

# **A Comparative Study of Classification Algorithms**



Tanmay Kalani - S20160010096

G. Mary Ankitha - S20160010029

# DATA MINING

**Data Mining** is a procedure of extraction of helpful data from extensive amount of raw data by methods of **Machine Learning, Statistics** and **Database Systems**.

- Goal is to extract information from data sets and transform it into fathomable shape.
- Data Mining is the analysis step of the “Knowledge Discovery in Databases” process or “KDD”.

## CLASSIFICATION

**Classification** is a **data mining** technique that assigns instances in a data set to target classes.

- It is utilized to discover in which class every datum occurrence is associated with inside a data set.
- Development of the various classifying models is finished utilizing a set of data instances for which related classes were known ahead of time.
- **Neural Networks, Bayes Classifier** and **Decision Trees** are the three classification techniques used here.

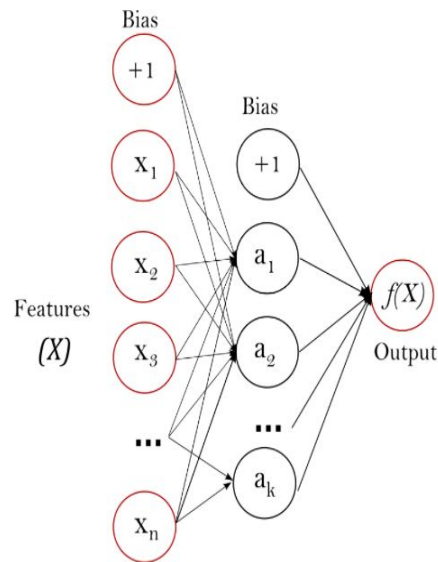
# NEURAL NETWORK

- Feed - Forward Neural Network
- Multi-Layer Perceptron (MLPClassifier)
- Learns using a function  $f(\cdot) : \mathbb{R}^m \rightarrow \mathbb{R}^o$
- Neural Node in the hidden layer transforms the values from the previous layer by weighted sum  $z = w_1x_1 + w_2x_2 + \dots + w_nx_n$
- Non - linear Hyperbolic activation function  $z = \frac{e^z - e^{-z}}{e^z + e^{-z}}$
- MLP uses **Stochastic Gradient Descent**

$$w \leftarrow w - \eta \left( \alpha \frac{\partial R(w)}{\partial w} + \frac{\partial Loss}{\partial w} \right)$$

$$Loss = \frac{1}{2} \sum_{r=1}^c (t_r - z_r)^2$$

- **Time Complexity** is  $O(n m h^k o i)$  where  $n$  is the number of training examples,  $m$  is the number of features,  $h$  is the number of hidden layers,  $k$  is the number of nodes in the hidden layer,  $o$  is the number of output nodes and  $i$  is the number of input nodes.



$\alpha = 0.0001$

$\eta = 0.001$

$\varepsilon = 0.01$

Epochs = 1000

# DECISION TREE

- Non-parametric supervised learning method used for classification
- Uses an optimized version of **CART** algorithm
- **GINI** impurity

$$H(X(m)) = \sum_k p_{mk}(1 - p_{mk})$$

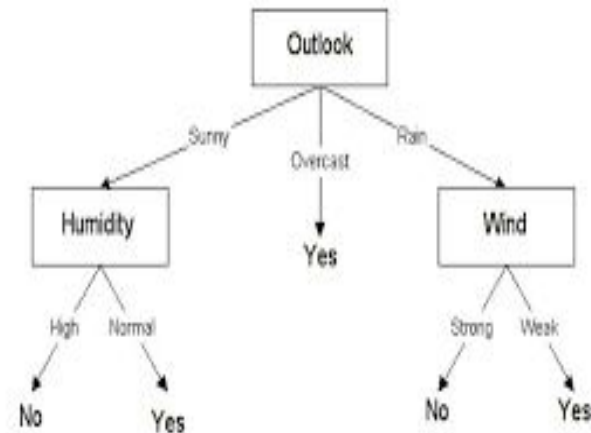
- **Cross-Entropy ( Information Impurity )**

$$H(X_m) = - \sum_k p_{mk} \log(p_{mk})$$

- **Misclassification**

$$H(X_m) = 1 - \max(p_{mk})$$

- **CART** is similar to **C4.5**. It creates tree utilizing the feature and threshold that yields the largest information gain at every node.



