Kavita Parekh
INST414
Data Science Lifecycle Option
27 April 2025

Improving Hyperlocal Temperature Forecasting Using Machine Learning
Git Repo Link: https://github.com/kparekh0424/sprints_inst414_project/tree/main

Executive Summary for Non-Technical Stakeholders:

Accurate weather forecasts are vital to daily decisions made by commuters, students, emergency responders, and public safety officials. However, many current forecasts are generalized and fail to provide hyperlocal insights that reflect conditions in specific neighborhoods or blocks. This project investigates how machine learning can improve short term, location specific temperature predictions by comparing forecasted and actual temperatures. Using free, available data from the National Weather Service[1] and National Oceanic and Atmospheric Administration's[2] METAR station at Newark Airport in New Jersey, I developed and evaluated two models. The models I used, Ridge Regression and Random Forest Regressor, aim to reduce the gap between forecasted and actual temperature values. Overall, the Ridge Regression model performs better for this task, and the framework can support future deployments from city emergency management departments like the New York City Emergency Management agency, helping them prepare for heatwaves, snow events, and more localized weather disruptions.

Problem Statement:

Weather forecasts play a critical role in daily decision making, yet they often lack accuracy and precision when applied to small geographic areas. However, some meteorological models are not always accurate. This is a significant problem for individuals and industries that rely on accurate weather information. This includes computers, students, outdoor workers, businesses, and even emergency responders where their work is dependent on weather conditions. Commuters who are traveling by car, train, or bus need to know the weather conditions in order to stay safe and be informed about delays, safety concerns, and other difficulties. Also, if a student is walking or biking to school, they have to be adequately prepared for the weather shifts. Inaccurate weather predictions impact agriculture industries, construction workers, outdoor event planning, and other logistics. Farmers depend on accurate weather forecasts for crop protection, pest control, and irrigation, yet generalized predictions are often

---

[1] National Weather Service. "Observation Equipment." National Weather Service, 2024, https://www.weather.gov/about/observation-equipment. Accessed 23 April. 2025.
[2] National Weather Service. (n.d.). API Web Services [Weather observations from METAR stations]. U.S. Department of Commerce. Retrieved April 23, 2025, from https://api.weather.gov/stations/KEWR/observations

wrong which can negatively affect farms and their crops. When forecasts are too broad, decision making can become uncertain which can lead to inefficiencies, lost revenue, and potential safety hazards.

Similarly, there are critical public safety implications that can occur due to poor hyperlocal weather forecasting. Emergency services rely on real time weather data to anticipate and respond to dangerous conditions like flash floods, snow storms, heat waves, or tornados. However, conventional forecasting models can often fail to provide real time insights that can improve disaster preparedness and response times. For example, with wildfire prone areas like California where wind direction, humidity, and temperature can vary significantly across the entire state, a failure to detect these variations in advance can lead to the difference between timely evacuation and disaster. Likewise, icy roads and unexpected storms can contribute to thousands of traffic accidents each year. Disasters like these can be mitigated with better predictive models that account for hyperlocal weather trends.

Therefore, traditional weather forecasts are too generalized and often lack high-resolution forecasting. I want to help industries optimize their planning and enhance public safety by providing better severe weather predictions for small-scale areas. With the advancement of IoT sensors, crowdsourced weather data, and machine learning, there is room and opportunity to improve location specific weather predictions that can benefit businesses, individuals, and even the government. By integrating real time data from traffic cameras, social media reports, satellites, and weather stations, predictive models can become significantly more precise and offer a better, more reliable forecast for individual neighborhoods. This could allow cities and towns to optimize road safety measures, emergency preparedness, and infrastructure planning based on dynamic, hyperlocal weather insights rather than relying on outdated, inaccurate regional forecasts. Addressing this issue is not only a matter of convenience but also a matter of economic impact, efficiency, and most importantly, public safety. With the right data driven approach, hyperlocal weather forecasting could have the potential to transform the way people interact and respond to the environment which would ultimately lead to smarter, safer communities and better informed decisions.

Proposed scope of work:

To improve the accuracy of immediate, hyperlocal weather forecasts, I aim to leverage a combination of machine learning, deep learning, and statistical predictive models that can use real time data sources. To create a final predictive model, I can combine multiple approaches and aggregate data from different sources to improve accuracy. I can evaluate the system using Mean Absolute Error and Root Mean Square Error for temperature and precipitation predictions, while also using precision and recall to assess the effectiveness of weather events, while also comparing traditional weather forecasting models to measure improvement in accuracy[3]. This

---

[3] Investopedia. "Predictive Modeling." Investopedia, 2024, https://www.investopedia.com/terms/p/predictive-modeling.asp. Accessed 23 April. 2025.

way, this project can deliver hyperlocal weather insights in a timely manner that goes beyond standard forecasting.

