# [https://avatars2.githubusercontent.com/u/4156894?v=3&s=100](http://www.calstatela.edu/centers/hipic) CIS5560 Term Project Tutorial

**Authors**: Maria Perez

**Instructor**: [Jongwook Woo](https://www.linkedin.com/in/jongwook-woo-7081a85)

**Date**: 4/1/2025

Lab Tutorial

Maria Perez (mpere110@calstatela.edu)

**Flight Prices Prediction Model using SparkML**

## Objectives

The aim of this tutorial is to build a model that predicts the prices of flights leaving from LAX based on the features of trip distance, flight duration, date of flight and destination airport using the following machine learning algorithms:

* Gradient Boost Tree Regression
* Random Forest Regression

## Platform Specifications

* Databricks Community Edition
* # of CPU cores: 8
* 9.1 LTS (includes Apache Spark 3.1.2, Scala 2.12)
* Hadoop Version – 3.3.3
* Pyspark Version: 3.2.1

## Dataset Specifications

* Dataset Name: Flight Prices
* Dataset size: 2.95GB
* Dataset Format.csv
* Dataset URL: <https://www.kaggle.com/datasets/dilwong/flightprices>

## Tasks to build/test the Machine Learning Models

### TASK 1: Get data manually from data source

1. Download the Flight Prices dataset from Kaggle: <https://www.kaggle.com/datasets/dilwong/flightprices>

### TASK 3: Databricks Community Edition

#### Step 1: Upload dataset to Databricks Community Edition

1. Login or create an account in Community Databricks: <https://community.cloud.databricks.com/>.
2. Once signed into Databricks Community Edition, click **Compute** on the left side, then **Create** **Compute**.
   1. Name the Cluster Name as CIS5560.
   2. Choose Databricks Runtime version as Runtime: 9.1 LTS (includes Apache Spark 3.1.2, Scala 2.12)

A screenshot of a computer

AI-generated content may be incorrect.

1. Now to upload the dataset, click on Catalog.

A screenshot of a computer

AI-generated content may be incorrect.

* 1. Click on Create Table.
  2. Drop the file to upload or click on the gray box under Files to upload the dataset.
  3. Click Create Table in Notebook.

A screenshot of a computer

AI-generated content may be incorrect.

* 1. If notebook does not automatically open, click on Workspace and open the notebook. Rename it to Flights-RFR.

#### Step 2: Prepare data and create training/test data

1. Insert the following to apply the necessary packages.

A screenshot of a computer program

AI-generated content may be incorrect.

1. Insert a new cell with code to be used to run Pyspark in CLI.

A screenshot of a computer

AI-generated content may be incorrect.

1. Insert a new cell with the code to be used to load dataset, create a DataFrame and display the DataFrame.

A screenshot of a computer

AI-generated content may be incorrect.

1. Insert a new cell to display the schema of the DataFrame.

A screenshot of a computer program

AI-generated content may be incorrect.

1. Insert a new cell and add the following code to convert the contents of the TravelDuration to the total number of minutes.

A screenshot of a computer

AI-generated content may be incorrect.

1. Insert a new cell and add the following code to create new columns and handle null values. This will make the dates and duration usable for the model.

A screenshot of a computer program

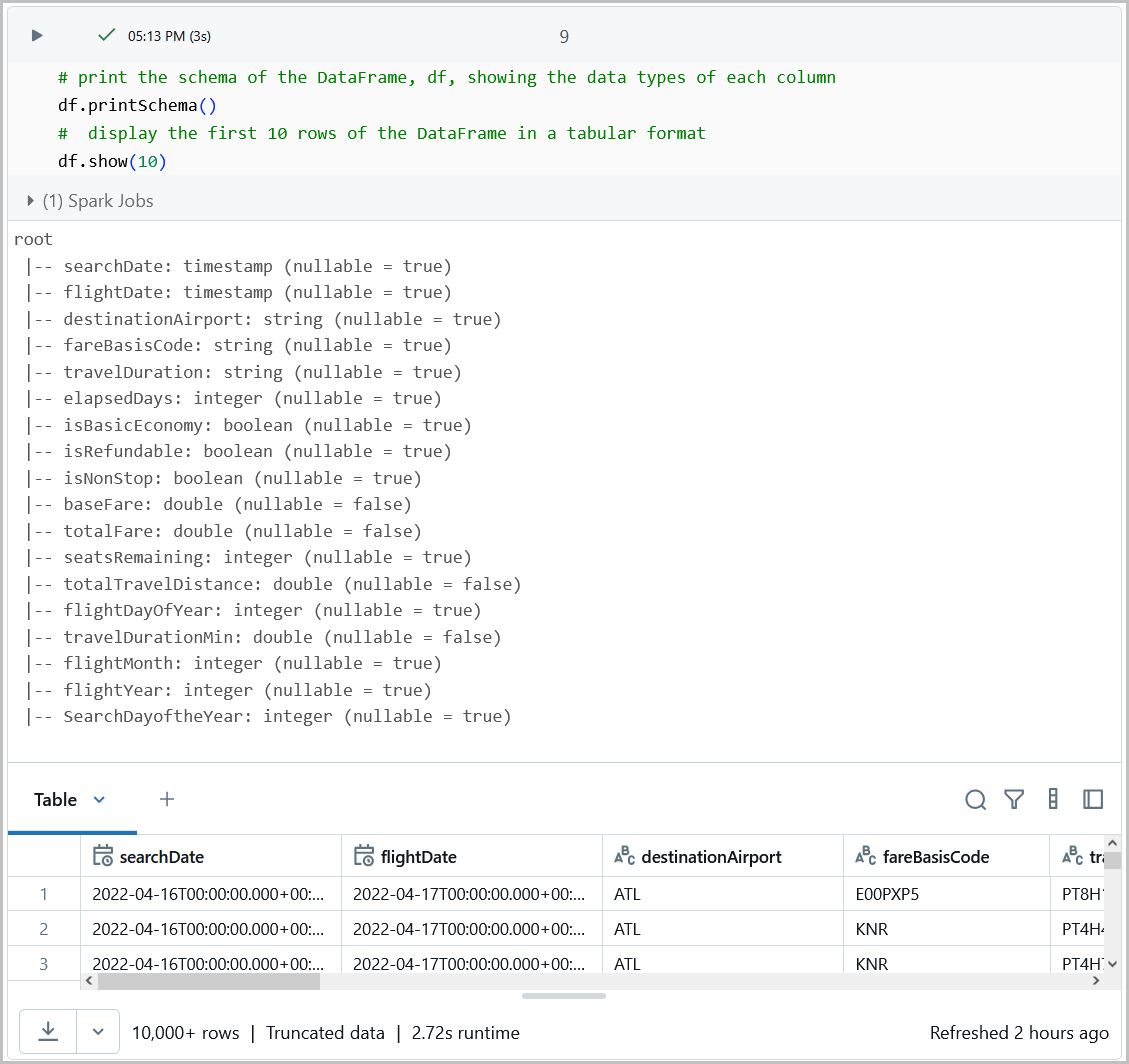
AI-generated content may be incorrect.

1. Insert a new cell and add the following code to drop unused columns.

A screenshot of a computer code

AI-generated content may be incorrect.

1. Insert a new cell and add the following code print the schema and tabular format of the DataFrame



1. Insert a new cell and add the following code to create a StringIndexer to convert the categorical feature into numerical indices. Then display the schema of the DataFrame.

A screenshot of a computer program

AI-generated content may be incorrect.

1. Insert a new cell and add the following code to identify the columns that are to be used as features.

A screenshot of a computer

AI-generated content may be incorrect.

1. Insert a new cell and add the following code that split the data into train and test in the ration of 70:30. Training set is used to build a model and testing set is used to test the model. Displays the number of rows used for training and the number of rows used for testing.

A screenshot of a computer program

AI-generated content may be incorrect.

#### Step 3: Feature Importance

Feature Importance refers to calculating the score for all the input features for a given model. This score indicates the importance of each feature. The higher the score, the larger the impact on the model. Feature Importance was performed using Random Forest Regression Model.

Insert the following to check for feature Importance.

A screenshot of a computer

AI-generated content may be incorrect.

#### Step 4: Create model: Random Forest Regression

In this task, run the Random forest regression algorithm using train split validation and cross validation.

1. Insert a new cell to create a VectorAssembler to combine multiple column into a single feature vector.

A screenshot of a computer

AI-generated content may be incorrect.

1. Insert a new cell to create a RandomForestRegressor Model. Set ‘totalFare’ as the target variable.

A screen shot of a computer

AI-generated content may be incorrect.

1. Insert a new cell to create a parameter grid. This is a collection of hyperparameters that can be tuned to improve the model’s performance.

A screenshot of a computer

AI-generated content may be incorrect.

1. Insert a new cell to create a RegressionEvaluator to asses the model’s performance.

A screenshot of a computer

AI-generated content may be incorrect.

1. Insert a new cell to define a Pipeline that will convert the categorical feature into a numerical representation.

A screenshot of a computer program

AI-generated content may be incorrect.

1. Insert a new cell to check how long the Cross Validator will take.

A white rectangular frame with black text

AI-generated content may be incorrect.

1. Insert a new cell with the CrossValidator.

A screenshot of a computer

AI-generated content may be incorrect.

1. Insert a new cell to fit the CrossValidator to the training data.

A screenshot of a computer

AI-generated content may be incorrect.

1. Insert a new cell to stop the time tracking the duration of the CrossValidator.

A screenshot of a computer

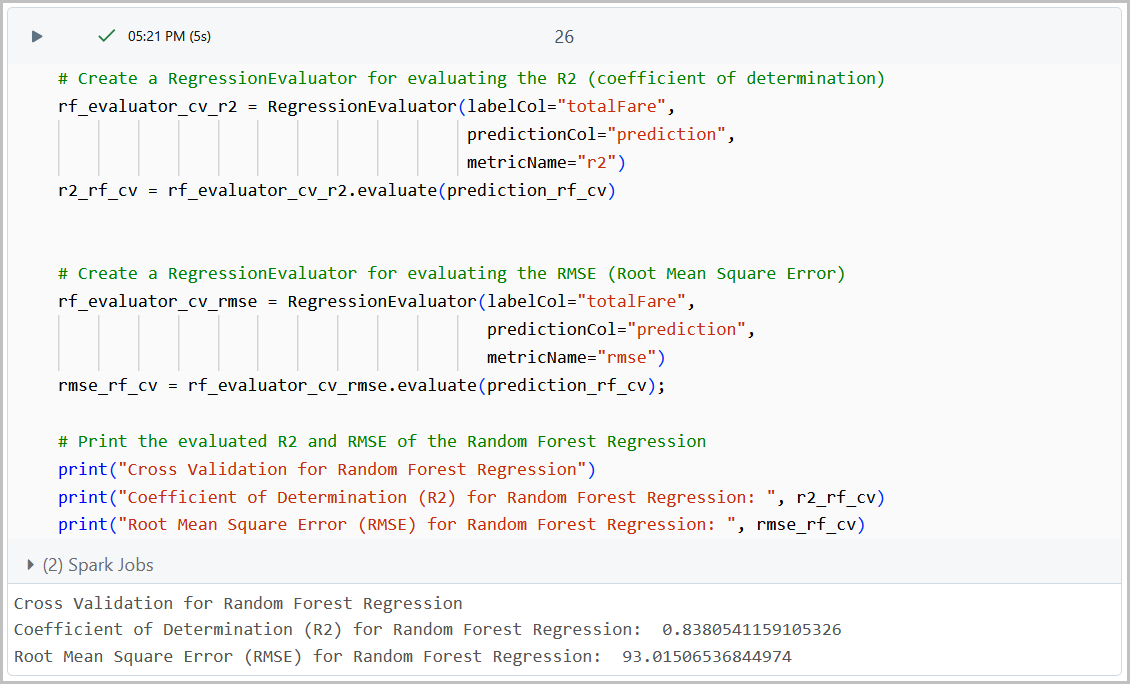
AI-generated content may be incorrect.

1. Insert a new cell to use the best model found by the cross validator to make predictions on test data. Display few rows of the predicted results.

A screenshot of a computer

AI-generated content may be incorrect.

1. Insert a new cell to print the RMSE and R2 results.



1. Next, train and validate using TrainValidationSplit. Insert a new cell to check how long the TrainValidationSplit will take.

A screen shot of a computer

AI-generated content may be incorrect.

1. Insert a new cell to create a TrainValidationSplit.

A screenshot of a computer program

AI-generated content may be incorrect.

1. Insert a new cell to train the model using TrainValidationSplit on the training data.

A screenshot of a computer

AI-generated content may be incorrect.

