CSE-575 - HW2

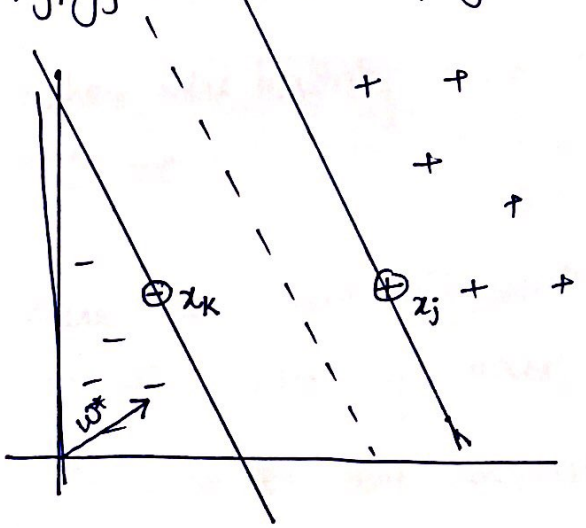
1) Support Vector Machines ↧

1.1) Hard-margin Linear SVM ↧

Given,

$$\underset{w,b}{\text{argmax}}\ d = d^+ - d^- = \frac{w^T x_j}{\|w\|^2} - \frac{w^T x_k}{\|w\|^2}$$

Now given that there are only two support vectors i.e. $x_j, y_j = 1$ & $x_k, y_k = -1$ with parameters w^* & b^*.



from above, we know that ↧

$$\vec{w^*} \cdot \vec{x_j} + b^* \geq 1 \quad \text{—— ①}$$

$$\vec{w^*} \cdot \vec{x_k} + b^* \leq -1 \quad \text{—— ②}$$

Introducing a new variable y_i such that

y_i = +1 for +ve samples
−1 for −ve samples.

Multiplying y_i with ① & ②

from ①, $y_i(\vec{w^*} \cdot x_j + b^*) \geq 1$ such that $y_i = +1$ —— ③

from ②, $y_i(\vec{w^*} \cdot x_j + b^*) \geq 1$ such that $y_i = -1$ —— ④

from ③ & ④,

$y_i (\vec{x_i} \cdot \vec{w^*} + b^*) - 1 \geq 0$ such that $y_i = \{+1, -1\}$

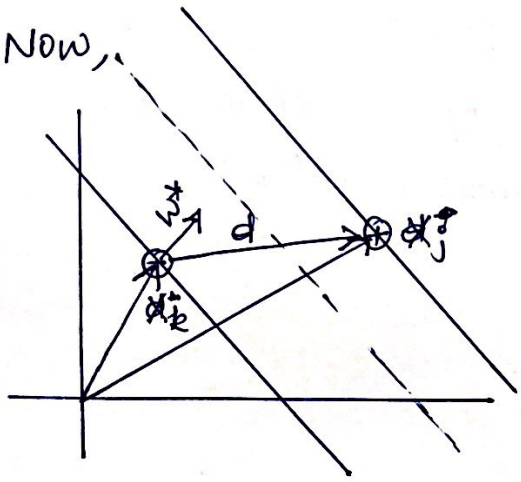
Also, ③ & ④ can be written as,

$1 (\vec{x_j} \cdot w^* + b^*) \geq 1$ —⑤

$-1 (\vec{x_j} \cdot w^* + b^*) \geq 1$ —⑥

} Two constraints

Now,



$d \neq (\vec{x^+} - \vec{x^-}) \cdot \frac{\vec{w^*}}{\|w^*\|}$

$d = (\vec{x_j} - \vec{x_k}) \cdot \frac{\vec{w^*}}{\|w^*\|}$

from ⑤ & ⑥, substituting the values of $\vec{x_j}$ & $\vec{x_k}$

$d = (1 - b^* + 1 + b^*) \cdot \frac{1}{\|w^*\|}$

$\underset{w}{\arg\max} \frac{2}{\|w^*\|} \Rightarrow \min \|w^*\| \Rightarrow \min \frac{1}{2} \|w^*\|^2$

1.2) Soft-Margin SVM ↓

a) In soft-margin SVM, we penalize the incorrectly classified examples by introducing a slack variable i.e. ξ_i

Now, we will minimize the following ↓

$$\min_{w,b,\xi_i \geq 0} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

$\sum_{i=1}^{n}$ → no of mistakes / incorrect classifications

[C → slack penality]

Such that,

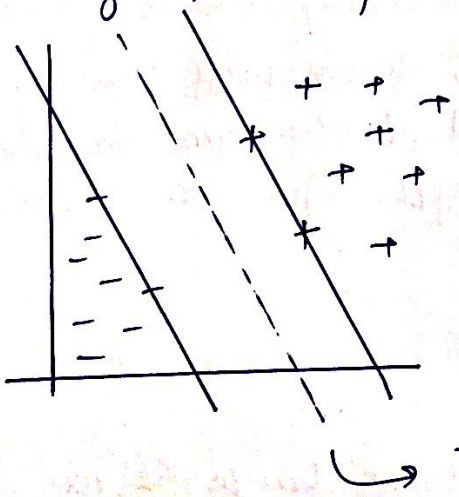
$$\forall i,\ y_i(\vec{w}\cdot\vec{x_i} + b) \geq 1 - \xi_i$$

when the training example is correctly classified then

$\xi_i = 0$

b) there are three different types of examples possible based on the value of slack variable i.e.

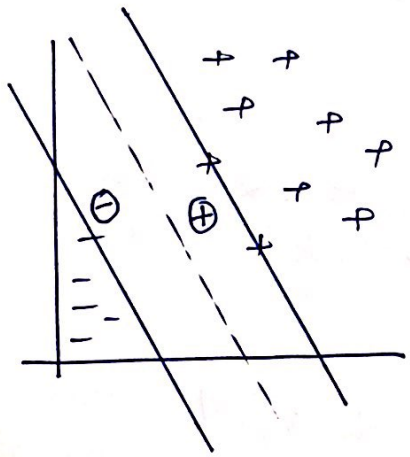
1) All examples are correctly classified, In this case, ξ_i will be zero, i.e. penalty will be zero.



↳ $\xi_i = 0$

Decision boundry will change only if we remove the only training examples that ~~are~~ is on the decision boundry or if that is a support vector.

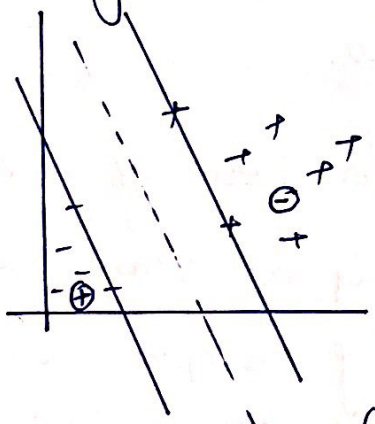
2) the second example will be, it is correctly classified but the confidence is low, so in this case the value of ξ_i will be positive but less than 1 i.e. $0 < \xi_i < 1$



there the two datapoints near to the decision boundry are correctly classified but confidence is low, hence they will be penalized, means ξ_i value will be b/w 0 & 1.

Removing these two datapoints will not change the decision boundry/margin.

3) the third example will be, when few examples are wrongly classified i.e. $\xi_i > 1$, in this case we will heavily penalize such examples and try to maximize our margin.



there two datapoints are wrongly classified, so we will heavily penalize them and at the same time maximise the margin b/w two classes.

$\hookrightarrow (wx+b)y_i \geq 1 - \xi_i$

Removing such example will not ~~have~~ change the margin/decision surface, but removing such example will make the penality zero and ~~it~~ data will be linearly classifiable.

2) Adaboost

2.1) Given, Accuracy = 50%

i.e. $\epsilon_t = 0.5$

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

$$= \frac{1}{2}\ln\left(\frac{0.5}{0.5}\right)$$

$$= \frac{1}{2}\ln(1) = 0$$

Since $\alpha_t = 0$, it means it will not have any say in the final classifier and also, the weight won't change in the next iteration, hence we will drop this weak classifier and go with some other classifier.