- Time Complexity is  $O(n_{features} n_{samples}^2 \log(n_{samples}))$
- If a target classification outcome on taking values  $0, 1, \dots, k-1$ , for node  $m$ , representing a region  $R_m$  with  $N_m$  observations let

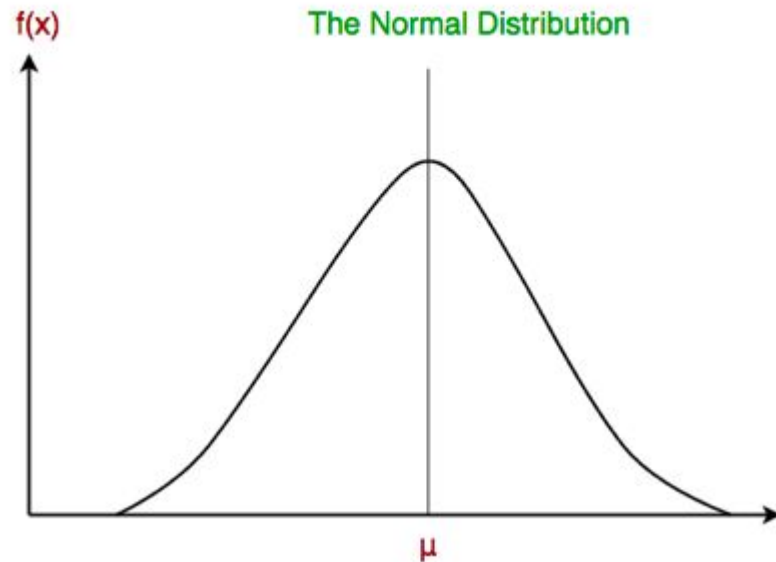
$$p_{m,k} = 1/N_m \sum_{x_i \in R_m} I(y_i = k)$$

# GAUSSIAN NAIVE BAYES

- **Naive Bayes** method are an arrangement of supervised learning algorithms dependent on applying **Bayes** theorem with an assumption of conditional independence between every pair pair of features
- Gaussian distribution is also called **Normal Distribution**.
- Based on probabilistic models

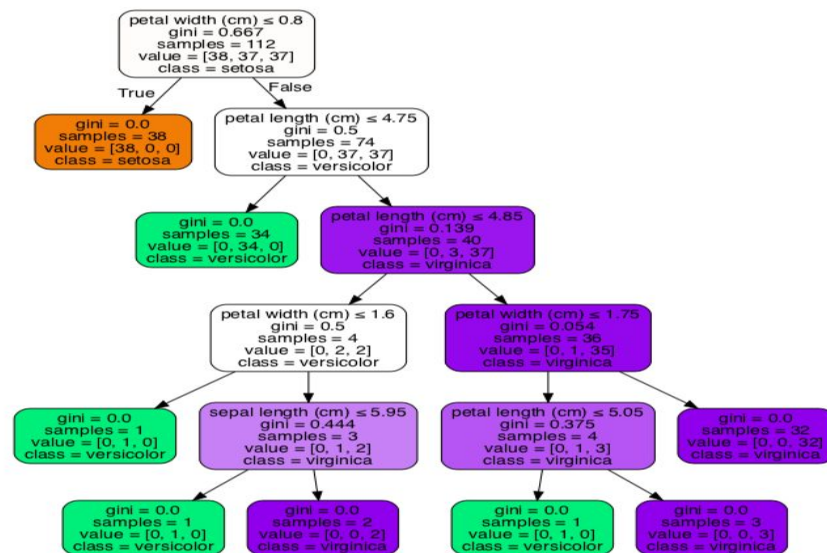
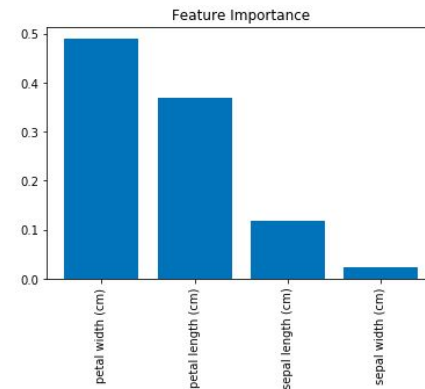
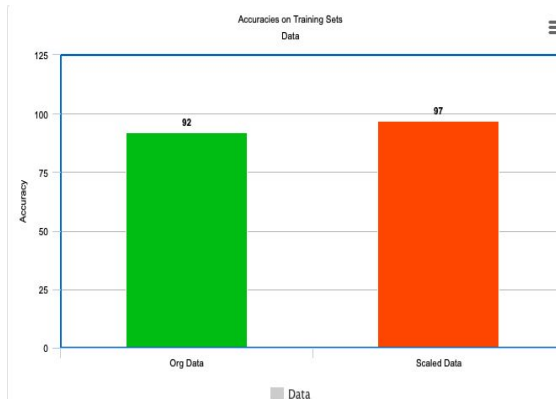
$$P(X_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

