Team Information

- **Team Members:** Pavan Sai Porapu

---

1. Selected API & Data Collection

- **API Name:** WeatherAPI
- **API Endpoint Used:** "https://air-quality.p.rapidapi.com/history/airquality"
- **Link to API Overview :** https://rapidapi.com/weatherbit/api/air-quality
- **Type of Data Retrieved:** JSON format data.
- **Frequency of Data Fetching:** Data of Air Quality over past three days of metropolitan cities in India.
- **Challenges in Data Retrieval & Solutions:**
    - API rate limits(25 request per day) and limited access to the api endpoint on rapidapi platform. I have managed this challenge by using multiple accounts to maximize the limi

2. Data Exploration & Understanding

- **Overview of Retrieved Data:** The retrieved data contains the following features,

> `lat` : Latitude (Degrees). `lon` : Longitude (Degrees). `timezone` : Local IANA Timezone. `city_name` : Nearest city name. `country_code` : Country abbreviation. `state_code` : State abbreviation/code. [{
>
> - `timestamp_local` : Timestamp at local time.
> - `timestamp_utc` : Timestamp at UTC time.
> - `ts` : Unix Timestamp at UTC time.
> - `aqi` : Air Quality Index [US - EPA standard 0 - +500]
> - `o3` : Concentration of surface O3 (µg/m³)
> - `so2` : Concentration of surface SO2 (µg/m³)
> - `no2` : Concentration of surface NO2 (µg/m³)
> - `co` : Concentration of carbon monoxide (µg/m³)
> - `pm25` : Concentration of particulate matter < 2.5 microns (µg/m³)
> - `pm10` : Concentration of particulate matter < 10 microns (µg/m³)
>
> }, ...]

3. Data Cleaning & Preprocessing

Handling Missing or Incomplete Data

- **Missing Values Found:** NO, there were no missing values found in the data.
- **Handling Strategy:**
    - If there are any missing values found I would have handled them by any one of the following strategies.
        - ❖ Mean/Median/Mode imputation
        - ❖ K Nearest Neighbours imputation

Data Type Transformation

- Applied OneHot encoding on Location(city_name) feature which a string datatype.
- Normalization continuous features using Min-Max Scaling or other strategy is needed to be applied based on the model being trained.

Feature Engineering

- **New Features Created:**
    - Created a new feature hour from datetime feature.
    - Added city_name column.
    - Removed unnecessary columns such as Timestamp, datetime, Unix time

```
data.head()
```

✓ 0.0s

	aqi	co	no2	o3	pm10	pm25	so2	city_name	hour
0	128	215.4	25.7	92.1	99.7	45.67	29.2	Ahmedabad	11
1	211	614.5	81.5	37.8	209.0	101.00	39.0	Ahmedabad	10
2	208	534.0	71.0	34.2	201.0	99.00	37.5	Ahmedabad	09
3	206	91.0	40.2	42.0	141.8	97.91	25.5	Ahmedabad	08
4	227	121.0	56.0	61.1	170.8	113.25	45.3	Ahmedabad	07

Figure 1Data after feature Engineering

4. Data Storage & Pipeline

- **Storage:**
    - Raw API data stored in comma separated files(.csv) format.
    - The stored data in csv format is converted into pandas DataFrames for processing.
- **Data Pipeline:**
    - **Extract:** Fetch data from WeatherAPI.
    - **Transform:** Clean, preprocess, and engineer features.
    - **Load:** Store processed data into the another CSV file.

5. Data Integrity & Quality Checks

- **Quality Checks Implemented:**
    - Checked for duplicate records and removed them.
    - Ensured AQI values remained within a valid range.
- **Outliers Detection & Handling:**
    - Used Z-score method to detect extreme values in pollutant concentrations.
    - Capped extreme values beyond 3 standard deviations.
- **Outliers Detection & Handling:** No duplicated columns found.

6. Preprocessed Data Structure & Readiness for Modeling

- **Final Dataset Overview:**
    - Well-structured dataset with numerical and categorical features properly transformed.
    - Missing values handled, and new features added.
- **Data Augmentation:** No data augmentation was applied.

✓ 0.0s Python

	aqi	co	no2	o3	pm10	pm25	so2	hour	city_name_Ahmedabad	city_name_Bengaluru	city_name_Chennai	city_name_Delhi	city_name_Gajuwaka	city_name_Hy
0	128	215.4	25.7	92.1	99.7	45.67	29.2	11	1.0	0.0	0.0	0.0	0.0	
1	211	614.5	81.5	37.8	209.0	101.00	39.0	10	1.0	0.0	0.0	0.0	0.0	
2	208	534.0	71.0	34.2	201.0	99.00	37.5	9	1.0	0.0	0.0	0.0	0.0	
3	206	91.0	40.2	42.0	141.8	97.91	25.5	8	1.0	0.0	0.0	0.0	0.0	
4	227	121.0	56.0	61.1	170.8	113.25	45.3	7	1.0	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
643	119	187.0	127.5	7.8	72.5	42.50	19.0	14	0.0	0.0	0.0	0.0	1.0	
644	104	302.0	127.0	5.8	78.0	37.00	18.0	13	0.0	0.0	0.0	0.0	1.0	
645	153	0.0	67.0	12.9	73.0	55.00	17.0	12	0.0	0.0	0.0	0.0	1.0	
646	121	302.0	133.0	6.6	92.0	43.00	17.0	11	0.0	0.0	0.0	0.0	1.0	
647	161	216.5	99.0	13.5	120.5	64.00	19.5	10	0.0	0.0	0.0	0.0	1.0	

648 rows × 17 columns

Figure 2Data after cleaning and OneHot encoding

7. Challenges & Solutions

- **Challenge:** API rate limitations.
  - **Solution:** used multiple accounts to get maximum request rate possible.
- **Challenge:** Handling inconsistent columns and unnecessary.
  - **Solution:** Used necessary techniques to resolve issue using documentation and AI tools.

8. Datasets & Notebooks

- **Github Repo:** https://github.com/SaiPawan01/ML-Hackathon-Inventun2K25.git
- **Directory structure :**
  - ❖ Data_retrieve directory : contains data extraction notebooks from api.
  - ❖ Dataset_preparation : contains data cleaning, feature extraction notebooks.
- **References:**
  - WeatherAPI: https://rapidapi.com/weatherbit/api/air-quality
  - Pandas: https://pandas.pydata.org/
  - Scikit-learn: https://scikit-learn.org/stable/user_guide.html