COMP 6915 - Machine Learning
Assignment 1 Report

by
© Soroush Baghernezhad
& Ramyar Zarza
& Mantra .

Instructor: Karteek Popuri
Department of Computer Science & Scientific Computing
Memorial University of Newfoundland

January 2025

St. John's Newfoundland

Question 1:

In part we trained a KNN model for different values of $k = \{1, 3, 5, 10, 20, 30, 50, 100, 150, 200\}$. As it can be seen in Figure 1, error values vary from $k = 1$ to $k = 200$, where $k = 30$ has the lowest test error of 16%.
Based on Figure 2, we can draw the following interpretations:

- $k = 1$: The training error rate is 0.0, while the test error rate is 0.22. This indicates perfect training accuracy but poor generalization due to high variance. The model memorizes the training data but fails to generalize for unseen samples.

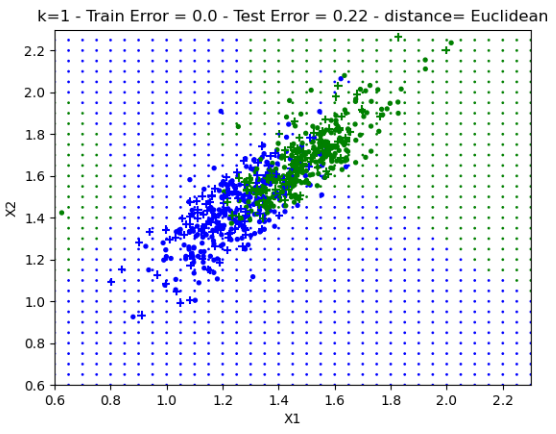
- $k = 3$ to $k = 10$: The test error decreases, suggesting better generalization as k increases. The model balances bias and variance.

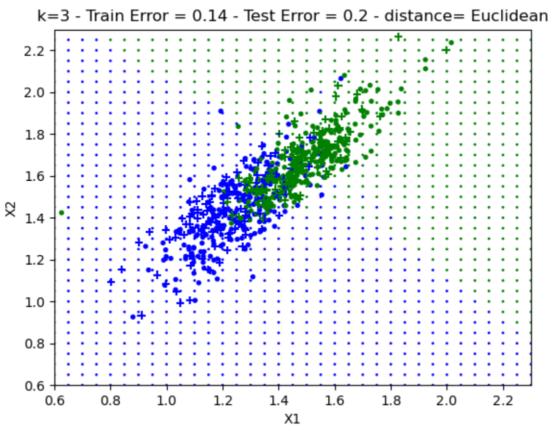
- $k = 20$ to $k = 30$: Test error stabilizes around 16-17%, and training accuracy remains high, which shows that the model is robust and less sensitive to noise.

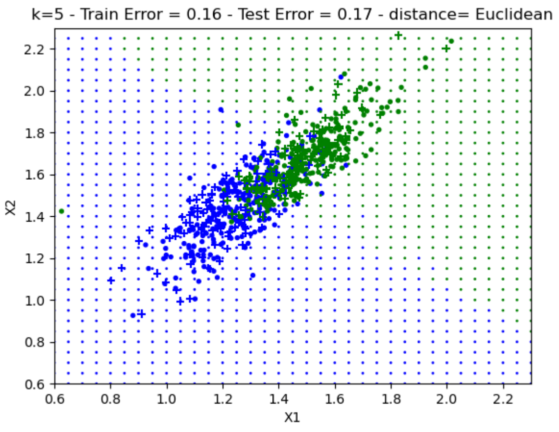
- $k = 100$ and beyond: Both test and training errors increase, indicating that the model is becoming overly simplified. It struggles to capture the finer details of the decision boundary, leading to higher bias.

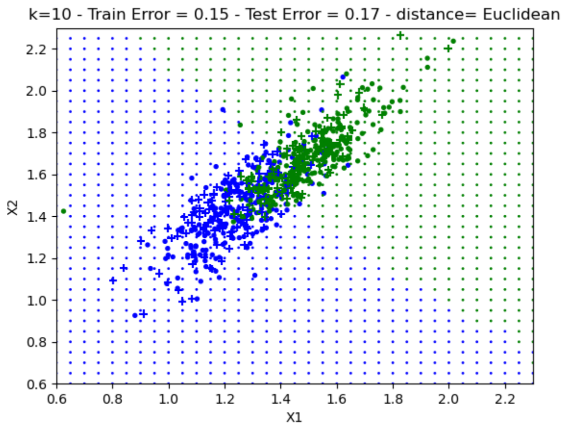
- At low k (e.g., $k = 1$), the model overfits the training data which leads to low training error but high test error due to its inability to generalize.

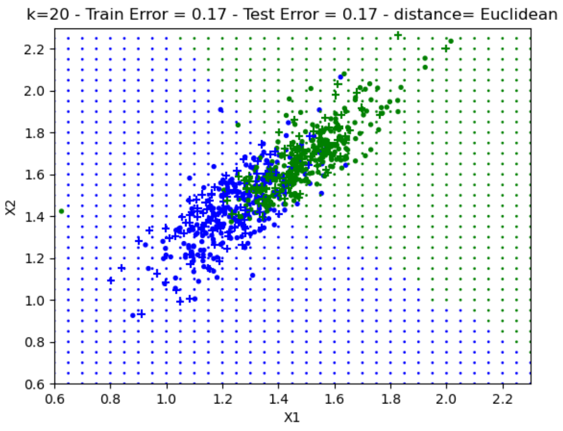
- At high k (e.g., $k = 200$), the model underfits both the training and test data which results in high error rates for both. The decision boundary becomes too smooth, failing to capture the underlying patterns.

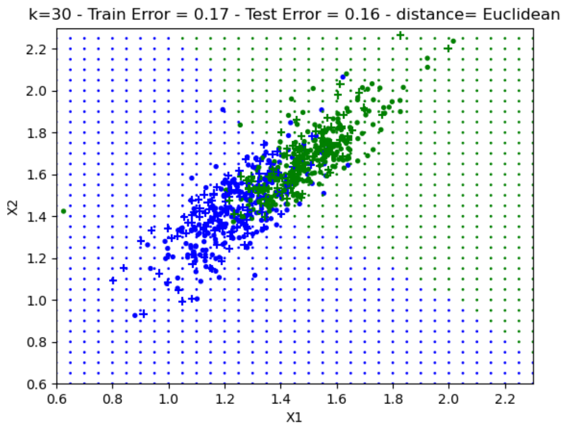


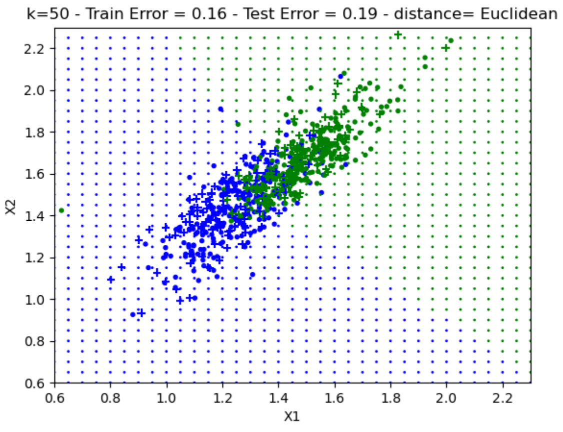


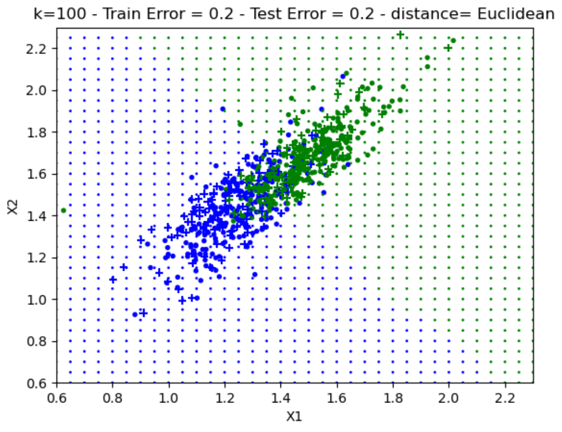

k=5 - Train Error = 0.16 - Test Error = 0.17 - distance= Euclidean
X1
X2

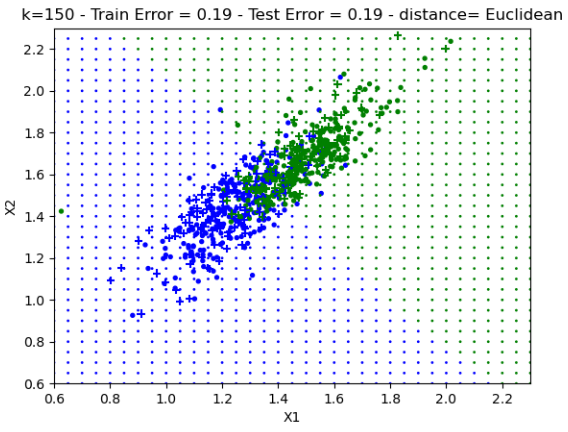

k=10 - Train Error = 0.15 - Test Error = 0.17 - distance= Euclidean
X1
X2


k=20 - Train Error = 0.17 - Test Error = 0.17 - distance= Euclidean
