**Data Science - Assignment 1**

**Submitted by**: Sneha Mathummal Parapurath, Deepa Krishnan, Punnosse Kozhuppakalam Thomas, Loshma Latha Babu

**Introduction**

The Assignment is to familiarize with the machine learning tasks like classification and regression. Classification is a supervised machine learning task to predict a categorical target variable. Here we are predicting an individual’s income is >50K or <=50K based on various features. Regression is another type of supervised learning technique to predict the discrete numerical value based on previous outcomes. In this task, we have predicted the median house value based on various features.

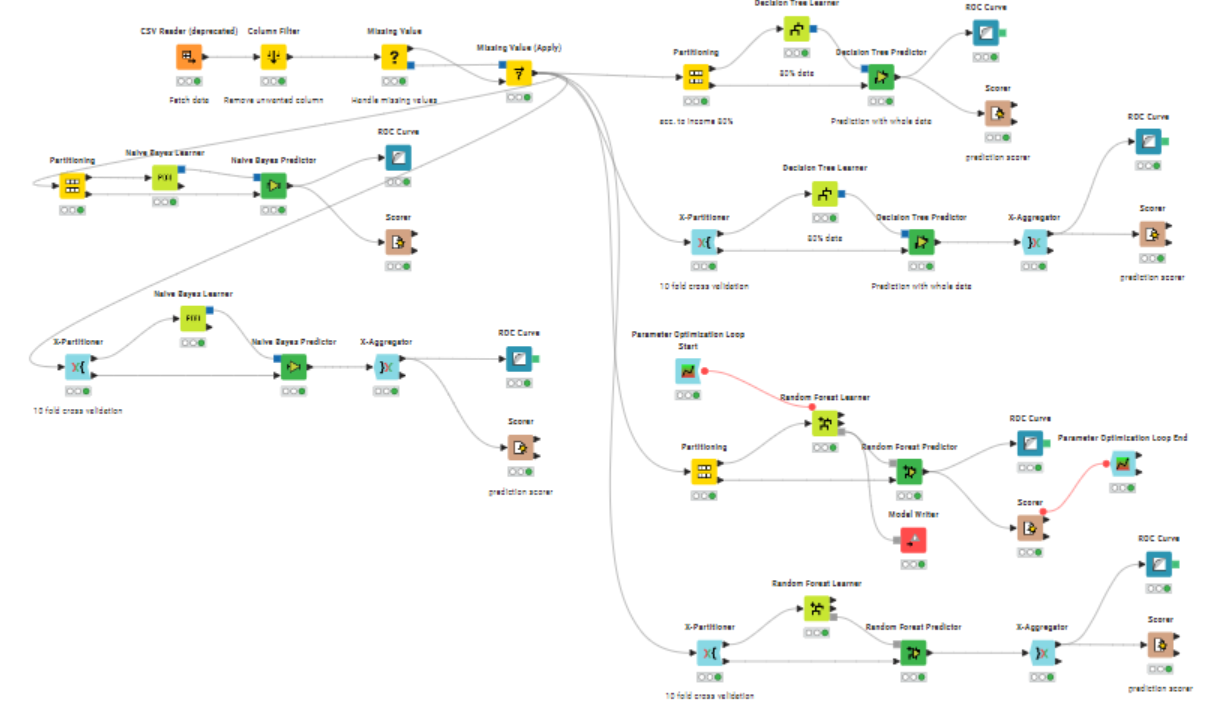
