Documentation of the Assignment:

Evaluation of GFS, IFS, and ICONEU Predictions Against Actual Weather Station Data

Contents

Goal of the Project

The predicted values from three global weather forecast models (IFS, GFS, and ICONEU) are compared with actual weather station data. Each forecast model is evaluated independently against the same data, and the performance of the models is discussed.

1. Non-Technical Analysis

1.1. Quick Look at Data Integrity

According to the information provided in the instructor's MS Word file, the weather station collecting real-time data is located in Munich, Germany, and provided data for the period 01-07/06/2024. The forecast data has been provided for the same area, i.e., Munich, Germany, and the same period, 01-07/06/2024. The weather station data starts on 2024-06-01 at 00:00:00, and the forecast data starts at the same time. The data appears to be consistent and reasonable.

1.2. Google Search for Weather in Munich at the same Period

The search is based on weather platforms and sites that provide historical weather data. The goal of this section is to gain a quick understanding of the weather conditions in Munich from 01 to 07/06/2024. The following tables were created by manually calculating the average temperature, wind speed, and the percentage of the most common wind direction from each platform. Saying 'manually' means that no weekly averages for temperature, wind speed or direction to cover the period from 01 to 07/06/2024 were provided and hence, the author's own calculations were necessary. The percentages for wind direction show how often the most common wind direction occurred during this period. For example, if most winds came from the west (either west-southwest or west-northwest), the tables will show the percentage of west-directed winds.

Weatherspark [1]

Table 1 is based on historical, daily weather data from Weatherspark for June 2024.

Table 1: Historical, daily (from 01 to 07) June 2024 data for Munich from "Weatherspark". [1]

Avg. Temp. (^oC)	Wind Speed (m/s)	Wind Direction
15.55	1 - 8	54.76% W

The large wind speed range occurred from the fact that the wind speeds fell mostly in the categories of "light air", "light breeze", "gentle breeze" and "moderate breeze". That is, the expected wind speed values are between 1 and 8 m/s.

Useful Note: Weatherspark mentions that wind direction is not measured for hours where the mean wind speed is lesser than 1 mph (≅ 0.45 m/s).

Weather Underground [2]

Table 2 is based on historical, daily weather data from Weather Underground for June 2024. Unfortunately, this platform does not provide wind direction information.

Table 2: Historical, daily (from 01 to 07) June 2024 data for Munich from "Weather Underground". [2]

Avg. Temp. (°C)	Wind Speed (m/s)	Wind Direction
15.87	3.23	-

Time and Date [3]

Table 3 is based on historical daily weather data from Time and Date for June 2024.

Table 3: Historical, daily (from 01 to 07) June 2024 data for Munich from "Time and Date". [3]

Avg. Temp. (°C)	Wind Speed (m/s)	Wind Direction
15.03	3.37	75% W

Conclusions of the Google Search:

The above quick search indicates that daily temperature values between 15 and 16 °C are considered near average for Munich in June. The wind speeds are ranged between 1 and 8 m/s with most values expected to be approximately 3 and 4 m/s. The dominant wind direction is from the west, with most values likely between 225° and 315°. Secondary wind directions are expected to be between 180° and 225°, as well as between 315° and 360°.

1.3. Google Search for Special Weather Phenomena in Munich during the same Period

Extreme storms that led to floods, resulting in human casualties, occurred in southern Germany (including Munich) from June 1 to June 3, 2024 [4]. This weather phenomenon may have affected the operational functionality of the weather station collecting the data or may have produced anomalies in the recorded data.

2. Technical Analysis

This section outlines all technical steps necessary for the completion of the project.

2.1. Import Libraries and Read Raw Data

In this step, all necessary libraries are imported. Also, pyarrow is installed to allow Pandas to read the original datasets, which are stored as '.parquet' files.

2.2. Data Preprocessing

This step encompasses all necessary data preprocessing prior to evaluation. It is a core part of the project, as careful step-by-step planning is essential to ensure the integrity of the evaluation.

2.2.1. Basic Preprocessing

This foundational step ensures that column names are consistent and meaningful, and that 'dates' are converted to the appropriate datetime data type.