X1
X2


k=30 - Train Error = 0.17 - Test Error = 0.16 - distance= Euclidean
X1
X2


k=50 - Train Error = 0.16 - Test Error = 0.19 - distance= Euclidean
X1
X2


k=100 - Train Error = 0.2 - Test Error = 0.2 - distance= Euclidean
X1
X2



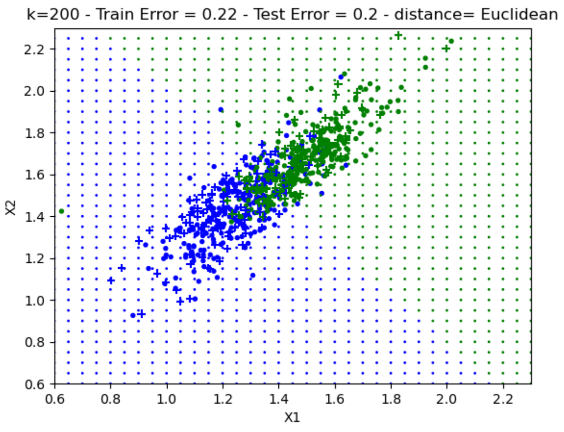


Figure 1: Plotting train and test data on 2D grid, train data is depicted by solid dots and test data is positive marker. Color green indicate negative class and color blue is the positive class.

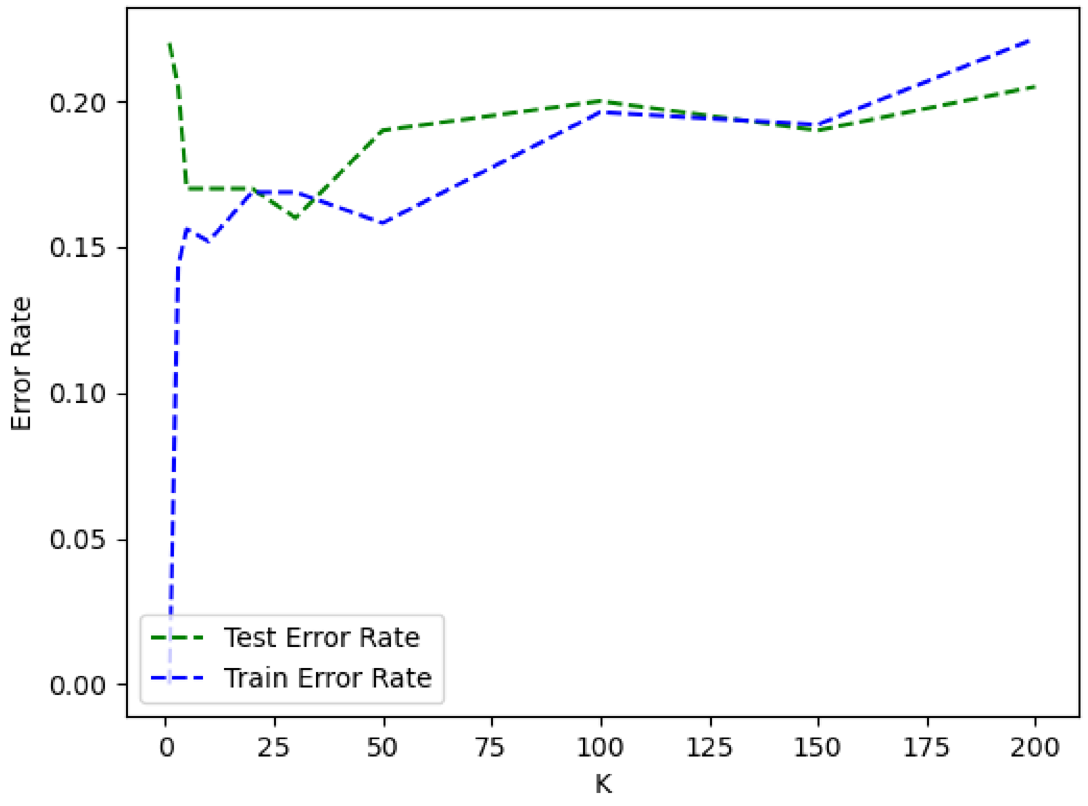


Figure 2: test and train error rate over different values of k.

