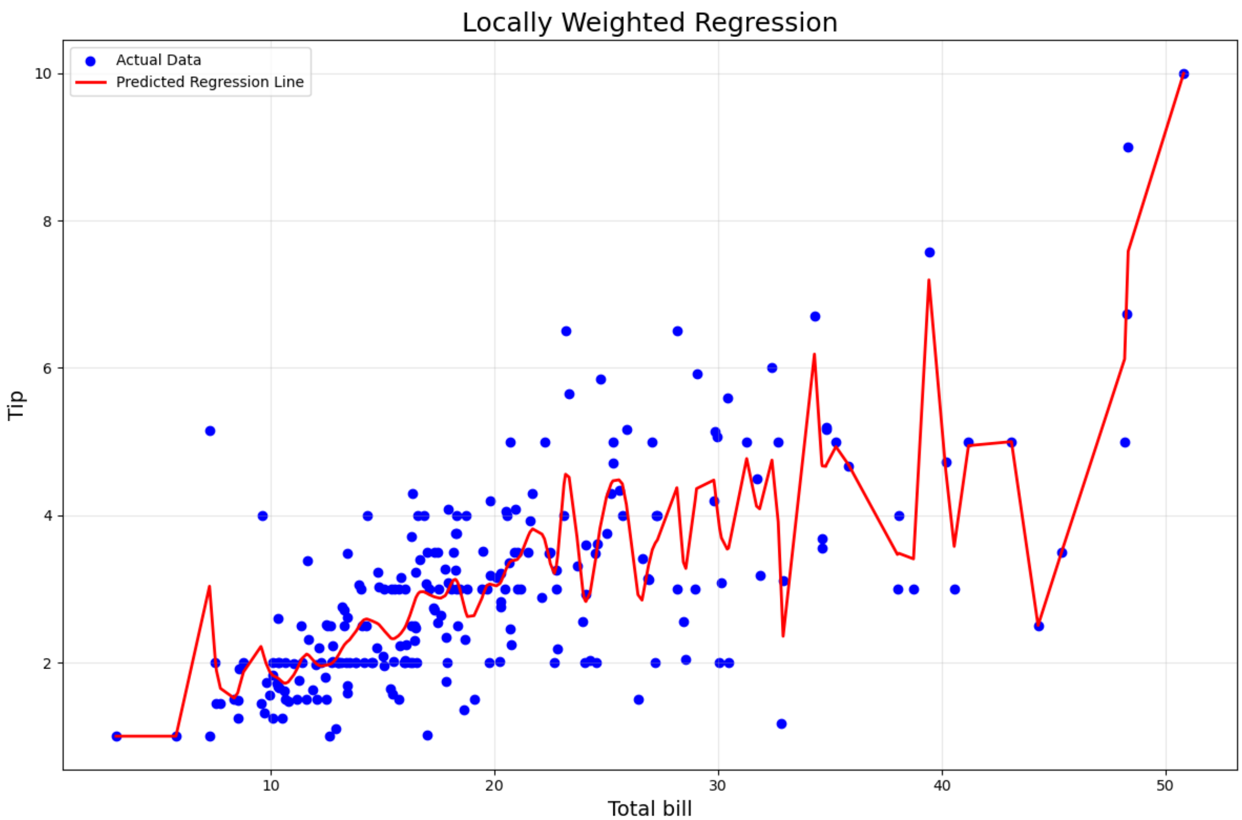
```
Point x60 (value: 0.8411) is classified as Class2
Point x61 (value: 0.6597) is classified as Class2
Point x62 (value: 0.3090) is classified as Class1
Point x63 (value: 0.2481) is classified as Class1
Point x64 (value: 0.2470) is classified as Class1
Point x65 (value: 0.9320) is classified as Class2
Point x66 (value: 0.9717) is classified as Class2
Point x67 (value: 0.8989) is classified as Class2
Point x68 (value: 0.4311) is classified as Class1
Point x69 (value: 0.0788) is classified as Class1
Point x70 (value: 0.0354) is classified as Class1
Point x71 (value: 0.8361) is classified as Class2
Point x72 (value: 0.4777) is classified as Class1
Point x73 (value: 0.2504) is classified as Class1
Point x74 (value: 0.6567) is classified as Class2
Point x75 (value: 0.2831) is classified as Class1
Point x76 (value: 0.9361) is classified as Class2
Point x77 (value: 0.6256) is classified as Class2
Point x78 (value: 0.4972) is classified as Class1
Point x79 (value: 0.3888) is classified as Class1
Point x80 (value: 0.4373) is classified as Class1
Point x81 (value: 0.4775) is classified as Class1
Point x82 (value: 0.2877) is classified as Class1
Point x83 (value: 0.8856) is classified as Class2
Point x84 (value: 0.2611) is classified as Class1
Point x85 (value: 0.3077) is classified as Class1
Point x86 (value: 0.9066) is classified as Class2
Point x87 (value: 0.9856) is classified as Class2
Point x88 (value: 0.3278) is classified as Class1
Point x89 (value: 0.5034) is classified as Class2
Point x90 (value: 0.5070) is classified as Class2
Point x91 (value: 0.9881) is classified as Class2
Point x92 (value: 0.8672) is classified as Class2
Point x93 (value: 0.0983) is classified as Class1
Point x94 (value: 0.9252) is classified as Class2
Point x95 (value: 0.8664) is classified as Class2
Point x96 (value: 0.9819) is classified as Class2
Point x97 (value: 0.5803) is classified as Class2
Point x98 (value: 0.6352) is classified as Class2
Point x99 (value: 0.9272) is classified as Class2
Point x100 (value: 0.0044) is classified as Class1


Classification complete.
```




```python
from os import listdir
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np1
import numpy.linalg as np
from scipy.stats import pearsonr

def kernel(point,xmat, k):
    m,n = np1.shape(xmat)
    weights = np1.mat(np1.eye((m))) # eye(m) returns identity matrix of size m
    for j in range(m):
```