1. Insert a new cell to stop the time tracking time duration of the TrainValidationSplit.

A screenshot of a computer error

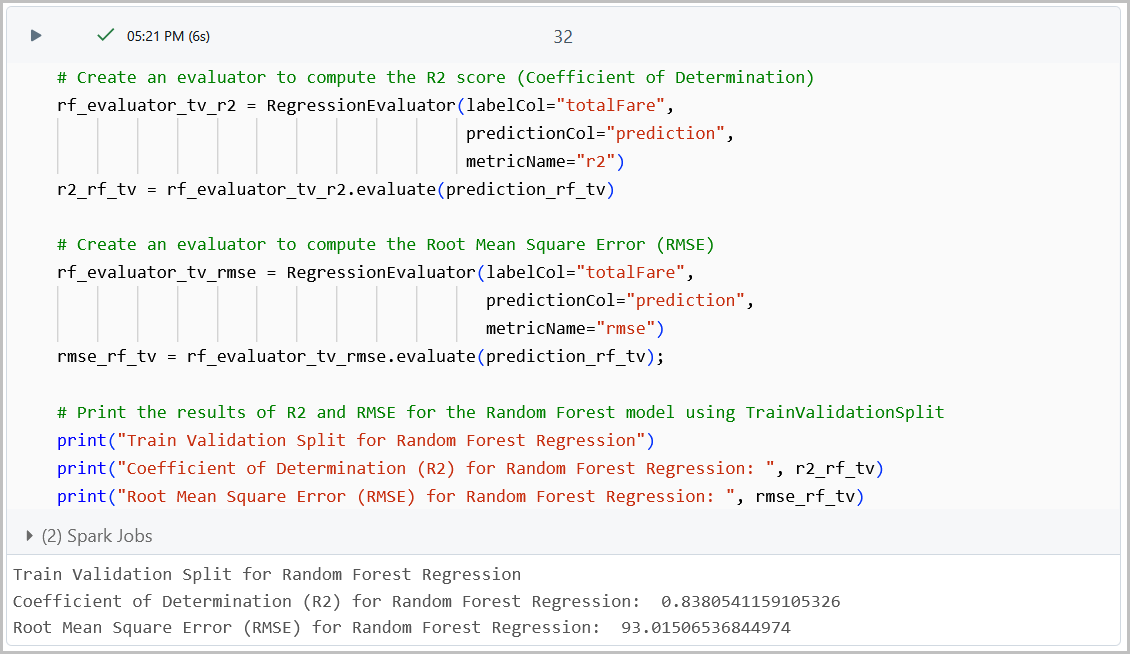
AI-generated content may be incorrect.

1. Insert a new cell to use the best model found by the TrainValidationSplit to make predictions on test data. Display few rows of the predicted results.

A screenshot of a computer code

AI-generated content may be incorrect.

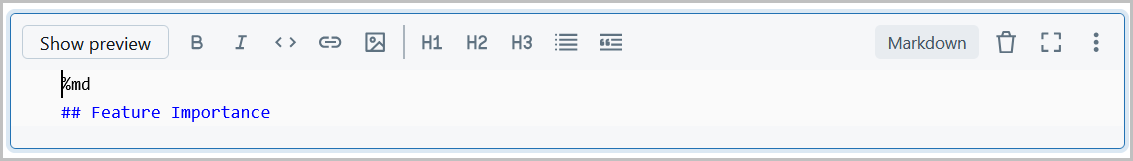
1. Print results for R2 and RMSE for the TrainValidationSplit.



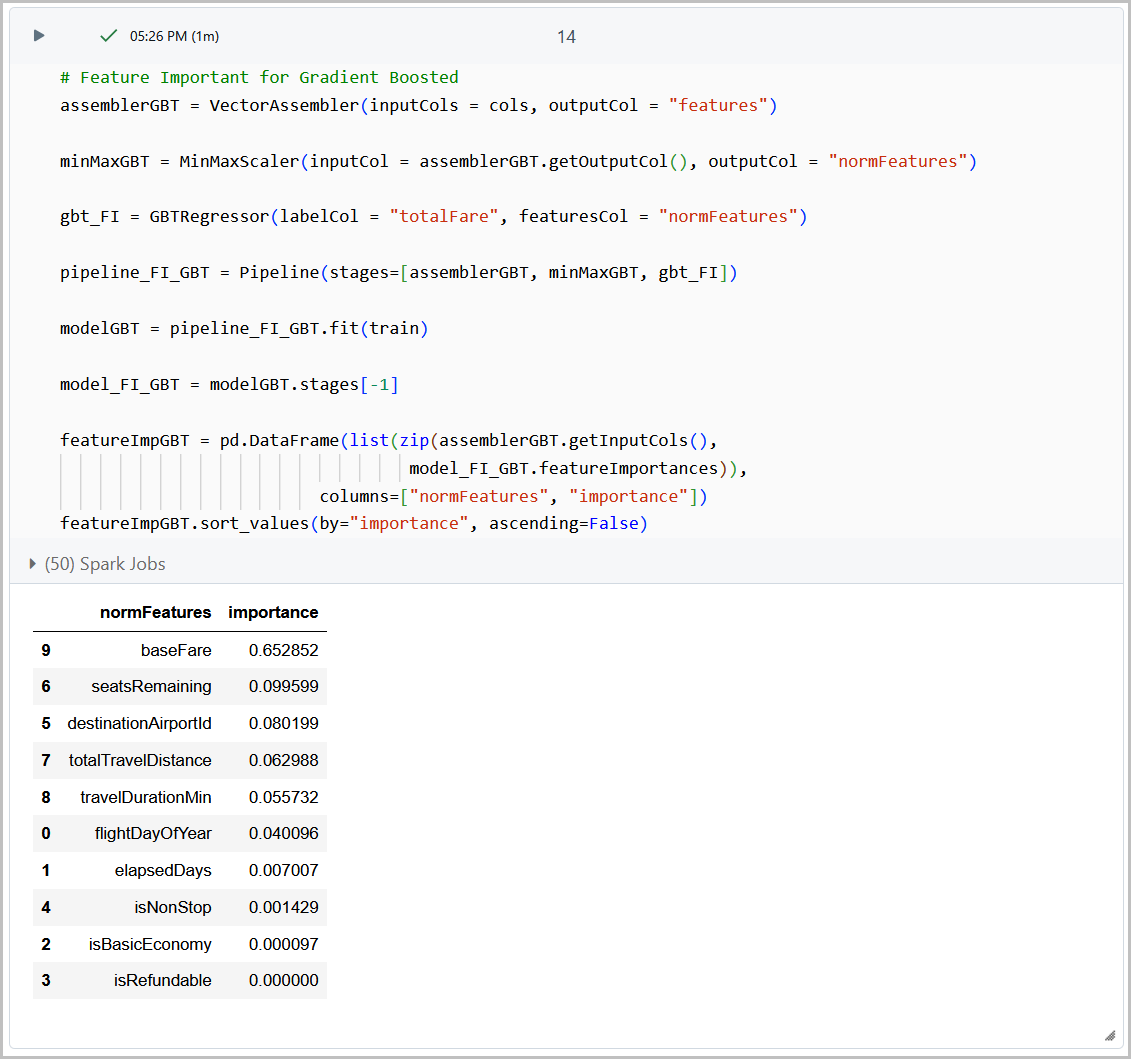
#### Step 5: Create model: Gradient Boost Regression

Run Gradient Boost Tree Algorithm using Train Split Validation and Cross Validation.

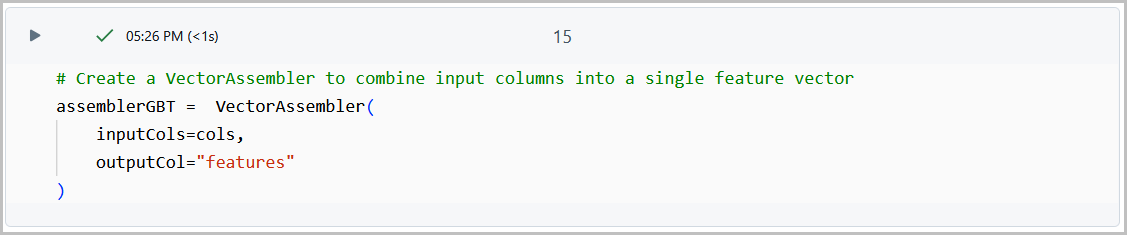
1. Create a new notebook by cloning Flights-RandomForest. Name the new file Flights-GradientBoost.
2. Delete all the cells after cell 12, which includes the training and testing rows of the data.
3. Insert a new cell with the heading for Feature Importance.



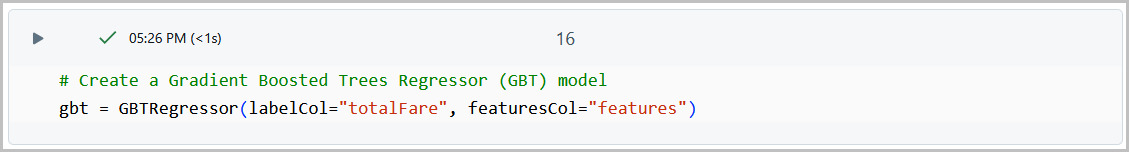
1. Insert a new cell to display the results of Feature Importance.



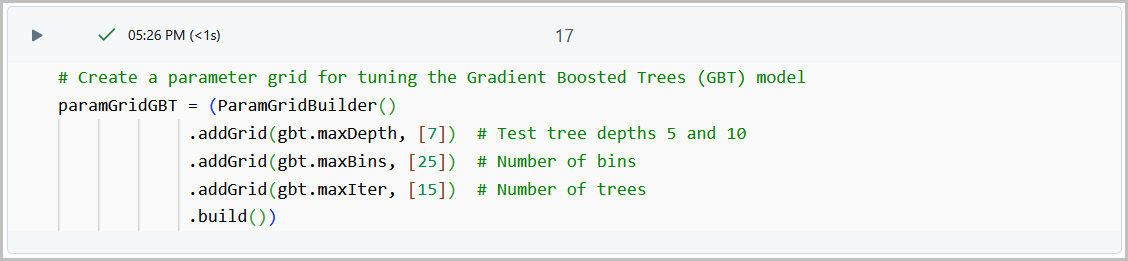
1. Insert a new cell to create a VectorAssembler to combine columns into a single feature vector.



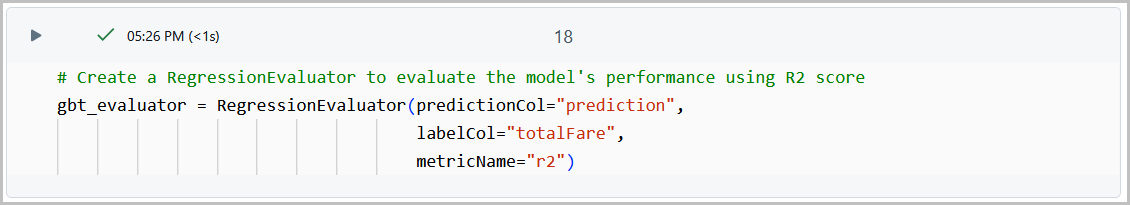
1. Insert a new cell to create a Gradient Boosted Trees Regressor.



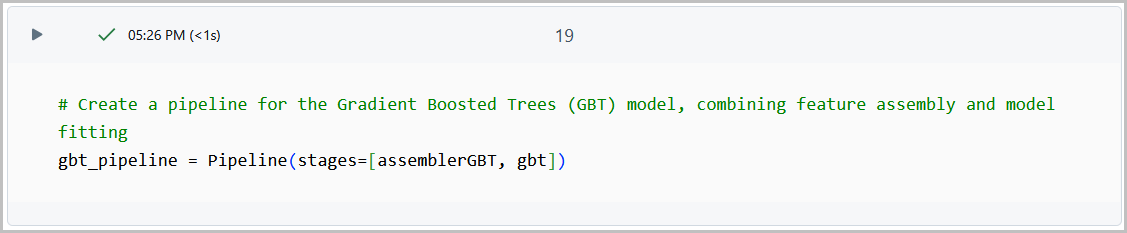
1. Insert a new cell to create a parameter grid to tune the GBT model.



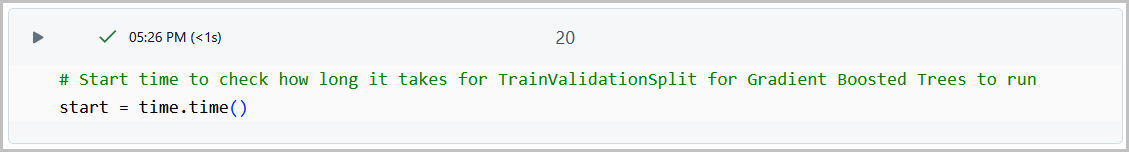
1. Insert a new cell to create a RegressionEvaluator to evaluate the model’s performance using the R2 score.



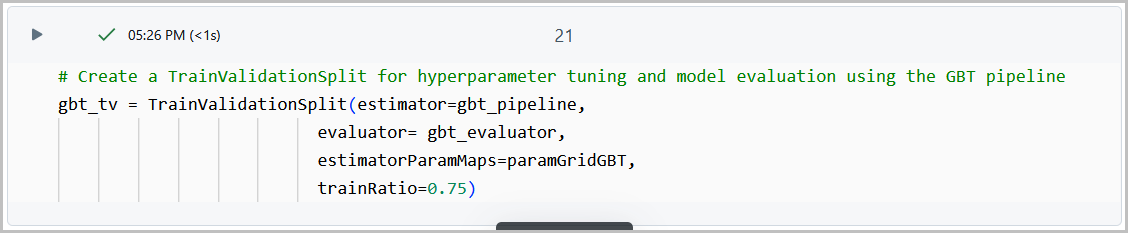
1. Insert a new cell to create a Pipeline for the GBT model.



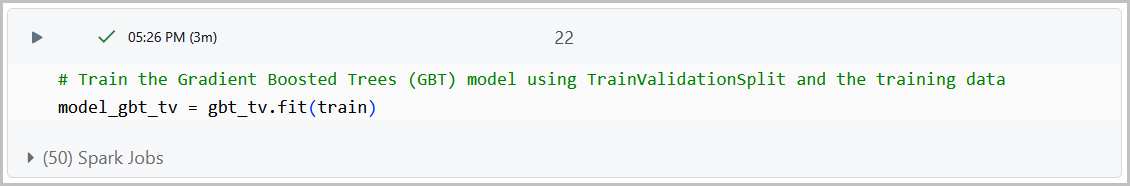
1. Insert a new cell to check how long the TrainValidationSplit will take.



