

**Computer machine learning**

**COMP4388**

Machine learning

Dr: [Radi Jarrar](https://ritaj.birzeit.edu/bzu-msgs/type?mttid=104&classid=183185)

3ed Assignment

* **Name: member:** Amir Eleyan.
* **Student ID:** 1191076.
* **Date:** Tuesday, January 18, 2022

# **Part one:**

## **Random Forest:**

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees like CART, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

Put simply: random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems.

The random forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, there's no need to combine a decision tree with a bagging classifier because you can easily use the classifier-class of random forest. With random forest, you can also deal with regression tasks by using the algorithm's regressor.

Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

## **Boosting:**

Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to minimize training errors. In boosting, a random sample of data is selected, fitted with a model, and then trained sequentially that is, each model tries to compensate for the weaknesses of its predecessor. With each iteration, the weak rules from each individual classifier are combined to form one, strong prediction rule.

Ensemble learning:

Ensemble learning gives credence to the idea of the “wisdom of crowds,” which suggests that the decision-making of a larger group of people is typically better than that of an individual expert. Similarly, ensemble learning refers to a group (or ensemble) of base learners, or models, which work collectively to achieve a better final prediction. A single model, also known as a base or weak learner, may not perform well individually due to high variance or high bias. However, when weak learners are aggregated, they can form a strong learner, as their combination reduces bias or variance, yielding better model performance.

Boosting algorithms can differ in how they create and aggregate weak learners during the sequential process. Three popular types of boosting methods include:

Adaptive boosting or AdaBoost, Gradient boosting, Extreme gradient boosting or XGBoost

* Benefits of boosting: ease of implementation, reduction of bias.
* Challenges of boosting: overfitting, intense computation.

## **Random forest Vs. XGBoost:**

|  |  |
| --- | --- |
| ***Random Forest*** | ***XGBoost*** |
| The decision trees are built independently so that if there are five trees in an algorithm, all the trees are built at a time but with different features and data present in the algorithm. | XGBoost builds one tree at a time so that each data pertaining to the decision tree is taken into account and the data is filled if there are any missing data. This helps to work with gradient algorithms along with the decision tree algorithm for better results. |
| Once all the decision trees are built, the results are calculated by taking the average of all the decision tree values. This makes us wait for building all the decision trees to the end and the cumulative results are taken into account. | While the model is building the decision trees, the results are calculated and added up for the next tree and hence the gradient of the results is considered. This helps to get an idea of the results even if the decision trees take time |
| Random Forest has many trees with leaves of equal weight so that high accuracy and precision can be obtained easily with the available data. This makes it's easy to add more features to the data and look at how it performs for all the data given to the algorithm. | XGBoost does not account for the number of leaves present in the algorithm. If the model predictability is not good, the algorithm performs better with more leaves in the decision tree. This improves the bias and the results completely depend on the data present in the algorithm. |

# **Part two:**

Procedure: First of all, read dataset and remove spaces from it, describe the data, min max, mean etc. Checking if there is no null value and if there are no rows with zero value, calculate the correlation between independent and target variable, drop the “ID” class; because has very low correlation with “status” variable. Shuffling the data set and split them to 80% for training and 20% for testing. Define the Random Forest model and these values were the result of building this model.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Measure name** | **Model score** | **Accuracy** | **Precision** | **Recall** | **F-score** | **MSE** | **Bias** | **Variance** |
| **Result** | 100.0% | 99.0% | 100.0% | 98.0% | 99.0% | 0.015 | 0.006 | 0.009 |

After that, define the C4.5 model, and these values were the result of building this model.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Measure name** | **Model score** | **Accuracy** | **Precision** | **Recall** | **F-score** | **MSE** | **Bias** | **Variance** |
| **Result** | 100.0% | 99.0% | 100.0% | 98.0% | 99.0% | 0.023 | 0.007 | 0.016 |