Kavita Parekh - Data Science Lifecycle Option - Sprint 1

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 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. l 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.

Review of similar solutions by other businesses or other stakeholders:

While hyperlocal weather prediction has attracted attention from various companies and stakeholders, gaps still exist. However, some similar solutions are:

- Private Technology Companies and Weather Apps: The Weather Company (IBM's weather channel)
    - IBM has used AI and machine learning into its Watson system to improve weather predictions. It uses data from airplane sensors, smartphones, IoT devices, and radar to make forecasts more precise. However, while this helps improve accuracy, it lacks any real time adaptation to hyperlocal variations and is limited. It serves large geographical areas rather than small towns and communities(The Weather Company).
    - Source:
        - The Weather Company. "The Weather Company." The Weather Company, 2024, https://www.weathercompany.com/. Accessed 14 Mar. 2025.
- AccuWeather
    - AccuWeather is an app that provides forecasts through its MinuteCast feature. It offers a breakdown of a standard forecast, yet still relies heavily on traditional meteorological data sources which can be inaccurate(AccuWeather).
    - Source:
        - AccuWeather. "Weather Matters." AccuWeather Corporate, 2024, https://corporate.accuweather.com/company/weather-matters/. Accessed 14 Mar. 2025.
- Smart City Initiatives
    - Some cities have started using sensor data from traffic cameras and weather stations to predict the climate. However, most cities lack the infrastructure to use a large-scale hyperlocal forecasting and these initiatives are not full proof yet(RikaSensor).
    - Source:
        - Rika Sensor. "How Weather Sensors Enhance Smart City Initiatives." Rika Sensor, 2024, https://www.rikasensor.com/a-news-how-weather-sensors-enhance-smart-city-initiatives.html. Accessed 14 Mar. 2025.
- The National Weather Service
    - This service is responsible for most public weather forecasts in the U.S. They utilize satellite imagery, Doppler radars, and weather balloons to generate

predictions. But, their models are often focused on large regional trends rather than localized forecasts(National Weather Service).

- Source:
    - National Weather Service. "Observation Equipment." National Weather Service, 2024, https://www.weather.gov/about/observation-equipment. Accessed 14 Mar. 2025.

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. The goal would be to create a system that enhances conventional forecasting methods by incorporating IoT sensor data, satellite imagery, traffic cameras, and social media reports. I could consider time series forecasting models. While researching, I came across LSTM's (Long Short-Term Memory Neural Networks), which is a deep learning model designed for time-series forecasting(Geeks For Geeks). This can learn more complex dependencies in weather data which would make it an effective choice for predicting microclimate changes over time. I could also use machine learning models for weather predictions. I could use gradient boosting machines (GBMs), which are a more powerful variation of decision trees and can capture complex patterns in weather data(Deepgram). There are also NLP's for social media and crowdsourced weather reports that I could use. These can analyze twitter posts, traffic cameras, and other broadcasts to detect accurate weather reports while they happen(IBM). 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.(Investopedia) This way, this project can deliver hyperlocal weather insights in a timely manner that goes beyond standard forecasting.

Sources:

Deepgram. "Gradient Boosting Machines." Deepgram AI Glossary, 2024, https://deepgram.com/ai-glossary/gradient-boosting-machines. Accessed 14 Mar. 2025.

GeeksforGeeks. "Deep Learning - Introduction to Long Short-Term Memory." GeeksforGeeks, 2024, https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/. Accessed 14 Mar. 2025.

IBM. "Natural Language Processing." IBM Think, 2024, https://www.ibm.com/think/topics/natural-language-processing. Accessed 14 Mar. 2025.

Investopedia. "Predictive Modeling." Investopedia, 2024, https://www.investopedia.com/terms/p/predictive-modeling.asp. Accessed 14 Mar. 2025.

Preliminary data sources, success metrics, and evaluation criteria. Data must be big enough for predictive analytics to be performed.

To create an effective model, I will need a large dataset that is diverse and accurate. The model will require data that captures historical weather patterns, real time environmental conditions, and external factors like social media reports. For success metrics and evaluation criteria, I could use Mean Absolute Error and Root Mean Square Error which can measure how much predicted temperature and precipitation values deviate from actual readings. I could also use precision and recall, and F1 scores to evaluate how well the model detects weather conditions. Finally, I can compare this model to models that are out there and adjust the model based on the gaps that I find. Using this information, I can make sure that the scalability is good and the model can work with different regions and that the model ensures accurate, timely updates.

Technical skills development plan (if needed)

Since this project uses multiple data sources and machine learning techniques, I can strengthen my python libraries skills, my API integration skills, my web scraping methods, and my geospatial data processing skills. I can learn how to process and clean weather data using Pandas and NumPy. I can use APIs like NOAA and Open Weather Maps. I can use Geopandas to visualize and analyze weather patterns based on locations. I can strengthen my time-series forecasting skills by working with LSTM networks, I can look into the gradient boosting machines to try and learn how to train models to recognize nonlinear weather trends, and I can look into NLPs for social media weather reports to extract insights from online. Overall, this project would have me taking a deep dive into machine learning and data engineering skills in order to provide an accurate model to solve my problem statement.

Some suggested resources to get you thinking

- https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/
- APIs:
    - https://openweathermap.org/api
    - https://www.wunderground.com/login?action=member-apikeys
- https://huggingface.co/docs/transformers/model_doc/bert

Data Science Lifecycle Option - Sprint 2

Research Question Update:

After exploring different data sources, I recognized that my original research question was not entirely feasible due to data limitations. However, the issue of inaccurate weather forecasting still remains relevant. To address this, I aim to investigate if machine learning models predict the temperature difference between forecasted and actual temperatures more accurately than a simple baseline approach?

Data Collection and Preprocessing:

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.

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

Data Cleaning and Feature Engineering:

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. All of my coding for this is in the jupyter notebook so that the data has been cleaned and prepped for modeling.

Model 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.

Evaluation Metrics:

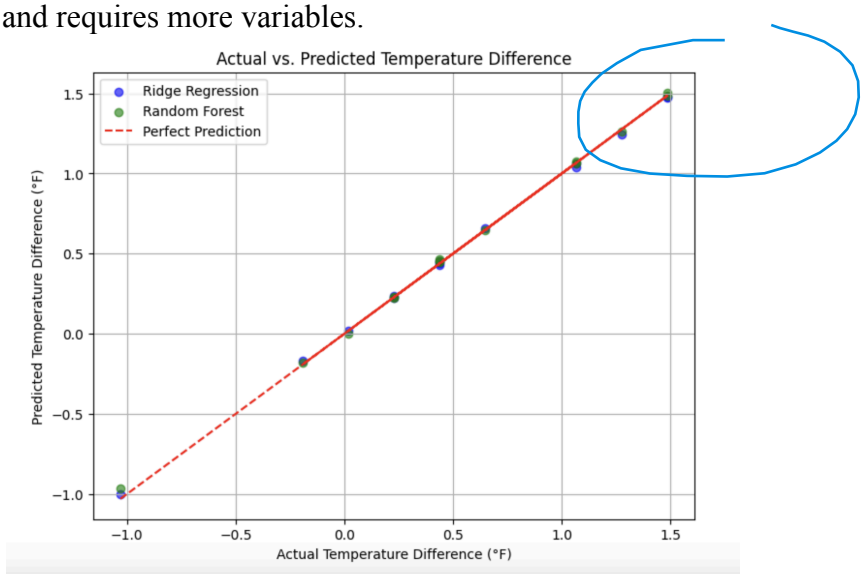
```
Ridge Regression Model Performance:
Mean Absolute Error (MAE): 0.0123
Mean Squared Error (MSE): 0.0003
R-Squared Score (R2): 0.9994

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
```

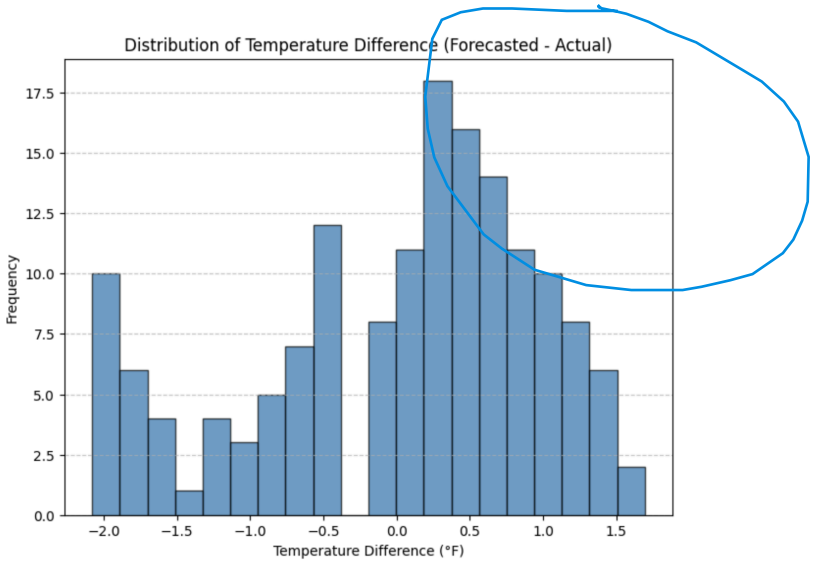
The Mean Absolute Error (MAE) measures the average magnitude of errors in predictions. A lower value means better performance and a higher value means a worse performance. The Mean Squared Error (MSE) punishes larger errors more than the MAE which makes it useful for looking and analyzing large deviations. 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.

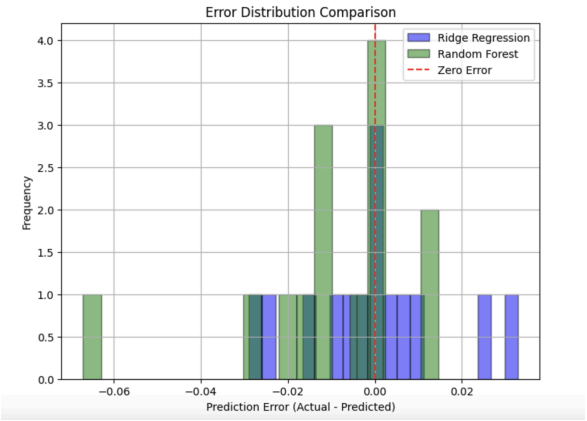


- This scatter plot compares the actual temperature difference with the predicted temperature difference from the Ridge Regression and Random Forest models. The red dashed line represents a perfect prediction and the blue and green dots show the predictions from each model. Since most points are on the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

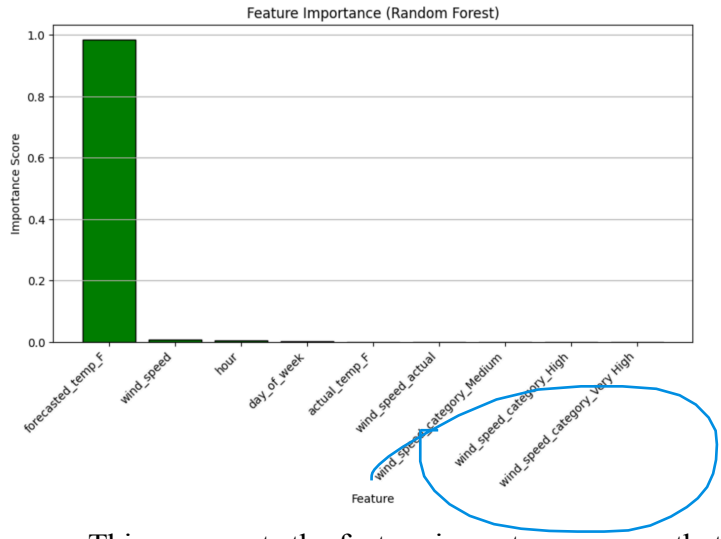
Visualizations and Insights:



- 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. However, in the future, I aim to introduce new relevant features to ensure that the model is not too simplistic.

Conclusion:

The Ridge Regression model was more accurate and had a more stable cross validation score than the Random Forest model did. This highlights how the Ridge Regression model was a simpler, regularized linear model that was effective. However, the dataset that I used was small, so there were some limitations on the results. For next steps, I aim to expand the dataset by incorporating additional real-time weather sources and apple feature selection techniques to refine specific variables. Also, I will experiment with deep learning models (LSTMs) to see if I can do time-series forecasting and get more accurate results and predictions for my research question. Machine learning can effectively predict temperature discrepancies, however I need to further refine the data analysis in order to enhance real-world usability and get more accurate weather predictions.