Machine Learning - Assignment 2

Group B

Ana Raquel Maceiras
Hélder Vieira
Miguel Tavares
Rui Vieira
FCUP

June 14, 2021

Overview

1. Introduction
2. Methods and Hyperparameters
3. Dataset Generation
4. Experimental Setup
5. Results
6. Conclusion

Introduction

- In supervised Machine Learning there are many methods for finding the best approximation to the unknown function that defines the data.
- Some method(s) may outperform other(s) in some specific scenarios

Objectives

Given a list of Machine Learning models (further referred as pool), the aim of this project is to:

- For each method (or subset of methods) generate datasets in which the classification obtained outperforms the other methods in the pool, using several performance metrics
- When possible hyperparameter tuning is used to select the optimal hyperparameters for each model
- Try to justify why each method is 'better' than the others in that specific scenarios, highlighting models' strengths and weaknesses

Methods and Hyperparameters

Methods	Predefined Parameters	Tuned Parameters
Logistic Regression	NA	None
Linear Discriminant Analysis (LDA)	NA	None
Quadratic Discriminant Analysis (QDA)	NA	None
Random Forest	NA	criterion min_samples_split min_samples_leaf min_impurity_decrease
Decision Trees	NA	min_samples_leaf min_samples_split max_depth
Support Vector Machines	linear rbf poly	C
Multi-Layer Perceptron	tanh ReLu	hidden_layer_sizes solver alpha learning_rate batch_size

NA - Not Applicable

Dataset Generation

Datasets were mostly created using functions from the scikit-learn library.

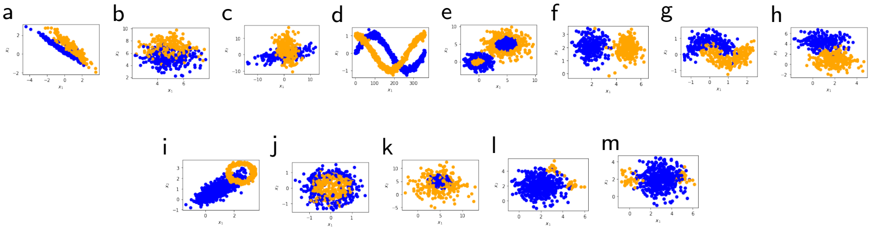

Figure: Datasets used in this study

Experimental Setup

1. All the methods will be applied to classify all the created datasets. In this process, a grid search CV was used to optimize the models that require hyperparameter tuning, with F1 score as selection metric.
2. Train-test split was applied (holdout 20%) and with the model fitted, accuracy, F1-score and ROC-AUC were computed and analyzed. A plot of the decision boundary was obtained to allow a better assessment of the model performance.
3. Lastly, all the training times were compared.

Logistic Regression

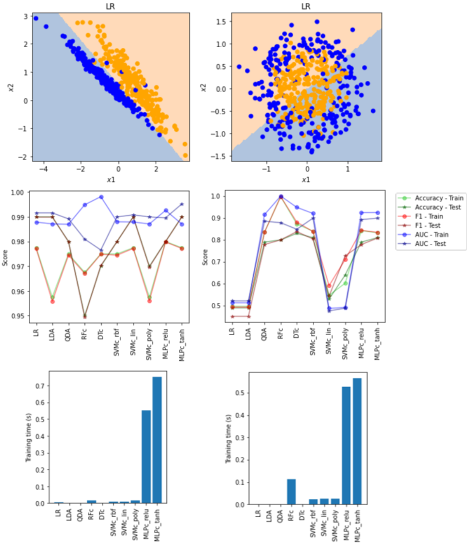

Figure: Logistic Regression performance for the generated datasets.