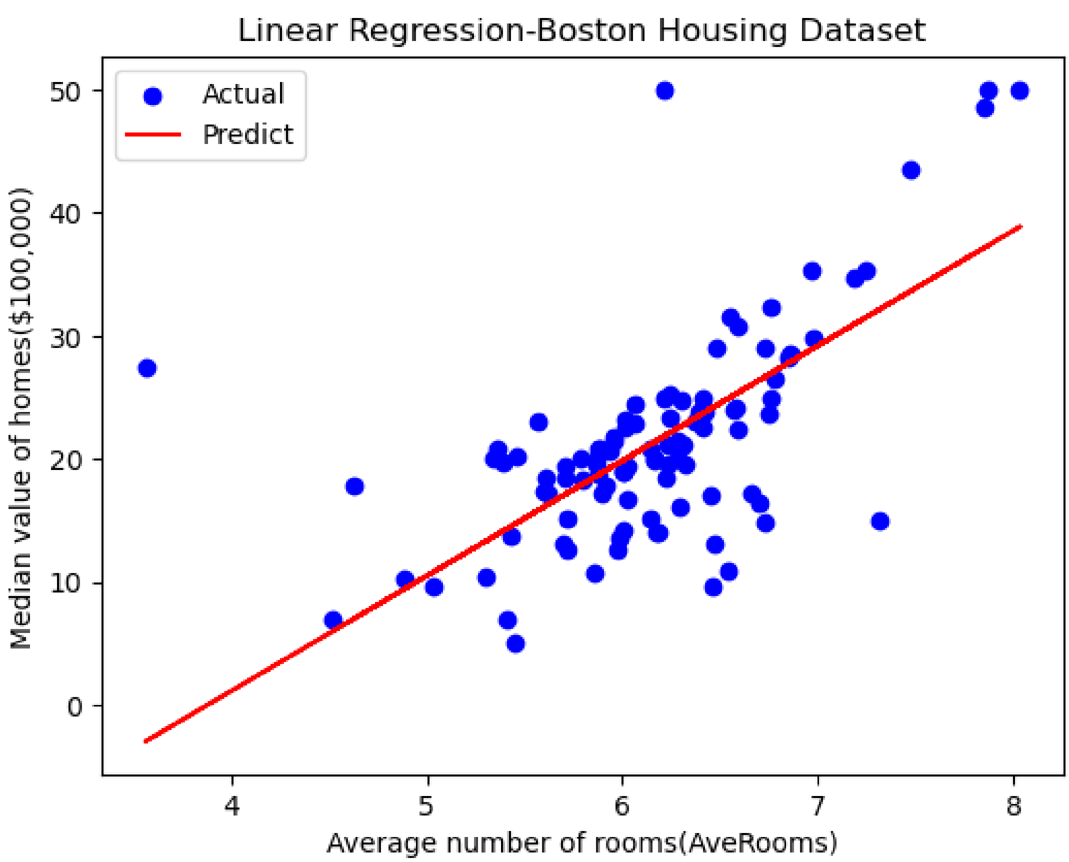
```
        diff = point - X[j]
        weights[j,j] = np1.exp(diff*diff.T/(-2.0*k**2))
    return weights

def localWeight(point,xmat,ymat,k):
    wei = kernel(point,xmat,k)
    XT_WX=X.T*(wei*X)
    XT_WX+=np1.eye(XT_WX.shape[0])*1e-5
    W = XT_WX.I*(X.T*(wei*ymat.T))

    return W

def localWeightRegression(xmat,ymat,k):
    m,n = np1.shape(xmat)
    ypred = np1.zeros(m)
    for i in range(m):
        ypred[i] = xmat[i]*localWeight(xmat[i],xmat,ymat,k)
    return ypred

data = pd.read_csv('/home/ewitise/Downloads/tips.csv')
bill = np1.array(data.total_bill)
tip = np1.array(data.tip)

mbill = np1.mat(bill)
mtip = np1.mat(tip)
m= np1.shape(mbill)[1]

one = np1.mat(np1.ones(m))
X= np1.hstack((one.T,mbill.T))

ypred = localWeightRegression(X,mtip,0.3)
SortIndex = X[:,1].argsort(0)
xsort = X[SortIndex][:,0]

fig = plt.figure(figsize=(12,8))
ax = fig.add_subplot(1,1,1)
ax.scatter(bill,tip, color='blue', label = 'Actual Data')
ax.plot(xsort[:,1],ypred[SortIndex], color = 'red', linewidth=2,
label='Predicted Regression Line')
plt.xlabel('Total bill', fontsize = 14)
plt.ylabel('Tip', fontsize = 14)

plt.title('Locally Weighted Regression', fontsize=18)
plt.legend(fontsize=10)
plt.grid(alpha=0.3)
plt.tight_layout()
plt.savefig('LWRReg.png')
plt.show();
```

```
/tmp/ipykernel_3362/4129896838.py:13: DeprecationWarning: Conversion
of an array with ndim > 0 to a scalar is deprecated, and will error in
future. Ensure you extract a single element from your array before
performing this operation. (Deprecated NumPy 1.25.)
  weights[j,j] = np1.exp(diff*diff.T/(-2.0*k**2))
/tmp/ipykernel_3362/4129896838.py:28: DeprecationWarning: Conversion
of an array with ndim > 0 to a scalar is deprecated, and will error in
future. Ensure you extract a single element from your array before
performing this operation. (Deprecated NumPy 1.25.)
  ypred[i] = xmat[i]*localWeight(xmat[i],xmat,ymat,k)
```




```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures,StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.metrics import mean_squared_error,r2_score

def linear_regression_boston():

b_housing=pd.read_csv('/home/ewitise/Downloads/BostonHousing(1).csv')
    X=b_housing[['rm']]
```

```
    y=b_housing['medv']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,rando
m_state=42)
    model=LinearRegression()
    model.fit(X_train,y_train)
    y_pred=model.predict(X_test)
    plt.scatter(X_test,y_test,color="blue",label="Actual")
    plt.plot(X_test,y_pred,color="red",label="Predict")
    plt.xlabel("Average number of rooms(AveRooms)")
    plt.ylabel("Median value of homes($100,000)")
    plt.title("Linear Regression-Boston Housing Dataset")
    plt.legend()
    plt.savefig('LinearRegression.png')
    plt.show()
    print("Linear Regression-Boston Housing dataset")
    print("Mean Squared Error:",mean_squared_error(y_test,y_pred))
    print("R^2 Score:",r2_score(y_test,y_pred))

linear_regression_boston()
```

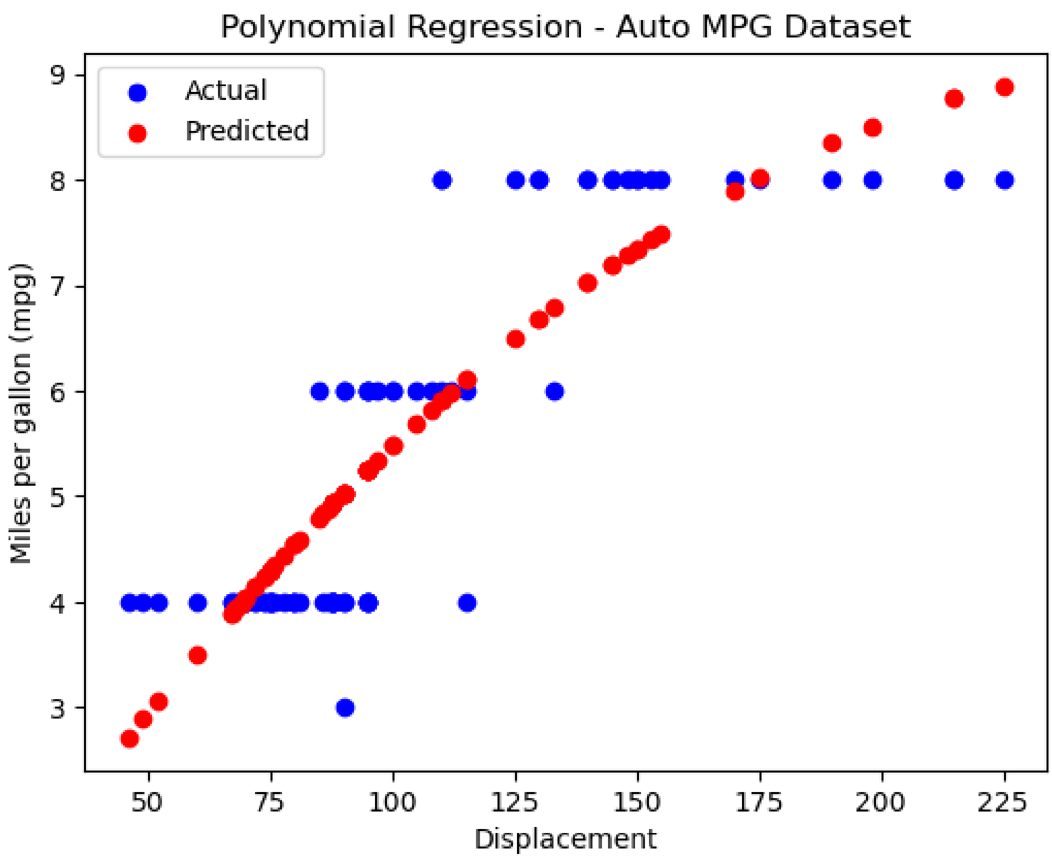


```
Linear Regression-Boston Housing dataset
Mean Squared Error: 46.144775347317264
R^2 Score: 0.3707569232254778
```

```
def polynomial_regression_auto_mpg():
    url =
"https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/
auto-mpg.data"
    column_names = ["mpg", "cylinders", "displacement", "horsepower",
"weight", "acceleration", "model_year", "origin"]
    data = pd.read_csv(url, sep='\s+', names=column_names,
na_values="?")
    data = data.dropna()

    X = data["displacement"].values.reshape(-1, 1)
    y = data["mpg"].values
    X_train, X_test, y_train,y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

    poly_model = make_pipeline(PolynomialFeatures(degree=2),
StandardScaler(), LinearRegression())
    poly_model.fit(X_train, y_train)

    y_pred = poly_model.predict(X_test)

    plt.scatter(X_test, y_test, color="blue", label="Actual")
    plt.scatter(X_test, y_pred, color="red", label="Predicted")
    plt.xlabel("Displacement")
    plt.ylabel("Miles per gallon (mpg)")
    plt.title("Polynomial Regression - Auto MPG Dataset")
    plt.legend()
    plt.savefig('PolynomialRegression.png')
    plt.show()
    print("Polynomial Regression - Auto MPG Dataset")
    print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
    print("R^2 Score:", r2_score(y_test, y_pred))
```

```
<>:4: SyntaxWarning: invalid escape sequence '\s'
<>:4: SyntaxWarning: invalid escape sequence '\s'
/tmp/ipykernel_3362/1489624616.py:4: SyntaxWarning: invalid escape
sequence '\s'
  data = pd.read_csv(url, sep='\s+', names=column_names,
na_values="?")
```

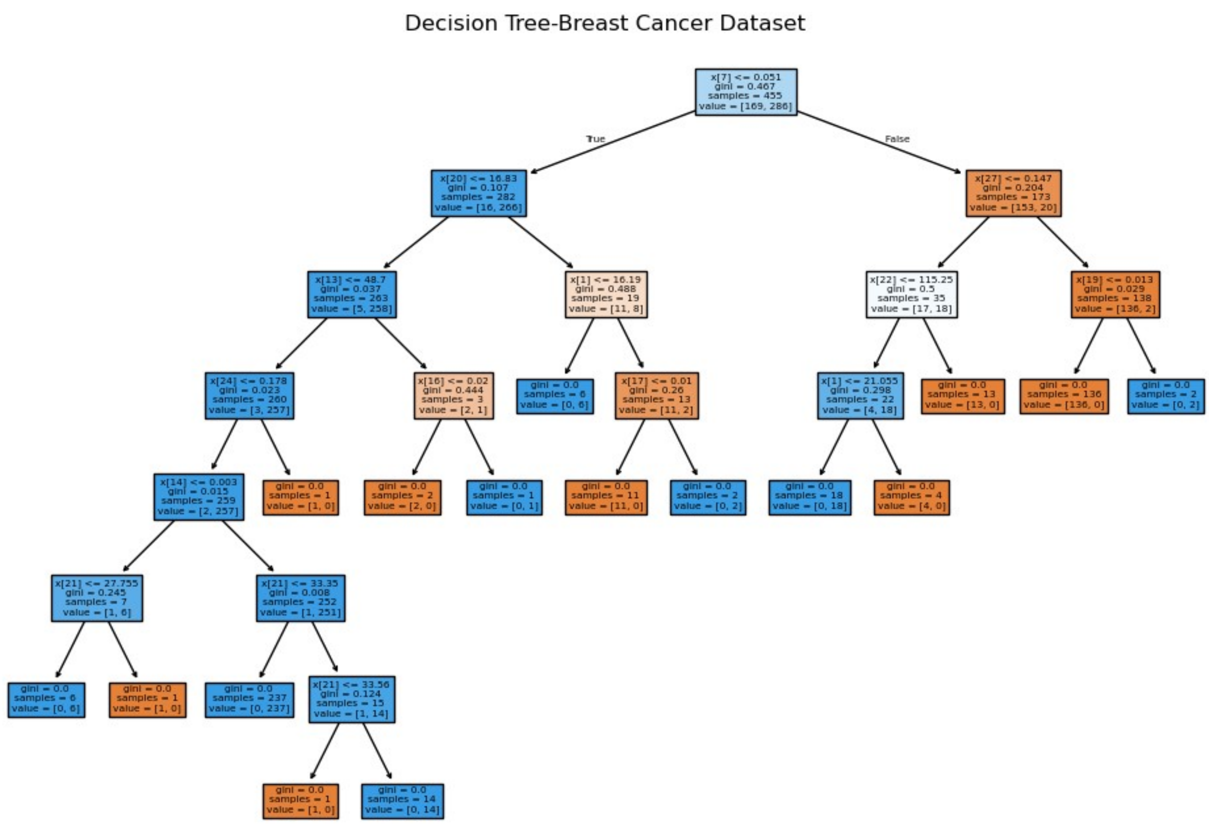
```
polynomial_regression_auto_mpg()
```




```
Polynomial Regression - Auto MPG Dataset
Mean Squared Error: 0.7431490557205862
R^2 Score: 0.7505650609469626

import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree

data = load_breast_cancer()
X = data.data
y = data.target

X_train, X_test, y_train,y_pred = train_test_split(X, y,
test_size=0.2, random_state=42)
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train,y_train)

accuracy = accuracy_score(y_test,y_pred)
print(f"Model Accuracy:{accuracy*100:.2f}%")
```

```
new_sample = np.array([X_test[0]])
prediction = clf.predict(new_sample)

prediction_class = "Benign" if prediction == 1 else "Malignant"
print(f"Prediction class for the new sample:{prediction_class}")

plt.figure(figsize=(12,8))
tree.plot_tree(clf,filled=True,label='all')
plt.title("Decision Tree-Breast Cancer Dataset")
plt.savefig('DecisionTree.png')
plt.show()
```

```
Model Accuracy:100.00%
Prediction class for the new sample:Benign
```




```
import numpy as np
from sklearn.datasets import fetch_olivetti_faces
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import matplotlib.pyplot as plt
```

```
data = fetch_olivetti_faces(shuffle=True, random_state=42)
X = data.data
y = data.target
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')
cross_val_accuracy = cross_val_score(gnb, X, y, cv=5,
scoring='accuracy')
print(f'\nCross-validation accuracy: {cross_val_accuracy.mean() *
100:.2f}%')
fig, axes = plt.subplots(3, 5, figsize=(12, 8))
for ax, image, label, prediction in zip(axes.ravel(), X_test, y_test,
y_pred):
    ax.imshow(image.reshape(64, 64), cmap=plt.cm.gray)
    ax.set_title(f"True: {label}, Pred: {prediction}")
    ax.axis('off')import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_breast_cancer
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.metrics import confusion_matrix, classification_report

data = load_breast_cancer()
X = data.data
y = data.target

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

kmeans = KMeans(n_clusters=2, random_state=42)
y_kmeans = kmeans.fit_predict(X_scaled)

print("Confusion Matrix:")
print(confusion_matrix(y, y_kmeans))
print("\nClassification Report:")
print(classification_report(y, y_kmeans))

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

df = pd.DataFrame(X_pca, columns=['PC1', 'PC2'])
df['Cluster'] = y_kmeans
df['True Label'] = y
```

```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='PC1', y='PC2', hue='Cluster',
palette='Set1', s=100, edgecolor='black', alpha=0.7)
plt.title('K-Means Clustering of Breast Cancer Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title="Cluster")
plt.savefig('ep101.png')
plt.show()

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='PC1', y='PC2', hue='True Label',
palette='coolwarm', s=100, edgecolor='black', alpha=0.7)
plt.title('True Labels of Breast Cancer Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title="True Label")
plt.savefig('ep102.png')
plt.show()

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='PC1', y='PC2', hue='Cluster',
palette='Set1', s=100, edgecolor='black', alpha=0.7)
centers = pca.transform(kmeans.cluster_centers_)
plt.scatter(centers[:, 0], centers[:, 1], s=200, c='red', marker='X',
label='Centroids')
plt.title('K-Means Clustering with Centroids')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title="Cluster")
plt.savefig('ep103.png')
plt.show()
plt.savefig('exp9-1.png')
plt.show()
```

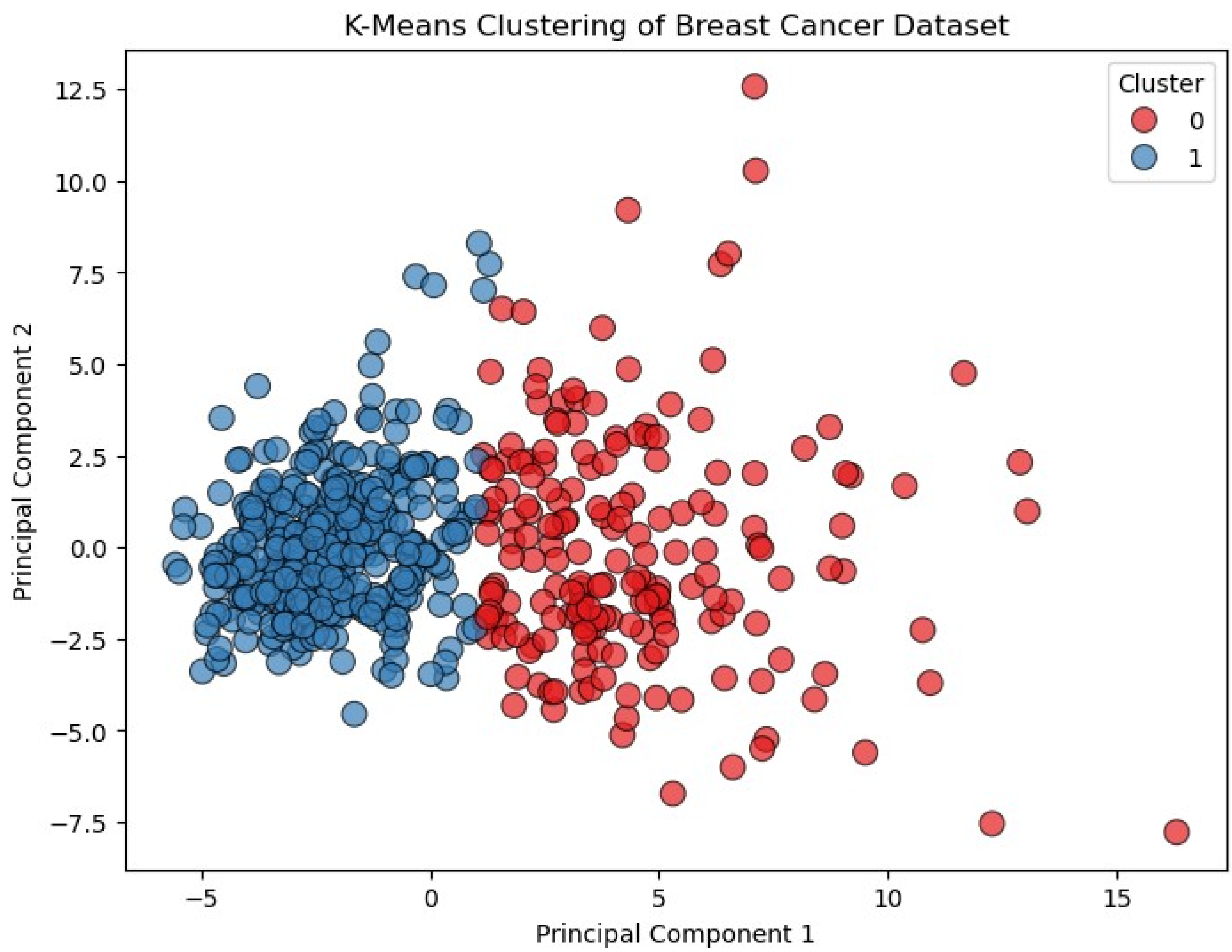
```
Accuracy: 80.83%

Cross-validation accuracy: 87.25%
```

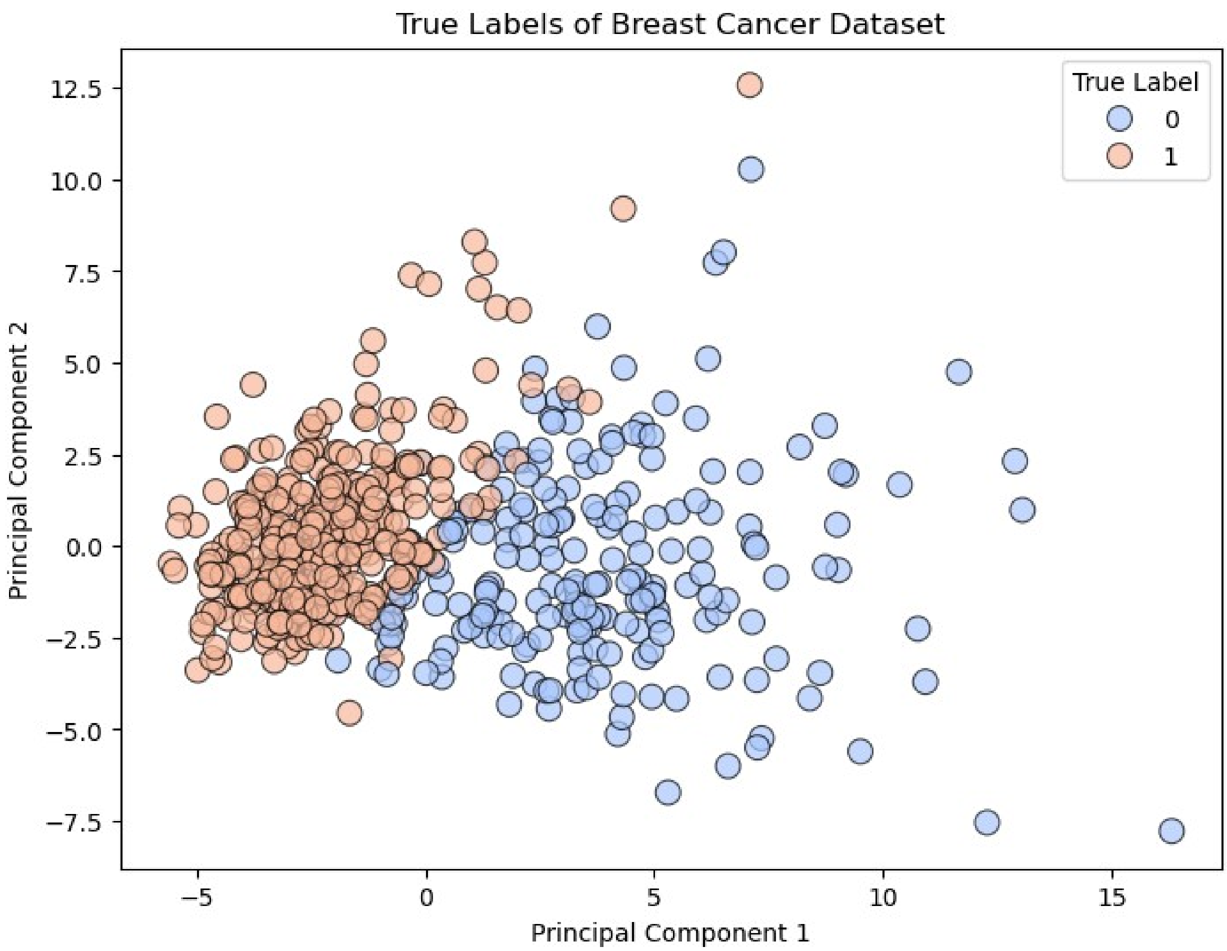

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_breast_cancer
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.metrics import confusion_matrix, classification_report

data = load_breast_cancer()
X = data.data
y = data.target

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

kmeans = KMeans(n_clusters=2, random_state=42)
y_kmeans = kmeans.fit_predict(X_scaled)

print("Confusion Matrix:")
print(confusion_matrix(y, y_kmeans))
print("\nClassification Report:")
print(classification_report(y, y_kmeans))
```

```
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

df = pd.DataFrame(X_pca, columns=['PC1', 'PC2'])
df['Cluster'] = y_kmeans
df['True Label'] = y

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='PC1', y='PC2', hue='Cluster',
palette='Set1', s=100, edgecolor='black', alpha=0.7)
plt.title('K-Means Clustering of Breast Cancer Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title="Cluster")
plt.savefig('ep101.png')
plt.show()

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='PC1', y='PC2', hue='True Label',
palette='coolwarm', s=100, edgecolor='black', alpha=0.7)
plt.title('True Labels of Breast Cancer Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title="True Label")
plt.savefig('ep102.png')
plt.show()

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='PC1', y='PC2', hue='Cluster',
palette='Set1', s=100, edgecolor='black', alpha=0.7)
centers = pca.transform(kmeans.cluster_centers_)
plt.scatter(centers[:, 0], centers[:, 1], s=200, c='red', marker='X',
label='Centroids')
plt.title('K-Means Clustering with Centroids')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title="Cluster")
plt.savefig('ep103.png')
plt.show()
```

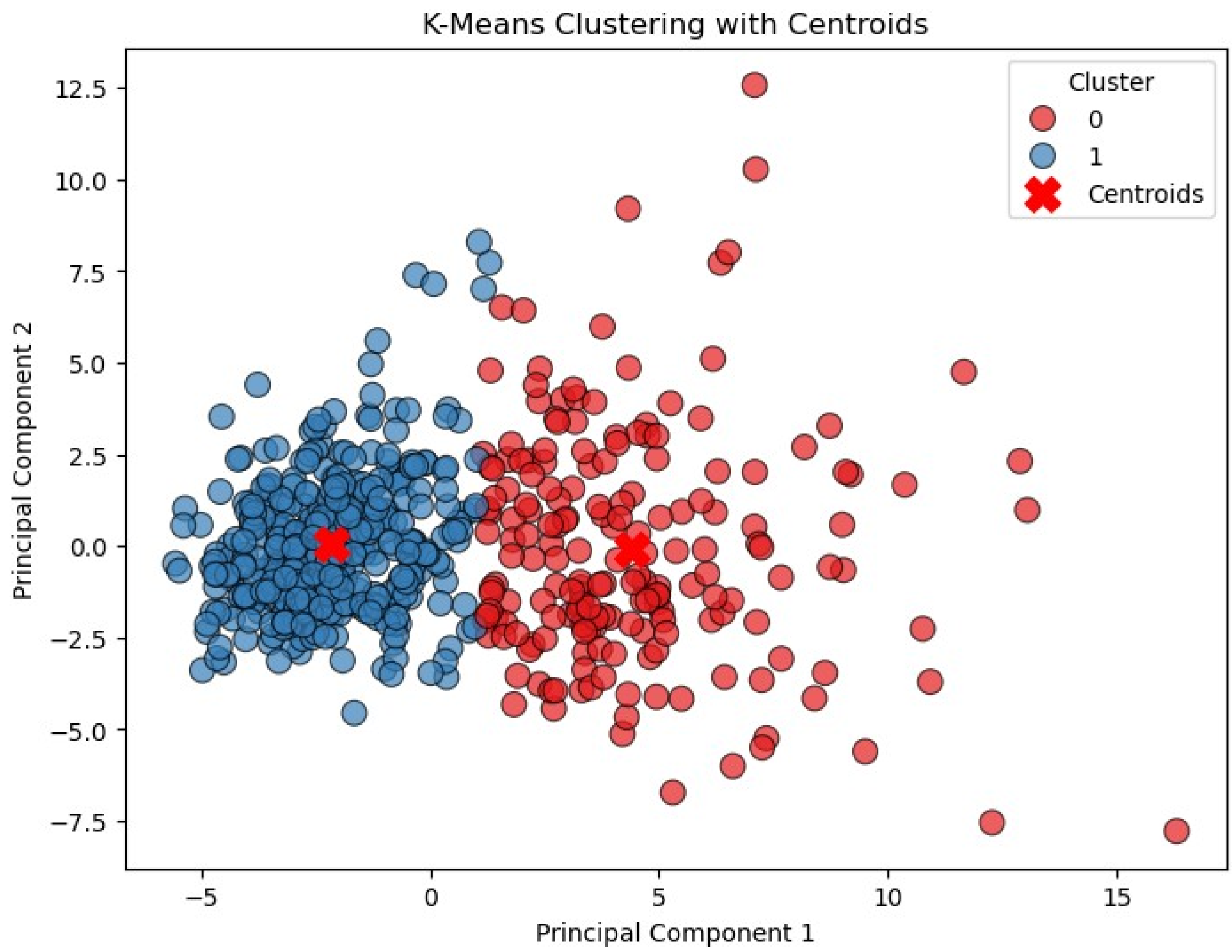
```
Confusion Matrix:
[[175  37]
 [ 13 344]]

Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.83      0.88       212
           1       0.90      0.96      0.93       357
```

```
    accuracy                           0.91       569
   macro avg       0.92      0.89      0.90       569
weighted avg       0.91      0.91      0.91       569
```





True Labels of Breast Cancer Dataset
True Label
0
1
Principal Component 2
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−5
0
5
10
15
Principal Component 1


K-Means Clustering with Centroids
Cluster
0
1
Centroids
Principal Component 1
Principal Component 2
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−5
0
5
10
15