2.2) We are given,

$$\epsilon_t = 1 - 0.45 = 0.55$$

here, we can flip the classifier and can get the accuracy equal to 0.55 which is better than the current ϵ_t.

i.e. if now we have,

$$C(x) = \begin{cases} +1 & x > 0 \\ -1 & x \le 0 \end{cases} \qquad \epsilon_t = 0.55$$

than after flipping, we will get,

$$C(x) = \begin{cases} +1 & x \le 0 \\ -1 & x > 0 \end{cases} \qquad \epsilon_t = 0.45$$

i.e 55% accuracy

Given,

x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1

$$C(x) = \begin{cases} +1 & x < \theta \\ -1 & x \geq \theta \end{cases}$$

2.3)

$\theta = 2.5$

Initially we have,

$$D_1(0) = D_1(1) = D_1(2) = D_1(3) = D_1(4) = D_1(5) = D_1(6) = D_1(7) = D_1(8) = D_1(9) = \frac{1}{10}$$

for Deriving the weights D_2,

$$D_2(i) = \frac{D_1(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

$$\epsilon_t = \frac{1}{\sum_{i=1}^{m} D_t(i)} \sum_{i=1}^{m} D_t(i)\, \delta(h_t(x_i) \neq y_i)$$

$$= 0+0+0+0+0+0+\frac{1}{10}\times 1 + \frac{1}{10}\times 1 + \frac{1}{10}\times 1 + 0$$

$$= \frac{3}{10}$$

$$\alpha_t = \frac{1}{2} \ln\left(\frac{7/10}{3/10}\right) = \frac{1}{2}\ln(7/3) = \frac{0.847}{2} = 0.423$$

$$D_2(i) = \frac{D_1(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

$$Z_t = \sum_{i=1}^{10} D_t(i)\exp(-\alpha_t y_i h_t(x_i))$$

$$D_2(0) = \frac{1}{10} \times \frac{\exp(-0.423)}{Z_t} = \frac{1}{10} \times \frac{0.655}{Z_t} = 0.0716$$

$$D_2(1) = \frac{1}{10} \times \frac{\exp(-0.423)}{Z_t} = \frac{1}{10} \times \frac{0.655}{Z_t} = 0.0716$$

$$D_2(3) = \frac{1}{10} \times \frac{\exp(-0.423)}{Z_t} = \frac{1}{10} \times \frac{0.655}{Z_t} = 0.0716$$

$$D_2(4) = \frac{1}{10} \times \frac{\exp(-0.423)}{Z_t} = \frac{1}{10} \times \frac{0.655}{Z_t} = 0.0716$$

$$D_2(5) = \frac{1}{10} \times \frac{\exp(-0.423)}{Z_t} = \frac{1}{10} \times \frac{0.655}{Z_t} = 0.0716$$

$$D_2(6) = \frac{1}{10} \times \frac{\exp(0.423)}{Z_t} = \frac{1}{10} \times \frac{1.52}{Z_t} = 0.166$$

$$D_2(7) = \frac{1}{10} \times \frac{\exp(0.423)}{Z_t} = \frac{1}{10} \times \frac{1.52}{Z_t} = 0.166$$

$$D_2(8) = \frac{1}{10} \times \frac{\exp(0.423)}{Z_t} = \frac{1}{10} \times \frac{1.52}{Z_t} = 0.166$$

$$D_2(9) = \frac{1}{10} \times \frac{\exp(-0.423)}{Z_t} = \frac{1}{10} \times \frac{0.655}{Z_t} = 0.0716$$

$$Z_t = \frac{1}{10} \times 0.655 \times 7 + \frac{1}{10} \times 1.52 \times 3 = 0.4585 + 0.456 = \underline{\underline{0.9145}}$$

2.4)

$$D_3(i) = \frac{D_2(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

$$\epsilon_t = \frac{1}{\sum_{i=1}^{10} D_t(i)} \sum_{i=1}^{10} D_t(i)\,\delta(h_t(x_i) \neq y_i)$$

Now for θ, we have 3 possible values, i.e.

