ASSIGNMENT-1

"Classification Accuracy on Fisher's Iris (flowers) data by KNNC, MkNNC and Decision Tree"

Submitted By

Sourabh Kumar

Stud ID: 15/8

Course Name- CPI 015 Pattern Recognition- Spring 2020

Under the guidance of:

Prof. MN Murthy

Department of Computer Science and Automation

Indian Institute of Science, Bangalore, India

Table of Contents

1. Problem Statement
2. Technology and Programming Resources Used
3. Experiments (1-6) around kNNC and MkNNC individually
4. Results, Plot and Analysis
5. Additional Experiment using Decision Tree classifier experiment and analysis

Problem Statement

1. Download Fisher’s Iris (flowers) data. There are 3 classes (Setosa, Virginica, and Versicolor) and4features(sepallength, sepalwidth,petallength, and petalwidth) describing each flower. There are total of 50 flowers in each class and the class label is also provided for each of the 150 flowers.

2. Run kNNC and MkNNC based on the following: (a) Split the set of 50 patterns from each class into 30 for training and 20 for testing. (b) Consider Setosa and Versicolor classes; this pair of classes will have a total of 60 training points and 40 test points, where each point is a 4-dimensional vector. Match the class label given by each of your classifiers on each of the 40 test patterns with the already provided class label and report the number of correct matches out of 40. Compute average value over 10 random splits of the data into training and test parts. (c) Perform the same on the pair Versicolor and Virginica. (d) Repeat your experiments by varying the value of k from 1 to 10. (e) Use L1, L2, L∞, and L1/2 norms. (f) Use both normalized and unnormalized data. (g) Compute the classification accuracy in each case.

3. Report your results appropriately using tables and graphs for different scenarios.

4. There port must be brief giving a page on the resources used and how they are used. Two-three pages on the results of your experiments.

Technology and Programming Resources Used

- Spyder Programming Editor
- Python Programming Language 3.7
- Sklearn python libraries
- Pandas Library
- Matplotlib library
- Fisher's Iris dataset - iris.data

Experiment-1 (kNNC)

Program Name- Assignment1_SourabhKumar_Satosa_Versicolor_KNN.py

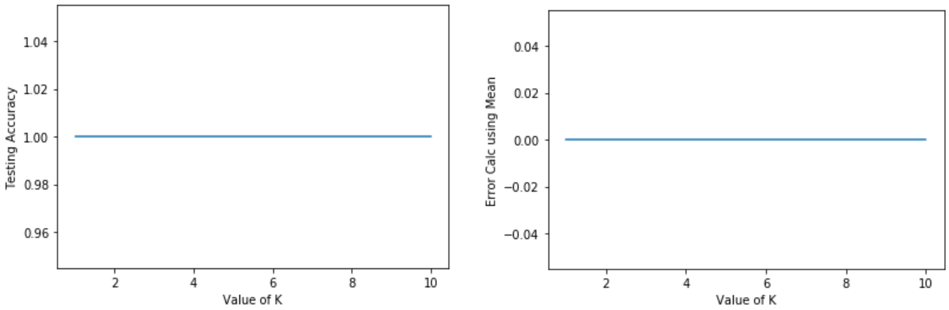
Variables and Inputs:

- Data set Contains - Satosa and Versicolor types
- Dataset normalized before split – yes and no
- Dataset split ratio for training and testing – 60 : 40 (random)
- Number of features used : 2, 3 and 4
- Distance functions – L1, L2, Lmax

Result Table:

Distance Function	Data Normalization	No. of Features	Max Accuracy rate	Min Error rate	K
L1	Yes	2	100	0	1
L1	Yes	3	100	0	1
L1	Yes	4	100	0	1
L1	No	2	100	0	1
L1	No	3	100	0	1
L1	No	4	100	0	1
L2	Yes	2	100	0	1
L2	Yes	3	100	0	1
L2	Yes	4	100	0	1
L2	No	2	100	0	1
L2	No	3	100	0	1
L2	No	4	100	0	1
Lmax	Yes	2	100	0	3
Lmax	Yes	3	100	0	1
Lmax	Yes	4	100	0	1
Lmax	No	2	100	0	1
Lmax	No	3	100	0	1
Lmax	No	4	100	0	1

Plots:



Analysis:

- Accuracy rate is 100% with K=1 for all inputs and variances.

