**Section A**

**1.** Extreme Gradient Boosting (XGBoost)

**2.** XGBoost is one of the popular machine learning algorithms these days. “Regardless of the type of prediction task at hand; regression or classification” [4] (Data Camp). Extreme Gradient Boosting “belongs to a family of boosting algorithm and uses the gradient boosting algorithm framework at its core” (Data camp). Let us talk about what is booting and how it works? Boosting is a sequential technique uses in Machine Learning that convert a weak learner to a strong one. “It combines a set of weak learners and delivers improved prediction accuracy. At any instant t, the model outcomes are weighed based on the outcomes of previous instant t-1” (Data camp) [ 4]. XGBoost has an in-built routine to handle missing values.

Extreme Gradient Boosting is a “scalable machine learning system for tree boosting” (Che, Tianqi & Guestrin, Carlos) [1]. The scalability of the algorithm is made possible because it uses a novel tree learning algorithm for handling sparse data, and parallel and distributed computing to make learning faster. It uses an approximation of the exact greedy algorithm to find the best possible split for the tree. It is an ensemble learning method, which is computationally fast compared to other gradient boosting algorithms (Brownlee, Jason) [2]. It is also excellent when used on structured or tabular datasets for classification or prediction problems [2].

The algorithm has a multitude of uses cases such as store and ad click prediction, and high energy physics event, malware, and web text classification.

**3.** **3.** for the regularized objective function

is for the additive model for tree boosting

**4.** (a) XGBoost can only handle numerical data, however it can take dummy numerical values to replace categorical variables. (Brownlee, Jason) [3]

(b) Some of the disadvantages of XGBoost is that the algorithm is not well documented [6]. Even though there are some API and examples, there are not very good.

(c) Cannot extract the linear combination of features [7]

**5.** It cannot be used for feature engineering or hyper parameter tuning. It is also possible to overfit the model to the test data. (Not a lot of information)

For section (a) and (b), we could not find solid information to back up our research.

**6.** (a) xgboost, pandas, matplotlib sklearn. metrics, sklearn.cross\_validation

(b) .Dmatrix(), .fit(), .predict(), .plot\_importance(), .plot\_tree(), accuracy\_score(), and .cv()

**7.** (a) In order to evaluate the performance of the model, we use the accuracy score method, which will compare the predicted values to the actual values for the dataset. This will return a score from 0 to 1, where 1 represents a perfect fit.

(c) accuracy\_score()

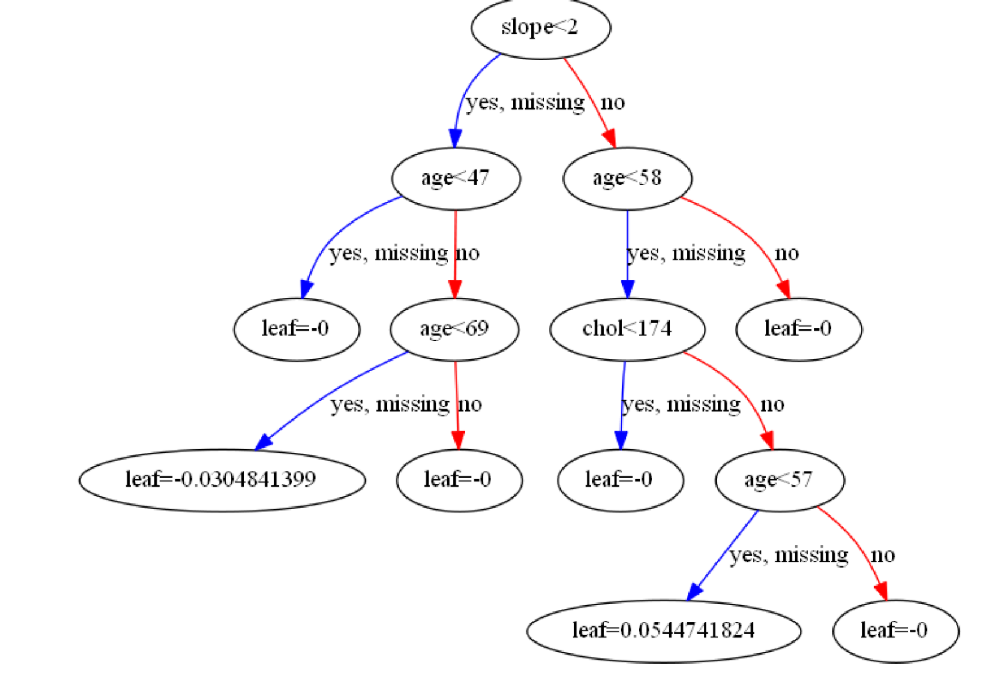
**Section B**

**8.** [This data set can be found at this link https://www.kaggle.com/ronitf/heart-disease-uci](file:///C:\Users\diall\Downloads\This%20data%20set%20can%20be%20found%20at%20this%20link%20%20%20%20https:\www.kaggle.com\ronitf\heart-disease-uci)

The dataset is one which contains 14 columns and 303 rows on heart disease related data. The goal of the dataset is to detect the presence of heart disease in patients, represented as 0 or 1. The variables are age, sex, trestbps (resting blood pressure), chol (serum cholesterol), restecg (resting electrocardiographic results), fbs (fasting blood sugar), cp (chest pain type), thalach (maximum heart rate achieved), exang (exercise induced angina), the slope of the peak exercise, ca (number of major vessels), oldpeak, thal, and target.

**9.** After applying the algorithm and receiving the accuracy score, we performed a K-fold cross validation to train the model to the original dataset used in training and validation. This was different from any of the previous algorithms we had come across, and it appears to have made a more robust model.

**10.**

****

*Figure 1: Visualization of Boosting Trees*

We are seeing every week a little bit of improvement in terms of researching and implementing a new algorithm.

Even though XGBoost is very powerful and popular Machine learning algorithm, we suggest for future research to look at its potential limitations and improve documentation.

**Section C**

**References:**

[1] <https://www.kdd.org/kdd2016/papers/files/rfp0697-chenAemb.pdf>

[2] <https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/>

[3] <https://machinelearningmastery.com/data-preparation-gradient-boosting-xgboost-python/>

[4] <https://www.datacamp.com/community/tutorials/xgboost-in-python>

[5] <https://xgboost.readthedocs.io/en/latest/parameter.html#general-parameters>

[6] <http://danielhnyk.cz/how-to-use-xgboost-in-python/>

[7] <https://acadgild.com/blog/xgboost-python>