**Project Proposal**

**Proposed Topic:**

**Boosting Ensemble Learning: A comparison of XGBoost to Decision Trees and Random Forest for Supermarket Sales Prediction**

**Motivation**

This project intends to examine and evaluate the performance of eXtreme Gradient Boosting popularly known as XGBoost, a tree-based ensemble machine learning algorithm based on boosting, on a supermarket sales dataset. It also compares its performance to a baseline linear regression model and two other tree-based algorithms namely; Decision Trees, and Random Forest.

Boosting is an ensemble learning technique that seeks to minimize training errors by building a powerful classifier out of a number of weak classifiers [1], [2], [8]. Firstly, a model is built on the training dataset, then another model is built on top of that to try to correct the errors of the previous model, this procedure is repeated until either the maximum number of models are added or training is completed [3], [5]. Similarly, XGBoost which stands for eXtreme Gradient Boosting was specifically designed to improve speed and performance, it is now a popular and efficient open-source implementation of the gradient boosted trees algorithm.

Over the past few years, XGBoost has gained traction in the data science/data mining ecosystem by helping individuals and teams on Kaggle win almost every structured tabular data competition as well as by being actively utilized by multiple organizations such as Delivery Hero, Compile Inc, BagelCode, BlaBlaCar etc. [3]. Its opensource characteristics has also resulted in a rising number of data scientists globally that are actively contributing to improving the codebase. For this project, the XGBoost algorithm is applied to build a predictive model that takes as input a set of independent variables describing a supermarket’s product, and outputs the supermarket product’s sales. The trained model would then be evaluated using the Root Mean Squared Error (RMSE) metric, and its performance compared with a baseline linear regression model and two other tree-based algorithms, namely: decision trees and random forest.

**Problem Definition**

A recent publication in the just concluded conference on Neural Information and Processing Systems (NeurIPS) confirmed that traditional machine learning models still outperform state of the art deep learning methods on tabular data. As excellent as neural networks are at classifying images, detecting objects in videos and natural language processing, it doesn’t beat tree-based models at tabular datasets. As such, I am interested in evaluating the performance of a tree-based ensemble technique, specifically the XGBoost model on a supermarket sales dataset, and comparing its RMSE to those obtained from decision tree and random forest algorithms which are also tree-based machine learning models. For this project scope, the input data is a supermarket sales dataset that will be obtained from Kaggle. This data will first be fed to a linear regression model to get a baseline Root Mean Square Error (RMSE) value, then to a decision tree and a random forest for an even better RMSE score, and finally we train an XGBoost model and compare the general predictive prowess and generalizability of the three models on data which it hasn’t seen.

**Methods**

**Data Preparation**

Before going ahead to build a model it is standard practice to at least inspect your dataset variables, to check for anomalies or outliers, this process is called exploratory data analysis (EDA). EDA is an approach for seeing what the data can tell us beyond the formal modelling or hypothesis testing task. The supermarket sales dataset contains 13 variables including the target variable, and of these 13 variables, 5 are continuous while the rest are categorical with 4990 unique data points. So we have to handle the categorical columns as some machine learning algorithms are not sophisticated enough to handle categorical algorithms by themselves.

**Dealing with missing data**

There are a number of techniques for dealing with missing data which can be subdivided into two broad methods namely:

1. Imputation
2. Removal of the data

On the one hand, the imputation method involves developing reasonable guesses for the missing rows, this could include replacing missing cells with its associated measures of central tendency e.g average, median, mode, or using some more sophisticated imputer method such as Regression imputation, KNN imputation, etc. On the other hand, the removal of data method essentially implies dropping the missing columns from the dataset and training the model with the rest of the data.

There are a number of techniques to choose from but for this project, we handle missing data using the imputation method by replacing empty cells with some arbitrary number “-999”,

**Evaluation Criteria**

The evaluation criterion will be the Root Mean Squared Error (RMSE), which is the standard deviation of the residuals. The RMSE is a measure of how dispersed the residuals are, or how consolidated the data is around the line of best fit in a regression model [11].

**Model Development**

Using sci-kit learn, a python based open-source library for data analysis and the gold standard for machine learning, a baseline linear regression model was built with no hyperparameter tuning, and the corresponding RMSE score was then obtained and used as a baseline score. A decision tree and a random forest model was also built and their respective RMSEs and predictive performance were compared to that of XGBoost. The toolset used includes: Pandas, Sci-kit learn, Jupyter notebooks, etc.

**Results**

**Hyperparameter selection**

**Conclusion**

**References**

[1] <https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/>

[2] <https://www.nvidia.com/en-us/glossary/data-science/xgboost/>

[3] <https://stackshare.io/xgboost>

[4] <https://www.ibm.com/cloud/learn/boosting>

[5] <https://www.mygreatlearning.com/blog/xgboost-algorithm/>

[6] <https://www.kaggle.com/competitions/dsn2018intercampus/overview/evaluation>

[7] <https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.html>

[8] <https://www.nvidia.com/en-us/glossary/data-science/xgboost/>

[10] <https://stackshare.io/xgboost>

[11] <https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/>

[xx] Cite the eda here https://en.wikipedia.org/wiki/Exploratory\_data\_analysis

<https://arxiv.org/abs/2207.08815> tree based model outperform cite

**Victor Irekponor | Ph.D. student**

**119079201**