**Group\_9\_Task 2**:

Prediction Algorithm, implement and predict using

1. Linear Regression
2. Random Forest

**Approach:**

**Dataset:** After obtaining the corrected and cleaned data from Task1 by removing the missing and out of range values we have written the output to an output file. Which is taken as the input to the Task2, and we implemented Linear Regression and Random Forest Regression for the corrected data.

We then selected few columns based on our input dataset for prediction, we didn’t have any direct relation between any of the columns for accurate predictions, we have tried to identify the closest columns which would give good prediction results with a correlation. We have used the predicted variable as Edu column.

**Brief description of the dataset and variable are predicting:**

We have found the null values in Income column and filled the null values using KNN imputer which is task1. The data set which is obtained from the task1 is used in the task2 for prediction using linear Regression and Random forest method. In the new data set, there will be three more columns are added which are Mar\_st, Edu and Dt. These columns are converted from String data type to Integer data type and appended them to our initial data set.

We have chosen the Edu column for the prediction. Since there are many columns in the data set choosing all the columns as input to predicted model would be complex. So we have chosen the input columns as Kidhome, Teenhome, Income, Mar\_st, MntWines, MntMeatProducts.Then trained our data using the Edu column with linear regression. Once the train data is ready, testing data also prepared with linear regression.

Both training and testing data are split into 80 percent and 20 percent for linear regression and 70 percent and 30 percent date for Random Forest. With these data sets we have predicted the Edu variable and outputted them for display.

The columns that we have used as input and output for predications are:

**Input Columns:** Kidhome, Teenhome, Income, Mar\_st, MntWines, MntMeatProducts

**Output Column:** Edu

* First we have imported all the required libraries. We have used the columns that have been all converted to a double type and cleaned from the task1 as input for task2.
* **Reading data:** We utilized data frame reader to read the data from the data source to load the path of the data specified for the specific question and to load the data frame in the same format which the tabular format.
* We have used VectorAssembler to define out input and output columns for predictions we have combined all the specified input column into a single feature, then fit() and transform() the model defined. We then split the dataset into train and test using randomSplit() function. ( 0.8 for training and 0.2 for testing for linear regression and 0.7 for training and 0.3 for testing for random forest regression) i.e., we have split the data set such that 80% training and 20% testing data for linear regression and 70% training and 30% testing data for random forest.
* Repeated the same process for both regressions and then applied the LinearRegression() and RandonForestRegression() for the input and output columns specified. Finally, we have obtained and printed out the predictions that were obtained from regression functions.

**Instructions to compile and run the program:**

**Imports Used:**

from pyspark.sql import SparkSession

from pyspark.sql import SQLContext

from pyspark.ml.linalg import Vectors

from pyspark.sql.functions import col

from pyspark.sql import functions as F

from pyspark.ml.regression import LinearRegression, RandomForestRegressionModel, RandomForestRegressor #importing the regressions and models required

import pandas as pd

from pyspark.ml.feature import VectorAssembler

from pyspark.sql.functions import count, mean, when, lit, create\_map, regexp\_extract #importing few necessary sql functions

from pyspark import SparkContext, SparkConf

**Input:**As mentioned, we have used the output of Task 1 as the input for Task 3. task\_1\_output.csv is the output file for Task1 which we use as input file using following line.

**Output:** The top 20 predictions are printed on the console. We have tried to write the output to csv file but could not save the vectors with is the feature input column into csv format, have also tried using an array but couldn’t get proper results.

1. Open the cluster and login into it.

2. Upload the Group\_9\_Task\_2.py file into session storage(local or csh)

3. Command: spark-submit Group\_9\_Task\_2.py

**Discussion of Results:**

By seeing the results, we could say that using the columns that have relation to the response variable that ‘Edu’ we could get accurate results. We have tried several combinations of input columns by combining them as features to check for the accurate prediction values and finally decided on the combination that gave us better prediction values for the actual values in case of both linear regression and random forest regression.

**References:** [Machine Learning Basics: Random Forest Regression | by Gurucharan M K | Towards Data Science](https://towardsdatascience.com/machine-learning-basics-random-forest-regression-be3e1e3bb91a)