Experiment-2 (MkNNC)

Program Name- Assignment1_SourabhKumar_Satosa_Versicolor_MKNN.py

Variables and Inputs:

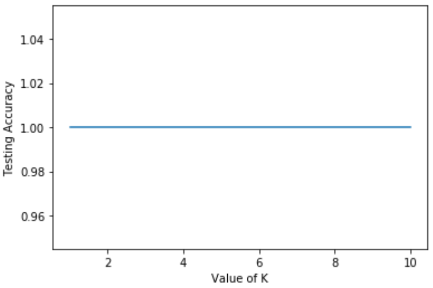
- Data set Contains - Satosa and Versicolor types
- Dataset normalized before split – yes and no
- Dataset split ratio for training and testing – 60 : 40 (random)
- Number of features used : 2, 3 and 4
- Distance functions – L1, L2, Lmax

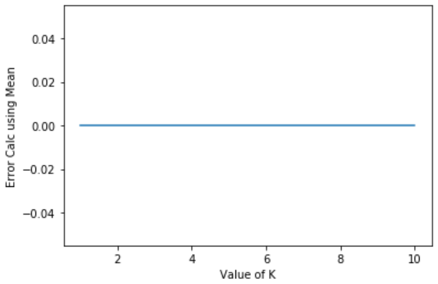
Result Table:

Distance Function	Data Normalization	No. of Features	Max Accuracy rate	Min Error rate	K
L1	Yes	2	100	0	1
L1	Yes	3	100	0	1
L1	Yes	4	100	0	1
L1	No	2	100	0	1
L1	No	3	100	0	1
L1	No	4	100	0	1
L2	Yes	2	100	0	1
L2	Yes	3	100	0	1
L2	Yes	4	100	0	1
L2	No	2	100	0	1
L2	No	3	100	0	1

L2	No	4	100	0	1
Lmax	Yes	2	100	0	1
Lmax	Yes	3	100	0	1
Lmax	Yes	4	100	0	1
Lmax	No	2	100	0	1
Lmax	No	3	100	0	1
Lmax	No	4	100	0	1

Plots:





Analysis:

- Accuracy rate is 100% with K=1 for all inputs and variances.

Experiment-3 (kNNC)

Program Name- Assignment1_SourabhKumar_Versicolor_Verginica_KNN.py

Variables and Inputs:

- Data set Contains - Versicolor and Verginica types
- Dataset normalized before split – yes and no
- Dataset split ratio for training and testing – 60 : 40 (random)
- Number of features used : 2, 3 and 4
- Distance functions – L1, L2, Lmax

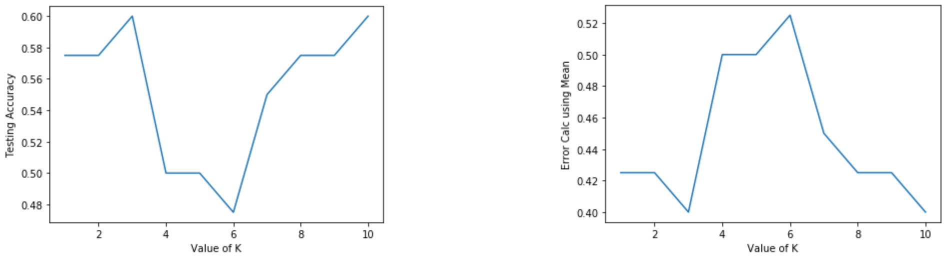
Result Table:

Distance Function	Data Normalization	No. of Features	Max Accuracy rate	Min Error rate	K
L1	Yes	2	62.5	37.5	3
L1	Yes	3	87.5	12.5	2
L1	Yes	4	92.5	7.5	2
L1	No	2	60	40	1
L1	No	3	90	10	5
L1	No	4	92.5	7.5	1
L2	Yes	2	65	35	3
L2	Yes	3	82.5	17.5	1