Linear Discriminant Analysis (LDA)

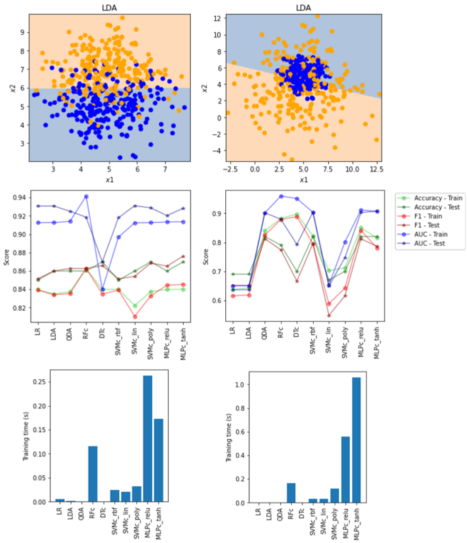

Figure: LDA performance for the generated datasets

Quadratic Discriminant Analysis (QDA)

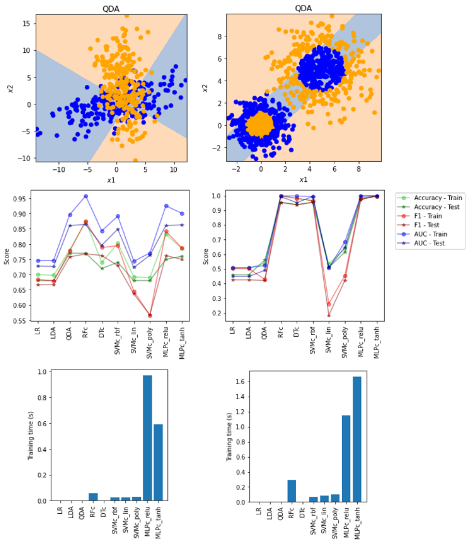

Figure: QDA performance for the generated datasets

Random Forest

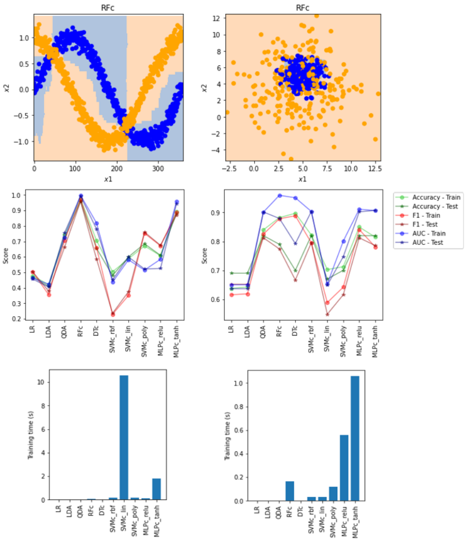

Figure: Random Forest performance for the generated datasets

Decision Trees

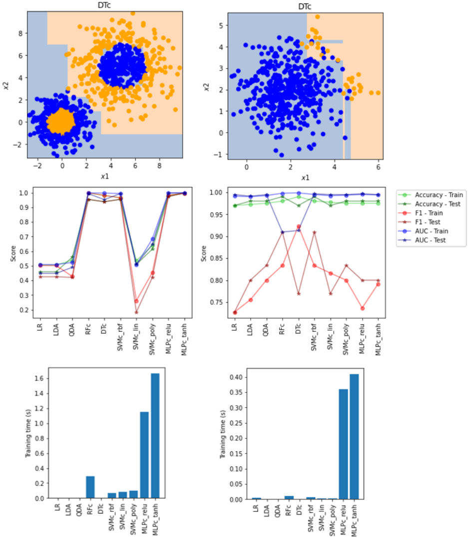

Figure: Decision Trees performance for the generated datasets

Support Vector Machines: Kernel - linear

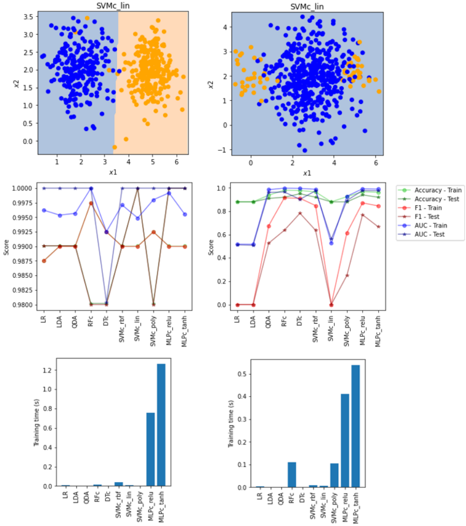

Figure: SVM with linear kernel performance for the generated datasets

Support Vector Machines: Kernel - radial basis

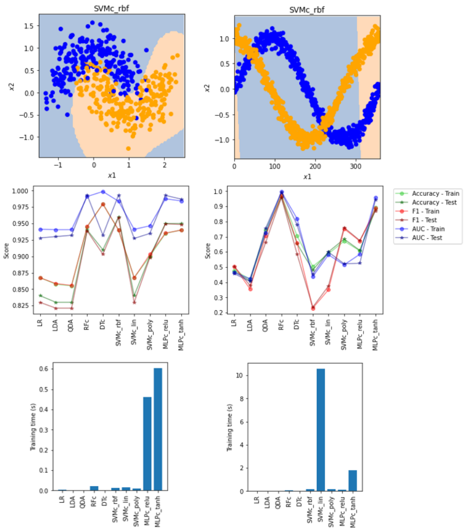

Figure: SVM with radial basis kernel performance for the generated datasets