2.2.2. Actual Data Preprocessing

Dealing with NaN Values:

The raw data contain meteorological angles representing wind directions. These meteorological directions will need to be converted in mathematical directions or wind components (u, v). Therefore, it is important to handle NaN values before creating new columns from the previous ones ensuring a solid approach. The NaN values in the actual data are filled using interpolation: the 'linear' method for temperature and wind speed, and the 'nearest' method for meteorological directions. The 'linear' method is not preferred for meteorological directions due to their circular nature. This is because the formula for 'linear' interpolation depends on calculations over regular values, and using circular values in the formula is likely to lead to incorrect results. However, the 'nearest' interpolation method needs to be used together with the forward fill method to fill the 100% of the meteorological direction column.

It is noteworthy that models evaluated with dropped NaNs as well. The evaluation results are almost identical either for filled or dropped NaN.

Dealing with the Circular Nature of Data:

After handling the NaN values, the next step is to resample the actual data to match the hourly format of the forecast data. A simple resampling by the mean is ineffective for meteorological directions.

For this reason, the meteorological directions are converted to mathematical directions in radians, using the following formula:

$$\varphi = [(270 - \alpha + 360)\%360]\frac{\pi}{180} \quad (\mathbf{1})$$

Here, α is the meteorological angle, and the modulo operator '%' ensures the result will not exceed 360°. The final results of the formula (1) are positive mathematical angles expressed in radians, meaning, they will range from 0 to 2π.

Then the zonal, u and meridional, v wind components can be calculated, assuming a wind speed magnitude of 1:

$$u = \cos(\varphi) \quad (\mathbf{2})$$

$$v = \sin(\varphi) \quad (\mathbf{3})$$

Resampling:

After calculating the wind components from the original meteorological directions, resampling can be effectively applied because the wind components represent Cartesian coordinates, which can be averaged. The resampling method uses 'first' for temperature and 'mean' for wind speed, u, and v parameters, while other parameters are automatically dropped from the DataFrame. The hourly format requires the first value for temperature timestamps, as temperatures are points in time.

Converting Wind Components back to Directions:

The final preprocessing step involves converting the wind components back to meteorological and mathematical directions, using formulas (4) and (5) respectively:

$$\varphi = \left[\left(\mathrm{arctan2(v,u)} * \frac{180}{\pi}\right) + 360\right]\%360 \quad (\mathbf{4})$$

$$\alpha = (270 - \varphi + 360)\%360 \quad (\mathbf{5})$$

Here, the $arctan2$ is a function available in many programming languages which effectively handles all special cases of the corresponding arctangent function ($arctan(v/u)$) in mathematics. It accurately calculates the angle for any component combination covering all four quadrants, while preventing divisions by zero.

The methodology described from the 'Dealing with NaN Values' subsection up to this point ensures that the wind direction information is properly processed and captured. At this stage, the actual dataset is almost fully preprocessed.

Final Actual Datasets:

Considering that the purpose of the assignment is to evaluate three different forecast models, each of which forecasted for different time intervals, the actual dataset must align with these intervals. Both the actual and forecast datasets start at the same timestamp. Since all datasets share the same starting timestamp, the differences mentioned relate to the ending timestamps. To be more specific, GFS forecasts data for the full 7 days, IFS stops at the end of the 6th day, and ICONEU stops at the end of the 5th day. To ensure the actual observations align with the forecasts, adjustments are made for the IFS and ICONEU models. Ultimately, the actual data to be compared with IFS forecasts will consist of rows 0 to 143 (144 rows) covering up to 2024-06-06 23:00:00, while the actual data to be compared with ICONEU will consist of rows 0 to 119 (120 rows) covering up to 2024-06-05 23:00:00.

2.2.3. Forecast Data Preprocessing

Adjusting Forecast Data:

Since temperature forecasts are specific to points in time, the first forecast row is only needed for comparison with the actual temperature measurement at 2024-06-01 00:00:00. The corresponding wind parameter forecasts relate to 2024-05-31 23:00:00 and should be dropped. Similarly, the final observation if the forecast data at 2024-06-08 01:00:00 is only relevant for wind parameters at 2024-06-07 23:00:00, while the temperature value cannot be compared with a corresponding actual measurement. Therefore, the wind parameters should be adjusted by shifting them -1, while the temperature values should remain in their original positions. The final observation of the adjusted forecast data should be dropped.

The same adjustments apply to the IFS temperature value at the timestamp 2024-06-07 01:00:00 and the ICONEU temperature value at 2024-06-06 01:00:00, both of which should be set to NaN. This methodology ensures that there are 168, 144 and 120 non-NaN values for GFS, IFS and ICONEU forecasts, respectively.

Final Forecast Datasets:

The final forecast datasets for IFS and ICONEU can be created by retaining all rows where the IFS and ICONEU forecast values are not NaN. The forecast dataset for GFS does not require any further actions.

2.2.4. Wind Direction Data Preprocessing

Calculating the Angular Differences:

The wind direction requires separate analysis due to the circular nature of the data. A new dataset is created with columns representing the angular differences between the actual meteorological directions and the forecasted. For instance, to calculate the angular differences between the GFS forecasts and the actual dataset, the following code can be applied:

$$\text{GFS Diff.} = \text{np.abs(actual meteo} - \text{GFS meteo)} \quad \textbf{(6)}$$

Here, "actual meteo" and "GFS meteo" represent the actual and GFS forecast value for meteorological directions, respectively, whereas the '.abs()' function keeps the absolute value of the above difference. A custom function then fills the angular differences column with:

$$return\; np.minimum(GFS\; Diff.,360 - GFS\; Diff.) \quad \textbf{(7)}$$

This approach guarantees that the final angular differences columns contain valid values. For instance, the difference between 10° and 350° is calculated as 20°, not 340°.

Final Wind Direction Datasets:

The same process used to create the final actual datasets is applied here as well.

2.3. Exploratory Data Analysis

The EDA is mainly presented within main.py file making this section brief by showing only the most interesting visualizations. The time series of wind speeds over time (figure 1) indicates that all three models perform even worse that the baseline (horizontal line).

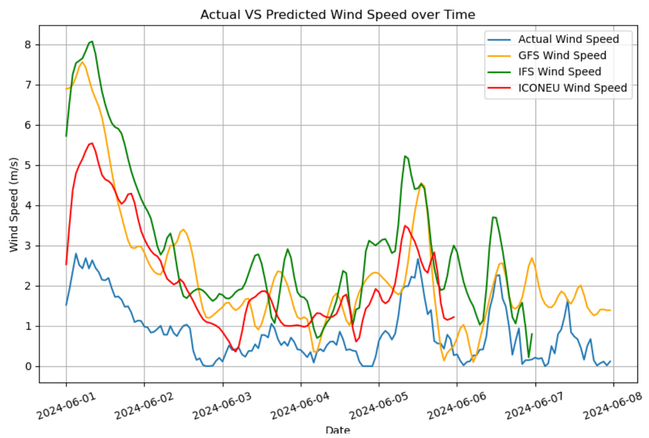

Figure 1: Actual VS Predicted Wind Speed over Time

However, fig. 2 indicates that IFS and ICONEU might have accurately predicted the wind directions.

