Project 3

Using MLlib with SparkSQL for Purchase Transaction Analysis and Prediction

Synopsis

This study describes how Apache Spark, in particular SparkSQL, was used for the processing and analysis of purchase transaction data, along with Spark's MLlib for predictive modelling. In order to understand and forecast purchasing behaviours, the project entails developing synthetic datasets representing customers and their buy transactions, which are then analysed and subjected to predictive modelling.

1. Overview Goal

This project's main goal is to use big data techniques to anticipate transaction volumes and understand customer buying behaviours. This entails using machine learning for predictive analysis and analysing massive datasets to get insightful information.

Tools Applied

Big data management made simple and effective with Apache Spark, an open-source distributed computing technology.

SparkSQL is an Apache Spark module that uses DataFrame APIs and SQL to process structured data.

2. Overview of Data Generation

Two synthetic datasets are created using PySpark as part of the data generation process: the Customers Dataset and the Purchases Dataset, which pertain to customer and buy transaction data, respectively.

Features of the Customers Dataset:

ID: An integer sequence ranging from 1 to 50,000.

Name: A 10-to 20-character string that is produced at random.

Age: Any number between 18 and 100 at random.

Random integer between 1 and 500 is the country code.

Pay: A random range from 100,000 to 10,000,000.

Features: The dataset mimics a heterogeneous clientele with a range of ages, income brackets, and ethnicities (represented by country codes).

Acquisitions Features of the Dataset:

TransID: An integer sequence ranging from 1 to 5,000,000.

CustID: A customer&#39;s ID, with an average of 100 transactions per client.

TransTotal: The purchase amount is represented by a random float between 10 and 2000.

TransNumItems: A random integer representing the quantity of products purchased, ranging from 1 to 15.

TransDesc: A 20–50 character random text string that describes the purchase.

Features: This dataset contains comprehensive transaction records with the purchase price, the things purchased, and a brief description.

3. Preprocessing and Data Loading

Spark DataFrames are used to load the created datasets, enabling quick processing and analysis. In order to comprehend the features of the datasets, preliminary preprocessing procedures include verifying data integrity, detecting null values, and conducting exploratory data analysis.

4. SparkSQL Data Analysis

Task 2.1: Purchase Filtering

Goal: Determine which purchases—those with totals more than $600—to exclude.

Methods: used the filtering features of SparkSQL to separate transactions according to the given criterion.

Results: To facilitate additional analysis, a filtered view of the purchasing dataset was constructed by creating a subset of transactions, designated as T1.

Task 2.2:Grouping by Number of Items

Goal: Examine expenditure trends in relation to the quantity of goods bought.

Method: The T1 dataset was grouped based on the number of items, and the median, lowest, and highest spending amounts were determined for each category.

Findings: By emphasising patterns and abnormalities, the study made it evident how spending behaviour is correlated with the quantity of products.

Task 2.3: Examining the Purchases Made by Young Clients

Goal: Specifically target clients between the ages of 18 and 25, compiling information about their purchases.

Method: Combined data based on customer ID and joined the T1 dataset with the young customer subset.

Summary: this assignment illuminated the purchase behaviours of younger consumers by offering a window into their preferences and spending patterns.

Task 2.4: Analysis of Customer Pairs

Goal: Find customer pairings in which the younger member spends more money but purchases fewer goods than the other.

Method: Filtering and selection methods were used to extract these pairs from the T3 dataset.

Summary: By concentrating on value rather than amount of purchases, the study of customer pairs provided insightful information about spending

Task 2.5) Data Preparation 1: The Dataset composed of customer ID, TransID, Age, Salary, TransNumItems and TransTotal is generated by joining purchases with customers on ID and selecting the relevant attributes on the joined table. This is then saved as a csv file on Jupyter.

Task 2.6) Data Preparation 2: The generated dataset is then split randomly using the randomSplit function with 0.8 and 0.2 as the parameters describing the fraction of the Dataset selected in both the train and test set. The seed is set to 22 for consistency to make sure that the same train and test sets are generated each time the code is run.

Task 2.7) Price prediction: pyspark.ml is the module used to access 3 machine learning models to predict TransTotal. The output is a continuous variable making this a regression problem. The 3 algorithms used are:

1. Linear Regression
2. Decision Tree Regressor
3. Random Forest Regressor

The feature columns are assembled using VectorAssembler to convert them to an acceptable input (single vector column) form for the models. Pipelines are then created with the input and models which are trained using the trainset. Predictions for TransTotal are made using the testset.