- Time Complexity is  $O(nm)$  where  $n$  is the number of training examples and  $m$  is dimensionality of data instances.



# IRIS DATASET

- Number of features = 4
- Classes - Setosa, Versicolor, Virginica
- Test\_ratio = 0.25
- MLPClassifier
  - Before data scaling - 92 %
  - After data scaling - 97 %
- Decision Trees
  - Cross Entropy Impurity
    - Overfit - Test Accuracy - 92.105 %
    - After Overfit - Test Accuracy - 94.30 %
  - GINI Impurity
    - Overfit - Test Accuracy - 92.105 %
    - After Overfit - Test Accuracy - 89.47 %
- Gaussian Naive Bayes
  - Accuracy - 97.368 %
- Most Important Feature - Petal Width



**Figure 10: Feature importance and decision tree**

**Feature Importance:**

Feature	Importance
worst fractal dimension	0.00
worst symmetry	0.00
worst concave points	0.18
worst concavity	0.05
worst compactness	0.00
worst smoothness	0.00
worst area	0.00
worst perimeter	0.00
worst texture	0.08
worst radius	0.62
fractal dimension error	0.01
symmetry error	0.00
concave points error	0.00
concavity error	0.00
compactness error	0.00
smoothness error	0.00
area error	0.00
perimeter error	0.08
texture error	0.00
radius error	0.02
mean fractal dimension	0.00
mean symmetry	0.00
mean concave points	0.00
mean concavity	0.00
mean compactness	0.00
mean smoothness	0.00
mean area	0.00
mean perimeter	0.00
mean texture	0.00
mean radius	0.00

