Advanced Topics in Algorithms Report

Supriya Sindigere Kumaraswamy
MSC(IT)
Technische Hochschule OWL
Lemgo, Germany
supriya.sindigere@stud.th-owl.de

***Abstract*— Technological developments have made human life better. People have started using technology in every field. This is the reason for a large collection of data. Maintaining this data and extracting important information from this data is a big challenge. Machine learning algorithms have contributed to resolving this challenge. In our paper, we have selected three machine learning classification algorithms to classify an autism dataset. Autism is a developmental disability where a person reacts or behaves in a different way. We have used Decision Tree, KNN, SVM to solve our problem of classifying people with autism and people without autism.**

Keywords*— *Autism, Decision Tree, KNN, SVM, Confusion Matrix, AUROC Curve.

I. Motivation

A huge amount of information is generated due to the multi-fold of technology in every sector. This is often the rationale for the boom within the data industry over the past ten years. To obtain decision-making insights, data that has been collected has to be supplemented with its analysis. Data analysis plays an important role in understanding vast volumes of information for further growth and development in any business or industry [1].

Analytical model building is automated by a method of data analysis i.e., Machine Learning. Machine Learning has gained importance because of this ability to analyze bigger, more complex data and automatically produce models which induce more accurate results. Profitable opportunities can be identified by the organization by building these precise models.[2]

Machine learning is classified into Supervised Learning and Unsupervised Learning. Labelled datasets define Supervised Learning. Machine Learning algorithms can supervise or train these labelled datasets into classifying data or predicting outcomes accurately. Supervised Learning has two types of problems classifying and regression. Unsupervised Learning is defined by unlabeled datasets. Without any human intervention, these algorithms can discover hidden patterns in data. Three main problems of Unsupervised Learning are Clustering, Association and Dimensionality Reduction. We use the classification problem of Supervised Learning. [3]

In classification, a given dataset is categorized into classes. We predict the class of a given dataset. These classes are addressed as target, label or categories [4]. Classification is used in health care to predict if a person is diagnosed with a certain disease or not, it is also used for Speech Tagging, Music identification, Quality Control [5]. Classifying a large dataset into different classes is a complex task and the Classification problem simplifies this task by developing a model automatically which can learn from a part of the dataset and predict the classes of remaining values of that dataset.

A person with autism spectrum disorder has problems taking the first step, smiling for the first time and waving bye-bye this is called development disability [6]. They face social, communication and behavioral challenges. People with ASD communicate, interact and learn things differently when compared to other normal people. Some people with ASD require a lot of help in their daily activities and some don't. As there is no medical test available for ASD the diagnosis of ASD is difficult. Early detection of ASD helps in providing necessary help for the person. To identify if a person has ASD or not, we use the Autism dataset which contains 20 features. We have used 3 classification problems Decision Tree Classifier, Support Vector Machine and K-Nearest Neighbor. We analyze the accuracy of these algorithms on the autism dataset.

II. State of Art

A. *Decision Tree:*

The decision tree comes under the supervised learning approach and is mainly used for classification and regression. It resembles a normal tree which includes roots, branches and leaves. Where a Decision tree includes root nodes, branches and leaf nodes. A decision tree is a tree, where each node represents an attribute, a test of an attribute is denoted by an internal (non-leaf) node, the outcome of this test is represented by a branch, a class label is denoted by a leaf node. The topmost node in a Decision tree is called a root node. Root node as the name suggests acts as a parent node [8].

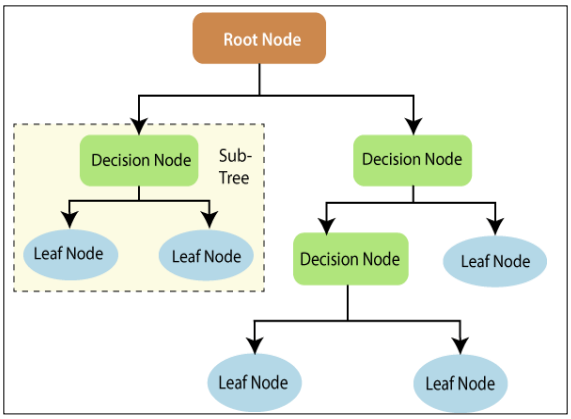

Fig 1: Decision Tree [9]

Fig 1 describes the structure of a Decision Tree. It consists of a root node which is the topmost node. Which further splits into two Decision nodes, this node is further subdivided into the Leaf node which represents the class label.