Task 2.8) Price prediction: The above models are evaluated on the basis of the following 3 metrics:

1. MAE (Mean absolute error)
2. R^2
3. RMSE (Root mean squared error)

RMSE and MSE are measures of error, with RMSE being the preferred choice when large errors are undesirable since RMSE gives a relatively higher weight to large errors since it squares the errors before they are averaged.

A statistical metric called R-squared is used to assess how well a regression model fits data. R-square values range from 0 to 1. When there is no difference between the predicted and actual values and the model fits the data exactly, we have an R-square of 1. R-square, on the other hand, equals 0 when the model learns no relationship between the independent and dependent variables and does not predict any variability in the model.

$$\text{RMSE} = \sqrt{\frac{\sum (y_i - y_p)^2}{n}}$$

$$\text{MAE} = \frac{|(y_i - y_p)|}{n}$$

y_i = actual value
y_p= predicted value
n = number of observations/rows

(https://miro.medium.com/max/552/1*5OQunI-NR-S0gAZFIit1Rw.png)

$$R^2 = 1 - \frac{\Sigma(y_i - \hat{y})^2}{\Sigma(y_i - \bar{y})^2}$$

Where,
$\hat{y}$ $- \textit{predicted value of } y$
$\bar{y} - \textit{mean value of } y$

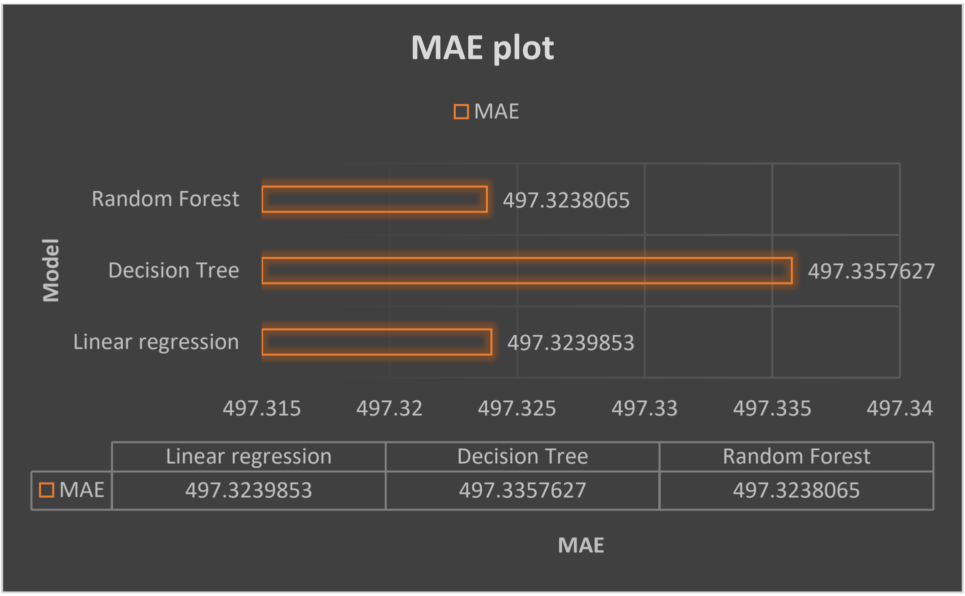
(https://4.bp.blogspot.com/-wG7IbjTfE6k/XGUvqm7TCVI/AAAAAAAAAZU/vpH1kuKTIooKTcVlnm1EVRCXLVZM9cPNgCLcBGAs/s1600/formula-MAE-MSE-RMSE-RSquared.JPG)

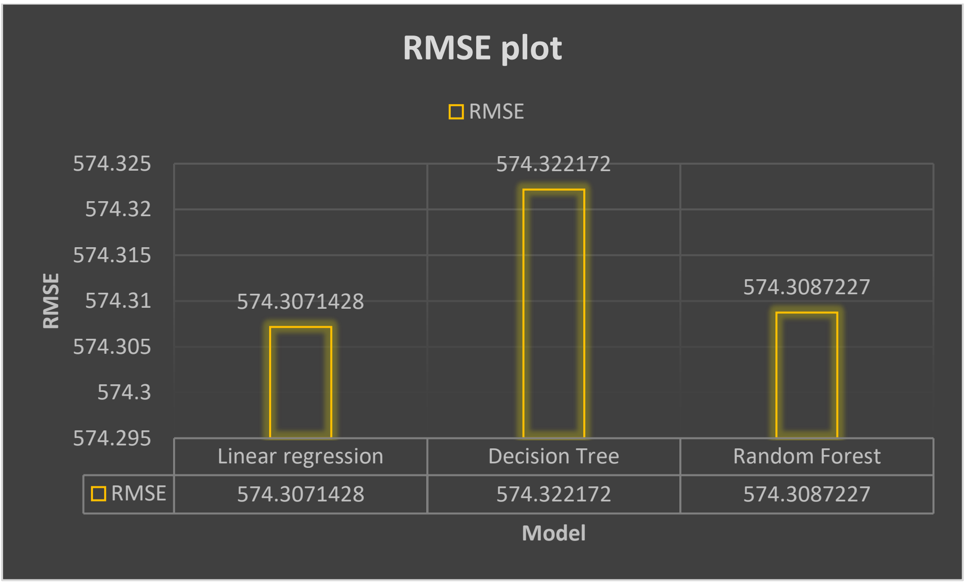
Results:

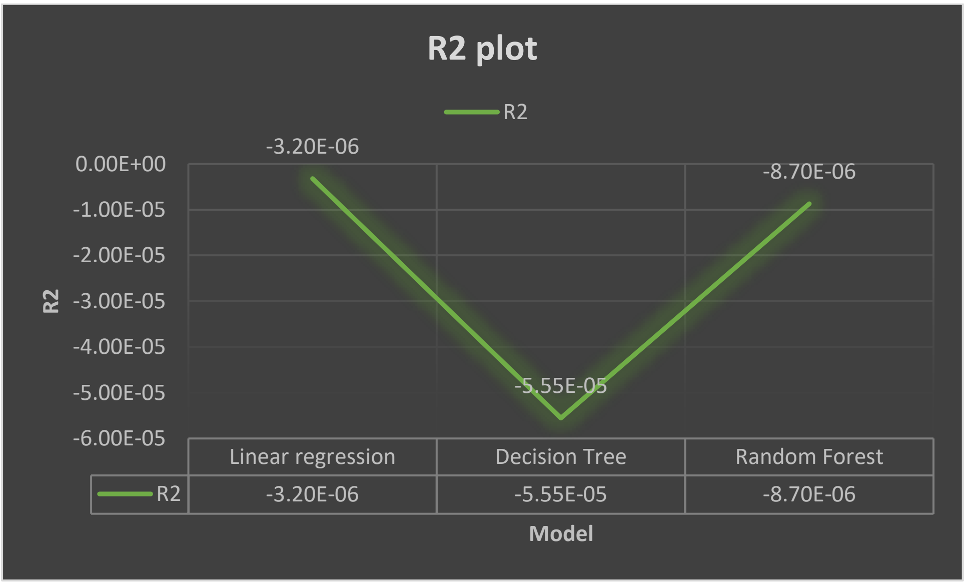
The results are illustrated in the table below.

Model	MAE	RMSE	R^2
Linear regression	497.32398530185117	574.3071428005508	-3.1955897634716735e-06
Decision Tree	497.33576271191635	574.3221719649848	-5.553486257525719e-05
Random Forest	497.3238065146036	574.3087226658814	-8.697428186099643e-06

The same are plotted below for better visualization


MAE plot
MAE
Model
Random Forest
Decision Tree
Linear regression
497.3238065
497.3357627
497.3239853
497.315
497.32
497.325
497.33
497.335
497.34
Linear regression
Decision Tree
Random Forest
MAE
497.3239853
497.3357627
497.3238065
MAE


RMSE plot
RMSE
RMSE
574.325
574.32
574.315
574.31
574.305
574.3
574.295
574.3071428
574.322172
574.3087227
Linear regression
Decision Tree
Random Forest
RMSE
574.3071428
574.322172
574.3087227
Model


R2 plot
R2
R2
0.00E+00
-1.00E-05
-2.00E-05
-3.00E-05
-4.00E-05
-5.00E-05
-6.00E-05
-3.20E-06
-5.55E-05
-8.70E-06
Linear regression
Decision Tree
Random Forest
R2
-3.20E-06
-5.55E-05
-8.70E-06
Model

Conclusion:

The R^2 value is lowest for Decision trees followed by random forest followed by linear regression.

The MAE is lowest for random forest followed by linear regression followed by decision trees.

RMSE is lowest for Linear regression, followed by random forest followed by decision tree.

Hence, we can conclude that Linear regression is the best model for our dataset followed by random forest followed by decision trees. However, the differences in value are miniscule.

From the R^2 values (close to zero) we can conclude that the 3 models perform poorly. This can be attributed to the fact that the dataset was generated randomly which probably led to no discernible patterns in the dataset from the model to learn from. A better result can be expected on real world datasets having meaningful correlations between the features and the output.

<u>Distribution of tasks:</u> Task 2.1- task 2.4 – Shubham Wagh, Task 2.4- task 2.8- Isha Jain

<u>Use of Generative AI –</u> ChatGPT was leveraged to help with the code and understand bugs encountered in task 2.