**RESEARCH PAPER**

# A Comprehensive Review of Machine Learning Models: Principles, Applications, and Optimal Model Selection

Amna Faisal[1], N.Z Jhanjhi[1], Humaira Ashraf[1], Sayan Kumar Ray[1] , Farzeen Ashfaq[1], Omar A. Alhawi[2] , Azeem Khan[3]

*[1]Taylors University, 47500 Subang Jaya, Selangor, Malaysia*
*Emails: 0360943@sd.taylors.edu.my, noorzaman.jhanjhi@taylors.edu.my, humaira.ashraf@taylors.edu.my, sayan.ray@taylors.edu.my, farzeen.ashfaq@sd.taylors.edu.my)*
*[2]College of Engineering University of Technology Bahrain Kingdom of Bahrain*
*Email : Oaalhawi @utb.edu.bh*
*[3]Faculty of Islamic Technology, Universiti Islam Sultan Sharif Ali (UNISSA), Brunei Darussalam,*
*Email : azeem@unissa.edu.bn*

**ABSTRACT**
Machine Learning (ML) continues to evolve rapidly, with revolutionary advancements emerging daily. Its pervasive impact extends across multiple sectors, including healthcare, finance, security, and business management, where it extracts value from previously untapped data reservoirs. Despite the burgeoning growth, there remains a significant void in the literature that concisely details the principles of ML algorithms and offers assistance for optimal model selection tailored for a given scenario. This paper fills this gap by providing an exhaustive review of supervised and unsupervised ML models. Through an in-depth analysis encompassing algorithmic intricacies, applications, strengths, weaknesses, ideal-case scenarios, pitfalls, essential preprocessing requirements, and suitability (SWiPES), this research provides an understanding of the intricate landscape of ML models. A critical contribution is the provision of strategic visualizations that succinctly encapsulate the SWiPES of each model, aiding in the swift identification of the most fitting model for a given real-world scenario. Furthermore, this paper provides a practical blueprint, a compass for researchers, practitioners, and ML enthusiasts, facilitating them to make informed decisions about the most appropriate model based on specific problem domains and dataset characteristics. This research intends to be the definitive ML review, unlocking the potential for precision and insight in navigating the complicated landscape of machine learning.
**Keywords**: Machine Learning, Strategic Guide, Unsupervised Learning, Supervised Learning, Best- fit Models
**How to cite this article:** Faisal A, Jhanjhi NZ, Ashraf H, Ray SK, Ashfaq F, Alhawi OA, Khan A., A Comprehensive Review of Machine Learning Models: Principles, Applications, and Optimal Model Selection .Int J Drug Deliv Technol. 2026;16(2s): 1-32; DOI: 10.25258/ijddt.16. 1-32
**Source of support:** Nil.
**Conflict of interest:** None

## INTRODUCTION

ACHINE Learning (ML) is a subset of artificial intelligence that employs statistical techniques and algorithms to enable machines to make human-like decisions [1] by gathering experience from vast data analysis. Speech Recognition [2], Natural Language Processing [3], and Computer Vision [4] are currently the most trending fields of ML. In speech recognition, ML has enabled computers to [5] transcribe spoken words and [6] recognize voice instructions. It is now essential to virtual assistants, voice-controlled products, transcription services, and hands-free communication, increasing productivity and accessibility. In Natural Language Processing (NLP), ML has revolutionized
[7] sentiment analysis, [8] topic modeling, [9] chatbots, [10] language translation, [11] fake advertisement detection, and [12] text generation, facilitating content creation, customer service, and cross-language communication. Then, in Computer Vision, ML models have advanced in understanding and interpreting visual data, enabling applications such as [13] object detection, [14] image classification, [15] limb gesture recognition, and [16] traffic management.
Not just this, machine learning has permeated all facets of modern life, influencing industries, services, and daily activities, as shown in Figure 1. In the healthcare sector, [17] medical diagnosis, [18] disease risk prediction, [19] drug discovery, and personalized treatment are all aided by machine learning. In finance, ML models power [20] fraud detection, [21] risk assessment, and [22] algorithmic trading. The transportation sector owes machine learning for [23] self-driving cars, [24] route optimization, and [25] demand forecasting in ride-sharing services. In the E-commerce sector, ML has enhanced [26] recommendation systems, [27] personalized marketing, and [28] inventory management. Lastly, [29] automated course suggestions, [30] intelligent tutoring systems, and [31] plagiarism

**Author for Correspondence: swapnas.sdc@saveetha.com*

detection are some contributions of ML to the education sector. Nevertheless, machine learning has emerged as a game changer, transform- ing [32]–[34] countless industries. It has almost limitless uses in enhancing businesses, services, and personal experiences.


**FIGURE 1. Fields benefited by Machine Learning**

Despite the rapid advancements in machine learning, [36] a considerable lack of research and guidance persists in selecting the best machine learning model for case-by-case scenarios. In other words, there is a scarcity of proper knowledge presentation to assist the selection of a high-performing model for a given dataset and problem. While various studies have attempted to demystify the complexity of machine learning, most are domain-specific. [37], discusses ML models for wireless network improvements, [38] presents a survey of ML models for wearable IoT devices, [39] does the same for ML models in computer architecture design, [40] surveys ML models in healthcare and, [41] reviews ML models for social media analysis. A comprehensive guide that properly navigates the dense maze of model selection remains elusive. No previous research, to the best of our knowledge, provides the depth and clarity required for optimal model selection. Table 1 summarizes existing surveys on machine learning and how our paper enhances on them

This research investigates a wide range of supervised and unsupervised learning techniques, and makes the following contributions:

Comprehensive analysis of 10 highly developed ma- chine learning models: 1) Linear Regression, 2) Logistic Regression, 3) Decision Trees, 4) Random Forest, 5) Naive Bayes Classifier, 6) Support Vector Machines,

7) K Nearest Neighbour, 8) K-means Clustering, 9) Gaussian Mixture Models, and 10) Principal Compo- nent Analysis.

Our research makes a groundbreaking contribution to the field of machine learning by disclosing a pioneer- ing decision-making blueprint, as shown in Figure 25,

designed for the most effective selection of machine learning models given a real-world scenario.

Engaging in a comprehensive exploration of machine learning intricacies, our research delves into the algorithmic specifics, practical applications, and the indispensable **S**trengths, **W**eaknesses, **I**deal case (Best- case), **P**itfalls (Worst-case), **E**ssential Preprocessing, and **S**uitability (**SWIPES**) of these models.

Our research expands beyond text analysis by creating novel visualizations that dynamically reflect the core of each machine-learning model.

The remaining paper is sectioned as follows: Section 2 presents a Literature Review, Section 3 briefs about the intuition and SWIPES of Supervised and Unsupervised Learning models, Section 4 guides about optimal model selection, Section 5 highlights the Future Work, and finally, Section 6 is the Conclusion.

## LITERATURE REVIEW

Machine Learning can broadly be categorized into two classes: Supervised Learning and Unsupervised Learning. This research investigates the intuition and SWIPES of the supervised and unsupervised models shown in the taxonomy in Figure 2

| Paper | Year | Major Contribution | Comparison with our research |
|---|---|---|---|
| [35] | 2022 | Explains the methods, implementation tools, advantages, and disadvantages of several supervised algorithms and concludes that each model has its unique way of extracting value from data. | Not only briefs the methods, implementation, pros, and cons of the supervised models discussed by [35], but also sheds light on their real-world applications and the best scenario to use them. |
| [36] | 2021 | Briefs about the principles and applicability of several ML algorithms and discusses the challenges and potential research directions for the ML domain. | Provides a comprehensive overview of 10 popularly used machine learning models, their strengths, weaknesses, and real-world applicability. Also resolves the key challenge [36] claims, is faced by the ML field. |
| [42] | 2021 | Briefly overviews some classification, regression, and clustering techniques and real-world applications of ML. | The intuition, merit, demerits, and real-world applications of the most popularly deployed classification, regression, clustering, and dimensionality reduction ML models are thoroughly discussed among their SWIPES. |
| [43] | 2019 | 10 ML and 1 DL models are reviewed, including their merits and demerits from the practical application perspective. | Because the scope of this research is constrained to ML only, 10 ML models are thoroughly comprehended, their essential preprocessing requirement to combat their demerits, and their ideal and worst real-world applications and datasets are discussed. |
| [44] | 2019 | Regression, classification, clustering, dimensionality reduction, association, instance-based learning, deep learning, and ensemble learning are concisely discussed along with their applications. | Regression, classification, clustering, dimensionality reduction, and ensemble learning algorithms are comprehensively explained, and their applications are mentioned across the literature. The ideal and worst cases of these models, their strengths and weaknesses, are also mentioned. |

.

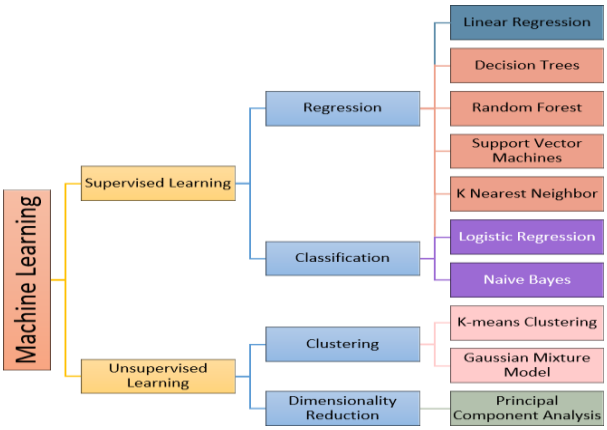


**FIGURE 2. A taxonomy of the ML models discussed in this paper. Note that some supervised learning models cater strictly to regression, some cater strictly to classification, while others can be used for both.**

***SUPERVISED LEARNING***

Supervised learning is a fundamental concept in machine learning, [45] where algorithms learn from labeled training data to make predictions about unseen data. This approach, encompassing classification and regression, enables algorithms to discern underlying patterns by mapping input features to target variable values. Classification and Regression are the two main branches of supervised learning. Some models are designed strictly for classification, such as [46] Naive Bayes and [47] Logistic Regression, while some are made strictly for regression, such as [48] Linear Regression. Others can be used for both purposes which include [49], [50] Decision Trees, [51], [52] Random Forest, [53], [54] K-Nearest Neighbor and [55], [56] Support Vector Machines. In classification, algorithms are trained to assign predefined categories or labels to input data based on patterns learned from labeled training data. The resourcefulness of classification lies in its ability to make automated decisions, aiding in complex tasks, a summarized overview of which is presented in Table 2.

Both classification and regression algorithms [57] discover latent patterns and relationships from labeled data and apply this knowledge to make predictions about new unlabeled samples. The only difference is that classification models predict discrete class labels while regression models predict continuous numerical values. Regression is like drawing a line through data points to represent their relationship best. Regression offers diverse utility in numerous fields, whose summary can be found in Table 3.

***UNSUPERVISED LEARNING***

Unsupervised learning is concerned with analyzing and extracting insights from an unlabeled dataset. That is to say that the data points in the dataset have no preconceived class labels. Clustering and Dimensionality Reduction fall under the umbrella of unsupervised learning.

Clustering involves grouping similar data points together based on their characteristics or features. It aims to identify inherent patterns, similarities, or relationships within a dataset without prior knowledge of the class labels or categories. In clustering, data points are assigned to clusters based on their similarity to each other, aiming to maximize the inter-cluster distance and minimize the intra-cluster distance. Data points that lie in close proximity to each other in the feature space are grouped in one cluster, while those far away are grouped in others. The similarity between

**TABLE 2. Applications of classification algorithms in Literature**

| Algorithm | Paper | Year | Application Domain | Usecase |
|---|---|---|---|---|
| | [47] | 2020 | Image classification | Multiclass indoor-outdoor scene classification |
| | [68] | 2020 | | Cardiovascular disease risk prediction |
| | [69] | 2020 | Healthcare | Analyzing social determinants in relation to elderly |
| Logistic Regression | | | | personal healthcare self-care |
| | [70] | 2020 | Customer Care | Churn prediction in telecommunication |
| | [49] | 2021 | Healthcare | Breast cancer classification |
| | [71] | 2019 | Materials Engineering | Buckling load classification |
| Decision Tress | [74] | 2020 | Banking | Intrusion detection in credit cards |
| | [52] | 2021 | Material Science | Classification of electrode properties in lithium-ion |
| | | | | batteries |
| Random Forest | [79] | 2020 | Stock Market | Feature selection in Chinese stock market prediction |
| | [81] | 2020 | Banking | Credit card holder default prediction |
| | [46] | 2022 | Healthcare | Corona Virus detection |
| | [85] | 2021 | | Spam email detection |
| | [88] | 2019 | Document Classification | Political article categorization |
| | [86] | 2020 | | Mood classification on medical data |
| Naive Bayes | [87] | 2019 | Sentiment Analysis | Hotel reviews classification |
| | [56] | 2020 | Cybersecurity | Malicious user classification in radio networks |
| | [96] | 2020 | Neuro-imaging | Early dementia diagnosis |
| Support Vector | [97] | 2023 | | Cancer classification |
| Machines | [98] | 2020 | Healthcare | Protein Fold Recognition |
| | [53] | 2020 | Anomaly detection | Sea-surface small target detection |
| | [108] | 2022 | Recommender System | User and article based recommender system for an e- |
| K Nearest Neighbor | | | | commerce platform |
| | [109] | 2022 | Heathcare | Parkinson Disease diagnosis |

**TABLE 3.** Applications of regression algorithms in Literature

| Algorithm | Paper | Year | Application Domain | Usecase |
|---|---|---|---|---|
| | [48] | 2022 | Automobile | Used car price estimation |
| | [62] | 2019 | Real Estate | House price prediction |
| | [63] | 2020 | Sales | Sales forecasting in big mart companies |
| Linear Regression | [64] | 2021 | Healthcare | Galectin-3 levels analysis for coronary heart dis- |
| | | | | eases |
| | [50] | 2020 | Agriculture | Estimation of soil moisture |
| | [73] | 2021 | Sales | Newly launched seasonal products sales forecast- |
| Decision Tress | | | | ing |
| | [75] | 2020 | Environmental Management | Water quality prediction. |
| | [51] | 2020 | Academia | Ed-tech impact examination on academic perfor- |
| Random Forest | | | | mance |
| | [80] | 2019 | Drug discovery | Dosage response prediction |
| Support Vector | [55] | 2020 | Agriculture | Prediction of soil properties in MIR spectroscopy |
| Machines | [124] | 2022 | Missing value prediction | Missing value imputation in environmental time |
| | | | | series data |
| | [54] | 2020 | Material Sciences | Remaining useful life estimation of lithium-ion |
| K Nearest Neighbor | | | | cells. |
| | [106] | 2023 | Sustainable Architecture | Thermal load prediction in buildings |

data points is typically determined using distance measures such as Euclidean distance or cosine similarity. Table 4 summarizes some of the applications of clustering algorithms found in the literature.

Dimensionality reduction is a form of unsupervised learning used to reduce the number of features in a dataset in a way that features containing the maximum information are preserved while others are discarded. Many real-world datasets can have a large number of features, which can lead to problems like the curse of dimensionality, overfitting, and reduced interpretability. Dimensionality reduction solves these issues by translating the data into a lower-dimensional space while keeping as much variability as feasible from the original data. Table 5 summarizes some of the applications of Principal Component Analysis as a dimensionality reduction algorithm from the literature.

## INTUITION AND SWIPES

This section presents a detailed analysis of the intuition, Strengths, Weaknesses, ideal-case (best-case), pitfalls (worst-case), Essential preprocessing requirements, and Suitability (**SWIPES**) of the ML models mentioned in Figure 2.

### *LINEAR REGRESSION*

Linear Regression is the simplest supervised learning algorithm to exist. It is a regression model used for predict

| Algorithm | Paper | Year | Application Domain | Usecase |
|---|---|---|---|---|
| | [114] | 2022 | Academia | Collge counselor scoring |
| | [115] | 2021 | Healthcare | Hepatitis disease diagnosis |
| | [116] | 2021 | Traffic Management | Vehicle license plate detection |
| K-means Clustering | [117] | 2020 | Document Clustering | Text documents clustering on several datasets |
| | [119] | 2021 | Economics | Human capital role analysis in industrialization of |
| Gaussian Mixture | | | | Iranian economy |
| Model | [120] | 2020 | Enviromental Management | Passive tracers density estimation |
| | [121] | 2019 | Anomaly Detection | Anomaly detection in Time Series data of EEG and |
| | | | | electrical equipment current. |

**TABLE 5. Applications of dimensionality reduction in Literature**

| Algorithm | Paper | Year | Application Domain | Usecase |
|---|---|---|---|---|
| | [122] | 2020 | Healthcare | Feature selection in kidney ultrasound images. |
| Principal Component | [123] | 2022 | Noise Reduction | Denoising in graph neural networks |
| Analysis | [125] | 2020 | Customer Care | Feature reduction in telecom customer segmentation |

ing continuous numerical values based on input features. As the name suggests, the model establishes [58] a linear relationship between the dependent (target) variable and the independent variables (features). Two commonly used linear regression models are simple linear regression and multivariate linear regression.

A simple linear regression model is one where a single independent variable models the dependent variable. This algorithm aims to find a best-fit line that [59] best minimizes the sum of squared differences between the predicted values and the actual values of the target variable. These differences are called residuals. Mathematically, the model can be defined using the equation below:

$$y_{target} = \beta 0 + \beta 1 x_{feature} \quad (1)$$

Here $y_{target}$ is the dependent variable, $x_{feature}$ is the independent variable, and $\beta 0$ and $\beta 1$ are the weights/coefficients. More precisely, $\beta 0$ is a bias coefficient, also called the intercept, which represents the value of $y_{target}$ when $x_{feature}$ is 0, and $\beta 1$ is the slope, representing the change in $y_{target}$ for one unit change in $x_{feature}$.