Support Vector Machines: Kernel- polynomial

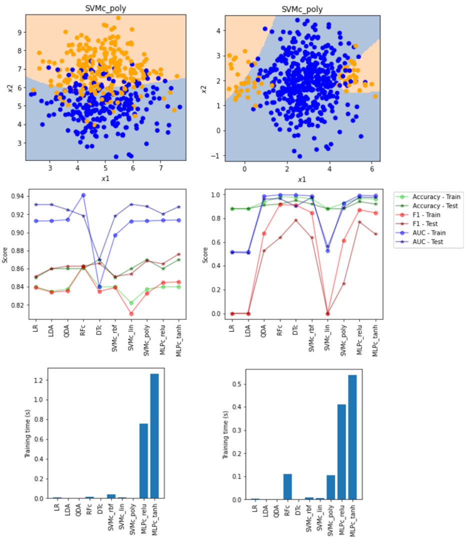

Figure: SVM with polynomial kernel performance for the generated datasets

Multilayer Perceptron: Activation - hyperbolic tangent function

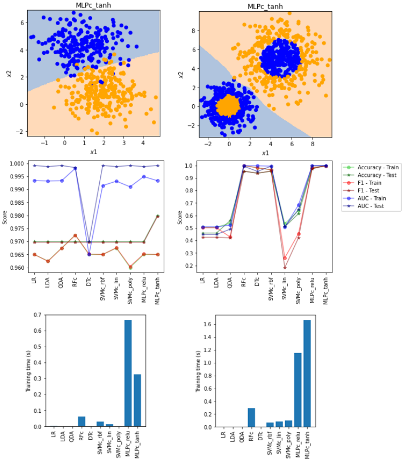

Figure: MLP with hyperbolic tangent (tanh) activation function performance for generated datasets

Multilayer Perceptron: Activation - rectified linear unit function

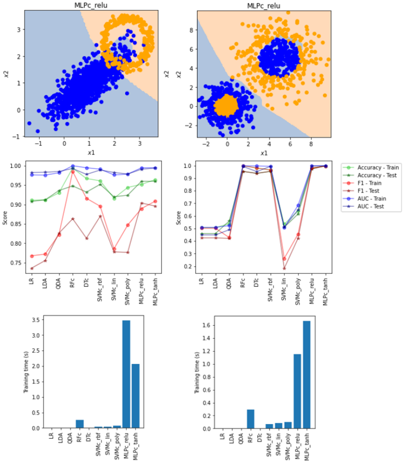

Figure: MLP with rectified linear unit (relu) activation function performance for datasets

Conclusion

Generally, for each method in the pool we have successfully created and presented a scenario in which for each method the approximation produced was significantly better than the majority of the methods, highlighting the models strengths and weaknesses.

The End