Decision Tree has made a significant difference in the field of machine learning, image processing and identification of patterns. A series of basic tests can be united efficiently and cohesively using a Decision Tree. In each test, we compare a numeric feature with a threshold value. We use a Decision tree mainly for grouping purposes [9]. Decision Tree is one of the most popular and widely used algorithms in supervised learning. The main reason for its popularity is its interface which can be easily understood by humans [10].

Different types of Decision Tree algorithms are Iterative Dichotomies 3 (ID3), Successor of ID3 (C4.5), Classification and Regression Tree (CART), CHi-squared Automatic Interaction Detector (CHAID), Multivariate Adaptive Regression Splines (MARS), Generalized, Unbiased, Interaction Detection and Estimation (GUIDE), Conditional Inference Trees (CTREE), Classification Rule with Unbiased Interaction Selection and Estimation (CRUISE), Quick, Unbiased and Efficient Statistical Tree (QUEST) [9].

B. K-Nearest Neighbours Algorithm

K-Nearest Neighbors is a simple yet effective non-parametric classification method. To classify a data point *t*, its *k* nearest neighbors are retrieved. To decide the classification for *t* majority voting among the data records in the neighborhood is considered. *k* is the number of nearest neighbors to be considered for the majority voting process. The accuracy of KNN mainly depends on the value of *k*. So, choosing a correct *k* value is an important step in KNN. A simple way of choosing the *k* value is by running the algorithm with different *k* values and choosing the one with maximum accuracy [11].

The first step in KNN is to transform the data points into their mathematical values (vectors) then we calculate the distance between these mathematical values. Distance between each data point and the neighbors is calculated and then the probability of similarity between the data point and neighbors is computed. Points that share the highest probability is considered for classification [12].

The Euclidian distance can be used to find the nearest neighbors. Euclidian distance formula for two points in the plane is given by,

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

Manhattan distance can also be used to calculate the distance between data points and the formula for Manhattan distance is given by,

$$d = (|(x_2 - x_1)| + |(y_2 - y_1)|)$$

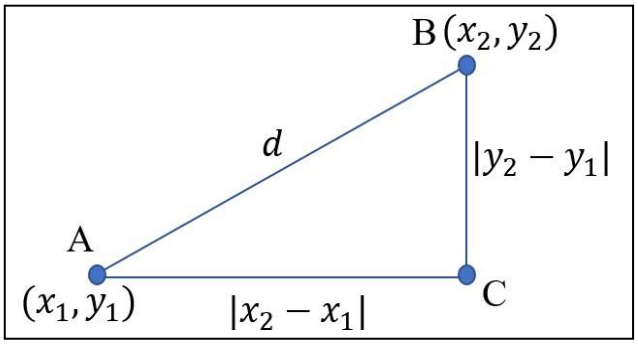

Fig 2: Visualization of Euclidian distance and Manhattan distance to calculate the distance in KNN algorithm

KNN has many advantages but it also has some drawbacks like low efficiency and its dependency on the correct value of *k* [13]. In fig 3 a green diamond is classified into blue class with *k*=8.

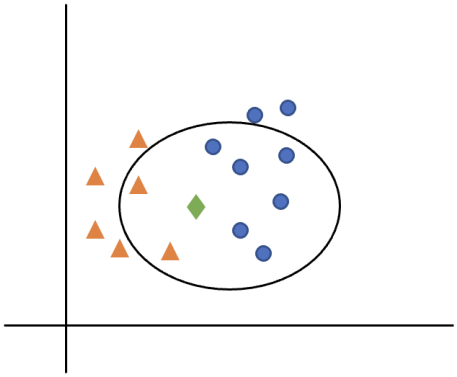

Fig 3: Green diamond classified as a blue class with k=8.

C. Support Vector Machine

The framework of statistical learning theory is used for the development of the Support Vector Machine. Various applications of SVM are time series prediction, face recognition, biological data processing for medical diagnosis [14].

Fig 4 depicts the hyperplane of an SVM and describes how it classifies the data points from the dataset.