Figure 3 visualizes the relationship between the dependent and independent variable. Here Line of Regression is the best-fit line that ideally minimizes the cost function of the model and helps understand the direction, strength, and trend of the relationship between the target and feature variables. In most linear regression models, Mean Squared Error (MSE) [60] is used as the cost function used. It is calculated by taking the average of squared residuals as shown in Equation 2

$$MSE = \frac{1}{n} \sum^{n} (y_{pred} - y_{actual})^2 \quad (2)$$

actual dependent variable value.

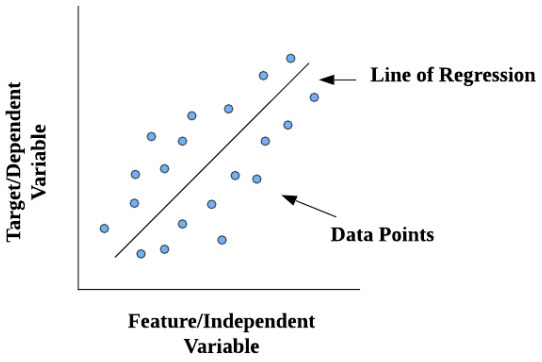


**FIGURE 3.** Graph of a simple linear regression model.

Once coefficients of the line of regression are determined, Equation 1 is used to make predictions of the dependent variable based on the value of the independent variable.

In the real world, however, data is complex, and for an accurate prediction of the dependent variable, several input features are needed. Here, the [58] multivariate linear regression model comes to the rescue, which considers multiple independent variables to predict the dependent variable. The equation for this algorithm goes as follows:

$$y_{target} = \beta 0 + \beta 1 x1 + \beta 2 x2 + \beta 3 x3 .... \beta n xn \quad (3)$$

Again $y_{target}$ is the dependent variable, and $\beta 0$ is the

Here $n$ is the total number of points in our dataset, $y\_pred$ is the predicted dependent variable value and $y\_actual$ is the intercept. $X1, X2 ..... Xn$ are the independent variables, and $\beta 1, \beta 2 ... \beta n$ are the weights of their respective features.
The objective of multivariate linear regression is the same as that of simple linear regression: to find a line of best fit, and predictions are made by plugging the values of features in Equation 3 .

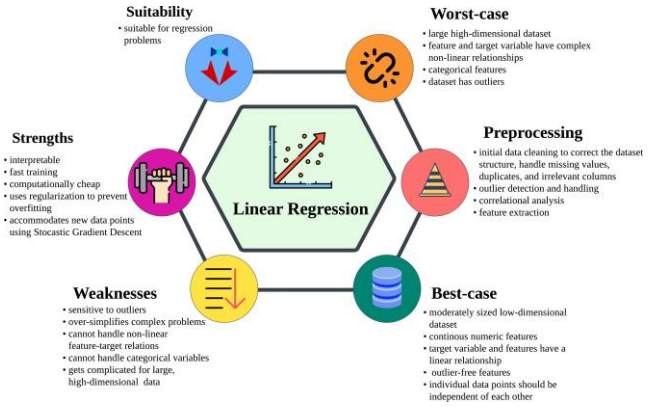


**FIGURE 4.** The SWIPES of Linear Regression.

Figure 4 explains the [61] SWIPES for Linear Regression. The model is straightforward, hence, best suited for simple real-world problems. For example, [62] a house price prediction model can ideally use linear regression, where the model can be trained on features like location, area of land, and number of rooms. A [63] sales forecasting model can deploy linear regression to predict future sales based on historical data, or it can be used to analyze how changes in advertisement expenditure impact sales figures. Linear Regression can also be used in risk assessment to quantify the relationship between risk factors and outcomes. For instance, [64] assessing the relationship between galectin-3 and the risk of heart disease

***LOGISTIC REGRESSION***

Logistic regression is a statistical modeling technique applied to classification problems, specifically binary classification. It is useful for modeling the relationship between a dichotomous categorical dependent variable and one or more independent variables. The model predicts the class of an instance by calculating the probability of an event occurring. Logistic regression utilizes a sigmoid function as its activation function to transform dependent variables into probability values bounded between 0 and 1. This transformation is called the logit transformation, which calculates the logarithm of odds. An odd is the ratio of the probability of an event occurring (presence of a class) to the probability of an event not occurring (absence of a class). The equation for the logistic function goes as follows:

Figure 5 shows the S-shaped curve of logistic regression, which is essentially a [65] sigmoid curve lying between 0 and 1. A decision threshold of 0.5 is set for this graph. Decision threshold is the cut-off probability value that decides which class a data point belongs to. If the probability of the data point is lower than the decision threshold, it belongs to the negative class; otherwise, it is assigned the positive class.

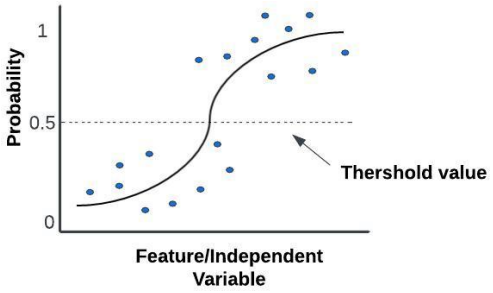


**FIGURE 5.** Sigmoid curve of a binary logistic regression model

To measure the performance of a model, a cost function is required. In logistic regression, a log-loss function (derived from the maximum likelihood estimation method (MLE)) is used to minimize the loss function by adjusting the weights of the model to maximize the likelihood of observing the given data under the assumed logistic regression model. The log-loss function is mathematically defined as:

$$logLoss = -\frac{1}{N}\sum_{i=1} -(y_i \log(P(Y_i)) + (1-y_i)\log(1-P(Y_i))) \quad (5)$$

Here $y_i$ is the actual target variable value, $P(Y_i)$ is the probability of $y_i$ being 1, and $(1 - P(Y_i)$ is the probability of $y_i$ being 0.

Optimization algorithms, such as [66] gradient descent, are commonly used to train logistic regression models. These algorithms iteratively update the regression coefficients to minimize the difference between the predicted probabilities and the actual outcomes. The optimization methods seek to find the best set of coefficients to fit the data and maximize the likelihood. Various performance criteria, such as accuracy, precision, recall, and the receiver operating characteristic (ROC) curve, are used to evaluate model outcomes. These metrics examine the model's ability to classify data points accurately between the positive and negative classes.

In [67] cases, when the dependent variable has three or more categories, multinomial logistic regression can be implemented. It is an extension of binary logistic regression, which aims to classify the dependent variable by modeling the probabilities of each category relative to a chosen reference category. Logit transformation is used independently for each category of the dependent variable. Using MLE, the model estimates several sets of regression coefficients,

$$y_{target} = \frac{1}{1 + e^{-x}}$$

Here $x$ represents the linear combination of predictor variables weighted by regression coefficients.

one for each category. The regression coefficients show the independent variables' effect on the log odds of each category relative to the reference category. Multinomial logistic regression equation is as follows


$$log(\frac{prob(class_z)}{prob(class_{reference})}) = \beta 0+\beta 1x1+\beta 2x2+\beta 3x3....\beta nxn$$

**FIGURE 7.** Structure of a decision tree. Each decision tree contains several sub-trees

Here $log(\frac{prob(class_z)}{prob(class_{reference})})$ is the log-odd ratio of class $z$ relative to the reference class.

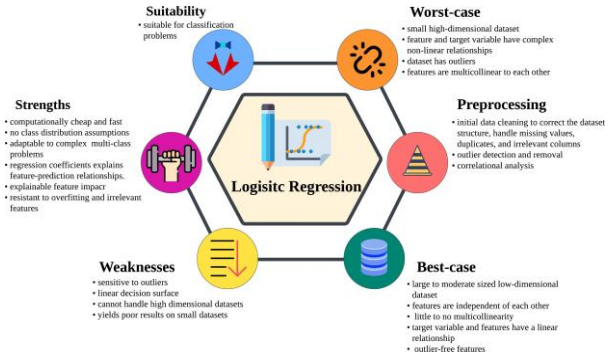


**FIGURE 6. The SWIPES of Logistic Regression.**

Figure 6 describes the SWIPES of a Logistic Regression model. The algorithm is often implemented on [68], [69] medical datasets to predict whether a person will contract a particular disease or not (1 for contracting the disease, 0 for not contracting the disease). To predict [70] whether a customer will churn or not (0 for not churning and 1 for churning), logistic regression can be implemented on features like customer demographics, article usage, and purchase history.

### *DECISION TREES*

Decision trees are graph structures used for classification and regression problems. They are intuitive models that mimic the shape of a tree, with three types of nodes: the root node, internal nodes, and leaf nodes, as shown in Figure 7. Each node corresponds to a specific feature or attribute, and each branch represents a possible outcome of that feature. The top most node is called the root node, while the ending nodes containing the predicted outcomes are called the leaf nodes. All intermediate nodes are called internal nodes that represent decisions based on a feature.

Here $log(\frac{prob(class_z)}{prob(class_{reference})})$ is the log-odd ratio of class $z$ relative to the reference class.

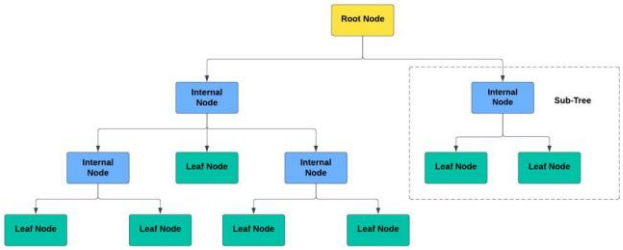


**FIGURE 6. The SWIPES of Logistic Regression.**

Figure 6 describes the SWIPES of a Logistic Regression model. The algorithm is often implemented on [68], [69] medical datasets to predict whether a person will contract a particular disease or not (1 for contracting the disease, 0 for not contracting the disease). To predict [70] whether a customer will churn or not (0 for not churning and 1 for churning), logistic regression can be implemented on features like customer demographics, article usage, and purchase history.

### *DECISION TREES*

Decision trees are graph structures used for classification and regression problems. They are intuitive models that mimic the shape of a tree, with three types of nodes: the root node, internal nodes, and leaf nodes, as shown in Figure 7. Each node corresponds to a specific feature or attribute, and each branch represents a possible outcome of that feature. The top most node is called the root node, while the ending nodes containing the predicted outcomes are called the leaf nodes. All intermediate nodes are called internal nodes that represent decisions based on a feature.

Decision trees work by recursively splitting the data based on the values of distinct features to produce a hierarchical structure of decisions. The algorithm begins by picking the best feature among the available features based on [71], [72] information gain, or Gini impurity. Information gain quantifies the effectiveness of a feature in splitting the data. The higher the information gain of a feature is, the more useful information it contains and hence the higher its position (closer to the root node) in the decision tree will be. Information gain is based on entropy, a concept used to measure impurity in a set of samples.

Entropy [72] is a measure of impurity or disorderliness in a sample set that estimates the uncertainty or randomness associated with the class labels of that sample. The entropy of a sample is inversely proportional to its purity. That is to say that the lower the entropy value, the purer the sample set will be, and vice versa. Mathematically, the entropy of a set $S$ can be written as:

$$Entropy(S) = -\sum_{i=1}^{N} p_i log_2 p_i \quad (7)$$

Here $p_i$ is the proportion of the *ith* feature in *S*.

The reduction in entropy achieved by splitting data along a specific attribute is its information gain. It is the difference between the entropy of the parent node and the weighted entropies of its child nodes after a split. It quantifies the usefulness and relevance of a feature (at the parent node) in predicting class labels. Mathematically,

$$IG(S,f) = E(S) - \sum_{v \in values(F)} \frac{|S_v|}{|S|} E(S_v) \quad (8)$$

Here *f* represents a specific attribute, *E(S)* is the total entropy of dataset *S*, $\frac{|S_v|}{|S|}$ is the proportion of values in $S_v$ to the number of values in *S* and $E(S_v)$ is the entropy of subset $S_v$.

At each split, information gain is recomputed on the newly formed subsets of our original data. The process is repeated recursively until a stopping condition is met. The stopping condition can be a maximum tree depth, a minimum number of samples per leaf, or when all our features are assigned nodes. When this recursive partitioning is complete, every leaf node is assigned a predicted class label or, in case of regression, a predicted value. Predictions for unseen data are made by traversing the tree from a root node to a leaf node following the decision rules that we just constructed.


**FIGURE 8. The SWIPES of Decision Trees.**

Figure 8 describes the SWIPES of Decision Trees. The best thing about this model is that influential features in the decision-making process can be identified by examining the splits and the resulting improvement in prediction. Hence, it's used for several real-world decision-making scenarios. For example, [73] an organization can deploy a decision tree to predict their product demand. They can be used [74] to analyze fraudulent transactions based on credit card data or [75] predict short term water quality metrics such as water temperature, dissolved oxygen, pH value, specific conductance, turbidity, and fluorescent dissolved organic matter.

***RANDOM FOREST***

Random Forest is an ensemble learning methods composed of multiple decision trees. Ensemble learning is a machine learning technique that trains a series of independent ML models known as 'base learners' on the same dataset or its subsets. Base learners are different algorithms or variations of the same algorithm with different parameters. Each base learner derives its own predictions from the data, and the final decision of an ensemble is based on the aggregate of individual base learners. A Random forest algorithm is an ensemble of weak decision trees, as Figure 9 shows. These trees are called weak because on its own, their accuracy is barely better than a random chance

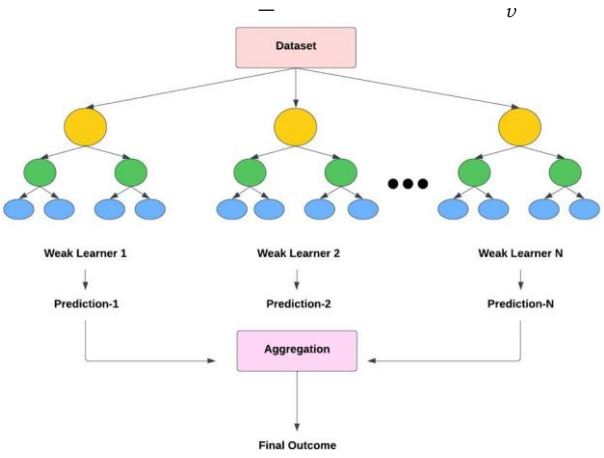


**FIGURE 9. A random forest algorithm comprising of several weak decision trees**

Random Forest is based on [76] bagging, short for bootstrap aggregation. Bootstrap is a sampling technique where subsets/samples of the original training data are selected with replacement. Each decision tree is separately trained on its respective subset, and the final decision of the random forest is an aggregate of individual decisions of the base learners, hence the term aggregation.

Unlike simple decision trees, which consider the entire feature space for splitting at each node, a random forest model only considers a subset of the original feature space, making them resistant to overfitting. After all, trees are trained, the model is ready for prediction. Predictions are made by aggregating the predictions of all decision trees. Different aggregation methods are used, the most common of which are majority voting and weighted voting. In [77] majority voting, the final decision is based on the prediction that receives the maximum number of votes. In [78] weighted voting, the decision of some trees is given more weightage than others. Trees that rely heavily on missing values are given less weightage than those based on no missing values.


**FIGURE 10.** The SWIPES of Random Forest.

Figure 10 describes the SWIPES of a Random Forest algorithm. Random Forests are widely used because of their

high accuracy and the ability to capture complex patterns in the data. They are resistant to overfitting than a simple decision tree because of the random feature selection at each split. This also helps to reduce the variance and correlation among the trees, improving generalization to unseen data. In real life, this ensemble is often implemented to for [79] stock price prediction or to [80] estimate dose-response of a medicine. It can also be used to predict whether a [81] credit card holder will default on his debt, or [82] to identify a patient's disease by analyzing his medical records.

### *NAIVE BAYES CLASSIFIER*

Naive Bayes is a supervised classification technique based on Bayes' theorem and the assumption that features are conditionally independent. That is to say that the algorithm makes a 'naive' assumption that the presence or absence of one feature in a class is independent of another. This helps simplify the computation of probabilities making the algorithm efficient.

Naive Bayes is a probabilistic algorithm that makes predictions based on Bayes theorem. A Bayes theorem calculates the conditional probability of an event, $S$, provided the occurrence of another event, $T$. The posterior probability $P(S|T)$ is the product of the likelihood of $T$ given $S$: $P(T|S)$, and the prior probability of $S$: $P(S)$, divided by the prior probability of $T$: $P(T)$. Mathematically,

$$P(S|T) = \frac{P(T|S)P(S)}{P(T)} \quad (9)$$

$$P(T|S) = \frac{X + a}{Y + na} \quad (10)$$

Here $X$ is the count of the total occurrence of $T$ with condition $S$, and $Y$ is the count of the total occurrence of $S$ in the dataset.

