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 in 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.

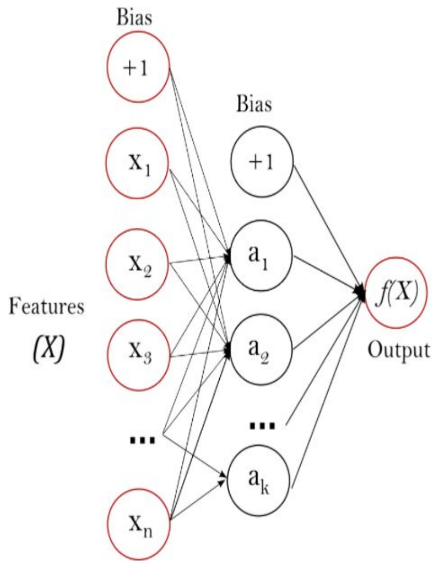
Neural Network

- Feed - Forward Neural Network
- Multi-Layer Perceptron (MLPClassifier)
- Learns using a function $f(\cdot) : R^m \rightarrow R^o$
- 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**

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

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

- **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.



α = 0.0001

η = 0.001

ε = 0.01

Epochs = 1000

Decision Tree

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

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

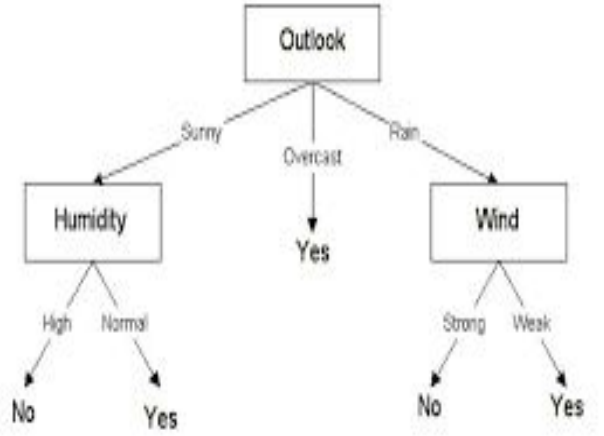
- **Cross-Entropy (Information Impurity)**

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

- **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*** $\boldsymbol{(n_{features} n_{samples}^2 log(n_{samples}))}$
- If a target classification outcome on taking values 0,1,...,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

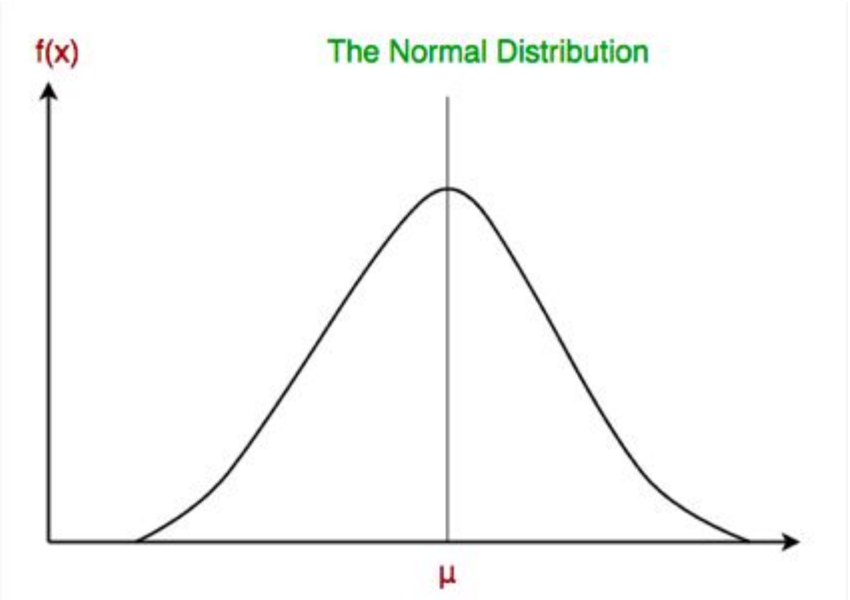
- **Naive Bayes** method are an arrangement of supervised learning algorithms dependent on applying **Bayes** theorem with an assumption of conditional independence between every pair pair of features

- Gaussian distribution is also called **Normal Distribution.**

- Based on probabilistic models

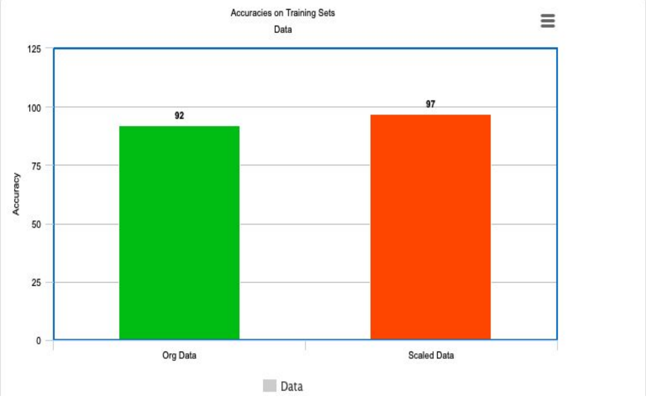
$$P(X_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(- \frac{(x_i - \mu_y)}{2\sigma_y^2} \right)$$

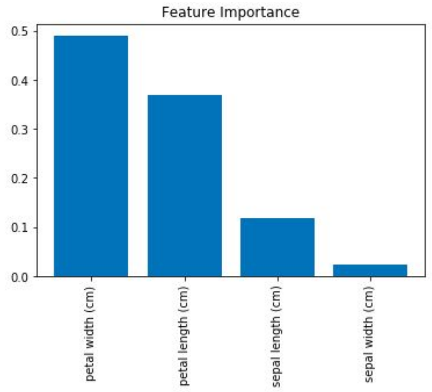
- Time Complexity is ***O (n m)*** where *n* is the number of training examples and *m* is dimensionality of data instances.

