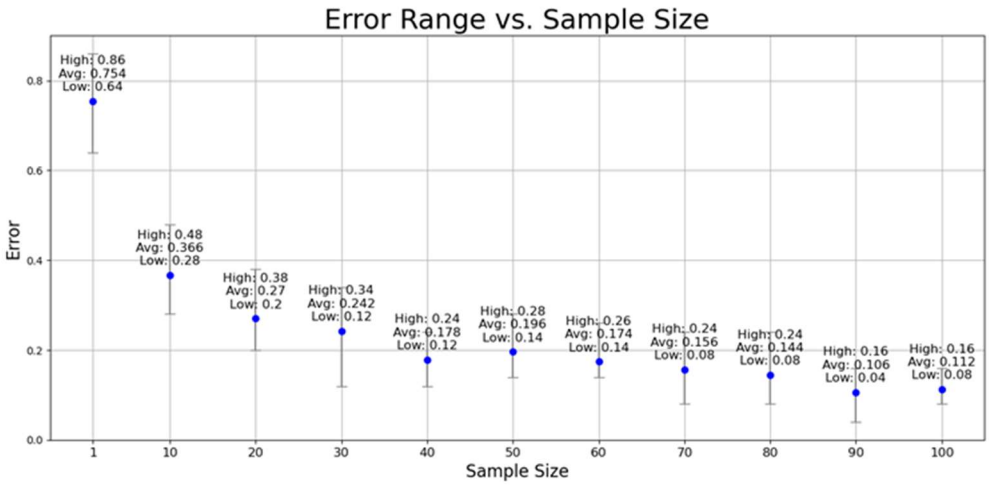
Intro To ML – KNN

Amit Zulan – 207299033, Dvir Ben Zikri - 315409508

1. Code is added separately.
2.
   a. First let us test our KNN with different sample sizes, we tested for sample size equal to 1 and then $10,20,30,\ldots,100$, we got:



   b. We can see that as we increase the sample size along a constant k=1 – 1NN we get smaller error.

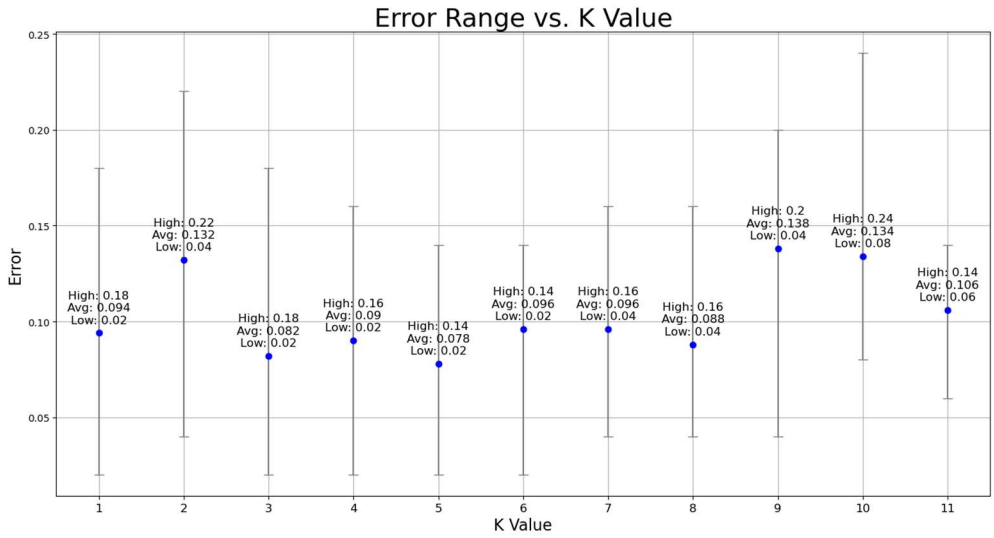
      This trend makes sense because as we increase the sample size, test samples have closer neighbors, increasing the probability that they have the same label – thus increasing the probability of getting a correct prediction.

   c. The results differ from run to run on the same sample size, since we used a random sample generator from a large dataset. This results in different, sample & test combinations (each run having different label distribution), causing variance in the results.

   d. We can see that the error bars get smaller as we increase the sample size.
      This trend can be explained by the fact that larger samples size results in closer neighbor for each test point, thus reducing the variance of the errors.