**Decision Tree:**

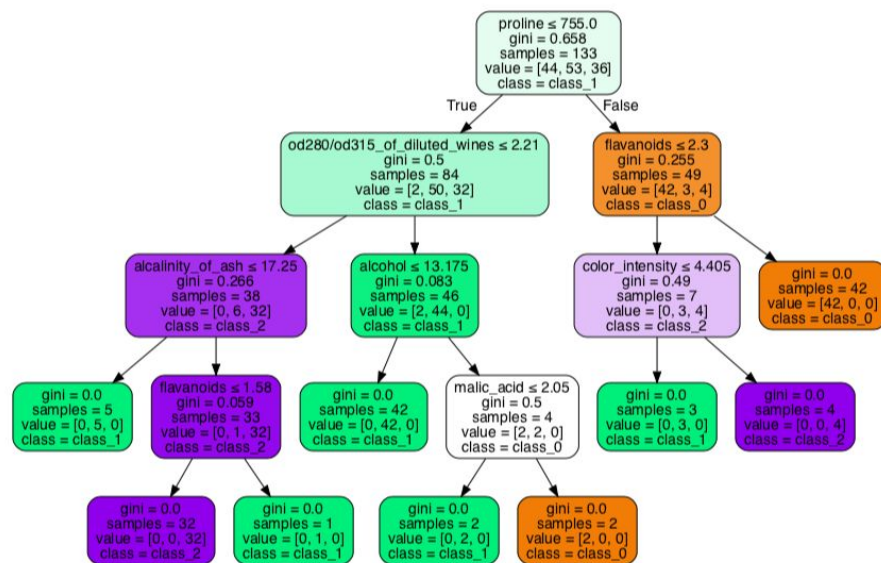
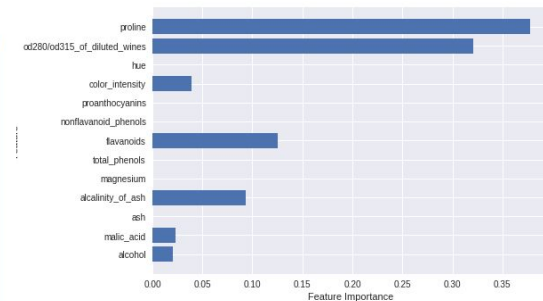
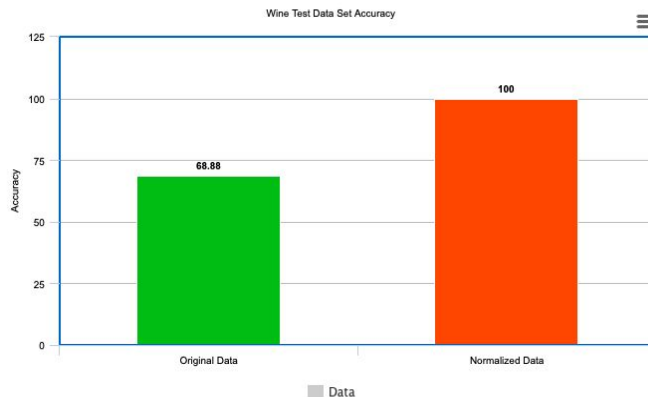
```

graph TD
    Root[ ] -- True --> Node1["worst radius ≤ 16.795  
gini = 0.468  
samples = 425  
value = [109, 267]  
class = benign"]
    Root -- False --> Node2["texture error ≤ 0.473  
gini = 0.106  
samples = 142  
value = [154, 8]  
class = malignant"]
    Node1 -- True --> Node3["worst concave points ≤ 0.136  
gini = 0.161  
samples = 284  
value = [25, 258]  
class = benign"]
    Node1 -- False --> Node4["radius error ≤ 1.048  
gini = 0.031  
samples = 130  
value = [6, 124]  
class = benign"]
    Node2 -- True --> Node5["worst texture ≤ 25.62  
gini = 0.451  
samples = 5  
value = [2, 3]  
class = malignant"]
    Node2 -- False --> Node6["gini = 0.9  
samples = 5  
value = [2, 3]  
class = benign"]
    Node3 -- True --> Node7["smoothness error ≤ 0.003  
gini = 0.025  
samples = 236  
value = [3, 248]  
class = benign"]
    Node3 -- False --> Node8["worst area ≤ 0.17  
gini = 0.375  
samples = 12  
value = [5, 7]  
class = benign"]
    Node4 -- True --> Node9["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node4 -- False --> Node10["mean texture ≤ 19.9  
gini = 0.375  
samples = 8  
value = [0, 9]  
class = benign"]
    Node5 -- True --> Node11["worst symmetry ≤ 0.268  
gini = 0.444  
samples = 2  
value = [1, 1]  
class = malignant"]
    Node5 -- False --> Node12["worst concavity ≤ 0.191  
gini = 0.041  
samples = 179  
value = [2, 3]  
class = malignant"]
    Node6 -- True --> Node13["worst texture ≤ 30.975  
gini = 0.0  
samples = 5  
value = [0, 5]  
class = benign"]
    Node6 -- False --> Node14["gini = 0.0  
samples = 2  
value = [0, 2]  
class = malignant"]
    Node7 -- True --> Node15["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node7 -- False --> Node16["area ≤ 48.7  
gini = 0.014  
samples = 243  
value = [0, 1]  
class = benign"]
    Node8 -- True --> Node17["mean compactness ≤ 0.168  
gini = 0.16  
samples = 19  
value = [0, 19]  
class = benign"]
    Node8 -- False --> Node18["fractal dimension error ≤ 0.002  
gini = 0.0  
samples = 3  
value = [1, 2]  
class = benign"]
    Node9 -- True --> Node19["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node9 -- False --> Node20["gini = 0.0  
samples = 1  
value = [0, 1]  
class = benign"]
    Node10 -- True --> Node21["gini = 0.0  
samples = 3  
value = [0, 3]  
class = benign"]
    Node10 -- False --> Node22["worst feature ≤ 32.35  
gini = 0.004  
samples = 2  
value = [1, 242]  
class = malignant"]
    Node11 -- True --> Node23["gini = 0.0  
samples = 2  
value = [1, 1]  
class = malignant"]
    Node11 -- False --> Node24["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node12 -- True --> Node25["gini = 0.0  
samples = 17  
value = [17, 0]  
class = malignant"]
    Node12 -- False --> Node26["gini = 0.0  
samples = 3  
value = [0, 3]  
class = benign"]
    Node13 -- True --> Node27["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node13 -- False --> Node28["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node14 -- True --> Node29["gini = 0.0  
samples = 2  
value = [0, 2]  
class = malignant"]
    Node14 -- False --> Node30["gini = 0.0  
samples = 3  
value = [0, 3]  
class = benign"]
    Node15 -- True --> Node31["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node15 -- False --> Node32["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node16 -- True --> Node33["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node16 -- False --> Node34["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node17 -- True --> Node35["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node17 -- False --> Node36["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node18 -- True --> Node37["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node18 -- False --> Node38["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node19 -- True --> Node39["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node19 -- False --> Node40["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node20 -- True --> Node41["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node20 -- False --> Node42["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node21 -- True --> Node43["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node21 -- False --> Node44["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node22 -- True --> Node45["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node22 -- False --> Node46["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node23 -- True --> Node47["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node23 -- False --> Node48["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node24 -- True --> Node49["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node24 -- False --> Node50["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node25 -- True --> Node51["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node25 -- False --> Node52["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node26 -- True --> Node53["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node26 -- False --> Node54["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node27 -- True --> Node55["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node27 -- False --> Node56["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node28 -- True --> Node57["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node28 -- False --> Node58["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node29 -- True --> Node59["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node29 -- False --> Node60["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node30 -- True --> Node61["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node30 -- False --> Node62["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node31 -- True --> Node63["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node31 -- False --> Node64["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node32 -- True --> Node65["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node32 -- False --> Node66["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node33 -- True --> Node67["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node33 -- False --> Node68["gini = 0.0  
samples = 1  
value = [1, 0]  
class = malignant"]
    Node34 -- True --> Node69["gini = 0.0  
samples = 1  
value = [0, 1]  
class = malignant"]
    Node34 -- False --> Node70["gini = 0.0  
samples = 1  
value
```

