SUPERVISED LEARNING MODELS

Introduction

This work investigates the nature of supervised learning models by applying them to two interesting datasets for binary classification. The models in question here include decision tree, knn, svm, neural networks and boosted decision trees. The attempt of this work allows us to understand how this two binary classification datasets yield interesting results that highlight the inner workings of these supervised learning models.

Datasets:

The ionospere dataset of data collected by a radar system in Goose Bay, Labrador. This system is composed of a phased array of 16 high frequency antennas to detect free elections in the ionosphere. In general, there are two types of structures in the ionosphere: “Good” and “Bad”. These radars detected these structures and passed the signal. There are 34 independent variables and one dependant, and 351 observations in total. The interesting thing is that this is an imbalanced dataset, so you can practice that as well. It is also not easy to achieve high accuracy on this dataset and the baseline performance is around 64%, while the top accuracy is around 94%. Despite that all the models performed very well on the test set with balanced accuracy scores above 80. Thus it leaves a great room for learning about how these models were able to perform well on the test set despite issues regarding low number of samples, high features and imbalanced class.

The sonar dataset consists metrics from sonar signals that distinguish between rocks and mines. It is an imbalanced dataset with 208 examples, 60 input features, and one output feature. This dataset has similar characteristics in number of samples and class imbalance when compared to the ionosphere dataset. The key difference here is the number of input features. However, for this dataset the same models performed comparatively with less accuracy. The baseline accuracy for this dataset is around 88% and all the models were unable to supersede that.

The following sections analyzes why we see these behaviors from the datasets from the perspective of the models.

SVM

Based on figure 4 and 5 the data data distribution of both the datsets are barely linearly seperable thus according to plots in figure 6 and 7 the rbf kernel had a higher f1 score. Despite the ionosphere being imbalanced the svm modeled performed with an f1 score of 0.96 as shown in figure 1 and the confusion matrix in figure 8 . Some impacts of imbalanced dataset could be results in the hyperplanes being skewed to the minority class during training. The second reason arises from the issue of an imbalanced support vector ratio, i.e. the ratio between the positive and negative support vectors becoming imbalanced and as a result, datapoints at the decision boundaries of the hyperplanes have a higher chance of being classified as negative. One reason for svm was able to handle this imbalance can be due the the correct balance of gamma and C values. The gamma and C values are 0.01 and 10 respectively. When gamma is very small, the model is too constrained and cannot capture the complexity or “shape” of the data. The region of influence of any selected support vector would include the whole training set. Smooth models (lower gamma values) can be made more complex by increasing the importance of classifying each point correctly (larger C values) hence this model has a lower

gamma value is complemented with a higher C value. This allows it to capture the complex decision boundaries that this datset can be plagued with without over fitting.

The sonar datset is similar to the first one with double number of features but similar number of samples and same issue of class imbalance. However, in this dataset the models performance are far worse. This could be due to the lack of training examples whose impact can be see on the learning curve on figure 10. Given the large gap between the train and validation graph and plateauing of the validation curve we can see that enough samples aren't there for inducing learning. Furthermore figure 9 shows an interesting situation where C is unchanged throughout. One can also observe that for some intermediate values of gamma we get equally performing models when C becomes very large. This suggests that the set of support vectors does not change anymore. The radius of the RBF kernel alone acts as a good structural regularizer. Increasing C further doesn't help, likely because there are no more training points in violation (inside the margin or wrongly classified), or at least no better solution can be found.

