Data Management Final Project,
Bill Zhang,
bz232
Git URL: https://github.com/billZhang232/data-management-final-project/tree/main

Project definition:
Problem statement:

- Core objective: This project seeks to create an integrated weather forecasting system that combines historical and real-time weather data for accurate, dynamic predictions.
    - Weather prediction remains a critical area of focus due to its far-reaching implications across industries such as transportation, agriculture, disaster management, and energy.
    - Many current systems lack the capability to quickly adapt to incoming real-time data, limiting their ability to predict sudden or highly localized weather changes effectively.
- Challenges in Existing Systems
    - Data Fragmentation: Historical and real-time data often exist in separate silos, making integration complex.
    - Computational inefficiencies: Processing large volumes of sequential data requires significant computational resources, making these systems less accessible for small-scale operations.
    - Scalability: Current systems may not scale efficiently when applied to new regions or larger datasets
- Proposed solution
    - Use a lightweight relational database to handle historical and real-time data efficiently
    - Apply machine learning algorithms to produce fast and accurate forecasts with minimal computational overhead

Importance
This project is important because it addresses the challenges of efficiently managing, analyzing, and forecasting weather data - a critical area with broad applications in environmental monitoring, agriculture, disaster preparedness, and urban planning. Accurate weather forecasting helps in mitigating risks, planning resources, and adapting to changing climatic conditions. Moreover, the integration of machine learning models like ARIMA adds a layer of complexity and relevance, making it a highly educational and practical endeavor. Additionally, weather prediction has real-world impact, making this project both intellectually stimulating and meaningful

I'm excited about this project because it combines multiple components we learned throughout the semester, such as data acquisition, storage, analysis, data cleaning, and machine learning, all into a cohesive workflow. I'm excited to finally utilize what we learned throughout the course into practice; to be able to design something of my choosing instead of following an exercise. Additionally, weather prediction has real-world impact, making this project both intellectually stimulating and meaningful.

Existing issues in current Data Management practices:

- Fragmented Data Sources; Weather data is often scattered across multiple sources, making it difficult to integrate into a single unified database for analysis
- Inefficient Storage and Querying: Many systems lack efficient database solutions for time-series data, leading to slower queries and limited scalability

- Data Quality Challenges: There are going to be missing or noisy data which often hampers the ability to perform accurate predictions
- Limited Predictive Capabilities: Since I am unable to get the api for accurate historical data, I had to generate historical data, which is far less accurate and thus greatly limits my training quality
- Complexity of other variables: real life has far more variables in play. Weather prediction in real life is much more difficult to predict. Without even the most advanced technology knowing some of these variables, it will make my model even less accurate.

Prior work

- NOAA's weather database: NOAA has provided a large, organized repository of weather data.
- Weather APIs, such as OpenWeather or WeatherStack: These services provide current and historical weather data, but they are primarily focused on data retrieval rather than comprehensive management or predictive analytics
- Time Series Forecasting with ARIMA: ARIMA is a well-established method for forecasting time-series data. Research papers and applications in domains like energy consumption, stock prices, and climate monitoring demonstrate its effectiveness.
- Machine Learning-Based Forecasting Models: Recent advancements integrate ARIMA with other methods, such as neural networks or ensemble learning, to enhance predictive accuracy in dynamic environments
- Data Lakes and Cloud-Based Solutions: Platforms like AWS and Google Cloud offer scalable storage and analytics solutions for weather data, but these require substantial resources and expertise

Data Usage

- The project used synthetic weather data and real-world weather data:
    - Synthetic Weather data was simulated for 30 days at hourly intervals. The simulation accounted for seasonal patterns, geographical effects, and random variations in temperature, humidity, and wind speed.
    - Real-world weather data is the current data, which was fetched from the OpenWeatherMap API using the latitude and longitude of New York City. The API provided actual temperature, humidity, and wind speed details.
- These datasets enabled the analysis of historical trends and the prediction of future weather conditions

Models/techniques/algorithms

- Data Management
    - SQLite was used to manage and store weather data efficiently. Both synthetic and real-time data were organized in a structured format for easy access and scalability.
- Data Processing:
    - Normalization of timestamps and scaling of data using MinMaxScaler prepared the dataset for time-series modeling and machine learning.
- Forecasting:
    - ARIMA was used to predict temperature trends based on the time-series data.
    - Humidity and wind speed predictions were derived from ARIMA's temperature forecasts, reflecting real-world weather patterns.
- Validation:
    - Historical data was cleaned and checked for missing or invalid timestamps to ensure high-quality inputs for modeling

Experiments and Results

- Synthetic Data Validation: Simulated weather data was generated to test the system's ability to handle both historical and real-time data effectively
- ARIMA Model Training: The ARIMA model was trained on hourly temperature data to forecast the next 7 days
- Multivariate Data Augmentation: I added predictions for humidity and windspeed as well based on temperature forecast for a more complete prediction
- Data Retrieval and Management: SQLite queries were tested for accuracy and performance to make sure data was properly handled.

Detailed Project Explanation:

First Block of Code:
Function: generate_synthetic_weather_data*start_date, latitude, longitude, days=5)

- Purpose: to generate synthetic historical weather data for a specific location and time range
    - Start from start_date and iterate over the number of hours in the specified number of days(days*24)
    - Base temperature is calculated as 30-abs(latitude)*.3 (cooler at higher latitudes) and adjusted with:
        - A sinusoidal function np.sin to simulate temperature fluctuations over the year
        - Added random noise to add realistic variation
    - Calculates humidity, which is inversely correlated with temperature. It's clamped between 20% and 100%
    - Wind speed is simulated as a random value between 0 to 15 m/s
        - This is another limitation of the project, as I could not figure out an accurate way of predicting wind speed while being able to incorporate it into this project
    - Append the generated data (timestamp, temperature, humidity, wind speed, latitude, longitude) to a list
    - Increment the time by 1 hour (timedelta(hour=1))
    - Finally, convert to datafram by using pandas.DataFrame to create a structured dataset with the appropriate columns, and outputs a data frame with synthetic weather data for the specified period.
- Generating and Displaying Data
    - The example location I used for this project was for New York City, at lat=40.71 and long=-74
        - For this project, I kept the location the same to avoid making the training model too complex. This was a way I used to reduce the complexity and introduce fewer variables to be taken into account for
    - I then calculate the 30 days prior to the current UTC time
    - Then I call the method to generate synthetic weather data and display the synthetic data for the first 5 rows to make sure the data I generated is acceptable and working

The following is on SQLite Database Initialization and for storing generated/fetched data:

- SQLite Database Initialization(Function: initialize_database())
    - The purpose of the function is to set up a local SQLite database to store weather data
    - The program first creates/connect to a database file named weather_data.db using SQLite
    - Then create a table weather, with columns for timestamp, temperature, humidity, wind speed, latitude, longitude
    - Finally commit changes to save the table and close the connection
- Storing synthetic data in SQLite(Function: store_synthetic_data(data, connection))
    - The purpose of this function is to insert the generated weather data into the weather table
    - It first loop through the rows of the DF using data.iterrows()
    - It then inserts each row into the database table using an SQL insert query
    - Finally saves the inserted rows to the database and close the cursor.
- Fetching Current Weather Data(Function: featch_current_weather(lat, lon, api_key)
    - The function will retrieve current weather data for a given location using OpenWeatherMap API, which I have created an account for to generate an API key
    - It first defines the API endpoint using the API URL, and passes the following parameters:
        - Lat, Lon are the location coordinates
        - Appid is the API key for authentication
        - Units indicates the metric system
    - An HTTP request is made after to send a GET request with the parameters, and it will raise an exception for any HTTP errors
    - Finally it will return the response as a JSON object
- Store Current Weather Data in SQLite(Function: store_current_weather(data, latitude, longitude)
    - This function will add the current weather data we fetched and store it into the weather table
    - It first retrieves temperature, humidity, and wind speed
    - It will then add these current weather information into the weather table with the current timestamp
- Finally, we will initialize the database, call the functions to generate the historical data as well as fetch the current weather data, and store these data into the database we created using SQLite

Second Block of Code:

- The query function retrieves the columns timestamp, temperature, humidity, wind speed, latitude, and longitude using an SQL query
    - The data is returned as a pandas DF for further processing
- The retrieved DF is checked for the presence of essential columns. If any of the previously mentioned columns are missing an error is raised.
    - Furthermore, timestamps are converted to datetime objects. Rows with invalid or missing timestamps are removed are part of the data cleaning process to ensure high quality data usage for training

- A frequency(h for hourly data) is assigned to the DF to indicate that data points occur at regularly intervals, and this ensures compatibility when we later use ARIMA to train our model
- Rows with missing values in other columns are removed to ensure the quality of the data
- ARIMA Model Training
    - The temperature column is extracted as the target time-series for ARIMA modeling, which is configured as (2, 1, 2) where
        - 2 refers to the number of autoregressive terms
        - 1 indicates differencing to make the series stationary
        - 2 specifies the number of moving average terms
    - The model is fitted using historical temperature data, which we generated.
- A new data frame forecast_df is created, which stores the forecasted temperatures along with their corresponding timestamps. Dummy adjustments(humidity inversely correlated with temperature, and wind speed simulated with random uniform values) are applied to generate predicted values for humidity and wind speed
- Display results at the end

Results:

```
Forecasted Temperatures for the Next 7 Days:
                                                  timestamp  \
2024-12-08 23:36:04.535993+00:00 2024-12-08 23:36:04.535993+00:00
2024-12-09 00:36:04.535993+00:00 2024-12-09 00:36:04.535993+00:00
2024-12-09 01:36:04.535993+00:00 2024-12-09 01:36:04.535993+00:00
2024-12-09 02:36:04.535993+00:00 2024-12-09 02:36:04.535993+00:00
2024-12-09 03:36:04.535993+00:00 2024-12-09 03:36:04.535993+00:00
...                                                            ...
2024-12-15 18:36:04.535993+00:00 2024-12-15 18:36:04.535993+00:00
2024-12-15 19:36:04.535993+00:00 2024-12-15 19:36:04.535993+00:00
2024-12-15 20:36:04.535993+00:00 2024-12-15 20:36:04.535993+00:00
2024-12-15 21:36:04.535993+00:00 2024-12-15 21:36:04.535993+00:00
2024-12-15 22:36:04.535993+00:00 2024-12-15 22:36:04.535993+00:00

                                  predicted_temperature
2024-12-08 23:36:04.535993+00:00              13.619941
2024-12-09 00:36:04.535993+00:00              13.745929
2024-12-09 01:36:04.535993+00:00              13.677128
2024-12-09 02:36:04.535993+00:00              13.735646
2024-12-09 03:36:04.535993+00:00              13.684485
...                                                 ...
2024-12-15 18:36:04.535993+00:00              13.708365
2024-12-15 19:36:04.535993+00:00              13.708365
2024-12-15 20:36:04.535993+00:00              13.708365
2024-12-15 21:36:04.535993+00:00              13.708365
2024-12-15 22:36:04.535993+00:00              13.708365

[168 rows x 2 columns]

Forecasted Weather Conditions for the Next 7 Days:
                                                  timestamp  \
2024-12-08 23:36:04.535993+00:00 2024-12-08 23:36:04.535993+00:00
2024-12-09 00:36:04.535993+00:00 2024-12-09 00:36:04.535993+00:00
2024-12-09 01:36:04.535993+00:00 2024-12-09 01:36:04.535993+00:00
2024-12-09 02:36:04.535993+00:00 2024-12-09 02:36:04.535993+00:00
2024-12-09 03:36:04.535993+00:00 2024-12-09 03:36:04.535993+00:00
2024-12-09 04:36:04.535993+00:00 2024-12-09 04:36:04.535993+00:00
2024-12-09 05:36:04.535993+00:00 2024-12-09 05:36:04.535993+00:00
2024-12-09 06:36:04.535993+00:00 2024-12-09 06:36:04.535993+00:00
2024-12-09 07:36:04.535993+00:00 2024-12-09 07:36:04.535993+00:00
2024-12-09 08:36:04.535993+00:00 2024-12-09 08:36:04.535993+00:00
2024-12-09 09:36:04.535993+00:00 2024-12-09 09:36:04.535993+00:00
2024-12-09 10:36:04.535993+00:00 2024-12-09 10:36:04.535993+00:00
2024-12-09 11:36:04.535993+00:00 2024-12-09 11:36:04.535993+00:00
2024-12-09 12:36:04.535993+00:00 2024-12-09 12:36:04.535993+00:00
2024-12-09 13:36:04.535993+00:00 2024-12-09 13:36:04.535993+00:00
2024-12-09 14:36:04.535993+00:00 2024-12-09 14:36:04.535993+00:00
2024-12-09 15:36:04.535993+00:00 2024-12-09 15:36:04.535993+00:00
2024-12-09 16:36:04.535993+00:00 2024-12-09 16:36:04.535993+00:00
2024-12-09 17:36:04.535993+00:00 2024-12-09 17:36:04.535993+00:00
2024-12-09 18:36:04.535993+00:00 2024-12-09 18:36:04.535993+00:00
2024-12-09 19:36:04.535993+00:00 2024-12-09 19:36:04.535993+00:00
2024-12-09 20:36:04.535993+00:00 2024-12-09 20:36:04.535993+00:00
2024-12-09 21:36:04.535993+00:00 2024-12-09 21:36:04.535993+00:00
2024-12-09 22:36:04.535993+00:00 2024-12-09 22:36:04.535993+00:00
```

	predicted_temperature	predicted_humidity	\
2024-12-08 23:36:04.535993+00:00	13.619941	87.843331	
2024-12-09 00:36:04.535993+00:00	13.745929	87.126334	
2024-12-09 01:36:04.535993+00:00	13.677128	88.345105	
2024-12-09 02:36:04.535993+00:00	13.735646	86.020828	
2024-12-09 03:36:04.535993+00:00	13.684485	86.087106	
2024-12-09 04:36:04.535993+00:00	13.729273	81.817885	
2024-12-09 05:36:04.535993+00:00	13.690061	97.092064	
2024-12-09 06:36:04.535993+00:00	13.724391	89.525974	
2024-12-09 07:36:04.535993+00:00	13.694335	85.329385	
2024-12-09 08:36:04.535993+00:00	13.720649	85.585813	
2024-12-09 09:36:04.535993+00:00	13.697611	95.492457	
2024-12-09 10:36:04.535993+00:00	13.717781	90.823583	
2024-12-09 11:36:04.535993+00:00	13.700122	100.000000	
2024-12-09 12:36:04.535993+00:00	13.715582	93.464578	
2024-12-09 13:36:04.535993+00:00	13.702047	87.404791	
2024-12-09 14:36:04.535993+00:00	13.713897	94.966230	
2024-12-09 15:36:04.535993+00:00	13.703523	88.194998	
2024-12-09 16:36:04.535993+00:00	13.712605	88.911215	
2024-12-09 17:36:04.535993+00:00	13.704653	84.808642	
2024-12-09 18:36:04.535993+00:00	13.711615	79.334351	
2024-12-09 19:36:04.535993+00:00	13.705520	87.594124	
2024-12-09 20:36:04.535993+00:00	13.710856	87.147885	
2024-12-09 21:36:04.535993+00:00	13.706185	83.976874	
2024-12-09 22:36:04.535993+00:00	13.710275	87.827772	

	predicted_wind_speed
2024-12-08 23:36:04.535993+00:00	9.751733
2024-12-09 00:36:04.535993+00:00	11.450746
2024-12-09 01:36:04.535993+00:00	11.221554
2024-12-09 02:36:04.535993+00:00	2.416377
2024-12-09 03:36:04.535993+00:00	13.459593
2024-12-09 04:36:04.535993+00:00	8.911758
2024-12-09 05:36:04.535993+00:00	1.281871
2024-12-09 06:36:04.535993+00:00	3.180424
2024-12-09 07:36:04.535993+00:00	11.941957
2024-12-09 08:36:04.535993+00:00	8.328210
2024-12-09 09:36:04.535993+00:00	14.575704
2024-12-09 10:36:04.535993+00:00	1.938423
2024-12-09 11:36:04.535993+00:00	5.968680
2024-12-09 12:36:04.535993+00:00	9.488888
2024-12-09 13:36:04.535993+00:00	8.386982
2024-12-09 14:36:04.535993+00:00	11.348174
2024-12-09 15:36:04.535993+00:00	2.561482
2024-12-09 16:36:04.535993+00:00	12.590771
2024-12-09 17:36:04.535993+00:00	7.653602
2024-12-09 18:36:04.535993+00:00	1.238099
2024-12-09 19:36:04.535993+00:00	12.121192
2024-12-09 20:36:04.535993+00:00	9.182502
2024-12-09 21:36:04.535993+00:00	3.198599
2024-12-09 22:36:04.535993+00:00	14.052157

Explanation:

1. Forecasted Temperatures for the Next 7 Days:
    - This table contains hourly temperature predictions for the next 7 days. The timestamp column marks the prediction's time, and the predicted_temperature column provides the hourly forecasted temperature.
    - There are 168 rows, representing 24 hours × 7 days. The temperatures seem relatively stable, indicating the ARIMA model captured a steady pattern from the historical data.
2. Forecasted Weather Conditions for the Next 7 Days:
    - This includes hourly forecasts for temperature, humidity, and wind speed:
        - Predicted Humidity: Derived as inversely correlated with temperature, with minor random noise adjustments to simulate variability.
        - Predicted Wind Speed: Random values between 0 and 15 m/s were generated, reflecting natural fluctuations.

    - The timestamps are consistent across all variables, ensuring proper time alignment for each forecast.
3. Observations:
    - The predictions for temperature, humidity, and wind speed follow a plausible trend but lack the complexity of real-world weather patterns due to simplified assumptions (e.g., inverse correlation for humidity and random wind speed).
    - Temperature predictions stabilize over time, likely reflecting the ARIMA model's inability to incorporate sudden or complex environmental changes (e.g., storms and heat waves).

These outputs demonstrate the system's capability to integrate historical data and predict weather variables over time, though with limitations in accuracy and complexity due to synthetic and simplified input data.

Overall Limitations and Possible Improvements

- We can optimize ARIMA hyperparameters(p, d, q) using automated techniques such as grid search.
- We can further improve and evaluate the model's accuracy and performance using metrics such as MAE
- Incorporate and use more data sources for better data quality. Using just one API is likely not good enough for an accurate weather prediction model.

Key Differences between Project Proposal and Final Project Report:

1. Implementation Simplifications: The final project scaled down the complexity, prioritizing simplicity and feasibility over the advanced design outlined in the proposal.
2. Limited Data Integration: While the proposal aimed to integrate diverse datasets, the final project used a combination of synthetic and real-time API data due to constraints.
3. Focus Shift: The final project concentrated more on implementing ARIMA and SQLite for a single geographical region (New York City) rather than exploring multi-regional scalability or diverse data sources.

The proposal outlined a more ambitious project leveraging advanced tools and large-scale infrastructure, while the final project focused on achievable goals within the given timeframe and resources. The biggest issue for this was the time constraint as well as accessibility to some of the APIs. Notably, the historical data API required a subscription to access, which was something difficult for me to acquire for this class project. Despite the differences, the final project still effectively demonstrated core data management, machine learning, and predictive analytics concepts.