Beijing PM2.5(Air Pollution) Prediction

Github Repo: https://github.com/pbtommy1209/TommyP1030-final-Report/tree/main
Baite Pang
12/14/2025
Brown Data Science Institute

Introduction

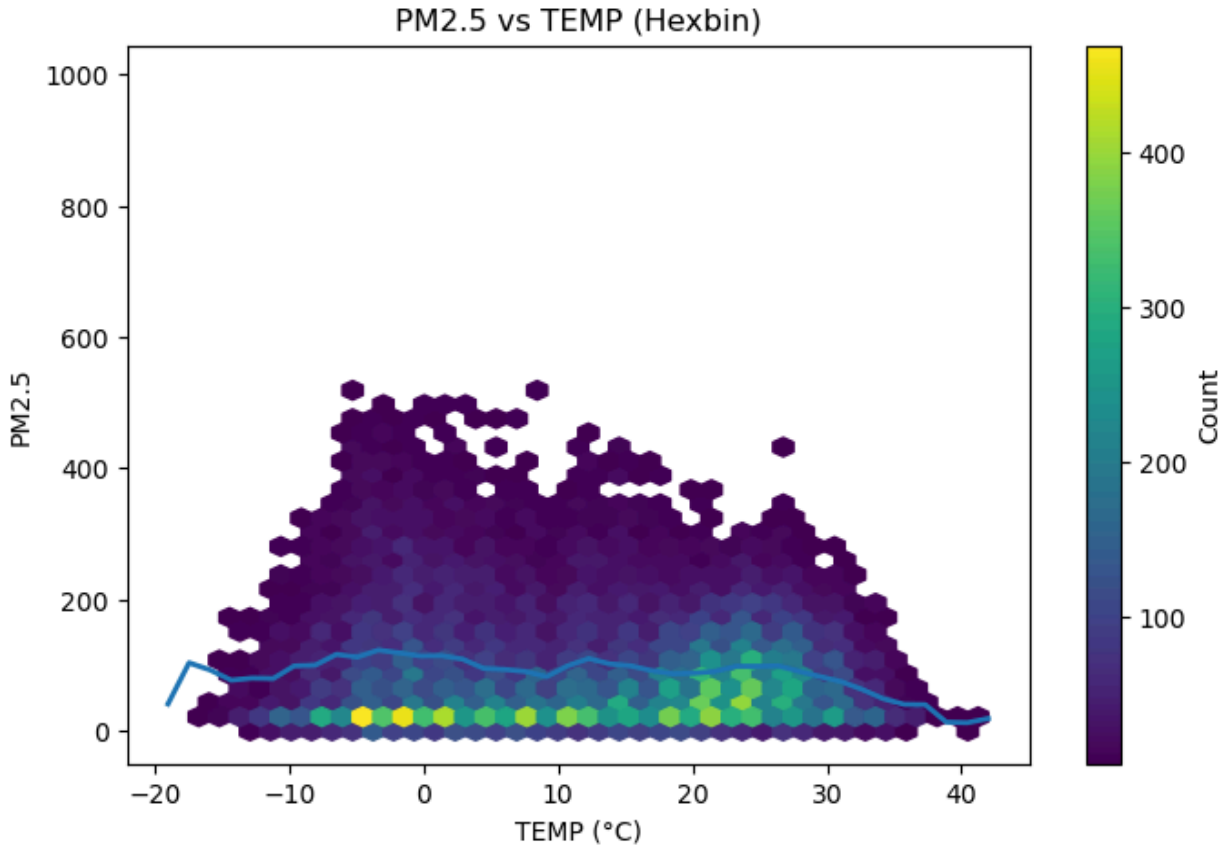
One of the most dangerous air pollutants is fine particulate matter (PM2.5), which has a diameter of less than 2.5 μm and is small enough to enter the bloodstream and deeply penetrate the lungs. Long-term PM2.5 exposure is linked to higher risks of stroke, respiratory and cardiovascular disorders. The ability to predict short-term concentrations is important for public health advisories and individual decision-making such as wearing masks, limiting outdoor activity, since hourly PM2.5 levels in Beijing frequently surpass these thresholds by significant margins.

The dataset in this project is the Beijing PM2.5 data set from the UCI Machine Learning Repository. It contains hourly measurements from 2010–2014 at the U.S. Embassy in Beijing (μg/m³), along with meteorological variables from Beijing Capital International Airport: dew point (DEWP), temperature (TEMP), pressure (PRES), cumulative wind speed (Iws), cumulative hours of snow (Is) and rain (Ir) and so on. The data form a multivariate time series with missing values flagged as “NA”, which makes it a natural testbed for time-series regression models that use meteorological and temporal information to predict short-term air-quality levels.

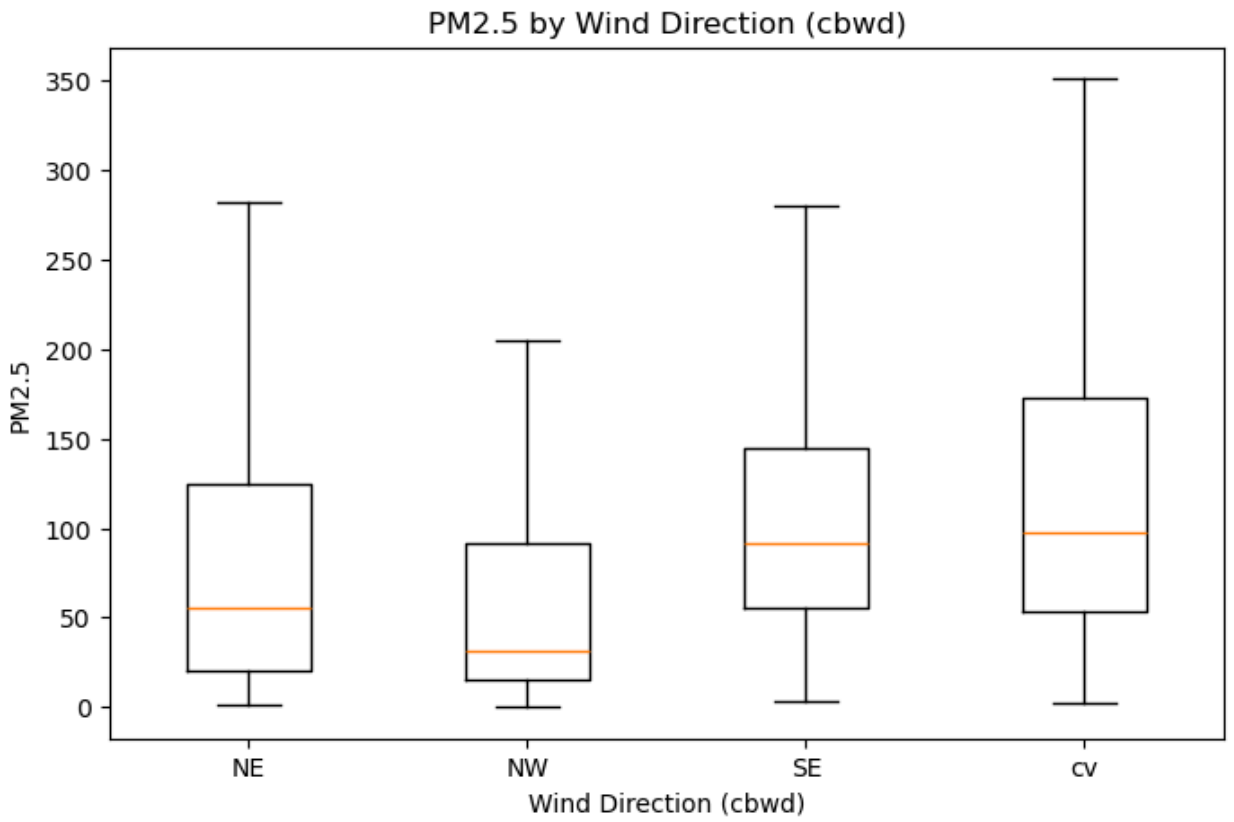
One closely related prior work is Zhang et al. (2023), who proposed a sparse attention–based Transformer network for forecasting hourly PM2.5 in Beijing using the same dataset as my project. They framed the task as a one-hour-ahead regression problem and compared their model against baselines such as SVR, and random forest. Their best model achieved an RMSE of about 19 μg/m³ on a 2014 test set which is similar to my linear model’s test RMSE of about 21.6 μg/m³ is broadly consistent with these results.

EDA

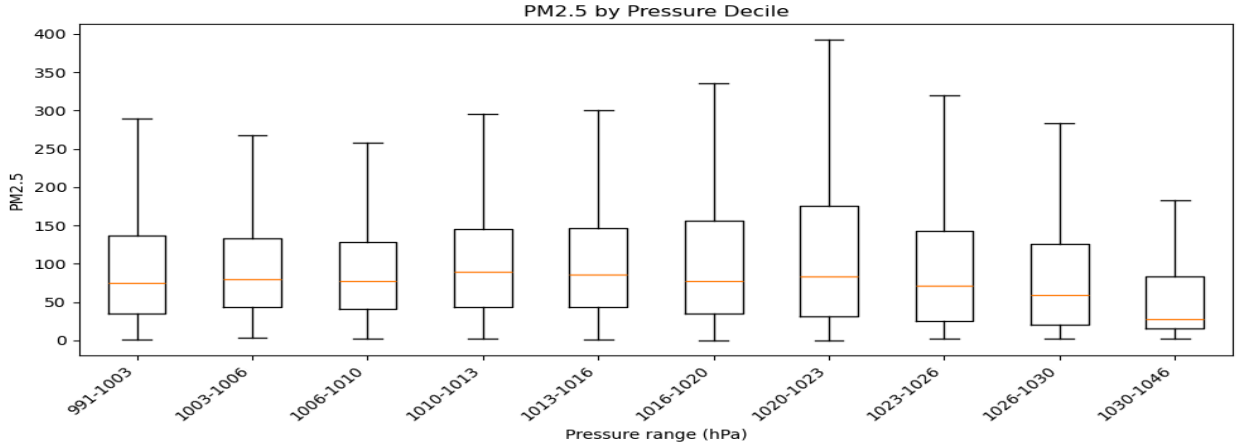
This first figure shows a hexbin plot of PM2.5 concentration against temperature. Pollution is highly variable at all temperatures, but the densest/highest PM2.5 values cluster around cool to mild temperatures rather than at the very hot or very cold extremes(from -20 to 30). This exactly matches the feature of seasonality.



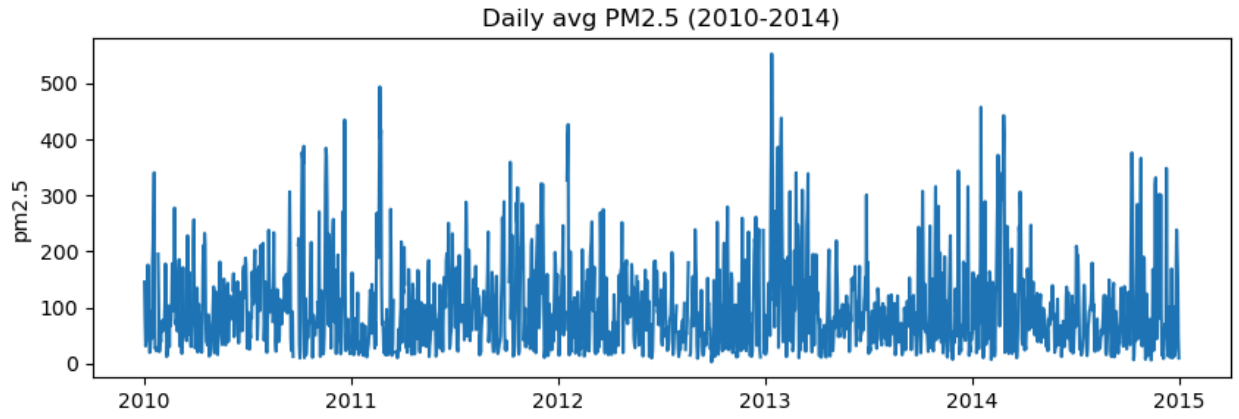
This figure shows boxplots of PM2.5 for each wind-direction category (cbwd), indicating that pollution levels are clearly direction-dependent. Hours with northwest (NW) winds have the lowest median PM2.5, while southeast (SE) and especially the “cv” (calm/variable) category show higher medians and more extreme upper values, suggesting that weak or variable winds are associated with the worst pollution episodes.



The third figure shows boxplots of PM2.5 across deciles of surface pressure. Median PM2.5 is generally higher at mid-range pressures (around 1013–1023 hPa) and lower in the very lowest and especially the highest pressure bins. This suggests that the most polluted hours tend to occur under moderate pressure conditions.



The fourth figure shows the daily average PM2.5 in Beijing from 2010–2014. The series has strong variability with frequent spikes above 300–400 μg/m³, indicating repeated severe pollution episodes. High values tend to cluster in certain periods (especially winters), suggesting pronounced seasonality and short-term persistence in PM2.5.



Methods

Feature construction

To capture temporal structure, I add hour-of-day (0–23) and encode it as two cyclical features, hour_sin and hour_cos = sin/cos(2π·hour/24), so that 23:00 and 0:00 are close in feature space. I then create lagged PM2.5 features: a 1-hour lag (pm2_1), a 24-hour lag (pm2_24), and a 24-hour rolling mean shifted by 1 hour (pm2_24_mean). These lags represent short-term persistence and daily history of pollution; rows without enough history to compute them are dropped.

To avoid heavy imputation, I use a reduced-features (pattern submodel) approach: I examine the missingness pattern across all candidate predictors and select the most common pattern. I then restrict the model to the subset of predictors that are fully observed under this pattern and to rows following that pattern. The final feature set includes:

- Meteorological variables: DEWP, TEMP, PRES, Iws, Is, Ir
- Temporal encodings: hour_sin, hour_cos
- Lagged pollution history: pm2_1, pm2_24, pm2_24_mean

Data splitting and cross-validation

All splits respect time order. After feature construction I sort the data chronologically and divide it into 80% “other” (earlier years) and 20% test (most recent months), without shuffling. The 20% test set is held out and never used during model selection.

Within the 80% block I apply 5-fold TimeSeriesSplit with expanding windows: in each fold the model is trained on an earlier window and validated on the immediately following window. Hyperparameters are tuned on these folds only, using the mean cross-validation RMSE as the selection criterion, and the chosen settings are then refit on the full 80% block and evaluated on the held-out test set.

Preprocess

For preprocessing, after sorting by time I dropped rows with missing PM2.5, then created time-based and lag features: hour of day (encoded as sine and cosine to capture daily periodicity) and three lagged PM2.5 features (1-hour lag, 24-hour lag, and a 24-hour rolling mean). These lags introduce missing values at the start of the series, so I removed those initial rows. For the

remaining columns I applied a reduced-features approach instead of imputation: after dropping rows without sufficient lag history, the remaining candidate features (DEWP, TEMP, PRES, Iws, Is, Ir, hour_sin, hour_cos, pm2_1, pm2_24, pm2_24_mean) had very little missingness, and the most common missingness pattern corresponded to having all 11 features observed. I therefore performed a complete-case analysis on rows with fully observed inputs and did not need to drop any additional features. Then standardized these predictors (zero mean, unit variance) using a StandardScaler fitted on the 80% "other" block; the same scaling was applied inside each fold of cross-validation and to the test set. All preprocessing steps were wrapped together with the model in a single scikit-learn Pipeline, ensuring that the scaler is always fit only on the training portion of each fold and preventing data leakage.

Choosing Metric

Each model is wrapped in a single scikit-learn Pipeline consisting of the preprocessing step (scaling) followed by the estimator. I evaluate performance using root mean squared error (RMSE), which matches the squared-error losses optimized by the regressors, is expressed in the same units as PM2.5 (μg/m³), and is the standard metric reported in previous Beijing PM2.5 studies.

Models and hyperparameter grids

I fit the following models:

Model	Best hyperparameters
SVR (RBF)	C = 300, ε = 1.0, RBF kernel (default γ)
Random Forest	max_depth = 30, max_features = 0.7, n_estimators = 300
Linear Regression	fit_intercept = True
Ridge Regression	alpha = 0.001
KNN Regressor	n_neighbors = 10, weights = 'distance'
Baseline (Mean)	no hyperparameters (mean predictor)

Uncertainty

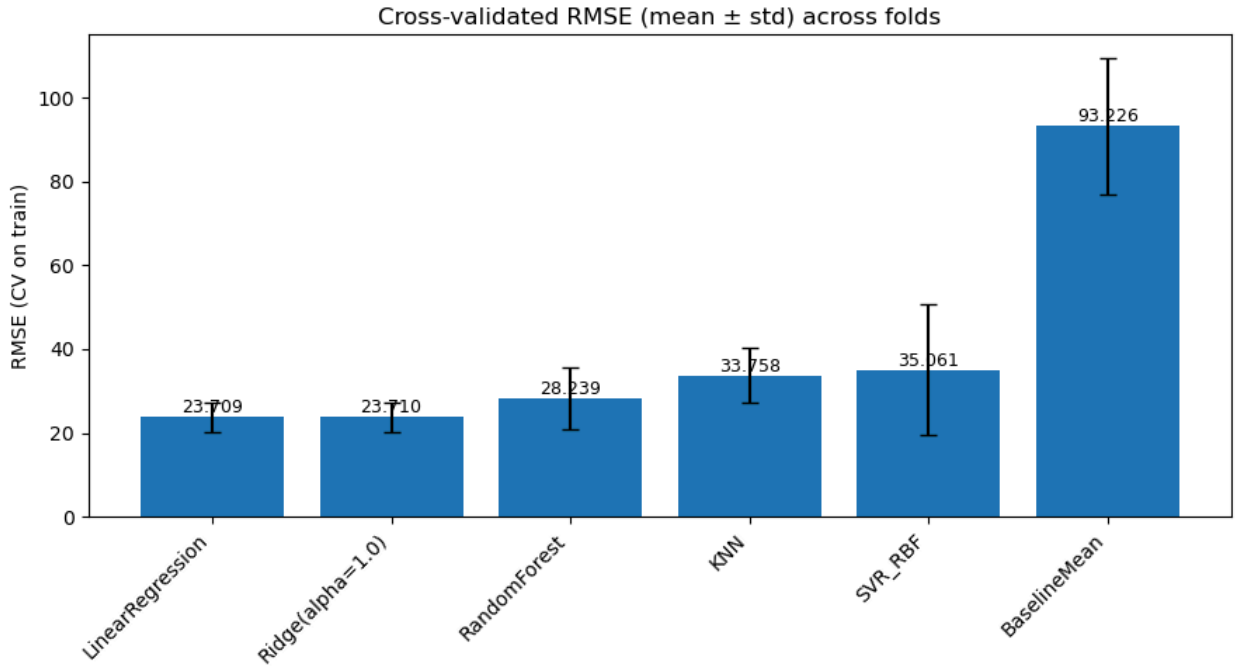
By using TimeSeriesSplit, I present two figures for every model on the 80% “other” block: the mean RMSE over the five folds and its standard deviation. The uncertainty resulting from the temporal splitting of the data is measured by the standard deviation of the cross-validation scores, which indicates how sensitive each model is to the precise time window used for validation. Because their performance is more dependent on the duration, models with comparable mean RMSE but significantly higher standard deviation are regarded as less dependable.

Each model’s final test performance on the held-out 20% test set is expressed as a single RMSE value. Similar to the differences between Linear, Ridge, KNN, and SVR, the fold-to-fold variability for the best models is only a few $\mu g/m^3$. This indicates that these top models are statistically very close and that apparent small gaps in RMSE should be interpreted cautiously.

Results

Baseline + Cross Validation(CV RMSE)

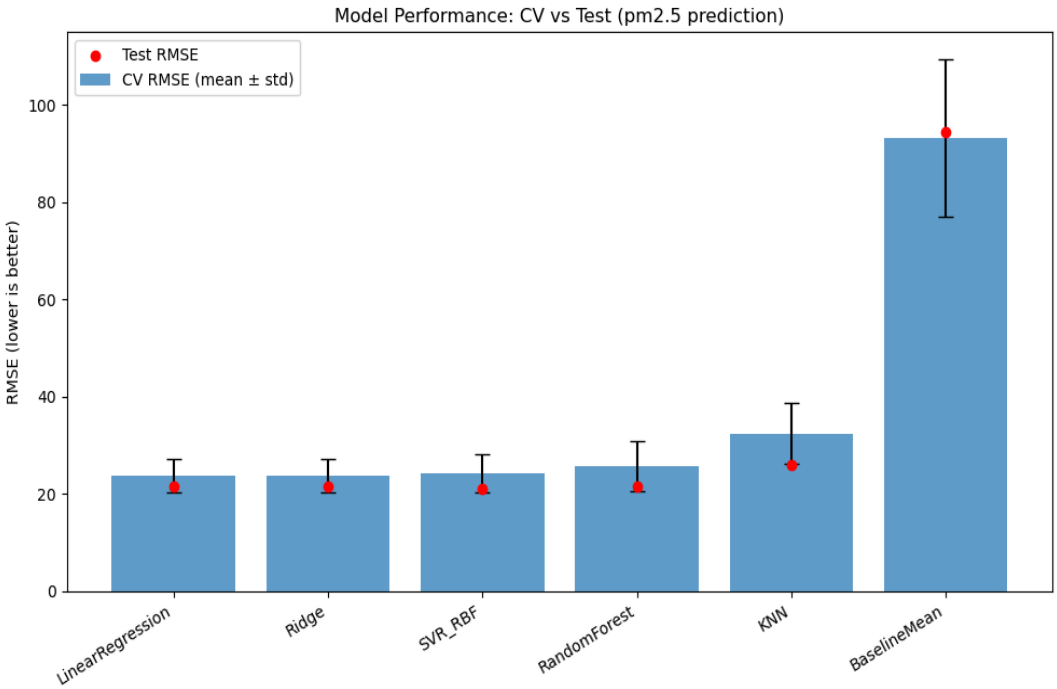
The graph below compares my models to a simple mean-predictor baseline. The baseline achieves a cross-validated RMSE of 93.226 ± 16.314 µg/m³, whereas all learned models reduce error by a large margin: Linear Regression obtains 23.709 ± 3.379 µg/m³, Ridge 23.710 ± 3.379, SVR 35.061 ± 15.489, Random Forest 28.239 ± 7.250, and KNN 33.758 ± 6.384.



Test Performance

The table and figure below show the CV RMSE using the best hyperparameters for each model. Both Linear and Ridge Regression achieve the lowest mean CV RMSE (23.72 ± 3.39 µg/m³), marginally better than SVR (24.18 ± 3.98) and Random Forest (25.66 ± 5.24), and obviously better than KNN (32.42 ± 6.35) and the baseline (93.22 ± 16.29) remain the same.

Based on this, I choose Linear Regression as my final model because it achieves the best CV RMSE and is much easier to understand than the other complex models. Its performance can be objectively estimated using the held-out test RMSE of 21.62 µg/m³. Plus, it can feed the raw features directly into a linear equation, and is computationally inexpensive and quick to train due to its straightforward, transparent structure.

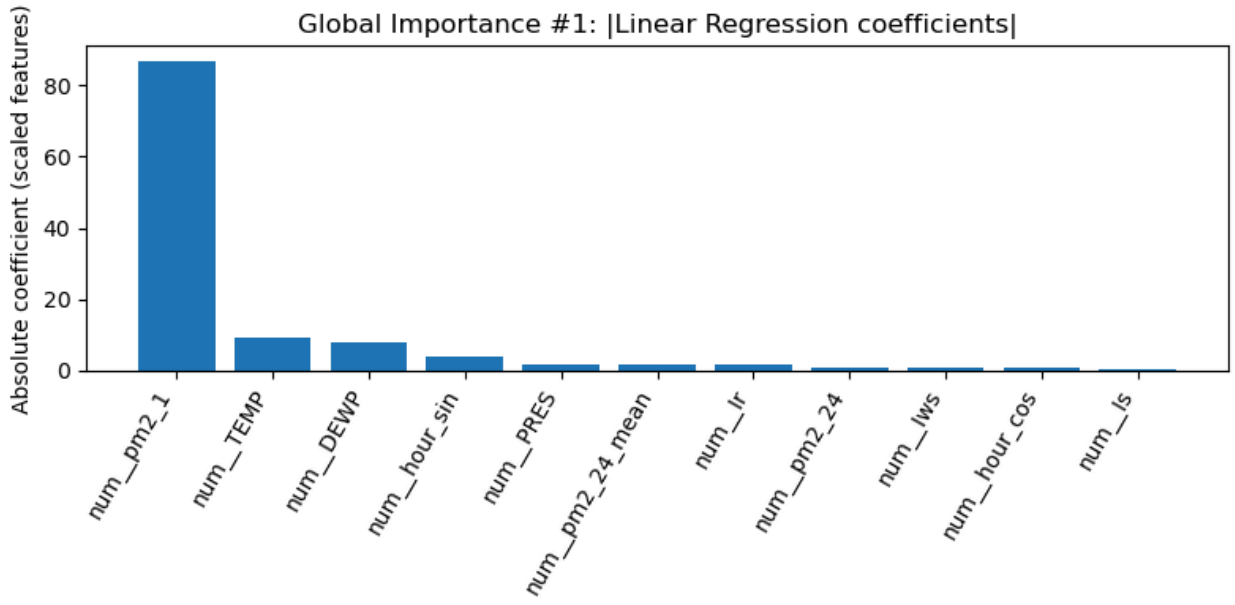


Model	CV RMSE (mean)	CV RMSE (std)	Test RMSE
SVR_RBF	24.181	3.981	21.071
RandomForest	25.659	5.237	21.576
LinearRegression	23.715	3.39	21.621
Ridge	23.715	3.39	21.621
KNN	32.421	6.349	26.018
BaselineMean	93.221	16.289	94.402

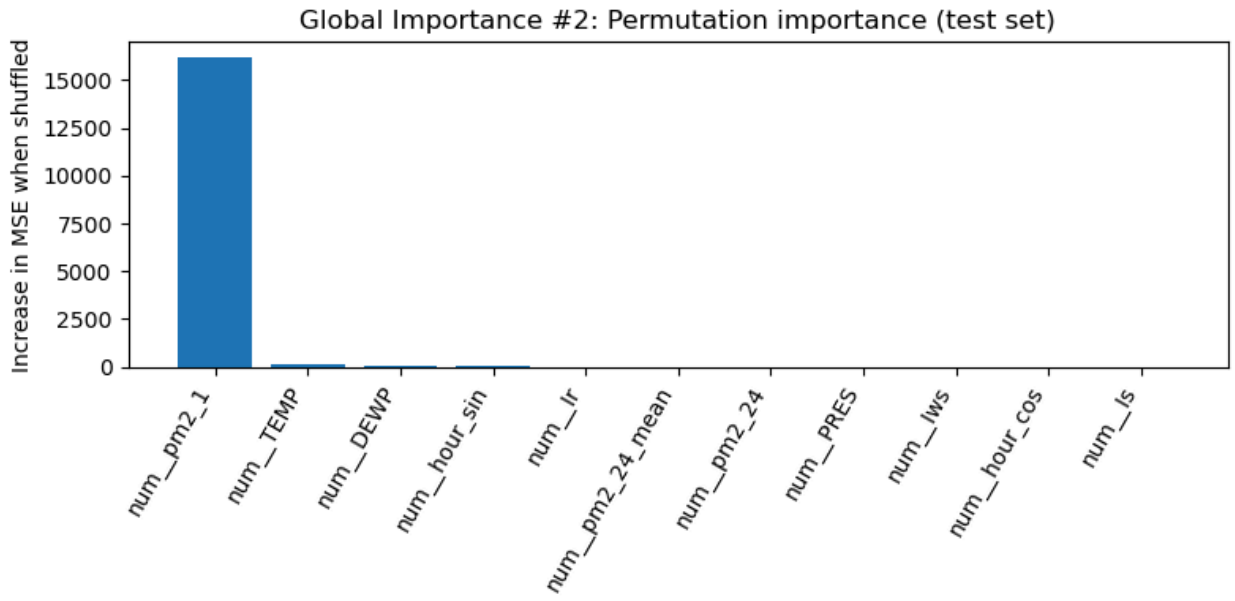
Global Importance Feature(Linear Regression)

I applied three global importance measures for the Linear Regression model: linear regression coefficients, permutation importance and Shap-based importance.

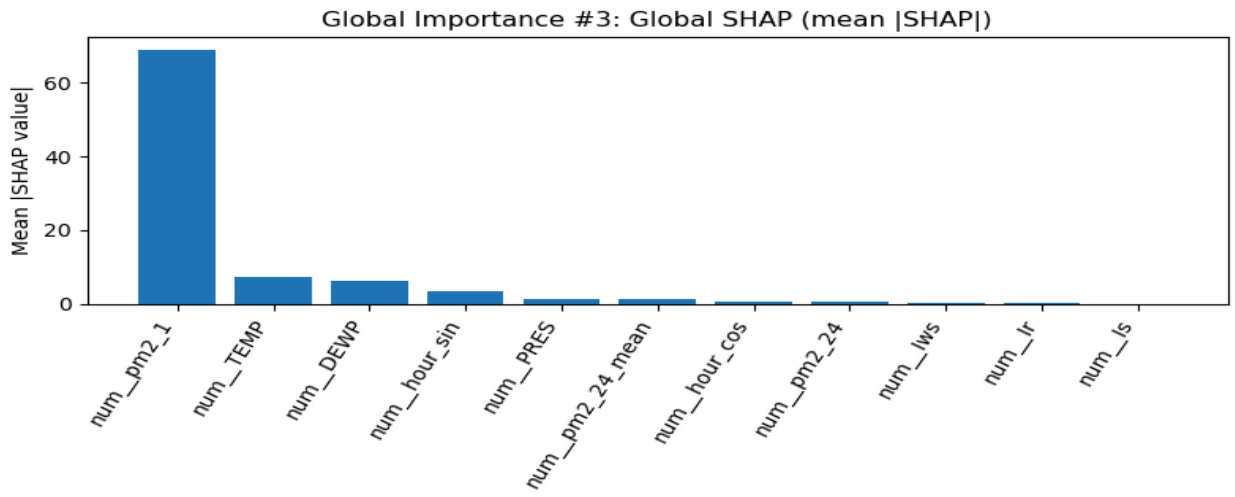
The coefficient-based ranking (Figure 1) is extremely concentrated. The 1-hour PM2.5 lag (pm2_1) dominates by a large margin, with much smaller but still noticeable contributions from temperature (TEMP), dew point (DEWP), and the hour-of-day sine term (hour_sin). Pressure (PRES), the 24-hour rolling mean (pm2_24_mean), and the remaining meteorological and temporal features have very small coefficients.



Permutation importance on the test set (Figure 2) yields a very similar picture. Shuffling pm2_1 produces by far the largest increase in RMSE, confirming that most of the model's predictive power comes from knowing the immediately preceding PM2.5 value. Shuffling TEMP, DEWP, and hour_sin causes modest error increases, while the other features have almost no effect when permuted.



Aggregated SHAP values (Figure 3) again highlights pm2_1 as the strongest driver, followed by TEMP, DEWP, and hour_sin.

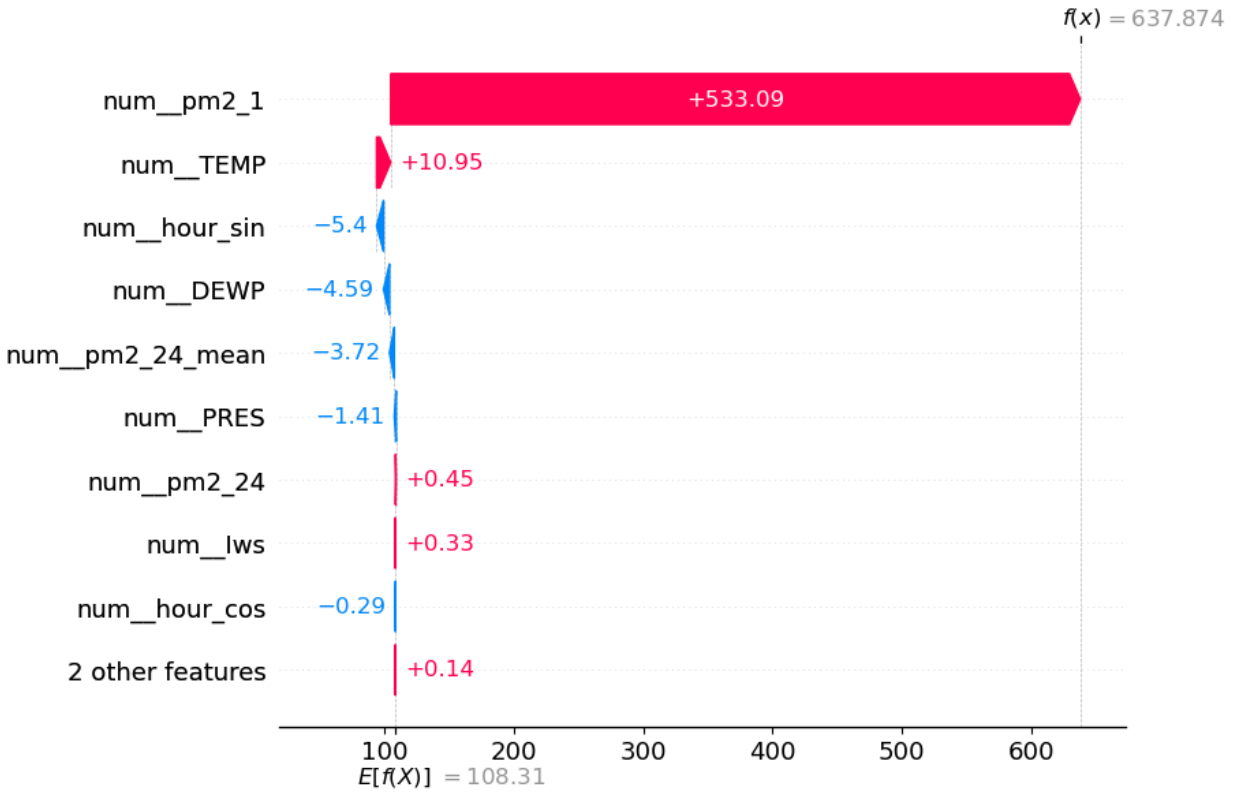


Across all three methods, the conclusion is consistent: very recent PM2.5 levels are the primary predictor, short-term meteorology and time-of-day play a secondary role, and the longer-lag pollution history and wind variables contribute very little to the final linear model.

Local Feature Importance with SHAP

The SHAP force plot for a single extreme pollution event is shown in the figure below. Based on a baseline prediction of 108 µg/m³, the model predicts approximately 638 µg/m³. The prediction is raised by more than +500 µg/m³ by the 1-hour lag pm2_1 alone, indicating that the model anticipates another severe spike primarily due to the extremely high PM2.5 level in the preceding hour. Features like hour_sin, DEWP, pm2_24_mean, and PRES slightly lower the prediction, temperature adds a smaller positive contribution, and variables like pm2_24, Iws, and hour_cos have minor impacts.

In conclusion, the story stays the same: pm2_1 always stands out as the main local driver. The other features barely move the needle—temperature, dew point, and time of day fine-tune things a bit, but that's about it. This fits with what most people expect: Beijing's PM2.5 just doesn't change much from one hour to the next. When the air gets really polluted, it usually sticks around for a while, and the weather only tweaks the concentration. Once you know the PM2.5 from the previous hour, adding longer-term history like pm2_24 or wind data barely adds anything. That's surprising. It means the model acts a lot like a simple one-step autoregressive setup, with just a few small meteorological adjustments layered on.



Outlook

Looking ahead, my main limitation is that the model relies very heavily on the 1-hour PM2.5 lag and only lightly uses longer-term structure. I would first enrich the feature set with clearer temporal signals (weekend vs weekday, month or season of a year) and slightly broaden the hyperparameter grids, for example trying more n_estimators for Random Forest. For interpretability, I would rerun SHAP and permutation importance on separate periods (winter vs summer) to check stability over time. Finally, adding external data such as traffic or emissions could further boost accuracy.

Reference

Chen, Song. "Beijing PM2.5." UCI Machine Learning Repository, 2015, https://doi.org/10.24432/C5JS49.

Zhang, Z., & Zhang, S. (2023). Modeling air quality PM2.5 forecasting using deep sparse attention-based Transformer networks. *International Journal of Environmental Science and Technology, 20*, 13535–13550. https://doi.org/10.1007/s13762-023-04900-1