In SVM we plot a hyperplane to distinctly classify the data points in N-dimensional space (N – The number of features). Hyperplane with a maximum margin between the data points of both the classes is chosen. In other words, the hyperplane should be as far as possible from Support vectors. Support Vectors are the extreme datapoints of the dataset. Data points are classified according to their position with respect to the hyperplane. The number of features defines the dimension of the hyperplane [15].

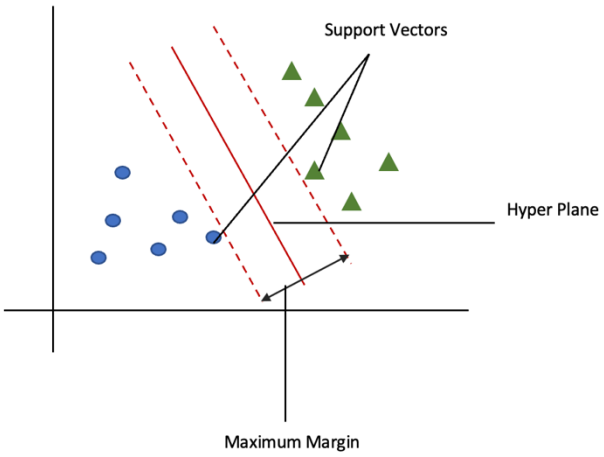

Fig 4: Hyperplane of SVM.

III. Performance Metrics

A. *Confusion Matrix:*

We measure the performance of our classification algorithms using Confusion Matrix. The confusion matrix is a performance measure of a machine learning classification problem with two or more classes as their output. It consists of four different combinations of predicted and actual values present in the form of an N*N Matrix [16].

		Actual Values: Positive (1)	Actual Values: Negative (0)
Predicted Values	Positive (1)	TP	FP
Predicted Values	Negative (0)	FN	TN

Fig 5: Confusion Matrix [16]

The confusion matrix consists of True positive (TP), False Positive (FP), False Negative (FN), True Negative (TN). True positive is when the model predicts a positive value and the actual value is also positive i.e., Both actual value and predicted value are the same. False Positive is also known as type 1 error here the model predicts the value as positive when the actual value is negative. False Negative is also known as type 2 error and the model predicts the value as negative when the actual value is positive. In True Negative, the model predicts the value as negative when the actual value is also negative [17].

We also calculate accuracy, precision, recall, f1-score for all the algorithms to know their performance. Accuracy is the ratio of correctly predicted observations to a total number of observations [18].

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision is the ratio of true positive values and the total number of correctly predicted values [18].

$$Precision = \frac{TP}{TP + FP}$$

Recall can be defined as the number of correctly predicted values from all the positive classes. It is also called sensitivity [16].

$$Recall(Sensitivity) = \frac{TP}{TP + FN}$$

The weighted average of precision and recall is defined as f1-score [18].

$$f1 - score = \frac{2 * Recall * Precision}{Recall + Precision}$$

B. *AUC-ROC Curve*

AUROC curve visualizes or indicates the performance of our classification algorithms. It is also called Area Under the Receiver Operating Characteristics (AUROC) [19].

ROC behaves like a probability curve and it is plotted with TPR (True Positive Rate) against FPR (False Positive Rate). The y-axis is represented by TPR, the x-axis is represented by FPR. The measure of separability is called AUC [19].

TPR is calculated using the formula [19],

$$TPR = \frac{TP}{TP + FN}$$

Specificity is calculated using the below formula [19],

$$Specificity = \frac{TN}{TN + FP}$$

FPR is calculated using, FPR = 1 – Specificity and [19]

$$FPR = \frac{FP}{TN + FP}$$

AUC near to 1 indicates an excellent model this explains that the model can separate the data points correctly into their respective classes [19]. If a model has an AUC value near 0 It predicts the classes of data points incorrectly. If the AUC value is 0.5 then the model does not predict the class of data points [19].

To plot the AUROC curve, we first predict the probabilities for each algorithm, then we calculate the probabilities for a positive outcome. Later we compute the AUROC values for each algorithm using the roc_auc_score function. We calculate the roc curve values using the roc_curve function. At last, we plot the roc curve using matplotlib.

IV. SOLUTIONS

Problem Description: Problem description is similar for all three classification algorithms. That is to identify if a person has autism or not by considering 20 features from the autism dataset.

