Temperature Prediction with Machine Learning

Joseph Sketl & Rohil Raina

DATA602

Professor: Mohammad Taghi Hajiaghayi

University of Maryland-College Park

December 12, 2023

1 Introduction

METAR is a standardized format for weather information through the International Civil Aviation Organization. It is one of the two most common forms of weather data used by pilots and is recognizable by pilots throughout the world.

A typical METAR format:

```
METAR KGGG 1617753Z AUTO 14021G26 3/4SM+ TSRA BR BKN008 OVC012CB
18/17 A2970 RMK PRESFR
```

This includes type of data (`METAR`), airport (`KGGG`), valid time (the 16th at 1753Z), modifier (`AUTO`; from an automated weather source), wind (from 140 degrees from true north at 21 knots gusts up to 26 knots), visibility (3/4 of a statute mile), precipitation (thunderstorms, rain showers, and mist), cloud cover (broken cloud cover at 800 feet, overcast at 1200 feet), air temperature and dew point (18 degrees and 17 degrees Celsius), altimeter pressure (29.70 Hg or inches of mercury), and any extra remarks (pressure falling rapidly).

Temperatures included in a METAR may differ slightly from a local area weather report due to some differences between surface temperatures and air temperatures, but the difference is negligible, and a METAR will provide accurate weather information for the area in question. METARs are a useful source of continuously updated data with METARs routinely posted at least hourly every day.

2 Data Collection

To initiate our project, we gathered METAR data specific to the College Park, Maryland area for the purpose of temperature prediction. College Park Airport, distinguished by the ICAO code KCGS, holds the distinction of being the world's oldest continuously operated airport. Initially, our plan was to procure data through the weather.gov weather API. However, we encountered a limitation – the API only provides historical data for up to two weeks, making it suboptimal for our project's needs.

In response to this limitation, we turned to the Iowa State University's Iowa Environmental Mesonet (IES), a comprehensive archive of automated airport weather observations globally. The IES enables the selection of specific METAR data within a date range, commencing from the first archived METAR at a chosen airport. For KCGS, METARs are available dating back to 1976, with a notable gap between 1981 and 2006 due to either non-archived data or the discontinuation of automated METAR broadcasts during that period. Our data collection spanned from May 31, 2006, to the most recent date available.

To facilitate this data collection, we implemented a straightforward web scraping function utilizing the beautiful soup package in Python. The IES provides a rich array of data, encompassing air temperature, dewpoint, relative humidity, heat index/wind chill, wind direction, wind speed, altimeter pressure in inches, mean sea level pressure, 1-hour precipitation, visibility, wind gust,

```
station,valid,tmpf,dwpf,relh,feel,drct,sped,alti,mslp,p01i,skyc1,skyc2,skyc3,wxcodes
CGS,2006-05-31 16:05,82.40,68.00,61.81,85.38,110.00,4.60,30.13,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 16:10,82.40,68.00,61.81,85.38,110.00,3.45,30.13,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 16:30,82.40,68.00,61.81,85.38,0.00,0.00,30.13,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 16:50,82.40,68.00,61.81,85.38,0.00,0.00,30.13,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 16:55,82.40,66.20,58.08,84.63,60.00,3.45,30.13,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 17:00,84.20,66.20,54.80,86.50,0.00,0.00,30.13,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 17:05,82.40,66.20,58.08,84.63,80.00,3.45,30.13,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 17:15,84.20,66.20,54.80,86.50,0.00,0.00,30.13,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 17:20,84.20,68.00,58.32,87.40,0.00,0.00,30.12,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 17:25,84.20,68.00,58.32,87.40,0.00,0.00,30.12,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 17:30,84.20,66.20,54.80,86.50,0.00,0.00,30.12,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 17:35,86.00,68.00,55.04,89.41,0.00,0.00,30.12,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 17:45,86.00,68.00,55.04,89.41,0.00,0.00,30.12,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 17:50,87.80,68.00,51.98,91.43,0.00,0.00,30.12,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 17:55,87.80,68.00,51.98,91.43,0.00,0.00,30.12,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 18:00,87.80,69.80,55.29,92.71,0.00,0.00,30.11,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 18:05,89.60,69.80,52.23,94.85,110.00,3.45,30.11,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 18:10,89.60,69.80,52.23,94.85,50.00,3.45,30.11,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 18:15,89.60,69.80,52.23,94.85,80.00,4.60,30.11,M,M,CLR,   ,   ,M,M
CGS,2006-05-31 18:20,89.60,69.80,52.23,94.85,100.00,3.45,30.11,M,M,CLR,   ,   ,M,M
```

Figure 1: HTML METAR data.

cloud coverage, cloud height, present weather codes, ice accretion, peak wind gust, peak wind time, snow depth, and the raw METAR.

