Preprocessing Document

Overview

This document explains the preprocessing steps applied in two predictive models:

1. **Service Completion Time Model** – Predicts how long it takes to complete a service appointment.

2. **Staffing Prediction Model** – Predicts the required number of employees on duty based on historical patterns.

Both models involve **data cleaning, feature engineering, encoding, scaling, and preparation** for machine learning training.

1. Service Completion Time Model

1.1 Data Sets

- **bookings_train.csv**
    - Includes appointment details such as booking date, appointment date, appointment time, check-in, and check-out times.
- **tasks.csv**
    - Contains task-specific information (`task_id`, `section_id`).
- **staffing_train.csv**
    - Contains staffing records (`date`, `employees_on_duty`, `total_task_time_minutes`).

1.2 Data Cleaning & Integration

- **Merge Operations**:
    - `booking + task` → merged using `task_id`.
    - `booking + staff` → merged using `appointment_date` and `section_id`.
- **Target Variable**:
    - `completion_time_minutes = check_out_time − check_in_time` (converted to minutes).
    - Records with **missing values** or **non-positive completion times** were removed.
- **Missing Values**:
    - `staff_load_ratio` and `employees_on_duty` were imputed with their **mean values**.

1.3 Feature Engineering

- **Temporal Features**:
    - `appointment_hour`: Extracted from appointment time (HH).
    - `appointment_weekday`: Day of week (0=Monday, 6=Sunday).
    - `month`: Month of appointment.
- **Categorical Encoding**:
    - `task_id` → `task_id_encoded` (LabelEncoder).
    - `section_id` → `section_id_encoded` (LabelEncoder).
- **Staff Features**:
    - `staff_load_ratio = total_task_time_minutes / employees_on_duty`.

    - `employees_on_duty`: Filled with mean if missing.

1.4 Feature Selection

Final input features for the model:

- `appointment_hour`
- `appointment_weekday`
- `month`
- `task_id_encoded`
- `section_id_encoded`
- `staff_load_ratio`
- `employees_on_duty`

Target variable:

- `completion_time_minutes`

1.5 Scaling

- **StandardScaler** applied to all numerical input features to normalize values before model training.

2. Staff Prediction Model

2.1 Data Sources

- **staffing_train.csv**
    - Contains daily staffing levels per section (`date`, `section_id`, `employees_on_duty`).
- **tasks.csv**
    - Used for section reference (if needed).

2.2 Data Cleaning

- **Date Conversion**:
    - `date` column converted to `datetime` object.
- **Missing Values**:
    - No explicit missing values found; if present, they would be handled with mean/mode imputation.

2.3 Feature Engineering

- **Temporal Features**:
    - `month`: Extracted from date.
    - `weekday`: Extracted from date (0=Monday, 6=Sunday).
- **Categorical Encoding**:
    - `section_id` → `section_id_encoded` (LabelEncoder).

2.4 Feature Selection

Final input features:

- `month`
- `weekday`
- `section_id_encoded`

Target variable:

- `employees_on_duty`

2.5 Scaling

- **StandardScaler** applied to normalize feature values.

3. Model Training & Evaluation

3.1 Common Steps

- **Train/Test Split**:
    - Data split into 80% training, 20% testing.
- **Model Used**:
    - XGBoost Regressor (`n_estimators=100, max_depth=5, random_state=42`).

3.2 Metrics

- **Service Completion Time Model**:
    - Metric: Mean Absolute Error (MAE) in minutes.
- **Staffing Prediction Model**:
    - Metric: Mean Absolute Error (MAE) in number of employees.

3.3 Persistence

- Models and encoders saved using **joblib** for future inference:
    - XGBoost model (`.pkl`)
    - Scaler (`.pkl`)
    - LabelEncoders (`.pkl`)

4. Rationale for Preprocessing Steps

- **Data Cleaning**: Removes invalid or incomplete records to avoid bias.
- **Feature Engineering**: Creates meaningful predictors (time-based + operational).
- **Encoding**: Converts categorical variables into numeric form compatible with ML models.
- **Scaling**: Normalizes features to improve XGBoost performance.
- **Model Saving**: Ensures reproducibility and reusability of trained models.