knn-algorithm

March 26, 2025

0.1 Develop a program to implement k-Nearest Neighbour algorithm to classify the randomly generated 100 values of x in the range of [0,1]. Perform the following based on dataset generated.

1. Label the first 50 points {x1,……,x50} as follows: if (xi 0.5), then xi Class1, else xi Class1
2. Classify the remaining points, x51,……,x100 using KNN. Perform this for k=1,2,3,4,5,20,30

[22]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

import warnings
warnings.filterwarnings('ignore')
```

[23]:
```
# Step 1: Generate dataset
np.random.seed(42)
values = np.random.rand(100)
print(values)
```

```
[0.37454012 0.95071431 0.73199394 0.59865848 0.15601864 0.15599452
 0.05808361 0.86617615 0.60111501 0.70807258 0.02058449 0.96990985
 0.83244264 0.21233911 0.18182497 0.18340451 0.30424224 0.52475643
 0.43194502 0.29122914 0.61185289 0.13949386 0.29214465 0.36636184
 0.45606998 0.78517596 0.19967378 0.51423444 0.59241457 0.04645041
 0.60754485 0.17052412 0.06505159 0.94888554 0.96563203 0.80839735
 0.30461377 0.09767211 0.68423303 0.44015249 0.12203823 0.49517691
 0.03438852 0.9093204  0.25877998 0.66252228 0.31171108 0.52006802
 0.54671028 0.18485446 0.96958463 0.77513282 0.93949894 0.89482735
 0.59789998 0.92187424 0.0884925  0.19598286 0.04522729 0.32533033
 0.38867729 0.27134903 0.82873751 0.35675333 0.28093451 0.54269608
 0.14092422 0.80219698 0.07455064 0.98688694 0.77224477 0.19871568
 0.00552212 0.81546143 0.70685734 0.72900717 0.77127035 0.07404465
 0.35846573 0.11586906 0.86310343 0.62329813 0.33089802 0.06355835
 0.31098232 0.32518332 0.72960618 0.63755747 0.88721274 0.47221493
```

```
 0.11959425 0.71324479 0.76078505 0.5612772  0.77096718 0.4937956
 0.52273283 0.42754102 0.02541913 0.10789143]
```

[24]:
```python
labels = []

for i in values[:50]:
    if i <=0.5:
        labels.append('Class1')
    else:
        labels.append('Class2')
print(labels)
```

```
['Class1', 'Class2', 'Class2', 'Class2', 'Class1', 'Class1', 'Class1', 'Class2',
'Class2', 'Class2', 'Class1', 'Class2', 'Class2', 'Class1', 'Class1', 'Class1',
'Class1', 'Class2', 'Class1', 'Class1', 'Class2', 'Class1', 'Class1', 'Class1',
'Class1', 'Class2', 'Class1', 'Class2', 'Class2', 'Class1', 'Class2', 'Class1',
'Class1', 'Class2', 'Class2', 'Class2', 'Class1', 'Class1', 'Class2', 'Class1',
'Class1', 'Class1', 'Class1', 'Class2', 'Class1', 'Class2', 'Class1', 'Class2',
'Class2', 'Class1']
```

[25]:
```python
labels += [None] * 50
print(labels)
```

```
['Class1', 'Class2', 'Class2', 'Class2', 'Class1', 'Class1', 'Class1', 'Class2',
'Class2', 'Class2', 'Class1', 'Class2', 'Class2', 'Class1', 'Class1', 'Class1',
'Class1', 'Class2', 'Class1', 'Class1', 'Class2', 'Class1', 'Class1', 'Class1',
'Class1', 'Class2', 'Class1', 'Class2', 'Class2', 'Class1', 'Class2', 'Class1',
'Class1', 'Class2', 'Class2', 'Class2', 'Class1', 'Class1', 'Class2', 'Class1',
'Class1', 'Class1', 'Class1', 'Class2', 'Class1', 'Class2', 'Class1', 'Class2',
'Class2', 'Class1', None, None, None, None, None, None, None, None, None, None,
None, None, None, None, None, None, None, None, None, None, None, None, None,
None, None, None, None, None, None, None, None, None, None, None, None, None,
None, None, None, None, None, None, None, None, None, None, None, None, None,
None]
```

[26]:
```python
print(labels)
```

```
['Class1', 'Class2', 'Class2', 'Class2', 'Class1', 'Class1', 'Class1', 'Class2',
'Class2', 'Class2', 'Class1', 'Class2', 'Class2', 'Class1', 'Class1', 'Class1',
'Class1', 'Class2', 'Class1', 'Class1', 'Class2', 'Class1', 'Class1', 'Class1',
'Class1', 'Class2', 'Class1', 'Class2', 'Class2', 'Class1', 'Class2', 'Class1',
'Class1', 'Class2', 'Class2', 'Class2', 'Class1', 'Class1', 'Class2', 'Class1',
'Class1', 'Class1', 'Class1', 'Class2', 'Class1', 'Class2', 'Class1', 'Class2',
'Class2', 'Class1', None, None, None, None, None, None, None, None, None, None,
None, None, None, None, None, None, None, None, None, None, None, None, None,
None, None, None, None, None, None, None, None, None, None, None, None, None,
None, None, None, None, None, None, None, None, None, None, None, None, None,
None]
```

[27]:
```python
data = {
    "Point": [f"x{i+1}" for i in range(100)],
    "Value": values,
    "Label": labels
}

print(data)
type(data)
```

```
{'Point': ['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8', 'x9', 'x10', 'x11',
'x12', 'x13', 'x14', 'x15', 'x16', 'x17', 'x18', 'x19', 'x20', 'x21', 'x22',
'x23', 'x24', 'x25', 'x26', 'x27', 'x28', 'x29', 'x30', 'x31', 'x32', 'x33',
'x34', 'x35', 'x36', 'x37', 'x38', 'x39', 'x40', 'x41', 'x42', 'x43', 'x44',
'x45', 'x46', 'x47', 'x48', 'x49', 'x50', 'x51', 'x52', 'x53', 'x54', 'x55',
'x56', 'x57', 'x58', 'x59', 'x60', 'x61', 'x62', 'x63', 'x64', 'x65', 'x66',
'x67', 'x68', 'x69', 'x70', 'x71', 'x72', 'x73', 'x74', 'x75', 'x76', 'x77',
'x78', 'x79', 'x80', 'x81', 'x82', 'x83', 'x84', 'x85', 'x86', 'x87', 'x88',
'x89', 'x90', 'x91', 'x92', 'x93', 'x94', 'x95', 'x96', 'x97', 'x98', 'x99',
'x100'], 'Value': array([0.37454012, 0.95071431, 0.73199394, 0.59865848,
0.15601864,
       0.15599452, 0.05808361, 0.86617615, 0.60111501, 0.70807258,
       0.02058449, 0.96990985, 0.83244264, 0.21233911, 0.18182497,
       0.18340451, 0.30424224, 0.52475643, 0.43194502, 0.29122914,
       0.61185289, 0.13949386, 0.29214465, 0.36636184, 0.45606998,
       0.78517596, 0.19967378, 0.51423444, 0.59241457, 0.04645041,
       0.60754485, 0.17052412, 0.06505159, 0.94888554, 0.96563203,
       0.80839735, 0.30461377, 0.09767211, 0.68423303, 0.44015249,
       0.12203823, 0.49517691, 0.03438852, 0.9093204 , 0.25877998,
       0.66252228, 0.31171108, 0.52006802, 0.54671028, 0.18485446,
       0.96958463, 0.77513282, 0.93949894, 0.89482735, 0.59789998,
       0.92187424, 0.0884925 , 0.19598286, 0.04522729, 0.32533033,
       0.38867729, 0.27134903, 0.82873751, 0.35675333, 0.28093451,
       0.54269608, 0.14092422, 0.80219698, 0.07455064, 0.98688694,
       0.77224477, 0.19871568, 0.00552212, 0.81546143, 0.70685734,
       0.72900717, 0.77127035, 0.07404465, 0.35846573, 0.11586906,
       0.86310343, 0.62329813, 0.33089802, 0.06355835, 0.31098232,
       0.32518332, 0.72960618, 0.63755747, 0.88721274, 0.47221493,
       0.11959425, 0.71324479, 0.76078505, 0.5612772 , 0.77096718,
       0.4937956 , 0.52273283, 0.42754102, 0.02541913, 0.10789143]), 'Label':
['Class1', 'Class2', 'Class2', 'Class2', 'Class1', 'Class1', 'Class1', 'Class2',
'Class2', 'Class2', 'Class1', 'Class2', 'Class2', 'Class1', 'Class1', 'Class1',
'Class1', 'Class2', 'Class1', 'Class1', 'Class2', 'Class1', 'Class1', 'Class1',
'Class1', 'Class2', 'Class1', 'Class2', 'Class2', 'Class1', 'Class2', 'Class1',
'Class1', 'Class2', 'Class2', 'Class2', 'Class1', 'Class1', 'Class2', 'Class1',
'Class1', 'Class1', 'Class1', 'Class2', 'Class1', 'Class2', 'Class1', 'Class2',
'Class2', 'Class1', None, None, None, None, None, None, None, None, None, None,
None, None, None, None, None, None, None, None, None, None, None, None, None,
```

```
None, None, None, None, None, None, None, None, None, None, None, None, None,
None, None, None, None, None, None, None, None, None, None, None, None, None,
None]}
```

[27]:
```
dict
```

[28]:
```
df = pd.DataFrame(data)
df.head()
```

[28]:

	Point	Value	Label
0	x1	0.374540	Class1
1	x2	0.950714	Class2
2	x3	0.731994	Class2
3	x4	0.598658	Class2
4	x5	0.156019	Class1

[29]:
```
df.nunique()
```

[29]:
```
Point    100
Value    100
Label      2
dtype: int64
```

[30]:
```
df.shape
```

[30]:
```
(100, 3)
```

[31]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Point   100 non-null    object
 1   Value   100 non-null    float64
 2   Label   50 non-null     object
dtypes: float64(1), object(2)
memory usage: 2.5+ KB
```

[32]:
```
df.describe().T
```

[32]:

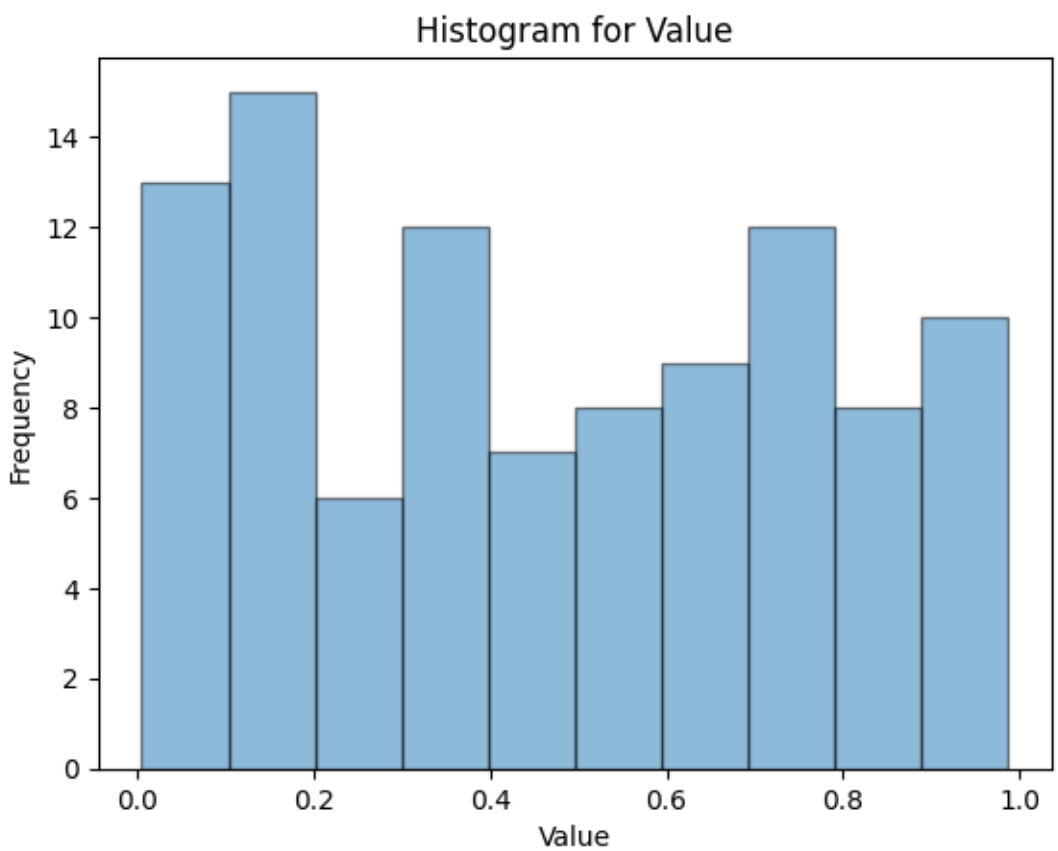
	count	mean	std	min	25%	50%	75%	\
Value	100.0	0.470181	0.297489	0.005522	0.193201	0.464142	0.730203	

	max
Value	0.986887

[33]:
```
df.isnull().sum()
```

[33]:
```
Point     0
Value     0
Label    50
dtype: int64
```

[34]:
```
num_col = df.select_dtypes(include=['int', 'float']).columns
for col in num_col:
    df[col].hist(bins=10, alpha=0.5, edgecolor='black',grid=False)
    plt.title(f'Histogram for {col}')
    plt.xlabel(col)
    plt.ylabel('Frequency')
    plt.show()
```



[35]:
```
# Split data into labeled and unlabeled
labeled_df = df[df["Label"].notna()]
X_train = labeled_df[["Value"]]
y_train = labeled_df["Label"]
print(X_train)
```

```python
print(y_train)
```

```
       Value
0   0.374540
1   0.950714
2   0.731994
3   0.598658
4   0.156019
5   0.155995
6   0.058084
7   0.866176
8   0.601115
9   0.708073
10  0.020584
11  0.969910
12  0.832443
13  0.212339
14  0.181825
15  0.183405
16  0.304242
17  0.524756
18  0.431945
19  0.291229
20  0.611853
21  0.139494
22  0.292145
23  0.366362
24  0.456070
25  0.785176
26  0.199674
27  0.514234
28  0.592415
29  0.046450
30  0.607545
31  0.170524
32  0.065052
33  0.948886
34  0.965632
35  0.808397
36  0.304614
37  0.097672
38  0.684233
39  0.440152
40  0.122038
41  0.495177
42  0.034389
43  0.909320
```

```
44  0.258780
45  0.662522
46  0.311711
47  0.520068
48  0.546710
49  0.184854
0     Class1
1     Class2
2     Class2
3     Class2
4     Class1
5     Class1
6     Class1
7     Class2
8     Class2
9     Class2
10    Class1
11    Class2
12    Class2
13    Class1
14    Class1
15    Class1
16    Class1
17    Class2
18    Class1
19    Class1
20    Class2
21    Class1
22    Class1
23    Class1
24    Class1
25    Class2
26    Class1
27    Class2
28    Class2
29    Class1
30    Class2
31    Class1
32    Class1
33    Class2
34    Class2
35    Class2
36    Class1
37    Class1
38    Class2
39    Class1
40    Class1
41    Class1
```

```
42    Class1
43    Class2
44    Class1
45    Class2
46    Class1
47    Class2
48    Class2
49    Class1
Name: Label, dtype: object
```

[36]:
```python
unlabeled_df = df[df["Label"].isna()]
X_test = unlabeled_df[["Value"]]
print(X_test)
```

```
       Value
50  0.969585
51  0.775133
52  0.939499
53  0.894827
54  0.597900
55  0.921874
56  0.088493
57  0.195983
58  0.045227
59  0.325330
60  0.388677
61  0.271349
62  0.828738
63  0.356753
64  0.280935
65  0.542696
66  0.140924
67  0.802197
68  0.074551
69  0.986887
70  0.772245
71  0.198716
72  0.005522
73  0.815461
74  0.706857
75  0.729007
76  0.771270
77  0.074045
78  0.358466
79  0.115869
80  0.863103
81  0.623298
82  0.330898
```

```
83  0.063558
84  0.310982
85  0.325183
86  0.729606
87  0.637557
88  0.887213
89  0.472215
90  0.119594
91  0.713245
92  0.760785
93  0.561277
94  0.770967
95  0.493796
96  0.522733
97  0.427541
98  0.025419
99  0.107891
```

[37]:
```python
# Generate true labels for testing (for accuracy calculation)
true_labels = ["Class1" if x <= 0.5 else "Class2" for x in values[50:]]
print(true_labels)
```

```
['Class2', 'Class2', 'Class2', 'Class2', 'Class2', 'Class2', 'Class1', 'Class1',
'Class1', 'Class1', 'Class1', 'Class1', 'Class2', 'Class1', 'Class1', 'Class2',
'Class1', 'Class2', 'Class1', 'Class2', 'Class2', 'Class1', 'Class1', 'Class2',
'Class2', 'Class2', 'Class2', 'Class1', 'Class1', 'Class1', 'Class2', 'Class2',
'Class1', 'Class1', 'Class1', 'Class1', 'Class2', 'Class2', 'Class2', 'Class1',
'Class1', 'Class2', 'Class2', 'Class2', 'Class2', 'Class1', 'Class2', 'Class1',
'Class1', 'Class1']
```

[38]:
```python
# Step 2: Perform KNN classification for different values of k
k_values = [1, 2, 3, 4, 5, 20, 30]
results = {}
accuracies = {}
print(results)
print(accuracies)
```

```
{}
{}
```

[39]:
```python
for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    predictions = knn.predict(X_test)
    results[k] = predictions

    # Calculate accuracy
    accuracy = accuracy_score(true_labels, predictions) * 100
```

```
    accuracies[k] = accuracy
    print(f"Accuracy for k={k}: {accuracy:.2f}%")

    # Assign predictions back to the DataFrame for this k
    unlabeled_df[f"Label_k{k}"] = predictions
```

```
Accuracy for k=1: 100.00%
Accuracy for k=2: 100.00%
Accuracy for k=3: 98.00%
Accuracy for k=4: 98.00%
Accuracy for k=5: 98.00%
Accuracy for k=20: 98.00%
Accuracy for k=30: 100.00%
```

[40]:
```
print(predictions)
```

```
['Class2' 'Class2' 'Class2' 'Class2' 'Class2' 'Class2' 'Class1' 'Class1'
 'Class1' 'Class1' 'Class1' 'Class1' 'Class2' 'Class1' 'Class1' 'Class2'
 'Class1' 'Class2' 'Class1' 'Class2' 'Class2' 'Class1' 'Class1' 'Class2'
 'Class2' 'Class2' 'Class2' 'Class1' 'Class1' 'Class1' 'Class2' 'Class2'
 'Class1' 'Class1' 'Class1' 'Class1' 'Class2' 'Class2' 'Class2' 'Class1'
 'Class1' 'Class2' 'Class2' 'Class2' 'Class2' 'Class1' 'Class2' 'Class1'
 'Class1' 'Class1']
```

[41]:
```
print(unlabeled_df)
```

```
   Point     Value Label Label_k1 Label_k2 Label_k3 Label_k4 Label_k5  \
50   x51  0.969585  None   Class2   Class2   Class2   Class2   Class2
51   x52  0.775133  None   Class2   Class2   Class2   Class2   Class2
52   x53  0.939499  None   Class2   Class2   Class2   Class2   Class2
53   x54  0.894827  None   Class2   Class2   Class2   Class2   Class2
54   x55  0.597900  None   Class2   Class2   Class2   Class2   Class2
55   x56  0.921874  None   Class2   Class2   Class2   Class2   Class2
56   x57  0.088493  None   Class1   Class1   Class1   Class1   Class1
57   x58  0.195983  None   Class1   Class1   Class1   Class1   Class1
58   x59  0.045227  None   Class1   Class1   Class1   Class1   Class1
59   x60  0.325330  None   Class1   Class1   Class1   Class1   Class1
60   x61  0.388677  None   Class1   Class1   Class1   Class1   Class1
61   x62  0.271349  None   Class1   Class1   Class1   Class1   Class1
62   x63  0.828738  None   Class2   Class2   Class2   Class2   Class2
63   x64  0.356753  None   Class1   Class1   Class1   Class1   Class1
64   x65  0.280935  None   Class1   Class1   Class1   Class1   Class1
65   x66  0.542696  None   Class2   Class2   Class2   Class2   Class2
66   x67  0.140924  None   Class1   Class1   Class1   Class1   Class1
67   x68  0.802197  None   Class2   Class2   Class2   Class2   Class2
68   x69  0.074551  None   Class1   Class1   Class1   Class1   Class1
69   x70  0.986887  None   Class2   Class2   Class2   Class2   Class2
70   x71  0.772245  None   Class2   Class2   Class2   Class2   Class2
```

71	x72	0.198716	None	Class1	Class1	Class1	Class1	Class1
72	x73	0.005522	None	Class1	Class1	Class1	Class1	Class1
73	x74	0.815461	None	Class2	Class2	Class2	Class2	Class2
74	x75	0.706857	None	Class2	Class2	Class2	Class2	Class2
75	x76	0.729007	None	Class2	Class2	Class2	Class2	Class2
76	x77	0.771270	None	Class2	Class2	Class2	Class2	Class2
77	x78	0.074045	None	Class1	Class1	Class1	Class1	Class1
78	x79	0.358466	None	Class1	Class1	Class1	Class1	Class1
79	x80	0.115869	None	Class1	Class1	Class1	Class1	Class1
80	x81	0.863103	None	Class2	Class2	Class2	Class2	Class2
81	x82	0.623298	None	Class2	Class2	Class2	Class2	Class2
82	x83	0.330898	None	Class1	Class1	Class1	Class1	Class1
83	x84	0.063558	None	Class1	Class1	Class1	Class1	Class1
84	x85	0.310982	None	Class1	Class1	Class1	Class1	Class1
85	x86	0.325183	None	Class1	Class1	Class1	Class1	Class1
86	x87	0.729606	None	Class2	Class2	Class2	Class2	Class2
87	x88	0.637557	None	Class2	Class2	Class2	Class2	Class2
88	x89	0.887213	None	Class2	Class2	Class2	Class2	Class2
89	x90	0.472215	None	Class1	Class1	Class1	Class1	Class1
90	x91	0.119594	None	Class1	Class1	Class1	Class1	Class1
91	x92	0.713245	None	Class2	Class2	Class2	Class2	Class2
92	x93	0.760785	None	Class2	Class2	Class2	Class2	Class2
93	x94	0.561277	None	Class2	Class2	Class2	Class2	Class2
94	x95	0.770967	None	Class2	Class2	Class2	Class2	Class2
95	x96	0.493796	None	Class1	Class1	Class2	Class2	Class2
96	x97	0.522733	None	Class2	Class2	Class2	Class2	Class2
97	x98	0.427541	None	Class1	Class1	Class1	Class1	Class1
98	x99	0.025419	None	Class1	Class1	Class1	Class1	Class1
99	x100	0.107891	None	Class1	Class1	Class1	Class1	Class1

	Label_k20	Label_k30
50	Class2	Class2
51	Class2	Class2
52	Class2	Class2
53	Class2	Class2
54	Class2	Class2
55	Class2	Class2
56	Class1	Class1
57	Class1	Class1
58	Class1	Class1
59	Class1	Class1
60	Class1	Class1
61	Class1	Class1
62	Class2	Class2
63	Class1	Class1
64	Class1	Class1
65	Class2	Class2
66	Class1	Class1

```
67    Class2    Class2
68    Class1    Class1
69    Class2    Class2
70    Class2    Class2
71    Class1    Class1
72    Class1    Class1
73    Class2    Class2
74    Class2    Class2
75    Class2    Class2
76    Class2    Class2
77    Class1    Class1
78    Class1    Class1
79    Class1    Class1
80    Class2    Class2
81    Class2    Class2
82    Class1    Class1
83    Class1    Class1
84    Class1    Class1
85    Class1    Class1
86    Class2    Class2
87    Class2    Class2
88    Class2    Class2
89    Class1    Class1
90    Class1    Class1
91    Class2    Class2
92    Class2    Class2
93    Class2    Class2
94    Class2    Class2
95    Class2    Class1
96    Class2    Class2
97    Class1    Class1
98    Class1    Class1
99    Class1    Class1
```

[42]:
```
df1 = unlabeled_df.drop(columns=['Label'], axis=1)
df1
```

[42]:
```
    Point     Value Label_k1 Label_k2 Label_k3 Label_k4 Label_k5 Label_k20  \
50    x51  0.969585   Class2   Class2   Class2   Class2   Class2    Class2
51    x52  0.775133   Class2   Class2   Class2   Class2   Class2    Class2
52    x53  0.939499   Class2   Class2   Class2   Class2   Class2    Class2
53    x54  0.894827   Class2   Class2   Class2   Class2   Class2    Class2
54    x55  0.597900   Class2   Class2   Class2   Class2   Class2    Class2
55    x56  0.921874   Class2   Class2   Class2   Class2   Class2    Class2
56    x57  0.088493   Class1   Class1   Class1   Class1   Class1    Class1
57    x58  0.195983   Class1   Class1   Class1   Class1   Class1    Class1
58    x59  0.045227   Class1   Class1   Class1   Class1   Class1    Class1
```

59	x60	0.325330	Class1	Class1	Class1	Class1	Class1	Class1
60	x61	0.388677	Class1	Class1	Class1	Class1	Class1	Class1
61	x62	0.271349	Class1	Class1	Class1	Class1	Class1	Class1
62	x63	0.828738	Class2	Class2	Class2	Class2	Class2	Class2
63	x64	0.356753	Class1	Class1	Class1	Class1	Class1	Class1
64	x65	0.280935	Class1	Class1	Class1	Class1	Class1	Class1
65	x66	0.542696	Class2	Class2	Class2	Class2	Class2	Class2
66	x67	0.140924	Class1	Class1	Class1	Class1	Class1	Class1
67	x68	0.802197	Class2	Class2	Class2	Class2	Class2	Class2
68	x69	0.074551	Class1	Class1	Class1	Class1	Class1	Class1
69	x70	0.986887	Class2	Class2	Class2	Class2	Class2	Class2
70	x71	0.772245	Class2	Class2	Class2	Class2	Class2	Class2
71	x72	0.198716	Class1	Class1	Class1	Class1	Class1	Class1
72	x73	0.005522	Class1	Class1	Class1	Class1	Class1	Class1
73	x74	0.815461	Class2	Class2	Class2	Class2	Class2	Class2
74	x75	0.706857	Class2	Class2	Class2	Class2	Class2	Class2
75	x76	0.729007	Class2	Class2	Class2	Class2	Class2	Class2
76	x77	0.771270	Class2	Class2	Class2	Class2	Class2	Class2
77	x78	0.074045	Class1	Class1	Class1	Class1	Class1	Class1
78	x79	0.358466	Class1	Class1	Class1	Class1	Class1	Class1
79	x80	0.115869	Class1	Class1	Class1	Class1	Class1	Class1
80	x81	0.863103	Class2	Class2	Class2	Class2	Class2	Class2
81	x82	0.623298	Class2	Class2	Class2	Class2	Class2	Class2
82	x83	0.330898	Class1	Class1	Class1	Class1	Class1	Class1
83	x84	0.063558	Class1	Class1	Class1	Class1	Class1	Class1
84	x85	0.310982	Class1	Class1	Class1	Class1	Class1	Class1
85	x86	0.325183	Class1	Class1	Class1	Class1	Class1	Class1
86	x87	0.729606	Class2	Class2	Class2	Class2	Class2	Class2
87	x88	0.637557	Class2	Class2	Class2	Class2	Class2	Class2
88	x89	0.887213	Class2	Class2	Class2	Class2	Class2	Class2
89	x90	0.472215	Class1	Class1	Class1	Class1	Class1	Class1
90	x91	0.119594	Class1	Class1	Class1	Class1	Class1	Class1
91	x92	0.713245	Class2	Class2	Class2	Class2	Class2	Class2
92	x93	0.760785	Class2	Class2	Class2	Class2	Class2	Class2
93	x94	0.561277	Class2	Class2	Class2	Class2	Class2	Class2
94	x95	0.770967	Class2	Class2	Class2	Class2	Class2	Class2
95	x96	0.493796	Class1	Class1	Class2	Class2	Class2	Class2
96	x97	0.522733	Class2	Class2	Class2	Class2	Class2	Class2
97	x98	0.427541	Class1	Class1	Class1	Class1	Class1	Class1
98	x99	0.025419	Class1	Class1	Class1	Class1	Class1	Class1
99	x100	0.107891	Class1	Class1	Class1	Class1	Class1	Class1

	Label_k30
50	Class2
51	Class2
52	Class2
53	Class2

54	Class2
55	Class2
56	Class1
57	Class1
58	Class1
59	Class1
60	Class1
61	Class1
62	Class2
63	Class1
64	Class1
65	Class2
66	Class1
67	Class2
68	Class1
69	Class2
70	Class2
71	Class1
72	Class1
73	Class2
74	Class2
75	Class2
76	Class2
77	Class1
78	Class1
79	Class1
80	Class2
81	Class2
82	Class1
83	Class1
84	Class1
85	Class1
86	Class2
87	Class2
88	Class2
89	Class1
90	Class1
91	Class2
92	Class2
93	Class2
94	Class2
95	Class1
96	Class2
97	Class1
98	Class1
99	Class1

[43]:
```python
# Display accuracies
print("\nAccuracies for different k values:")
for k, acc in accuracies.items():
    print(f"k={k}: {acc:.2f}%")
```

```
Accuracies for different k values:
k=1: 100.00%
k=2: 100.00%
k=3: 98.00%
k=4: 98.00%
k=5: 98.00%
k=20: 98.00%
k=30: 100.00%
```

[ ]:
```python

```