During the feature selection process for our predictive model, we excluded certain variables, such as visibility, peak wind gust, peak wind time, ice accretion, and cloud height. These variables are pertinent to a pilot's considerations, involving aircraft operations and fuel levels, but lack direct relevance to atmospheric conditions conducive to temperature prediction.

The chosen features extracted from METAR data included air temperature (°F), dewpoint temperature (°F), relative humidity (%), heat index/wind chill (°F), wind direction, wind speed (MPH), altimeter (inches), mean sea level pressure (mb), 1-hour precipitation (inch), cloud coverage, snow depth, and weather codes. Each of these features was selected with specific considerations in mind. For instance, air temperature served as our target variable for prediction, while dewpoint represented the temperature at which air becomes saturated with water vapor. Relative humidity, inversely proportional to air temperature, was deemed significant for its potential role in temperature prediction.

Similarly, heat index/wind chill, reflecting the 'feel' of the temperature, incorporated the interplay of windspeed and air temperature. Wind speed and direction were considered due to their potential impact on air temperature dynamics. Altimeter and mean sea level pressure, measuring atmospheric pressure, were included as pressure and temperature exhibit a proportional relationship. Cloud coverage factored in due to its influence on temperature through precipitation or UV-ray reflection, potentially causing temperature fluctuations. Additionally, 1-hour precipitation, weather codes, and snow depth were included to explore their potential contributions to temperature prediction.

In summary, our selection of these features was grounded in the belief that they collectively contribute to the intricate task of temperature prediction.

```
df
```

valid	tmpf	dwpf	relh	feel	drct	sped	alti	p01i	skyc1	wxcodes	skyc1_score	wxcode_score
2006-05-31 16:05:00	82.4	68.0	61.81	85.38	110.0	4.60	30.13	NaN	NaN	NaN	0.0	0
2006-05-31 16:10:00	82.4	68.0	61.81	85.38	110.0	3.45	30.13	NaN	NaN	NaN	0.0	0
2006-05-31 16:30:00	82.4	68.0	61.81	85.38	0.0	0.00	30.13	NaN	NaN	NaN	0.0	0
2006-05-31 16:50:00	82.4	68.0	61.81	85.38	0.0	0.00	30.13	NaN	NaN	NaN	0.0	0
2006-05-31 16:55:00	82.4	66.2	58.08	84.63	60.0	3.45	30.13	NaN	NaN	NaN	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
2023-12-10 22:30:00	61.0	60.6	98.59	61.00	0.0	0.00	29.72	0.04	NaN	NaN	1.0	2
2023-12-10 22:50:00	60.6	60.0	97.89	60.60	0.0	0.00	29.71	0.12	NaN	NaN	1.0	2
2023-12-10 23:10:00	59.4	59.0	98.58	59.40	0.0	0.00	29.70	0.07	NaN	NaN	1.0	1
2023-12-10 23:30:00	59.7	59.4	98.94	59.70	310.0	3.45	29.68	0.09	NaN	NaN	1.0	2
2023-12-10 23:50:00	59.7	59.4	98.94	59.70	0.0	0.00	29.69	0.09	NaN	NaN	1.0	2

433997 rows × 12 columns

Figure 2: 'Df' Data Frame .

3 Data Cleaning

Initially, we aggregated the collected data into a consolidated dataframe, denoted as 'df', comprising over 400,000 rows. Missing values within the dataset were encoded as 'M.' These gaps primarily stemmed from non-existent data at the time, such as instances with no precipitation, resulting in the absence of corresponding weather codes. Notably, the most prevalent missing values were observed in the 1-hour precipitation (`p01i`), cloud coverage (`skyc1`), and weather code (`wxcodes`) columns.

Mean sea level pressure, a potentially valuable feature for temperature prediction, was unavailable for the entire dataset. Given the presence of the altimeter column, mean sea level pressure adjusted for reporting station elevation, the mean sea level pressure data was deemed non-essential and consequently omitted from the web scraping request. Snow depth, another initially considered feature, was also absent from the dataset and subsequently removed.

To address the remaining missing values, our strategy involved replacing them with the column median. This choice was motivated by several factors. The altimeter column, reflecting mean sea level pressure at the reporting site's elevation, tends to remain relatively constant at 29.92 inches of mercury. Therefore, imputing missing values with the column median was deemed suitable. For columns such as `p01i`, `skyc1`, `wxcodes`, `drct` (wind direction), and `sped` (wind speed), where values were absent, indicating their non-occurrence in the atmospheric conditions, would consequently become 0 with this strategy. Wind direction, influenced by prevailing wind currents shaped by the Coriolis effect, exhibited minimal variation for KCGS, predominantly originating from the North-Northwest direction.

Subsequently, for the `wxcodes` and `skyc1` columns, both text-based, we devised a numerical scoring system for each value. `Wxcodes` comprised text identifiers and standardized codes such as 'RA' for rain or 'SN' for snow. Similarly, `skyc1` included codes like 'CLR' for clear skies and 'OVC' for overcast condi-

```
wxcode_mapping = {
    'RA BR': 2,  # Rain, mist
    'RA': 2,     # Light rain
    'BR': 1,     # Mist (light precipitation)
    'SN': 3,     # Snow
    'TS': 4,     # Thunderstorm
    'DZ': 1,     # Drizzle (similar to light rain)
    'PL': 3,     # Ice pellets (similar to snow)
    'GR': 3,     # Hail (similar to snow)
    'FG': 1,     # Fog
    'FZ': 3,     # Freezing
    'UP': 1,     # Unknown Precipitation
    'DZ BR': 1,  # Combination of drizzle, mist
    'UP BR': 1,  # Unknown Precipitation, mist
    'TSRA': 3,   # Thunderstorm, rain
    'SHRA': 2,   # Rainshowers
    'SHRA FG': 2,# Rainshowers, fog
    'TSRA FG': 3,# Thunderstorms, rain, fog
    'TS DZ': 3,  # Thunderstorms, drizzle
    'FU': 1,     # Smoke
    'TSRA DZ': 3,# Thunderstorms, drizzle
    'DZ FG': 0,  # Drizzle, fog
    'HZ': 0,     # Haze
    'FZFG': 3,   # Freezing, fog
    'RA FG': 1,  # Rain, fog
    'SN BR': 3,  # Snow, mist
    'M': 0,      # Missing
```

Figure 3: WX Code Meanings.

tions. A dictionary was crafted to assign a numerical score to each key, removing plus and minus signs indicating precipitation intensity before mapping. The resulting columns were then populated with medians to replace missing values in numerical columns.

To streamline the dataset, we discarded columns such as 'station,' 'skyc1,' and 'wxcodes' that were deemed surplus to requirements. The 'valid' datetime column was renamed 'date' for enhanced readability. Further data processing involved grouping entries into semi-monthly periods, differentiating between the first half (e.g., Dec 1) and the second half (e.g., Dec 16) of each month. The data was subsequently averaged within these groups, yielding a new data frame reflecting these averages. This approach was adopted to reduce the total number of data points, in an effort to enhance computational efficiency.

4 Prediction Model

Moving on to the development of our temperature prediction model, it's crucial to acknowledge the complexity of weather prediction, which relies on intricate formulas to simulate atmospheric conditions. The project initially faced uncertainty regarding the feasibility of utilizing historical weather data for accurate temperature predictions. Nevertheless, contemporary meteorologists employ machine learning (ML) as an integral part of their weather analysis.

In our literature review, Fisher et al.'s paper, "Accurate long-term air temperature prediction with Machine Learning models and data reduction techniques," provided valuable insights into ML and artificial intelligence (AI) frameworks used for long-term air temperature prediction. The paper showcased successful applications of various algorithms, including linear regression, polynomial regression, lasso, AdaBoost, decision trees, and random forest.

