Learn to Build an End-to-End Machine Learning Pipeline - Part 1 Project Overview

Overview

The project addresses a critical challenge faced by the logistics industry. Delayed truck shipments not only result in increased operational costs but also impact customer satisfaction. Timely delivery of goods is essential to meet customer expectations and maintain the competitiveness of logistics companies.

By accurately predicting truck delays, logistics companies can:

- Improve operational efficiency by allocating resources more effectively
- Enhance customer satisfaction by providing more reliable delivery schedules
- Optimize route planning to reduce delays caused by traffic or adverse weather conditions
- Reduce costs associated with delayed shipments, such as penalties or compensation to customers

This project is the first part of a three-part series aimed at solving the truck delay prediction problem. In this initial phase, we will utilize PostgreSQL and MySQL in AWS RDS to store the data, set up an AWS Sagemaker Notebook, perform data retrieval, conduct exploratory data analysis, create feature groups with Hopsworks, data processing, and feature engineering. We aim to build a pipeline that prepares the data for model building.

Note: AWS Usage Charges

This project leverages the AWS cloud platform to build the end-to-end machine learning pipeline. While using AWS services, it's important to note that certain activities may incur charges. We recommend exploring the AWS Free Tier, which provides limited access to a wide range of AWS services for 12 months. Please refer to the **AWS Free Tier page** for detailed information, including eligible services and usage limitations.

Aim

The primary objective of this project is to create an end-to-end machine learning pipeline for truck delay classification. This pipeline will encompass data fetching, creating a feature store, data preprocessing, and feature engineering.

Data Description

The project involves the following data tables:

- City Weather: Weather data for various cities
- Routes: Information about truck routes, including origin, destination, distance, and travel time
- Drivers: Details about truck drivers, including names and experience
- Routes Weather: Weather conditions specific to each route
- Trucks: Information about the trucks used in logistics operations
- Traffic: Traffic-related data
- Truck Schedule: Schedules and timing information for trucks

Tech Stack

➔ Language: Python, SQL
➔ Libraries: NumPy, Pandas, PyMySQL , Psycopg2, Matplotlib, Seaborn
➔ Data Storage: PostgreSQL,MySQL, AWS RDS, Hopsworks
➔ Data Visual Tool(SQL): MySQL Workbench, Pgadmin4
➔ Feature Store: Hopsworks
➔ Cloud Platform: AWS Sagemaker

Approach

- Introduction to End-to-End Pipelines:
    - Understanding the fundamental concepts and importance of end-to-end pipelines

- Database Setup:
    - Creating AWS RDS instances for MySQL and PostgreSQL
    - Setting up MySQL Workbench and pgAdmin4 for database management

- Data Analysis:
    - Performing data analysis using SQL on MySQL Workbench and pgAdmin4

- AWS SageMaker Setup

- Exploratory Data Analysis (EDA):
    - Conducting exploratory data analysis to understand essential features and the dataset's characteristics

- Feature Store:

    - Understanding the concept of a feature store and its significance in machine learning projects
    - Understanding how Hopsworks works to facilitate project creation and feature group management

- Data Retrieval from Feature Stores

- Fetching data from feature stores for further analysis

- Data Preprocessing and Feature Engineering

- Data Storage:
    - Storing the final engineered features in the feature store for easy access and consistency

Code Folder Overview:

Once you unzip the code.zip file, you can find the following folders:

```
├─ Data
│  ├─ Database_backup
│  │  ├─ truck-eta-mysql.sql
│  │  └─ truck-eta-postgres.sql
│  ├─ data_description.pdf
│  └─ Training_data
│     ├─ city_weather.csv
│     ├─ drivers_table.csv
│     ├─ routes_table.csv
│     ├─ routes_weather.csv
│     ├─ traffic_table.csv
│     ├─ trucks_table.csv
│     └─ truck_schedule_table.csv
├─ Notebook
│  └─ Truck_Delay_Classification.ipynb
├─ References
│  ├─ readme.md
│  ├─ requirements.txt
│  └─ solution methdology.pdf
└─ SQL Commands
   ├─ truck-delay-mysql.sql
   └─ truck-delay-postgres.sql
```

Here is a brief information on the folders:

1. Data
2. Notebook
3. References
4. SQL Commands

1. The data folder contains database backup files in both MySQL and PostgreSQL formats. These files can be used to restore a database. It also contains training data files in CSV format and data description.

2. The notebook folder contains the original ipython notebook as in the lectures. Note that this notebook contains code as well as observation and other information as per videos.

3. The reference folder contains a README.md file, which contains documentation and instructions, and a requirements.txt file has all the required libraries with respective versions and solution methodology of the project.

4. The SQL commands folder contains MySQL and PostgreSQL analysis scripts.

Project Takeaways

1. What are end-to-end machine learning pipelines, and why are they important?
2. How to create an AWS RDS instance?
3. How to connect with the database using MYSQL Workbench and Pgadmin4?
4. How to recover data using bak files in a database?
5. How to do Data Analysis using SQL?
6. How to set up an AWS Sagemaker Notebook?
7. How to fetch data from AWS RDS using Python?
8. Understanding Business Insights through Exploratory Data Analysis
9. Understand the significance of Feature Stores
10. Learn about Hopsworks Feature Store
11. How to create feature groups in Hopsworks?
12. How to fetch data from Hopsworks?
13. Understand data preprocessing to ensure quality and consistency
14. Learn different feature engineering techniques