Case Study 1: Retail Sales Forecasting

Background:
A retail company wants to leverage Azure Databricks to forecast sales for its multiple store locations. They have historical sales data and want to build a predictive model to improve inventory management.

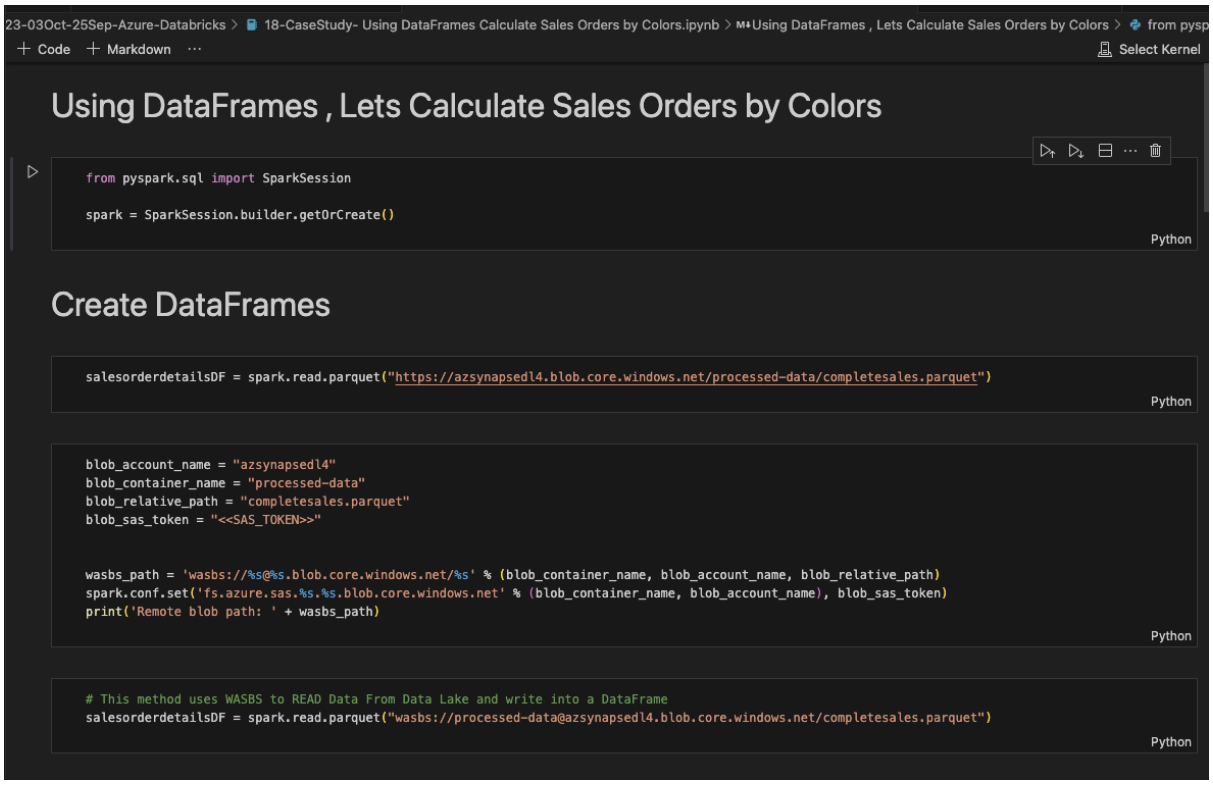
Tasks:

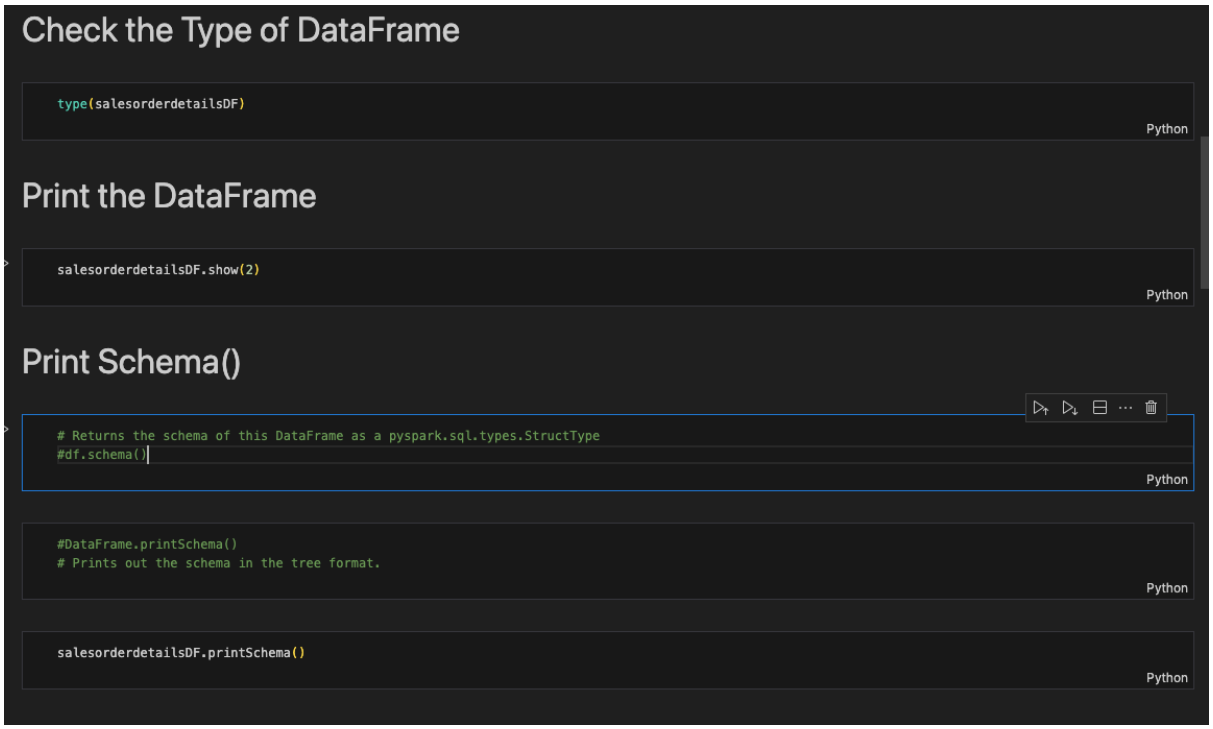
- Data ingestion and preprocessing using Databricks DataFrames.
- Learning about the sales by Various Paramaters
- Creating visualizations and dashboards in Databricks.

Inputs:

```
# For Tables
Customers   :   [SalesLT].[Customer]

Sales       :   [SalesLT].[SalesOrderDetail]

Product     :   [SalesLT].[Product]

# For Database
Database    :   bltmainretaildb

Username    :   bltadmin

Password    :   Database@1
```

Solution


23-03Oct-25Sep-Azure-Databricks > 18-CaseStudy- Using DataFrames Calculate Sales Orders by Colors.ipynb > Using DataFrames , Lets Calculate Sales Orders by Colors > from pysp
+ Code + Markdown ···
Select Kernel
Using DataFrames , Lets Calculate Sales Orders by Colors
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
Python
Create DataFrames
salesorderdetailsDF = spark.read.parquet("https://azsynapsedl4.blob.core.windows.net/processed-data/completesales.parquet")
Python
blob_account_name = "azsynapsedl4"
blob_container_name = "processed-data"
blob_relative_path = "completesales.parquet"
blob_sas_token = "<<SAS_TOKEN>>"
wasbs_path = 'wasbs://%s@%s.blob.core.windows.net/%s' % (blob_container_name, blob_account_name, blob_relative_path)
spark.conf.set('fs.azure.sas.%s.%s.blob.core.windows.net' % (blob_container_name, blob_account_name), blob_sas_token)
print('Remote blob path: ' + wasbs_path)
Python
# This method uses WASBS to READ Data From Data Lake and write into a DataFrame
salesorderdetailsDF = spark.read.parquet("wasbs://processed-data@azsynapsedl4.blob.core.windows.net/completesales.parquet")
Python


Check the Type of DataFrame
type(salesorderdetailsDF)
Python
Print the DataFrame
salesorderdetailsDF.show(2)
Python
Print Schema()
Returns the schema of this DataFrame as a pyspark.sql.types.StructType
#df.schema()
Python
#DataFrame.printSchema()
Prints out the schema in the tree format.
Python
salesorderdetailsDF.printSchema()
Python

Reading Data from Parquet File

```python
# spark.read.parquet(d).show()
```

Python

```python
# This method uses WASBS to READ Data From Data Lake and write into a DataFrame
salesorderdetailsDF = spark.read.parquet("wasbs://processed-data@azsynapsedl4.blob.core.windows.net/completesales.parquet")
```

Python

```python
salesorderdetailsDF.show(5)
```

Python

```python
display(salesorderdetailsDF)
```

Python

```python
salesorderdetailsDF.printSchema()
```

Python

```python
salesorderformattedparquetDF = salesorderdetailsDF.select(['SalesOrderID','OrderQty','UnitPrice','UnitPriceDiscount','Color'])
```

Python

```python
salesorderformattedparquetDF.show()
```

Python

```python
type(salesorderformattedparquetDF)
```

Python

CREATE A VIEW USING DATAFRAME

```python
#df.createGlobalTempView("people")

OR

# df.createTempView("people")
```

Python

```python
salesorderformattedparquetDF.createTempView('salesorderbycolor')
```

Python

WRITING A SQL QUERY

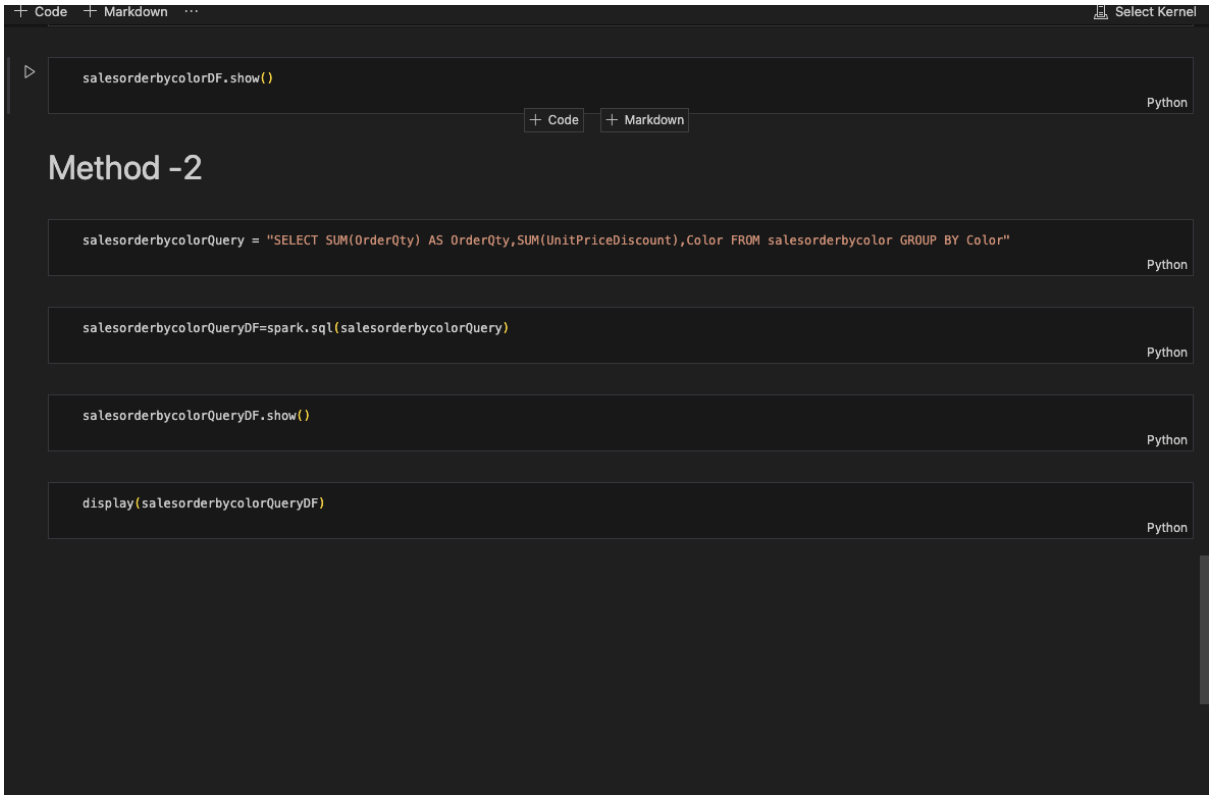
Method -1

```python
salesorderbycolorDF = spark.sql("""SELECT * FROM salesorderbycolor WHERE Color='Blue'""")
```

Python

```python
salesorderbycolorDF.show()
```

Python


Code
Markdown
Select Kernel
salesorderbycolorDF.show()
Python
Code
Markdown
Method -2
salesorderbycolorQuery = "SELECT SUM(OrderQty) AS OrderQty,SUM(UnitPriceDiscount),Color FROM salesorderbycolor GROUP BY Color"
Python
salesorderbycolorQueryDF=spark.sql(salesorderbycolorQuery)
Python
salesorderbycolorQueryDF.show()
Python
display(salesorderbycolorQueryDF)
Python