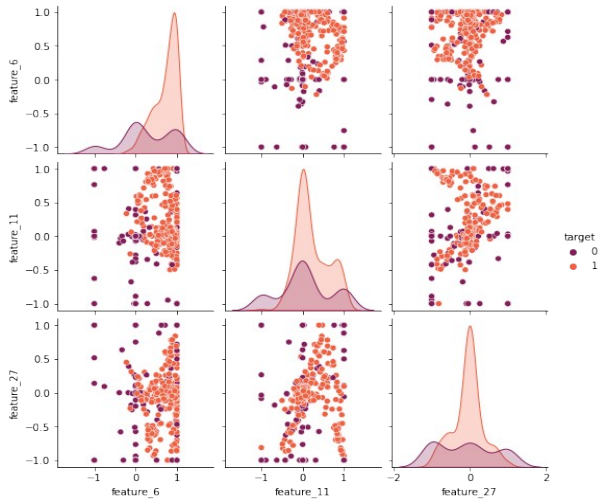


Fig 4 Ionosphere dataset

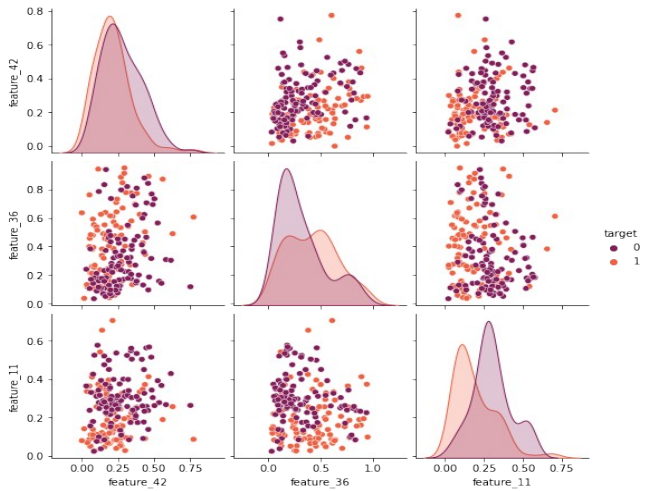


Fig 5 Sonar dataset scatter plot

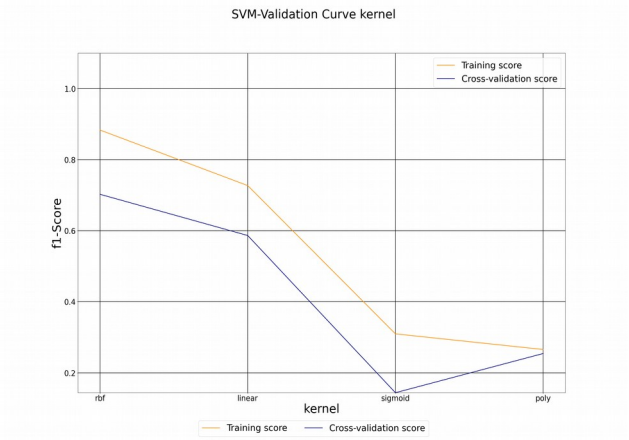


Fig 6 Kernel validation curve-sonar

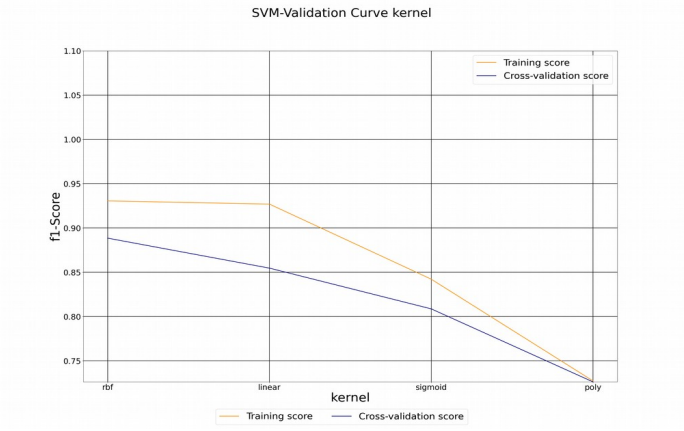


Fig 7 Kernel validation curve-ionosphere

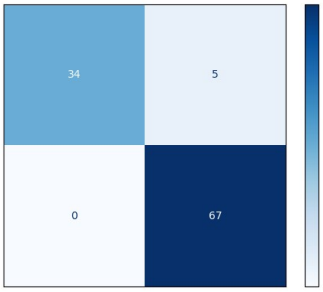


Fig 8 Confusion Matrix Ionosphere

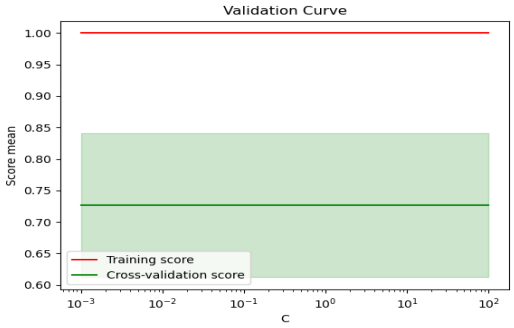


Fig 9 C-Validation curve sonar

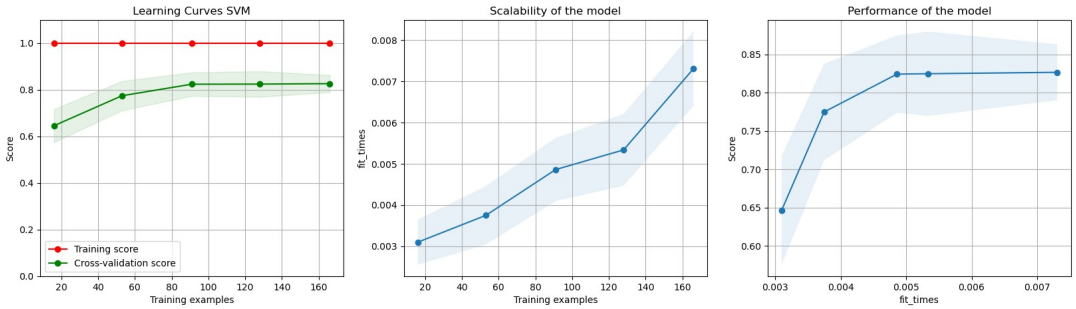


Fig 10 Learning Curves Sonar

KNN

Knn is an algorithm that can be susceptible to class imbalance. As we know that the class label assigned to a query vector is determined by a majority vote amongst the k-nearest neighbors of the query vector. The inductive bias of knn classification in this case assumes that the space of all samples can be partitioned into two disjoint subspaces. If training set is large enough and balanced, it'd seem as if the subspaces were sampled uniformly for generating the training set. If the training set were to be imbalanced, and if the uniform distribution assumption were to still hold, the probability that the k nearest neighbors of any random query point will belong to the class with more examples becomes higher. So, the closest neighbor of the query point may still belong to the class with less examples, but if rest (k-1) points belong to the other class (because of its higher density in the space), the point will get misclassified. However, both the datasets performed considerably well using KNN as summarized in table 1

Model	Parameters	f1-score
KNN-ionosphere	{'weights': 'uniform', 'n_neighbors': 4, 'metric': 'chebyshev', 'algorithm': 'brute'}	0.928