The basic workflow of a Naive Bayes classifier is represented in Figure 11. The algorithm calculates the posterior probability of each class given the input features and then selects the class with the highest probability as the predicted class

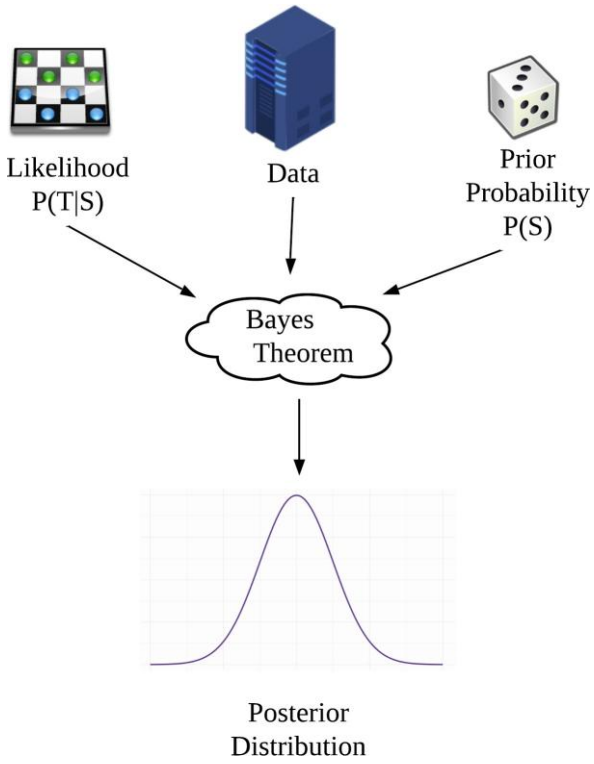


**FIGURE 11. The workflow behind a Naive Bayes Classifier [24]**

To deal with the problem of zero probabilities and avoid potential difficulties when handling feature combinations that were not present in the training data, a small positive value, $\alpha$, is added while calculating likelihood, $P(T|S)$. The intuition behind adding $\alpha$ is called smoothing, or more precisely, [83] Laplace smoothing. The fraction of likelihood now becomes Smoothing is required when a feature value is observed in the test data but not in the training data. In such instances, the probability estimation for that feature would be 0, resulting in a zero probability for the entire class. This can lead to the Naive Bayes classifier becoming overly confident and as- signing zero probabilities to certain classes causing incorrect predictions. Hence, by adding a smoothing coefficient, a, the algorithm ensures that even if a feature value is unseen in the training data, it still has a non-zero probability in the posterior calculation

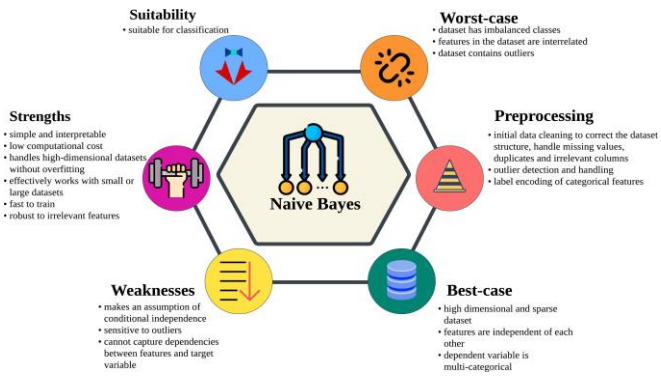


**FIGURE 12.** The SWIPES of Naive Bayes.

Figure 12 describes the SWIPES of a Naive Bayes classifier. Text classification is one of the most popular applications of this algorithm, specifically for [85] identifying spam emails. Be it sentiment analysis of [86] medical data, or [87] hotel reviews, [88] article categorization, or [89] fake news classification, Naive Bayes is popular among NLP tasks because of its ability to handle high-dimensional datasets.

***SUPPORT VECTOR MACHINES***

Support Vector Machines (SVMs) are supervised machine learning algorithms commonly used for classification problems (can be used for regression as well). They work by finding an optimal hyperplane that separates data into their respective classes by maximizing the margin between the hyperplane and support vectors.

Support vectors, hyperplanes, and margins are the key concepts of SVMs. Figure 13 is a pictorial representation of these concepts. A hyperplane is the decision boundary that separates the data points into different classes. It is a subspace of *Y-1* dimensions where *Y* is the total number of dimensions in the feature space. For example, in a 2D feature space, the hyperplane would be 1D (line), while in a 3D feature space, it would be 2D (plane). Support vectors are data points from the training set that lie closest to the hyperplane and influence its determination. They are the focal points for SVMs as the algorithm makes decisions on the basis of them instead of the entire training set. Margins are the distances between the hyperplane and the support vectors. They represent the separation between different classes, and an SVM's objective is to maximize this margin to achieve [90] a well-generalized and robust decision boundary (hyperplane). The larger the margin, the more confident the SVM would be in its classification, as it allows for better separation between classes and reduces the risk of misclassification.

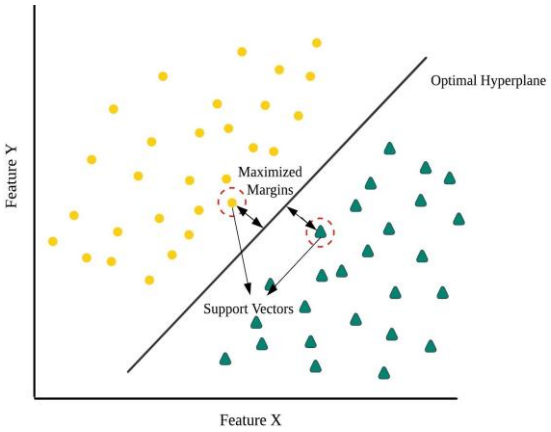


**FIGURE 13.** Hyperplane, support vectors and margin in an SVM.

In some cases, data may not be perfectly separable by a hyperplane. For example, in the presence of outliers, having a strict hyperplane that accurately classifies these outliers may also lead to wrong predictions on unseen data. In these scenarios, we allow our SVM to misclassify some data points. This is called [91] soft margin classification, where the model achieves a trade-off between maximizing the margin and making some errors. The trade-off is controlled by regularization parameters.

Finding a hyperplane is easy in problems where the data is linearly separable. However, when the data is non-linear, we need to introduce a kernel to find an optimal hyperplane. A kernel is a function that converts low-dimensional data into high-dimensional data to find a line of separation between classes. They enable SVMs to capture nonlinear relationships between features without explicitly computing the transformation. Different kinds of kernels are commonly used in SVMs:

Linear Kernel: The simplest and most commonly used kernel in SVMs that performs a [92] linear transforma- tion of the input data, maintaining the original feature space. It is suitable when the data is linearly separable.

Radial Basis Function (RBF) Kernel: Uses a [93] Gaussian-like transformation to assign weights to each data point based on their proximity to a reference point called the kernel center. It can capture complex nonlin- ear decision boundaries and is suitable when there is no prior knowledge about the distribution of data points in the dataset.

Polynomial Kernel: Applies a [94] polynomial transformation to the training data. It allows SVMs to cap- ture nonlinear relationships by introducing higher-order polynomial terms. The degree parameter of the polyno- mial kernel determines the complexity of the decision boundary.

Sigmoid Kernel: Applies a sigmoid function to the input data, allowing SVMs to capture nonlinear relationships similar to artificial neural networks. It is often used in binary classification problems but [95] may not perform as well as other kernel functions in many scenarios.

Nevertheless, the kernel choice depends on the dataset's characteristics and the problem at hand.

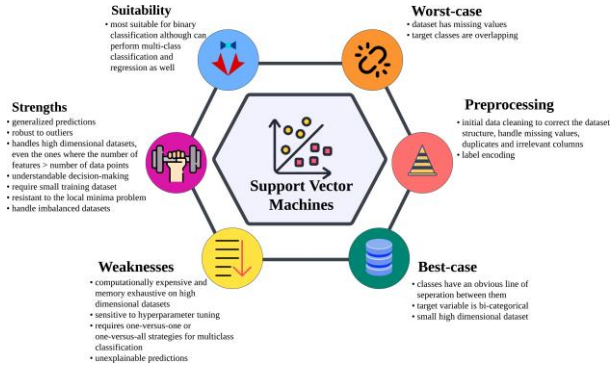


**FIGURE 14. The SWIPES of Support Vector Machines.**

Figure 14 describes the SWIPES [90] of an SVM. SVMs are particularly useful in several image classification and localization use cases, such as [96] neuroimaging analysis.

In bioinformatics, they are sometimes deployed for [97] cancer classification and [98] protein fold recognition. The algorithm also yields good results for [99] non-English handwritten digit recognition and [100] anomaly detection.

### *K NEAREST NEIGHBOUR ALGORITHM*

K-nearest neighbor is a non-parametric machine learning algorithm for regression and classification problems. Given an unseen data point, the model makes predictions by finding K training points closest to it in the feature dimension. These points are called its nearest neighbors. For classification problems, the unseen data point is assigned the class of the majority of its k nearest neighbors, and for regression, the average value of its k nearest neighbors is used.

Figure 15 represents the intuition behind the KNN algorithm where K=4. Different distance metrics are used to find the distance between the unseen data point and the training samples. Euclidean distance, Manhattan distance, and Cosine similarity are the most common. The list below briefly dis- cusses each of these measures.

Euclidean Distance: The square root of the sum of the squared differences between two coordinates. It repre- sents the length of the straight-line path connecting the two points in the [101] Euclidean space. For two points $X$ and $Y$, in a multidimensional space with coordinates $(x_1, y_1, \ldots n_1)$ and $(x_2, y_2, \ldots n_2)$, the Euclidean distance will be

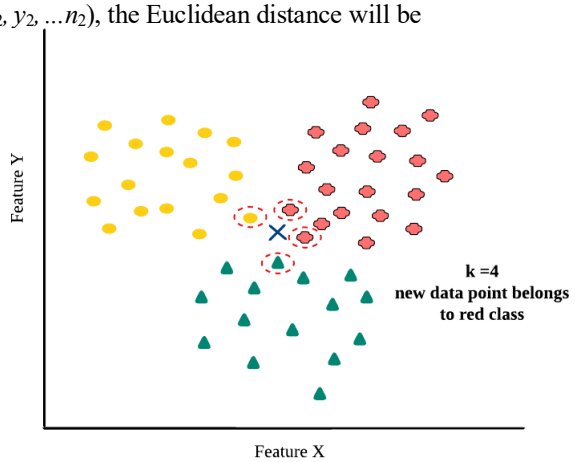


**FIGURE 15. A KNN model with K = 4 to solve a multiclass classification problem**

Manhattan Distance: Also known as the L1 distance, the [102] Manhattan distance calculates the distance between two points in a multidimensional space. Unlike Euclidean distance, which calculates the straight-line distance, Manhattan distance considers only the hori- zontal and vertical movements along axes to measure the distance. It is the sum of the absolute differences between the corresponding coordinates of two points. For two points $X$ and $Y$, with coordinates $(x_1, y_1, \ldots n_1)$ and $(x_2, y_2, \ldots n_2)$, the Manhattan distance is calculated as:

$$d = |x_2 - x_1| + |y_2 - y_1| + \ldots + |n_2 - n_1| \quad (12)$$

Cosine Similarity: Calculates similarity between two vectors by measuring the [103] cosine of the angle between them. Cosine similarity ranges from -1 to 1. A cosine similarity of 1 indicates that the vectors are in the same direction and are perfectly similar. A cosine similarity of -1 indicates that the vectors are in opposite directions and perfectly dissimilar. A cosine similarity of 0 indicates that the vectors have no similarity. For two points $X$ and $Y$, with coordinates $(x_1, y_1, z_1 \ldots n_1)$ and $(x_2, y_2, z_2, \ldots n_2)$, the cosine similarity is calculated as:

$$cosine - similarity = \frac{XY}{||X|| * ||Y||} \quad (13)$$

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + \ldots (n_2 - n_1)^2}$$

The choice of metric heavily influences the model's results; therefore, it is important to carefully analyze the nature of the dataset before making a suitable choice. Based on this distance, k nearest neighbors are selected, where k is a user-defined value representing the number of neighbors to choose.

K value should be carefully determined because having a small k value can lead to overfitting and noise sensitivity

while having a large k value can oversimplify the algorithm and lead to reduced accuracy. Usually, an [104] optimal k value is determined by finding the square root of the total number of samples in our training dataset. Mathematically:

$$k = \sqrt{N} \quad (14)$$

Where N is the total number of samples in the training dataset.

Since KNN relies on distance calculations, it is important to normalize features before training. This ensures that no single feature dominates the distance calculations due to its scale.

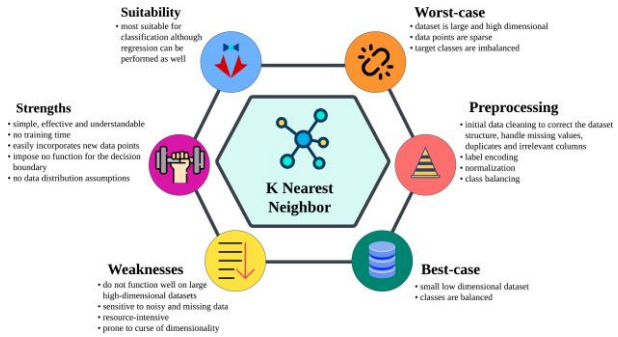


**FIGURE 16.** The SWIPES of K Nearest Neighbor algorithm.

Figure 16 describes the SWIPES of a KNN model. The algorithm requires no training time because it does not explicitly learn a model but stores the training instances in memory. K-Nearest Neighbors (KNN) are deployed for a wide range of use cases. They are used for several recognition

tasks such as [105] gait phase detection, [106] long-term thermal load prediction, and [107] water quality classification. In [108] recommendation systems, the model is used to suggest products to users based on their similarity to other users. Then, in the healthcare sector, KNN can be used for disease diagnosis such as the [109] diagnosis of Parkinson's disease among elderly people. In Geographical data analysis, KNN can be used for [110] route distance optimization and
[111] traffic flow prediction.

### *K-MEANS CLUSTERING*

K-means clustering is an unsupervised learning algorithm that works by grouping similar data points together and dissimilar data points in separate clusters. It is used when there is a large amount of unlabelled data. K is a user-defined value that denotes the number of clusters to be created for a dataset.

The algorithm starts by taking a user-defined k and randomly initializing k centroids. Centroids are the center points of a cluster, and each data point is assigned to its nearest centroid based on distance. Different metrics are used for distance calculation between a data point and a centroid, the most common of which is the [112] Euclidean distance. For a data point, $d$, in feature space with clusters $c_1, c_2, c_3, \ldots c_k$, $d$ will be assigned to the cluster that has the least distance from it as shown in Equation 15.

$$d_{cluster} = min(dist_{c}1\ , dist_{c}2\ , dist_{c}3\ \ldots dist_{c}n) \quad (15)$$

Once all data points are assigned to clusters, the mean of each cluster's data points is computed, becoming the updated centroid. Cluster assignment and centroid updating process is repeated iteratively until there is no significant change in the assignment of clusters or the centroid value. Final clustering is obtained where each data point belongs to a specific cluster based on the nearest centroid. Figure 17 shows how a k- means model with K=2 groups an unlabelled dataset.

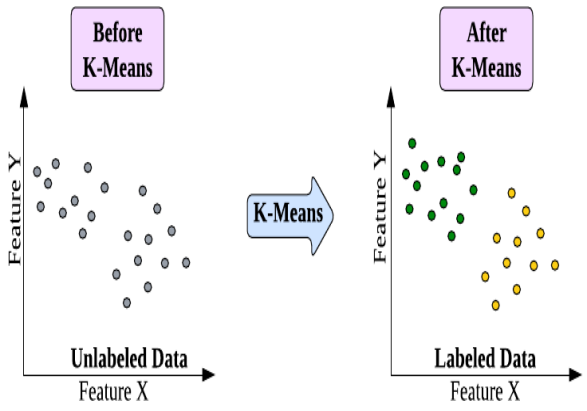


**FIGURE 17. A K-means model with K = 2**

The k value is usually determined using an [113] elbow graph. The elbow graph runs k-means clustering on a range of values starting from k=1 and going up to a suitable point. It plots the sum of the squared distance between the data point and centroid, against k values in the predefined range. The point where the rate of decrease in the mean distance (sum of squared distance) decreases is the elbow point and usually the optimal k value. Figure 18, for example, shows an elbow graph with an elbow at K=3, indicating it to be a suitable k value for the model it is plotted for

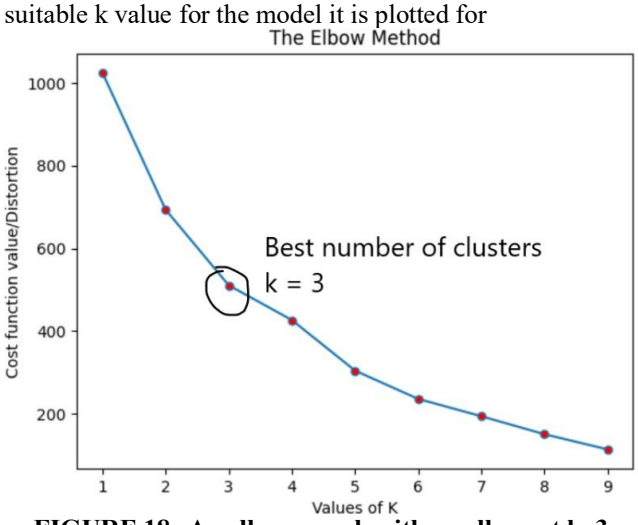


**FIGURE 18. An elbow graph with an elbow at k=3**

Anyhow, the resulting clusters from a k-means model are easily interpretable. Each cluster is represented by its centroid, which makes it simple to understand each cluster's characteristics and central tendencies

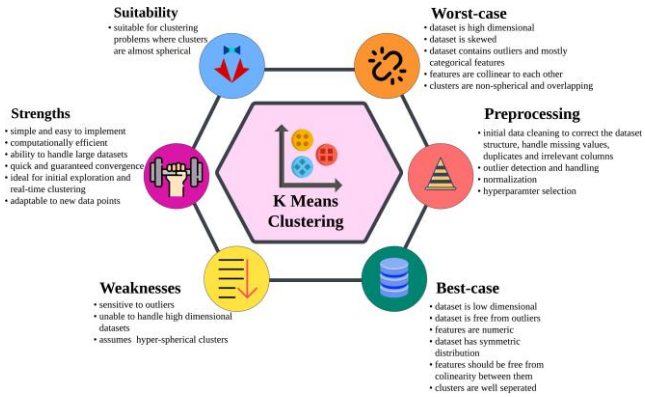


**FIGURE 19. The SWIPES of K-means clustering.**

Figure 19 describes the SWIPES of a K-means clustering model. The algorithm is widely used in several use cases; for example, automatic performance evaluation systems like the one developed by [114] use K-means clustering to score college counselors. In diagnostic systems, k-means is used to create smart medical decision support systems, such as the ones used to [115] treat liver ailments. In the traffic sector, it can be used for [116] vehicle license plate detection and

segmentation, and in NLP, it can be used for [117] document clustering.

