DPDzero Data Ops Assignment - Documentation

Overview

This task is a modular Python data pipeline designed for an organization that runs daily call campaigns for loan collections. It processes campaign data from three CSV files covering call logs, agent rosters, and disposition summaries to generate a daily performance report per agent. It includes data validation, merging, metric computation, and outputs:

- A detailed CSV report of agent performance
- A Slack-style summary message

The code uses object-oriented design (DPDClass) and supports command-line execution with logging.

Given datasets:

- **call_logs.csv**: Contains details of individual call attempts.
- **agent_roster.csv**: Contains metadata about the agents (name, office location, etc.).
- **disposition_summary.csv**: Contains information about agent logins for a given date.

1. Data Ingestion & Validation

Implemented via DPDClass.ingest_validate_data(data_path, primary_key_cols).

Parameters:

- data_path: path to csv input file.
- primary_key_cols: Columns that should not be null or no missing values.

Function:

- Reads CSVs using pandas.
- Ensures that required columns are present or not.
- Drops duplicates.
- Handles missing values with isna() function and if null values found it will be Flagged.
- Formats date columns to consistent YYYY-MM-DD format.
- Logs each step with helpful context.

Returns: It returns ingested, cleaned, and validated dataframe to the assigned variable.

2. Join Logic:

Implemented via DPDClass.merge_dataframes(df1, df2, on_cols, mtype).

Parameters:

- df1, df2: Two dataframes which to be joined/merged.
- on_cols: List of columns that were used for the merging dataframes.
- mtype: it represents the merge type such as left, right, inner joins.

Function:

- Merging the two data frames using the given join type and columns.
- Ensuring there is no data loss while merging.
- Log warns if there is any agent in call_logs are not found in roster.

Returns: A merged dataframe.

3. Feature Engineering

Implemented via DPDClass.feature_engg(df).

Parameters:

- df: Alone dataframe that will be used to calculate the metrics.

Function:

- For each agent on each date, compute:
- Total Calls: Calculates the total number of calls made by each agent.
- Unique Loans Contacted: Counts the number of unique loans contacted by each agent.
- Connect Rate: Computes the percentage of successful calls (marked as completed).
- Average Call Duration: Determines the average duration of calls in minutes.
- Presence: Indicates whether the agent was logged in (present) for that day.
- The data is then grouped by agent_id, call_date, and other relevant features to compute these metrics.

Returns: A computed features and metrices dataframe.

4. Output

Implemented via slack_summary_message(df, date).

Parameters:

- df: A dataframe to get a details for the slack style message.
- date: The date when the summary message is to be generated.

Function:

- Filters the performance summary based on the given date.
- Identifies the top-performing agent.
- Calculates and logs the average call duration and the total number of active agents.
- Formats the results into a given slack styled message format.

Returns: A Multi-line String of slack styled summary message.

DPDZero_Task file

Implemented via dpdtask(path1, path2, path3).

Parameters:

- Path1, path2, path3: The dataset file path which contains the csv file for the task.

Function:

- This is the main script that manages the entire data pipeline and accepts file paths as input parameters for processing.
- It creates a log file to trace every move of the pipeline and the logs/flags stored in a 'logs/pipeline.log' path
- An object of the **DPDClass** is created and represented by the variable **dpd**.
- Data ingestion, cleaning, validation, merging, and feature engineering are all performed within this script.
- Saves the agent performance summary DataFrame to the specified file path.
- Finally, generates and returns a Slack-style summary message as the output.

Returns: It Saves a CSV file named agent_performance_summary and returns a Slack-formatted summary message from the data pipeline.

To Execute this Pipeline

- Install requirement python libraries such as pandas, os, logging, pathlib, argparse using pip install.
- Ensures that the DPDClass.py and DPDZero_Task.py/DPDZero_Task.ipynb are in the same directory.
- Ensure the input CSV files are in the right directory.

In the command line use this snippet to run the pipeline:

- python DPDZero_Task.py
  --call_logs <call_logs.csv path>
  --agent_roster <agent_roster.csv path>
  --disposition_summary <disposition_summary.csv path>

The processed dataframe will be saved in the result/agent_performance_summary.csv path and the output was the slack-styled summary message.

Output:

Agent Summary for 2025-04-28
Top Performer: AgentFirst3 AgentLast3 (38% connect rate)
Total Active Agents: 17
Average Duration: 0.12

Submitted by,
Vigneshwaran S
Email: