Mine detection with machine-learning

Anudeep Reddy Pothula
MS in Data Science
Stevens Institute of Technology
Hoboken, USA
apothula1@stevens.edu

Benson Moses Palaparthi
MS in Computer Science
Stevens Institute of Technology
Hoboken, USA
bpalapar@stevens.edu

Yunwei Zheng
MS in Computer Science
Stevens Institute of Technology
Hoboken, USA
yzheng58@stevens.edu

***Abstract*—Detection of underwater mines is performed using Support Vector Machine, Decision Tree, Random Forest Models, K-Nearest Neighbours, and a Neural Network.**

I. Introduction

Mines have been traditionally detected using expensive machinery or manual effort. The presence of mines in erstwhile war-zones poses a great risk to life. Therefore, a need arises for the passive and accurate detection of mines from alternate methods. The use of machine learning algorithms is a step towards achieving this in a more efficient manner.

The dataset was gathered by Jerry Sejnowski, now at the Salk Institute and the University of California at San Deigo et al. The dataset consists of 59 features that were obtained by from multiple sonar readings off a rock or a metal cylinder (mine). The response is a 1 for a mine and 0 for a rock.

We will use instance based methods like K-Nearest Neighbours, Decision Tree, Random Forest Models, Support Vector Machine along with building an ANN to predict the dependent variable. K-Nearest Neighbours, Decision Tree, Random Forest Models, Support Vector Machine are less intensive algorithms to implement compared to Neural Network models.

II. Related Work

DeCoste et al [1] employed Cholesky Factorization methods to speedup SVM and Kernel Fisher classifiers on the data.

Demiriz et al [2] used clustering, albeit with a genetic algorithm employed to solve the objective function.

Gorman and Sejnowski et al [3] identified specific signal features in the hidden layer of the neural network. The performance was comparable to that of trained human listeners.

Gurbel and Moore et al [4] created a probabilistic model which generates training data without noise.

Tan and Dowe et al [5] employed a multivariate decision tree using Minimum Message Length (MML) principle as an objective function, which returned a better precision on both 0 and 1 than decision tree C4.5 and C5 programs.

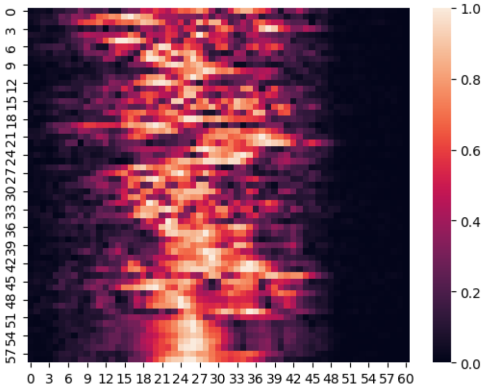
Zhou and Jiang et al [6] employed a neural ensemble based decision tree, NeC4.5, in which a neural network generates a new training dataset on which C4.5 decision tree is employed.

III. Our Solution

A. Description of Dataset

The dataset has been taken from the UC Irvine database. The dataset has four features- Voltage in the sensor, height of the sensor from the ground, type of soil. Voltage and height are continuous variables. The type of soil is a categorical variables with 6 values. The dependent variable is the mine-type of which 5 values exist.

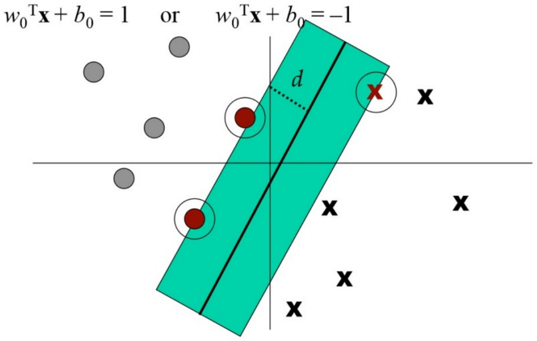
The values of the predictors are normalized and range from 0 to 1. Some features are predominantly on the higher side, as the heat map shows, and some on the lower side.