Given a training dataset D: X -> Y, where D $\subseteq$ X * Y and X $\subseteq$ F, where F is a feature space with 20 features and Y is a set of labels. We need a learning function *f* that learns from the training dataset and classifies the testing dataset T,
f: T -> Y. *f* acts as a classifier [21].

A. Decision Tree

1) Formalization:

Given a training dataset $D: X -> Y$, where $D \subseteq X * Y$ and $X \subseteq F$, where F is a feature space with 20 features and Y is a set of labels. We pick the best possible feature $\theta^* \in F$, out of all the features in F. θ^* splits D into $D_{left}(\theta^*)$ and $D_{right}(\theta^*)$ subsets. These subsets are recursed until a maximum allowable depth is reached [20][21].

2) Algorithm

$Algorithm\ Decision\ Tree$: $training\ vector\ x_i \in R^n\ i = 1 \dots l, label\ vector\ y \in R^l. Q_m \rightarrow\ data\ at\ node\ m, N_m \rightarrow\ Samples, \theta = (j, t_m) \rightarrow\ candidate\ spit, j \rightarrow\ feature, t_m \rightarrow\ threshold.$ [20]

1. Procedure $DT(Q_m, N_m, \theta)$
2. $\theta = (j, t_m)$-> $Q_m^{left}(\theta)$ and $Q_m^{right}(\theta)$
$$Q_m^{left}(\theta) = \{(x, y) | x_j <= t_m\}$$
$$Q_m^{right}(\theta) = Q_m \setminus Q_m^{left}(\theta)$$
3. Loss function $H(\,)$ is used to compute the quality of θ
$$G(Q_m, \theta) = \frac{N_m^{left}}{N_m} H(Q_m^{left}(\theta)) + \frac{N_m^{right}}{N_m} H(Q_m^{right}(\theta))$$
4. Parameter with minimum impurity is chosen.
$$\theta^* = \text{argmin}_\theta\, G(Q_m, \theta)$$
5. For $N_m > min_{samples}\ or\ N_m! = 1$
$$DT(Q_m^{left}(\theta)\ and\ Q_m^{right}(\theta))$$

3) Analysis:

When we applied The Decision Tree classifier for our Autism dataset it gave an accuracy of 100%. So, Decision Tree classified all the data points correctly into their classes. The performance of DT is explained in the experiments and results section.

4) Implementation:

The software which we have used is python 3, Jupyter 6.3.0 for coding. For the decision Tree, we have used the sklearn Machine Learning library.

B. KNN

1) Formalization:

The algorithm takes Training dataset D as input.

$x -> Sx: S_x \subseteq D_{s.t.} |S_x| = k | S_x| = k\ and\ \forall (x', y') \in D \setminus S_x\, \forall (x', y') \in D \setminus Sx \rightarrow dist(x, x') \geq \max(x'', y'') \in S_x dist(x, x'') \rightarrow h(x) = mode(\{y'': (x'', y'') \in S_x\})$
Where $S_x \rightarrow$ Set of neighbours, $x \rightarrow$ Test Point, $mode(.) ->$ Label of highest occurrence [22].

2) Algorithm:

Input: Test point x from the data set D
Output: Class of Test point x

1. Select Sx set of k nearest neighbours.
2. Calculate the Euclidian distance between test point and values from Sx
$$d = \left(\sqrt{(x_2 - x_1)^2} + (y_2 - y_1)^2\right)$$
3. The distance of all the points not in Sx should be far from the test point t.
$$dist(x, x') \geq max_{(x'', y'' \in S_x)}\ dist(x, x'')$$
4. classifier $h(\,)$ returns the most common label for the test point t.
$$h(x) = mode(\{y'': (x'', y'') \in Sx\})$$
5. Test point t is assigned with class label of its nearest neighbour with minimum d value

3) Analysis

For the autism dataset KNN algorithm was performed with an accuracy of 95.75% with a k value 10. All the other Performance metrics are explained in the next section.

4) Implementation:

We have used sklearn.neighbors import KNeighbors Classifier library to implement the KNN algorithm for our dataset.

C. SVM

1) Formalization:

The algorithm takes $\mathbb{D} = \{(\vec{x_i}, y_i)\}$ as input,

$$\mathbb{D} = \{(\vec{x_i}, y_i)\} \rightarrow f(\vec{x}) = sign(\vec{w}^T \vec{x} + b) \rightarrow \{+1\ x_i \rightarrow class\ 1, -1\ x_i \rightarrow class\ 0\}$$

Where x_i is the test data point and y_i is the class label [23].

