Mid-projects examples

---

2. Weather Data ETL

Requirements:

- **Dataset:** Global Weather Data
- **Tasks:**
- Extract, transform, and load data into a SQLite database.
- Analyze temperature trends, rainfall patterns, and seasonal variations.
- Generate a CSV report and a bar chart for average monthly temperatures.

```
Bonuse:
•   Provide CLI options to view temperature trends or rainfal
l patterns by city and time.
•   Store the cleaned weather data in a database for regional
or temporal queries.
•   Offer dynamic filtering of extreme weather events based o
n user-defined thresholds.
```

---

3. Movie Rating Dashboard

Requirements:

- **Dataset:** IMDb Movie Ratings
- **Tasks:**
- Clean and transform data to include genres, release year, and user ratings.
- Create a CLI using argparse for sorting/filtering movies by rating, genre, or year.
- Output insights as visualizations and summaries.

```
Bonuse:
•   Add CLI commands for sorting/filtering movies by genre, r
ating, or year.
•   Load the dataset into a database for interactive explorat
ion of movies.
•   Enable dynamic analysis based on user-selected genres or
minimum rating values.
```

---

4. Real Estate Price Analysis

Requirements:

- **Dataset:** Ames Housing Dataset
- **Tasks:**
- Analyze house price trends based on location, size, and year of sale.
- Create a dynamic analysis tool that allows users to input location and view related insights.
- Store cleaned data in a SQLite database.

```
Bonuse:
•   Use a CLI to sort properties by price, location, or size
dynamically.
•   Save the processed real estate data into a database for l
ocation-based queries.
•   Provide filters for specific property types, years, or pr
ice ranges.
```

---

5. COVID-19 Vaccination Data Analysis

Requirements:

- **Dataset:** COVID-19 Vaccination Dataset
- **Tasks:**

- Perform ETL to extract vaccination rates across different regions.
- Visualize vaccination progress over time using line plots.
- Output the cleaned dataset to a JSON file.

```
Bonuse:
•   Create CLI commands for viewing vaccination progress by r
egion or country.
•   Store vaccination data in a database for queries on speci
fic time frames or regions.
•   Enable filtering by population percentage vaccinated or d
ate ranges dynamically.
```

---

6. Employee Productivity Tracker

Requirements:

- **Dataset:** Employee Productivity
- **Tasks:**
- Transform data to calculate average productivity scores and detect outliers.
- Build a CLI for users to retrieve employee-specific performance reports.
- Save results to CSV and SQL formats.

```
Bonuse:
•   Create CLI commands for viewing vaccination progress by r
egion or country.
•   Store vaccination data in a database for queries on speci
fic time frames or regions.
•   Enable filtering by population percentage vaccinated or d
ate ranges dynamically.
```

---

8. Air Quality Monitoring

Requirements:

- **Dataset:** Air Quality Data
- **Tasks:**
- Analyze and clean the dataset for pollutants like PM2.5 and NO2.
- Create a dashboard to show pollutant levels across cities.
- Output the data to both JSON and CSV formats.

```
Bonuse:
•   Use a CLI to explore pollutant levels dynamically by cit
y, date, or type.
•   Store air quality data in a database for advanced query c
apabilities.
•   Allow users to filter for locations with pollutant levels
exceeding safety thresholds.
```

---

9. Food Delivery Performance Analysis

Requirements:

- **Dataset:** Food Delivery Data
- **Tasks:**
- Analyze data to determine peak hours, average delivery time, and top-rated dishes.
- Build a CLI tool for filtering data by restaurant or delivery zone.
- Store the cleaned dataset in a database.

```
Bonuse:
•   Enable CLI commands to sort delivery times or ratings by
zone, restaurant, or day dynamically.
•   Load delivery data into a database for performance metric
s queries.
```

```
•   Allow filtering of high-performing restaurants or peak de
livery hours.
```

---

10. Global Renewable Energy Trends

Requirements:

- **Dataset:** Renewable Energy Production
- **Tasks:**
- Analyze the contribution of different energy sources (solar, wind, hydro) over time.
- Use Pandas for data manipulation and Plotly for visualizations.
- Save processed data to a JSON file and generate a textual summary.

```
Bonuse:
•   Provide CLI options for sorting energy sources by contrib
ution percentage dynamically.
•   Store energy production data in a database for temporal o
r regional queries.
•   Add dynamic filters to focus on renewable energy types or
production ranges.
```

Bonuses (Applied to All Projects):

1. **Use** argparse **to Create a CLI:**

- Implement a command-line interface to enable dynamic interaction with the dataset.
- Allow users to select specific analysis outputs through different commands.

2. **Store the Dataset in a Database:**

- Preload the cleaned and transformed dataset into a database (e.g., SQLite or PostgreSQL).
- Provide query functionality to retrieve specific data subsets.

3. **Create Dynamic Analysis:**

• Add features to sort and filter the dataset based on user input criteria, such as dates, categories, or numerical ranges.

4. **Document the Process:**

• Write detailed documentation explaining the original dataset, transformation steps, and the value generated by the analysis.

• Include code explanations and usage guides for the implemented CLI and database integration.