Question 2:

Replacing Euclidean distance with Manhattan distance and using the best k from the previous section (k=30) yields the result shown in Figure 3, where the test error is 0.17, which is higher than the values from Question 1. For this dataset, Euclidean distance is a better choice than Manhattan distance, as it aligns better with the data's inherent structure and yields a higher test accuracy of 84%.

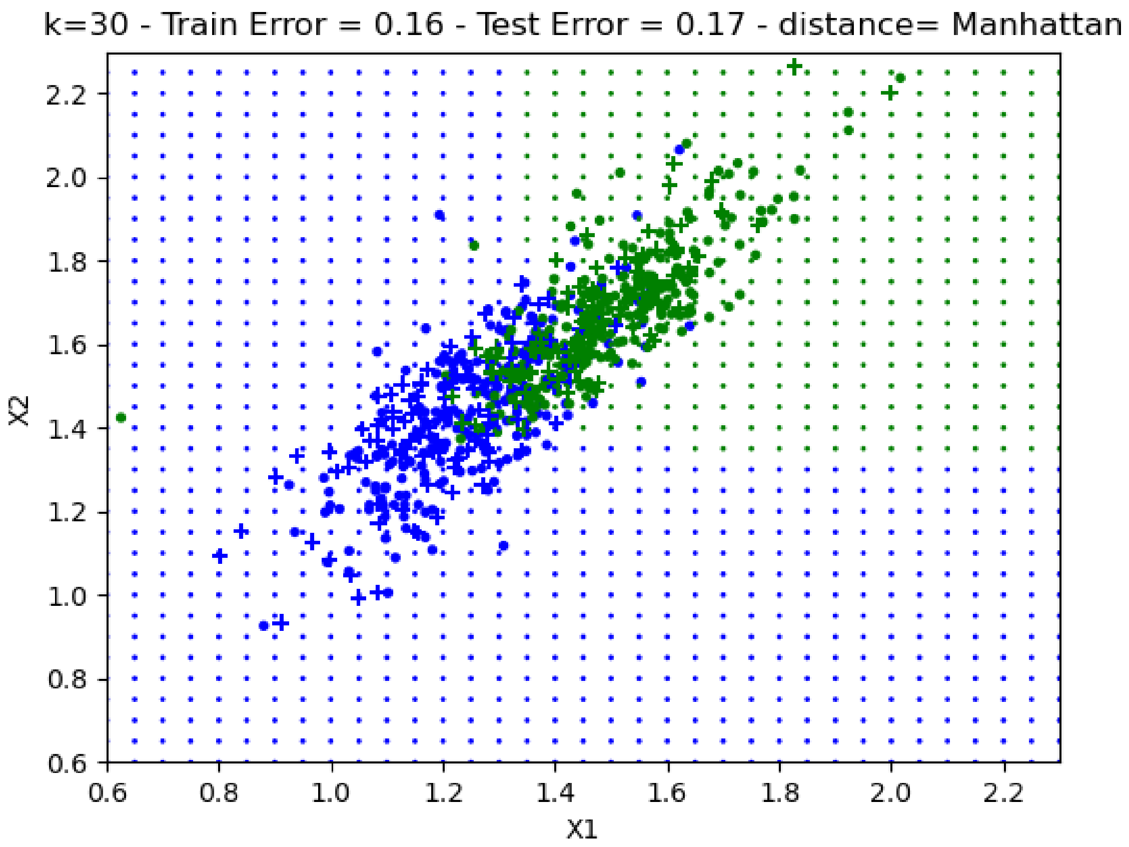


Figure 3: 2D plot of test and train data and the boundaries for Question 2.

