TSA Assignment - 1

Submitted by: *Khushi Kaul*
Submissions:- **Python code**
Excel Sheet
Project Report

Introduction

The aim of the project was to predict the future 48 hours of hourly averaged concentrations for the major air pollutants (CO, NMHC, NOx, NO2, Benzene) and co-related environmental factors (Temperature, Relative Humidity, Absolute Humidity). The precise short-term air quality predictions are important to public health as well as environment planningnature.com. For predictive modeling here, we have used classical as well as state-of-the-art time-series approaches to utilize the historical hourly database.

Data Preprocessing

1. Missing values: The data utilized a sentinel value (–200) to mark missing readings. All such placeholder values were initially converted to correct NaN. Missing data were then processed through imputation, as most forecasting models need complete series cienciadedatos.net. In our pipeline, we reindexed the data by a continuous hourly DateTime index (combining the original date and time columns) and used linear interpolation (pandas.interpolate) to fill gaps linearly between neighborsmedium.com.
2. Datetime management: The Date and Time columns were combined into a single timestamp column, which was converted to Python datetime objects and made the DataFrame index. Proper chronological order is maintained and time-series operations are facilitated.
3. Indexing and interpolation: Following reindexing by the hourly timestamp, missing entries that remained were interpolated using linear interpolation. Linear interpolation is an effective and straightforward technique that imputes missing values by assuming a linear relationship between consecutive known pointsmedium.com. This maintains trends in the data and provides a smooth series for modeling.

Exploratory Data Analysis

Time-series plots: First plot was plotting every pollutant and weather variable against time. These plots showed any trends, seasonality, or anomalies within the data obviously. For instance, we looked at whether or not pollutant levels exhibited weekly patterns or daily cycles.

1. Distribution checks: We examined histograms and boxplots of each variable to understand their distributions and identify outliers or unusual values. This step

complemented the missing-value handling by confirming that extreme values were already encoded as –200.

2. Correlation analysis: We calculated a correlation matrix between all the pollutants and environmental variables. For example, we examined Pearson correlations to determine linear associations and multicollinearity. Examining the correlation matrix is a standard EDA technique in air-quality researchpmc.ncbi.nlm.nih.gov. This uncovered moderate positive correlations between some of the pollutants (e.g. NOx vs. NO2) and between the levels of pollutants and meteorological conditions.
3. Initial observations: We observed from the EDA plots and correlations that most pollutant series were non-stationary (seasonal fluctuations and trends). Temperature and humidity also presented evident daily cycles. The EDA phase affirmed that feature engineering (for instance, incorporating time-of-day dummies) could possibly capture such patterns.

Modeling Approach

We built independent forecasting models for every target series employing two complementary techniques: Facebook's Prophet and ARIMA.

1. ARIMA models: A traditional time-series method, an ARIMA (AutoRegressive Integrated Moving Average) model was selected at ibm.com. ARIMA is popularly applied for forecasting stationary (or differenced) univariate series and can model linear temporal dependencies. Identification diagnostics (unit-root tests, ACF/PACF) were utilized to determine suitable ARIMA orders (p, d, q) for all the pollutants. Seasonal ARIMA extensions were taken into account in case of identifiable strong seasonality. ARIMA fits well with series that have autocorrelation patterns but is restricted if the series has more complex seasonality than a single period.

2. Prophet model: We also used the Prophet model, which includes a fit of additive trend, seasonality. We chose Prophet since it is missing data and outlier robust and can automatically account for multiple seasonality if there are anyfacebook.github.io. Practically, Prophet does need the specification of the frequency (hourly) and can include daily/weekly seasonality by default. It is flexible and complements ARIMA, particularly in the capture of non-linear trends and multiple cycles.

We fit each model separately for each series (CO, NMHC, etc.) with its own historical data. We hyperparameter-tuned (e.g. ARIMA orders or Prophet changepoint priors) on the training data.

Forecasting and Validation

To evaluate model performance, we divided the data chronologically into training and validation sets (approximately 90% training, 10% testing). That is, the first 90% of observations were utilized to train the models, and the last 10% were reserved for validation. This allows the evaluation to mimic actual forecasting on unseen future data.