$\theta = 2.5$
or $\theta = 5.5$
or $\theta = 8.5$

Now for each θ value, we will check for the minimum error and select that θ is decision boundry for our current weak classifier.

$\theta = 2.5$

$$C(x) = \begin{cases} +1 & x < \theta \\ -1 & x \geq \theta \end{cases}$$

$\epsilon_t = 0.166 \times 3$
$= 0.498$

$$C(x) = \begin{cases} -1 & x < \theta \\ +1 & x \geq \theta \end{cases}$$

$\epsilon_t = 0.0716 \times 7$
$= 0.502$

$\theta = 5.5$

$$C(x) = \begin{cases} +1 & x < \theta \\ -1 & x \geq \theta \end{cases}$$

$\epsilon_t = 0.0716 \times 3 + 0.166 \times 3$
$= 0.2148 + 0.498$
$= 0.7128$

$$C(x) = \begin{cases} -1 & x < \theta \\ +1 & x \geq \theta \end{cases}$$

$\epsilon_t = 0.0716 \times 4$
$= 0.2864$

$\theta = 8.5$

$$C(x) = \begin{cases} +1 & x < \theta \\ -1 & x \geq \theta \end{cases}$$

$\epsilon_t = 0.0716 \times 3$

$= \boxed{0.214}$ MINIMUM

$$C(x) = \begin{cases} +1 & x < \theta \\ +1 & x \geq \theta \end{cases}$$

$\epsilon_t = 0.0716 \times 4 + 0.166 \times 3$

$= 0.2864 + 0.498$

$= 0.7844$

for all possible 6 error values, we will select the θ & $C(x)$ with minimum error value. i.e

$\theta = 8.5$

$$C(x) = \begin{cases} +1 & x < \theta \\ -1 & x \geq \theta \end{cases}$$

} this is our current weak learner for next iteration.

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - 0.214}{0.214}\right)$$

$$= \frac{1}{2} \ln\left(\frac{0.706}{0.214}\right)$$

$$= \frac{1}{2} \ln(3.67) = \frac{1}{2} \times 1.30 = 0.65$$

Now we will calculate $D_3(0) \ldots D_3(9)$ using above α_t & $C(x)$.

$$D_3(0) = \frac{D_2(0)\exp(-\alpha_t y_0 h_t(0))}{Z_t}$$

$$= \frac{0.0716 \times \exp(-0.65)}{Z_t} = \frac{0.037}{Z_t} = \frac{0.037}{0.817} = 0.045$$

$$D_3(1) = \frac{0.0716 \times \exp(-0.65)}{Z_t} = \frac{0.037}{Z_t} = \frac{0.037}{0.817} = 0.045$$

$$D_3(2) = \frac{0.0716 \times \exp(-0.65)}{Z_t} = \frac{0.037}{Z_t} = \frac{0.037}{0.817} = 0.045$$

$$D_3(3) = \frac{0.0716 \times \exp(0.65)}{Z_t} = \frac{0.137}{Z_t} = \frac{0.137}{0.817} = 0.167$$

$$D_3(4) = \frac{0.0716 \times \exp(0.65)}{Z_t} = \frac{0.137}{Z_t} = \frac{0.137}{0.817} = 0.167$$

$$D_3(5) = \frac{0.0716 \times \exp(0.65)}{Z_t} = \frac{0.137}{Z_t} = \frac{0.137}{0.817} = 0.167$$

$$D_3(6) = \frac{0.166 \times \exp(-0.65)}{Z_t} = \frac{0.086}{Z_t} = \frac{0.086}{0.817} = 0.105$$

$$D_3(7) = \frac{0.166 \times \exp(-0.65)}{Z_t} = \frac{0.086}{Z_t} = \frac{0.086}{0.817} = 0.105$$

$$D_3(8) = \frac{0.166 \times \exp(-0.65)}{Z_t} = \frac{0.086}{Z_t} = \frac{0.086}{0.817} = 0.105$$

$$D_3(9) = \frac{0.0716 \times \exp(-0.65)}{Z_t} = \frac{0.037}{Z_t} = \frac{0.037}{0.817} = 0.045$$

$$Z_t = 0.037 \times 4 + 0.086 \times 3 + 0.137 \times 3$$

$$= 0.148 + 0.258 + 0.411 = 0.817$$

for D_4, again we have 3 possible values for θ i.e. 2.5, 5.5 & 8.5.

Now, we will calculate error at each θ & will choose the θ with min error.

$\theta = 2.5$

$$C(x) = \begin{cases} +1 & x < \theta \\ -1 & x \geq \theta \end{cases}$$

$$\epsilon_t = 0.105 \times 3 = 0.315$$

$$C(x) = \begin{cases} -1 & x < \theta \\ +1 & x \geq \theta \end{cases}$$

$$\epsilon_t = 0.045 \times 4 + 0.167 \times 3 = 0.681$$

$\theta = 5.5$

$$C(x) = \begin{cases} +1 & x < \theta \\ -1 & x \geq \theta \end{cases}$$

$$\epsilon_t = 0.167 \times 3 + 0.105 \times 3$$
$$= 0.504 + 0.315 = 0.819$$

$$C(x) = \begin{cases} -1 & x < \theta \\ +1 & x \geq \theta \end{cases}$$

$$\epsilon_t = 0.045 \times 4$$
$$= \boxed{0.18} \quad \text{MINIMUM}$$

$\theta = 8.5$

$$C(x) = \begin{cases} +1 & x < \theta \\ -1 & x \geq \theta \end{cases}$$

$$\epsilon_t = 0.167 \times 3 = 0.501$$

$$C(x) = \begin{cases} -1 & x < \theta \\ +1 & x \geq \theta \end{cases}$$

$$\epsilon_t = 0.045 \times 4 + 0.105 \times 3$$
$$= 0.18 + 0.315 = 0.49$$

Out of all 6 possible error, we will select the minimum i.e

$\theta = 5.5$

$$C(x) = \begin{cases} -1 & x < \theta \\ +1 & x \geq \theta \end{cases}$$

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-0.18}{0.18}\right) = \frac{1}{2} \times 1.504 = 0.752$$

$$D_4(i) = \frac{D_3(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

$$D_4(0) = \frac{0.045\exp(0.75)}{Z_t} = \frac{0.045\times 2.11}{Z_t} = \frac{0.095}{Z_t} = 0.124$$

$$D_4(1) = \frac{0.045\exp(0.75)}{Z_t} = \frac{0.045\times 2.11}{Z_t} = \frac{0.095}{Z_t} = 0.124$$

$$D_4(2) = \frac{0.045\exp(0.75)}{Z_t} = \frac{0.045\times 2.11}{Z_t} = \frac{0.095}{Z_t} = 0.124$$

$$D_4(3) = \frac{0.167\exp(-0.75)}{Z_t} = \frac{0.167\times 0.472}{Z_t} = \frac{0.078}{Z_t} = 0.102$$

$$D_4(4) = \frac{0.167\exp(-0.75)}{Z_t} = \frac{0.167\times 0.472}{Z_t} = \frac{0.078}{Z_t} = 0.102$$

$$D_4(5) = \frac{0.167\exp(-0.75)}{Z_t} = \frac{0.167\times 0.472}{Z_t} = \frac{0.078}{Z_t} = 0.102$$

$$D_4(6) = \frac{0.105\exp(-0.75)}{Z_t} = \frac{0.015\times 0.472}{Z_t} = \frac{0.049}{Z_t} = 0.064$$

$$D_4(7) = \frac{0.105\exp(-0.75)}{Z_t} = \frac{0.015\times 0.472}{Z_t} = \frac{0.049}{Z_t} = 0.064$$

