**DS USING PYTHON LAB**

**EXPERIMENT: 9 B**

**AIM:**

a)Perform data Visualization using Power BI

b)Perform linear and logistic regression using Apache Spark

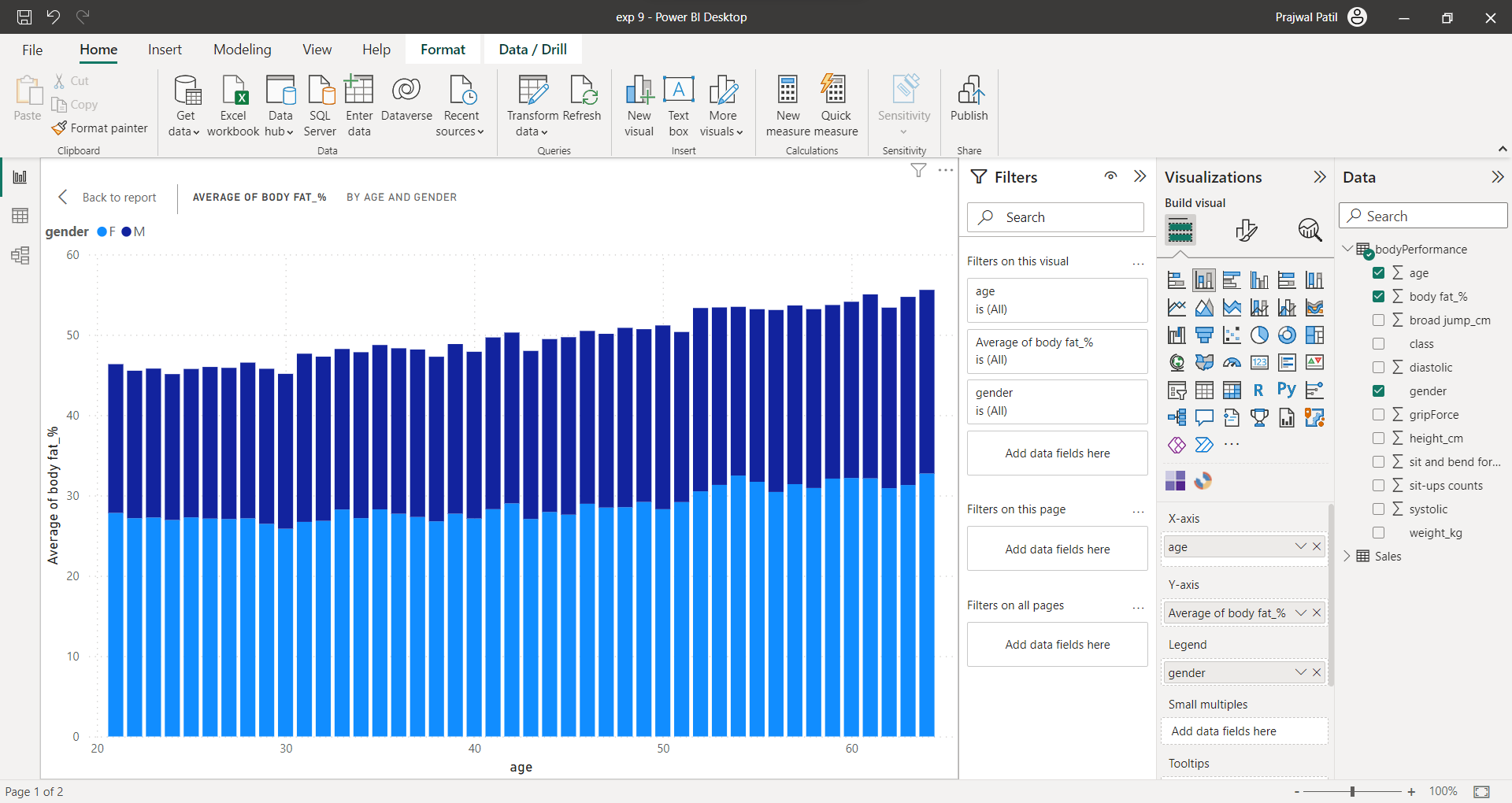
**IMPLEMENTATION:**

**Visualization using PowerBi:**

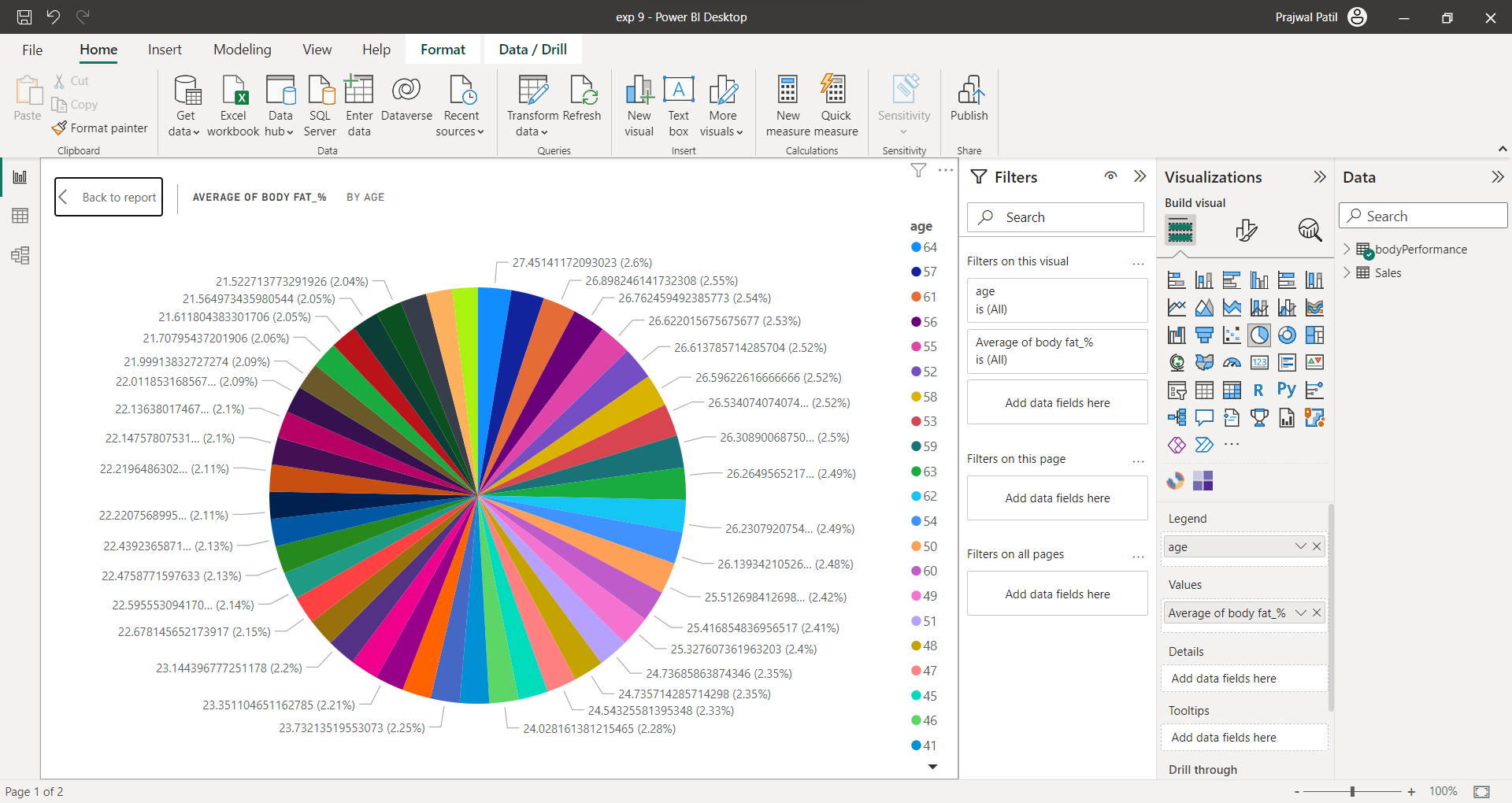
**Dataset:** <https://www.kaggle.com/datasets/kukuroo3/body-performance-data>

The above dataset contains various measures of Body Performance such as age, Gender, Height\_cm, weight\_kg, body fat\_%, diastolic, systolic, gripForce, sit and bend, forward\_cm, sit-ups counts, broad jump\_cm, class.

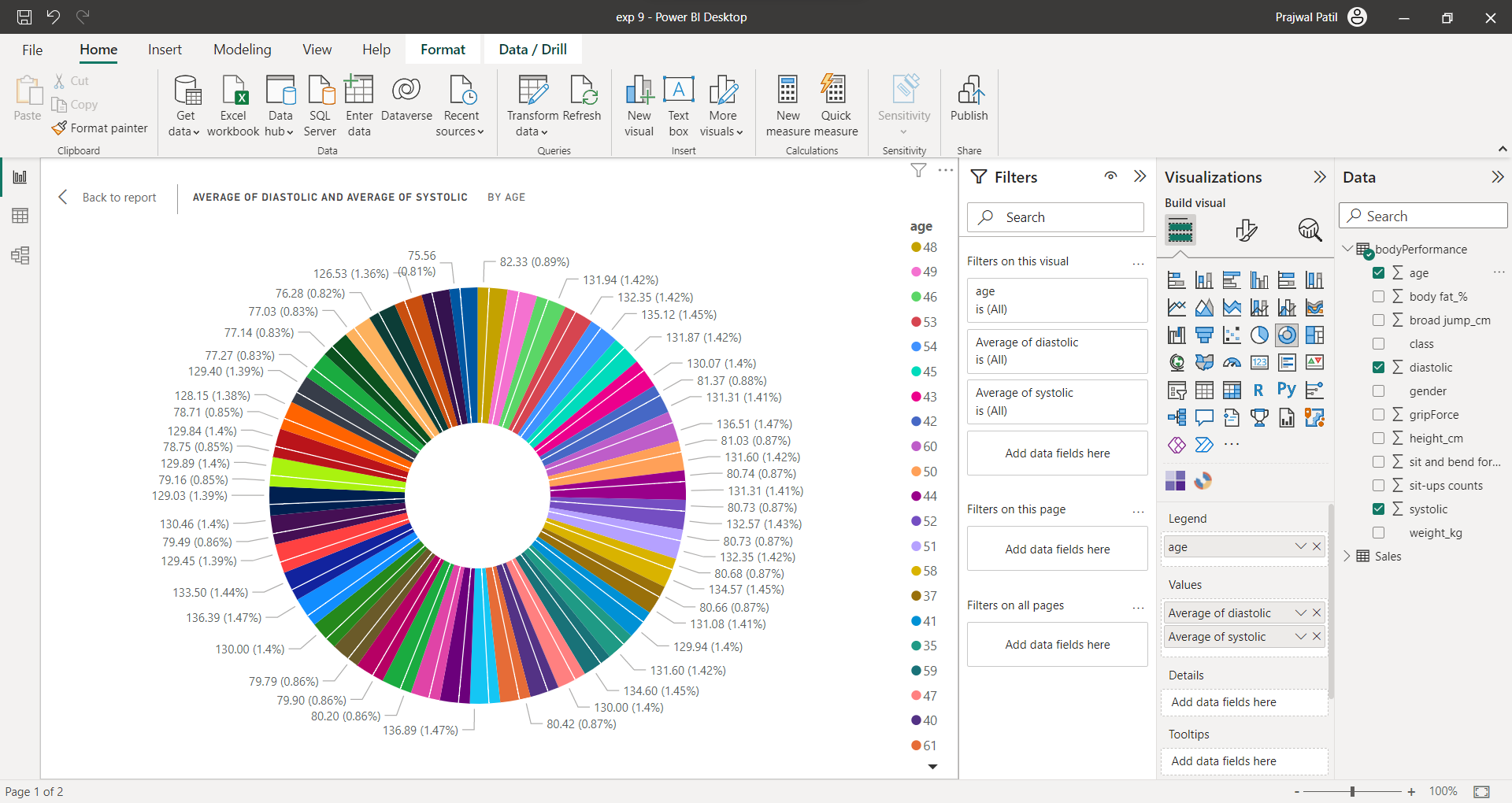
1. Bar plot of Average Body\_Fat% vs Age based on Gender

****

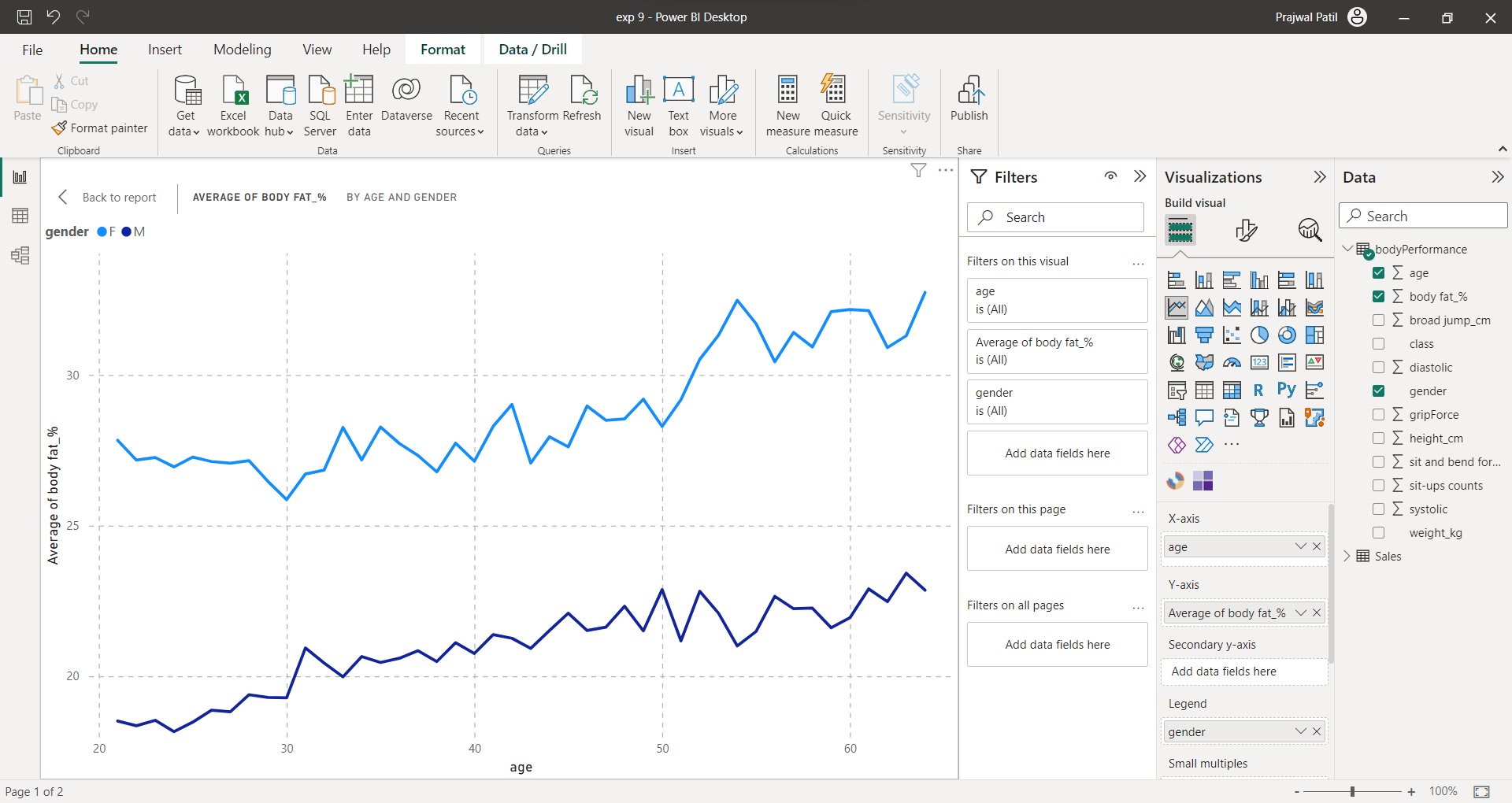
1. Pie Chart of Average Body\_Fat% based on Age

****

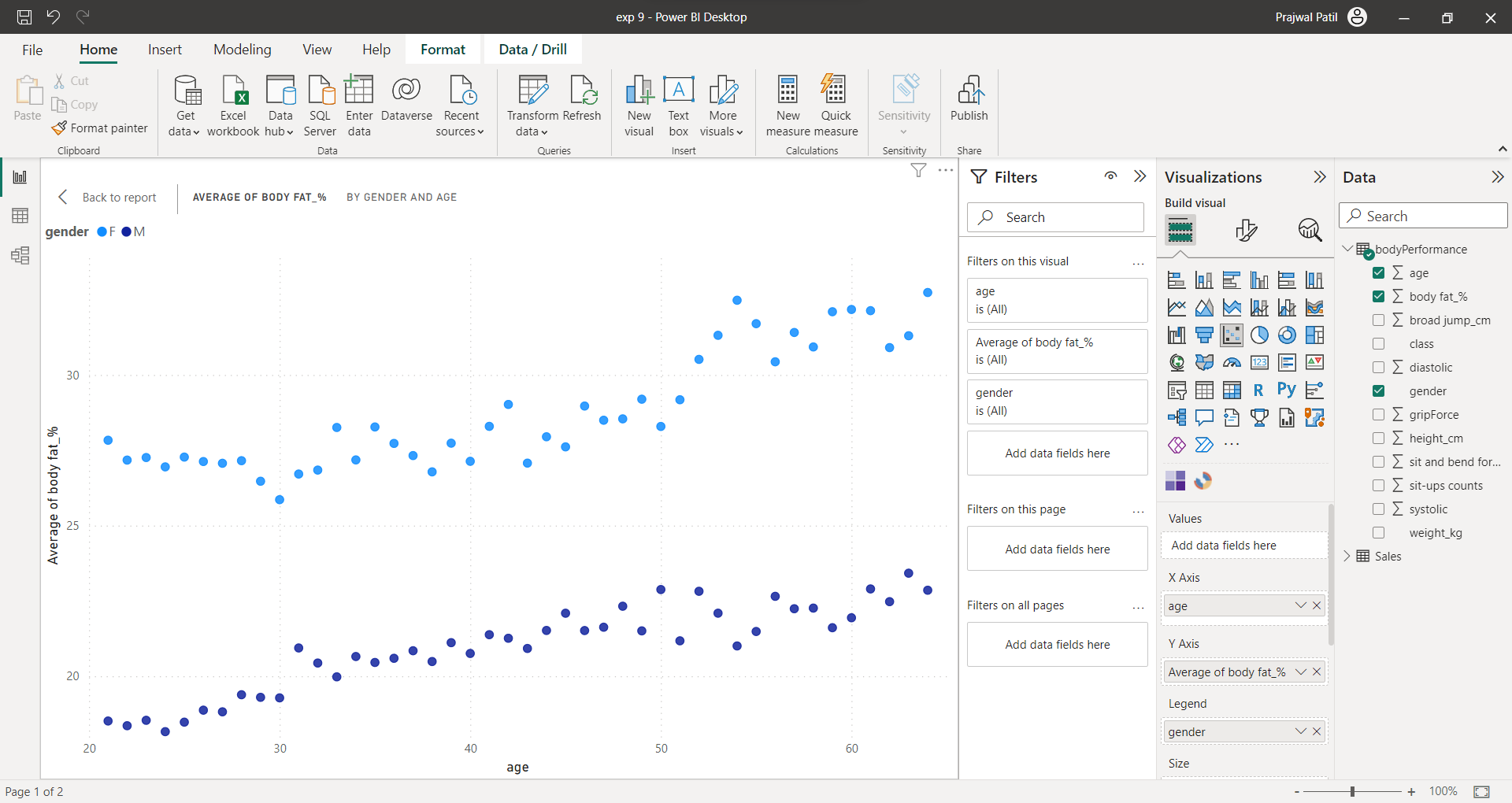
1. Donut Chart of Average Systolic vs Average Diastolic based on Age

****

1. Line Plot of Average Body\_Fat% vs Age based on Gender

****

1. Scatter Plot of Average Body\_Fat% vs Age based on Gender

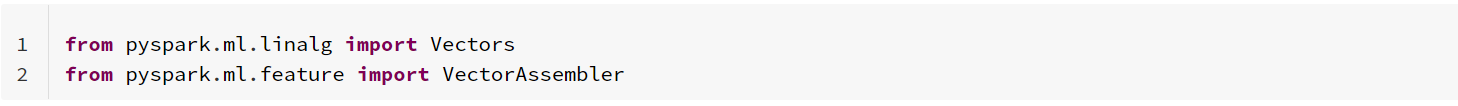
****

**Linear Regression using Apache Spark**

Dataset:<https://github.com/LeondraJames/Hyundai-Cruise-Ship-Crew-Prediction/blob/master/cruise_ship_info.csv>

1.Import library





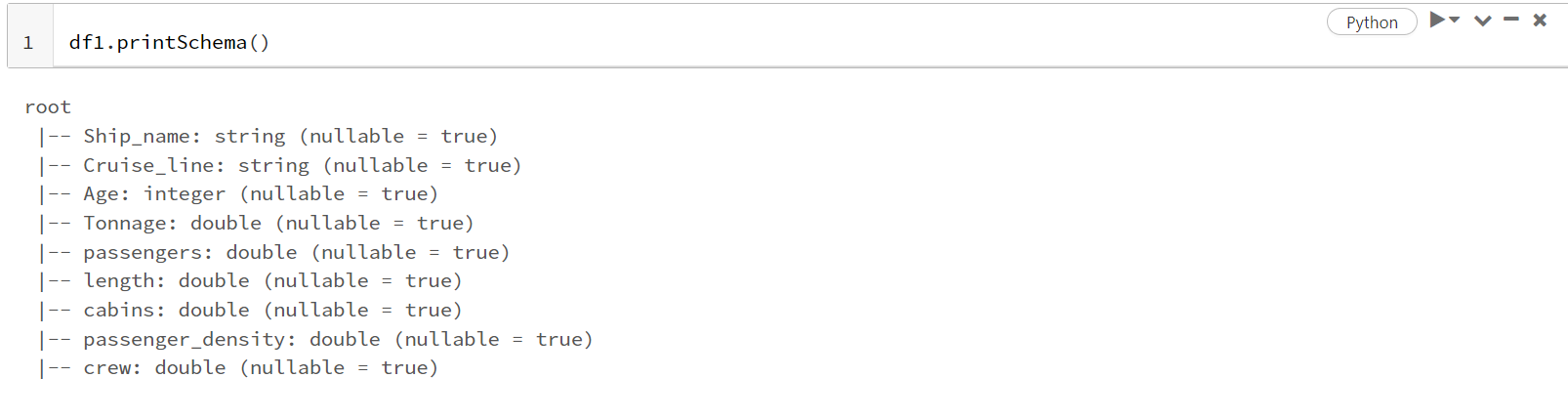
2.Create a new SparkSession using the builder pattern.



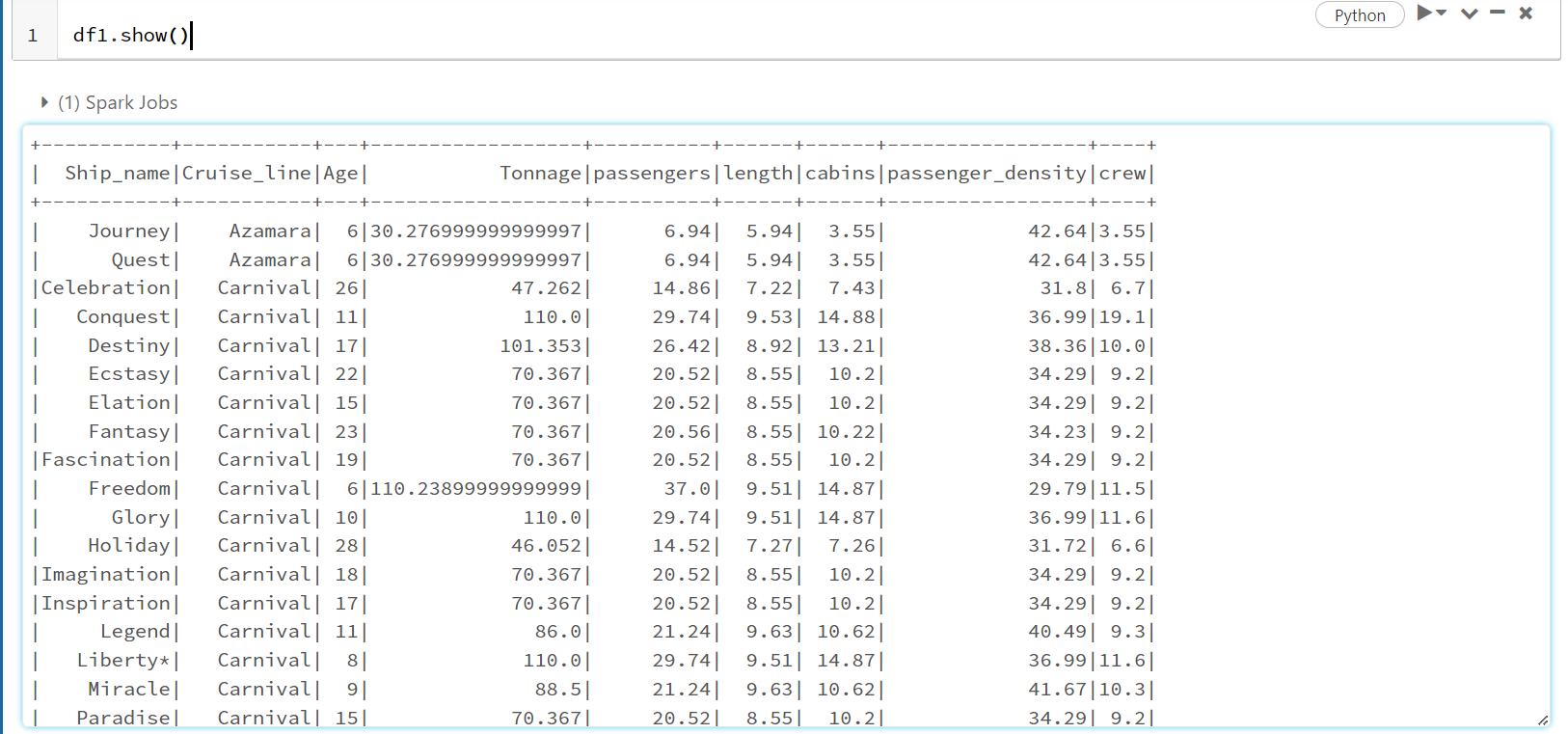
3.Read the dataset

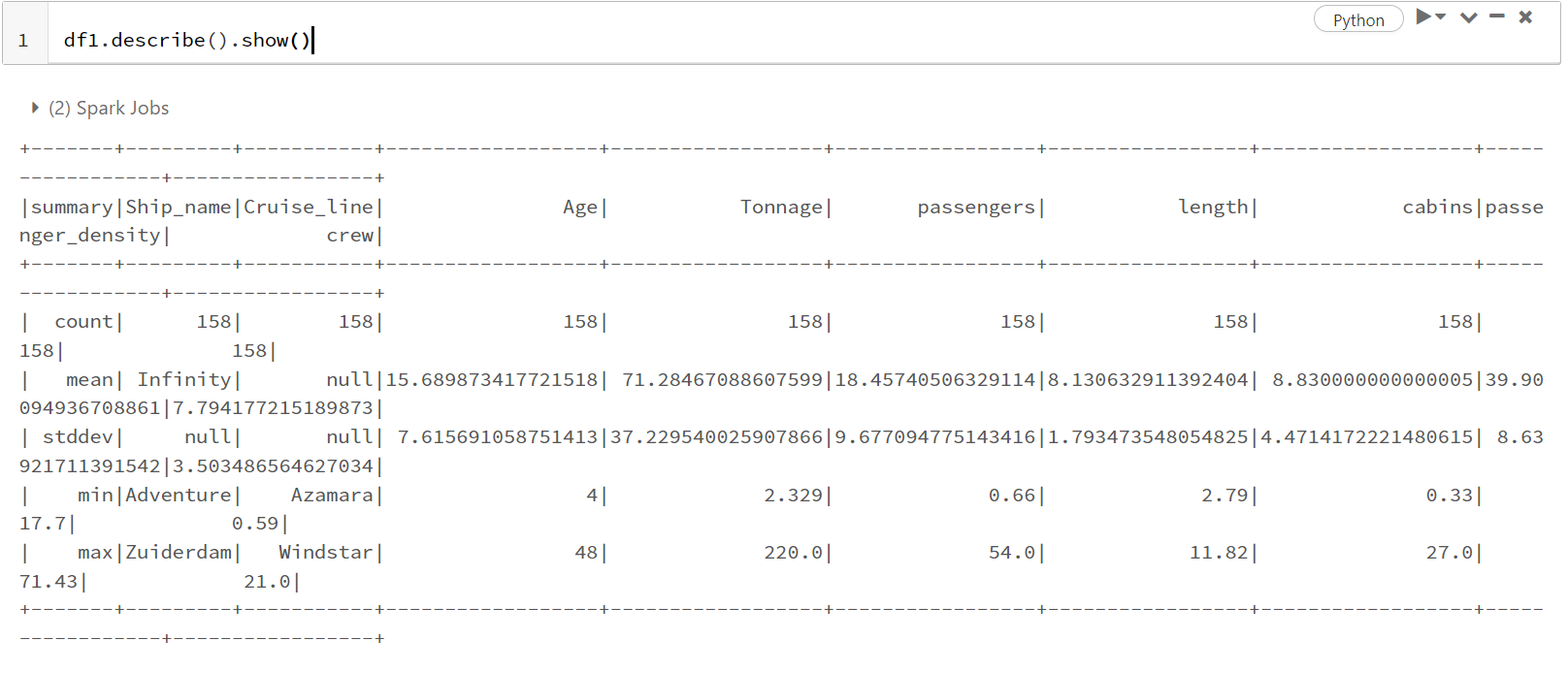


4.Printing the schema of dataset

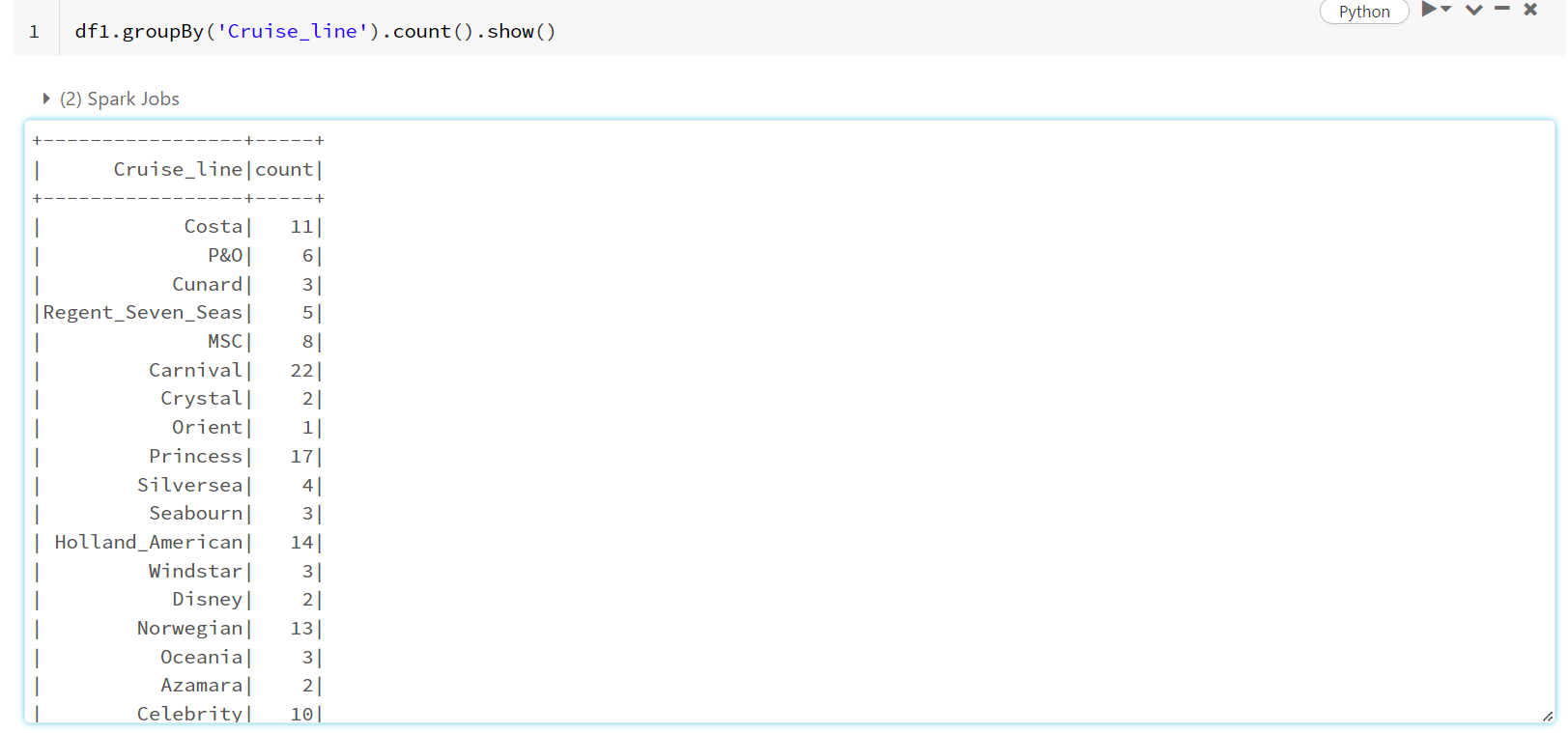


5.Understanding the contents of dataset





6. Count the number of occurrences of each unique value in this column



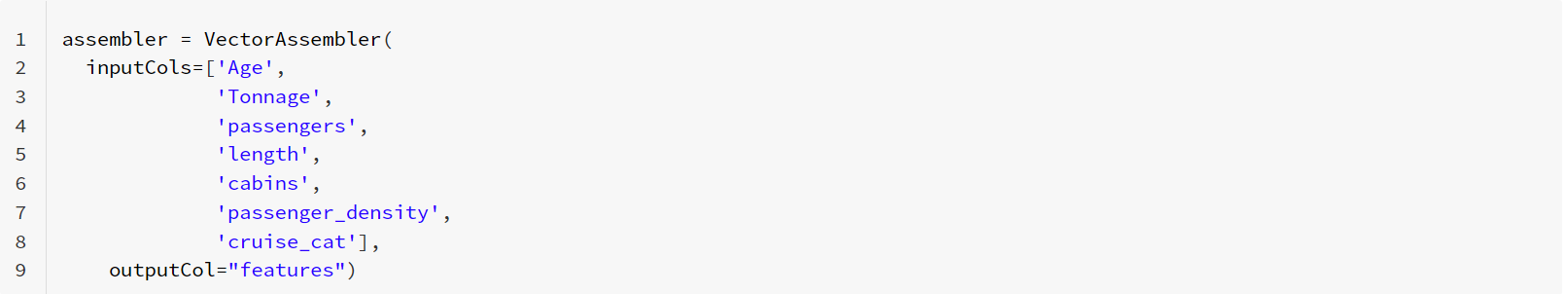
7.Convert the categorical column "Cruise\_line" in the DataFrame df1 to a numerical index column "cruise\_cat" in a new DataFrame indexed



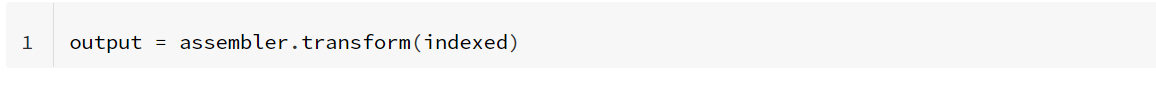
8.Printing new dataframe

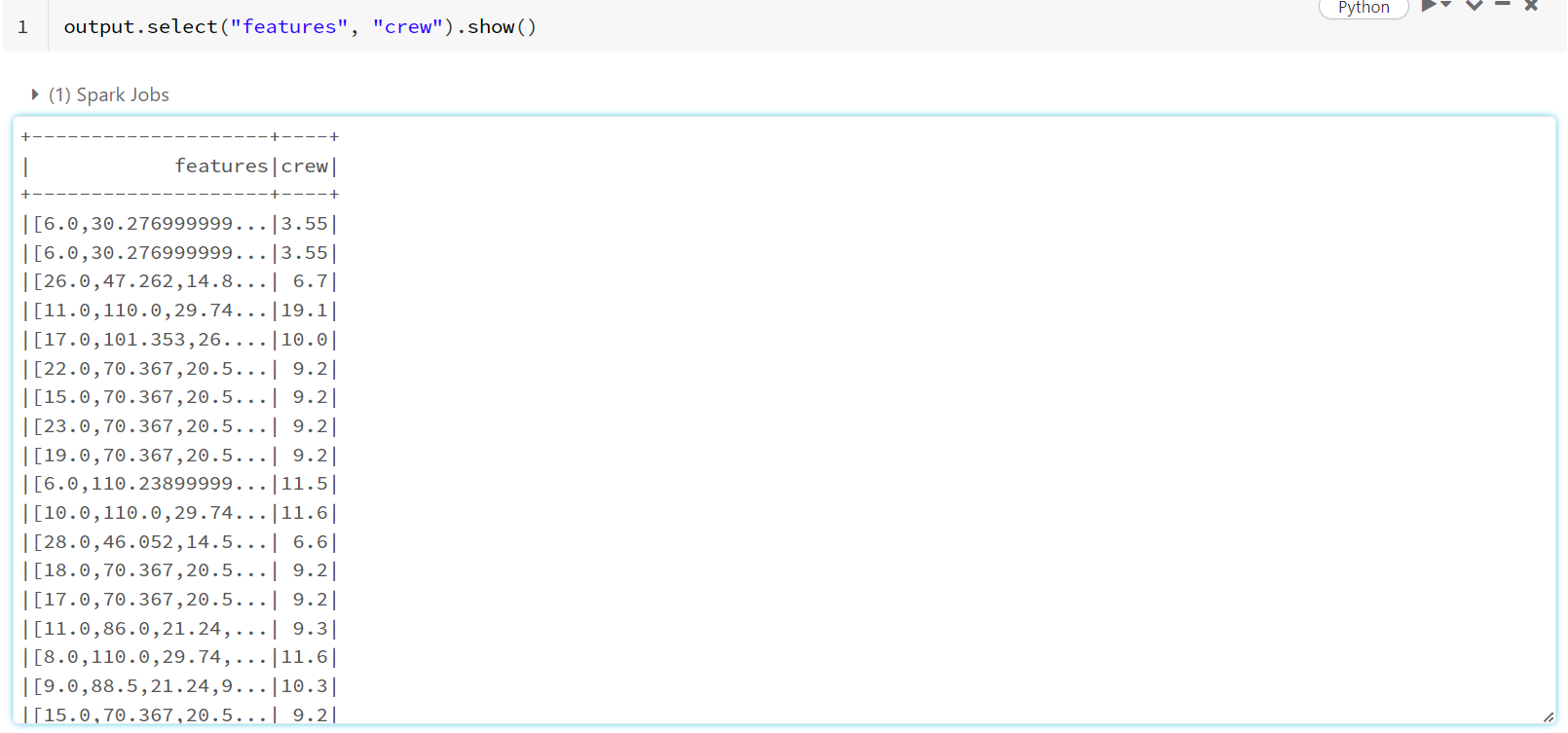


9. Create a VectorAssembler object from the pyspark.ml.feature module that combines multiple input columns in a DataFrame into a single output column of type Vector.

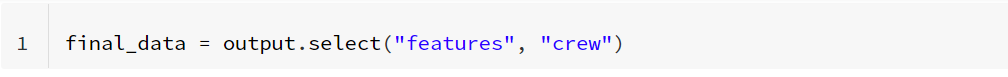


10.Applied the VectorAssembler transformation to the indexed DataFrame created earlier, which creates a new DataFrame output that contains a new column called "features" that combines all the specified input columns into a single vector column.

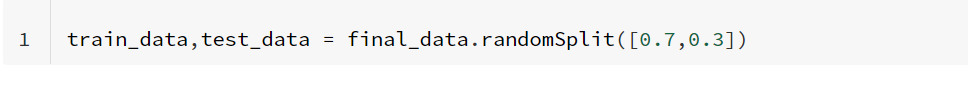




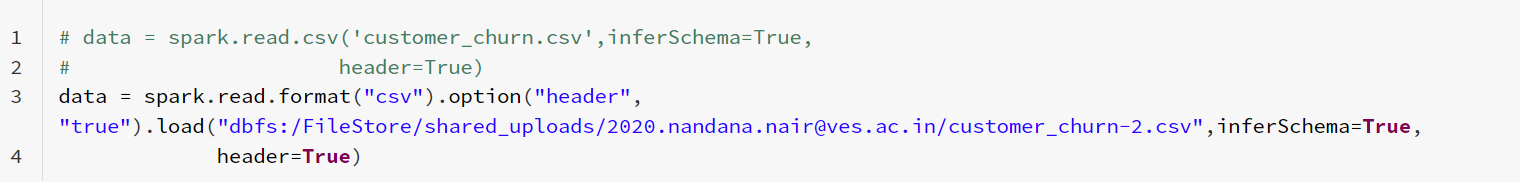
11.Select the "features" column and the "crew" column from the output DataFrame



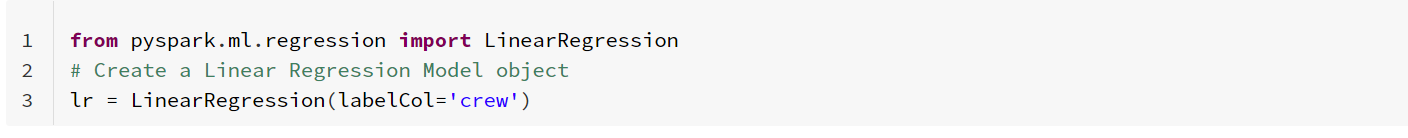
12. Splitting the dataset into train and test dataset



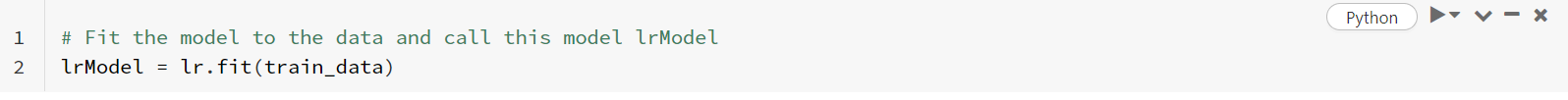
13.Importing the dataset

****

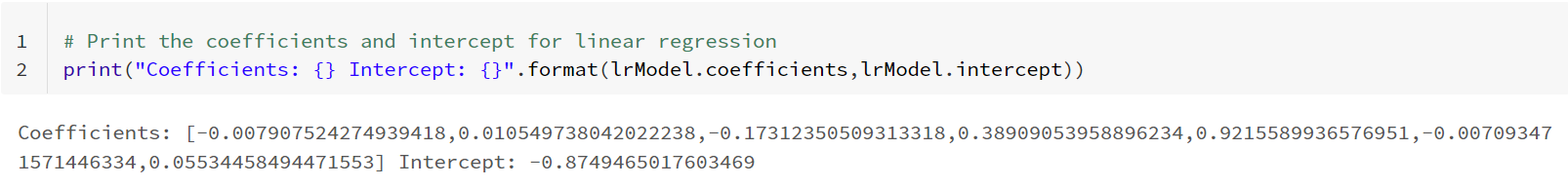
14.Model creation



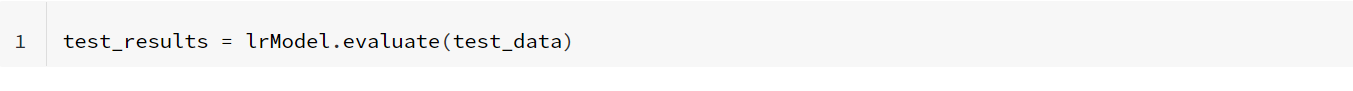
15.Fitting the model

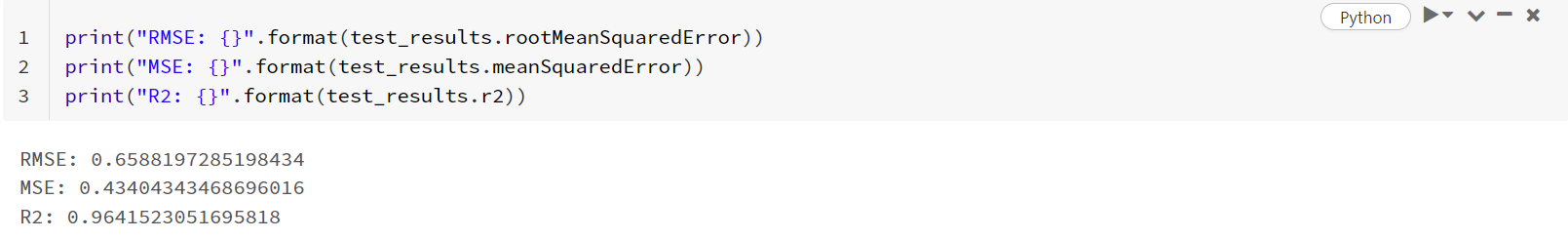


16.Printing coefficients and intercept

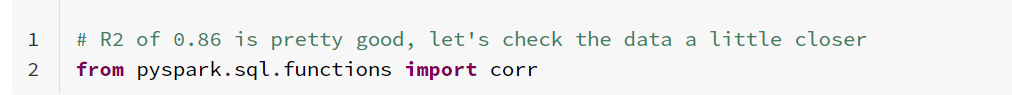


17.Model evaluation





18.Performing correlation

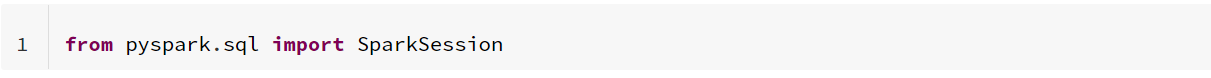




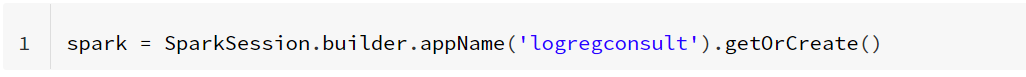
**Logistic regression using Apache spark**

Dataset:<https://github.com/SkalskiP/pySpark_Tutorial/blob/master/Sekcja_12_Logistic_Regression/new_customers.csv>

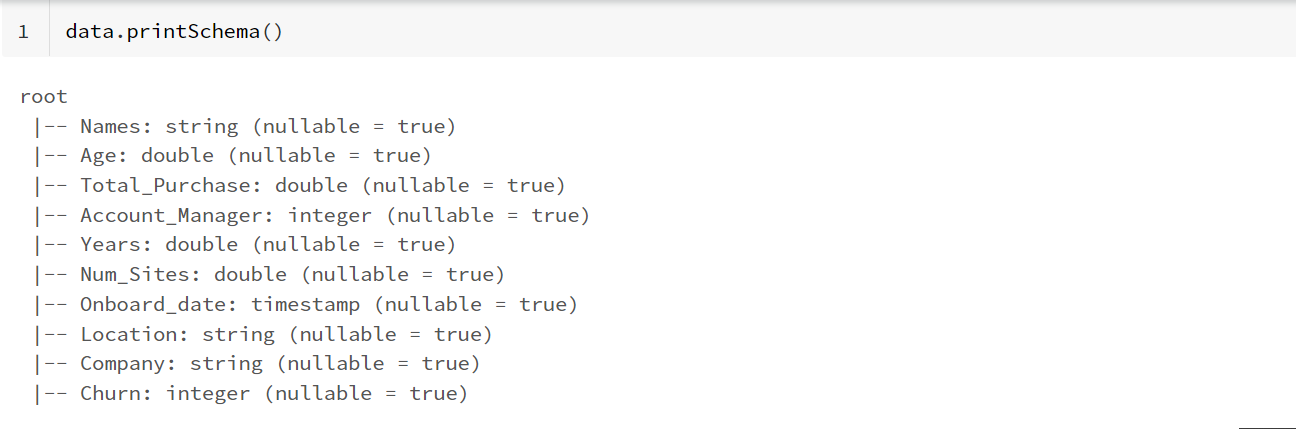
1. Importing pyspark

****

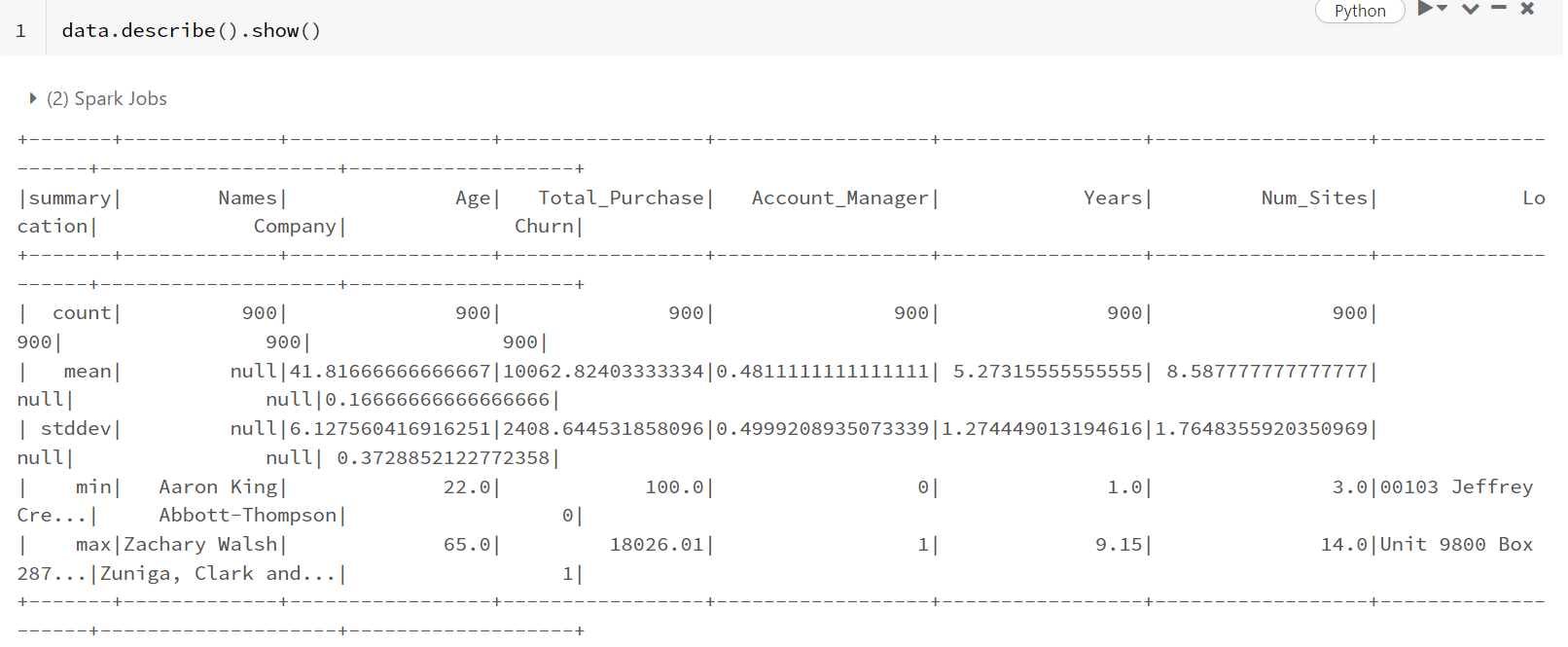
1. Initilize a new SparkSession

****

1. Printing the schema for the dataset.

****

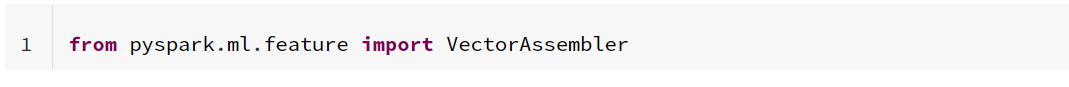
1. Computes and displays descriptive statistics of a DataFrame

****

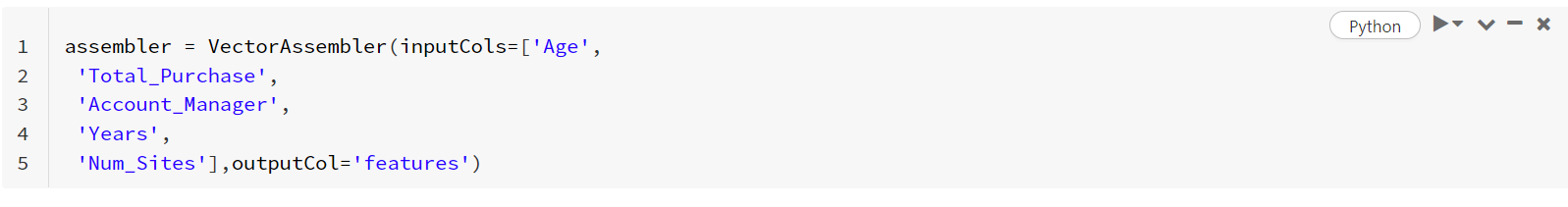
1. Print the columns

****

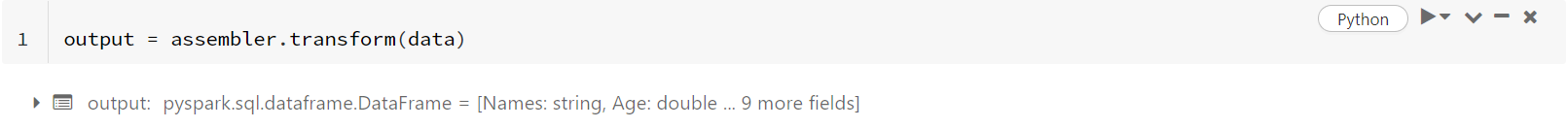
1. The VectorAssembler is a transformer in PySpark that is used to combine a given list of columns into a single vector column.

****

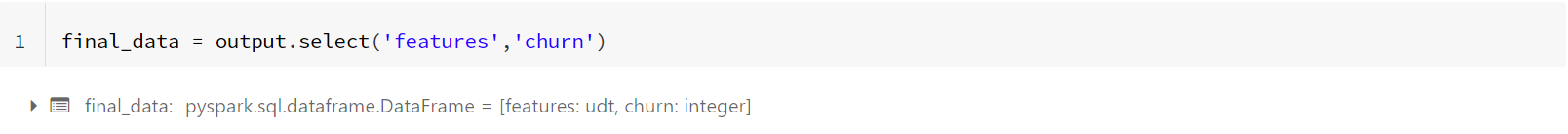
1. Here were taking the features as input and then outputting those features in a single vector column.

****

1. This code snippet applies the VectorAssembler to the DataFrame data and produces a new DataFrame output with an additional column that contains the vector representation of the input columns

****

1. This line of code creates a new DataFrame final\_data by selecting two columns from the output DataFrame: the features column and the churn column.

****

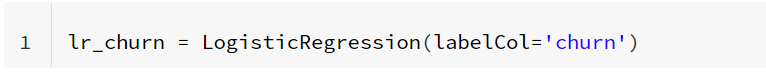
1. Splitting the dataset into train and test set

****

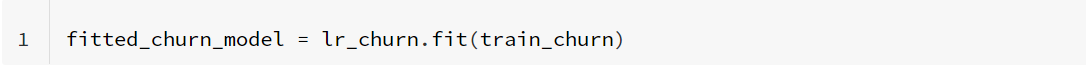
1. Importing LogisticRegression library

****

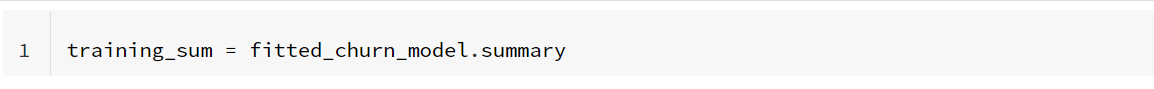
1. This line of code initializes a new LogisticRegression estimator in PySpark with the label column set to churn.

****

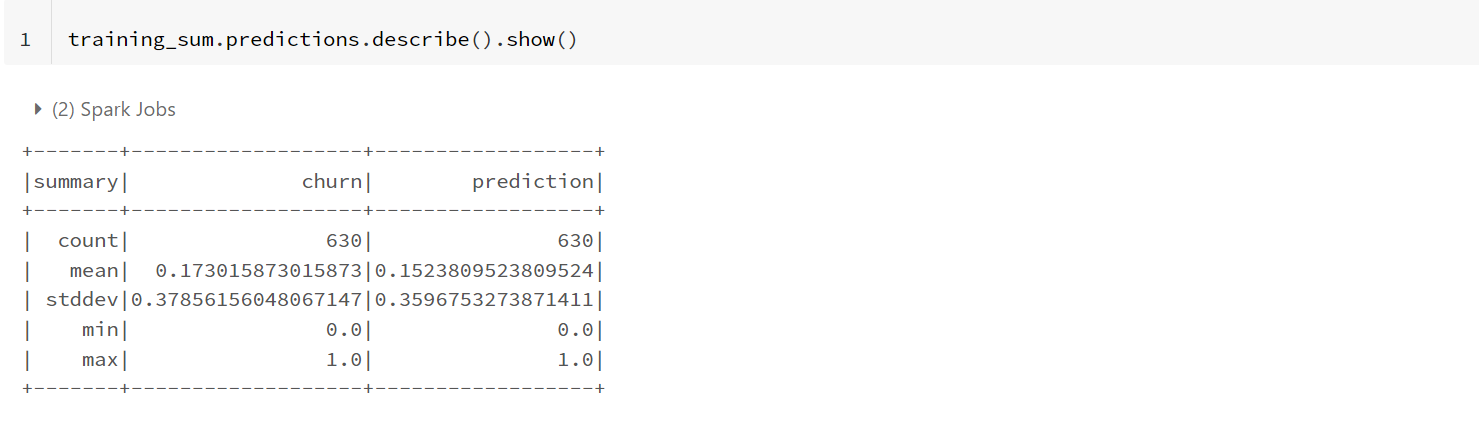
1. Fitting the train set in the model.

****

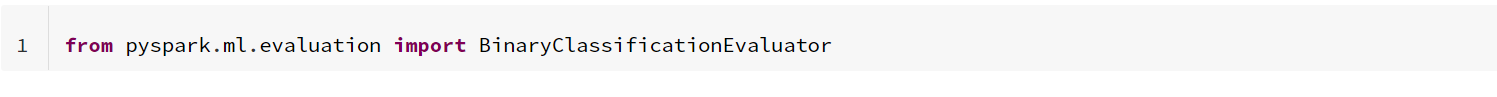
1. This code computes the summary statistics of the logistic regression model trained in the previous step and stores them in the training\_sum object.

****

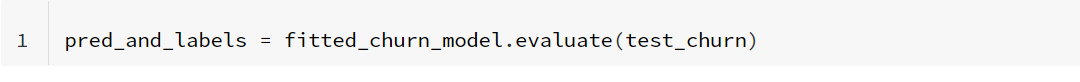
1. This code computes summary statistics of the predictions made by the logistic regression model on the training data, and displays them using the show() method.

****

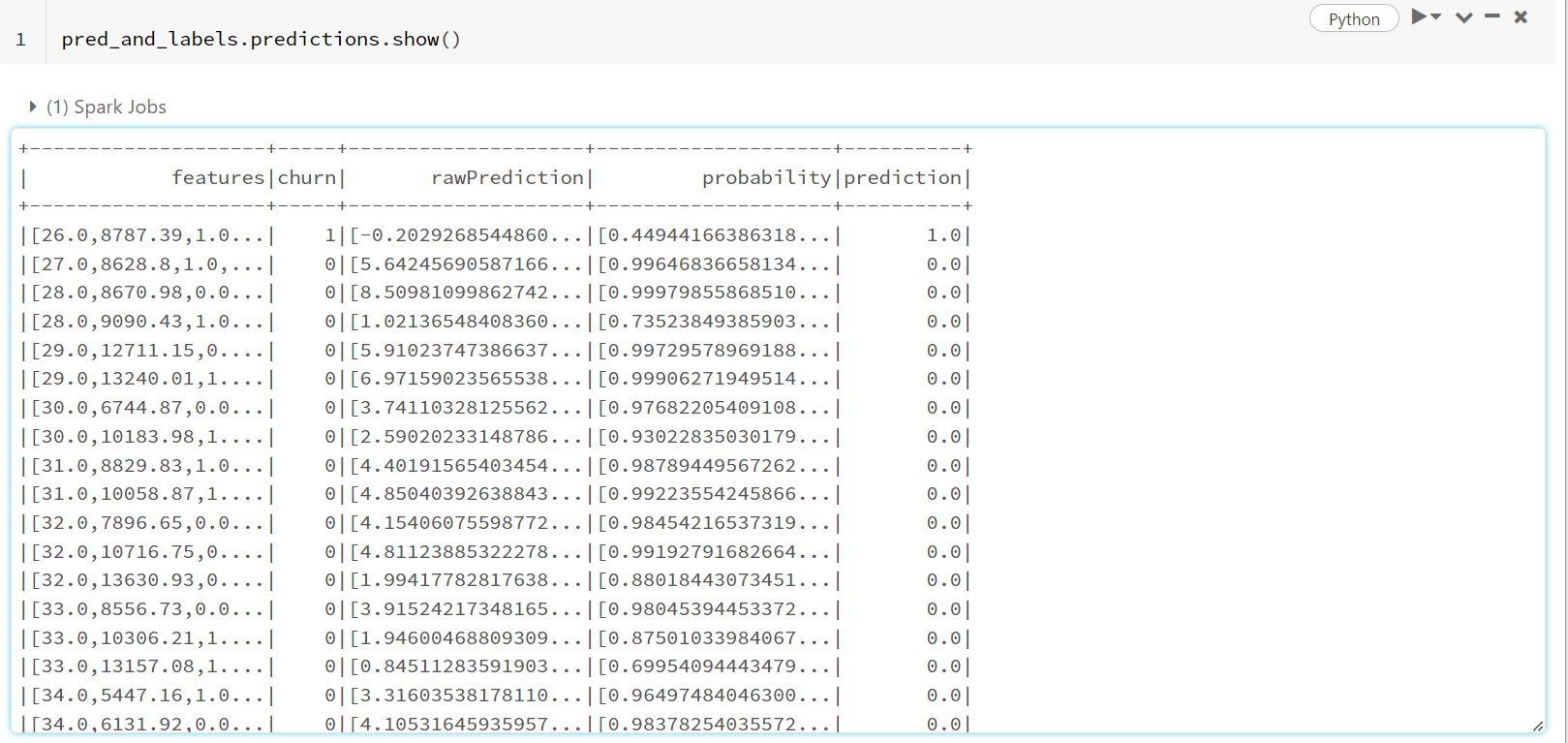
1. Importing Binary Classification Evaluator.

****

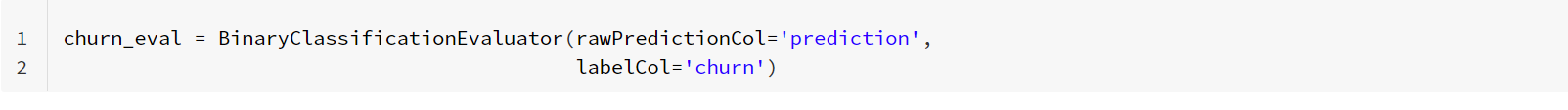
1. This code applies the trained logistic regression model fitted\_churn\_model to the test data test\_churn using the evaluate() method and produces a BinaryLogisticRegressionSummary object pred\_and\_labels as the output.

****

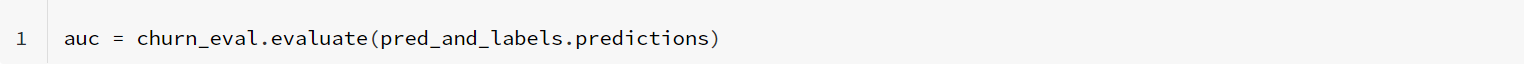
1. Display the predictions made by the Logistic Regression model.

****

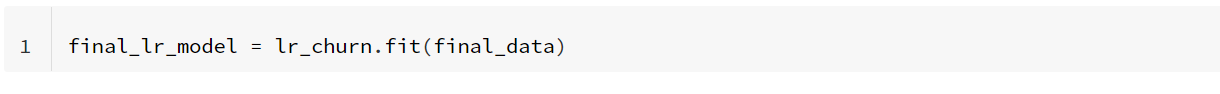
1. This code creates a BinaryClassificationEvaluator object churn\_eval with the parameters rawPredictionCol='prediction' and labelCol='churn'.

****

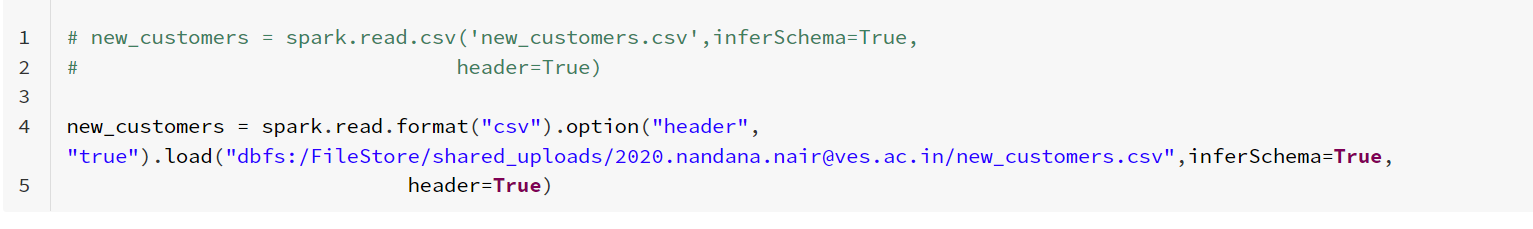
1. This code evaluates the performance of the logistic regression model on the test data

****

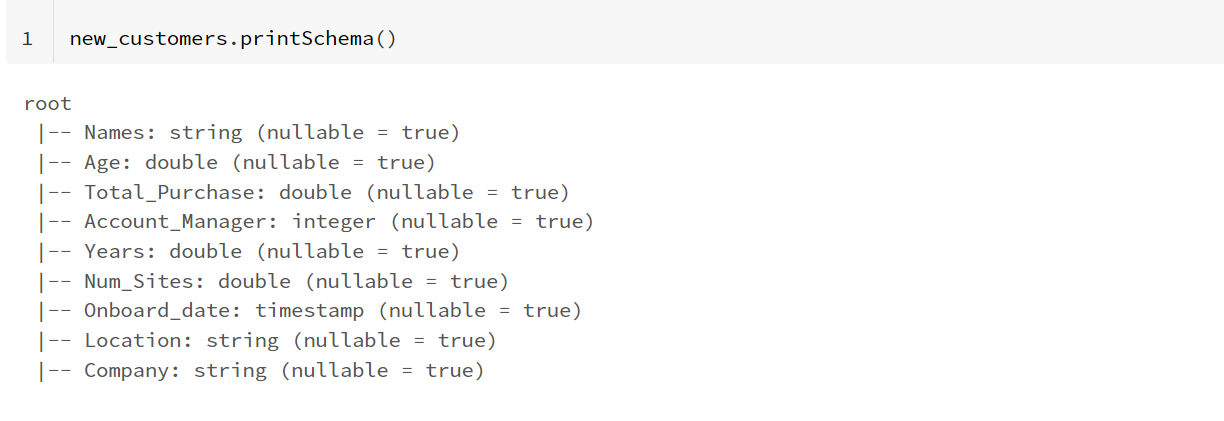
1. This code trains a logistic regression model on the entire dataset final\_data by calling the fit() method of the LogisticRegression object lr\_churn and assigning the resulting trained model to the variable final\_lr\_model

****

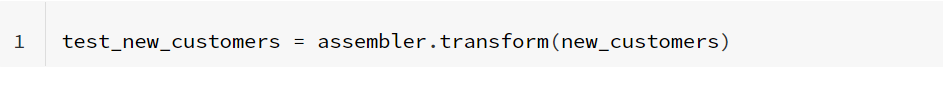
1. Uploading the dataset

****

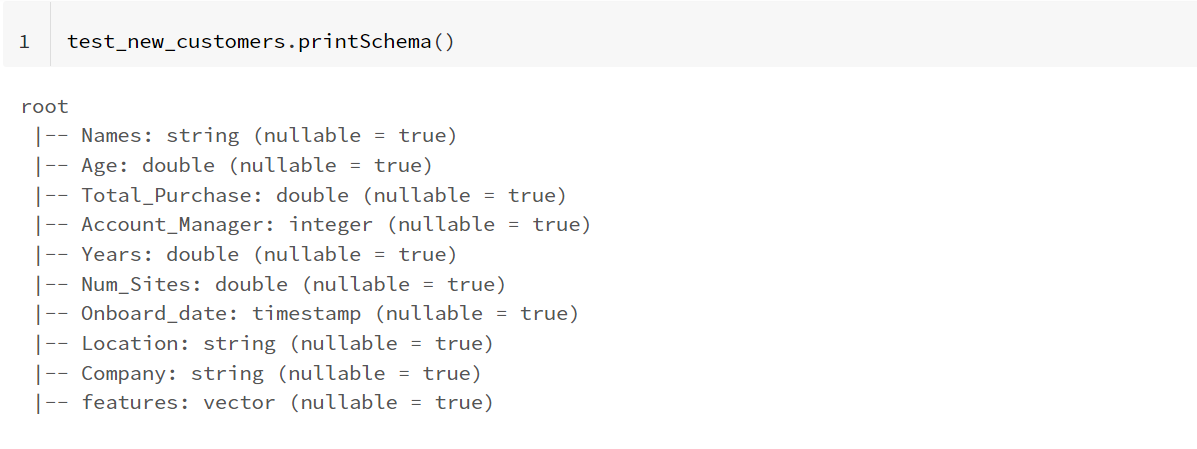
1. Printing the Schema

****

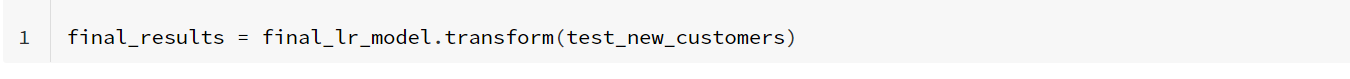
1. This code uses the transform() method of the VectorAssembler object assembler to transform the new\_customers DataFrame into a new DataFrame test\_new\_customers with a vector column features that contains the input features for each record.

****

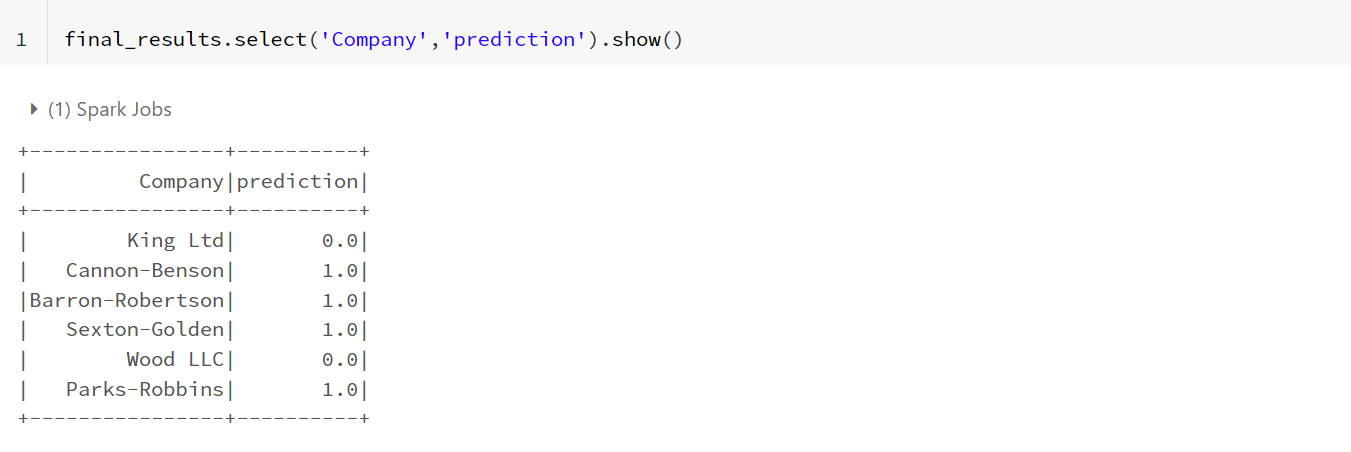
1. Printing Schema of the new Dataset

****

1. This code uses the transform() method of the trained logistic regression model final\_lr\_model to make predictions on new data in the DataFrame test\_new\_customers, and assigns the resulting DataFrame to the variable final\_results.

****

1. Printing the final predictions.

****

**CONCLUSION:**

In this experiment,we have performed data Visualization using Bar plot,Pie Chart,Donut Chart,Line Plot,Scatter Plot on Power BI.We have also performed linear and logistic regression using Apache Spark on Databricks.