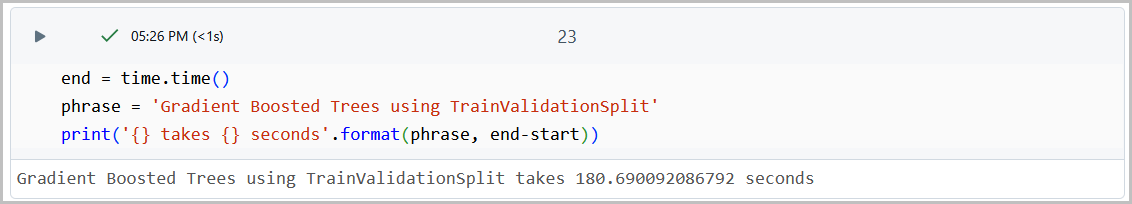
1. Insert a new cell to create a TrainValidationSplit for hyperparameter tuning and model evaluation using the GBT pipeline.



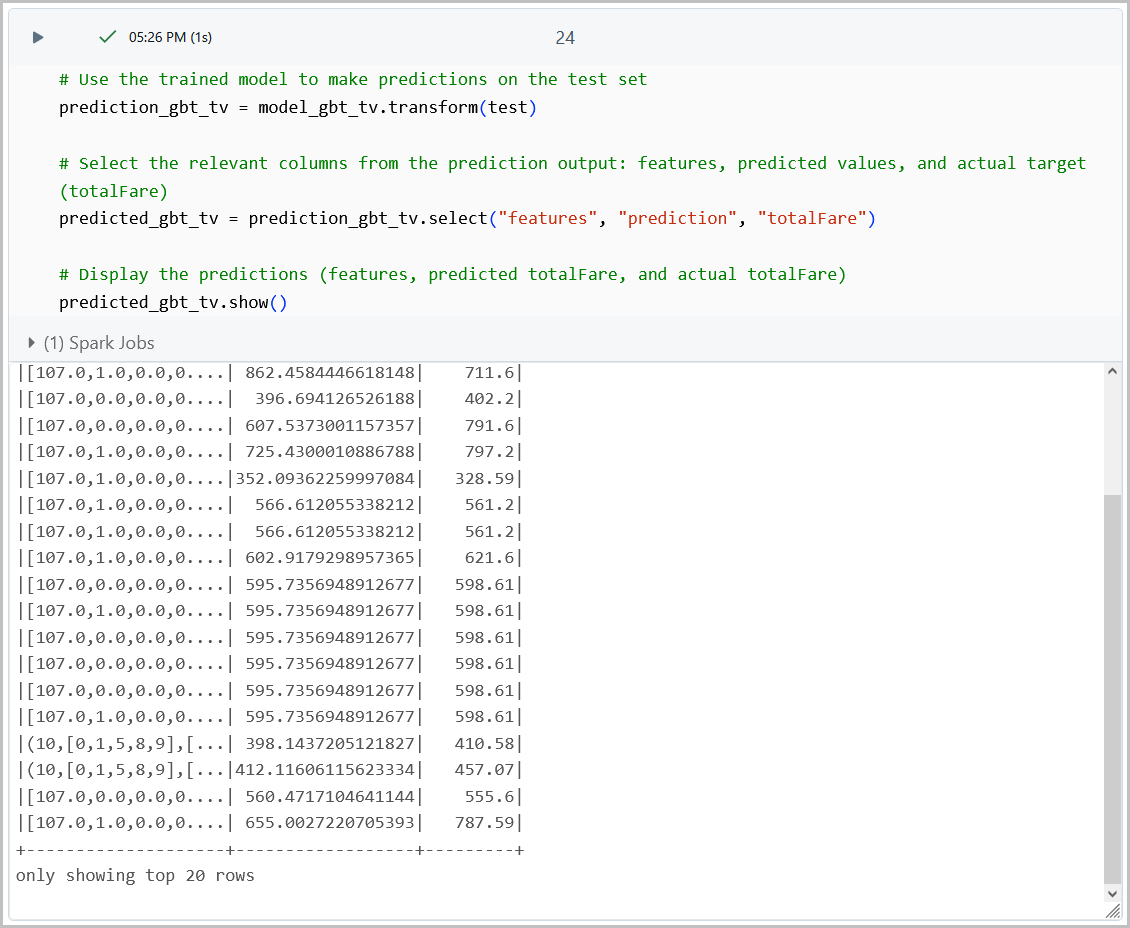
1. Insert a new cell to train GBT model using TrainValidationSplit



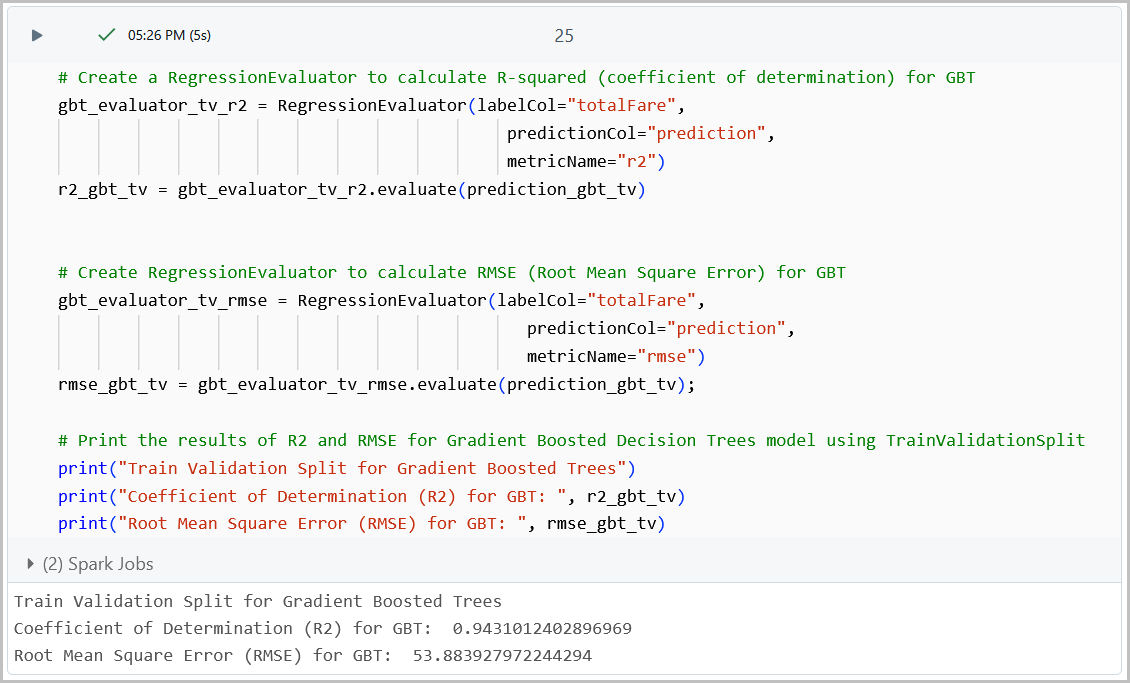
1. Insert a new cell to stop the time tracking the duration of the TrainValidationSplit for the GBT model.



