TAXI FARE PREDICTION SYSTEM

USER MANUAL

INTRO TO BIG DATA CSCI 6444

Environment Setup:

- Create a .env file in the same directory as your Node.js scripts with your AWS credentials and other required environment variables as outlined in the source code files.
- Install the required Node.js packages by running npm install in the directory of your scripts.
- Ensure Python with PySpark is installed for data processing and that you have the necessary Python packages installed.

Running the Pipeline:

Step 1: Uploading Raw Data to S3

- Use the upload-taxi-data.js script to upload raw taxi data from NYC TLC to an S3 bucket for a specified year.
- Execute the script with the following command:
  ‘node upload-taxi-data.js’

Step 2: Converting Parquet to CSV and Cleaning

- Use the parquet-csv.js script to list parquet files from S3, convert them to CSV, clean them, and then upload them to another S3 bucket.
- Execute the conversion script by passing the year as an argument:
  ‘**node parquet-csv.js 2018**’ (do for all the years till 2024)

This script will internally call process.py to utilize PySpark for data cleansing and transformation.

Step 3: Feature Engineering, Model Training, and Running Colab Notebook

- Open your Google Colab/Jupyter Notebook that is prepared for data cleaning, EDA, feature engineering, and model training.
- To run the analysis, simply use the Run all feature in Google Colab, which will execute all cells in the notebook sequentially.

This process will include:

- Further cleaning of the CSV data.
- Exploratory Data Analysis (EDA).
- Feature engineering.
- Splitting the data into training and test sets.
- Training the machine learning model using algorithms like Distributed Random Forest, Gradient Boosting, etc.
- Evaluating the model with performance metrics like RMSE and R^2.

Step 4: Making Predictions and Visualization

- The notebook should also contain cells that use the trained model to make predictions on test data.
- Visualizations of predicted fares and model performance can be done using libraries like Matplotlib and Seaborn within the notebook.

Final Notes:

- Verify you have the correct permissions for the S3 buckets used in the scripts.
- The process.py script should be properly referenced in parquet-csv.js and be present in the same directory or have the paths set correctly.
- Update any bucket names and URLs in the scripts to match your AWS environment and data source.
- Always monitor the output logs for errors and address any issues with AWS permissions by adjusting your IAM user policies as needed.

By following these steps and using the ‘*Run all’* feature in Google Colab, you can ensure that the entire data pipeline runs smoothly from raw data upload to data transformation and analysis.