- Number of features = 30
- Classes - Malignant, Benign
- Test\_ratio = 0.25
- MLPClassifier
  - Before data scaling - 92.30 %
  - After data scaling - 95.804 %
- Decision Trees
  - Cross Entropy Impurity
    - Overfit - Test Accuracy - 94.405 %
    - After Overfit - Test Accuracy - 95.804 %
  - GINI Impurity
    - Overfit - Test Accuracy - 93.706 %
    - After Overfit - Test Accuracy - 95.104 %
- Gaussian Naive Bayes
  - Accuracy - 92.30 %
- Most Important Feature - Worst Radius

# WINE DATASET

- Number of features = 13
- Classes - Class\_0, Class\_1, Class\_2
- Test\_ratio = 0.25
- MLPClassifier
  - Before data scaling - 68.88 %
  - After data scaling - 100 %
- Decision Trees
  - Cross Entropy Impurity
    - Overfit - Test Accuracy - 93.33 %
    - After Overfit - Test Accuracy - 94.33 %
  - GINI Impurity
    - Overfit - Test Accuracy - 88.88 %
    - After Overfit - Test Accuracy - 88.88 %
- Gaussian Naive Bayes
  - Accuracy - 95.55 %
- Most Important Feature - Proline





## SUMMARY

- All classifiers were trained and tests on same dissemination of training and testing data sets.
- Each of these algorithms gave palatable outcomes considering they were prepared with not very many data instances.
- Gaussian Naive Bayes scored the highest accuracy ( 97.368 % ) for IRIS data set.
- Decision Tree ( GINI ) impurity scored highest accuracy ( 95.804 % ) for Breast Cancer Data set.
- Decision Tree ( GINI ) impurity scored highest accuracy ( 95.55 % ) for Wine Data set.

# CONCLUSION

- Gaussian Naive Bayes gave the most palatable results for all the three data sets.
- Decision Tree has the lowest time complexity.
- Neural Network have the highest accuracy yet they take the longest time to generate the classifier and have extensibility to due to their large and complex nature.
- Bayesian classifier is a probabilistic model which can end up in many stable states.
- On the off chance that decision trees are permitted to grow totally they may result in an over-fitted model, which gives high accuracy on training data sets but may not perform well on test set.
- Decision trees are based on greedy algorithms where locally ideal decisions are made at every node which does not ensure all round optimal decision tree.
- Neural Network utilizes a non-linear activation function. It has capability to learn non-linear functions, If the arbitrary weights chosen at the start at the algorithm are not appropriate, it can wind up in a local minimum with different validation accuracy.
- It is imperative scaling is done before training and testing.