Data Loading and Exploration using PySpark

Introduction:

In this experiment, we use **PySpark** to load a CSV dataset, inspect its schema, and perform basic data exploration. PySpark is a powerful distributed computing framework that allows handling large datasets efficiently. This task demonstrates how to read data, examine its structure, and perform simple transformations.

Code Implementation:

```
# Step 1: Install and Import PySpark
!pip install pyspark

from pyspark.sql import SparkSession

# Step 2: Create a Spark Session
spark = SparkSession.builder.appName("PySparkDataLoad").getOrCreate()

# Step 3: Download a sample CSV file
import urllib.request
url = "https://people.sc.fsu.edu/~jburkardt/data/csv/airtravel.csv"
local_file = "airtravel.csv"
urllib.request.urlretrieve(url, local_file)

# Step 4: Load the CSV into a Spark DataFrame
df = spark.read.csv(local_file, header=True, inferSchema=True)

# Step 5: Show the schema of the DataFrame
df.printSchema()

# Step 6: Display the first 5 rows
df.show(5)

# Step 7: Display summary statistics
df.describe().show()

# Step 8: Stop the Spark session
spark.stop()
```

Conclusion:

In this lab, we successfully used PySpark to read a CSV file, inspect its schema, view the data, and calculate summary statistics. This demonstrates the ease and efficiency of PySpark for handling structured data. Such techniques are useful for data preprocessing and analysis in large-scale projects.