

**Ain Shams University**

**Faculty of Computer and Information Science**

**Scientific Computing department**

**Ain shams university**

**Faculty of computer and information science**

**Bioinformatics department**

**Project Title**

**Airline Ticket Price Prediction**

**PHASE 2**

**Modeling Phase**

**- We use 8 models for this project:**

**Text

Description automatically generated**

**General Notes:**

The hypered parameters in the models effect on the accuracy of the model so it should be picked carefully.

**AdaBoost Classifier:**

An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

**The results of the model on the dataset:**

Text

Description automatically generated with medium confidence

**GB Classifier:**

GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n\_classes\_ regression trees are fit on the negative gradient of the loss function, e.g. binary or multiclass log loss. Binary classification is a special case where only a single regression tree is induced.

**The results of the model on the dataset:**

A screenshot of a computer

Description automatically generated with low confidence

**Bagging Classifier:**

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

**The results of the model on the dataset:**

A screenshot of a computer

Description automatically generated with low confidence

**RF Classifier:**

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max\_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

**The results of the model on the dataset:**

A screenshot of a computer

Description automatically generated with low confidence

**XGB Classifier:**

The XGBoost or Extreme Gradient Boosting algorithm is a decision tree based machine learning algorithm which uses a process called boosting to help improve performance. Since it’s introduction, it’s become of one of the most effective machine learning algorithms and regularly produces results that outperform most other algorithms, such as logistic regression, the random forest model and regular decision trees.

**The results of the model on the dataset:**

A screenshot of a computer

Description automatically generated with low confidence

**DT Classifier:**

re a non-parametric supervised learning method used for [classification](https://scikit-learn.org/stable/modules/tree.html#tree-classification) and [regression](https://scikit-learn.org/stable/modules/tree.html#tree-regression). The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

**The results of the model on the dataset:**

A screenshot of a computer

Description automatically generated with low confidence

**HGB Classifier:**

This estimator is much faster than GradientBoostingRegressor for big datasets

This estimator has native support for missing values (NaNs). During training, the tree grower learns at each split point whether samples with missing values should go to the left or right child, based on the potential gain. When predicting, samples with missing values are assigned to the left or right child consequently. If no missing values were encountered for a given feature during training, then samples with missing values are mapped to whichever child has the most samples.

**The results of the model on the dataset:**

A screenshot of a computer

Description automatically generated with low confidence

**HGB Classifier:**

This class implements a meta estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

**The results of the model on the dataset:**

A screenshot of a computer

Description automatically generated with medium confidence