KNN-sonar	{'weights': 'distance', 'n_neighbors': 4, 'metric': 'euclidean', 'algorithm': 'brute'}	0.905

Table 1 KNN results summary

In addition to class imbalance we can also see how these dataset pose a challenge associated with the curse of dimensionality. For the ionosphere dataset as the number of features grow linearly, the amount of data we need to accurately generalize the grows exponentially. If we have a lot of features and we treat them all with equal importance, we're going to need a LOT of data to determine which ones are more relevant than others. Thus a common expectation would be for this sample to suffer from the curse of dimensionality. Despite that the best estimator trained on this model chose uniform weights instead of basing it on a distance function where the nearest points would have more impact on the given datapoint. also the number of features for this dataset is considerably high compared to the available samples. I believe for this dataset the chebyshev distance metric allows it to perform well. Chebyshev looks at max distance between an axis rather than the exact datapoint location in the feature space. Thus this may allow it to handle the imbalance nature of the data set my ensuring that the majority class despite being nearer to a point leaves room for consideration of another point in the minority class and of its one of its axis might be closer.

We can conclude the same for the sonar dataset that it may be more prone to the curse of dimensionality given it has the almost the double number of features compared to ionosphere. But the euclidean and distance based weights could remedy that as now the importance of datasets are based on the distance from the point of interest. Figures 11 and 12 show the confusion matrix for both datasets.

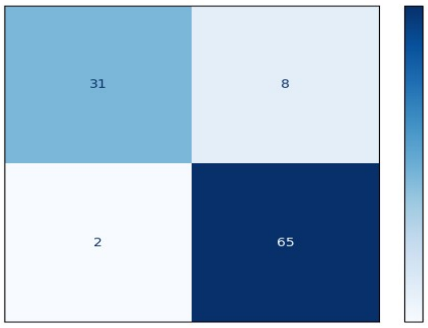


Figure 11 Ionosphere

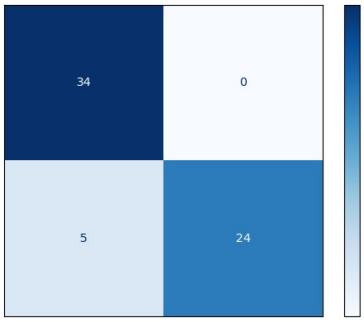


Figure 12 Sonar

Adaboost:

The adaboost classifier for both databases consists of a decision tree classifier as the base estimator. For both the datasets we can see a considerable jump in f1 score and balanced accuracy when compared to the single decision tree model as shown in table 2. The key reasoning here could be due to the relationship between learning rate and the number of estimators. Decreasing the learning rate makes the regularization coefficients smaller, which reduces the amplitude of the sample_weights at each step. This translates into smaller variations of the weighted data points and therefore fewer differences between the weak classifier decision boundaries. Increasing the number of weak classifiers M increases the number of iterations, and allows the sample weights to gain greater amplitude. This translates into more weak classifiers to combine at the end, and more variations in the decision boundaries of these classifiers. Thus putting these together effects tend to lead to more complex overall decision boundaries. For both datasets we see a higher number of estimators that could be due to the complex decision boundaries that describes the dataset. This comlpex decision boundaries could have been missed by a single decision tree model but multiple weak classifiers put together by adaboost is able to

capture these boundaries better. However, adjusting completely to the complex decision boundaries can lead to overfitting which in this case is being regularized by the low learning rate.

Model	Parameters	f1
DT-sonar	{'splitter': 'random', 'max_depth': 6, 'criterion': 'gini', 'ccp_alpha': 0.013505}	0.75
Ada-sonar	{'n_estimators': 5, 'learning_rate': 0.01, 'base_estimator__splitter': 'random', 'base_estimator__max_depth': 5}	0.87
DT-ionosphere	{'splitter': 'random', 'max_depth': 9, 'criterion': 'gini', 'ccp_alpha': 0.00757}	0.91
Ada-ionosphere	{'n_estimators': 100, 'learning_rate': 0.16, 'base_estimator__splitter': 'random', 'base_estimator__max_depth': 5}	0.99

ANN

As we have established in the previous scatter plots of ionosphere dataset we can see that the data is hardly linearly seperable. MLP classifier is composed of multiple layered perceptrons. The inate nature of perceptrons is to compute lines or hyperplanes in n dimensions that will attempt to linearly seperate the data. Given the complex distribution of this dataset the hyperparameter search returned a 3 hidden layers with 66 nodes in each. The higher the number of layers as shown in figure 13 and nodes the better the ability of the the mlp classifier to find more complex boundaries between each classification label. Furthermore an interesting aspect is that despite the imbalanced nature of the dataset the mlp classifier was able to tackle that by assigning a very low value for alpha. As shown in figure 14 alpha is a parameter for regularization term that handles overfitting by constraining the size of the weights. Increasing alpha may fix high variance (a sign of overfitting) by encouraging smaller weights, resulting in a decision boundary plot that appears with lesser curvatures. Similarly, decreasing alpha may fix high bias (a sign of underfitting) by encouraging larger weights, potentially resulting in a more complicated decision boundary. This appears to complement the high number of nodes and hidden layer that was chosen to capture these complicated decision boundaries.

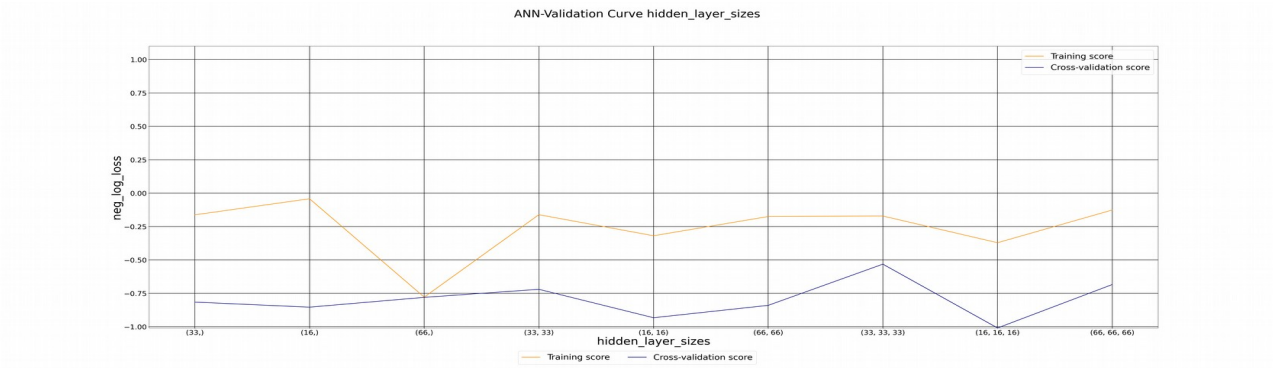
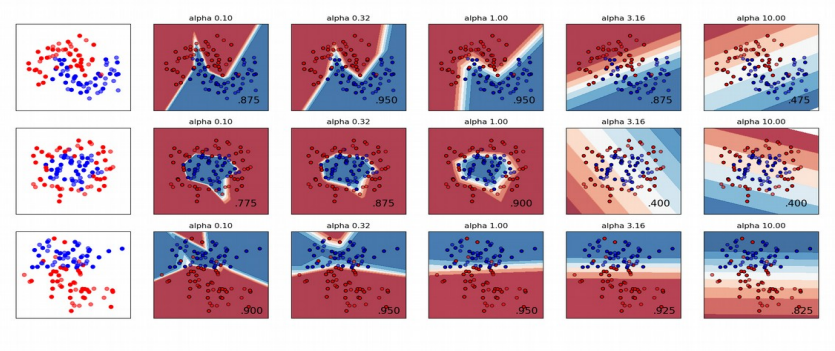


Figure 13 Hidden layer size validation curve ionosphere



For the sonar dataset we see similar results with the number of hidden layers being higher as shown in figure 15. This can be attributed to the fact that this dataset has double the features and datapoints. Furthermore this data set is also very far from being linearly seperable as the two classifications have highly overlapping datapoints adding more complexity to the decision boundaries.

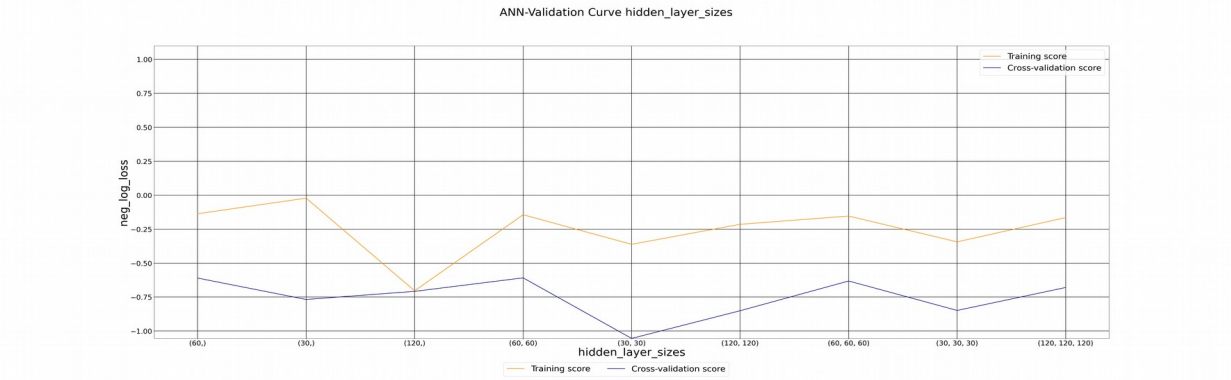


Figure 15 Hidden layer validation curve for sonar

Decision Trees

Decision trees are known to have orthogonal decision boundaries. Thus given that the ionosphere datapoints are overlapping as shown in the scatter plot previously it can be expected that the decision tree model to not perform so well on this dataset as it can make it difficult to determine a decision boundary that will generalize without overfitting. Despite that, the model acheived a balanced accuracy score above 0.8 on the test dataset. One key reason for this could be the preference bias of decision trees being trees with lower depth. Eventually a decision tree with a max depth of 9 was able to generalize better. In addition to optimizing the max depth parameter a form of pruning has been implemented to prevent overfitting. These experiments optimized the the alpha parameter which is associated with the minimal cost complexity pruning. For the ionosphere dataset it came out to 0.013. The alpha value is used to determine which subtree is selected to be dropped. If alpha is 0 then the biggest subtree is chosen as it removes the complexity penalty term. We can see that play out for both datasets in the learning curve in figure 16 and 19 where the training error shows a slight dip with the validation error going up showing the trees ability to generalize better due to pruning. Figure 17 and 18 shows the alpha validation curve for both datasets.