**Classification**

**Data pre-processing and feature selection**

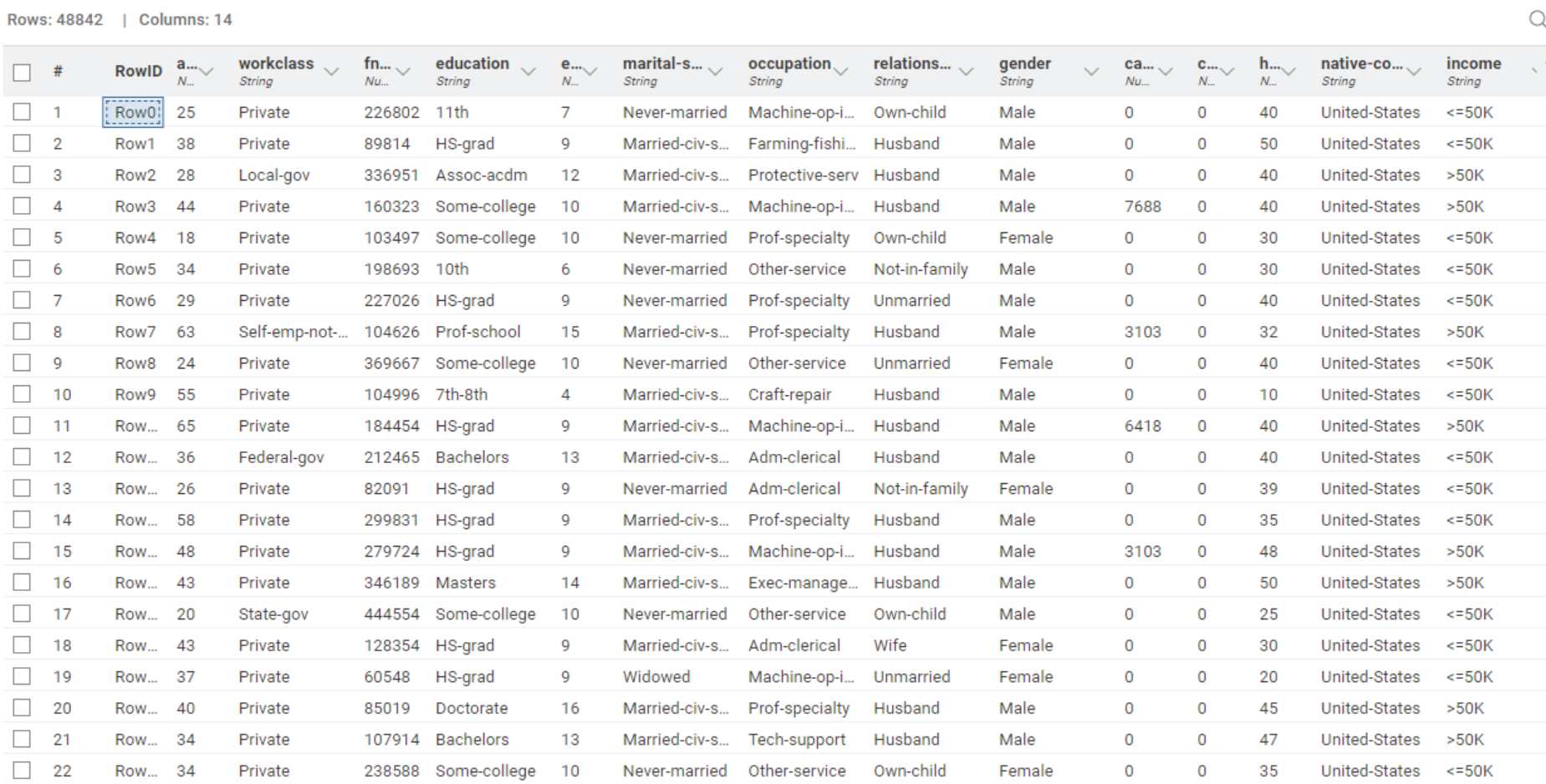
This step includes loading the data, cleaning the data, missing values handling, and converting categorical data into numerical values if required for the specific model.

Adult\_update.csv dataset provides information on the attributes of a group of people. Feature of the data set are Age, fnlwt, Work-class, Education, educational-num, Marital Status, Occupation, Relationship, Gender, Capital-gain, Capital-loss, Hours-per-week, Native-country, Income. This dataset consists of both numerical and categorical data, but we didn’t transform categorical data into numerical data as we have used only classifiers which can predict despite of the data type.