B. Machine Learning Algorithms

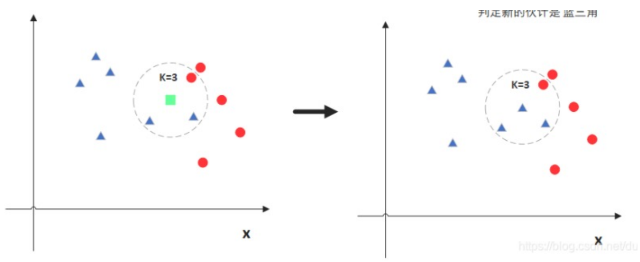
Various machine learning techniques have been applied for classification tasks.

SVM: Support Vector Machines (SVMs) are commonly used and extendable to both binary and multi-class classification problems. Random Forests, an ensemble method that combines multiple decision trees, are also a popular choice for classification tasks.



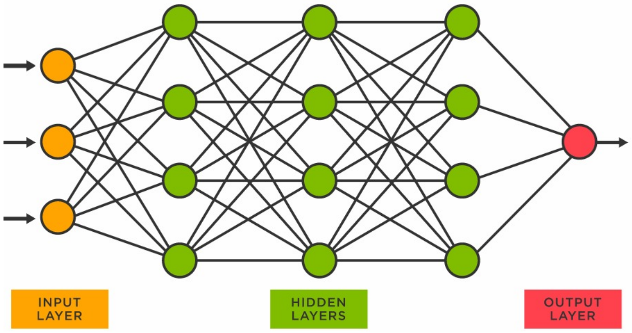
KNN is a non-parametric algorithm that can be used for both classification and regression tasks. In the case of classification, KNN assigns a label to a new observation based on the majority class of its k-nearest neighbors. In the case of regression, KNN assigns a value to a new observation based on the average value of its k-nearest neighbors.

KNN is a useful algorithm for dealing with non-linear decision boundaries and noisy data. However, as pointed out, the computational cost can become a challenge for large datasets, and the value of k should be tuned to achieve optimal performance.

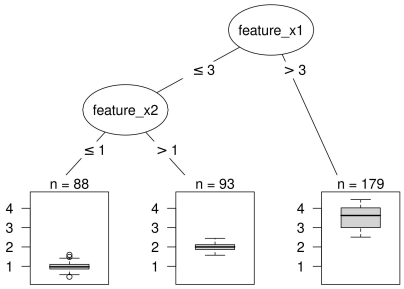


ANN: Artificial Neural Networks (ANNs) mimic biological neural networks and consist of interconnected nodes arranged in layers. It can handle vast amounts of complex nonlinear data and execute tasks such as prediction, classification, and optimization. The fundamental structure of an ANN is composed of multiple layers of interconnected neurons. Each neuron obtains input signals from other neurons and processes them to generate an output signal.

The connections between neurons have varying weights, which can be adjusted during a process called training. This process involves tuning the weights based on the desired output. ANN is a versatile tool in machine learning and has been utilized in diverse fields such as natural language processing, image recognition, and financial forecasting.



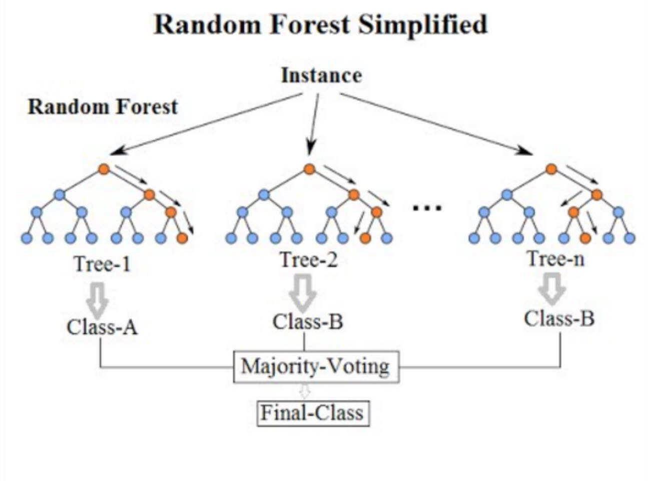
Decision Tree: Decision Trees are a simple but powerful technique that partition the feature space and assign labels based on the majority label of training samples within each partition.



Random Forest Models: Random Forest is a popular ensemble learning algorithm used for both classification and regression tasks in machine learning. It is a collection of decision trees that are trained on random subsets of the training data, and then their predictions are combined to make the final prediction.

In Random Forest, each decision tree is built independently using a different subset of the training data and a random subset of the features. This helps to reduce overfitting and improve the model's generalization ability.

When making a prediction, the Random Forest algorithm aggregates the predictions of all the individual decision trees to arrive at a final prediction. The final prediction is typically determined by a majority vote in classification tasks, or by taking the average of the predictions in regression tasks.



These methods offer different strengths and have been successfully applied in various classification tasks

C. Implementation Details

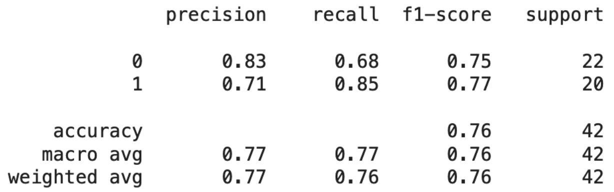
SVMs are a type of supervised machine learning algorithm used for classification and regression analysis. Which constructs a hyperplane to separate data points into different classes, maximizing the margin between the closest data points from each class.

It uses kernel functions to map input data into a higher-dimensional space, where the data becomes linearly separable, allowing SVMs to capture non-linear relationships between features in the data.

The Support Vector Machine has been implemented using the inbuilt modules of the SciKitLearn module of Python. A

C-support support vector machine with a linear kernel has been selected. SVMs are efficient in small datasets. They are less prone to overfitting, which is required in a dangerous problem such as classifying mines.

This algorithm provides a 76% precision on the test set.

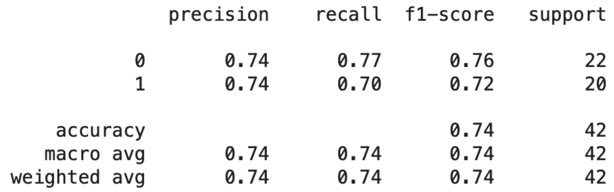


```
              precision    recall  f1-score   support

           0       0.83      0.68      0.75        22
           1       0.71      0.85      0.77        20

    accuracy                           0.76        42
   macro avg       0.77      0.77      0.76        42
weighted avg       0.77      0.76      0.76        42
```

A decision tree is fit using SciKit Learn's inbuilt module. It reports a 74% precision.

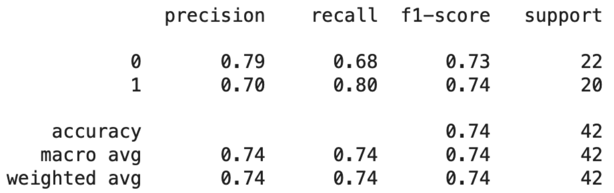


```
              precision    recall  f1-score   support

           0       0.74      0.77      0.76        22
           1       0.74      0.70      0.72        20

    accuracy                           0.74        42
   macro avg       0.74      0.74      0.74        42
weighted avg       0.74      0.74      0.74        42
```

A random forest was fit using SciKit Learn's inbuilt module. It reports a 74% precision.

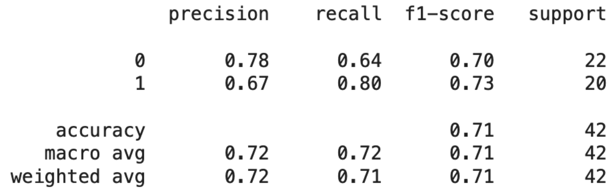


```
              precision    recall  f1-score   support

           0       0.79      0.68      0.73        22
           1       0.70      0.80      0.74        20

    accuracy                           0.74        42
   macro avg       0.74      0.74      0.74        42
weighted avg       0.74      0.74      0.74        42
```

A K-Nearest Neighbours was fit using SciKit Learn's inbuilt module. It reports a 74% precision.