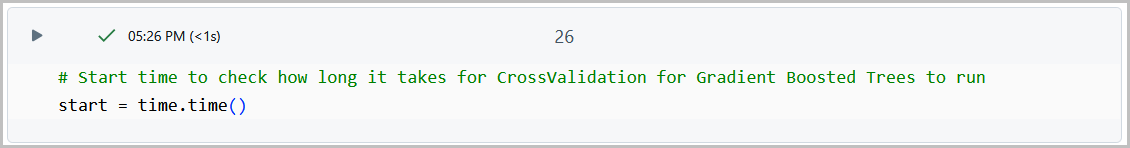
1. Insert a new cell to use the best model found by the TrainValidationSplit to make predictions on test data. Display few rows of the predicted results.



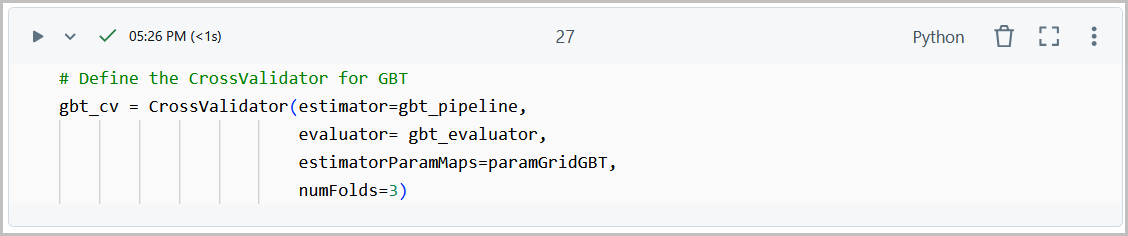
1. Insert a new cell to print the results for R2 and RMSE for the GBT model.



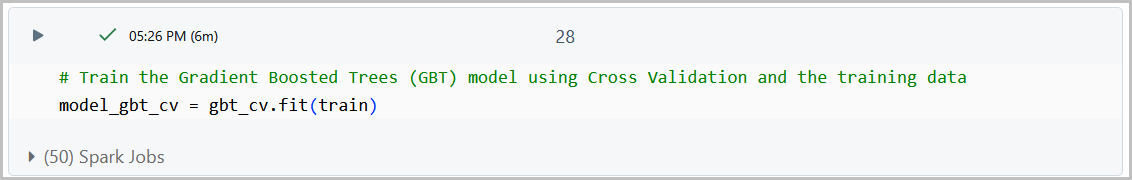
1. Insert a new cell to start the time to track the duration of the CrossValidation for the GBT model.



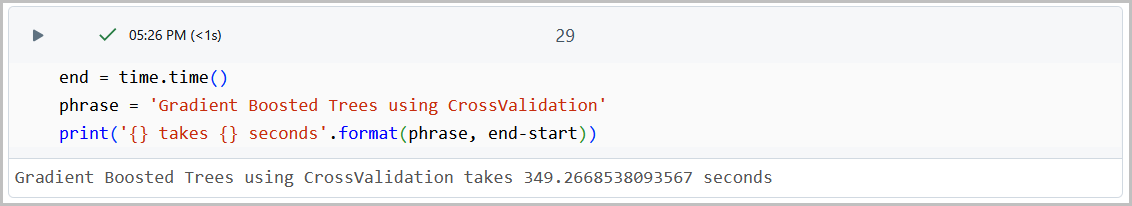
1. Insert a new cell to define the CrossValidator for GBT.



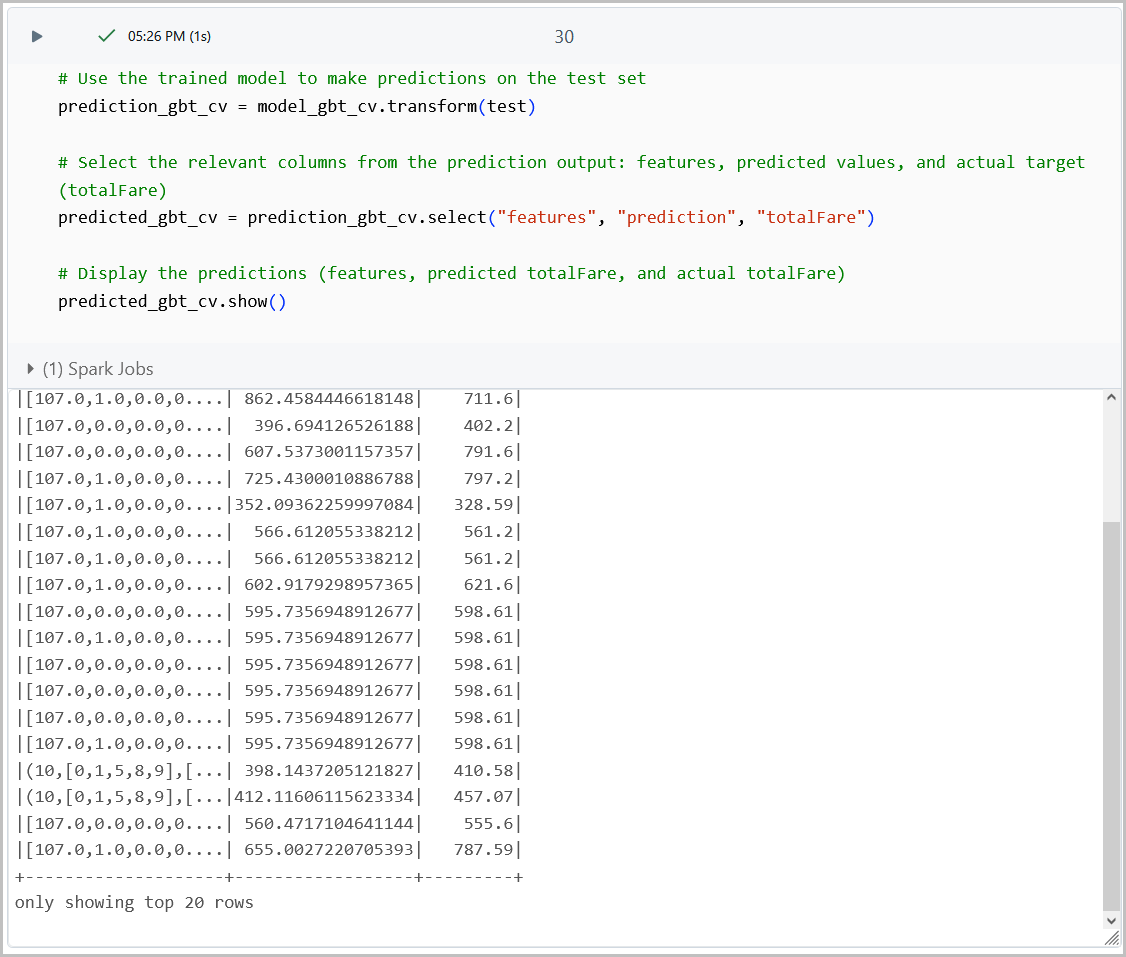
1. Insert a new cell to train the GBT model using CrossValidation.



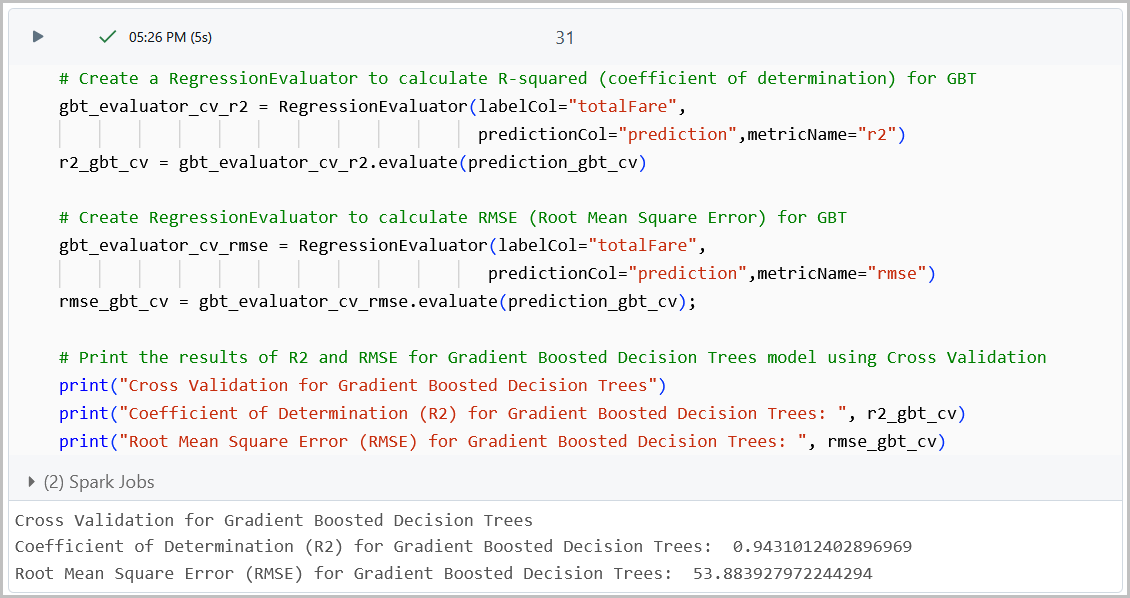
1. Insert a new cell to stop the time tracking the duration of the CrossValidator for the GBT.



1. Insert a new cell to show the predictions vs the actual values of cross validation.



1. Insert a new cell to print results of R2 and RMSE of the Cross Validation for the Gradient Boosted Decision Trees.

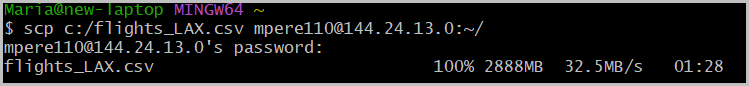


### TASK 3: Hadoop File System

#### STEP 1: Upload the dataset to the Hadoop File System

1. Open a shell terminal.
2. To upload the dataset file to the Hadoop File System (HDFS), first transfer the file to the local directory using scp commands. Replace mpere110 with your username and your HDFS server IP address.

scp C:/flights\_LAX.csv **mpere110**@**HDFS server IP Address**:~/



1. Open another shell terminal and enter the ssh command to connect to the Hadoop Spark cluster.

$ ssh mpere110@129.153.214.22

A black screen with white text

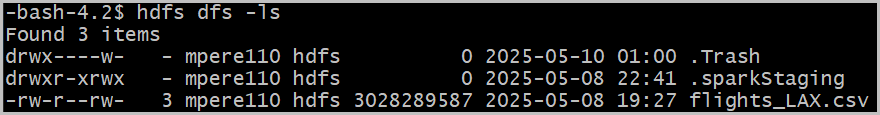
AI-generated content may be incorrect.

1. Upload the dataset file to HDFS
   1. Run the following shell command to put file in Final-Project directory.

hdfs dfs -put flights\_LAX.csv

* 1. Run the following to make sure if the csv file is uploaded to Final-Project directory. A list of all the files in HDFS will be displayed.

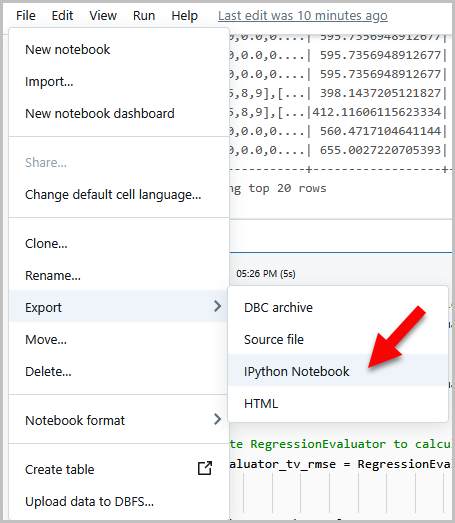
hdfs dfs -ls



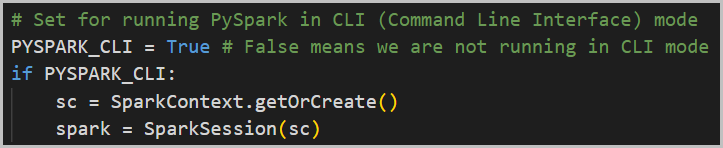
#### Step 2: Prepare .py for HDFS

Since Databricks has a limit to the size of the dataset, HDFS will be used to process the full size of the dataset.

1. Start by exporting both .py file from Databricks. Go to File > Export > IPython Notebook.

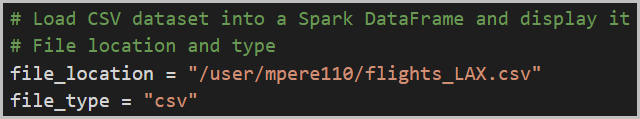


1. Open both files using a text editor and make the following changes.
   1. Change PySpark to True.



* 1. Change the file location to the one listed below.

ile\_location = "/user/mpere110/flights\_LAX.csv"

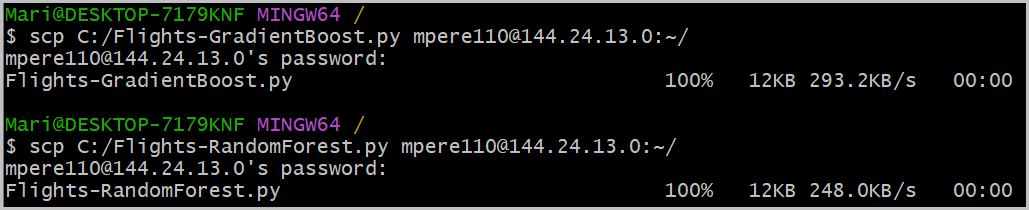


#### Step 3: Calculate R2 and RMSE using HDFS

1. Open a shell terminal and upload the Flights-GradientBoost.py and Flights-RandomForest.py

scp C:/Flights-GradientBoost.py [mpere110@129.153.214.22:~/](mailto:mpere110@129.153.214.22:~/)

scp C:/Flights-RandomForest.py [mpere110@129.153.214.22:~/](mailto:mpere110@129.153.214.22:~/)