Methodology: Data Collection

To answer my research question, I collected weather forecast data from the National Weather Service (NWS) API and real-time weather observations from the METAR API from the National Oceanic and Atmospheric Administration (NOAA). The dataset will consist of forecasted temperature in Fahrenheit and wind speed in miles per hour from the NWS API, and actual temperature in Fahrenheit and wind speed in miles per hour from the METAR API. To make this more specific, I used New York City coordinates and Newark Airport coordinates to test the data in a specific geographic location. I came across numerous challenges when searching for my datasets and aligning my data. For example, a lot of weather APIs are not free. The Open Weather API that I mentioned in my Sprint 1 was not free when I looked to use it. When I tried to use other APIs, they were unresponsive and had limited data to work with. This was a challenge for me because it took awhile to find free APIs that I could pull from and use. In addition to this, when I finally found data that worked, the timestamps from the two sources did not align perfectly so I had to merge them properly and match forecasted data with the nearest available actual temperature. There were missing values that I had to deal with as well. After I got the data organized and in a merged table, I was able to make a data dictionary for the merged dataframe.

METAR raw data:

[4]:

	datetime	actual_temp_F	wind_speed_actual
0	2025-03-14 20:00:00+00:00	51.98	20.52
1	2025-03-14 19:00:00+00:00	51.98	16.56
2	2025-03-14 18:00:00+00:00	53.06	20.52
3	2025-03-14 17:00:00+00:00	53.06	7.56
4	2025-03-14 16:00:00+00:00	51.08	5.40

National Weather Service raw data:

[3]:

	datetime	forecasted_temp_F	wind_speed	short_forecast
0	2025-03-14 17:00:00-04:00	51	7 mph	Sunny
1	2025-03-14 18:00:00-04:00	51	7 mph	Mostly Clear
2	2025-03-14 19:00:00-04:00	50	7 mph	Partly Cloudy
3	2025-03-14 20:00:00-04:00	48	7 mph	Partly Cloudy
4	2025-03-14 21:00:00-04:00	46	6 mph	Mostly Cloudy

Methodology: Data Preparation

I merged the data from the two APIs using timestamps. Since exact timestamps did not always align, I used a merge-asof technique to match each forecasted hour with the nearest actual recorded value. Missing values were forward-filled and timezone information was removed to standardize datetime entries. To handle missing values and outliers, I used forward fill so that actual temperature values were filled into the missing records. In addition to this, I wanted to remove any outliers that could skew the data. To do this, I used the IQR method which removes any outliers in temp_difference by using statistical formulas to understand what the outliers were, and then removed them. This helps prevent skewed model results and will make the models more accurate.

For feature engineering, I wanted to create new features to improve model performance. I created the rolling mean of the temperature difference so that this variable can capture trends and smooth fluctuations in temperatures. I put the wind speed into bins which categorized them into meaningful groups. Wind can affect temperature so I wanted to have this variable to show that relationship. I also introduced a lag feature which will allow the models to recognize patterns from previous observations. Finally, I added an hour and day of the week to capture cyclical trends in weather variation.

Merged Data:

[5]:

	datetime	forecasted_temp_F	wind_speed	short_forecast	merge_key_x	actual_temp_F	wind_speed_actual	merge_key_y	temp_difference
0	2025-03-14 17:00:00	51	7 mph	Sunny	2025-03-14 17:00	53.06	7.56	2025-03-14 17:00	-2.06
1	2025-03-14 18:00:00	51	7 mph	Mostly Clear	2025-03-14 18:00	53.06	20.52	2025-03-14 18:00	-2.06
2	2025-03-14 19:00:00	50	7 mph	Partly Cloudy	2025-03-14 19:00	51.98	16.56	2025-03-14 19:00	-1.98
3	2025-03-14 20:00:00	48	7 mph	Partly Cloudy	2025-03-14 20:00	51.98	20.52	2025-03-14 20:00	-3.98
4	2025-03-14 21:00:00	46	6 mph	Mostly Cloudy	2025-03-14 21:00	51.98	20.52	2025-03-14 20:00	-5.98

Removing Outliers Data:

[7]:

	datetime	forecasted_temp_F	wind_speed	short_forecast	merge_key_x	actual_temp_F	wind_speed_actual	merge_key_y	temp_difference
0	2025-03-14 17:00:00	51	7 mph	Sunny	2025-03-14 17:00	53.06	7.56	2025-03-14 17:00	-2.06
1	2025-03-14 18:00:00	51	7 mph	Mostly Clear	2025-03-14 18:00	53.06	20.52	2025-03-14 18:00	-2.06
2	2025-03-14 19:00:00	50	7 mph	Partly Cloudy	2025-03-14 19:00	51.98	16.56	2025-03-14 19:00	-1.98
3	2025-03-14 20:00:00	48	7 mph	Partly Cloudy	2025-03-14 20:00	51.98	20.52	2025-03-14 20:00	-3.98
4	2025-03-14 21:00:00	46	6 mph	Mostly Cloudy	2025-03-14 21:00	51.98	20.52	2025-03-14 20:00	-5.98

Standard Scaler Data:

	datetime	forecasted_temp_F	wind_speed	short_forecast	merge_key_x	actual_temp_F	wind_speed_actual	merge_key_y	temp_difference
0	2025-03-14 17:00:00	-0.194950	-1.069609	Sunny	2025-03-14 17:00	8.774964	-11.904493	2025-03-14 17:00	-0.418191
1	2025-03-14 18:00:00	-0.194950	-1.069609	Mostly Clear	2025-03-14 18:00	8.774964	0.100469	2025-03-14 18:00	-0.418191
2	2025-03-14 19:00:00	-0.404689	-1.069609	Partly Cloudy	2025-03-14 19:00	-0.113961	-3.567714	2025-03-14 19:00	-0.401427
3	2025-03-14 20:00:00	-0.824167	-1.069609	Partly Cloudy	2025-03-14 20:00	-0.113961	0.100469	2025-03-14 20:00	-0.820532
4	2025-03-14 21:00:00	-1.243645	-1.252569	Mostly Cloudy	2025-03-14 21:00	-0.113961	0.100469	2025-03-14 20:00	-1.239637

Feature Engineering:

[11]:

	datetime	forecasted_temp_F	wind_speed	short_forecast	merge_key_x	actual_temp_F	wind_speed_actual	merge_key_y	temp_difference	hour	day_of_	rolling_temp_diff	wind_speed_category	temp_difference_lag1
33	2025-03-16 02:00:00	0.014789	0.028148	Areas Of Fog	2025-03-16 02:00	-0.113961	0.100469	2025-03-14 20:00	0.017678	2		-0.191875	Low	-0.191875
34	2025-03-16 03:00:00	0.014789	0.211107	Areas Of Fog	2025-03-16 03:00	-0.113961	0.100469	2025-03-14 20:00	0.017678	3		-0.052173	Low	0.017678
35	2025-03-16 04:00:00	0.224528	0.394067	Areas Of Fog	2025-03-16 04:00	-0.113961	0.100469	2025-03-14 20:00	0.227230	4		0.087528	Low	0.017678
36	2025-03-16 05:00:00	0.224528	0.577026	Areas Of Fog	2025-03-16 05:00	-0.113961	0.100469	2025-03-14 20:00	0.227230	5		0.157379	Low	0.227230
37	2025-03-16 06:00:00	0.224528	0.759986	Areas Of Fog	2025-03-16 06:00	-0.113961	0.100469	2025-03-14 20:00	0.227230	6		0.227230	Low	0.227230

Data Dictionary:

[6]:

	Variable Name	Description	Data Type	Source
0	datetime	Timestamp of the forecasted weather observation (hourly)	datetime	NWS API
1	forecasted_temp_F	Forecasted temperature in Fahrenheit from the NWS API	float	NWS API
2	wind_speed	Forecasted wind speed in mph from the NWS API	string	NWS API
3	short_forecast	Brief text description of forecasted weather conditions	string	NWS API
4	merge_key_x	Key used to merge forecast and actual observations (datetime format)	string	Computed Feature
5	actual_temp_F	Actual recorded temperature in Fahrenheit from METAR (Newark Airport)	float	METAR API (NOAA)
6	wind_speed_actual	Actual recorded wind speed in mph from METAR (Newark Airport)	float	METAR API (NOAA)
7	merge_key_y	Key used for merging METAR actual temperature readings	string	Computed Feature
8	temp_difference	Difference between forecasted and actual temperature (forecasted_temp_F - actual_temp_F)	float	Computed Feature

Methodology: Modeling Approaches and Implementation

For model implementation, I used two machine learning models. The first one I used was Ridge Regression. Ridge Regression is a linear regression model with L2 regularization that helps prevent overfitting by reducing the effect of large coefficients. Similarly, I used Random Forest Regressor as a machine learning model. This is a tree-based model that captures complex relationships by averaging multiple decision trees. Both models were trained using an 80-20 train test split and the features were standardized using StandardScaler to ensure that the numbers were stable. I also used cross validation methods to help evaluate how strong the models are. I used a 5-fold cross validation analysis to ensure that performance metrics were not solely dependent on a single train-test split. This will help assess model generalizability across different data sections and can help reduce any overfitting the model may cause.

