

**INT 354**

**MACHINE LEARNING - I**

**PROJECT REPORT**

**ON**

**LANDMINES**

|  |  |  |
| --- | --- | --- |
| **Submitted by** | | |
| NAME | REG. NO. | ROLL NO. |
| Shalini Tigga | 12205663 | RKM068B28 |

Under the guidance of

**DR. USHA MITTAL(20339)**

School of Computer Science and Engineering

Lovely Profession University, Jalandhar,

Punjab, India

**Table of Content**

Catalog

[1. ABSTRACT 1](#_Toc32298)

[2. INTRODUCTION 1](#_Toc11618)

[3. METHODOLOGY 2](#_Toc21266)

[4. DISCUSSION 4](#_Toc8933)

[5. CONCLUSION 7](#_Toc8183)

[6. ABBREVIATION 8](#_Toc5522)

[7. REFERENCE 8](#_Toc5881)

1. **ABSTRACT**

The detection of buried mines is crucial for ensuring the safety of life and property. Despite various methods being employed, achieving 100% success remains elusive. The mine detection process involves sensor design, data analysis, and decision algorithm phases. The magnetic anomaly method operates by measuring anomalies in the magnetic field caused by objects, which helps determine conditions such as motion and position. This method has been utilized since the 1970s for determining parameters like position, depth, or direction of motion.

In this study, we aim to identify the most effective algorithm for mine detection using magnetic anomaly data. This can lead to improved accuracy in identifying mines and reducing false positives. Identifying the most effective algorithms can also lead to more efficient and cost-effective mine detection processes. Algorithms that perform well can help reduce the time and resources required for mine detection operations. It can also improve the safety and reliability of mine detection operations. By reducing the need for manual detection methods, which can be dangerous and error-prone, these algorithms can help protect the lives of mine detection personnel.

Based on our research, the *Support Vector Machine (SVM)* seems to perform the best with the highest best score of 0.62 and the highest average accuracy in cross-validation of 0.52. We compare several machine learning algorithms to determine their accuracy in detecting mines, to improve the overall success rate of mine detection processes.

1. **INTRODUCTION**

*Machine learning (ML)* is a branch of artificial intelligence (AI) and computer science that focuses on using data and algorithms to enable AI to imitate the way that humans learn, gradually improving its accuracy. It enables computers to learn from data and improve their performance on tasks without being explicitly programmed. It focuses on the development of algorithms that can analyze and interpret data, identify patterns, and make decisions or predictions based on data.

The key principle behind machine learning is the ability to automatically learn and improve from experience. This is achieved through the use of statistical techniques that allow computers to "learn" from data, identify patterns, and make decisions or predictions without human intervention.

Machine learning algorithms play a pivotal role in extracting meaningful insights and making predictions from data. Selecting the most suitable algorithm for a given dataset is crucial for achieving optimal performance. This research project focuses on comparing and analyzing different machine-learning algorithms to determine the most effective approach for mine detection using magnetic anomaly data.

This paper presents a comprehensive comparative analysis of Machine learning algorithms on landmine dataset for optimal performance. The study evaluates and compares the performance of algorithms such as Perceptron, Linear Regression, SVM, KNN, and Decision Trees to identify the most effective approach for the dataset.

The primary objective of this research is to identify the most effective machine learning algorithm for mine detection using magnetic anomaly data to compare performance metrics (accuracy, precision, recall, F1 score, and Best Score) of different machine learning algorithms on the landmine dataset. The performance of each algorithm will be evaluated using cross-validation on the landmine dataset.

The significance of this paper lies in its contribution to the field of machine learning algorithm selection. By providing a systematic comparison of different algorithms on a specific dataset, this paper offers valuable insights into the strengths and weaknesses of each approach. The findings of this research can help researchers and practitioners in selecting the most suitable algorithm for similar datasets, ultimately improving the efficiency and accuracy of the machine learning model.