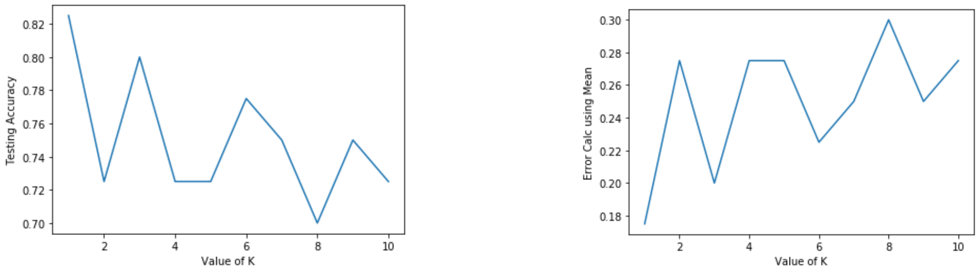
L2	Yes	4	92.5	7.5	1
L2	No	2	62.5	37.5	1
L2	No	3	90	10	3
L2	No	4	92.5	7.5	1
Lmax	Yes	2	60	40	3
Lmax	Yes	3	82.5	17.5	1
Lmax	Yes	4	95	5	1
Lmax	No	2	62.5	37.5	1
Lmax	No	3	85	15	1
Lmax	No	4	90	10	1

Plots:

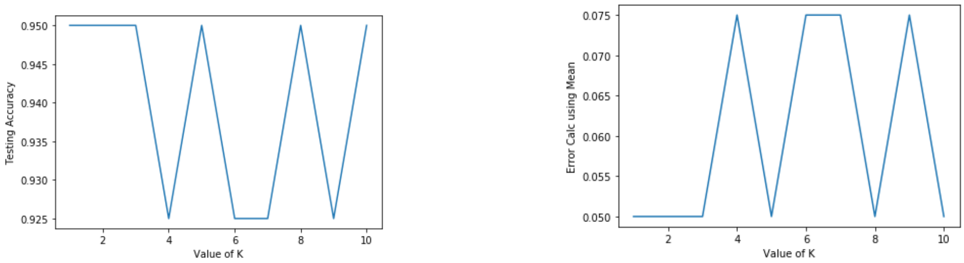
~ 60% accuracy with 2 features:



80-90% accuracy with 3 features:



92.5% accuracy with 4 features:



Analysis:

- If only 2 features are used then accuracy rate is low, ~ 60%
- If only 3 features are used then accuracy is between 80% and 90%

- If all 4 features are used then accuracy is above 92.5%

Experiment-4 (MkNNC)

Program Name- Assignment1_SourabhKumar_Versicolor_Verginica_MKNN.py

Variables and Inputs:

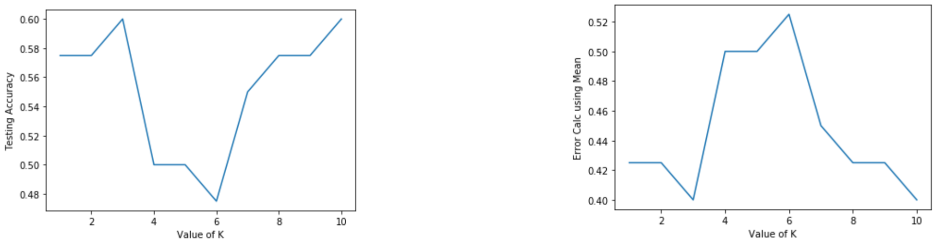
- Data set Contains - Versicolor and Verginica types
- Dataset normalized before split – yes and no
- Dataset split ratio for training and testing – 60 : 40 (random)
- Number of features used : 2, 3 and 4
- Distance functions – L1, L2, Lmax

Result Table:

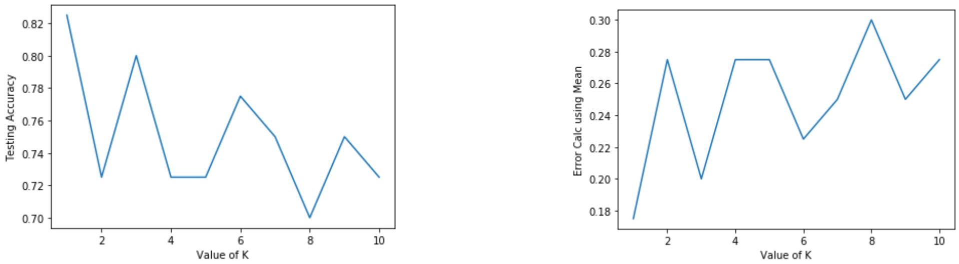
Distance Function	Data Normalization	No. of Features	Max Accuracy rate	Min Error rate	K
L1	Yes	2	65	35	3
L1	Yes	3	85	15	1
L1	Yes	4	90	10	1
L1	No	2	65	35	7
L1	No	3	87.5	12.5	2
L1	No	4	92.5	7.5	1
L2	Yes	2	65	35	3
L2	Yes	3	85	15	3
L2	Yes	4	92.5	7.5	1
L2	No	2	65	35	3
L2	No	3	90	10	3
L2	No	4	92.5	7.5	1
Lmax	Yes	2	60	40	3
Lmax	Yes	3	82.5	17.5	1
Lmax	Yes	4	95	5	1
Lmax	No	2	62.5	37.5	1
Lmax	No	3	85	15	1
Lmax	No	4	92.5	7.5	2

Plots:

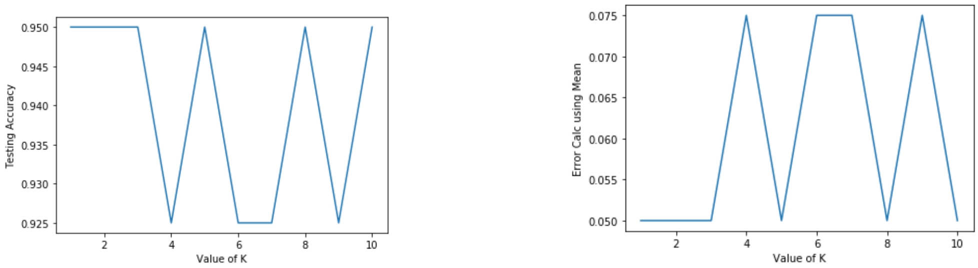
~ 60% accuracy with 2 features:



80-90% accuracy with 3 features:



92.5% accuracy with 4 features:



Analysis:

- If only 2 features are used then accuracy rate is low, ~ 60%
- If only 3 features are used then accuracy is between 80% and 90%
- If all 4 features are used then accuracy is above 92.5%

Experiment-5 (kNNC)

Program Name- Assignment1_SourabhKumar_IRIS_KNN.py

Variables and Inputs:

- Data set Contains – Setosa, Versicolor and Verginica types
- Dataset normalized before split – yes and no
- Dataset split ratio for training and testing – 60 : 40 (non random)
- Number of features used : 2, 3 and 4
- Distance functions – L1, L2, Lmax

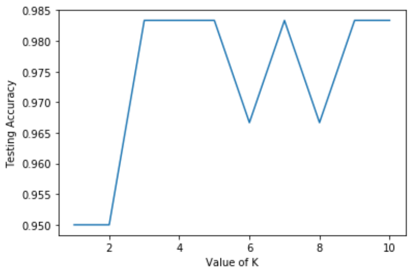
Result Table:

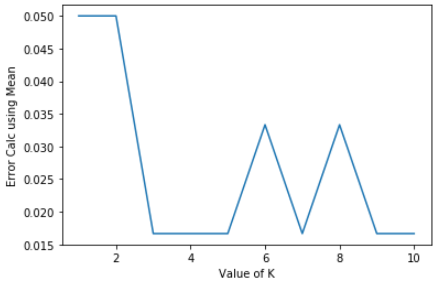
Distance Function	Data Normalization	No. of Features	Max Accuracy rate	Min Error rate	K
L1	Yes	2	80	20	5