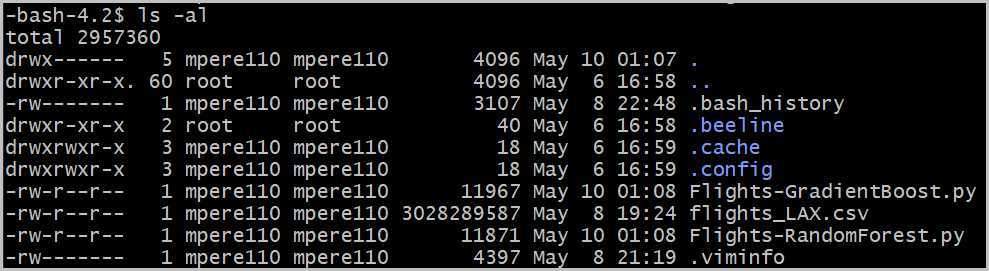


1. Open another Git Bash window, and log into using your username and password.

ssh mpere110@129.153.214.22

1. Using ls -al to check all files uploaded correctly.

ls -al

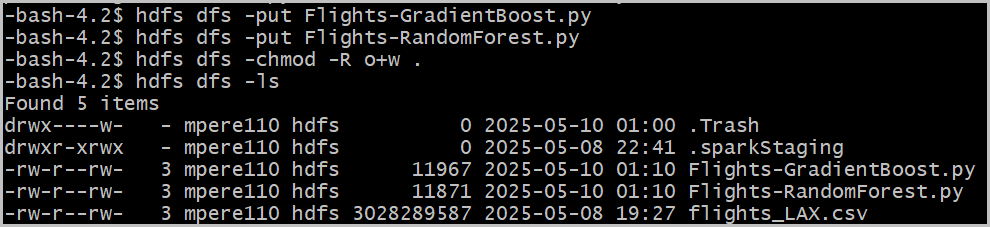


1. Run the following to put all the dataset to HDFS and give read/write permission.

hdfs dfs -put Flights-GradientBoost.py

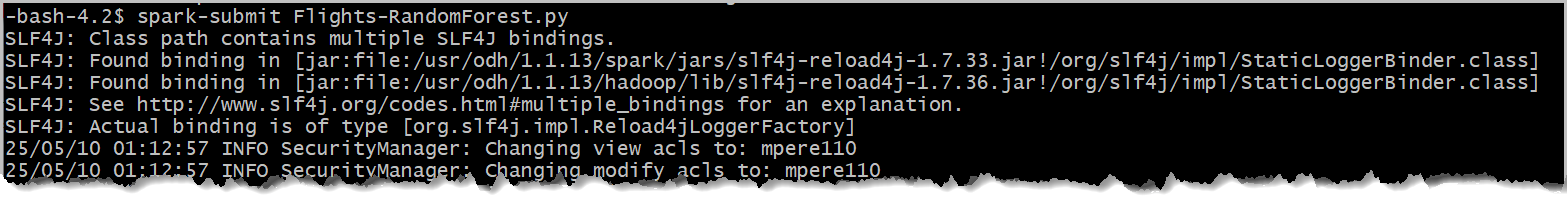
hdfs dfs -put Flights-RandomForest.py

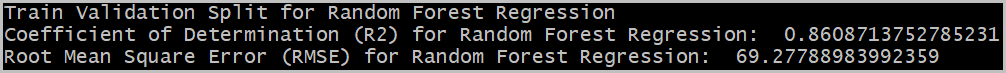
hdfs dfs -chmod -R o+w .



1. Let’s get the results for Random Forest Regression. Run the py file.

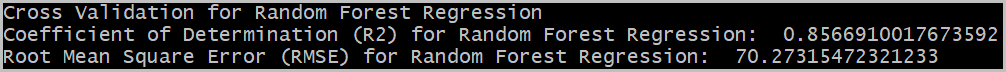
spark-submit Flights-RandomForest.py



1. Note the results of Feature Importance, Cross Validation, TrainValidation Split and the model train times.
   1. Train Validation Split for Random Forest and its training time.



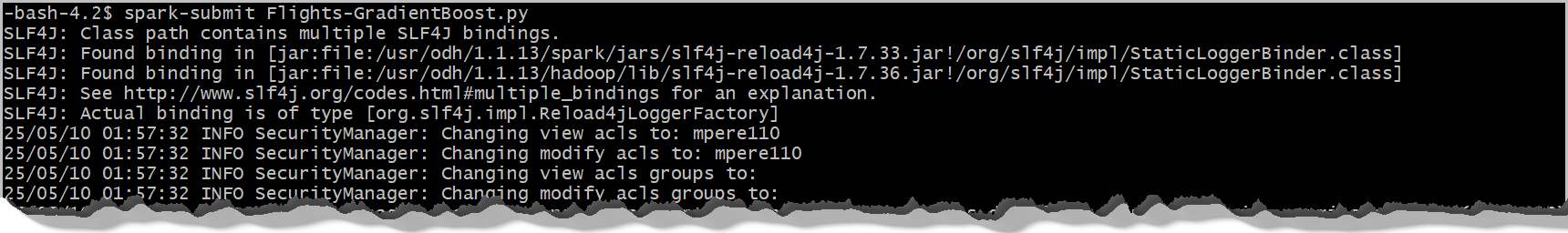
* 1. Cross Validation for Random Forest and its training time.



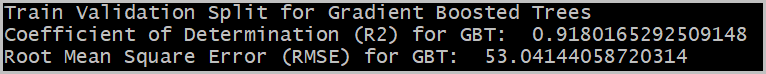


1. Next, get the results for Gradient Boosted Regression. Run the py file.

spark-submit Flights-GradientBoost.py

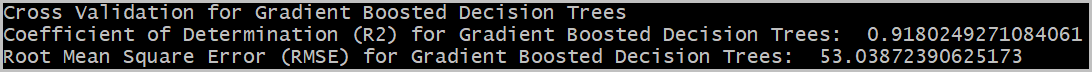


1. Note the results of Feature Importance, Cross Validation, TrainValidation Split and the model train times.
   1. Train Validation Split for Gradient Boosted and its training time.





* 1. Cross Validation for Gradient Boosted and its training time.





### TASK 4: Compare Results

Compare results of both regression algorithms.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Test | R2 | RMSE | Model Training Time(s) |
| Random Forest | Cross Validation | 0.85669 | 70.27315 | 209.4498 |
| Random Forest | TrainValidation Split | 0.86087 | 69.277889 | 99.8695 |
| Gradient Boost | Cross Validation | 0.918024 | 53.03872 | 275.0281 |
| Gradient Boost | TrainValidation Split | 0.918016 | 53.04144 | 158.61868 |

## References

* URL of Data Source: <https://www.kaggle.com/datasets/dilwong/flightprices>
* URL of your Github: <https://github.com/mperez13/flightPricesML.git>