Overall, this paper provides a valuable contribution to the field of machine learning and data science by offering a detailed analysis of algorithm performance on a real-world dataset and providing practical recommendations for algorithm selection in machine learning applications.

1. **METHODOLOGY**
   1. **The Used Classifiers**

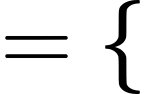
3.11 Perceptron

*Perceptron* is a type of artificial neural network that serves as a fundamental building block in the field of machine learning and neural networks. It was developed by *Frank Rosenblatt* in the late 1950s and is a simple model inspired by how neurons in the human brain work.

The perceptron takes a set of input features and each of which is associated with a weight . These weights determine the importance of each feature in the classification. The weighted sum of the inputs is then passed through an activation function, typically a step function. The activation function determines whether the perceptron "fires" (outputs a 1) or not (outputs a 0) based on the weighted sum. Mathematically, this can be represented as:

If 1, > threshold

0, otherwise



The perceptron is trained using a supervised learning approach, where it is presented with labeled training data and adjusts its weights to minimize classification errors.

Advantage: It can be computationally efficient, especially for linearly separable problems, as it converges to a solution in a finite number of steps.

Disadvantage: The perceptron can only classify data that is linearly separable, meaning it cannot handle non-linear relationships between features and classes.

3.12 Linear Regression

*Logistic regression* is a statistical model used for binary classification tasks, where the goal is to predict the probability that an instance belongs to a particular class (usually denoted as 1 or 0). Despite its name, logistic regression is a classification algorithm, not a regression algorithm.

The logistic regression model uses the *logistic function* (also called the sigmoid function) to model the relationship between the independent variables and the probability of belonging to a certain class. The logistic function is defined as:

Where is the probability is the probability that the instance belongs to class 1 given the features

are the coefficients of the model, which are estimated during the training process, e is the base of the natural logarithm.

Advantages: Logistic regression is less prone to overfitting compared to more complex models when the number of features is small.

Disadvantages: Logistic regression is inherently binary, meaning it can only be used for binary classification tasks. Extensions like multinomial logistic regression can handle multiple classes, but it's still limited compared to some other models.

3.13 Support Vector Machine

*Support Vector Machine (SVM)* is a powerful supervised machine learning algorithm used for classification and regression tasks. It works by finding the hyperplane that best separates different classes in the feature space.SVM aims to find the hyperplane that not only separates the data but also maximizes the margin, making it more robust to noise and outliers.

SVM can handle non-linearly separable data by using a technique called the kernel trick. The kernel trick maps the input data into a higher-dimensional space where it may be linearly separable. Common kernels used in SVM include linear, polynomial, *radial basis function* (RBF), and sigmoid kernels.

SVM also includes a regularization parameter (C) that controls the trade-off between maximizing the margin and minimizing the classification error. A higher value of C allows for a smaller margin but fewer misclassifications, while a lower value of C allows for a larger margin but more misclassifications.

Advantages: SVMs use a subset of training points (support vectors) in the decision function, making them memory efficient, especially for datasets with large numbers of features.

Disadvantages: Training an SVM can be computationally expensive, especially for large datasets, due to the need to solve a quadratic optimization problem.

3.14 K-Nearest Neighbour

*K-Nearest Neighbors (KNN)* is a simple, instance-based learning algorithm used for classification and regression tasks. In KNN, the output value for a given data point is determined by the majority class (for classification) or the average of the closest k data points (for regression) in the training dataset.

Advantages: KNN is an instance-based learning algorithm, which means it does not require a training phase. The model simply memorizes the training data and makes predictions based on the closest neighbors.

Disadvantages: KNN can be computationally expensive, especially for large datasets, as it requires calculating the distance between the test data point and all other points in the training dataset.

3.15 Decision Tree

A *Decision Tree* is a popular and widely used machine learning algorithm that can be used for both classification and regression tasks. It is a tree-like structure where:

* Each internal node represents a "decision" based on the value of a feature.
* Each branch represents the outcome of the decision.
* Each leaf node represents the final decision or the target variable.

Advantages: Decision trees can handle both numerical and categorical data without the need for data normalization or scaling.

Disadvantages: Decision trees are prone to overfitting, especially with deep trees. Techniques like pruning and setting a minimum number of samples per leaf node can help mitigate this issue.

* 1. **Experiment**

3.21 Hardware Requirement

* Processor: Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
* Memory(RAM): 8.00 GB (7.89 GB usable)
* Storage: 256 GB

3.22 Software Requirement

* OS: Windows
* Programming Language: Python
* Libraries: NumPy, pandas, matplotlib, sci-kit learn

3.23 Dataset

In this paper, we used "Land Mine" dataset from UCI Machine Learning Repository. The creator of this dataset is Hamdi Tolga KAHRAMAN.

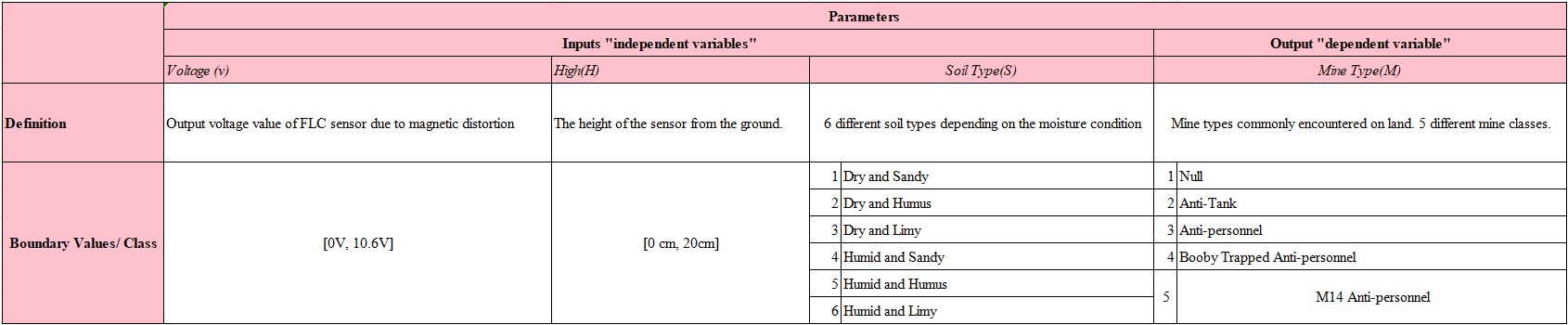


Fig 3.23 Parameter of out dataset

3.24 Data Preprocessing

Data preprocessing is a crucial step in the data mining and machine learning pipeline that involves transforming raw data into a format that is suitable for analysis. Therefore, we performed the following on our data:

* Normalisation: It is scaling numerical features to a standard range. We used StandardScaler to perform the same.
* Encoding: It is converting categorical variables into numerical values. We used LabelEncoder to perform the same.

We didn’t have to perform dropna or imputation because our data had no missing values.

3.25 Performafance Metrics

* TP(True Positive): True positive predicts the positive cases positive.
* TN(True Negative): True negative predicts the negative cases positive.
* FP(False Positive): False positive predicts the negative cases as positive.
* FN(False Negative): False Negative predicts the positive cases as negative.
* These are the following formulas used to calculate the performance metrics like accuracy, precision, recall.
* Accuracy: Accuracy measures the proportion of correct predictions (both true positives and true negatives) out of the total number of predictions. It provides an overall assessment of how well the model performs across all classes. However, accuracy may not be the best metric for imbalanced datasets, as it can be misleading when one class dominates the dataset.

Accuracy = (TP + TN)/(TP+FP+FN+TN)

* Precision: Precision measures the proportion of correctly predicted positive case(true positives) out of all predicted positive cases (true positives + false positives). It indicates the ability of the model to avoid false positives. A high precision indicates that when the model predicts a positive outcome, it is likely to be correct.

Precision = TP/(TP+FP)

* Recall (Sensitivity): Recall measures the proportion of correctly predicted positive cases (true positives) out of all actual positive cases (true positives + false negatives). It indicates the ability of the model to capture all positive cases. A high recall indicates that the model is good at identifying positive cases.

Recall = TP/(TP+FN)

1. **DISCUSSION**

**4.1 Best Score**

The best score typically refers to the best performance metric achieved by the algorithm during hyperparameter tuning or model selection. It represents the highest level of performance that the algorithm was able to achieve on the given dataset and can be used to compare the algorithms' relative performance.

The best score of the following algorithms are:

* Perceptron: 0.46690196078431373
* Logistic Regression: 0.4939607843137255
* SVM: 0.6282352941176471
* KNN: 0.5967058823529412
* Decision Tree: 0.5491764705882354

As we can see that SVM has the highest best score.This indicates that SVM achieved the highest performance on our dataset among the algorithms listed.

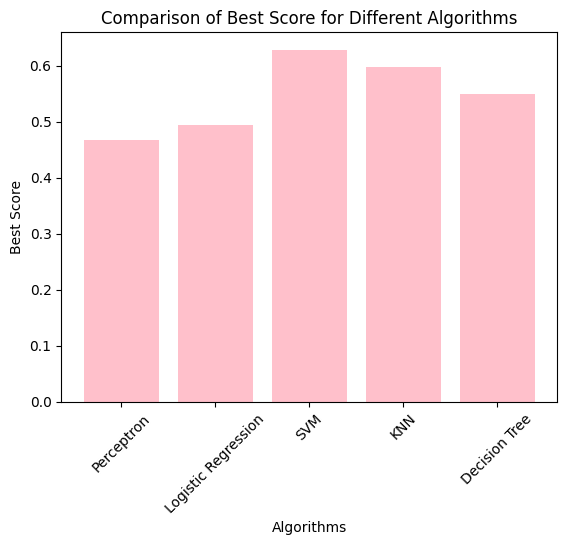


Fig 4.1 Comparison of Best Score for different algorithms

**4.2 Training Results**

F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall. It is useful when you want to consider both false positives and false negatives and need a single metric to evaluate the model's performance.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Training Accuracy** | **Precision Accuracy** | **Recall score** | **F1 Score** |
| **Perceptron** | 0.3675889328063241 | 0.37764162575326915 | 0.3675889328063241 | 0.34428589570024876 |
| **Logistic Regression** | 0.5177865612648221 | 0.48109970276543934 | 0.5177865612648221 | 0.4812318527523994 |
| **SVM** | 0.6521739130434783 | 0.657764505229124 | 0.6521739130434783 | 0.6244910554813298 |
| **KNN** | 0.7391304347826086 | 0.7515386416333287 | 0.7391304347826086 | 0.7296296694178803 |
| **Decision Tree** | 0.6561264822134387 | 0.6613542530653105 | 0.6561264822134387 | 0.6481631791086315 |

Fig 4.21 Training Result Table

As we can see KNN has the best statistics out of the 5 indicating that it performed the best on the training dataset.

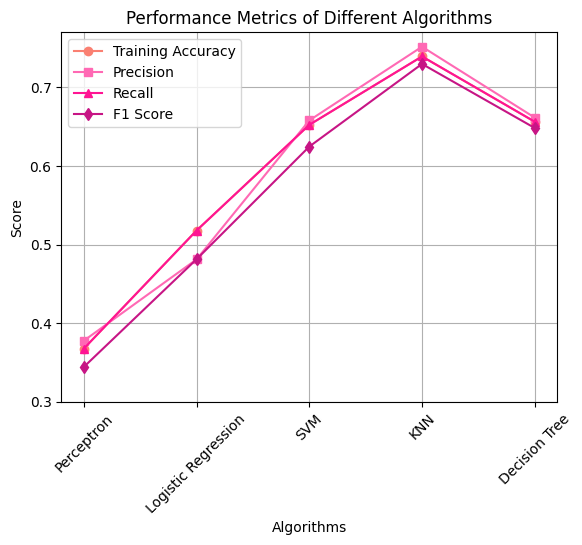


Fig 4.22 Performance Metric of Different Algorithms

**4.3 Testing Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Training Accuracy** | **Precision Accuracy** | **Recall score** | **F1 Score** |
| **Perceptron** | 0.43529411764705883 | 0.4860130718954248 | 0.43529411764705883 | 0.4334865726346457 |
| **Logistic Regression** | 0.4823529411764706 | 0.4986309369725711 | 0.4823529411764706 | 0.4555699692333496 |
| **SVM** | 0.5294117647058824 | 0.5326797385620915 | 0.5294117647058824 | 0.50741408307389 |
| **KNN** | 0.5176470588235295 | 0.49798761609907116 | 0.5176470588235295 | 0.4945221224645358 |
| **Decision Tree** | 0.5294117647058824 | 0.4800150829562595 | 0.5294117647058824 | 0.49762874071584773 |

Fig 4.31 Testing Result Table

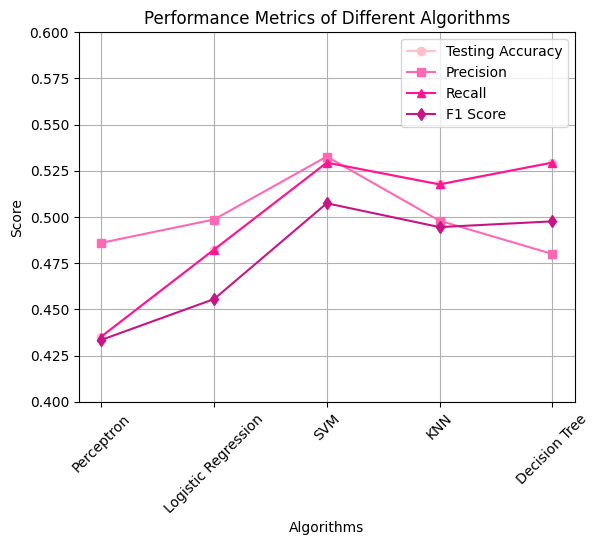


Fig 4.32 Performance Metrics of Different Algorithms

As we can see SVM has the best statistics out of 5 indicating that it performed the best on the testing dataset.

**4.4 Cross Validation**

|  |  |  |
| --- | --- | --- |
|  | **Average Accuracy** | **Standard Deviation** |
| **Perceptron** | 0.3917647058823529 | 0.05400044142385828 |
| **Logistic**  **Regression** | 0.46674509803921566 | 0.06699912490032005 |
| **SVM** | 0.5258039215686274 | 0.04158963583075931 |
| **KNN** | 0.4828235294117647 | 0.07216172191652377 |
| **Decision Tree** | 0.49035294117647055 | 0.03689327063186708 |

Fig 4.41 Cross Validation Table

As we can see that SVM has the highest average accuracy and relatively low standard deviation indicating that it performs well across different folds.