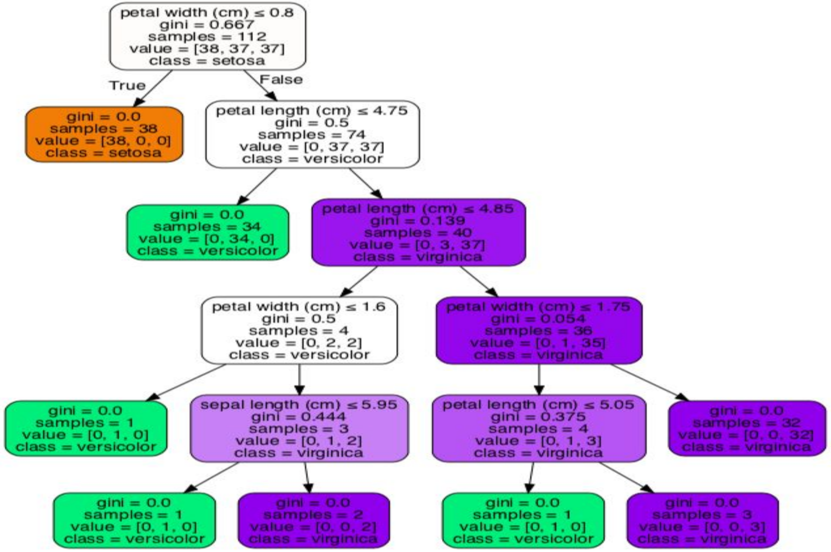


Iris Dataset

- 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

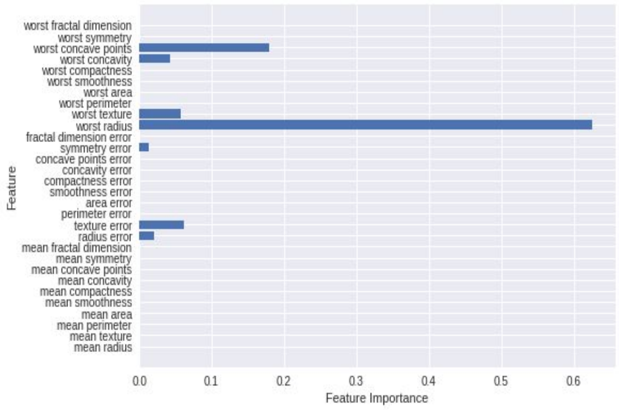


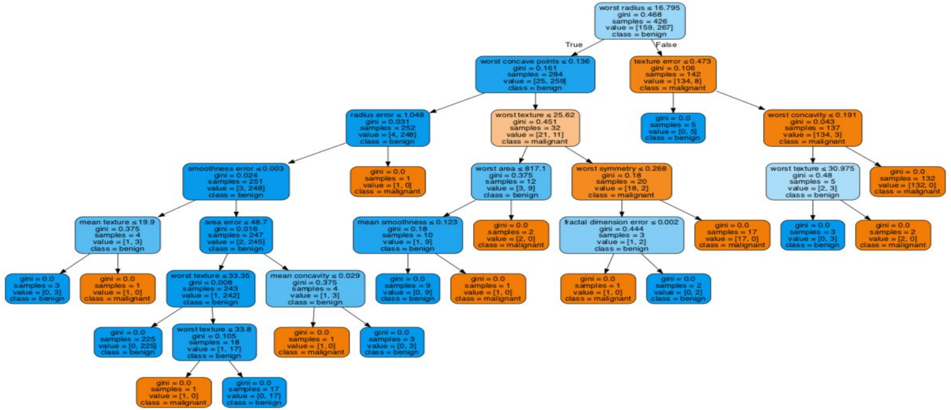




Breast Cancer Dataset

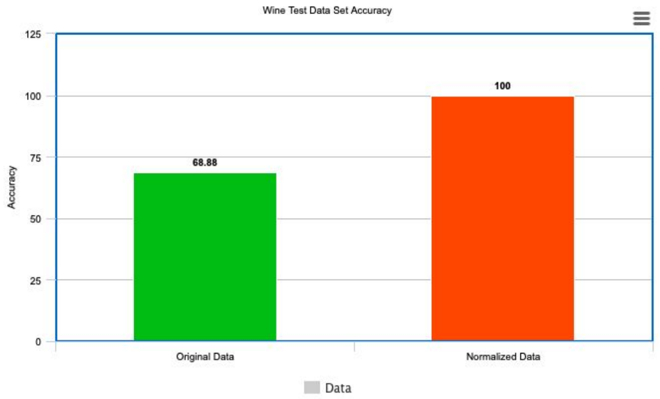
- Number of features = 30
- Classes - Malignant, Benign
- Test_ratio = 0.25
- MLPClassifier
  - Before data scaling - 92.30 %
  - After data scaling - 95.804 %
- Decision Trees
  - Cross Entropy Impurity
    - Overfit - Test Accuracy - 94.405 %
    - After Overfit - Test Accuracy - 95.804 %
  - GINI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

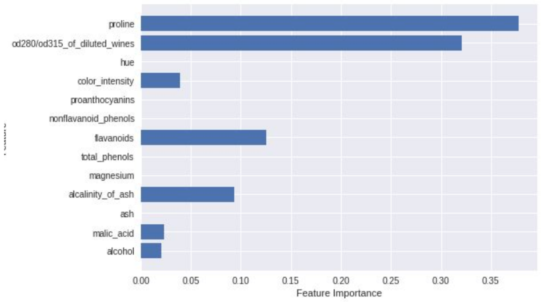


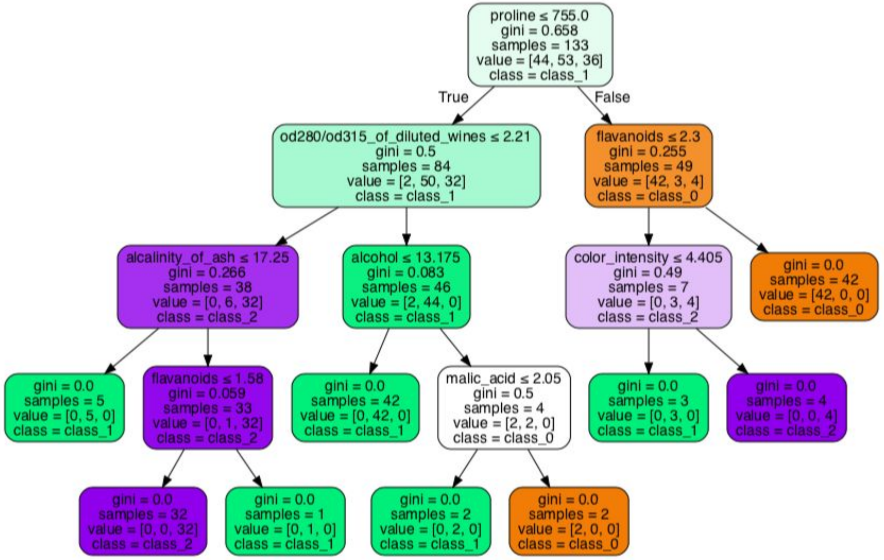


Wine Dataset

- 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.