Question 3:

From the experiments in Question 1 and Question 2, Euclidean distance was selected because it achieved the lowest test error rate of 0.16%. Based on Figure 4, we can categorize our model regarding capacity and generalization into three distinct zones:

- **Underfitting** (high k, low $1/k$): Both training and test error rates are high due to high bias, as the model is too simple to capture the data's complexity.
- **Optimal fit** (moderate k): Test error is minimized, as it's achieving a balance between bias and variance. The model generalizes well to unseen data.
- **Overfitting** (low k, high $1/k$): Training error approaches zero, but test error increases due to high variance, as the model captures noise in the data.

is it possible to plot Bayes Classifier error using only the information you have?

$$P_{\text{error}} = \int_{R^d} \min(P(Y = 0|X = x), P(Y = 1|X = x))p(x)\,dx \tag{1}$$

where:

- $P(Y|X)$ is the **posterior probability** of class Y given feature vector X.
- $p(x)$ is the **marginal probability density** of X.

Based on Eq 1, in order to get the true Bayes classifier error, we need to know the actual distribution of our data which is unknown and its our objective, hence, we cannot find the bayes classifier error since the posterior probability is unknown to us, however, we can estimate this error but this estimation is not reliable and it is not setting any lower bound for our classification errors.

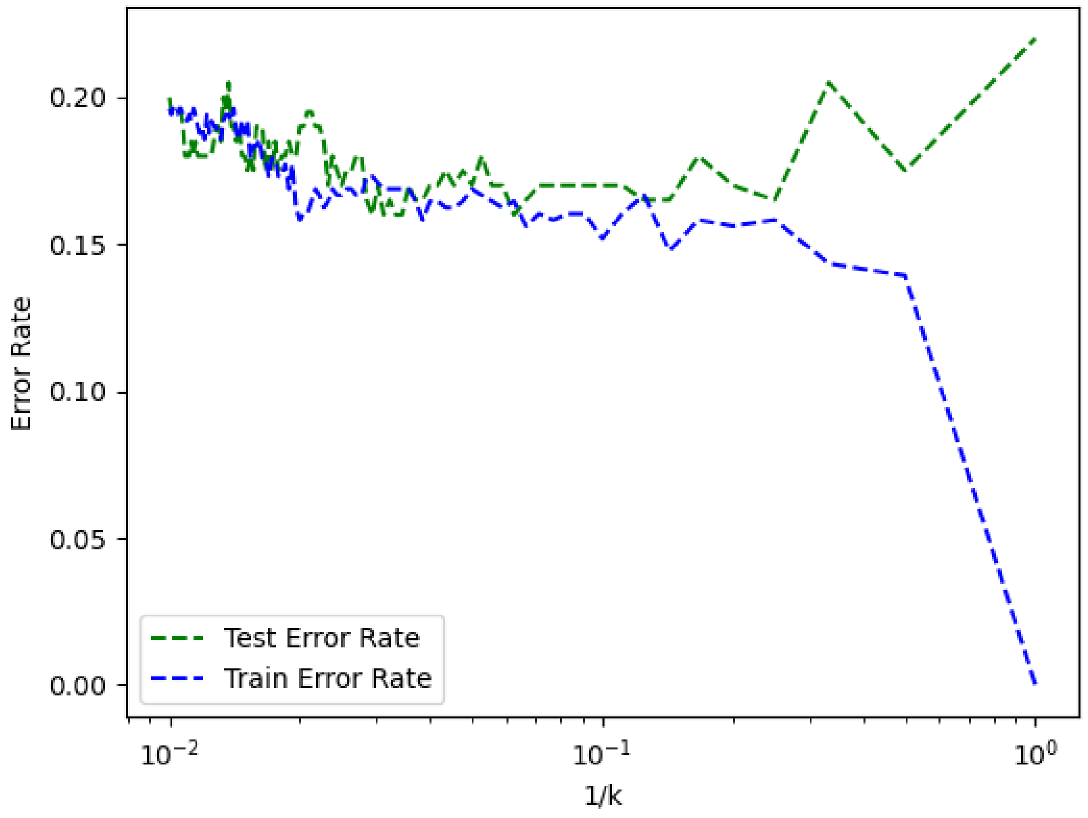


Figure 4: A comparison of model capacity in x axis and error rate for Question 3.

Question 4:

We experimented with two different distance metrics: Manhattan and Euclidean. Additionally, we applied data denoising using DBSCAN and enhanced our dataset through data augmentation by adding an extra feature. For data normalization, we tested MinMaxScaler, StandardScaler, and RobustScaler. We also explored feature weighting based on distance. To identify the best model, we evaluated all possible combinations of these techniques across different values of k (ranging from 1 to 100).Our final model achieved an accuracy of 85% when evaluated on the test data. Achieving significantly higher accuracy may not be feasible given the current dataset, as we likely lack sufficient information.