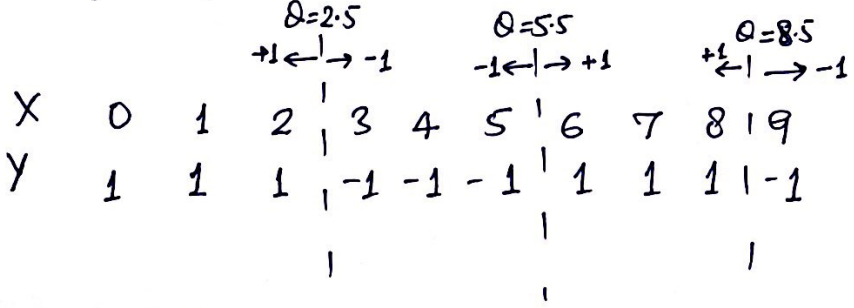
$$D_4(8) = \frac{0.105\exp(-0.75)}{Z_t} = \frac{0.015\times 0.472}{Z_t} = \frac{0.049}{Z_t} = 0.064$$

$$D_4(9) = \frac{0.045\exp(0.75)}{Z_t} = \frac{0.045\times 2.11}{Z_t} = \frac{0.095}{Z_t} = 0.124$$

$$Z_t = 0.095\times 4 + 0.078\times 3 + 0.049\times 3$$
$$= 0.38 + 0.234 + 0.148 = 0.762$$

2.5)

After four iterations, we have



from above we can see that our data has been classified correctly, hence classification error is zero.

So our final classifier will have those 3 weak classifiers (decision tree stumps) weighted with their α_t values i.e.

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Since our classifier have reach 100% classification accuracy, we can stop the iteration and take weighted sum of above three classifiers as our final classifier.

3) K-Nearest Neighbour classifier ↓

3.1) A Lazy Classifier :-

a) When a new training example becomes available, we have to retrain the models from scratch for Naive Bayes classifier and Support Vector Machines. for both the algorithm we have to retune our hyper-parameters based on the new training example as these new incoming training example can change the decision boundry/surface over-time. So to get this new boundry/surface, we have to retrain our model from scratch and learn new hyper-parameters.

Whereas, for KNN, since it is a instance based classifier, and there is no model or hyper-parameters are involved, new training example will not make any difference. Just when new test example comes, we have to calculate the distance from all examples including new training example.

b) KNN will need the most computation to infer the class label for this new test example as it will have to calculate the distance of new test example from each of the training example to find the K-nearest neighbours and use them to label the new example.

time complexity → $O(n) + O(n) \Rightarrow O(n)$

($O(n)$ for distance, $O(n)$ for sorting (using Quickselect))

while SVM & Naive Bayes will take $O(1)$ time to classify new example.

3.2) Implement KNN →

a) Step 1 → pre-process the data. Since we have used just first 10% of the data for training set and test set.

Step 2 → for each of the test example, calculate its distance from each of the training example, and get the top K-nearest neighbours for that particular example. we have used the euclidean distance as our distance metric.

e.g. A (x_1, y_1) B (x_2, y_2)

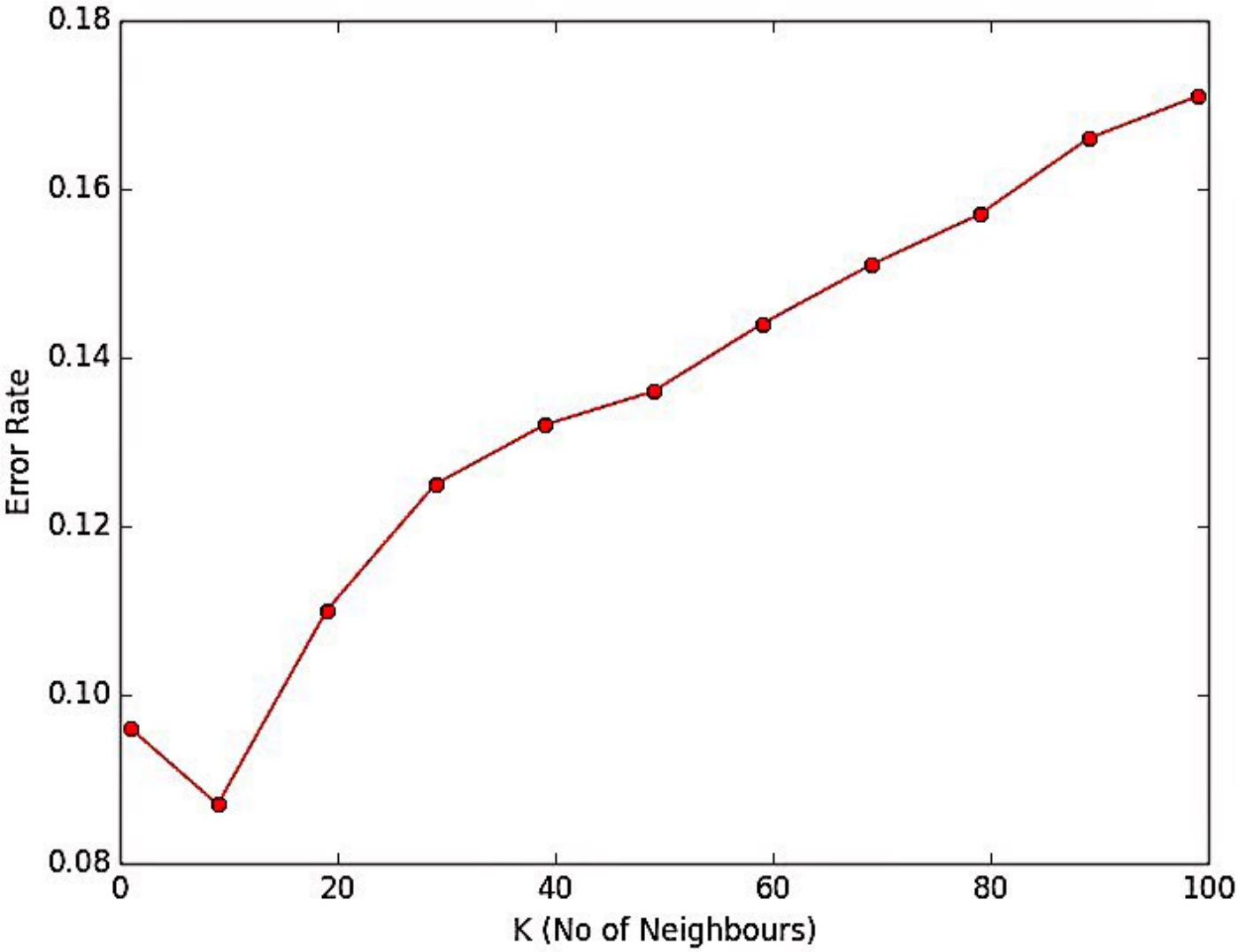
$$d = \sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

In general,

$$d = \sqrt{\sum_{i,j}^{n} (x_i - x_j)^2}$$

Step 3 → Now that we have the top-k neighbours, we will predict the label of our ~~test~~ example by taking the vote of each of the $^{top\text{-}k}$ neighbour's label. Highest voted label will be assigned to the new test example.

Step 4 → Accuracy can be calculated based on the actual label vs the predicted lables for particular test example.



```
For K =  1
Test Error:  0.096
No of Incorrect Labels:  96
-----------------------------------------
For K =  9
Test Error:  0.086
No of Incorrect Labels:  86
-----------------------------------------
For K =  19
Test Error:  0.11
No of Incorrect Labels:  110
-----------------------------------------
For K =  29
Test Error:  0.123
No of Incorrect Labels:  123
-----------------------------------------
For K =  39
Test Error:  0.132
No of Incorrect Labels:  132
-----------------------------------------
For K =  49
Test Error:  0.136
No of Incorrect Labels:  136
-----------------------------------------
For K =  59
Test Error:  0.143
No of Incorrect Labels:  143
-----------------------------------------
For K =  69
Test Error:  0.151
No of Incorrect Labels:  151
-----------------------------------------
For K =  79
Test Error:  0.156
No of Incorrect Labels:  156
-----------------------------------------
For K =  89
Test Error:  0.166
No of Incorrect Labels:  166
-----------------------------------------
For K =  99
Test Error:  0.171
No of Incorrect Labels:  171
-----------------------------------------
```