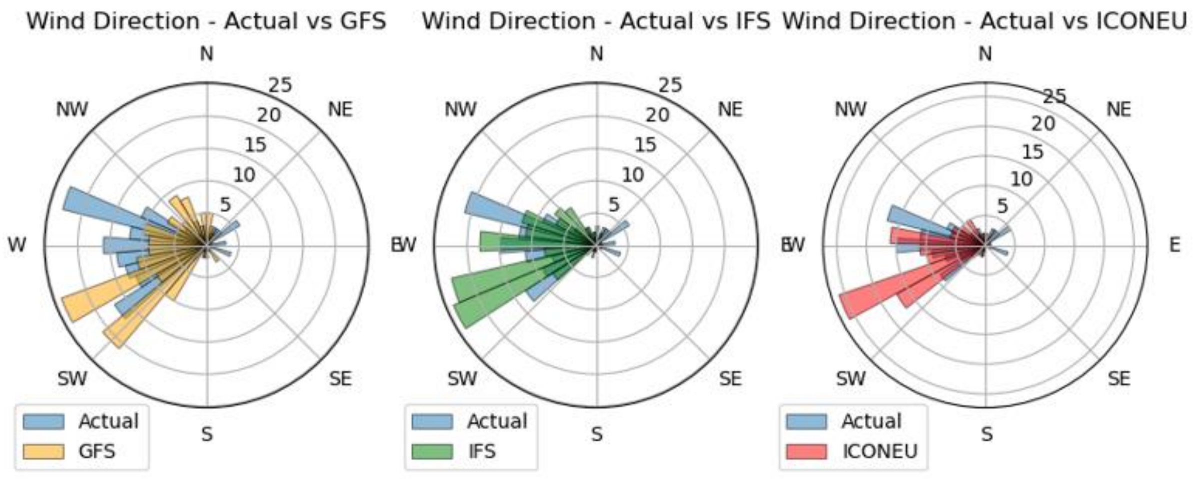

Figure 2: Wind Directions for all three models compared to the actual wind directions.

2.4. Evaluation (Results and Discussion)

The metrics used for evaluating the errors between the actual and forecasted temperature and wind speed values are the Mean Absolute Error (MAE), the Mean Squared Error (MSE), and the Root Mean Squared Error (RMSE), while the metric used to assess how well each model fits the data is R-squared (R^2). For the wind directions the error metrics used is MAE and RMSE and Median Absolute Error (Med. AE).

2.4.1. Metrics Formulas and Interpretation

Mean Absolute Error (MAE):

$$MAE = \frac{\sum_{i=1}^{n} | y_i - \hat{y}_i |}{n} \quad (\mathbf{8})$$

Equation (8) indicates that MAE is a metric that calculates the average magnitude of the errors between the forecasted values, $\hat{y}_i$ and the actual values y_i for a sample of n observations. This is an easy-to-interpret metric providing the errors with the original parameter units with the lower values indicating better model performance.

Mean Squared Error (MSE):

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n} \quad (\mathbf{9})$$

Equation (9) indicates that MSE is the average of the squared differences between the actual and the forecasted values (sum of squares error). This metric penalizes larger errors more significantly due to squaring the differences, making it sensitive to outliers. However, the problem with this metric is that it complicates interpretability, as the original parameter units are squared.

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad \mathbf{(10)}$$

Equation (10) converts the MSE back to the original units by applying a square root to MSE. This metric enhances interpretability, allowing for a direct comparison of RMSE values with the actual data. RMSE is very useful as it balances between sensitivity to larger errors and interpretability.

R-squared (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad \mathbf{(11)}$$

Where $\bar{y}$ is the mean of the actual values. Equation (11) indicates that as the R^2 value approaches 1, the fit of the model to the actual data improves. This metric evaluates how well a model fits the actual data rather than the prediction errors. However, it is not suitable for evaluating wind direction because it involves the mean of the actual values in the denominator. Due to the circular nature of wind direction, calculating the average meteorological direction is inappropriate.

Median Absolute Error (Med. AE):

There is no single formula for calculating this metric. However, it can be computed programmatically by sorting the properly calculated angular differences and finding the middle value. The NumPy library provides a built-in function 'np.median()' that can execute this process. Median is not sensitive to outliers, offering a strong measure to central tendency for the angular differences. For instance, a median angular error of 25° indicates that 50% of the angular differences are less than 25°, whereas the remaining 50% are greater.

2.4.2. Temperature

The GFS model is the most effective in fitting the actual data, as it has the highest R^2 value of 0.774. However, in terms of error metrics, GFS does not perform so well. Specifically, the ICONEU model demonstrates very good performance, with a MAE of 1.337, a MSE of 3.527, and a RMSE of 1.878. The IFS model is the least effective for temperature prediction.

Table 4: The performance of all models on predicting the actual temperature values.

Models Performance on Predicting Temperature				
Model / Metric	**MAE (°C)**	**MSE (°C)^2**	**RMSE (°C)**	R^2
GFS	1.646	3.811	1.952	0.774
IFS	1.602	5.895	2.428	0.466
ICONEU	1.337	3.527	1.878	0.418

2.4.3. Wind Speed

Considering that Weatherspark does not provide wind direction measurements for wind speeds less than 1 mph (approximately 0.447 m/s), it is the author's speculation that speeds below this threshold are not frequently encountered. However, the actual wind speed data includes 15,214 observations that are less than or equal to 0.447 m/s, which represents 40.25% of the sample!