Moreover, generally smaller sample sizes have a higher probability of missing a label within the sample. Therefore, a certain sample might not predict a specific label at all.

e. Now let us test our KNN with different k values, we check for k equal to 1 up to 11, we got:



It seems like there is no trend in the graph above.
It seems like the optimal value for k is either 3 or 5 but that varies between runs.
Since k is small relatively to sample size, it's unlikely we need to take many neighbors, thus including far away neighbors (with irrelevant labels).
In addition, different digits (represented in a 2-D array, gray scale) should be far away from each other, meaning that we probably don't need to consider many neighbors.

Lastly, we can see that 2NN performs poorly (having a high average error and variance between different runs).
This can be explained by the fact that some test points have its two closest neighbors, with different labels thus resulting in a 50% chance to get the wrong label (assuming one of the neighbors has the correct label).

3.

a. If $(x_1, y_1), (x_2, y_2) \in S$ therefore, $(x_1, y_1), (x_2, y_2) \in D$.
By definition if D is C-Lipschitz w.r.t. Euclidian distance then $|\eta(x_1) - \eta(x_2)| \leq c \cdot \big||x_1 - x_2|\big|$. Since both $(x_1, y_1), (x_2, y_2)$ are in the sample which is sampled from D, and D has 0 bayes error meaning it's deterministic, then we know $P_{(X,Y)\sim D}[Y = y_1 | X = x_1] = 1$ and the same goes for (x_2, y_2). Therefore, we can deduce $\eta(x_1) = y_1, \eta(x_2) = y_2$, and $|\eta(x_1) - \eta(x_2)| = 1$ since $y_1 \neq y_2$.
Combining everything together we get:
$$|\eta(x_1) - \eta(x_2)| = 1 \leq c \cdot \big||x_1 - x_2|\big| \xrightarrow{yields} \big||x_1 - x_2|\big| \geq \frac{1}{c}$$

b. We would like to calculate the $err(f_S^{nn}, D) = 0$. Meaning, for some x_i in D what is the probability $P[f(x_i) \neq y_i]$. We know that $\forall x_i \in D, \exists B_i \; s.t. \; x_i \in B_i$. In addition, $\forall B_i \; \exists x_j \in S, s.t \; x_j \in B_i$. From that we can deduce that $\forall x_i \in D, \exists B_i \; s.t. \; x_i, x_j \in B_i, x_j \in S$.
So, both the random prediction point x_i and the sample point x_j are in some ball B_i. Since they're both in the same ball with radius $\frac{1}{3c}$ the maximum distance between x_i, x_j is the diameter of the ball $\frac{2}{3c} < \frac{1}{c}$.
Now we can use the C-Lipschitz property of D, and we know that both x_i, x_j has the same label y_i. The nearest neighbor prediction rule will output $f_S^{nn}(x_i) = y_i$ which we know for a fact that is the true label of x_i due to the C-Lipschitz property. We didn't assume anything about x_i, meaning this is true $\forall x_i \in D$.
Thus $\forall x_i \in D, f_S^{nn}(x)$ predicts the correct label. Another thing to note is that if $\exists x_k$ that is closer to x_i than x_j than they would still have the same label due to the C-Lipschitz property. To sum up, $\forall x_i \in D, f_S^{nn}$ predicts the correct label and therefore $err(f_S^{nn}, D) = 0$.