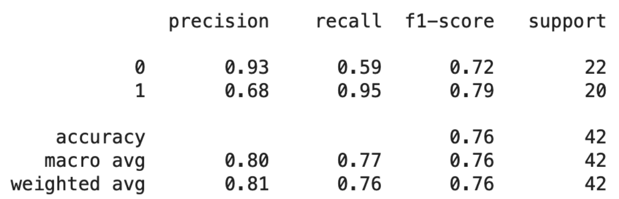


```
              precision    recall  f1-score   support

           0       0.78      0.64      0.70        22
           1       0.67      0.80      0.73        20

    accuracy                           0.71        42
   macro avg       0.72      0.72      0.71        42
weighted avg       0.72      0.71      0.71        42
```

A sequential neural-network is employed from the inbuilt Kera module with ReLu and Sigmoid activation functions. An 'adam' optimizer is used and the loss function is set to binary cross-entropy. The neural network gives a precision of 80%.



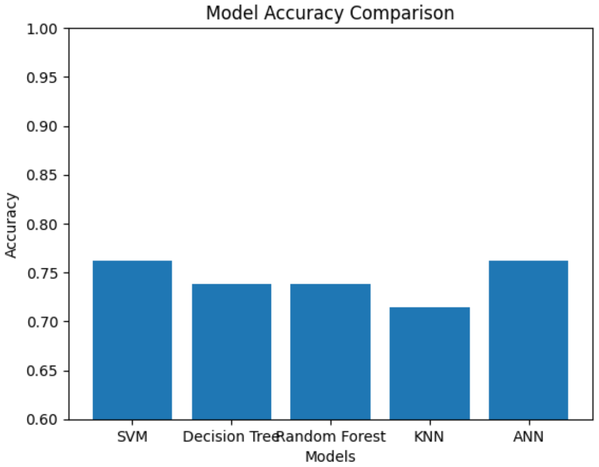
```
              precision    recall  f1-score   support

           0       0.93      0.59      0.72        22
           1       0.68      0.95      0.79        20

    accuracy                           0.76        42
   macro avg       0.80      0.77      0.76        42
weighted avg       0.81      0.76      0.76        42
```

IV. Comparison

The neural network returns the highest precision among all models, followed by the support-vector machine. The random forest and the decision tree are next with the same accuracy. The K Nearest Neighbours has the least precision. The F-scores are also in that order. The Support Vector Machine has performed the best, though the neural network has better numbers. This is because the support vector machine is less computationally intense than the neural network.



This section includes the following: 1) comparing the performance of different machine learning algorithms that you used, and 2) comparing the performance of your algorithms with existing solutions if any. Please provide insights to reason about why this algorithm is better/worse than another one.

V. Future Directions

Resampling methods like bagging, boosting, bootstrapping can be implemented. The algorithms employed can be more fine-tuned particularly for employing on this dataset.

VI. Conclusion

We have achieved a better accuracy compared to a naive Bayes Classifier using the above techniques.

References

[1] DeCoste, D. (2003). Anytime Query-Tuned Kernel Machines Via Cholesky Factorization. https://doi.org/10.1137/1.9781611972733.17

[2] Demiriz, A., Bennett, K. P., Embrechts, M. J. (2002). A Genetic Algorithm Approach for Semi-Supervised Clustering. International Journal of Smart Engineering System Design, 4(1), 21–30. https://doi.org/10.1080/10255810210623

[3] Gorman, R. P., Sejnowski, T. J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. Neural Networks, 1(1), 75–89. https://doi.org/10.1016/0893-6080(88)90023-8

[4] Gurbel, P. A., Moore, A. (2003). Probabilistic noise identification and data cleaning. https://doi.org/10.1109/icdm.2003.1250912

[5] Tan, P. P. C., Dowe, D. L. (2004). MML Inference of Oblique Decision Trees. In Lecture Notes in Computer Science (pp. 1082–1088). Springer Science+Business Media. https://doi.org/10.1007/978-3-540-30549-1$_1$05

[6] Zhou, Z., Jiang, Y. (2004). NeC4.5: neural ensemble based C4.5. IEEE Transactions on Knowledge and Data Engineering, 16(6), 770–773. https://doi.org/10.1109/tkde.2004.11

[7] Burges, C. (1998). A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2), 121-167.

[8] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[9] Nayak, A. R., Nayyar, A. (2019). Comparison of SVM and CNN for Image Classification: A Case Study with Small Datasets. In 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon) (pp. 121-126). IEEE.

[10] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[11] Geitgey, A. (2018). SVM vs. CNN for image classification. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/svm-vs-cnn-for-image-classification/