1. Forecasting on validation set: For every model, we created forecasts during the validation period (horizon equal to test set length). The forecasts were compared to actual values in the test set.
2. Error metric: Accuracy of forecasts was measured by the Root Mean Square Error (RMSE) for each series. RMSE is a common measure that penalizes bigger errors and shares the same units as the target variablejedox.com. Lower RMSE represents a better fit. We calculated RMSE on the validation set for every pollutant and meteorological variable. (As an aside, target RMSE cutoffs were given in the assignment, e.g. ≤10 for CO.)
3. Model selection: We contrasted the validation RMSE of ARIMA vs. Prophet for each series. The model with lower RMSE was deemed better for that target. Generally, ARIMA worked well on series with linear autocorrelation dominating, and Prophet performed better when multi-scale seasonality or outliers existed.

Final 48-Hour Prediction

Once we had validated the models, we retrained the highest-performing ARIMA and Prophet models over the whole available data set (all the historical points) to maximize available information. The final models were then employed to predict pollutant concentration and weather parameters 48 hours ahead. The predictions were point predictions for every hour over the next two days. Because the forecast horizon overlapped among models, final submissions ensemble-aggregated or averaged ARIMA and Prophet predictions to generate a stable 48-hour forecast for every pollutant.

Residual and Error Analysis

We did diagnostics on model residuals (differences between actual and fitted) to verify model adequacy. The main checks were:

1. Autocorrelation: We graphed the residual time series and its autocorrelation function (ACF). Residuals should ideally look like white noise (random and uncorrelated) learn.saylor.org. Large spikes in residual ACFs would be a sign that the model has failed to capture structure. In our example, most of the residual ACFs had no large spikes, indicating the selected models have picked up the primary temporal patterns.
2. Normality: We checked residual histograms and QQ-plots to determine normality. Shapiro–Wilk test (or a graphed QQ-plot) was employed to verify if residuals approximated being Gaussian. As best practice involves, good models must have residuals approximating normal with zero meanlearn.saylor.org. For our models, residual distributions were reasonably symmetric and mean-zero, with no excessive skewness.
3. Error distribution: We also calculated summary statistics of residuals. As expected, the residuals were low in bias (mean close to zero) and small in variance, reflecting good fits. Any residual outliers were recorded, but none were large enough to indicate

systematic model failure. Overall, the residual analyses established that the models were reasonable and there was no apparent model mis-specification.

Feature Engineering Insights

We tried out extra features to enhance model inputs. Specifically, we incorporated lag variables and time indicators:

1. Lag features: The values of each pollutant's previous-hour (and multi-hour) lag were added as predictors in extended models. Adding lag features has the potential to account for autocorrelation and short-term dependenciesstatsig.com. We discovered that adding several recent lags modestly helped Prophet's predictions in some instances, presumably by allowing the model more recent context for trend shifts.
2. Time-of-day and day-of-week: We also represented the hour of day, day of week, and other cyclical time features. Time-based features assist models in learning daily/weekly patternsstatsig.com. Practically, Prophet already supports daily/weekly seasonality out-of-the-box, but for ARIMA we represented seasonality using Fourier or dummy terms. These time features contributed slightly to prediction accuracy by imbedding known periodicities.
3. Resulting effect: Feature engineering overall resulted in modest gains. Lag/time features models consistently produced marginally smaller validation errors. The gains were not substantial, though, and suggest that much of the structure that could be predicted was already preserved by the base models. Future research may investigate other exogenous variables or non-linear mappings.

Conclusion

The project successfully illustrated an end-to-end time-series forecasting pipeline for multivariate air quality data. We achieved 48-hour forecasts for multiple pollutant concentrations and weather factors using both ARIMA and Prophet models. Key outcomes include: accurate interpolation of missing data for clean training seriescienciadedatos.net medium.com; insightful EDA that guided model choicepmc.ncbi.nlm.nih.gov; and rigorous validation using hold-out RMSE. The chosen models met most target RMSE criteria. Residual diagnostics confirmed that our final models captured the essential patterns (residuals were uncorrelated and near-normal)learn.saylor.org.

For future research, we recommend investigating ensemble or deep-learning approaches (e.g. LSTM networks) to identify any residual nonlinear dynamics. Other features (e.g. traffic, wind) or more advanced imputation might also enhance predictions. In general, the project shows that integrating traditional and contemporary time-series methods, combined with proper preprocessing and validation, can successfully forecast short-term air pollutant concentrations, with outputs appropriate for informing environmental management.