### *GAUSSIAN MIXTURE MODEL*

Gaussian Mixture Models (GMMs) are probabilistic models that cluster a dataset on the assumption that each data point is generated from a Gaussian distribution. They are advanced clustering algorithms used when a dataset has non-circular clusters. A Gaussian mixture model is initialized when the number of clusters k, their mean, covariance matrix, and mixing coefficients are specified. Mixing coefficients determine the relative contribution of each Gaussian component to the mixture.

The expectation-maximization (EM) algorithm [118] is used to optimize the model's parameters iteratively. An EM algorithm consists of two steps: Expectation step (E): In this step, the algorithm uses the current parameter estimates to compute the posterior probabilities for each data point belonging to each Gaussian mixture. Maximization step (M): In this step, the algorithm updates the parameters: mean, covariance matrices, and mixing coefficients, based on the computed posterior probabilities. Hence, data points with a higher probability of belonging to that Gaussian mixture will contribute more. Expectation and Maximization steps are iteratively repeated until a point of convergence is achieved. Convergence occurs when the stopping criteria is met: when a log-likelihood change threshold or a maximum number of iterations is achieved.

After the model has converged, data points are assigned clusters with the highest probability. A probability density function (PDF) can be used to quantify the likelihood of witnessing a data point given the GMM's parameters. It describes the shape of each component's distribution in the mixture model. A GMM's PDF is a weighted sum of the individual Gaussian distributions, with each Gaussian component contributing to the overall distribution according to its weight. Mathematically

$$P(s|\theta) = \sum_{i} w_i \times N(s|\mu_i, \sum_{i}) \qquad (16)$$

Here $\sum_i$ represents the covariance matrix of the $i^{th}$ Gaussian component, $\mu_i$ is its mean vector, and $w_i$ represents its mixing coefficient. $N(s\,\mu_i, \sum_i)$ denotes the probability density function of the $i^{th}$ Gaussian component.

GMMs are also used for density estimation, where the learned parameters are used to calculate the PDF of each cluster. This way, new data points can be generated from the estimated distributions.

Although GMMs are complex, they provide soft cluster assignments where each data point is assigned to a cluster based on probability, hence they can capture complex patterns and shapes in the data. They use a probabilistic framework, enabling them to handle missing values by filling them up with imputed ones. Unlike K-means clustering, GMMs can model clusters of different shapes and sizes. They can capture clusters with varying orientations, variances, and correlation structures. Figure 20 shows GMM's ability to draft differently shaped clusters

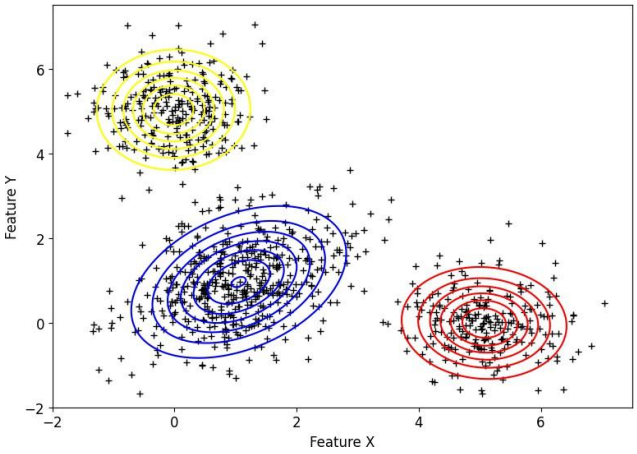


**FIGURE 20.** A GMM with 3 Gaussian mixtures

Gaussian Mixture Models yield maximum performance when the dataset is normally distributed and contains numerical features. While they have the ability to handle non-spherical clusters, GMMs work well only when the clusters are roughly elliptical in shape. If the dataset is too irregularly shaped, the model may not perform as expected.

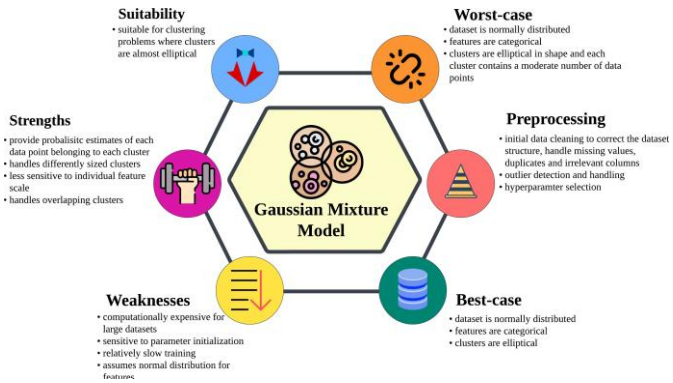


**FIGURE 21.** The SWIPES of the Gaussian Mixture Model.

Figure 21 describes the SWIPES of a Gaussian Mixture Model. In daily life, GMM is used in several applications, including [119] time-series analysis, [120] density estimation, and [121] anomaly detection.

### *PRINCIPAL COMPONENT ANALYSIS*

Principal Component Analysis (PCA) is a dimensionality reduction technique influenced by the intuition of clustering. It is used to scale a dataset from a high-dimensional feature space to a low-dimensional feature space in such a way that its maximum variation is preserved. Hence the main objective of the algorithm is to find features/dimensions that capture maximum information about the dataset. These features are called principal components (PCs).

To find principal components, the dataset is first normalized to bring all features to the same scale and ensure that each variable contributes its fair share. Next, an $n \quad n$ covariance matrix is calculated to capture all the possible covariances in the feature space. A covariance matrix tells the estimated variance in individual random variables and whether they are correlated or not. The matrix is then decomposed to obtain the eigenvalues and eigenvectors. Each eigenvector represents a PC, and its eigenvalue denotes the amount of variance explained by that PC. Eigenvectors are arranged in a descending order according to their eigenvalues to get a list of all principal components. First $k$ PCs are selected based on the desired level of variance retention. The greater the value of $k$, the more the variation is captured. However, choosing a too large k value defeats the purpose of applying PCA; hence k must be selected with care. Lastly, data is reoriented from its original axis to the axis of the first principal component. The selected PCs are projected to the new feature dimension.

Figure 22 visually clarifies the concept behind PCA. PC1 and PC2 were chosen based on the data variability along these axes. PC1 captures the maximum spread of data, and PC2 captures the second-highest data spread.

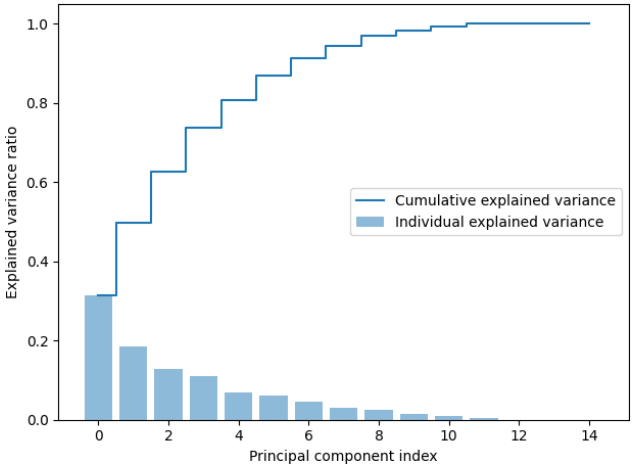


**FIGURE 22.** How PCA reduces a high dimensional feature space to 2 principal components

The number of principal components to choose depends on many factors but, most importantly, better judgment of the problem, the dataset, the available resources, and the requirements. Below are a few factors influencing the number of PC to choose:

Explained Variance: Every PC has an eigenvalue representing the amount of variance explained by that component. Plotting a cumulative explained variance against the number of principal components helps identify how many PCs to choose. The point where adding more components does not significantly increase the variance is generally considered the ideal number of PCs. Look- ing at the cumulative and individual explained variance graphs in Figure 23, it can be observed that the curves level out after the first 9 PCs, making it the ideal number of PCs for the model.

Scree Plot: Scree plot for the eigenvalues of all PCs can be plotted in descending order. The "elbow" point where the variance seems to level off can indicate a suitable number of PCs.

Computational Efficiency: The number of components is inversely related to computational efficiency

Application Requirements: Depending on the requirements of an algorithm, the number of PCs can be determined.

Anyhow, finding a balance between dimensionality reduction and information preservation is important. Dropping too many components can lead to information loss while preserving too many can over complicate an algorithm.

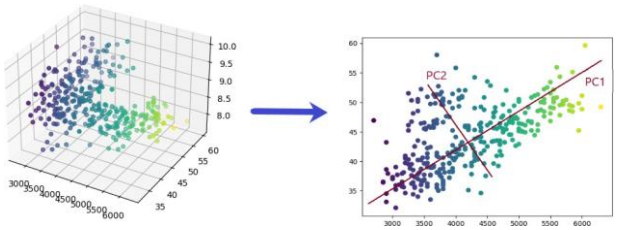


**FIGURE 23.** A plot of explained variance to determine the ideal number of principal components for PCA.

PCA provides easy visualization of the relationship between different features and is robust to overfitting. Because it is a dimensionality reduction method, it is most suited to highly dimensional datasets that may or may not contain redundant and multicollinear features.

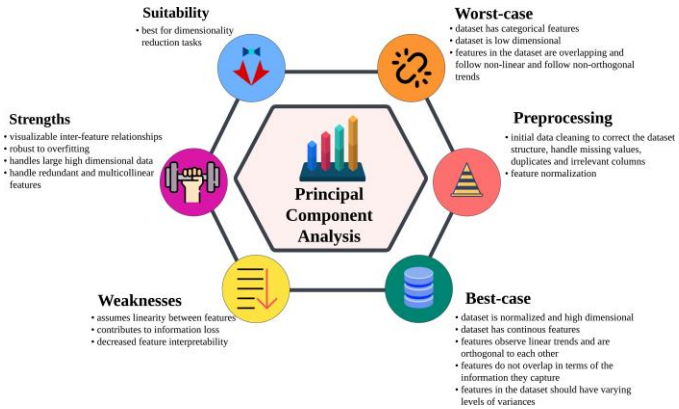


**FIGURE 24.** The SWIPES of Principal Component Analysis.

Figure 24 describes the SWIPES of Principal Component Analysis. The model prefers features in the dataset to have varying levels of variances, as it aims to capture the maximum variance in the data, so having variables with significantly different variances allows the PCs to differentiate and capture the primary sources of variability. The model is used in several applications but mostly during the preprocessing

and exploratory data analysis (EDA) stages and is barely used as a standalone model. It is paired with other machine learning models to speed up their performances. The model finds its applications in [122] feature extraction for image processing or [123] noise reduction from graph neural networks.

**SMART MODEL SELECTION**

Figure 25 presents a flowchart that provides an organized approach to the selection of the best machine learning (ML) algorithm for a given problem and dataset. This methodical decision-making process enables practitioners, and even novices, to make informed decisions that are consistent with the features of their dataset and the requirements of their task. The decision-making process begins by first determining whether or not the dataset is labeled. This divides the problem into two types: supervised learning, which applies when the dataset is labeled, and unsupervised learning, which applies when the dataset is unlabeled.

In the case of labeled data, the ensuing decision is centered on determining feature independence. If features are interdependent, the dataset is best suited for deep neural networks. However, if features are independent of each other, classification and regression algorithms can smoothly tackle the dataset. The choice of classification or regression model then relies on assessing whether the relationship between features and the target variable is linear or not. In instances of linear relationship, Logistic Regression is the best ML model for classification problems, whereas Linear Regression is the best ML model for regression problems.

If, on the other hand, the relationship between features and the target variable is non-linear, and there exists a clear line of separation between different classes of the target, SVM become a favorable choice. Although SVMs can be used for regression problems, they offer more straightforward applicability for classification, particularly binary classification. Conversely, if the line of separation is ambiguous, the next thing to determine is if the dataset is high-dimensional. If it is so, the answer lies between Random Forest classifiers and Naive Bayes, where in the case of sparse data Naive Bayes is the optimal choice, while for abundant datasets, it's better to go with Random Forest Classifiers.

Moving forward, if the dataset is not high-dimensional, decision trees and KNN can both be used. Yet, in this scenario, the final decision can be accredited to the availability of resources. When fast, real-time decision-making and memory efficiency are crucial, decision trees are more suitable than KNNs. This is because KNNs are lazy learners, and instead of learning a distinctive function from the training data in advance, they memorize it and parse the whole dataset at the instance of prediction.

Alternatively, for unlabeled datasets, PCA is recommended for dimensionality reduction. Whereas, when tackling clustering, with spherical and well-separated clusters, K-means clustering is useful. Lastly, when clusters are overlapping and non-spherical, Gaussian Mixture Model should be used

**DISCUSSION**

In this section, we provide a detailed discussion of various machine learning algorithms, focusing on their performance across different datasets and application scenarios. The empirical results presented here are derived from recent studies, offering insights into the practical effectiveness and limitations of each algorithm. For clarity, we present the results in separate tables for each algorithm, summarizing key findings and performance metrics.

***LINEAR REGRESSION***

Linear Regression remains a fundamental algorithm for predicting continuous outcomes. It is a widely used algorithm in various applications for its simplicity and effectiveness in modeling relationships between variables. It is most effective when the underlying relationship between features and target variables is linear. For example, in real estate, it predicts property prices based on features like square footage and location. In sales forecasting, it helps estimate future sales by analyzing historical data and marketing efforts. Similarly, in healthcare, it can forecast patient outcomes from treatment variables. By establishing a linear relationship between inde- pendent variables and a dependent variable, linear regression provides clear insights and predictions, making it an essential tool in fields ranging from economics to education. Table 6 highlights some of the key empirical results of this algorithm.

Linear Regression performed exceptionally well in the above scenarios, demonstrating an $R^2$ of average 0.8. This indicates a strong fit of the model to the data, provided that the data is linear in nature. However, for datasets exhibiting non-linear relationships or significant outliers, the model's performance may degrade.

Also the many other studies suggests that the Linear regression is a widely used predictive model in statistics and machine learning. For example predicting multiple component content in food and is valued for its simplicity and efficiency [130]. However, assessing model assumptions such as linearity, normality, and equal variance is crucial for selecting the best regression model. When these assumptions are violated, strategies like variable transformation or spline models can be employed to improve the model [131]. Multiple Linear Regression (MLR), a generalization of simple linear regression, is commonly used in multivariate statistical analysis. Understanding MLR's basic principles, application examples, conditions, and diagnostics is essential for correct implementation in research [132]. While linear regression is widely applicable, other methods like polynomial regression and fuzzy neural networks may offer better prediction accu- racy in certain scenarios.

***DECISION TREE***

In the regard of diverse application scenarios, decision trees, due to their intuitive structure and ease of interpretation are verstalie tools for both classification and regression tasks, providing clear interpretability. Their primary strength lies in their ability to model decision-making processes by splitting data into distinct branches based on feature values, leading to a tree-like diagram of decisions and their possible conse- quences. For instance, in healthcare, the algorithm can pre- dict patient outcomes by analyzing symptoms and

medical history, making them invaluable for diagnosing diseases and recommending treatments. In finance, they assist in credit scoring by categorizing loan applicants based on their finan- cial history and behavior, thereby simplifying the decision- making process. Additionally, many studies reveal that they are being are employed in marketing to segment customers based on purchasing patterns and preferences, enabling tar- geted advertising strategies. Their capacity to handle both categorical and numerical data, coupled with the ability to visualize decision paths, makes Decision Trees a versatile tool in data science and machine learning. Table 8 highlights some of the key empirical results of this algorithm.

Decision trees have diverse applications in various fields, including bioinformatics, education, and healthcare. In bioinformatics, visually tuned decision trees can achieve good comprehensibility and classification performance, particularly for datasets with binary class attributes and numerous potentially redundant attributes [135]. In higher education, the Decision Tree Wrapper Sampling Class Imbalance Knowledge Assimilation (DT-WSCIKA) method has shown significant improvements in classification metrics and student performance outcomes, aiding in course planning, student advising, and resource allocation [136]. Decision trees are also valuable in healthcare research, such as studying anemia among rural children, where they can identify and rank important factors influencing the condition [137].

***RANDOM FOREST***

Random Forest is like a team of experts working together to make better decisions. Imagine when trying to predict whether a customer will buy a product based on their past behavior. A single decision tree might give the answer, but it might be influenced by a quirk in the data. This algorithm takes multiple decision trees, each looking at different aspects of the data, and combines their predictions. This collective approach makes it more reliable. For instance, in healthcare, a Random Forest model could help predict patient outcomes by integrating various factors like medical history and symptoms, leading to more accurate diagnoses. In finance, it can flag fraudulent transactions by analyzing patterns across many different variables. This method's ability to handle a lot of data and still provide clear insights makes it a go-to for many complex problems. Table **??** showcases how this technique has been successfully used across various fields.

This algorithm has proven effective in various applications, particularly for large datasets. It has been successfully applied to classify acute coronary syndrome cases, achieving 83.45% accuracy with a 70:30 learning scenario [140]. RF's

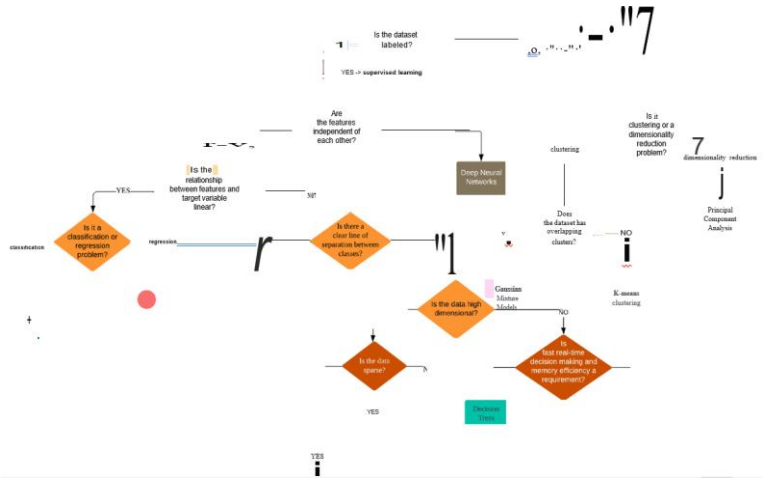


**TABLE 6. Empirical Results for Linear Regression in Different Application Scenarios**

| Study | Dataset | Application Scenario | Performance Metrics | Key Findings |
|---|---|---|---|---|
| [126] | Daily new positive, active, de- ceased, and recovered cases in Odisha and India (March 22, 2020 - July 4, 2020) | Predict daily active COVID-19 cases | **Odisha:** MAE = 73.86, MSE = 11320.16, RMSE = 106.40; $R^2$ = 0.9956 **India**: MAE = 245838.76, MSE = 74851765386.71, RMSE = 273590.51; $R^2$ = 0.9749 | Simple linear regression shows strong predictive ca- pability with high $R^2$ values. Multiple linear regres- sion improves accuracy with $R^2$ values of 0.9985 (Odisha) and 1.0 (India). Forecasts indicate an up- ward trend in active COVID-19 cases. |

| | | | | |
|---|---|---|---|---|
| [127] | Fluoride levels in drinking water, urine, maternal serum, and cord blood from pregnant women | Predict fluoride levels in various biological samples based on drinking water fluoride concentration | $R^2$ = 0.083 to 0.837 for different variables | High fluoride levels in drinking water correlate with higher fluoride levels in urine, maternal serum, and cord blood. Significant statistical differences were found in fluoride concentrations between low/optimum and high fluoride groups. |
| [128] | Time delay data from a space teleobot system, including communication, transmission, and processing delays | Predict time delays in space teleobot systems using modified sparse multivariate linear regression | Performance metrics not explicitly detailed, but modified SMLR showed superior efficiency and precision | The modified SMLR method outperforms traditional SMLR, AR, NN, and CBMB methods in predicting time delays, particularly in complex scenarios with significant jitters. The method demonstrated higher prediction accuracy and efficiency, especially in handling severe time delays in space environments. |
| [129] | Field measurement data from seven water channels of the Zayandeh-Rud irrigation channels in Isfahan province, Iran | Predict seepage from unlined earthen channels using finite element method and multivariable nonlinear regression | **SEEP/W Model:** $R^2$ = 0.879, RMSE = 6.604 **Moritz Method:** $R^2$ = 0.373, RMSE = 12.356 | The SEEP/W numerical model demonstrates high accuracy in predicting seepage rates compared to other methods. Nonlinear regression equations show good performance with $R^2$ values up to 0.930 and RE values as low as -24.18%. |

**TABLE 7. Empirical Results for Decision Tree in Different Application Scenarios**

| Study | Dataset | Application Scenario | Performance Metrics | Key Findings |
|---|---|---|---|---|
| [133] | Z-Alizadeh Sani dataset (303 patients with 55 parameters) | Diagnosis of coronary artery disease (CAD) | Accuracy = 91.09% (with 18 features), TPR = 91.10%, TNV = 91.10% | The CART model with 18 features achieved high accuracy and classification performance, with perfect True Positive Rate (TPR) and True Negative Value (TNV). |
| [134] | DEAP dataset (32 participants, 32 EEG channels) | Emotion recognition from EEG signals | Binary Classification Accuracy = 99.12 ± 0.4814%, Four-class Classification Accuracy = 98.95 ± 0.84%, Eight-class Classification Accuracy = 97.58 ± 2.311% | The Adaptive Neural Decision Tree (ANT) outperforms traditional methods in emotion recognition with high accuracy across various classification tasks and demonstrates efficient parameter use and enhanced interpretability. The model also automates parameter and structure optimization, reducing the need for manual tuning. |

**TABLE 8. Empirical Results for Random Forest in Different Application Scenarios**

| Study | Dataset | Application Scenario | Performance Metrics | Key Findings |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| [138] | Lake Champlain tributaries | Estimating nutrient concentrations and loads | $R^2$ (test) = 0.54 ± 0.07 (TP), RMSE (test) = 0.028 ± 0.003 (TP), % Flux Bias (test) = 1.8 ± 19.9 (TP) | The Random Forest model significantly outperformed WRTDS and its Kalman filtering extension in esti- mating nutrient loads, showing better predictive accuracy and providing useful visualization for process insights. |
| [139] | Dublin, Ireland and New York City | Aerial urban LiDAR point cloud classification | Precision = 0.92, Recall = 0.91, F1-score = 0.92 | The PCVC approach demonstrated strong scalability and improved classification performance compared to previous methods, achieving higher precision, recall, and F1-score while reducing the number of points needing sequential processing by 99%. |

versatility extends to network intrusion detection, email spam filtering, gene classification, credit card fraud detection, and text classification [141]. In benthic habitat mapping using satellite imagery, RF demonstrated consistent performance across different parameter scenarios, with accuracy variations of 1.3% and 5.8% for two islands [142]. Feature selection techniques can enhance RF's performance in medical data analysis. An empirical evaluation of four feature selection methods found that the Boruta algorithm yielded the best results when used with RF classifiers [143].

***SUPPORT VECTOR MACHINE***

As already discussed above SVM involve finding the best possible boundary to separate different groups. Example scenario where may be to sort emails into "spam" and "not spam." SVMs particularly useful when the data isn't easily separable by a simple line, which is often the case in real-world problems. For example, in image recognition, SVM can help identify whether an image contains a cat or a dog by drawing a clear boundary between these classes. Similarly, in finance, SVM can be used to detect unusual patterns that might indicate fraudulent activity. Its strength in handling complex classification tasks makes it a favorite in many analytical scenarios. Table 9 highlights practical successes of SVM in various applications.

Fpr SVM empirical studies have demonstrated that SVM ensembles, particularly bagged ensembles with polynomial kernels, often outperform single SVMs [146]. SVMs exhibit robustness to noise in input datasets, with unweighted combi- nations of data types performing well in simple cases, though

| **Study** | **Dataset** | **Application Scenario** | **Performance Metrics** | **Key Findings** |
|---|---|---|---|---|
| [144] | Gearbox fault data | Intelligent fault identification using PARAFAC and APSO with SVM | Correction Rate = 98%, Running Time = 16.593s | The hybrid PARAFAC-APSO-SVM model achieved the highest fault classification accuracy and retained important features while eliminating redundant information. |
| [145] | Vibration data from induction motors | Fault diagnosis in induction motors using various machine learning models including SVM | SVM Accuracy (Average) = 87.58% to 100% across iterations | The SVM model demonstrated high accuracy in fault diagnosis, achieving up to 100% accuracy in some cases. |

weighted approaches may be superior for multiple noisy datasets [147]. When dealing with imbalanced data, SVMs generally outperform other classifiers like Naive Bayes and decision trees in terms of sensitivity and specificity, especially for large datasets [148]. Comparative studies of SVM optimization criteria reveal that linear programming-based heuristic SVMs achieve similar classification accuracy and generalization performance as quadratic programming-based optimal SVMs, but with fewer support vectors positioned at the furthermost borders of classes [149]. These findings highlight the versatility and effectiveness of SVMs in various classification scenarios.

***K NEAREST NEIGHBOUR***

In the context of a recommendation system, KNN might recommend products or movies based on the preferences of similar users. In healthcare, KNN can predict patient outcomes by comparing new cases with historical cases that have similar characteristics. In healthcare, it can help predict patient outcomes by comparing new cases to similar historical cases. Table 10 provides examples of how KNN has been applied effectively.

Nearest Neighbor (NN) search is a fundamental operation in various domains, but its effectiveness diminishes in high-dimensional spaces. [152] demonstrated that as dimensionality increases, the distance between nearest and farthest neighbors converges, rendering NN search less meaningful. This effect can occur with as few as 10-15 dimensions. [153] conducted a comprehensive evaluation of approximate NN search algorithms, proposing a new method that achieves high efficiency and recall. [154] investigated the difficulty of finding an optimal k value for k-NN classification, concluding that it's generally impossible to determine a priori. [155] compared exact NN algorithms, finding little advantage in using Metric Trees or Cover Trees over KD-Trees for standard NN search. These studies highlight the challenges and limitations of NN techniques, emphasizing the need for careful evaluation and adaptation in high-dimensional spaces.

### *LOGISTIC REGRESSION*

Despite its name, logistic regression is used for classification rather than regression. For instance, in healthcare, logistic regression can predict the likelihood of a patient developing a particular disease based on risk factors such as age, gender, and lifestyle. In finance, it is used to assess the probability of a customer defaulting on a loan based on their credit history and other financial indicators. In marketing, logistic regression helps in predicting whether a customer will respond to a marketing campaign based on their past behavior and demographic data. Table 11 highlights some key examples and results of logistic regression in different applications, showcasing its effectiveness in solving classification problems.

As already discussed above, Logistic regression is a widely used for analyzing binary outcomes in various fields, including education and finance [158], [159]. While it offers advantages over multiple regression for certain types of data, recent research has highlighted limitations in its application and interpretation. Many studies fail to properly report and interpret effect sizes, often misinterpreting odds ratios as relative risk [158]. In high-dimensional settings, classical maximum-likelihood theory can lead to unreliable inferences, with biased estimates and incorrect distribution of the likelihood-ratio test [160]. To improve the reliability and interpretability of logistic regression results, researchers recommend reporting marginal effects and predicted probabilities [158], using logistic regression alongside multivariate analyses for verification [159], and employing new theoretical approaches to account for high-dimensional data [160]. Additionally, guidelines for modeling strategies and reporting standards have been proposed to enhance the quality of logistic regression analyses [161].

### *NAIVE BAYES*

Naive Bayes works well when you need a straightforward, yet effective approach to classification. Imagine when setting up a system to filter out spam emails. Naive Bayes uses the assumption that each word in an email contributes independently to its spam likelihood, which simplifies the problem while still delivering strong results. In healthcare, it might predict whether a patient has a certain condition based on their symptoms and medical history. Despite its simplic- ity, it's quite effective, especially in text-based applications like sentiment analysis or document categorization. It's this straightforward yet powerful approach that makes Naive Bayes a handy tool in various real-world scenarios. Table 12 illustrates some real-life successes with Naive Bayes.

Naive Bayes classifiers, despite their simplistic independence assumption, often perform well in practice. Empirical studies have shown that Naive Bayes achieves good classification accuracy, particularly with low-entropy feature distributions [164], [165]. Surprisingly, it performs best in two contrasting scenarios: completely independent features

**TABLE 10. Empirical Results for K-Nearest Neighbors**

| Study | Dataset | Application Scenario | Performance Metrics | Key Findings |
|---|---|---|---|---|
| [150] | FMCG product reviews | Aspect-Based Sentiment Analysis (ABSA) using Fuzzy-Rough Near- est Neighbour (FRNN) methods | Aspect F1 CV = 0.9036, Sentiment F1 CV = 0.7289, Positive Emotions F1 CV = 0.8273, Negative Emotions F1 CV = 0.7025 | The FRNN-OWA and FROVOCO methods achieved high performance in aspect, sentiment, and emotion prediction tasks, with improvements over previous approaches and good alignment with Dutch dataset results. |
| [151] | Satellite images | Classification of satellite images us- ing Kohonen maps and K-Nearest Neighbor (KNN) algorithm | Not provided directly | Kohonen maps combined with KNN offered im- proved classification accuracy and dimensionality re- duction over K-medoid clustering methods, effec- tively categorizing satellite image components into land types. |

**TABLE 11. Empirical Results for Logistic Regression**

| Study | Dataset | Application Scenario | Performance Metrics | Key Findings |
|---|---|---|---|---|
| [156] | Listed companies in Thailand (SET) | Predicting financial performance of publicly listed companies using lo- gistic regression and random forest algorithms | Logistic Regression: Odds Ratios and p-values for predictors; Random For- est: Model performance not detailed | The random forest algorithm outperformed logistic regression in predicting financial performance. Sig- nificant determinants include liquidity, leverage, asset turnover, and IPO funds. Financial performance pre- dictions varied over time, with excessive IPO funds potentially leading to adverse outcomes. |
| [157] | Wearable ECG-monitoring device | Personalized seizure detection us- ing patient-adaptive logistic regres- sion machine learning (LRML) | Sensitivity = 78.2%, False Alarm Rate (FAR) = 0.62/24 h | The patient-adaptive LRML algorithm significantly reduced the FAR by 31% compared to previous meth- ods while maintaining similar sensitivity. Using re- sponders for training the algorithm improved perfor- mance over generic and non-adaptive approaches. |

**TABLE 12. Empirical Results for Naive Bayes**

| Study | Dataset | Application Scenario | Performance Metrics | Key Findings |
|---|---|---|---|---|
| [162] | Tourism reviews for Jimbaran Beach and Kuta Beach | Comparative review analysis using Naïve Bayes classifier | Jimbaran Beach: Accuracy = 82%-89%; Kuta Beach: Accuracy = 66%-79% | The Naïve Bayes classifier demonstrated variable accuracy across different iterations, with Jimbaran Beach achieving higher accuracy compared to Kuta Beach. The results highlight the effectiveness of the model in sentiment analysis and classification tasks. |
| [163] | Geospatial data for the Gomoa Area, Ghana | Mineral prospectivity mapping us- ing SVM and Naïve Bayes classi- fiers | SVM AUC = 0.90; Naïve Bayes AUC = 0.83; Prospective zones: SVM = 181.62 km², NB = 296.02 km² | The SVM-derived model exhibited higher accuracy (AUC = 0.90) compared to the Naïve Bayes model (AUC = 0.83). Both models effectively delineated prospective mineral zones, with the SVM model cov- ering a smaller area but showing superior predictive performance. |

and functionally dependent features [164], [165]. The classifier's accuracy is better predicted by the amount of class information lost due to the independence assumption rather than the degree of feature dependencies [164], [165]. In ranking tasks, Naive Bayes outperforms decision tree algorithms and competes well with more sophisticated models [166]. It achieves optimal ranking for certain problem types, even when its classification is suboptimal [166]. While Naive Bayes demonstrates good accuracy compared to other mod- els, the advantage of its low-complexity inference may not be as significant as previously thought [167].

***K MEANS CLUSTERING***

The goal of K-Means is to minimize the variance within each cluster, thereby grouping similar data points together while separating dissimilar ones. For example, in market segmentation, K-Means can be used to categorize customers into distinct groups based on their purchasing behaviors, allowing companies to target their marketing efforts more effectively. In image compression, K-Means helps in reducing the num- ber of colors in an image by clustering similar colors, thus enabling more efficient storage and processing. In healthcare,
K-Means can cluster patient data to identify common patterns in symptoms or treatment responses. Table 13 showcases various applications and results of K-Means clustering across different domains, highlighting its practical utility and effectiveness.

K-means clustering is a widely studied algorithm in data mining, with various modifications and applications. Empiri- cal studies have compared K-means with its variants, such as Bisecting K-means, Fuzzy C-means, and Genetic K-means, using internal and external validity indices ( [170]. While K-means and Bisecting K-means show similar performance, Fuzzy C-means and Genetic K-means often outperform the standard K-means algorithm [170], [171]. The choice of dis- tance/similarity metric in K-means can significantly impact its accuracy, performance, and reliability [172]. The behavior of K-means is also influenced by the dataset characteristics, with well-isolated clusters resulting in more local minima compared to overlapping clusters [173]. These empirical evaluations provide valuable insights into the strengths and limitations of K-means and its variants, guiding researchers and practitioners in selecting appropriate clustering algo- rithms for specific applications

| Study | Dataset | Application Scenario | Performance Metrics | Key Findings |
|---|---|---|---|---|
| [168] | Grape leaf images | Diagnosis of grape leaf diseases using K-means clustering and SVM | SVM Accuracy: GLCM = 98.71%, PCA = 98.97%; CNN = 86.82%, Google Net = 94.05% | The proposed method using SVM achieved high accuracy (98.97% with PCA) and was faster compared to deep learning methods. K-means clustering effectively separated disease areas for feature extraction. |
| [169] | Home health care routing | Multi-objective evolutionary approach for routing and scheduling | NSGA-II, SPEA2, Hybrid with K-means | The hybrid approach with K-means improved Pareto set quality for balancing service time and tardiness. |

### *GAUSSIAN MIXTURE MODEL*

GMMs are particularly useful for identifying subpopulations within a dataset, capturing complex patterns, and providing flexible and probabilistic clustering solutions. For example, in customer segmentation, GMMs can classify customers into distinct groups based on purchasing behavior, allowing businesses to tailor marketing strategies more effectively. In image processing, GMMs can segment different objects or regions within an image by modeling the pixel intensity distributions. In finance, GMMs are employed to model asset returns and identify underlying market regimes or anomalies. Table 14 provides examples and results of how GMMs have been effectively applied across various fields, demonstrating their versatility and capability in handling complex data modeling tasks.

GMMs have been widely applied in various domains, particularly for skin color modeling and large-scale data analysis. In skin color detection, GMMs outperform single Gaussian models for medium to high true positive rates, although all models perform similarly at low false positive rates [176]. GMMs have also been used effectively for skin pixel learning across different ethnicities [177]. For large-scale applications, coresets can significantly reduce training time for GMMs while maintaining accuracy. These coresets have size polynomial in dimension and number of mixture components, independent of dataset size [178]. In-memory analysis of GMMs using shared-nothing relational data management systems like Myria has shown promising results, performing up to an order of magnitude faster than Hadoop for large astronomy and oceanography datasets [179]. These studies demonstrate the versatility and efficiency of GMMs in handling diverse and large-scale data analysis tasks.

### *PRINCIPAL COMPONENT ANALYSIS*

PCA is widely used in various applications due to its effectiveness in reducing the number of features while retaining the essential structure of the data. It is particularly valuable when dealing with high-dimensional data, as it helps in uncovering the underlying patterns and reducing computational complexity. For instance, in image processing, PCA can compress image data by capturing the most significant features, allowing for more efficient storage and processing. In finance, PCA is used to analyze and reduce the complexity of financial portfolios by identifying the principal components that explain the majority of the variance in asset returns. In genomics, PCA helps in identifying patterns in gene expression data, which can be critical for understanding genetic variations and their impact on diseases. Table 15 illustrates some of the notable applications and results of PCA in various domains.

PCA empirical studies have shown that PCA can efficiently decompose and represent radiated emissions from circuits, achieving significant efficiency savings in representing emitted radiation [182]. However, researchers caution that PCA results can differ substantially from those of common factor analysis and maximum likelihood factor analysis, particularly in the magnitudes and signs of factor loadings, and in factor interpretation [183]. Despite these differences, patterns produced by PCA, image component analysis, and factor analysis have been found to be remarkably similar across multiple datasets [184]. When applying PCA, it is crucial to consider proper scaling of variables, outlier de- tection, and determination of significant components through methods like cross-validation [185]. These empirical findings highlight both the utility and potential limitations of PCA in various research contexts.

## FUTURE WORK

A solid understanding of numerous machine learning techniques and their applications has been established by our current study. To improve the effectiveness and application of these algorithms, however, a number of important areas need to be further investigated and refined as this field develops. This section describes our future work plans, which centre on expanding into the field of deep learning and resolving the limits found in our study.

***OPTIMIZING MACHINE LEARNING MODELS***

Despite the insights gained from our current research, there remain several areas within machine learning where improvements can be made. Our future work will focus on the following aspects:

Feature Engineering

A crucial aspect of machine learning that has a big impact on model performance is feature engineering. To improve model accuracy and interpretability, we intend to explore more deeply into sophisticated feature engineering approaches in our upcoming study. This entails investigating dimensionality reduction strategies, automated feature selection methodolo- gies, and feature extraction algorithms. Through the applica- tion of cutting-edge feature engineering techniques combined with domain-specific knowledge, our goal is to identify the most informative features that can improve model performance.

Regularization Techniques

In order to reduce overfitting and increase the generalisability of machine learning models, regularisation techniques are crucial. We plan to look into a number of regularisation strategies, including ensemble approaches, dropout, and L1 and L2 regularisation. Evaluating these methods' effects on model stability and robustness is our main objective, especially when dealing with high-dimensional data. We will also investigate new regularisation techniques that may improve the resilience and performance of the model even more.

**TABLE 14. Empirical Results for Gaussian Mixture Model**

| Study | Dataset | Application Scenario | Performance Metrics | Key Findings |
|---|---|---|---|---|
| [174] | Generalized Category Discovery | EM-like framework for representation learning and class number estimation | Semi-supervised GMM with prototypical contrastive learning | Achieved state-of-the-art performance in both generic and fine-grained image datasets, with high accuracy across various settings. |
| [175] | Intrusion Detection | Stacked Sparse Autoencoder and Improved Gaussian Mixture Model | Non-linear dimensionality reduction with stacked sparse autoencoder; joint optimization with improved Gaussian mixture model | SIGMOD outperforms traditional methods on UNSW-NB15 dataset with significant improvements in accuracy and robustness. |

**TABLE 15. Empirical Results for Principal Component Analysis**

| Study | Dataset | Application Scenario | Performance Metrics | Key Findings |
|---|---|---|---|---|
| [180] | Structured Illumination Microscopy (SIM) | Principal Component Analysis (PCA-SIM) | Efficient and robust algorithm for super-resolution imaging; non-iterative, accurate parameter estimation with low SNR | PCA-SIM achieves high-speed, artifact-free super-resolution imaging of live cells with accuracy below 0.01-pixel wave vector and 0.1° relative phase. |
| [181] | Spatial Distribution - Principal Component Analysis (SD-PCA) | A model integrating spatial attributes with PCA to assess soil pollution | Combines spatial distribution and linear transformation for effective pollution assessment; identifies sources of heavy metals in urban areas | Achieves efficient identification of pollution sources with agriculture being the largest contributor (65.5%). Shows varying degrees of contamination and correlations among metals like Cr and Mn/Cu. |

Normalization and Standardization
When preparing data for machine learning models, standardisation and normalisation are essential steps. Subsequent investigations will concentrate on assessing various normali- sation methodologies, such as robust scaling, z-score normal- isation, and min-max scaling. We will evaluate how well they mitigate problems with scale, outliers, and data distribution. Our goal is to increase the convergence speed and overall performance of machine learning algorithms by optimising data preprocessing stages.

Hyperparameter Tuning
Hyperparameter tuning is a crucial aspect of optimizing machine learning models. We plan to explore advanced hyperparameter optimization techniques, such as grid search, random search, Bayesian optimization, and evolutionary algorithms. By systematically evaluating different hyperparameter con- figurations, we aim to identify the optimal settings for various models and enhance their predictive accuracy. Additionally, we will investigate the trade-offs between computational
efficiency and model performance in the context of hyperparameter tuning.

## *ADVANCEMENTS IN DEEP LEARNING*

Deep learning is becoming a more potent paradigm in machine learning that has the ability to solve challenging real-world issues. We plan to investigate the following areas of deep learning in the future:

1) Exploring Popular Deep Neural Networks
Convolutional neural networks , recurrent neural networks , and transformer models are among the deep neural networks that we intend to study. We seek to assess each of these archi- tectures' performance in various scenarios, as they each have unique applications and capabilities. We will offer important insights into these models' applicability for particular tasks and domains by evaluating their advantages and disadvan- tages.

## *INTEGRATING MACHINE LEARNING AND DEEP LEARNING APPROACHES*

Future research will also explore the integration of machine learning and deep learning techniques. Combining these approaches can leverage the strengths of both paradigms and address their individual limitations. We will investigate hybrid models that incorporate machine learning algorithms for feature extraction and deep learning models for advanced pattern recognition. This integrative approach has the potential to enhance predictive performance and provide more comprehensive solutions to complex problems [186].

## *ETHICAL CONSIDERATIONS AND RESPONSIBLE AI*

As machine learning and deep learning technologies become increasingly prevalent, ethical considerations and responsible AI practices are of paramount importance. Our future research will address issues related to fairness, transparency, and accountability in AI systems. We will explore methods for mitigating biases, ensuring data privacy, and promoting ethical use of technology. By incorporating these principles into our research, we aim to contribute to the development of AI systems that are both effective and socially responsible.

## CONCLUSION

The area of machine learning is a broad one, where each problem presents its share of obstacles and opportunities. While every machine learning model has its set of SWIPES, there is no universal algorithm that's a one-for-all and can manage every case effortlessly. Instead, achieving the best results in machine learning depends as much on your better judgment as on a model's strengths.
As we progress in the field of ML, it's critical to remember that experience and judgment are still necessary companions to the algorithms we use. The research offered here provides practitioners with the knowledge they need to make informed decisions, increasing the efficiency and efficacy of their ML efforts. The presented explication algorithms, their SWIPES, and the final strategic blueprint are useful tools for unlocking the full potential of ML models and navigating the convo- luted path to success in an ever-changing profession.
Our research serves as a compass in this treacherous landscape, seeking to help your decision-making process. We have provided elaborate information about the intuition and SWIPES of 10 popular machine learning models and developed a pioneering flowchart to aid in the selection of the best model for a specific dataset and problem. This flowchart, we believe, is unrivaled in its capacity to simplify and speed the model selection process.

Appendixes, if needed, appear before the acknowledgment

## REFERENCE

1. Janiesch, C., Zschech, P. and Heinrich, K., 2021. Machine learning and deep learning. Electronic Markets, 31(3), pp.685-695.
2. Wang, D., Wang, X. and Lv, S., 2019. An overview of end-to-end auto- matic speech recognition. Symmetry, 11(8), p.1018.
3. Priyadarshini, I., Chatterjee, J.M., Sujatha, R., Jhanjhi, N., Karime, A. and Masud, M., 2022. Exploring internet meme activity during COVID- 19 lockdown using Artificial Intelligence techniques. Applied Artificial Intelligence, 36(1), p.2014218.
4. Gouda, W., Sama, N.U., Al-Waakid, G., Humayun, M. and Jhanjhi, N.Z., 2022, June. Detection of skin cancer based on skin lesion images using deep learning. In Healthcare (Vol. 10, No. 7, p. 1183). MDPI.
5. Wahyutama, A.B. and Hwang, M., 2022. Auto-Scoring Feature Based on Sentence Transformer Similarity Check with Korean Sentences Spoken by Foreigners. Applied Sciences, 13(1), p.373.

6. Li, J. and Cheng, Y., 2020, August. Design and implementation of voice- controled intelligent fan system based on machine learning. In 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA) (pp. 548-552). IEEE.
7. Zaman, N., Ghazanfar, M.A., Anwar, M., Lee, S.W., Qazi, N., Karimi, A. and Javed, A., 2023. Stock market prediction based on machine learning and social sentiment analysis
8. Gaur, L., Jhanjhi, N.Z., Bakshi, S. and Gupta, P., 2022, February. Analyz- ing Consequences of Artificial Intelligence on Jobs using Topic Modeling and Keyword Extraction. In 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM) (Vol. 2, pp. 435-440). IEEE.
9. Suta, P., Lan, X., Wu, B., Mongkolnam, P. and Chan, J.H., 2020. An overview of machine learning in chatbots. International Journal of Me- chanical Engineering and Robotics Research, 9(4), pp.502-510.
10. Popel, M., Tomkova, M., Tomek, J., Kaiser, Ł., Uszkoreit, J., Bojar,
11. O. and Žabokrtský, Z., 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. Nature communications, 11(1), p.4381.
12. Zaheer, A., Tahir, S., Humayun, M., Almufareh, M.F. and Jhanjhi, N.Z., 2022, November. A novel Machine learning technique for fake smart watches advertisement detection. In 2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS) (pp. 1-5). IEEE.
13. Li, J., Tang, T., Zhao, W.X., Nie, J.Y. and Wen, J.R., 2022. Pre-
14. trained language models for text generation: A survey. arXiv preprint arXiv:2201.05273.
15. Li, Y., Mao, H., Girshick, R. and He, K., 2022, October. Exploring plain vi- sion transformer backbones for object detection. In European Conference on Computer Vision (pp. 280-296). Cham: Springer Nature Switzerland.
16. Sharma, R., Singh, A., Jhanjhi, N.Z., Masud, M., Jaha, E.S. and Verma, S., 2022. Plant Disease Diagnosis and Image Classification Using Deep Learning. Computers, Materials & Continua, 71(2).
17. Adhikari, S., Gangopadhayay, T.K., Pal, S., Akila, D., Humayun, M., Alfayad, M. and Jhanjhi, N.Z., 2023. A Novel Machine Learning-Based Hand Gesture Recognition Using HCI on IoT Assisted Cloud Platform. Computer Systems Science & Engineering, 46(2).
18. Humayun, M., Ashfaq, F., Jhanjhi, N.Z. and Alsadun, M.K., 2022. Traffic management: Multi-scale vehicle detection in varying weather conditions using yolov4 and spatial pyramid pooling network. Electronics, 11(17), p.2748.
19. Hrizi, O., Gasmi, K., Ben Ltaifa, I., Alshammari, H., Karamti, H., Krichen, M., Ben Ammar, L. and Mahmood, M.A., 2022. Tuberculosis disease diagnosis based on an optimized machine learning model. Journal of Healthcare Engineering, 2022.
20. Humayun, M., Khalil, M.I., Almuayqil, S.N. and Jhanjhi, N.Z., 2023. Framework for detecting breast cancer risk presence using deep learning. Electronics, 12(2), p.403.
21. Patel, V. and Shah, M., 2022. Artificial intelligence and machine learning in drug discovery and development. Intelligent Medicine, 2(3), pp.134- 140.
22. Razaque, A., Frej, M.B.H., Bektemyssova, G., Amsaad, F., Almiani, M., Alotaibi, A., Jhanjhi, N.Z., Amanzholova, S. and Alshammari, M., 2022. Credit Card-Not-Present Fraud Detection and Prevention Using Big Data Analytics Algorithms. Applied Sciences, 13(1), p.57.
23. Bussmann, N., Giudici, P., Marinelli, D. and Papenbrock, J., 2021. Ex- plainable machine learning in credit risk management. Computational Economics, 57, pp.203-216.
24. Lei, K., Zhang, B., Li, Y., Yang, M. and Shen, Y., 2020. Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading. Expert Systems with Applications, 140, p.112872.
25. Chiat, G.B., Ahmad, M., Jhanjhi, N.Z. and Malik, Y., 2022. Machine intelligence as a foundation of self-driving automotive (sda) systems. In Empowering Sustainable Industrial 4.0 Systems With Machine Intelligence (pp. 154-173). IGI Global.
26. Abduljabbar, R., Dia, H., Liyanage, S. and Bagloee, S.A., 2019. Appli- cations of artificial intelligence in transport: An overview. Sustainability, 11(1), p.189.
27. Ham, S.W., Cho, J.H., Park, S. and Kim, D.K., 2021. Spatiotemporal demand prediction model for e-scooter sharing services with latent feature and deep learning. Transportation research record, 2675(11), pp.34-43.
28. Sarwar, S., Tahir, S., Humayun, M., Almufareh, M.F., Jhanjhi, N.Z. and Hamid, B., 2022, November. Recommendation of smart devices using collaborative filter approach. In 2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS) (pp. 1-4). IEEE.
29. Cheung, K.W., Kwok, J.T., Law, M.H. and Tsui, K.C., 2003. Mining customer product ratings for personalized marketing. Decision Support Systems, 35(2), pp.231-243
30. Kumar, N.P., Bhaskar, S., Srinidhi, S.P., Shashank, D. and Karanam, S.G., 2022, December. Machine Learning

Based Predictive Analytics For Agriculture Inventory Management System. In 2022 Fourth International Conference on Cognitive Computing and Information Processing (CCIP) (pp. 1-7). IEEE.

31. Jena, K.K., Bhoi, S.K., Malik, T.K., Sahoo, K.S., Jhanjhi, N.Z., Bhatia, S. and Amsaad, F., 2022. E-Learning Course Recommender System Using Collaborative Filtering Models. Electronics, 12(1), p.157.

32. Chang, M., D'Aniello, G., Gaeta, M., Orciuoli, F., Sampson, D. and Si- monelli, C., 2020. Building ontology-driven tutoring models for intelligent tutoring systems using data mining. IEEE Access, 8, pp.48151-48162.

33. Awale, N., Pandey, M., Dulal, A. and Timsina, B., 2020. Plagiarism detection in programming assignments using machine learning. Journal of artificial intelligence and capsule networks, 2(3), pp.177-184.

34. Zahra, F., Jhanjhi, N.Z., Brohi, S.N., Khan, N.A., Masud, M. and AlZain, M.A., 2022. Rank and wormhole attack detection model for RPL-based internet of things using machine learning. Sensors, 22(18), p.6765.

35. Aherwadi, N., Mittal, U., Singla, J., Jhanjhi, N.Z., Yassine, A. and Hossain, M.S., 2022. Prediction of fruit maturity, quality, and its life using deep learning algorithms. Electronics, 11(24), p.4100.

36. Menon, S., Anand, D., Kavita, Verma, S., Kaur, M., Jhanjhi, N.Z., Ghoniem, R.M. and Ray, S.K., 2023. Blockchain and Machine Learning Inspired Secure Smart Home Communication Network. Sensors, 23(13), p.6132.

37. Gupta, M. and Pandya, S.D., 2022. A Comparative Study on Supervised Machine Learning Algorithm. International Journal for Research in Ap- plied Science and Engineering Technology (IJRASET), 10(1), pp.1023-1028.