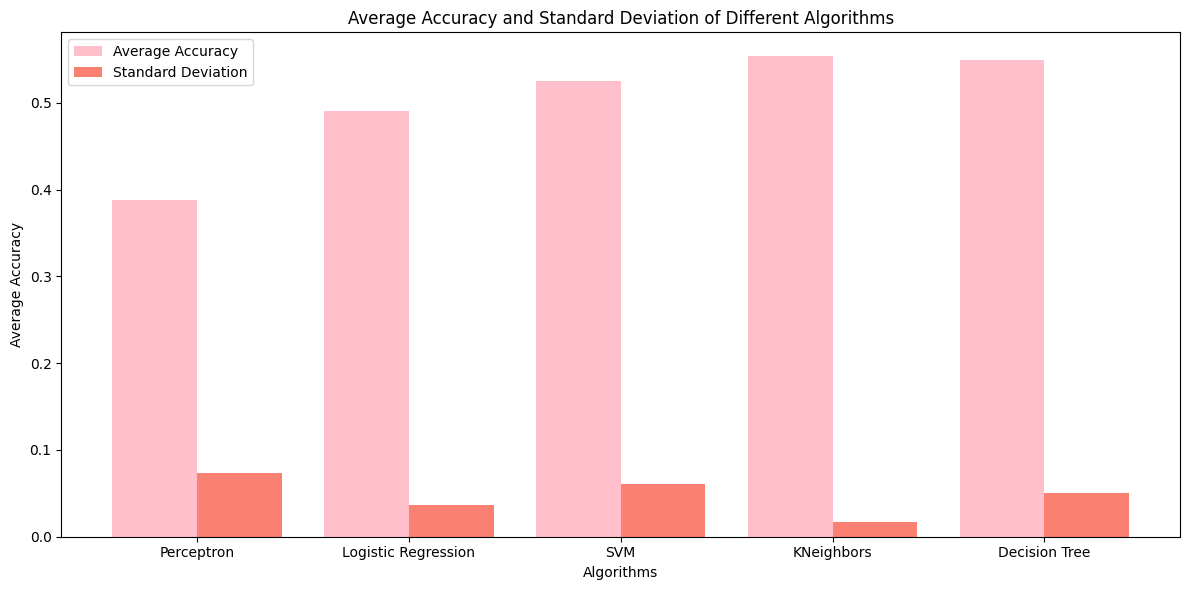


Fig 4.41 Cross Validation Graph

1. **CONCLUSION**

In this study, we compared several machine learning algorithms and found that the SVM algorithm outperformed others on our dataset. SVM achieved the highest testing accuracy, precision, recall, and F1 score, along with the highest best score during hyperparameter tuning. This indicates its effectiveness and suitability for our dataset. While the KNN algorithm showed the highest training accuracy, precision, recall, and F1 score, its performance on the testing dataset was slightly lower, suggesting potential overfitting to the training data.

Based on our findings, we recommend the use of SVM as the best algorithm for our dataset, considering its strong performance on both training and testing datasets, as well as during hyperparameter tuning.

The superior performance of the SVM algorithm suggests its potential for real-world applications requiring high predictive performance. For instance, in medical diagnostics or financial forecasting, where accuracy and reliability are crucial, SVM could lead to more accurate predictions and better decision-making.

Additionally, the insights gained from our study can serve as a valuable reference for researchers and practitioners selecting machine learning algorithms for similar datasets. Our findings guide choosing algorithms for classification tasks, especially when dealing with datasets that share characteristics similar to ours.

Overall, the use of SVM could significantly enhance predictive performance in various real-world applications, making it a promising choice for practitioners seeking reliable and accurate machine learning models.

1. **ABBREVIATION**

* ML - Machine Learning
* AI - Artificial Intelligence
* SVM - Support Vector Machine
* KNN - K-Nearest Neighbour
* RBF - Radial Basis Function

1. **REFERENCE**

[1] Yilmaz, C., Kahraman, H. T., & Söyler, S. (2018). Passive mine detection and classification method based on hybrid model. IEEE Access, 6, 47870-47888.

[2] Ganesha Bhumireddy, Venkata Ajay Surendra Manikanta Anala (2022). "Comparison of Machine Learningalgorithms on detecting the confusion of students while watching MOOCs." Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden.

[3] Kapil Sethi, Ankit Gupta, Gaurav Gupta, Varun Jaiswal (2019). "Comparative Analysis of Machine Learning Algorithms on Different Datasets"

[4] Cemal Yilmaz, Hamdi Tolga Kahrahman,Salih Söyler (2018). "Passive Mine Detection and Classification Method Based on Hybrid Model"