Model Evaluation and Selection:

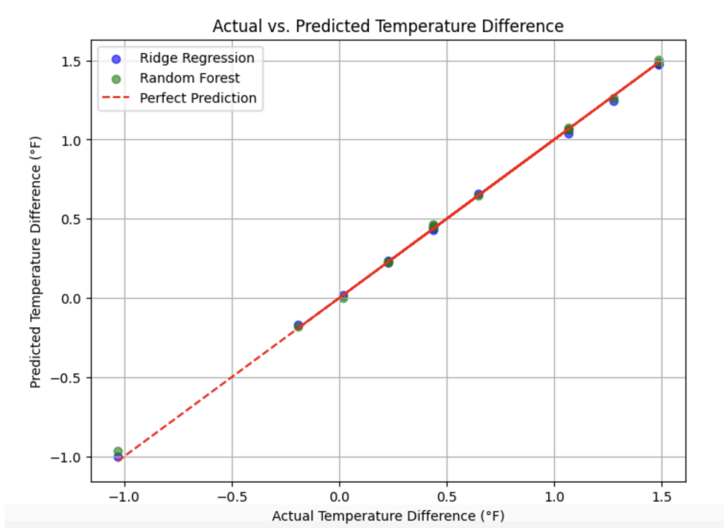
Two machine learning models, Ridge Regression and Random Forest Regressor, were developed to predict the temperature difference between forecasted and actual weather conditions. Ridge Regression is a linear model that uses L2 regularization to reduce overfitting by shrinking large coefficient values, making it ideal for datasets with multicollinearity. In contrast, Random Forest Regressor is an ensemble learning method that aggregates the outputs of multiple decision trees, offering robustness and the ability to capture nonlinear relationships in the data. Both models were trained on an 80-20 train-test split and evaluated using key performance metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), R^2 score, and 5-fold cross-validation. The Ridge Regression model demonstrated superior performance across most metrics, achieving lower MAE and higher R^2 scores compared to the Random Forest model. Additionally, Ridge Regression showed more stable results across different validation folds, suggesting better generalizability and reduced risk of overfitting.

The Mean Absolute Error (MAE) measures the average magnitude of errors in predictions. A lower value means better performance and a higher value means a worse performance. The Mean Squared Error (MSE) punishes larger errors more than the MAE which makes it useful for looking and analyzing large deviations. For the R^2 value, this indicates the proportion of variance explained by the model and a value closer to 1 means a better fit. The cross validation scores ensure that the model can be generalized across numerous training splits. This highlights how the Ridge Regression model has a score closer to 1 for the R^2 and is therefore a better fit than the Random Forest Model.

For the Ridge Regression model, there were a lot of strengths to choosing this particular machine learning model. For example, this mode prevents overfitting and has a higher accuracy than other models. However, it may not perform as well on non-linear relationships. With the Random Forest model, it is able to capture complex relationships and is easy to interpret results. While the Random Forest model is good with complex relationships, it has lower generalizability and requires more variables.

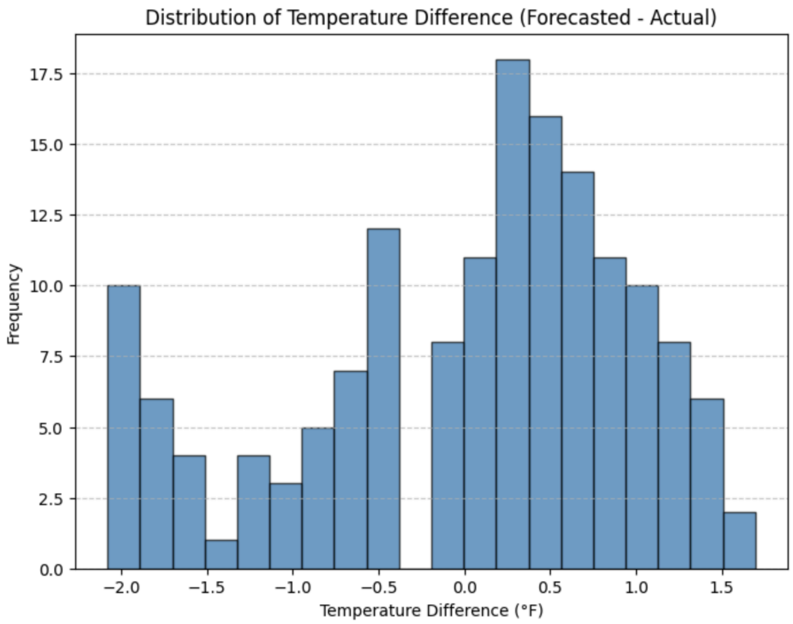
Evaluation Metrics:
Supporting visualizations included actual vs. predicted temperature difference scatter plots, error distribution histograms, and feature importance charts. These visualizations confirmed that forecasted temperature was the most significant predictor in the model. Collectively, the results highlight Ridge Regression as the more effective model in this context, while also validating the overall utility of machine learning for refining temperature forecasts.

```
Ridge Regression Model Performance:
Mean Absolute Error (MAE): 0.0123
Mean Squared Error (MSE): 0.0003
R-Squared Score (R2): 0.9994

Random Forest Model Performance:
Mean Absolute Error (MAE): 0.0142
Mean Squared Error (MSE): 0.0005
R-Squared Score (R2): 0.9989

Cross Validation Scores:
Ridge Regression: 0.9921
Random Forest: 0.9382
```

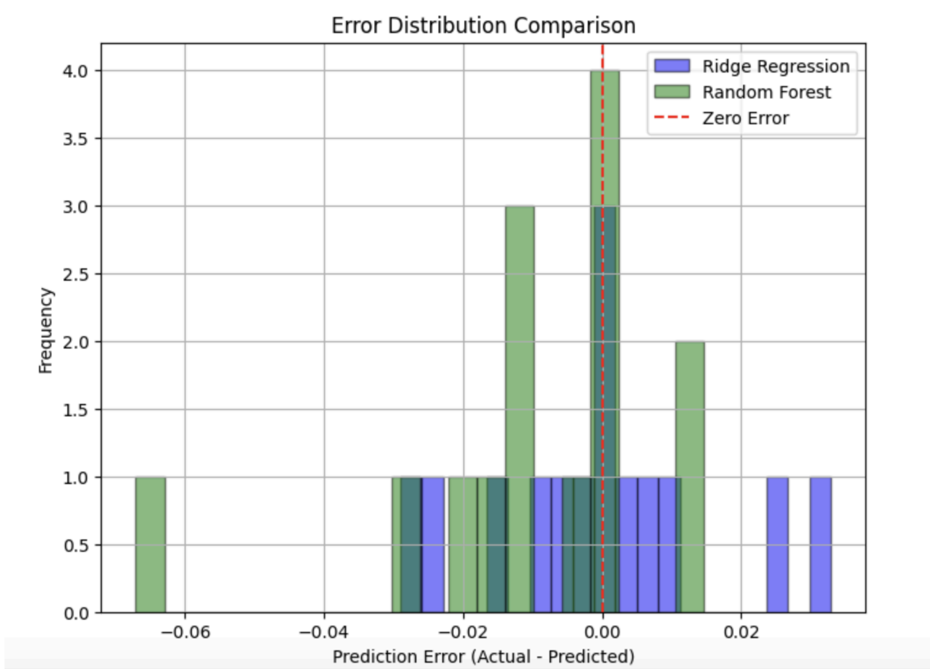


This scatter plot compares the actual temperature difference with the predicted temperature difference from the Ridge Regression and Random Forest models. The red dashed line represents a perfect prediction and the blue and green dots show the predictions from each model. Since most points are on the line, it shows that the model is performing well.

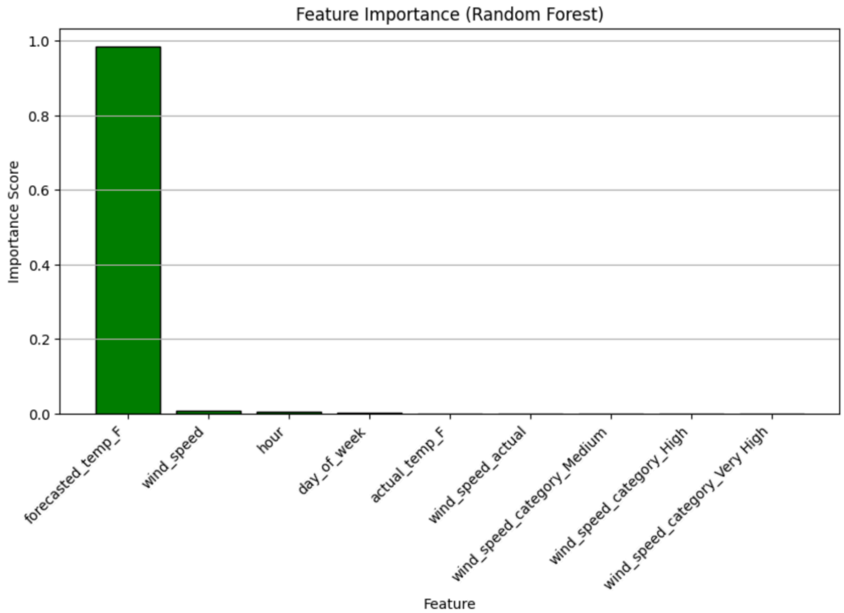
Visualizations and Insights:



This histogram shows the frequency distribution of temperature differences where it represents how often certain forecast errors occurred. The peaks and clusters show that forecast errors are not evenly distributed and instead occur within specific ranges. The two peaks show that temperature forecasts can either slightly overestimate or slightly underestimate actual temperatures.



This histogram shows the distribution of prediction errors for both models. The blue bars show the Ridge Regression errors and the green bars show the Random Forest errors. The red dashed line shows a perfect prediction. Since most errors are around 0, it highlights that the models are relatively accurate. However, the spread of the errors shows the variability in the predictions where more spread may lead to a less consistent model.



This represents the feature importance scores that the Random Forest model determined. It quantifies how much each feature contributes to predicting the temperature differences. It highlights how all of the variables do not have much impact on temperature difference, but the most dominant feature is the forecasted temperature. This makes sense since the model is heavily dependent on this variable.

Implementation Strategy and Business Value:

The implementation of this project holds value for public sector agencies and municipalities. The predictive models can be deployed as accurate, location specific adjustment tools to improve traditional forecasts. One proposed application is a dashboard system integrated within emergency management agencies like the New York City Emergency Management (NYCEM)[4]. With this dashboard, more agencies could receive timely updates about forecast errors and respond more accurately to temperature-sensitive operations.

Potential use cases include alerting public transit authorities about unexpected heat spikes or cold fronts, optimizing plowing or salting strategies for snow removal based on specific neighborhoods' locations, and notifying vulnerable populations during dangerous weather conditions. Businesses involved in construction, agriculture, outdoor events, and working that commute can leverage these insights to reduce downtime and plan logistics with more accuracy

---

[4] New York City Emergency Management. (n.d.). NYC Emergency Management. The City of New York. https://www.nyc.gov/site/em/index.page

and confidence. The improved accuracy and geographic specificity of the model can help improve operational efficiencies, economic savings, and enhance public safety outcomes.

Ethical Considerations and Limitations:

The ethical considerations of this project are critical for responsible implementation. One key consideration is the risk of geographic bias. Since the current model was trained using data from a single station, it may not generalize well to regions with different climates or fewer data sources. The model would need to be bigger and include multiple specific locations. This raises concerns about fairness and inclusivity where low-income or under monitored communities might not receive the most accurate predictions. Expanding the data sources to include diverse geographic and socioeconomic regions would help improve the model's equity to combat this limitation. Similarly, the model's performance is limited by the size and quality of available data. Since the amount of free weather data is limited, the training dataset is small and only covers a short time span. Consequently, the models may not fully capture seasonal variations, big natural disasters, or rare events. While the model focuses on temperature, future extensions can incorporate precipitation, natural disasters, or other relevant variables.

Any model that informs emergency response or public safety decisions must be interpretable and transparent. Stakeholders need to understand not only what the model predicts but also why. To improve interpretability, future iterations of this project could incorporate tools like SHAP-Supported Social Explanation Protocol to quantify feature influence on predictions or Contrastive Counterfactual Explanation Systems (CCES) to identify the minimal changes required to alter a prediction[5]. Ethical AI design principles must guide the development of the system and ensure that outputs are justifiable, auditable, honest, and serve the public interest.

[5] Jain, A. (n.d.). Introduction to SHAP values for machine learning interpretability. DataCamp. https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability