Pollution Forecasting

Problem Description :

Background Information: Air pollution is a significant environmental concern that affects the health and well-being of individuals. Particulate Matter (PM) is a major air pollutant, and PM2.5 refers to fine particles with a diameter of 2.5 micrometers or smaller, which can penetrate deep into the respiratory system. Accurate forecasting of PM2.5 levels is crucial for public health management, environmental planning, and policy-making. In this project, we aim to develop machine learning models to forecast the PM2.5 levels in Jeongnim-Dong from 2018 to 2022.

Data and Data Preprocessing: The dataset contains historical PM2.5 values for various cities. We filter the data to select the city 'Jeongnim-Dong' and extract the PM2.5 values for the specified date range (2018-2022). The dataset is then preprocessed to handle missing dates and ensure a daily frequency of data. The missing PM2.5 values are filled by taking the previous day's value, and the data is sorted by date.

Data Visualization: We visualize the time series data of PM2.5 values for Jeongnim-Dong over the years (2018-2022) using line plots. We also split the data into training and testing sets based on the timestamp, where the training data contains values up to December 31, 2020, and the testing data contains values from January 1, 2021, to January 1, 2022.

Modeling: Two machine learning models, namely the CNN Model (Convolutional Neural Network) and the Mixed Model (combination of Convolutional and Bidirectional LSTM), are implemented to forecast the PM2.5 values.

CNN Model: The CNN Model uses Convolutional layers to extract patterns from the input sequence. The model consists of two Conv1D layers with ReLU activation

functions, followed by Global Average Pooling, Flatten, and Dense layers. The model is compiled using the Huber loss function and Adam optimizer with Mean Absolute Error (MAE) as the evaluation metric. The model is trained for 50 epochs with a custom learning rate scheduler and a custom callback function to stop training when the MAE is less than 10.0.

Mixed Model: The Mixed Model combines Convolutional and Bidirectional LSTM layers. It follows a similar architecture to the CNN Model, with additional Bidirectional LSTM layers for capturing bidirectional temporal patterns. The model is also compiled with the Huber loss function and Adam optimizer and trained for 50 epochs using the same learning rate scheduler and custom callback as the CNN Model.

Forecasting: The models are used to forecast PM2.5 values for the testing data. The data is windowed with a specified window size, and the predictions are obtained for each window. The MAE scores are computed for both models to evaluate their forecasting performance.

Future Work:

1. Experiment with different window sizes to observe the impact on forecasting accuracy.
2. Explore other machine learning models, such as Time Series models (ARIMA, SARIMA) and advanced deep learning models (GRU, Transformer).
3. Incorporate additional environmental and weather-related features into the models to improve forecasting accuracy.
4. Apply hyperparameter tuning techniques to optimize model performance.
5. Implement an ensemble approach to combine multiple models for more robust predictions.
6. Analyze the residuals to identify any patterns or model deficiencies.
7. Deploy the best-performing model as an online forecasting service for real-time PM2.5 predictions.

Possible Framework :

Background Information: Air pollution is a major environmental concern that affects the health and well-being of individuals worldwide. Particulate Matter (PM) is a significant air pollutant, consisting of fine particles with a diameter of 2.5 micrometers or smaller, known as PM2.5. These particles can penetrate deep into the respiratory system and have adverse effects on human health, leading to respiratory problems, heart diseases, and even premature death. Moreover, PM2.5 also contributes to climate change and environmental degradation.

Dataset Information: The dataset used for this project contains historical PM2.5 values for various cities, including the city 'Jeongnim-Dong.' The dataset is time-series data with timestamps representing the date and time of each observation. The PM2.5 values represent the concentration of fine particulate matter in the air, measured in micrograms per cubic meter ($\mu g/m^3$).

Problem Statement: The primary objective of this project is to develop machine learning models capable of accurately forecasting PM2.5 levels in Jeongnim-Dong over a specific time period (2018-2022). By predicting future PM2.5 concentrations, we can provide valuable information for public health management, environmental planning, and policy-making. Accurate forecasting of PM2.5 levels can help authorities take proactive measures to mitigate the impact of air pollution on public health and the environment.

Dataset Preprocessing: The dataset is preprocessed to extract the PM2.5 values for the city 'Jeongnim-Dong' and filter the data within the specified time range (2018-2022). Missing dates in the time series data are handled by filling the missing values with the previous day's PM2.5 concentration. The data is sorted chronologically to ensure a daily frequency of observations.

Data Visualization: Data visualization techniques, such as line plots, are employed to visualize the time series of PM2.5 values in Jeongnim-Dong from 2018 to 2022. This allows us to gain insights into the patterns, trends, and seasonality of the PM2.5 concentrations over the years.

Modeling: Two machine learning models, the CNN Model (Convolutional Neural Network) and the Mixed Model (Combination of Convolutional and Bidirectional LSTM),

are implemented to forecast PM2.5 values. The CNN Model leverages Convolutional layers to extract patterns from the input sequence, while the Mixed Model combines Convolutional and Bidirectional LSTM layers to capture bidirectional temporal patterns.

Forecasting and Evaluation: The trained models are used to forecast PM2.5 values for the testing data. The performance of each model is evaluated using the Mean Absolute Error (MAE) metric, which measures the average absolute difference between the actual and predicted PM2.5 concentrations. Lower MAE scores indicate more accurate forecasts.

Future Applications: Accurate pollution forecasting can have significant implications for public health management and environmental planning. By implementing the best-performing model, this project can be extended to provide real-time and future PM2.5 predictions for Jeongnim-Dong and other cities, aiding decision-makers in implementing effective pollution control strategies and safeguarding the health and well-being of the population.

Conclusion: The Pollution Forecasting project aims to develop machine learning models that can forecast PM2.5 levels in Jeongnim-Dong over the years 2018 to 2022. By leveraging historical data and advanced modeling techniques, this project contributes to efforts in mitigating the impact of air pollution on human health and the environment.

Code Explanation :

*If this section is empty, the explanation is provided in the .ipynb file itself.

Future Work :

Forecasting pollution levels is a crucial task that can have significant implications for public health and environmental planning. While the provided code serves as a good starting point, there are several avenues for future work to improve the accuracy and robustness of the pollution forecasting model. Here is a step-by-step guide on how to implement the future work:

1. Data Augmentation and Feature Engineering:

- Collect additional data sources like meteorological variables, geographical features, and traffic patterns that could influence pollution levels.
- Perform feature engineering to create new relevant features based on domain knowledge.
- Explore techniques like lag features, moving averages, and seasonality to capture temporal patterns in the data.

2. Hyperparameter Tuning:

- Optimize hyperparameters of the existing models (CNN and Mixed Model) using grid search or random search techniques.
- Adjust the learning rate, number of layers, units in layers, and dropout rates to fine-tune the model's performance.

3. Model Ensemble:

- Implement an ensemble of multiple models (e.g., CNN, LSTM, Gradient Boosting) to combine their predictions.
- Use techniques like bagging, boosting, or stacking to create a more robust and accurate ensemble model.

4. Time Series Cross-Validation:

- Adopt time series cross-validation methods like TimeSeriesSplit or Walk-Forward Validation to evaluate the model's performance on out-of-sample data.
- This will provide a more reliable estimate of the model's generalization ability.

5. Long-Term Forecasting:

- Modify the forecasting horizon to predict pollution levels for longer timeframes (e.g., monthly or yearly).
- Adjust the window size and sequence length accordingly to capture long-term patterns.

6. External Data Integration:

- Integrate data from other relevant sources, such as satellite imagery or social media data, to capture real-time pollution events or unexpected events.

7. Model Explainability:

- Use techniques like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) to interpret the model's predictions and understand its underlying decision-making process.

8. Deployment and Automation:

- Deploy the best-performing model as a web service or API to provide real-time pollution forecasts.
- Automate the data collection and preprocessing pipelines to ensure regular updates of the model.

Step-by-Step Guide for Future Work Implementation:

1. **Data Collection and Integration:**
- Gather additional data related to pollution, meteorology, and other relevant features.
- Preprocess the data and merge it with the existing pollution dataset.
2. **Feature Engineering and Data Augmentation:**
- Create new features based on domain knowledge and engineering techniques.
- Augment the dataset to include additional relevant observations.
3. **Hyperparameter Tuning:**
- Define a grid of hyperparameters to be tested for each model.
- Use a validation set to evaluate the performance of different hyperparameter combinations.
- Select the best hyperparameters based on the validation results.
4. **Model Ensemble:**
- Train multiple models (CNN, LSTM, etc.) with different hyperparameter settings.

- Combine the predictions of individual models using weighted averaging or majority voting.

5. **Time Series Cross-Validation:**
- Split the dataset using time series cross-validation methods.
- Evaluate the models on multiple validation folds to estimate their performance.

6. **Long-Term Forecasting:**
- Adjust the forecasting horizon and modify the input sequences accordingly.
- Train and evaluate the models for long-term forecasting.

7. **External Data Integration:**
- Obtain and preprocess external data sources to include in the forecasting model.
- Merge the external data with the existing pollution dataset.

8. **Model Explainability:**
- Use SHAP or LIME to explain the model's predictions and gain insights into its behavior.

9. **Deployment and Automation:**
- Deploy the best-performing model as a web service or API.
- Automate the data collection and preprocessing pipelines for regular updates.

Concept Explanation :

Oh, hello there! So, you want to know all about how we forecast pollution using time series neural networks? Buckle up, my friend, because we're going on a data-driven adventure!

What's the Hype About Time Series?

Imagine you have this super cool gadget that measures pollution levels every day in your city. You end up with a big bunch of pollution readings, each one tied to a specific date. Now, you have yourself a time series – a fancy term for data that's ordered by time!

Let's Dive into Neural Networks:

Neural networks? Yeah, they are like a team of super clever friends who can crunch numbers like nobody's business! Each neuron in the network is like a detective, trying to find hidden patterns in our data. They work together, passing messages and learning from the past to predict the future (just like fortune-tellers, but with data instead of crystal balls!).

Our Two Superhero Models:

1. **The Convolutional Neural Network (CNN):** Picture this: CNN is like a nosy neighbor, peeping through different windows to catch patterns in the data. It slides its window of observation through time and looks for pollution patterns. It's good at finding short-term trends!
2. **The Mixed Model (LSTM with Convolution):** Meet the Mixed Model – the ultimate combo! It's a bit like a superhero duo. LSTM (Long Short-Term Memory) is like the memory genius, remembering stuff from way back. CNN and LSTM work together to tackle short and long-term patterns, making predictions from both perspectives!

Training the Models:

Our superhero models need training – no, not to fight crime, but to understand pollution patterns! We take our time series data and cut it into little sequences, just like how we chop up a big pizza into slices. These sequences become the training data for our models. They look at the past pollution values and try to predict the next one.

The Magical Learning Process:

During training, the models get feedback – kinda like a cooking show, where the judges tell you how your dish tastes. The models adjust their superpowers (weights and biases) to improve predictions with each slice of data they see. They keep learning until they can predict pollution levels with super accuracy!

Testing the Heroes:

Once our models are trained, it's time to put them to the test – like in a superhero battle! We give them some new data, and they make predictions. Will they get it right? Let's find out!

Ensemble: Joining Forces

Two superheroes are good, but why not have an Avengers-style team-up? Our ensemble model brings together CNN and Mixed Model predictions. They combine their strengths, like friends working together on a tough problem. This way, we get even better forecasts!

Presenting the Winners!

The models make their predictions, and we compare them to the actual pollution levels – just like playing "Guess the Pollution." The model with the lowest error wins the contest! Drumroll, please...

Future Forecasting:

Now, these models are like fortune-tellers for pollution! They can forecast future pollution levels, helping us plan and take action to make our cities cleaner and healthier. We've got our superhero data team on the job, and the future looks brighter already!

So there you have it – time series neural networks, the pollution-fighting superheroes! They analyze historical data, learn from the past, and use their superpowers to predict pollution levels. Let's thank our neural friends for helping us make the world a better place, one pollution forecast at a time! ⍰⍰

Exercise Questions :

Exercise 1: How can you preprocess the dataset to ensure a continuous time series with no missing dates?

Answer: To ensure a continuous time series, we can use the Pandas library to create a date range from the start to end date. Then, we can check for missing dates in the original data and fill them with the previous day's pollution value.

Exercise 2: What is the purpose of using the Huber loss function in model compilation?

Answer: The Huber loss function is robust to outliers, which means it reduces the impact of extreme pollution values during training. It combines the benefits of the Mean Absolute Error (MAE) and Mean Squared Error (MSE) loss functions, providing a balance between them.

Exercise 3: How does the Convolutional Neural Network (CNN) model capture short-term trends in pollution data?

Answer: The CNN model uses convolutional layers that slide through the time series data, capturing short-term patterns in the pollution readings. These convolutional filters act like windows, looking at small sequences of pollution values and finding patterns that help in short-term trend forecasting.

Exercise 4: Explain the purpose of the Learning Rate Scheduler in the model training process.

Answer: The Learning Rate Scheduler adjusts the learning rate during model training. In this project, it starts with a high learning rate and reduces it gradually as the training progresses. This technique helps the model to converge faster and achieve better performance.

Exercise 5: What role does the LSTM (Long Short-Term Memory) layer play in the Mixed Model?

Answer: The LSTM layer in the Mixed Model is responsible for capturing long-term dependencies in the pollution data. It can remember past pollution values, which is crucial for forecasting pollution trends that occur over longer periods.

Exercise 6: How do you assess the performance of the models after training?

Answer: The performance of the models is assessed using Mean Absolute Error (MAE). It measures the average absolute difference between the predicted and actual pollution values. Lower MAE indicates better model performance.

Exercise 7: Can you explain the concept of time windowing in the training data preparation process?

Answer: Time windowing involves creating sequences of pollution values from the time series data. Each sequence (window) contains a certain number of consecutive pollution readings. These windows are then used as training samples to predict the next pollution value.

Exercise 8: What is the purpose of using Dropout layers in the model architectures?

Answer: Dropout layers help prevent overfitting in the models. During training, some neurons in the layers are randomly deactivated, forcing the network to learn more robust features and avoid relying too heavily on specific neurons.

Exercise 9: How does the ensemble model combine the predictions from the CNN and Mixed Models?

Answer: The ensemble model takes the predictions from both the CNN and Mixed Models and averages them to get the final prediction. This combination leverages the strengths of both models and provides more accurate and stable forecasts.

Exercise 10: In which scenario would you choose the CNN model over the Mixed Model and vice versa?

Answer: The CNN model is more suitable when short-term pollution trends are of particular interest, as it excels in capturing immediate patterns. On the other hand, the Mixed Model is preferred when both short-term and long-term trends are essential, as it combines the power of CNN and LSTM to handle both scenarios effectively.