REPORT:

In this document, I will mention the choices that I make for the project and the problems that I faced.

FEATURE PIPELINE:

Data Fetching:

I used two APIs for fetching relevant data.

- Open Weather : for AQI data.
- Open Meteo: for weather data.

Historical data that I fetched was from january01,2023, 00: 00: 00 PST.

Data Pre-processing:

There were 16 gaps in AQI data. 15 gaps were 24-28 hours long. One gap was 120 hours long.

- I used Linear Interpolation for 24-48 hours gap.
- 120 hours gap was at the end of april 2025. I used previous data to train an LSTM model, and then predicted that 5 day gap.

Engineering Features:

I used three main types of features for training:

AQI features:

- Raw concentrations of six pollutants (pm2.5, pm10, no2, so2, co, o3)
- Lag features on the raw concentrations of pollutants (lag_1, lag_3, lag_6) and AQI value (lag_1, lag_3, lag_6, lag_12).
- Rolling features on raw concentrations of pollutants and AQI value (rolling_6, rolling_12, rolling_24) .

Weather features:

- Temperature, surface pressure, relative humidity, shortwave radiation, wind direction_sin, wind_direction_cos, wind speed, precipitation
- Lag features on Temperature, relative humidity, precipitation and surface pressure. (lag_1, lag_3, lag_6)
- Rolling features on temperature, relative humidity, precipitation, surface pressure. (rolling_3, rolling_6, rolling_12)

Time features:

- Sin and cos encoded values of hour of the day, day of the week, month of the year.

Hopsworks Feature-Store Schema:

Feature Groups:

Above mentioned features were arranged in following feature groups:

- Aqi_data (raw concentrations of six pollutants, aqi value, aqi category (1-5))
- Weather_data (raw values of temperature, surface pressure, relative humidity, precipitation, shortwave radiation, wind direction cos, wind direction sin, wind speed)
- Aqi_lag (lag values of pollutants and aqi)
- Weather_lag (lag values of weather features)
- Aqi_rolling (rolling values on pollutants and aqi value)
- Weather_rolling (rolling values on weather features)
- Time_features (sin and cos encoded values of time features)

All these feature groups have primary keys as datetime_unix (unix timestamp).

Feature Views:

Three main feature views for three different models:

- lstm_features : aqi_data (all features except aqi and aqi value) + weather_data (all features) + time_features (all features)
- Ensemble_features: all features from all feature groups
- Classifier_features: aqi value not included.

Problems that I faced:

- **Timestamp in utc:** Open Weather returns timestamp in unix_utc format. I didn't noticed it at first, and it took some time and a lot of frustration to realize this.
- **Just AQI category**: Open Weather returned aqi as category and not the value. So, I calculated the AQI value by myself. I used the US-EPA method. But I didn't find any proper documentation for the method for calculating this. There were some confusions. And contradicting rules for calculating aqi. I choosed the following options.
    - I used concentration units of $\mu g/m^3$ for pm2.5 and pm10, ppb for so2, no2, o3 and ppm for co.
    - I used 24 hours mean for values of pm 2.5 and pm10, 8 hour rolling mean for 03 and co, 1 hour mean for so2 and no2.
    - For pm2.5, in case when the current hour was less than 18, I used the Now Cast algorithm for calculating mean from last 12 hours.

TRAINING PIPELINE:

Models:

Three models as following:

LSTM:

Raw aqi + weather + time data for training.

Predicting just the concentrations of pollutants. (sequence for 72 hours predicted)
Lstm_features feature view was used for reading data from hopsworks for this model.

XGBOOST
Lag and rolling features, time features along with raw data used for training.
Aqi value was predicted.
Ensemble_features feature view, for reading data from hopsworks for this model.

SVM
Raw values and selective lag and rolling features for training the model.
Predicting just the aqi category.
Classifier_features feature view, for reading data from hopsworks for this model.

For LSTM, incremental training was used to train the model on daily hourly data.
For XGBOOST and SVM, training was done from scratch for daily hourly data.

Problems that I faced:

"THE LSTM": I had prior experience of training classifiers and other traditional ML models, but no adequate prior experience of training deep learning models especially RNNs. So, training LSTM was challenging. I made changes to the code so many times. Many new concepts that I learned. The concept of sequences and the shape of matrix, this took a while to settle. First time, I trained the model, it gave me very very horrible performance. Values in tens were predicted to be in hundreds.

CI/CD PIPELINE:

Two separate workflows for both pipelines.
Hourly event alert for feature pipeline.
Daily event alert for training pipeline.

Problems that I faced:

When I was working on my CI/CD pipeline, hopsworks updated it's backend (saas), and my system degraded. Starting from the version mismatch of library.. I continued to face some very weird problems. I contacted them on their slack channel, and they said they are trying to investigate such kind of problems and it appear to them that it's related to the platform update.