Report

Dataset Details:

1. Title: Tic-Tac-Toe Endgame database

2. Source Information

-- Creator: David W. Aha (aha@cs.jhu.edu)

-- Donor: David W. Aha (aha@cs.jhu.edu)

-- Date: 19 August 1991

3. Number of Instances: 958 (legal tic-tac-toe endgame boards)

4. Number of Attributes: 9, each corresponding to one tic-tac-toe square

5. Attribute Information: (x=player x has taken, o=player o has taken, b=blank)

1. top-left-square: {x,o,b}
2. top-middle-square: {x,o,b}
3. top-right-square: {x,o,b}
4. middle-left-square: {x,o,b}
5. middle-middle-square: {x,o,b}
6. middle-right-square: {x,o,b}
7. bottom-left-square: {x,o,b}
8. bottom-middle-square: {x,o,b}
9. bottom-right-square: {x,o,b}
10. Class: {positive,negative}

6. Missing Attribute Values: None

7. Class Distribution: About 65.3% are positive (i.e., wins for "x")

Colab Notebook Link:

https://colab.research.google.com/drive/1mpKPob7q5rNPOGwz6-TjzdU_PdQ8VAo1 • The scoring parameter for tuning the hyper parameters is Precision_macro.

Algorithms - Best Parameters

Algorithm	Best Parameters
Decision Tree	{'max_depth': 1, 'min_impurity_decrease': 0.05, 'min_samples_leaf': 1, 'min_samples_split': 2}
Neural Net	{'activation': 'tanh', 'hidden_layer_sizes': 50, 'learning_rate': 'adaptive', 'max_iter': 5000}
Support Vector Machine	{'C': 1000, 'gamma': 0.001, 'kernel': 'rbf', 'max_iter': 1000, 'random_state': 1}
Gaussian Naive Bayes	{‘priors’: None}
Logistic Regression	{'C': 1, 'fit_intercept': 'True', 'max_iter': 100, 'multi_class': 'ovr'}
k-Nearest Neighbors	{'algorithm': 'kd_tree', 'n_neighbors': 30, 'p': 2, 'weights': 'distance'}
Bagging	{'max_features': 4, 'max_samples': 30, 'n_estimators': 20, 'random_state': 4}
Random Forest	{'criterion': 'entropy', 'max_depth': 5, 'max_features': 3, 'n_estimators': 50}
AdaBoost	{'algorithm': 'SAMME.R', 'learning_rate': 1.5, 'n_estimators': 150, 'random_state': 1}
Gradient Boosting	{'learning_rate': 1.0, 'loss': 'exponential', 'min_impurity_decrease': 0.05, 'n_estimators': 50}
XGBoost	{'booster': 'gbtree', 'learning_rate': 1.0, 'min_child_weight': 1, 'n_estimators': 150}

Algorithms - Score Report

Algorithm	Average Precision	Average Recall	Accuracy Score
Decision Tree	0.41	0.64	0.64
Neural Net	0.81	0.81	0.81
Support Vector Machine	0.75	0.75	0.75
Gaussian Naive Bayes	0.74	0.71	0.71
Logistic Regression	0.66	0.68	0.68
k-Nearest Neighbors	0.79	0.74	0.74
Bagging	0.78	0.75	0.75
Random Forest	0.87	0.85	0.85
AdaBoost	0.96	0.95	0.95
Gradient Boosting	0.98	0.97	0.97
XGBoost	1.00	1.00	1.00

- XGBoost performed the best from the given group of algorithms achieving an accuracy of 1.00 which means all the data items are correctly classified into their respective classes. Gradient Boosting and AdaBoost followed XGBoost by achieving accuracy scores of 0.97 and 0.95 respectively.

- XGBoost, Gradient Boosting and AdaBoost performed the best in the classification task. All the three algorithms are Boosting algorithms which generally train weak learners by sequentially correcting their predecessors to make them strong learners. As this sequence of corrections improve the performance of the learner, these algorithms performed the best from the given group of algorithms.

- In order to improve the results, various data pre-processing techniques can be applied on the dataset such as MinMaxScaler, Normalization, etc. in order to make data representation more suitable for the learning task.