We have handled missing values using missing values node. We are replacing the string value with the most frequent value and numeric value with the mean of that parameter.



We have used all features for this assignment to predict the income. We tried removing some features and predicted the model, but it was not helpful to increase the accuracy.



**Implementation**

We have implemented the model using three classifiers. Here we have used:

* Decision Tree
* Random Forest
* Naïve Bayes

The data was split into training and testing set in the percentage (80:20) for model building by stratified sampling based on income and some random seed value. We have used these algorithms because these models can use categorical data directly and hence no need of Normalizer nodes. We have used learner and predictor nodes according to the algorithm chosen. Predictor node helps to predict between two classes of income which is <=50k or >50k. After training a model we have evaluated the accuracy and area under the curve values using Scorer and ROC Curve nodes. There are also other criteria we can choose to compare the classifiers like Cohen’s kappa, error percentage etc.

ROC Curve visualises the performance comparing the actual value and predicted value. If the value is closer to 1, better the model. Scorer calculates the accuracy, error, and Cohen’s kappa. Accuracy which we have chosen as the criteria is the ratio of correctly predicted observation to the total observation.

A graph with a line

Description automatically generated

ROC Curve of Decision Tree Model

A screenshot of a computer

Description automatically generated

Screenshot of Decision Tree learner node

**Comparison**

We got the better AUC and accuracy for the Random Forest model when compared to other two Classifier models. We have also incorporated 10-fold validation technique for all the model and observed that it has led to smaller accuracy. We have got 86.795 % for the Random forest model, while for other models it was 82.629% (Decision Tree) and 83.54%(Naïve Bayes) respectively.

A screenshot of a computer

Description automatically generated

Confusion matrix of Naïve Bayes model

A graph with a line

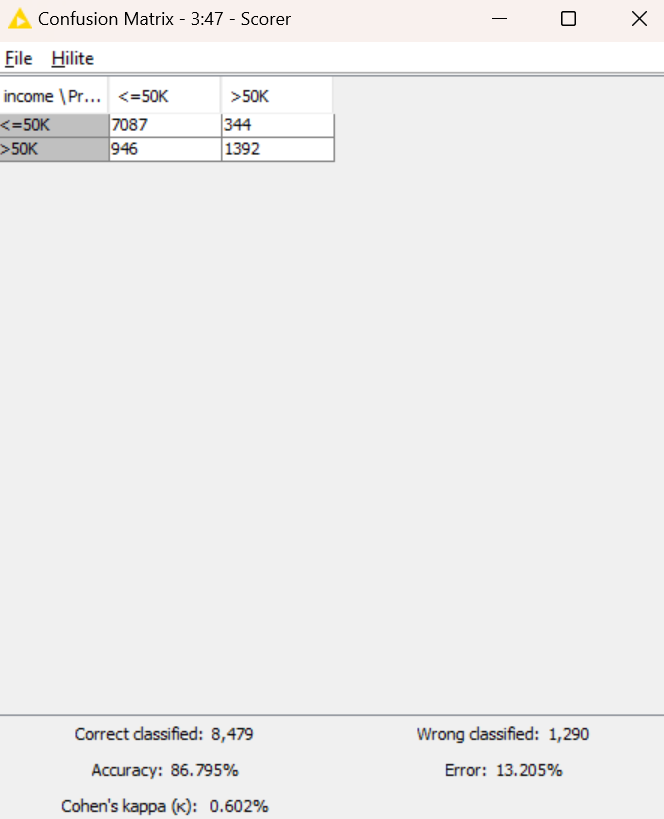
Description automatically generated

ROC Curve of Random Forest model

**Parameter Optimization and Saving the model**

It is an important step in machine learning task as it can enhance the performance of a model. We have used Parameter optimization Loop to get a structured approach to explore various combinations of parameters. This will help to identify which optimal setting can improve the accuracy of a model. We have used Parameter optimization loop only for the Random Forest model as it gave us better model when compared to other models.