38. Sarker, I.H., 2021. Machine learning: Algorithms, real-world applications and research directions. SN computer science, 2(3), p.160.

39. Kulin, M., Kazaz, T., De Poorter, E. and Moerman, I., 2021. A survey on machine learning-based performance improvement of wireless networks: PHY, MAC and network layer. Electronics, 10(3), p.318.

40. Al-Turjman, F. and Baali, I., 2022. Machine learning for wearable IoT- based applications: A survey. Transactions on Emerging Telecommunica- tions Technologies, 33(8), p.e3635.

41. Penney, D.D. and Chen, L., 2019. A survey of machine learning applied to computer architecture design. arXiv preprint arXiv:1909.12373.

42. Qayyum, A., Qadir, J., Bilal, M. and Al-Fuqaha, A., 2020. Secure and robust machine learning for healthcare: A survey. IEEE Reviews in Biomedical Engineering, 14, pp.156-180.

43. Balaji, T.K., Annavarapu, C.S.R. and Bablani, A., 2021. Machine learning algorithms for social media analysis: A survey. Computer Science Review, 40, p.100395.

44. Kaur, Randeep. (2021). A Survey on Machine Learning Techniques with Applications.

45. Ray, S., 2019, February. A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 35-39). IEEE.

46. Nagar, R. and Singh, Y., 2019. A literature survey on machine learning algorithms. Journal of Emerging Technologies and Innovative Research, 6(4), pp.471-474.

47. Uddin, S., Khan, A., Hossain, M.E. and Moni, M.A., 2019. Comparing different supervised machine learning algorithms for disease prediction. BMC medical informatics and decision making, 19(1), pp.1-16.

48. Mansour, N.A., Saleh, A.I., Badawy, M. and Ali, H.A., 2022. Accurate detection of Covid-19 patients based on Feature Correlated Naïve Bayes (FCNB) classification strategy. Journal of ambient intelligence and human- ized computing, pp.1-33.

49. Ahmed, A., Jalal, A. and Kim, K., 2020. A novel statistical method for scene classification based on multi-object categorization and logistic regression. Sensors, 20(14), p.3871.

50. Khan, Z., 2022. Used car price evaluation using three different variants of linear regression. International Journal of Computational and Innovative Sciences, 1(1), pp.12-20.

51. Ghiasi, M.M. and Zendehboudi, S., 2021. Application of decision tree- based ensemble learning in the classification of breast cancer. Computers in biology and medicine, 128, p.104089.

52. Pekel, E., 2020. Estimation of soil moisture using decision tree regression. Theoretical and Applied Climatology, 139(3-4), pp.1111-1119.

53. Tzenios, N., 2020. Examining the Impact of EdTech Integration on Academic Performance Using Random Forest Regression. ResearchBerg Review of Science and Technology, 3(1), pp.94-106

54. Liu, K., Hu, X., Zhou, H., Tong, L., Widanage, W.D. and Marco, J., 2021. Feature analyses and modeling of lithium-ion battery manufacturing based on random forest classification. IEEE/ASME Transactions on Mechatron- ics, 26(6), pp.2944-2955.

55. Guo, Z.X. and Shui, P.L., 2020. Anomaly based sea-surface small target detection using K-nearest neighbor classification. IEEE Transactions on Aerospace and Electronic Systems, 56(6), pp.4947-4964.

56. Zhou, Y., Huang, M. and Pecht, M., 2020. Remaining useful life estimation of lithium-ion cells based on k-nearest neighbor regression with differen- tial

evolution optimization. Journal of Cleaner Production, 249, p.119409.
57. Deiss, L., Margenot, A.J., Culman, S.W. and Demyan, M.S., 2020. Tuning support vector machines regression models improves prediction accuracy of soil properties in MIR spectroscopy. Geoderma, 365, p.114227.
58. Khan, M.S., Khan, L., Gul, N., Amir, M., Kim, J. and Kim, S.M., 2020. Support vector machine-based classification of malicious users in cogni- tive radio networks. Wireless Communications and Mobile Computing, 2020, pp.1-11.
59. Sen, P.C., Hajra, M. and Ghosh, M., 2020. Supervised classification algo- rithms in machine learning: A survey and review. In Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018 (pp. 99-111). Springer Singapore.
60. Maulud, D. and Abdulazeez, A.M., 2020. A review on linear regression comprehensive in machine learning. Journal of Applied Science and Technology Trends, 1(4), pp.140-147.
61. Kim, H.Y., 2019. Statistical notes for clinical researchers: simple linear regression 3–residual analysis. Restorative dentistry & endodontics, 44(1).
62. Chicco, D., Warrens, M.J. and Jurman, G., 2021. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science, 7, p.e623.
63. Ray, S., 2019, February. A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 35-39). IEEE.
64. Madhuri, C.R., Anuradha, G. and Pujitha, M.V., 2019, March. House price prediction using regression techniques: A comparative study. In 2019 International conference on smart structures and systems (ICSSS) (pp. 1-5). IEEE.
65. Bajaj, P., Ray, R., Shedge, S., Vidhate, S. and Shardoor, N., 2020. Sales prediction using machine learning algorithms. International Research Jour- nal of Engineering and Technology (IRJET), 7(6), pp.3619-3625.
66. Isola, G., Polizzi, A., Alibrandi, A., Williams, R.C. and Lo Giudice, A., 2021. Analysis of galectin-3 levels as a source of coronary heart disease risk during periodontitis. Journal of Periodontal Research, 56(3), pp.597- 605.
67. Fan, Y., Bai, J., Lei, X., Zhang, Y., Zhang, B., Li, K.C. and Tan, G., 2020. Privacy preserving based logistic regression on big data. Journal of network and computer applications, 171, p.102769.
68. Zou, X., Hu, Y., Tian, Z. and Shen, K., 2019, October. Logistic regression model optimization and case analysis. In 2019 IEEE 7th international conference on computer science and network technology (ICCSNT) (pp. 135-139). IEEE.
69. Bayar, Y., Sezgin, H.F., Öztürk, Ö.F. and S¸as¸maz, M.Ü., 2020. Financial literacy and financial risk tolerance of individual investors: Multinomial logistic regression approach. Sage Open, 10(3), p.2158244020945717.
70. Nusinovici, S., Tham, Y.C., Yan, M.Y.C., Ting, D.S.W., Li, J., Sa- banayagam, C., Wong, T.Y. and Cheng, C.Y., 2020. Logistic regression was as good as machine learning for predicting major chronic diseases. Journal of clinical epidemiology, 122, pp.56-69.
71. Connelly, L., 2020. Logistic regression. Medsurg Nursing, 29(5), pp.353- 354.
72. Jain, H., Khunteta, A. and Srivastava, S., 2020. Churn prediction in telecommunication using logistic regression and logit boost. Procedia Computer Science, 167, pp.101-112.
73. Wagner, H.N.R., Köke, H., Dähne, S., Niemann, S., Hühne, C. and Khakimova, R., 2019. Decision tree-based machine learning to optimize the laminate stacking of composite cylinders for maximum buckling load and minimum imperfection sensitivity. Composite Structures, 220, pp.45- 63.
74. Charbuty, B. and Abdulazeez, A., 2021. Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), pp.20-28.
75. Kharfan, M., Chan, V.W.K. and Firdolas Efendigil, T., 2021. A data-driven forecasting approach for newly launched seasonal products by leveraging
76. machine-learning approaches. Annals of Operations Research, 303(1-2), pp.159-174.
77. Hammed, M. and Soyemi, J., 2020. An implementation of decision tree algorithm augmented with regression analysis for fraud detection in credit card. International Journal of Computer Science and Information Security (IJCSIS), 18(2), pp.79-88.
78. Lu, H. and Ma, X., 2020. Hybrid decision tree-based machine learn- ing models for short-term water quality prediction. Chemosphere, 249, p.126169.
79. Tyralis, H., Papacharalampous, G. and Langousis, A., 2019. A brief review of random forests for water scientists and practitioners and their recent history in water resources. Water, 11(5), p.910.
80. Kim, S.Y. and Upneja, A., 2021. Majority voting ensemble with a decision trees for business failure prediction during economic downturns. Journal of Innovation & Knowledge, 6(2), pp.112-123.
81. Maheshwari, A., Mehraj, B., Khan, M.S. and Idrisi, M.S., 2022. An opti- mized weighted voting based ensemble model for DDoS attack detection and mitigation in SDN environment. Microprocessors and Microsystems,

89, p.104412.
82. Yuan, X., Yuan, J., Jiang, T. and Ain, Q.U., 2020. Integrated long-term stock selection models based on feature selection and machine learning algorithms for China stock market. IEEE Access, 8, pp.22672-22685.
83. Rahman, R., Dhruba, S.R., Ghosh, S. and Pal, R., 2019. Functional random forest with applications in dose-response predictions. Scientific reports, 9(1), p.1628.
84. Schonlau, M. and Zou, R.Y., 2020. The random forest algorithm for statistical learning. The Stata Journal, 20(1), pp.3-29.
85. Jackins, V., Vimal, S., Kaliappan, M. and Lee, M.Y., 2021. AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes. The Journal of Supercomputing, 77, pp.5198-5219.
86. Geetha, S. and Maniyosai, R., 2019. An improved Naive Bayes classifier on imbalanced attributes. International Journal of Organizational and Collective Intelligence (IJOCI), 9(2), pp.1-15.
87. Wickramasinghe, I. and Kalutarage, H., 2021. Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. Soft Computing, 25(3), pp.2277-2293.
88. Jaiswal, M. and Das, S., 2021. Detecting spam e-mails using stop word TF- IDF and stemming algorithm with Naïve Bayes classifier on the multicore GPU. International Journal of Electrical & Computer Engineering (2088- 8708), 11(4).
89. Boyko, N. and Boksho, K., 2020, November. Application of the Naive Bayesian Classifier in Work on Sentimental Analysis of Medical Data. In IDDM (pp. 230-239).
90. Farisi, A.A., Sibaroni, Y. and Al Faraby, S., 2019, March. Sentiment analysis on hotel reviews using Multinomial Naïve Bayes classifier. In Journal of Physics: Conference Series (Vol. 1192, No. 1, p. 012024). IOP Publishing.
91. Abd, D.H., Sadiq, A.T. and Abbas, A.R., 2019, September. Political arti- cles categorization based on different naïve bayes models. In International Conference on Applied Computing to Support Industry: Innovation and Technology (pp. 286-301). Cham: Springer International Publishing.
92. Adiba, F.I., Islam, T., Kaiser, M.S., Mahmud, M. and Rahman, M.A., 2020. Effect of corpora on classification of fake news using naive Bayes classifier. International Journal of Automation, Artificial Intelligence and Machine Learning, 1(1), pp.80-92.
93. Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L. and Lopez, A., 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, pp.189-215.
94. Hekmatmanesh, A., Wu, H., Jamaloo, F., Li, M. and Handroos, H., 2020. A combination of CSP-based method with soft margin SVM classifier and generalized RBF kernel for imagery-based brain computer interface applications. Multimedia Tools and Applications, 79, pp.17521-17549.
95. Tang, X., Ma, Z., Hu, Q. and Tang, W., 2019. A real-time arrhythmia heart- beats classification algorithm using parallel delta modulations and rotated linear-kernel support vector machines. IEEE Transactions on Biomedical Engineering, 67(4), pp.978-986.
96. Gopi, A.P., Jyothi, R.N.S., Narayana, V.L. and Sandeep, K.S., 2023. Classification of tweets data based on polarity using improved RBF kernel of SVM. International Journal of Information Technology, 15(2), pp.965- 980.
97. Gontumukkala, S.S.T., Godavarthi, Y.S.V., Gonugunta, B.R.R.T., Subra- mani, R. and Murali, K., 2021, July. Analysis of Image Classification using SVM. In 2021 12th International Conference on Computing Communica- tion and Networking Technologies (ICCCNT) (pp. 01-06). IEEE
98. Al-Mejibli, I.S., Alwan, J.K. and Abd Dhafar, H., 2020. The effect of gamma value on support vector machine performance with different ker- nels. International Journal of Electrical and Computer Engineering, 10(5), p.5497.
99. Zhou, T., Thung, K.H., Liu, M., Shi, F., Zhang, C. and Shen, D., 2020. Multi-modal latent space inducing ensemble SVM classifier for early dementia diagnosis with neuroimaging data. Medical image analysis, 60, p.101630.
100. Houssein, E.H., Hassan, H.N., Samee, N.A. and Jamjoom, M.M., 2023. A Novel Hybrid Runge Kutta Optimizer with Support Vector Machine on Gene Expression Data for Cancer Classification. Diagnostics, 13(9), p.1621.
101. Yan, K., Wen, J., Liu, J.X., Xu, Y. and Liu, B., 2020. Protein fold recognition by combining support vector machines and pairwise sequence similarity scores. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 18(5), pp.2008-2016.
102. Bouchene, M.M. and Boukharouba, A., 2022. Features extraction and reduction techniques with optimized SVM for Persian/Arabic handwritten digits recognition. Iran Journal of Computer Science, 5(3), pp.247-265.
103. Yang, K., Kpotufe, S. and Feamster, N., 2021. An efficient one-class SVM for anomaly detection in the Internet of Things. arXiv preprint arXiv:2104.11146.
104. Singh, S.A. and Majumder, S., 2019. Classification of unsegmented heart sound recording using KNN classifier. Journal of Mechanics in Medicine and Biology, 19(04), p.1950025.

105. Gao, X. and Li, G., 2020. A KNN model based on manhattan distance to identify the SNARE proteins. Ieee Access, 8, pp.112922-112931.
106. Singh, R.H., Maurya, S., Tripathi, T., Narula, T. and Srivastav, G., 2020. Movie recommendation system using cosine similarity and KNN. Inter- national Journal of Engineering and Advanced Technology, 9(5), pp.556- 559.
107. Chaudhari, P., Agarwal, H. and Bhateja, V., 2021. Data augmentation for cancer classification in oncogenomics: an improved KNN based approach. Evolutionary Intelligence, 14, pp.489-498.
108. Rattanasak, A., Uthansakul, P., Uthansakul, M., Jumphoo, T., Phap- atanaburi, K., Sindhupakorn, B. and Rooppakhun, S., 2022. Real-time gait phase detection using wearable sensors for transtibial prosthesis based on a kNN algorithm. Sensors, 22(11), p.4242.
109. Liang, Y., Pan, Y., Yuan, X., Jia, W. and Huang, Z., 2023. Surrogate mod- eling for long-term and high-resolution prediction of building thermal load with a metric-optimized KNN algorithm. Energy and Built Environment, 4(6), pp.709-724.
110. Nababan, A.A., Khairi, M. and Harahap, B.S., 2022. Implementation of K-Nearest Neighbors (KNN) algorithm in classification of data water quality. Jurnal Mantik, 6(1), pp.30-35.
111. Mensouri, D., Azmani, A. and Azmani, M., 2022, May. Towards an E-commerce Personalized Recommendation System with KNN Classi- fication Method. In International Conference on Advanced Intelligent Systems for Sustainable Development (pp. 364-382). Cham: Springer Nature Switzerland.
112. Fang, Z., 2022, February. Improved KNN algorithm with information entropy for the diagnosis of Parkinson's disease. In 2022 International Conference on Machine Learning and Knowledge Engineering (MLKE) (pp. 98-101). IEEE.
113. Paithane, P., Wagh, S.J. and Kakarwal, S., 2023. Optimization of route distance using k-NN algorithm for on-demand food delivery. System research and information technologies, (1), pp.85-101.
114. Babaei, M. and Behzadi, S., 2023. Spatial Data-Driven Traffic Flow Prediction Using Geographical Information System. Journal of Soft Com- puting in Civil Engineering, 7(4).
115. Hossain, M.Z., Akhtar, M.N., Ahmad, R.B. and Rahman, M., 2019. A dynamic K-means clustering for data mining. Indonesian Journal of Electrical engineering and computer science, 13(2), pp.521-526.
116. Cui, M., 2020. Introduction to the k-means clustering algorithm based on the elbow method. Accounting, Auditing and Finance, 1(1), pp.5-8.
117. Wang, Z. and Tian, Q., 2022. Performance Appraisal and Automatic Scoring System for College Counselors Based on Kmeans Clustering. Mathematical Problems in Engineering, 2022.
118. Singh, A., Mehta, J.C., Anand, D., Nath, P., Pandey, B. and Khamparia, A., 2021. An intelligent hybrid approach for hepatitis disease diagnosis: Combining enhanced k-means clustering and improved ensemble learning. Expert Systems, 38(1), p.e12526
119. Chen, D.J.I.Z., 2021. Automatic vehicle license plate detection using K- means clustering algorithm and CNN. Journal of Electrical Engineering and Automation, 3(1), pp.15-23.
120. Abasi, A.K., Khader, A.T., Al-Betar, M.A., Naim, S., Alyasseri, Z.A.A. and Makhadmeh, S.N., 2020. A novel hybrid multi-verse optimizer with K-means for text documents clustering. Neural Computing and Applica- tions, 32, pp.17703-17729.
121. Ponzi, V., Russo, S., Wajda, A., Brociek, R. and Napoli, C., 2022. Analysis pre and post covid-19 pandemic rorschach test data of using em algorithms and gmm models. In CEUR Workshop Proceedings (Vol. 3360, pp. 55-63).
122. Mozaffari, Z. and Ghaderi, S., 2021. The role of human capital in the industrialization of Iranian economy; Applications of fuzzy logic and GMM in time series.
123. Crawford, A., 2020. The use of Gaussian mixture models with atmo- spheric Lagrangian particle dispersion models for density estimation and feature identification. Atmosphere, 11(12), p.1369.
124. Liu, J., Zhu, H., Liu, Y., Wu, H., Lan, Y. and Zhang, X., 2019, April. Anomaly detection for time series using temporal convolutional networks and Gaussian mixture model. In Journal of Physics: Conference Series (Vol. 1187, No. 4, p. 042111). IOP Publishing.
125. Kumar, D., 2020. Feature extraction and selection of kidney ultrasound images using GLCM and PCA. Procedia Computer Science, 167, pp.1722- 1731.
126. Dong, W., Woz´niak, M., Wu, J., Li, W. and Bai, Z., 2022. Denoising ag- gregation of graph neural networks by using principal component analysis. IEEE Transactions on Industrial Informatics, 19(3), pp.2385-2394.
127. Camastra, F., Capone, V., Ciaramella, A., Riccio, A. and Staiano, A., 2022. Prediction of environmental missing data time series by Support Vector Machine Regression and Correlation Dimension estimation. Envi- ronmental Modelling & Software, 150, p.105343.
128. Alkhayrat, M., Aljnidi, M. and Aljoumaa, K., 2020. A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA. Journal of Big Data, 7, pp.1-23.