4.

a. $\chi = \left[x_{age}, x_{weight}\right]^T, x_{age} \in (0,42]\ and\ x_{weight} \in (0,5]$
$Y = y_{color}, y_{color} \in \{black, white\}$

b.

$$h_{bayes} = \{\ black, if\ x = [8,4]\ or\ x = [15,1]$$
$$white, if\ x = [15,2]\}$$

c. $bayes - optimal\ error = \sum_{x \in X} P[X = x] \cdot (1 - \eta_{h(x)}(x)) =$
$0.48_{x=[8,4]} \cdot \left(1 - \frac{0.42}{0.48}\right) + 0.28_{x=[15,1]} \cdot \left(1 - \frac{0.21}{0.28}\right) + 0.24_{x=[15,2]}$
$\cdot \left(1 - \frac{0.24}{0.24}\right) = 0.06 + 0.07 + 0 = 0.13 = 13\%$

d. $H = \{h_{black}(x) = black,\ h_{white}(x) = white\}$
$err(h_{black}, D) = 0.06 + 0.07 + 0.24 = 0.37 = 37\%$
$err(h_{white}, D) = 0.42 + 0.21 = 0.63 = 63\%$
$\rightarrow err_{app} = \min_{h \in H} err(h, D) = 0.37 = 37\%$

e. Generally, the expected memorize error is:

$$E[err_{mem}] = E\left[\frac{|Y| - 1}{|Y|} \cdot \sum_{x \in D \setminus S} P[X = x]\right]$$
$$= \frac{|Y| - 1}{|Y|} \cdot E\left[\sum_{x \in D} P[X = x] \cdot I[x \notin S]\right]$$
$$= \frac{|Y| - 1}{|Y|} \cdot \sum_{x \in D} P[X = x] \cdot E\left[I[x \notin S]\right]$$
$$= \frac{|Y| - 1}{|Y|} \cdot \sum_{x \in D} p_x \cdot (1 - p_x)^m$$

$E[err_{mem}] = 0.5$
$\cdot (0.08 \cdot 0.92^3 + 0.3 \cdot 0.7^3 + 0.47 \cdot 0.53^3 + 0.15$
$\cdot 0.85^3) \cong 0.163643$

The reason we can't use the Memorize method on D and we can use it on D'' is because D'' is a deterministic distribution and D is an indeterministic distribution. Thus, the memorization would not be able to know which indeterministic label to memorize. Even if we decide on a decision rule between different labels, we would always be wrong on the x that has the other label from which we memorized.

5.

a. We can see from the fig. 2 that $f_{ERM}^{rec}(x) = f_{5,5}^{rec}(x)$, since these thresholds would label wrong only 2 apples from the sample S. $\widehat{err}\left(f_{5,5}^{rec}(x), S\right) = \frac{1}{m}\sum_{i=1}^{m} I[h(x_i) \neq y_i] = \frac{2}{20} = 0.1 = 10\%$.

b.

- $g(x_h, x_v) = \left|\left|(x_h, x_v)\right|\right|_2 = \sqrt{x_h^2 + x_v^2}$, the Euclidean distance formula from $(0, 0)$.
- $H_g = \left\{I\left[\left|\left|x\right|\right|_2 \leq \tau \,\middle|\, \tau \in R\right]\right\}$
- $f_{euc,6}(x) = I\left[\left|\left|x\right|\right| \leq 6\right]$ will be the ERM prediction rule selected from the above H_g. It achieves 0 empirical error on the sample S, since all blue samples are closer than 6 units from $(0, 0)$, and all red sample are further away. This is not the only function that achieves 0 empirical error on sample S. There is a range of thresholds that can achieve this.

Moreover, there is an extended hypothesis class that can also achieve 0 empirical error and might have a better prediction rule than the suggested H_g from above.

Choosing $g(x_h, x_v) = \sqrt{\alpha \cdot x_h^2 + \beta \cdot x_v^2} \mid \alpha, \beta \in R$. This means there are more prediction rules to choose from but could reduce the approximation error. Real life reasons for this could be that the trunk is thicker or healthier in some axis.