Random Forest Learner allow us to change many parameters in KNIME like the max number of levels, no of models, node size, child size etc. We have changed several parameters and have not observed much difference in the performance of the model, but changing the tree depth had changed improved the performance of metrics to a certain extent and got new accuracy to 86.9%. Hence, we can say that the tree depth in a random forest model is a crucial hyperparameter that influences the trade-off between model complexity and generalization. With the help of looping concept in KNIME we have trained the model and obtained the best parameter with better accuracy than before.



Confusion matrix of Random Forest

After optimization, we got the best parameter value as 24.

A white rectangular object with a black stripe

Description automatically generated

We have saved the model using Model Writer node and created a new Workflow to run entire data to see the prediction. Model Reader node has been used to get the saved model and fed the model with entire dataset.

A diagram of a computer model

Description automatically generated

Prediction generation for entire dataset

A screenshot of a computer

Description automatically generated

Table with prediction for entire dataset

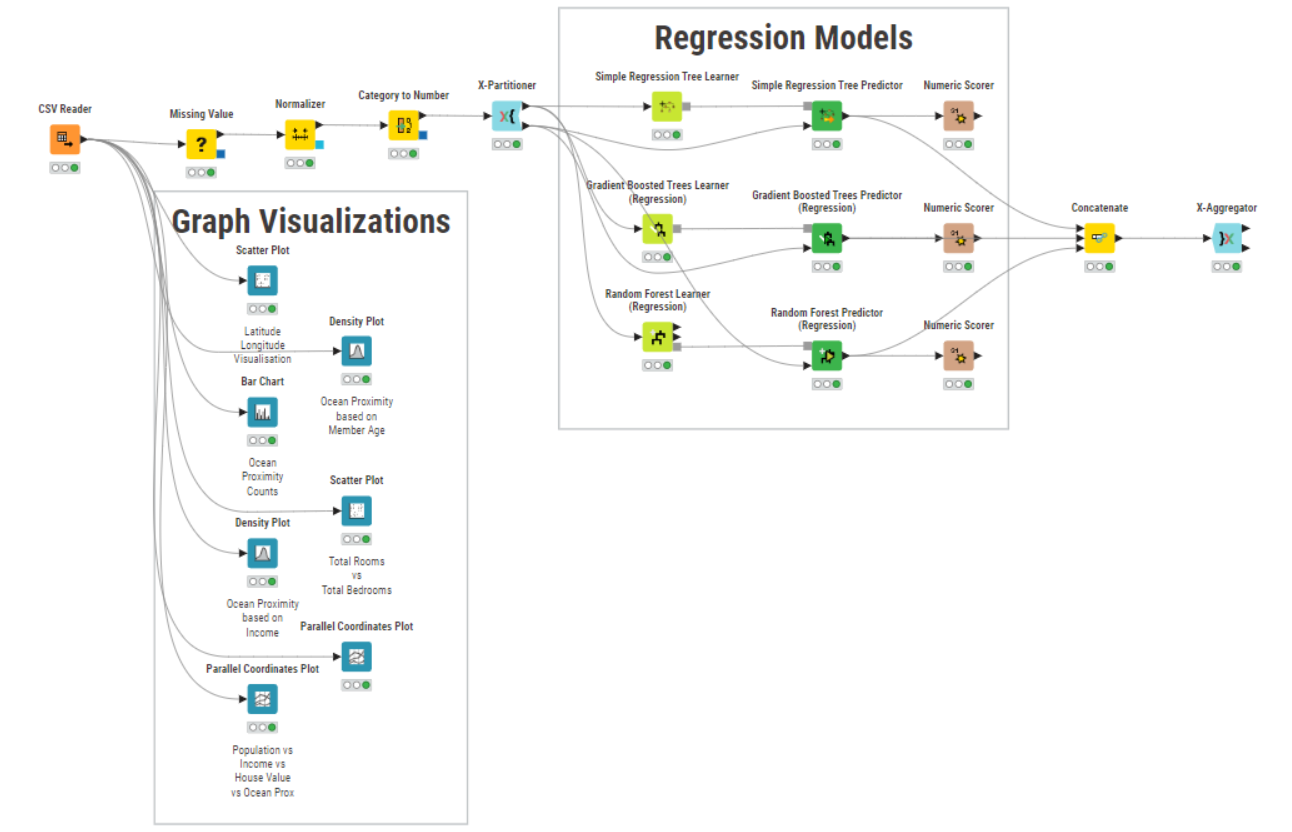
**Regression**

**Data pre-processing and feature selection**

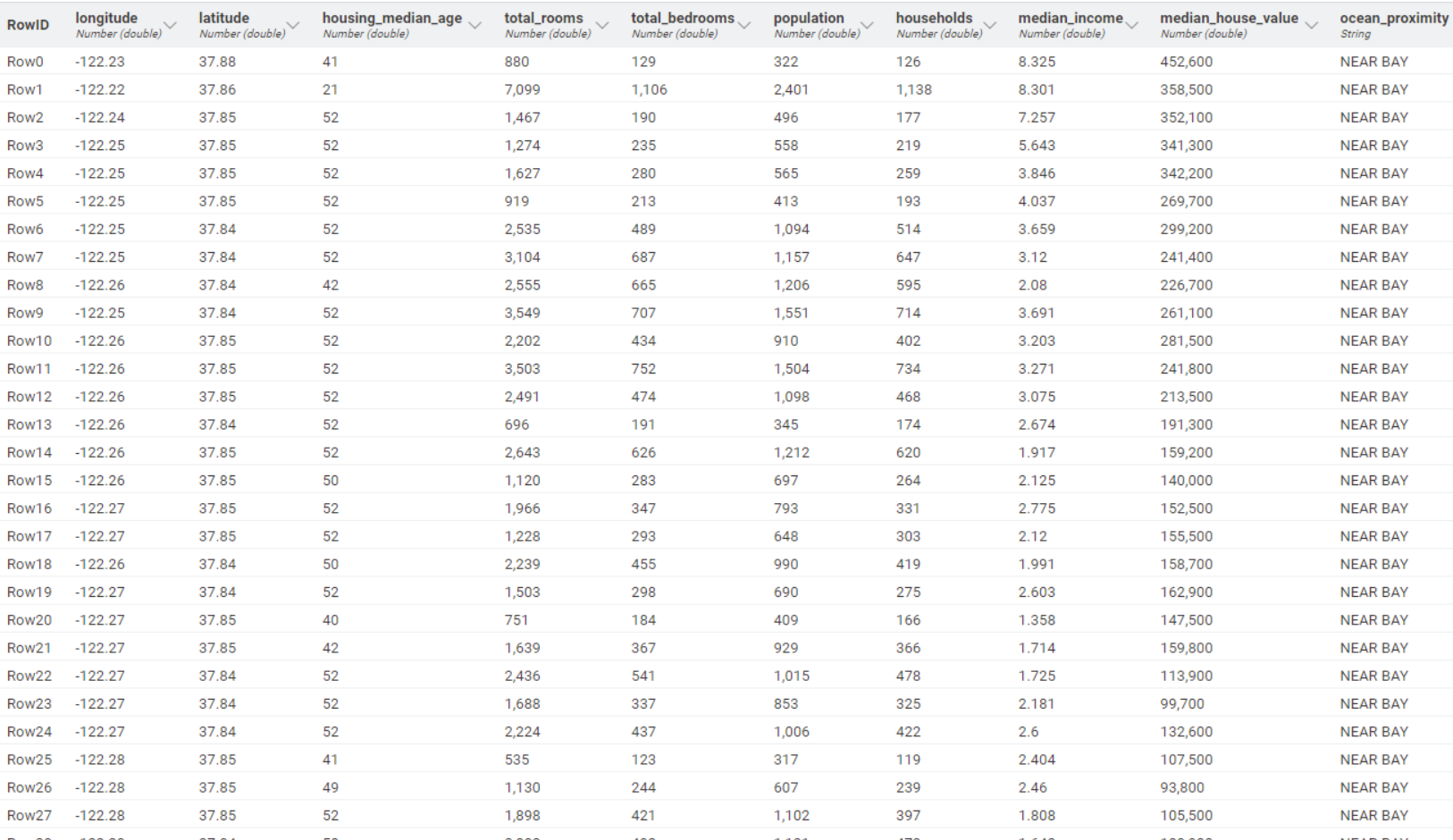
This step includes loading the data, missing values handling, normalizing and converting categorical data into numerical values if required for the specific model.