129. Rath, S., Tripathy, A., & Tripathy, A. R. (2020). Prediction of new active cases of coronavirus disease (COVID-19) pandemic using multiple linear regression model. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, 14(5), 1467–1474. Elsevier.

130. Thippeswamy, H.M., Kumar, M. Nanditha, Girish, M., Prashanth, S.N., & Shanbhog, R. (2021). Linear regression approach for predicting fluoride concentrations in maternal serum, urine and cord blood of pregnant women consuming fluoride containing drinking water. *Clinical Epidemiology and Global Health*, **10**, 100685. Elsevier.

131. Chen, Haifei, Huang, Panfeng, Liu, Zhengxiong, and Ma, Zhiqiang, *Time delay prediction for space telerobot system with a modified sparse multivariate linear regression method*, Acta Astronautica, vol. 166, pp. 330–341, 2020.

132. Farzin Salmasi and John Abraham, *Predicting seepage from unlined earthen channels using the finite element method and multi variable non- linear regression*, *Agricultural Water Management*, vol. 234, p. 106148, 2020.

133. Xiyu Xie, *Analysis on the Application of Linear Regression in Various Fields*, 2020. Available online: https://api.semanticscholar.org/CorpusID: 222113242.

134. Ningjie Huang, *Scenarios Where Utilizing a Spline Model in Developing a Regression Model Is Appropriate*, 2014. Available online: https://api. semanticscholar.org/CorpusID:4683823.

135. Y. H. Hu, S. C. Yu, X. Qi, W. J. Zheng, Q. Q. Wang, and Hong-yan Yao, *An overview of multiple linear regression model and its applica- tion*, *Zhonghua Yu Fang Yi Xue Za Zhi [Chinese Journal of Preventive Medicine]*, vol. 53, no. 6, pp. 653–656, 2019. Available online: https:

136. //api.semanticscholar.org/CorpusID:182947293.

137. M. M. Ghiasi, S. Zendehboudi, and A. A. Mohsenipour, "Decision tree- based diagnosis of coronary artery disease: CART model," *Computer Methods and Programs in Biomedicine*, vol. 192, p. 105400, 2020.

138. Yongqiang Zheng, Jie Ding, Feng Liu, and Dongqing Wang, "Adaptive neural decision tree for EEG based emotion recognition," *Information Sciences*, vol. 643, pp. 119160, 2023.

139. Gregor Štiglic, Simon Kocbek, Igor Pernek, and Peter Kokol, "Com- prehensive Decision Tree Models in Bioinformatics," *PLoS ONE*, vol. 7, 2012. https://api.semanticscholar.org/CorpusID:18888480

140. Xiaohu He, "Application of Decision Tree Algorithm in Teaching and Learning Management in Colleges and Universities," *Journal of Electrical Systems*, 2024. https://api.semanticscholar.org/CorpusID:269880866

141. Yu-gang Ma, Yu-xue Bi, Hong-jian Yan, Li-na Deng, Wei-feng Liang, Bei rong Wang, and Xue-li Zhang, "The application of decision tree in the research of anemia among rural children under 3-year-old," *Zhonghua Yu Fang Yi Xue Za Zhi [Chinese Journal of Preventive Medicine]*, vol. 43, no. 5, pp. 434-437, 2009. https://api.semanticscholar.org/CorpusID:27830757

142. Isles, Peter D. F. (2024). *A random forest approach to improve estimates of tributary nutrient loading*. *Water Research*, **248**, 120876. Elsevier.

143. Harith Aljumaily, Debra F. Laefer, Dolores Cuadra, and Manuel Velasco, *Point cloud voxel classification of aerial urban LiDAR using voxel attributes and random forest approach*, *International Journal of Applied Earth Observation and Geoinformation*, **118**, 103208 (2023).

144. Eka Pandu Cynthia, Alwis Nazir, and Fadhilah Syafria. *Random Forest Algorithm to Investigate the Case of Acute Coronary Syndrome*. 2021.

145. Available at: https://api.semanticscholar.org/CorpusID:235224488

146. Mohammed Zakariah. *Classification of Large Datasets Using Random Forest Algorithm in Various Applications: Survey*. 2014. Available at: https://api.semanticscholar.org/CorpusID:212489042

147. Pramaditya Wicaksono and Wahyu Lazuardi. *Random Forest Classifi- cation Scenarios for Benthic Habitat Mapping Using Planetscope Im- age*. In *IGARSS 2019 - 2019 IEEE International Geoscience and Re- mote Sensing Symposium*, 2019, pp. 8245-8248. Available at: https://api. semanticscholar.org/CorpusID:208035687

148. S. V. N. Santhosh Kumar and Talal Shaikh. *Empirical Evaluation of the Performance of Feature Selection Approaches on Random Forest*. In *2017 International Conference on Computer and Applications (ICCA)*, 2017, pp. 227-231. Available at: https://api.semanticscholar.org/CorpusID:35470060

149. Shaoyi Li, Hanxin Chen, Yongting Chen, Yunwei Xiong, and Ziwei Song, *Hybrid method with parallel-factor theory, a support vector machine, and particle filter optimization for intelligent machinery failure identification*, *Machines*, **11**(8), 837 (2023).

150. Min-Chan Kim, Jong-Hyun Lee, Dong-Hun Wang, and In-Soo Lee, *Induction motor fault diagnosis using support vector machine, neural networks, and boosting methods*, *Sensors*, **23**(5), 2585 (2023).

151. Shijin Wang, Avin Mathew, Yan Chen, Li-feng Xi, Lin Ma, and Jay Lee, *Empirical analysis of support vector machine ensemble classifiers*, *Expert Systems with Applications*, **36**, 6466–6476 (2009), https://api. semanticscholar.org/CorpusID:18547221.

152. Darrin P. Lewis, Tony Jebara, and William Stafford Noble, *Support vector machine learning from heterogeneous data: an empirical analysis using protein*

*sequence and structure*, *Bioinformatics*, **22**(22), 2753–2760 (2006), https://api.semanticscholar.org/CorpusID:14641265.

153. Shu Zhang, Samira Sadaoui, and Malek Mouhoub, *An Empirical Analysis of Imbalanced Data Classification*, *Computer and Information Science*, **8**, 151–162 (2015), https://api.semanticscholar.org/CorpusID:32412739.

154. Alexey Nefedov and Jiankuan Ye, *Experimental Study of Support Vec- tor Machines Based on Linear and Quadratic Optimization Criteria*, in *Proceedings of the 2009 International Conference on Computational Intelligence and Security*, (2009), https://api.semanticscholar.org/CorpusID: 117896183.

155. O. Kaminska, C. Cornelis, and V. Hoste, "Fuzzy rough nearest neighbour methods for aspect-based sentiment analysis," Electronics, vol. 12, no. 5,p. 1088, 2023.

156. S. S. Priscila, S. S. Rajest, R. Regin, T. Shynu, et al., "Classification of Satellite Photographs Utilizing the K-Nearest Neighbor Algorithm," Central Asian Journal of Mathematical Theory and Computer Sciences, vol. 4, no. 6, pp. 53–71, 2023.

157. K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is 'near- est neighbor' meaningful?," in International Conference on Database The- ory, 1999. [Online]. Available: https://api.semanticscholar.org/CorpusID: 206634099

158. W. Li, Y. Zhang, Y. Sun, W. Wang, W. Zhang, and X. Lin, "Approxi- mate nearest neighbor search on high dimensional data — experiments, analyses, and improvement," IEEE Transactions on Knowledge and Data Engineering, vol. 32, pp. 1475–1488, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:1364239

159. F. J. Ferrer-Troyano, J. S. Aguilar-Ruiz, and J. C. Riquelme Santos, "Em- pirical evaluation of the difficulty of finding a good value of k for the near- est neighbor," in International Conference on Conceptual Structures, 2003. [Online]. Available: https://api.semanticscholar.org/CorpusID:31951805

160. M. Kibriya and E. Frank, "An empirical comparison of exact nearest neighbour algorithms," in European Conference on Principles of Data Mining and Knowledge Discovery, 2007. [Online]. Available: https://api. semanticscholar.org/CorpusID:16525027

161. P. Supsermpol, S. Thajchayapong, and N. Chiadamrong, "Predicting financial performance for listed companies in Thailand during the transi- tion period: A class-based approach using logistic regression and random forest algorithm," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 9, no. 3, p. 100130, 2023.

162. J. Jeppesen, J. Christensen, P. Johansen, and S. Beniczky, "Personal- ized seizure detection using logistic regression machine learning based on wearable ECG-monitoring device," *Seizure: European Journal of Epilepsy*, vol. 107, pp. 155–161, 2023.

163. L. Niu, "A review of the application of logistic regression in educational research: common issues, implications, and suggestions," *Educational Review*, vol. 72, pp. 41-67, 2018. Available: https://api.semanticscholar. org/CorpusID:149736042

164. J. Fang, "Why Logistic Regression Analyses Are More Reliable Than Multiple Regression Analyses," in *Proceedings*, 2013. Available: https:// api.semanticscholar.org/CorpusID:199477137

165. P. Sur and E. J. Candès, "A modern maximum-likelihood theory for high- dimensional logistic regression," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, pp. 14516-14525, 2018.

166. Available: https://api.semanticscholar.org/CorpusID:4024186

167. C.-Y. J. Peng and T.-S. H. So, "Logistic Regression Analysis and Re- porting: A Primer," *Understanding Statistics*, vol. 1, pp. 31-70, 2002.Available: https://api.semanticscholar.org/CorpusID:121962352

168. A. V. D. Sano, A. A. Stefanus, E. D. Madyatmadja, H. Nindito, A. Purnomo, and C. P. M. Sianipar, "Proposing a visualized comparative review analysis model on tourism domain using Naïve Bayes classifier," *Procedia Computer Science*, vol. 227, pp. 482-489, 2023. Available: Elsevier.

169. E. D. Forson and P. O. Amponsah, "Mineral prospectivity mapping over the Gomoa Area of Ghana's southern Kibi-Winneba belt using support vector machine and naive bayes," *Journal of African Earth Sciences*, vol. 206, p. 105024, 2023. Available: Elsevier.

170. I. Rish, "An empirical study of the naive Bayes clas- sifier," in *Proceedings of the 2001 IJCAI Workshop on Empirical Methods in Artificial Intelligence*, 2001. Available: https://api.semanticscholar.org/CorpusID:61568722.

171. T. J. Watson, "An empirical study of the naive Bayes classifier," in *Proceedings of the 2001 IJCAI Workshop on Empirical Methods in Artificial Intelligence*, 2001. Available: https://api.semanticscholar.org/CorpusID:14891965.

172. H. Zhang and J. Su, "Naive Bayes for optimal ranking," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 20, pp. 79-93, 2008. Available: https://api.semanticscholar.org/CorpusID:12800860.

173. J. D. Nielsen and M. Jaeger, "An Empirical Study of Efficiency and Accuracy of Probabilistic Graphical Models," in *Proceedings of the Eu- ropean Workshop on Probabilistic Graphical Models*, 2006. Available:

https://api.semanticscholar.org/CorpusID:14000052.
174. S. M. Javidan, A. Banakar, K. A. Vakilian, and Y. Ampatzidis, "Diagnosis of grape leaf diseases using automatic K-means clustering and machine learning," *Smart Agricultural Technology*, vol. 3, pp. 100081, 2023.
175. M. Belhor, A. El-Amraoui, A. Jemai, and F. Delmotte, "Multi-objective evolutionary approach based on K-means clustering for home health care routing and scheduling problem," *Expert Systems with Applications*, vol. 213, pp. 119035, 2023.
176. S. M. Banerjee, A. Choudhary, and S. Pal, "Empirical evaluation of K- Means, Bisecting K-Means, Fuzzy C-Means and Genetic K-Means cluster- ing algorithms," in *2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2015, pp. 168-172. [Online]. Available: https://api.semanticscholar.org/CorpusID:15028862.
177. S. Sivarathri and A. Govardhan, "Experiments on Hypothesis "Fuzzy K- Means is Better than K-Means for Clustering," *International Journal of Data Mining & Knowledge Management Process*, vol. 4, pp. 21-34, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:11666076.
178. M. K. Gupta and P. Chandra, "An Empirical Evaluation of K-Means Clustering Algorithm Using Different Distance/Similarity Metrics," in *Proceedings of ICETIT 2019*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:204082752.
179. K. Javed, H. A. Babri, and M. Saeed, "The behavior of k-Means: An em- pirical study," in *2008 Second International Conference on Electrical Engineering*, 2008, pp. 1-6. [Online]. Available: https://api.semanticscholar. org/CorpusID:13111535.
*180.* B. Zhao, X. Wen, and K. Han, "Learning semi-supervised Gaussian mixture models for generalized category discovery," in *Proceedings of*
181. *the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 16623–16633.
182. T. Zhang, W. Chen, Y. Liu, and L. Wu, "An intrusion detection method based on stacked sparse autoencoder and improved Gaus- sian mixture model," *Computers & Security*, vol. 128, p. 103144, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/ pii/S0167404823001184.
183. T. S. Caetano, S. D. Olabarriaga, and D. A. C. Barone, "Performance evaluation of single and multiple-Gaussian models for skin color mod- eling," in *Proceedings. XV Brazilian Symposium on Computer Graphics and Image Processing*, 2002, pp. 275-282. [Online]. Available: https:
184. //api.semanticscholar.org/CorpusID:12216582.
185. D. Gokalp, "Learning Skin Pixels in Color Images Using Gaussian Mix- ture," in *Proceedings of the 2005 Conference*, 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:35700294.
186. M. Lucic, M. Faulkner, A. Krause, and D. Feldman, "Training Gaussian Mixture Models at Scale via Coresets," *J. Mach. Learn. Res.*, vol. 18, pp. 160:1-160:25, 2017. [Online]. Available: https://api.semanticscholar.org/ CorpusID:44070760.
187. R. Maas, J. Hyrkas, O. G. Telford, M. Balazinska, A. J. Connolly, and
188. B. Howe, "Gaussian Mixture Models Use-Case: In-Memory Analysis with Myria," in *Proceedings of the 3rd VLDB Workshop on In-Memory Data Management and Analytics*, 2015. [Online]. Available: https://api. semanticscholar.org/CorpusID:15485905.
189. J. Qian, Y. Cao, Y. Bi, H. Wu, Y. Liu, Q. Chen, and C. Zuo, "Structured illumination microscopy based on principal component analysis," *ELight*, vol. 3, no. 1, pp. 4, 2023. [Online]. Available: https://link.springer.com/ article/10.1186/s43540-023-00004-4.
190. J. Liu, H. Kang, W. Tao, H. Li, D. He, L. Ma, H. Tang, S. Wu, K. Yang,
191. and X. Li, "A spatial distribution–Principal component analysis (SD-PCA) model to assess pollution of heavy metals in soil," *Science of The Total Environment*, vol. 859, p. 160112, 2023. [Online]. Available: https://www. sciencedirect.com/science/article/pii/S0048969723011277.
192. L. R. Arnaut, C. Obiekezie, and D. W. P. Thomas, "Empirical Emission Eigenmodes of Printed Circuit Boards," *IEEE Transactions on Electro- magnetic Compatibility*, vol. 56, pp. 715-725, 2014. [Online]. Available:
https://api.semanticscholar.org/CorpusID:39691811.
193. R. Hubbard and S. J. Allen, "A Cautionary Note on the Use of Princi- pal Components Analysis," *Sociological Methods & Research*, vol. 16,
194. pp. 301-308, 1987. [Online]. Available: https://api.semanticscholar.org/ CorpusID:120561111.
195. W. F. Velicer, "An Empirical Comparison Of The Similarity Of Prin- cipal Component, Image, And Factor Patterns," *Multivariate Behavioral Research*, vol. 12, no. 1, pp. 3-22, 1977. [Online]. Available: https://api. semanticscholar.org/CorpusID:25353582.
196. S. Wold, K. H. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, pp. 37-52, 1987. [Online]. Available:
https://api.semanticscholar.org/CorpusID:265962113.
197. A. Faisal, N. Z. Jhanjhi, H. Ashraf, S. K. Ray, and F. Ashfaq, "A Com- prehensive Review of Machine

Learning Models: Principles, Applications, and Optimal Model Selection," *Authorea Preprints*, 2025