For our project, we opted for the lasso, AdaBoost, and SARIMAX algorithms for temperature prediction, leveraging their proven success in climate prediction models.

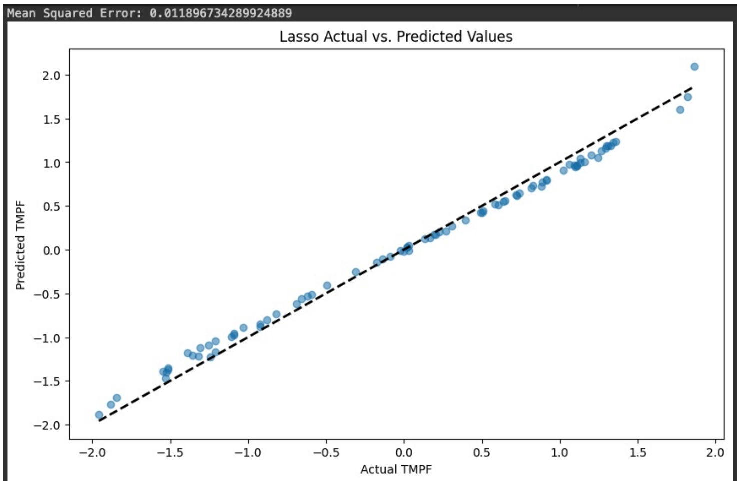
Moving forward, our feature selection process was driven by considering predictors deemed useful for temperature prediction, and all collected columns from the METAR data represented these features. To refine our choices, we calculated Pearson's correlation coefficients between each column and temperature. These coefficients, measuring linear correlation, served as a baseline for determining the relevance of features to temperature prediction.

The analysis revealed a strong positive correlation between temperature and dewpoint, as well as wind chill. Surprisingly, a positive correlation was found between temperature and relative humidity, contrary to the inversely proportional relationship mentioned earlier. Additionally, a relatively strong negative correlation was observed between temperature and altimeter pressure, wind speed, wind direction, and cloud coverage. Weather codes and 1-hour precipitation exhibited weaker correlations with temperature, likely due to limited precipitation recorded in the area. Despite this, we included them in our analysis, anticipating potential utility.

For the Lasso prediction, date features were extracted from the datetime column, and a subset was excluded for subsequent predictions. We utilized a standard scaler from the sklearn package to normalize input features before model training. The model was evaluated using the mean squared error (MSE) function, yielding an MSE of approximately 0.01. Predictions for the excluded values were then made, with the scaler reversed to restore temperatures to their original scale. The predicted temperatures, though slightly deviating from the actual values, were within 80% accuracy. Challenges in scaling inversion prompted exploration of alternative methods, acknowledging potential inaccuracies. Nevertheless, the predictions demonstrated reasonable success.

A similar test, maintaining the original individual data format instead of grouping into semi-monthly periods, was conducted to assess the impact on model accuracy. The model achieved an MSE of around 0.04, with predicted

Mean Squared Error: 0.011896734289924889



	Date	Predicted_TMPF
0	2023-10-01	50.90
1	2023-10-16	47.53
2	2023-11-01	37.69
3	2023-11-16	33.48
4	2023-12-01	34.97

Figure 4: Lasso Model.

temperatures also within 80% accuracy. While informative, this test was not included in the final notebook.

Subsequently, AdaBoost, a popular ensemble learning method, was employed for temperature prediction. AdaBoost, short for Adaptive Boosting, is an ensemble technique that combines the predictions of multiple weak learners to create a robust model. In this case, the AdaBoost Regressor was imported from the sklearn library, and the model's performance was assessed using the Mean Squared Error (MSE) metric, consistent with the Lasso model evaluation. The procedural steps mirrored those of the Lasso model, with the exception that values were not scaled this time. Notably, the MSE for AdaBoost was 0.76, significantly higher than the 0.01 obtained for the Lasso model.

Using the same excluded dataset, AdaBoost predictions yielded values of 61.92, 56.03, 47.61, 43.8, and 43.97. Intriguingly, these predictions were notably closer to the actual values, despite the higher MSE, indicating potential strengths of the AdaBoost algorithm in capturing underlying patterns.