housing.csv dataset provides information on the attributes of a set of houses. Feature of the data set are **longitude**, **latitude**, **housing\_median\_age**, **total\_rooms**, **total\_bedrooms**, **population**, **households**, **median\_income**, **median\_house\_value**, and **ocean\_proximity**. This dataset consists of mostly numerical data with only the **ocean\_proximity** parameter having categorical data and we transform this single categorical data into numerical values for better regression results.

We have handled missing values using missing values node. We are replacing the string value with the most frequent value and numeric value with the median of that parameter.



We have used all features for this assignment to predict the income. We tried removing some features and predicted the model, but it was not helpful to increase the accuracy.



**Implementation**

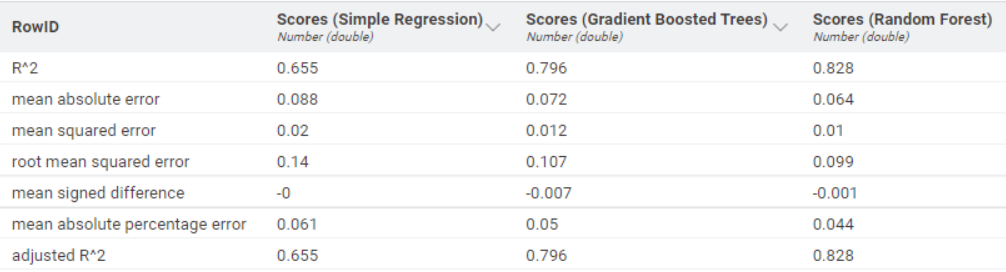
We have implemented the model using three classifiers. Here we have used:

* Simple Regression Tree
* Gradient Boosted Trees
* Random Forest (Regression)

The data was first cleaned by clearing out missing values using the Missing Value node and then normalized to the range of 1-2 using the Normalizer node. After which, the Category to Number node was used to convert categorical string data in the **ocean\_proximity** column to numerical data. The data was then split into training and testing set by the 70-30 rule for model building by stratified sampling based on **median\_house\_value** and a random seed value of 5 (best of multiple random trial values). We have used learner and predictor nodes according to the algorithm chosen. Predictor node helps to predict the **median\_house\_value** based on the remaining parameters. After training a model we have evaluated the Mean Squared Error and R2 values. There are other criteria we can choose to compare the regressors, like Mean Absolute Error, Root Mean Squared Error, Mean Signed Difference, Mean Absolute Percentage Error and Adjusted R2 value.

**Comparison**

We got the better Mean Squared Error and R2 values for the Random Forest Regression model when compared to other two Regression models. We have also incorporated 10-fold cross-validation technique for all the model and observed that it has led to minimally noticeable difference in errors. We have got 0.01 Mean Squared Error and 0.828 R2 for the Random Forest model, while for other models it was 0.02 MSE & 0.655 R2 (Simple Regression) and 0.012 MSE & 0.796 R2 (Gradient Boosted Trees Regression) respectively.



**Parameter Optimization and Saving the model**

It is an important step in machine learning task as it can enhance the performance of a model. We have used Parameter optimization Loop to get a structured approach to explore various combinations of parameters. This will help to identify which optimal setting can improve the accuracy of a model. We have used Parameter optimization loop for the X-Partitioner node to determine a better value for Random Seed and to determine what sampling works best for our purpose.