L1	Yes	3	93.33333333	6.666666667	1
L1	Yes	4	95	5	2
L1	No	2	80	20	1
L1	No	3	98.33333333	1.666666667	6
L1	No	4	98.33333333	1.666666667	6
L2	Yes	2	80	20	3
L2	Yes	3	93.33333333	6.666666667	1
L2	Yes	4	95	5	1
L2	No	2	81.66666667	18.33333333	5
L2	No	3	98.33333333	1.666666667	7
L2	No	4	98.33333333	1.666666667	6
Lmax	Yes	2	81.66666667	18.33333333	3
Lmax	Yes	3	95	5	1
Lmax	Yes	4	93.33333333	6.666666667	1
Lmax	No	2	81.66666667	18.33333333	1
Lmax	No	3	98.33333333	1.666666667	6
Lmax	No	4	98.33333333	1.666666667	3

Plots:

Highest Accuracy- 98.33%





Analysis:

- non-normalized data is better than normalized in case of IRIS, accuracy or error rate is not very different
- 3 looks the most optimal number of features to classify the test points accurately
- Distance function is not playing very significant role
- Highest Accuracy = 98.33%
- Minimum error rate in prediction = 1.67%
- Euclidean distance, K = 3 looks the most optimal nearest neighbour

Experiment-6 (MkNNC)

Program Name- Assignment1_SourabhKumar_IRIS_MKNN.py

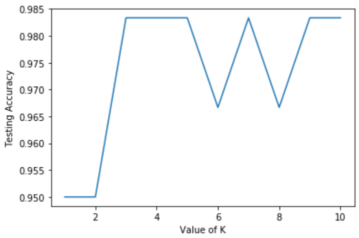
Variables and Inputs:

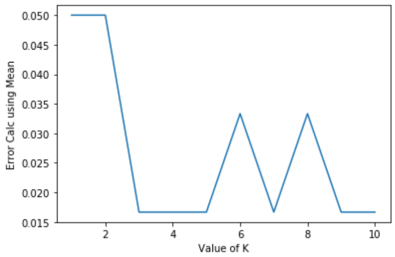
- Data set Contains – Setosa, Versicolor and Verginica types
- Dataset normalized before split – yes and no
- Dataset split ratio for training and testing – 60 : 40 (non random)
- Number of features used : 2, 3 and 4
- Distance functions – L1, L2, Lmax
- Non random split

Result Table:

Distance Function	No. of Features	**Max Accuracy rate**	**Min Error rate**	K
L1	2	81.66666667	18.33333333	8
L1	3	98.33333333	1.666666667	8
L1	4	98.33333333	1.666666667	5
L2	2	81.66666667	18.33333333	7
L2	3	98.33333333	1.666666667	7
L2	4	98.33333333	1.666666667	6
Lmax	2	81.66666667	18.33333333	1
Lmax	3	96.66666667	3.333333333	1
Lmax	4	98.33333333	1.666666667	3

Plots:





Analysis:

- Lmax seems to be performing better than other two distance functions
- Using all 4 features look optimum
- highest accuracy rate = 98.33%
- K = 3 looks optimum in mknn
- Normalization is not used

Decision Tree Classifier

Program Name- Assignment1_SourabhKumar_IRIS_DecTree.py

Input Parameters	**Accuracy Rate**

criterion="entropy"/"gini",
splitter="best",
max_depth=2/3/any,
data normalized = yes/no
	96.7%

criterion="entropy"/"gini",
splitter="random",
max_depth=2/3/any,
data normalized = yes/no
	Varying 60% to ~99%

Analysis:

- With Best Splitter used, Accuracy is 96.7 %
- With random splitter technique accuracy is varying in every execution from 60% to ~99% with mean as 96.7% as above.

It is monitored below the frequency of accuracies over few experiment iterations with random split.

Accuracy	Frequencies
95%	3
96.7%	4
93.33%	3
98.33%	1
60%	1
82%	2