To explore further, the model was tested on the non-semi-monthly dataset to assess its performance without temporal grouping. Surprisingly, the model exhibited diminished accuracy with an elevated MSE of 16.6. All predicted temperatures converged around 44.55.

The reasons behind the model's underperformance with the new dataset remain unclear. We suspect potential temporal patterns that are discernible when grouped into semi-monthly intervals may be less evident in the individual date dataset. Other contributing factors might include the smaller time frame for predictions compared to the semi-monthly format. The semi-monthly model

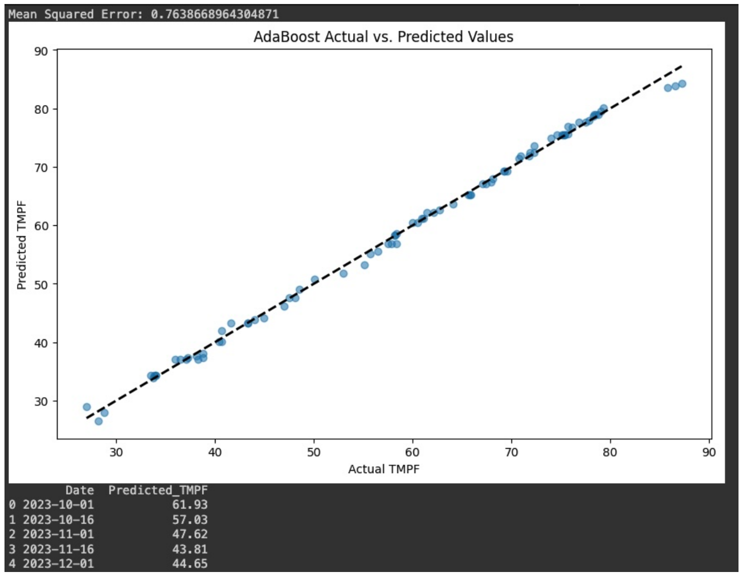


Figure 5: AdaBoost Model.

predicts temperatures for October, November, and December, while the individual date dataset focuses on 5 days in December. This test outcome was also excluded from the final notebook.

Finally, the SARIMAX model was employed for temperature prediction. SARIMAX, an acronym for Seasonal Autoregressive Integrated Moving Average with Exogenous Regressor, is a robust time series forecasting model that places a greater emphasis on the datetime column compared to the previously discussed models. Unlike the Lasso and AdaBoost models, the semi-monthly dataset used in earlier analyses could not be employed due to the absence of an inferred frequency recognized by Python. Consequently, a resampling of the original data frame 'df' was executed to obtain a dataset with a daily frequency.

SARIMAX utilizes exogenous, or external, data to predict a time series based on historical values, incorporating seasonality into its predictions, unlike its ARIMA counterpart. Given our objective of leveraging past temperature values to forecast future ones and considering the inherently seasonal nature of weather data, SARIMAX proved advantageous. This model acknowledges external variables, such as relative humidity, which can impact temperatures despite the overall seasonality of weather patterns. For instance, increased humidity, or moisture in the air, can elevate winter temperatures compared to drier winter conditions.

The SARIMAX model is represented by a formula involving parameters variables p, d, q, P, D, Q, and m in our Python implementation. The effectiveness of the model depends on the accurate selection of these parameters.

To optimize parameter selection, a systematic loop was implemented, eval-

```python
# Parameter tuning (grid search)
p = d = q = range(0, 2)
P = D = Q = range(0, 2)
m = [3, 6, 12]
param_grid = itertools.product(p, d, q, P, D, Q, m)
best_rmse = float('inf')
best_params = None
for params in param_grid:
    try:
        order = (params[0], params[1], params[2])
        seasonal_order = (params[3], params[4], params[5], params[6])

        # Create and fit the SARIMA model
        sarima_model = SARIMAX(train[target], order=order, seasonal_order=seasonal_order, enforce_stationarity=False)
        sarima_results = sarima_model.fit()
        # Make predictions on the test set
        sarima_predictions = sarima_results.get_forecast(steps=len(test))
        sarima_mean = sarima_predictions.predicted_mean

        # Evaluate the SARIMA model
        mse_sarima = mean_squared_error(test[target], sarima_mean)
        rmse_sarima = mse_sarima ** 0.5

        # Check if current parameters yield a better result
        if rmse_sarima < best_rmse:
            best_rmse = rmse_sarima
            best_params = params

    except Exception as e:
        print(f"Error for params {params}: {e}")
print(f"Best Parameters: {best_params}")
```

Figure 6: SARIMAX Parameter Loop.

uating various combinations of variables p, d, q, P, D, Q, and m, calculating the root mean squared error (RMSE) for each combination. The best parameter combination was determined by identifying the set with the lowest RMSE. After exploring multiple combinations, the parameters (1, 0, 0, 1, 1, 6) were selected as the most optimal configuration, as illustrated below.

Subsequently, the SARIMAX model was instantiated, and a forecast period of 12 days was designated. While the projected temperatures may not consistently match those anticipated by local weather stations, the model exhibits a commendable ability to capture variations in weather temperatures. This suggests the potential for a more comprehensive and nuanced prediction.

A notable illustration of this capability is observed in the transition from a temperature of 50 degrees on 12—16–2023 to 40 degrees on 12–17–2023. This shift aligns reasonably closely with the forecast for the upcoming week, albeit with a slight deviation of a day or so.

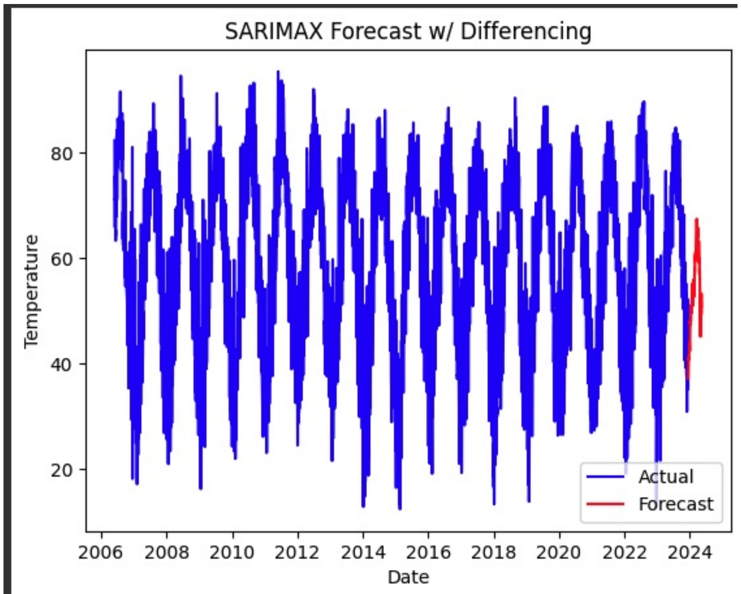
An identified issue arose in the SARIMAX model due to convergence problems, potentially stemming from non-stationary characteristics in the 'tmpf' column of the time series data. To address this, an Augmented Dickey-Fuller (ADF) Test was employed to inspect the stationarity of the temperature column. The initial test yielded a p-value above 0.05, indicating non-stationarity. To remedy this, the differencing function .diff() was applied, followed by a reevaluation of the ADF test, resulting in a p-value below 0.05 and confirming stationarity.

The model parameters were revisited through a parameter search loop, revealing that the optimal configuration for the SARIMAX model was (0, 0, 1, 1, 0, 0, 3). The SARIMAX model was rerun using the differenced temperatures. To facilitate visualization, the differencing was reversed using '.cumsum()'.

Upon inspecting the extended forecast, it became apparent that the new

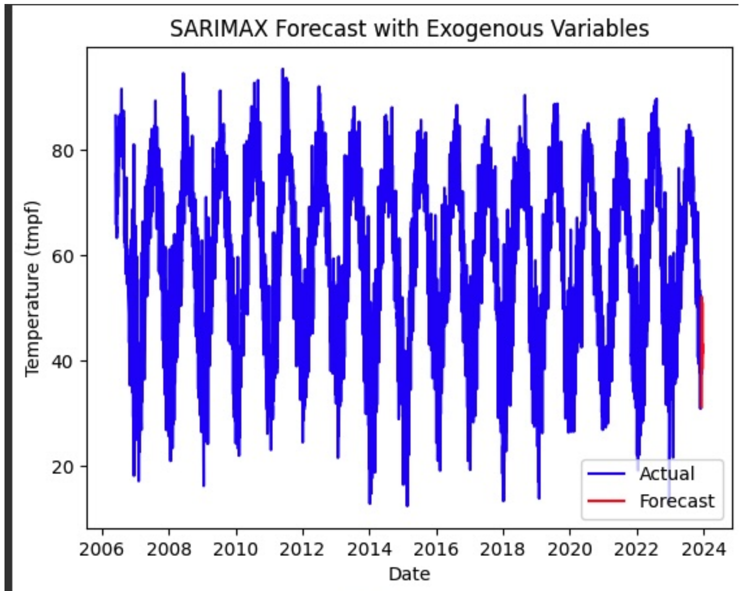
```
CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
Best Parameters: (1, 0, 0, 0, 1, 1, 6)
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            3     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  3.30942D+00    |proj g|=  9.69875D-02
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: No frequency information
was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: No frequency information
was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
 This problem is unconstrained.

At iterate    5    f=  3.22808D+00    |proj g|=  7.28709D-03

At iterate   10    f=  3.21641D+00    |proj g|=  2.21993D-03

At iterate   15    f=  3.21241D+00    |proj g|=  2.02130D-02

At iterate   20    f=  3.21191D+00    |proj g|=  2.76475D-04
```

Figure 7: Best Parameters Found



	date	predicted_tmpf
0	2023-12-13	37.10
1	2023-12-14	38.24
2	2023-12-15	39.16
3	2023-12-16	40.10
4	2023-12-17	40.91

Figure 8: SARIMAX Model w/ Differencing.



	date	predicted_tmpf
0	2023-12-11	31.13
1	2023-12-12	38.27
2	2023-12-13	39.57
3	2023-12-14	49.05
4	2023-12-15	52.05
5	2023-12-16	50.30
6	2023-12-17	40.47

Figure 9: SARIMAX Model.

predictions align more closely with historical data. However, they lack the variance observed in the original predictions. The values now exhibit a gradual increase and decrease without abrupt movements above or below the current values. While this revised model may be conceptually sound, the initial forecast appears more accurate in capturing the observed variance. Both forecast plots were included in the notebook for comparison.

5 Website Building

To showcase our research findings, we employed the Dash framework within Google Colab, creating an interactive web application. The application comprises three distinct pages, each dedicated to one of the predictive models we utilized: Lasso Regression, AdaBoost Regression, and SARIMAX Forecast. Let's delve into the key features of each page.

5.1 Lasso Regression Page

Upon navigating to the Lasso Regression page, users are greeted with a detailed plot illustrating the comparison between actual and predicted temperature values. The plot includes an ideal line for reference. Additionally, a table presents the predicted temperatures generated by the Lasso model, offering a more granular view of the forecasted data.

5.2 AdaBoost Regression Page

The AdaBoost Regression page follows a similar structure, providing a visual representation of actual versus predicted temperature values through a scatter plot. Accompanying the plot is a table showcasing the specific temperatures predicted by the AdaBoost model. Despite a higher Mean Squared Error (MSE) compared to Lasso, the model's performance is discussed, and potential reasons for varying accuracies are explored.

5.3 SARIMAX w/o Differencing Forecast Page

The SARIMAX w/o Differincing Forecast page introduces users to time series forecasting using the SARIMAX w/o Differencing model. A line plot captures the actual temperature values alongside the forecasted temperature. The SARIMAX predicted temperatures are also presented in tabular form, offering a comprehensive insight into the model's performance.

5.4 SARIMAX w/o Differencing Forecast Page

Similarly, a SARIMAX w/ Differencing page is included. The only difference from the previous page is the use of the differenced temperatures and forecasts.

5.5 Website Navigation

A convenient navigation bar is implemented across all pages, facilitating seamless transitions between Lasso Regression, AdaBoost Regression, and SARIMAX Forecast sections. Users can easily explore the specific models of interest.

5.6 Hosting in Google Colab

To make our web application accessible within the Colab environment, we utilized a script that accesses the virtual machine's proxy URL. A strategic line of JavaScript code opens the application in a new browser window, ensuring a user-friendly experience.

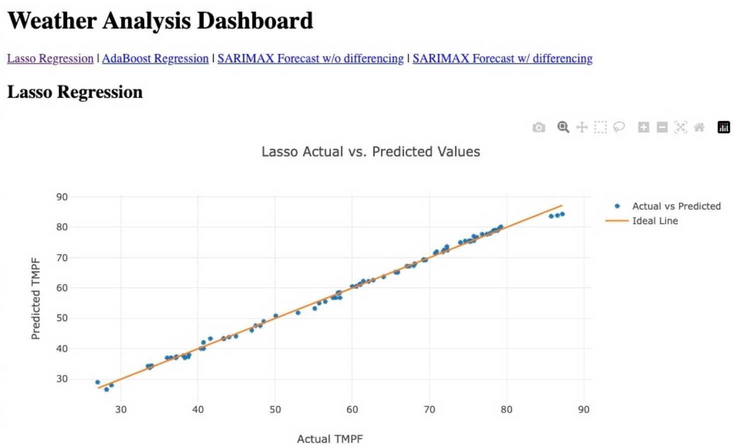


Figure 10: Website Dashboard.

In summary, our Dash web application serves as an interactive platform to present, analyze, and compare the outcomes of three distinct predictive models. The combination of visual plots and tabular data enhances the user's understanding of the models' forecasting capabilities. Accessible within the Colab environment, our web app provides a convenient means of sharing and discussing our project outcomes.

6 Conclusion

To conclude, this project experienced both successes and challenges. Going forward, it is crucial to investigate the issues related to the non-semi-monthly data, as preliminary assessments indicated variations in Pearson's correlation coefficients when data is kept in its individual date time format. Understanding how data relationships differ when grouped versus ungrouped is essential.

An identified issue involved an error in the initial data cleaning code, which failed to replace NaN values with column medians. The simple imputer from the sklearn library was used in the Lasso and AdaBoost models to deal with missing values at the time. Attempting to rectify the initial code negatively impacted model performance and was kept in its unchanged form. While the code remains unchanged for now, future improvements could involve replacing the initial data cleaning steps with the simple imputer for a more streamlined approach.

Moreover, exploring the SARIMAX forecast's effectiveness over an extended period would be insightful, especially considering the limitations encountered when forecasting into the next year. Additionally, enhancing the AdaBoost model by incorporating it as a booster for another classifier, such as a Random Forest algorithm, could lead to improved predictive capabilities.

Regarding the Dash app, while it serves its purpose, there is room for enhancement in terms of design and user-friendliness. Future iterations could focus on refining the app's aesthetics and functionality, making it more visually appealing and intuitive for users. This project marked our initial attempt into app development, and lessons learned could pave the way for a more polished and sophisticated application in the future.

References

[1] D. Fister, J. Pérez-Aracil, C. Peláez-Rodríguez, J. Del Ser, S. Salcedo-Sanz, *Accurate long-term air temperature prediction with Machine Learning models and data reduction techniques*, *Applied Soft Computing*, **Volume 136**, 2023, 110118, ISSN 1568-4946, `https://doi.org/10.1016/j.asoc.2023.110118`. `https://www.sciencedirect.com/science/article/pii/S1568494623001369`

[2] National Weather Service, *Why dewpoint is important: The difference between dewpoint and humidity*, `https://www.weather.gov/arx/why_dewpoint_vs_humidity#:~:text=The%20dew%20point%20is%20the,water%20in%20the%20gas%20form.`

[3] National Weather Service, *Relative Humidity*, `https://www.weather.gov/lmk/humidity`

[4] University of Melbourne, *Temperature and Relative Humidity - A Guide for Musicians*, `https://arts.unimelb.edu.au/__data/assets/pdf_file/0008/3922271/3.2-Temperature-and-relative-humidity.pdf`

[5] College Park Airport, `https://collegeparkairport.aero/`

[6] Iowa State University Mesonet, `https://mesonet.agron.iastate.edu/request/download.phtml`