2) Algorithm:

1. The hinge loss Function is used for maximizing the margin[15].

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

2. We add a regularization function to maximize the margin, so the cost function formula changes [15].

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

3. To find our gradients, we take the partial derivatives with respect to weights [15].

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} (1 - y_i \langle x_i, w \rangle)_+ = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

4. Gradient update when there is no misclassification [15]

$$w = w - \alpha \cdot (2\lambda w)$$

5. Gradient update when there is a misclassification.[15]

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$$

3) Analysis:

In SVM we experimented with different kernel functions and each of them performed with different accuracy values which are explained in the next section.

4) Implementation:

from sklearn.svm import SVC, from sklearn import svm are the libraries used to implement the SVM algorithm.

V. Experiments and Results

We have used Decision Tree Classifier, KNN, SVM classification algorithms to solve our problem. Autism is a development disorder that can be cured by early detection. Early detection of this disorder helps is proving necessary help for the patient. The performance of the algorithms to solve the problem was different. Each algorithm has a different accuracy score, precision, recall and f1 score. This is shown in table 1.

Table 1: Performance metrics of classification algorithms

Algorithms	Accuracy	Precision	Recall	F1-Score
Decision Tree	1.0	1.0	1.0	1.0
KNN (k=10)	0.9575	0.98	0.8596	0.9158
SVM(kernel=linear)	1.0	1.0	1.0	1.0
SVM(kernel=rbf)	0.9952	1.0	0.9824	0.9911
SVM(kernel=poly degree=3)	0.8726	1.0	0.5263	0.6896
SVM(kernel=poly Degree=4)	0.8301	1.0	0.3684	0.5384
SVM(kernel=poly Degree=6)	0.8207	1.0	0.3333	0.5

Decision Tree and linear SVM has the highest accuracy of 1.0. Their precision, recall and f1-score are also 1.0. These values indicate that Decision Tree and linear SVM performed the best when compared with other algorithms on the autism dataset. KNN has an accuracy of 0.9575 with 10 k nearest neighbors. After trying different integer values for k, k=10 was chosen because of its high accuracy. Radial Basis Function (rbf) SVM has the second-highest accuracy with a value of 0.9952.

As recall value is important for the autism dataset because it indicates if the algorithms identified the relevant data accurately or not. Our algorithms should identify if a person has autism or not accurately. Decision Tree and linear SVM has a recall value of 1.0, indicating that these algorithms accurately identified if a person has autism or not [24].

We have considered linear, rbf, degree 3 polynomial, degree 4 polynomial, and degree 6 polynomial for the SVM algorithm. Linear SVM performed the best among these.

We have constructed an AUROC curve for all three algorithms. From this curve, we understood the Decision Tree and SVM perform the best because they have an AUROC value of 1.0, indicating that they accurately identified the classes of data points.

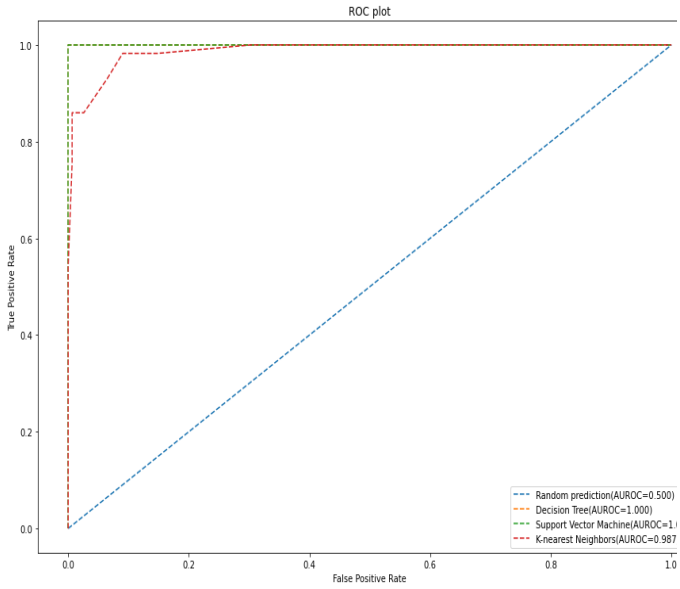

Fig 6: AUROC Curve

VI. Summary and Outlook

We used three popular classification algorithms to solve our problem of identifying autism in a person. Decision Tree and SVM were the best algorithms with the highest accuracy, precision, recall and f1-recall values. One disadvantage is that the dataset has limited data values and in order to identify any disease we need a bigger dataset to build a model with at most accuracy.

References:

1. TechnologyHQTechnologyHQ is a platform about business insights. “Importance of Data Analytics in the Modern World.” *TechnologyHQ*, 11 Sept. 2021, https://www.technologyhq.org/importance-data-analytics-modern-world/.
2. “Machine Learning: What It Is and Why It Matters.” *SAS*, https://www.sas.com/en_us/insights/analytics/machine-learning.html#machine-learning-importance.
3. “Supervised vs. Unsupervised Learning: What's the Difference?” *IBM*, https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning.
4. “Classification in Machine Learning: Classification Algorithms.” *Edureka*, 16 Dec. 2021, https://www.edureka.co/blog/classification-in-machine-learning/.
5. “Classification Problems.” *Brilliant Math & Science Wiki*, https://brilliant.org/wiki/classification/.
6. “Facts about Developmental Disabilities.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 13 Sept. 2021, https://www.cdc.gov/ncbddd/developmentaldisabilities/facts.html.
7. “What Is Autism Spectrum Disorder?” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 25 Mar. 2020, https://www.cdc.gov/ncbddd/autism/facts.html.
8. Patel, Harsh & Prajapati, Purvi. (2018). Study and Analysis of Decision Tree Based Classification Algorithms. International Journal of Computer Sciences and Engineering. 6. 74-78. 10.26438/ijcse/v6i10.7478.
9. Jijo, Bahzad & Mohsin Abdulazeez, Adnan. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. Journal of Applied Science and Technology Trends. 2. 20-28.
10. S. Pathak, I. Mishra and A. Swetapadma, "An Assessment of Decision Tree based Classification and Regression Algorithms," 2018 3rd International Conference on Inventive Computation Technologies (ICICT), 2018, pp. 92-95, doi: 10.1109/ICICT43934.2018.9034296.
11. Guo, Gongde & Wang, Hui & Bell, David & Bi, Yaxin. (2004). KNN Model-Based Approach in Classification. https://www.researchgate.net/publication/2948052_KNN_Model-Based_Approach_in_Classification.
12. Pandey, Ashwin. “The Math behind Knn.” *Medium*, Artificial Intelligence in Plain English, 8 Jan. 2021, https://ai.plainenglish.io/the-math-behind-knn-7883aa8e314c.
13. jabbar, M. A., Deekshatulu, B. L., & Chandra, P. (2013). Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm. *Procedia Technology*, *10*, 85–94. https://doi.org/10.1016/j.protcy.2013.12.340.
14. Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12. https://www.researchgate.net/publication/221621494_Support_Vector_Machines_Theory_and_Applications/citation/download
15. Gandhi, Rohith. “Support Vector Machine - Introduction to Machine Learning Algorithms.” *Medium*, Towards Data Science, 5 July 2018, https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.
16. Narkhede, Sarang. “Understanding Confusion Matrix.” *Medium*, Towards Data Science, 15 June 2021, https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62.
17. “Confusion Matrix for Machine Learning.” *Analytics Vidhya*, 23 July 2021, https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/
18. Solutions, Exsilio, and * Name. “Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures.” *Exsilio Blog*, 11 Nov. 2016, https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/.
19. Narkhede, Sarang. “Understanding AUC - Roc Curve.” *Medium*, Towards Data Science, 15 June 2021,

https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.
20. "1.10. Decision Trees." *Scikit*, https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart.
21. Jackermeier, Mathias. "DtControl: Decision Tree Learning for Explainable Controller Representation." *MediaTUM*, 1 Jan. 1970, https://mediatum.ub.tum.de/1547107?id=1547107.
22. Lecture 2: K-Nearest Neighbors / Curse of Dimensionality, https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote02_kNN.html.
23. *Support Vector Machines: The Linearly Separable Case*, https://nlp.stanford.edu/IR-book/html/htmledition/support-vector-machines-the-linearly-separable-case-1.html.
24. "Precision vs Recall: Precision and Recall Machine Learning." *Analytics Vidhya*, 9 Mar. 2021, https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/.