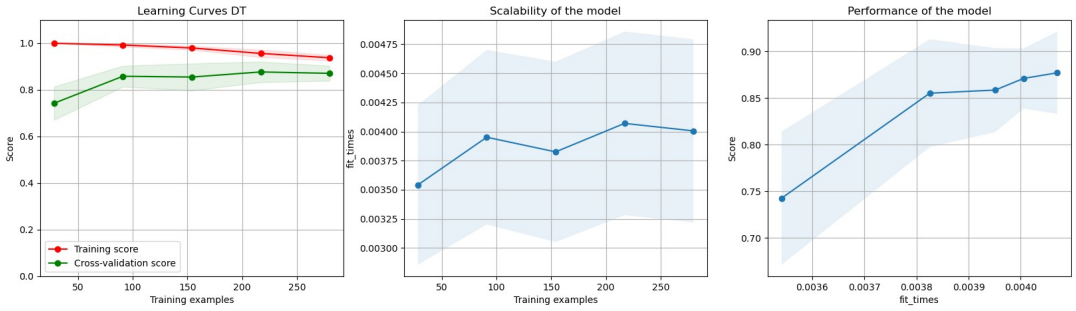


Figure 16 Ionosphere learning curve Decision Tree

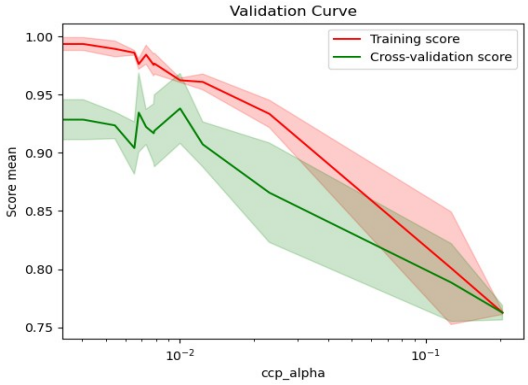


Fig 17 alpha- ionosphere

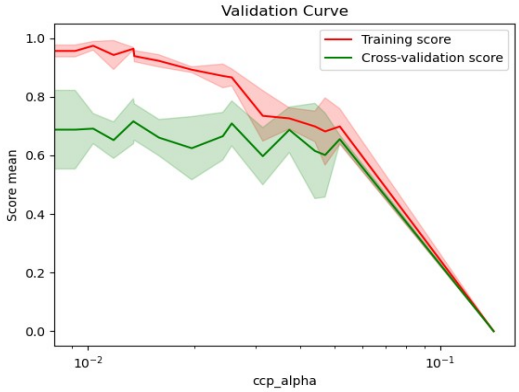


Fig 18 alpha-sonar

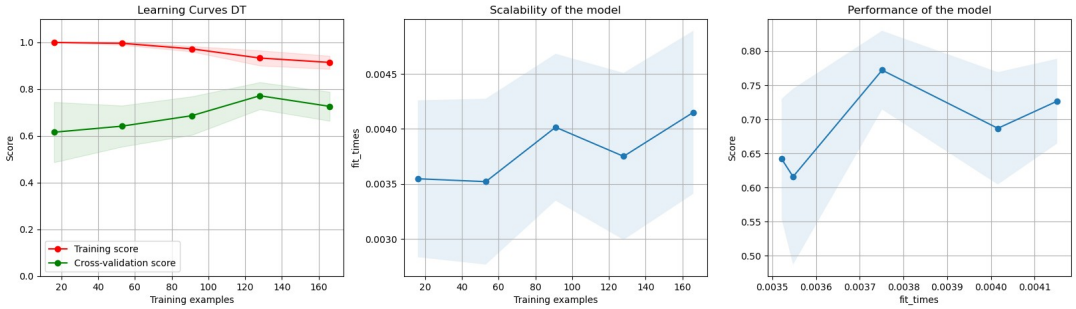


Fig 19 sonar learning curve

Conclusion

The following figures summarizes the results of all the models in both the datasets. The ionoshpere dataset yields models that perform higher than its baseline of 64% whereas the sonar dataset is hovering on the baseline of 88%.

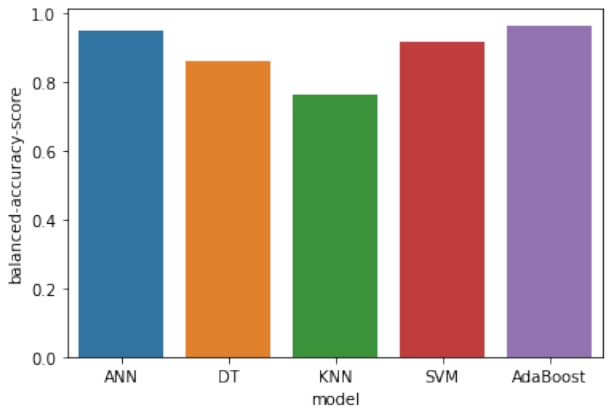


Fig 20 Ionosphere results

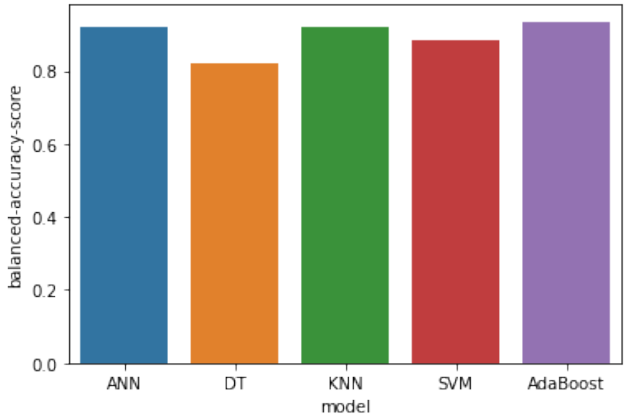


Fig 21 Sonar results

Model	Parameters	f1
ANN	{'solver': 'adam', 'learning_rate_init': 0.004, 'learning_rate': 'adaptive', 'hidden_layer_sizes': (66, 66, 66), 'alpha': 0.0031 'activation': 'relu'}	0.97
DT	{'splitter': 'random', 'max_depth': 9, 'criterion': 'gini', 'ccp_alpha': 0.00757}	0.91
KNN	{'weights': 'uniform', 'n_neighbors': 4, 'metric': 'chebyshev', 'algorithm': 'brute'}	0.87
SVM	{'kernel': 'rbf', 'gamma': 0.01, 'C': 10.0}	0.95
AdaBoost	{'n_estimators': 100, 'learning_rate': 0.16, 'base_estimator__splitter': 'random', 'base_estimator__max_depth': 5}	0.97

Table 3 Ionosphere Results

Model	Parameters	f1
ANN	{'solver': 'adam', 'learning_rate_init': 0.004, 'learning_rate': 'adaptive', 'hidden_layer_sizes': (120, 120, 120), 'alpha': 0.0031, 'activation': 'relu'}	0.91
DT	{'splitter': 'random', 'max_depth': 6, 'criterion': 'gini', 'ccp_alpha': 0.013}	0.8
KNN	{'weights': 'distance', 'n_neighbors': 4, 'metric': 'euclidean', 'algorithm': 'brute'}	0.91
SVM	{'kernel': 'poly', 'gamma': 100.0, 'C': 10.0}	0.87
AdaBoost	{'n_estimators': 5, 'learning_rate': 0.01, 'base_estimator__splitter': 'random', 'base_estimator__max_depth': 5}	0.93

Table 4 Sonar Results

REFERENCES

https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/

https://online.stat.psu.edu/stat508/lesson/11/11.8/11.8.2

https://www.analyticsvidhya.com/blog/2020/10/cost-complexity-pruning-decision-trees/

https://rubikscode.net/2021/07/19/top-23-best-public-datasets-for-practicing-machine-learning/

https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a