Air Flow Rate and Energy Forecasting Using Sensor and Time-Series Data

April 6, 2025

1 Dataset Exploration

1.1 Automotive OBD-II Dataset: Air Flow Rate Prediction

The Automotive OBD-II dataset comprises **2,693,824 time-series entries** collected via an onboard diagnostic tool from a passenger vehicle during real-world driving. The objective is to predict the **air flow rate** using various engine and sensor parameters. The dataset includes:

- **Engine Metrics:** RPM, throttle position, air flow rate
- **Sensor Data:** Intake/ambient/coolant temperature, pressure
- **Driver Input:** Accelerator pedal positions D and E
- **Time Features:** Month, day of week, hour

Target variable: Air Flow Rate from Mass Flow Sensor [g/s].
Dataset link: radar.kit.edu
Target variable: **Air Flow Rate from Mass Flow Sensor [g/s]**.

1.2 PJME Energy Forecasting (Dataset2)

The dataset contains hourly power consumption (MW) from the PJME regionDataset 2, showing strong seasonal trends influenced by time of day and year.

Key preprocessing steps:

- Parsing and setting the `Datetime` column as index.
- Handling missing values via resampling and time-based interpolation.
- Filtering the data from 2015 onwards.
- Adding `Year`, `Month`, and `Weekday` features.

2 Feature Engineering and Selection

2.1 Dataset1

Due to the large size, a 10% random subset was used.

Correlation Analysis (Figure 1) showed:

- Intake Manifold Absolute Pressure [kPa] (0.86)
- Engine RPM [RPM] (0.77)
- Vehicle Speed Sensor [km/h] (0.69)
- Month, Seconds Since Start

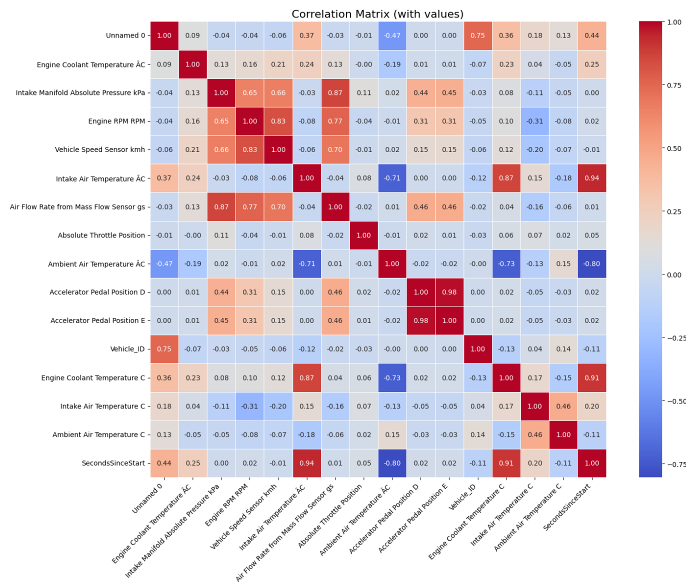

Figure 1: Correlation Matrix of Engine Sensor Features

2.2 Dataset2

- Extracted `Hour`, `Month`, and `Weekday` for seasonality.
- Added a 1-year (8760 hours) rolling average for trend detection (Figure 2).

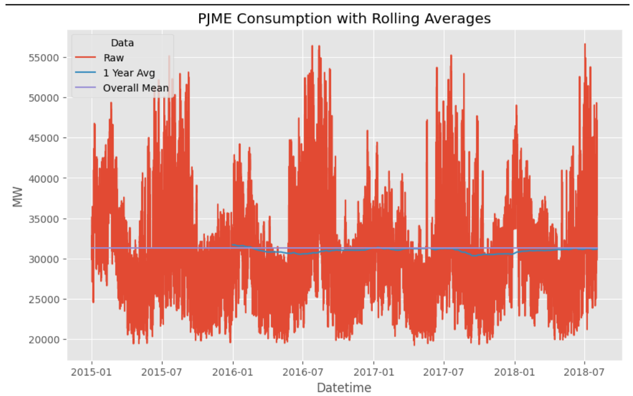

Figure 2: PJME Consumption with 1-Year Rolling Average and Overall Mean

3 Modeling Approach

3.1 Air Flow Prediction (Dataset1)

Input:

- Engine RPM, Intake Manifold Pressure, Vehicle Speed Sensor
- Temporal: Month, Seconds Since Start

Output: Air Flow Rate from Mass Flow Sensor [g/s]
Train-Test Split: 80% training, 20% testing **Preprocessing:** Standardization using `StandardScaler`

Air Flow Prediction Pseudocode

```
1. Load and subset the top features based on correlation
2. Apply StandardScaler to normalize features
3. Split data into training and testing sets (80-20)
4. Train three models:
   a. LinearRegression()
   b. RandomForestRegressor()
   c. XGBRegressor()
5. Predict on test data
6. Evaluate using R2 and RMSE
```

3.2 Time Series Forecasting (Dataset2)

Input:

- Historical hourly PJME electricity consumption (MW)
- Engineered features: Hour, Month, Weekday, Year

Output: Future electricity demand (MW)
Train-Test Split: Train on pre-2018, test on 2018 onwards

Time Series Forecasting Pseudocode

```
1. Load PJME data and filter from 2015 onward
2. Resample hourly data, fill missing values via time interpolation
3. Create features: Hour, Month, Weekday, Year
4. Split data: pre-2018 for training, 2018+ for testing
5. Fit models:
   a. ARIMA(p=3, d=0, q=6)
   b. SARIMA(p=1,d=0,q=1)(P=1,D=1,Q=1)[24]
6. Forecast future values
7. Compare predictions using RMSE (monthly + overall)
```

4 Evaluation Metrics

4.1 Regression Models (Dataset1)

Metrics:

- R^2 Score
- Root Mean Squared Error (RMSE)

Table 1: Regression Model Performance

Model	R^2 Score	RMSE
Linear Regression	0.8221	6.8729
Random Forest	**0.9347**	**4.1638**
XGBoost	0.9183	4.6576

4.2 Time Series Models (Dataset2)

Overall RMSE:

Model	Overall RMSE (MW)
ARIMA (3,0,6)	3257.23
SARIMA (1,0,1)(1,1,1)[24]	**3162.69**

Table 2: Overall RMSE Comparison

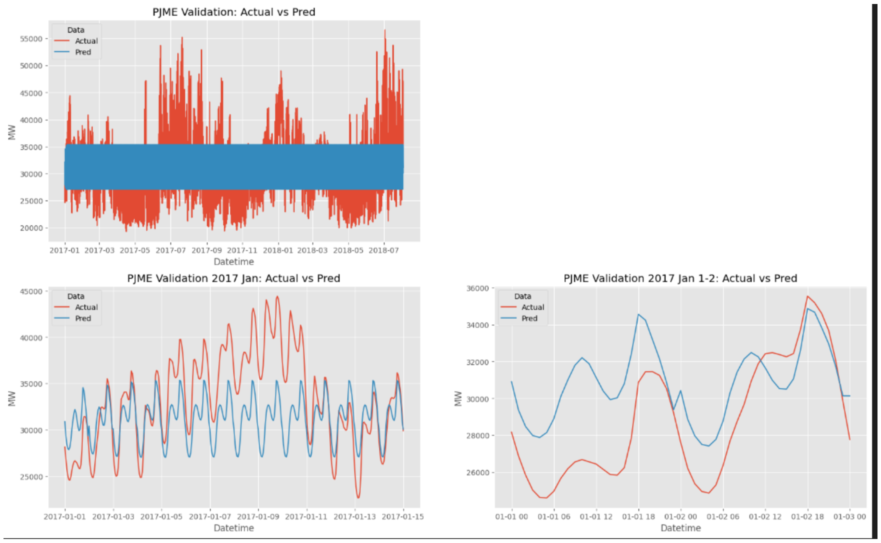

Figure 3: Prediction Sarima

5 Conclusions:

- The **Random Forest Regressor** provided the best accuracy for air flow rate prediction.
- The **SARIMA model** outperformed ARIMA in forecasting electricity demand by effectively capturing seasonality.

6 References:

7 References

– https://github.com/parasnaren/Predictive-Vehicle-Health
– https://raw.githubusercontent.com/Welkro/Vehicle-Fuel-Consumption-Analysis/refs/heads/main/dataset/fuel.csv
– https://www.generatorsource.com/Diesel_Fuel_Consumption.aspx
– https://github.com/hayatu4islam/Automotive_Diagnostics/tree/main/OBD-II-Dataset
– https://radar.kit.edu/radar/en/dataset/bCtGxdTklQlfQcAq#
– https://colab.research.google.com/drive/1D8EAj93VQxNbGvzg1vAhTzyPwHHMnJd5?usp=sharing
– https://colab.research.google.com/drive/1b2kv5u555ZMXAilZ6Vl2vGcMuKdC3SjN?usp=sharing
– https://colab.research.google.com/drive/1ooNutVU6lyhnY7URKFkzlFkxiBxxzPov?usp=sharing