Data Preparation and Documentation for Project Data

Title: Data Preparation and Documentation for Project Data

ILO 5.1 Responsible: Alexi Kehayias(Group 24)

Data Preparation

1. Combining Relevant Tables:

- **Objective**: The project integrates data from various tables, including vehicle-related tables (safe_driving, breda_road) and weather-related tables (precipitation, temperature, greenery), for machine learning purposes.
- **Justification**:
    - Time series models require comprehensive datasets that capture temporal relationships and contextual factors.
    - Integrating vehicle-related and weather-related data ensures that models can learn from a holistic view of the driving environment and conditions.
    - This approach helps capture interactions between different data sources, such as how weather conditions impact driving behaviour and safety.

---

2. Database Connection:

- **Steps**:
    - Establish a connection to the PostgreSQL database using provided credentials to access the required tables.
- **Justification**:
    - Direct access to the database allows efficient retrieval and management of large datasets.
    - This is essential for training deep learning models that require substantial computational resources and ensures that the data is up-to-date and readily available for analysis.

---

3. Displaying Tables:

- **Steps**:
  - Retrieve and display the list of tables available in the specified schema (group24_warehouse) to confirm the data structure.
- **Justification**:
  - Understanding the data structure and available tables is crucial for ensuring that all necessary data is included and appropriately linked.
  - This step helps identify the relationships between tables, ensuring that the time series analysis captures all relevant data points.

---

4. Dropping Unnecessary Tables:

- **Steps**:
  - Remove any redundant tables from the database to streamline the data processing workflow.
- **Justification**:
  - Removing redundant tables reduces clutter and potential confusion, streamlining the workflow and focusing resources on relevant data for model training.
  - This helps in maintaining a clean and organized database, essential for efficient data processing.

Data Cleansing and Preprocessing

1. Reading Data from Tables:

- **Steps**:
  - Load data from safe_driving, breda_road, precipitation, temperature, and greenery tables into Pandas DataFrame.
- **Justification**:
  - Loading data into DataFrame allows for flexible and powerful data manipulation.
  - DataFrame provide a convenient structure for handling large datasets and performing operations such as merging, filtering, and transformation, which are essential for preparing data for time series models.

---

2. Data Cleansing:

- **Convert date columns to datetime format**:
  - **Justification**: Ensures that temporal data is accurately interpreted and can be used for time-based indexing, aggregation, and resampling. These operations are fundamental in time series analysis to capture trends, seasonal patterns, and other temporal dependencies.
- **Handling missing values**:
  - **Justification**: Time series models are sensitive to missing values. Filling them with default values or interpolating ensures continuity in the data, preventing disruptions in the learning process. This step is crucial for maintaining the integrity of the time series data and avoiding gaps that could lead to inaccurate predictions.
- **Adding latitude and longitude**:
  - **Justification**: Geographical information is critical for contextualizing time series data, especially when analyzing events that are spatially distributed. This helps in creating models that account for location-based variations, which can significantly impact the outcomes being predicted.

---

3. Detecting and Handling Outliers:

- **Location-based outliers**:
  - **Justification**: Ensures that the data used for training is geographically consistent and representative of the study area. This reduces noise and improves model accuracy by focusing on data points relevant to the specific geographic context.
- **IQR method for weather data**:

- **Justification**: Removes extreme values that could skew the model training, ensuring that the model learns from typical conditions rather than anomalies. This step helps in building robust models that generalize well to new data.

---

4. Feature Engineering:

- **Adding `time_of_day`:**
  - **Justification**: Capturing the time of day can reveal daily patterns and trends, which are crucial for time series models to learn temporal dependencies. For example, traffic patterns and weather conditions may vary significantly between morning, afternoon, evening, and night.
- **Calculating Euclidean distance to landmarks**:
  - **Justification**: Provides additional contextual information that can influence time series patterns. For instance, the proximity to known locations may affect driving behaviour and incident severity, adding valuable context to the analysis.

Creating and Populating Normalised Tables

1. Creating Normalised Tables:

- **Steps**:
  - Define the schema for normalised tables in the database to store cleaned and pre-processed data.
- **Justification**:
  - Structured and normalised data tables ensure consistency, reduce redundancy, and enhance data integrity. This organization is essential for reliable time series analysis, as it facilitates efficient querying and data management.

---

2. Inserting Data:

- **Steps**:
  - Insert the cleaned and pre-processed data into the newly created tables, ensuring the structure aligns with the defined schema.
- **Justification**:
  - Storing pre-processed data in the database ensures that the data is readily available for model training. This reduces preprocessing time and potential errors during model development, allowing for a smoother and more efficient workflow.

Joining and Consolidating Data

1. Joining Tables:

- **Steps**:
  - Perform SQL joins to combine data from safe_driving, breda_road, precipitation, temperature, and greenery tables.
  - Merge these tables into a single consolidated DataFrame for further analysis.
- **Justification**:
  - Combines various data sources into a single, comprehensive dataset. This is necessary for capturing the full context of each time series event, enabling models to learn from a rich set of features that include temporal, spatial, and contextual information.

---

2. Handling Joined Data:

- **Steps**:
  - Convert columns to appropriate data types.
  - Handle missing values post-join to ensure data integrity.
- **Justification**:
  - Ensures data consistency and completeness, critical for maintaining the quality and reliability of time series inputs. Properly formatted and clean data is essential for effective model training and accurate predictions.

---

3. Inserting Joined Data:

- **Steps**:
  - Insert the consolidated DataFrame into a new table (pre_normalised) in the database.
- **Justification**:
  - Consolidated data in a single table simplifies access and analysis, making it easier to feed into time series models. This step ensures that all relevant data is available in a unified format, ready for further processing and analysis.

Creating and Utilizing Views

1. Creating Database Views:

- **Steps**:
  - Define and create SQL views in the database to provide simplified access to the consolidated data.
- **Justification**:
  - Provides a simplified and consistent interface for accessing the consolidated data. Views facilitate easier data exploration and analysis by presenting a unified and organized perspective of the data.

---

2. Analysing Data:

- **Steps**:
  - Analyse the distribution of the output class (incident severity) to understand class imbalance.
  - Visualize the class distribution using bar plots.
- **Justification**:
  - Understanding class distributions helps in addressing class imbalance issues, ensuring that time series models are trained on balanced and representative datasets. Visualizing the data aids in identifying patterns and anomalies that may need further investigation.

Data Splitting and Class Imbalance Handling

1. Splitting Data:

- **Steps**:
  - Split the normalized data into training, validation, and test sets for machine learning model development.
  - Ensure the splits are balanced and representative.
- **Justification**:
  - Properly splitting the data into training, validation, and test sets ensures that the time series models are trained, tuned, and evaluated on different data. This enhances their generalizability and robustness, ensuring reliable performance on unseen data.

---

2. Handling Class Imbalance:

- **Steps**:
  - Use techniques such as random sampling with replacement and class weighting during model training to address class imbalance.
- **Justification**:
  - Techniques like random sampling and class weighting help in addressing class imbalance. This is crucial for preventing model bias and ensuring that the model performs well across all classes, which is important for making accurate predictions in real-world scenarios.

Saving Preprocessing Steps

1. Custom Data Preprocessor:

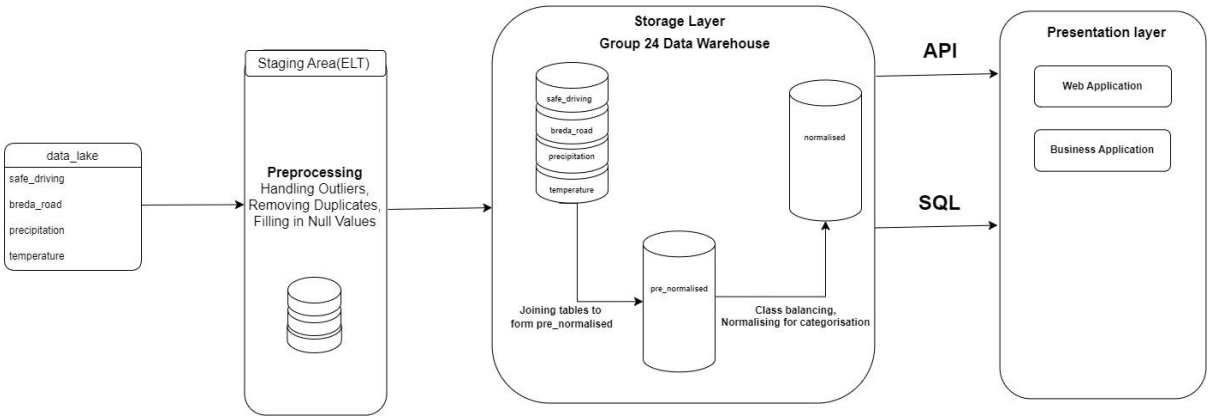
- **Steps**:
  - Implement a custom transformer for data cleaning and feature engineering, integrated into a scikit-learn pipeline.
  - Ensure preprocessing steps are reusable for future data integration.
- **Justification**:
  - A reusable preprocessing pipeline ensures that future data can be processed consistently. This maintains the integrity of the time series data and ensures that the model's performance remains stable, facilitating continuous improvement and scalability.

---

2. Pipeline Integration:

- **Steps**:
  - Define a pipeline that incorporates the custom preprocessing steps.
  - Apply the pipeline to the fetched and joined data to maintain consistency.
- **Justification**:
  - Integrating preprocessing steps into a pipeline automates and standardizes the data preparation process. This reduces errors and ensures that the data fed into deep learning models is always properly cleaned and formatted, leading to more reliable and accurate models.

SQL Data Warehouse Diagram



Conclusion

By justifying each cleansing and preprocessing step with the needs of time series deep learning models, we ensure that the data is prepared in a way that enhances model accuracy, reliability, and performance.

This comprehensive approach addresses data quality and consistency challenges, providing a robust foundation for advanced MLP analysis and machine learning model development.