Additionally, a Google search indicated that wind speeds in Munich during this period were typically between 3 and 4 m/s, significantly higher than the average actual weather station measurement of 0.83 m/s.

For these reasons, the weather station operational functionality might need investigation, especially given the extreme weather phenomena that occurred in Munich during this period, which could have affected the sensors. Another option would be to compare the actual measurements with those from the nearest weather station to see if there are any surprising discrepancies.

As expected from the exploratory data analysis, all models perform worse than the baseline. If one must be chosen, it would be ICONEU. However, it is important to note that simply calculating a historical mean may yield more accurate values for wind speed than the predictions made by ICONEU.

Table 5: The performance of all models on predicting the actual wind speed values.

Models Performance on Predicting Wind Speed				
Model / Metric	**MAE (m/s)**	**MSE (m/s)^2**	**RMSE (m/s)**	R^2
GFS	1.481	3.433	1.852	-0.308
IFS	2.092	5.979	2.445	-0.771
ICONEU	1.282	2.230	1.493	-0.255

2.4.4. Wind Direction

Keeping in mind that the weatherspark does not calculate wind direction when the speed is < 1 mph the actual wind direction data might include anomalies here as well. However, the predictions of the wind directions are more accurate than the wind speed predictions.

ICONEU outperforms the other two models in all metrics for wind direction, having the lowest MAE (37.930), Med. AE (26.340), and RMSE (52.476). This suggests that ICONEU provides the most accurate predictions for wind direction.

Table 6: The performance of all models on predicting the actual wind direction values.

Models Performance on Predicting Wind Direction			
Model / Metric	**MAE (degrees)**	**Med. AE (degrees)**	**RMSE (degrees)**
GFS	54.181	39.934	70.629
IFS	39.857	27.925	55.722
ICONEU	37.930	26.340	52.476

2.4.5. Future Work and Technical Problems

Evaluation with Cosine Similarity:

It would be valuable to apply cosine similarity between the angles instead of the custom function for angular differences. This evaluation of the models using cosine similarity could then be compared with the corresponding evaluation results from the angular difference.

However, significant differences are not expected. Applying cosine similarity instead of angular differences will result in a new column that includes values ranging from -1 to 1, reducing the order of magnitude by 1 to 3 units. This would be useful if creating machine learning models to predict the target (the cosine similarity between the actual and forecasted directions) as it elegantly applies manual feature scaling to the target variable. Since the purpose of this assignment is to evaluate models rather than optimize them, cosine similarity is unlikely to add significant value to the process.

Problem in Loading Circular Histograms:

There is a minor issue with properly loading the circular histogram figure. However, since the figure displays correctly on screen, this is a minor concern. I plan to correct it in the future.

3. Author’s Thoughts on the Project

Approach to This Project as a Daily Work Project:

If this project were a core part of daily working tasks, it would be approached differently. During the initial period at work (1-2 months), the priority would be on producing results, regardless of the quality of the approach. That is, the focus would be exclusively to delivering correct results, without considering many other factors. As experience is gained, automations would be built for daily tasks, continuously improving time management, the quality of the approach and the organization of modular programming.

However, for this project, the aim was to combine everything at once, not only to increase the chances of producing accurate metrics, but also to present a well-organized project from every perspective. Inevitably, completing this within a week was challenging, even though a week might seem like sufficient time for an experienced individual who is already familiar with similar meteorological datasets.

Mistakes during the Project:

The most significant mistake was spending too much time trying to automate everything and create 100% dynamic functions. Since this project is not related to anything else and serves more as training for potential recruits, spending so much time in a fully automated modular environment might not have been very helpful and could be considered out of scope.

That said, I learnt a lot about meteorological data and possible metrics, refreshed my knowledge of trigonometry, solved math problems, organized my step-by-step thinking, and had the opportunity to present a professional project to anyone interested. Thus, focusing on 100% automation would only be in scope if this were a work project that someone would be involved with on a repeated basis.

References

[1] https://weatherspark.com

[2] https://www.wunderground.com

[3] https://www.timeanddate.com

[4] www.climameter.org