Project Writeup

Kenneth Wirjadisastra

June 8, 2025

Note: To access hyperlinks in this PDF, ***please download and open it locally.***
GitHub's built-in PDF preview may disable or hide clickable links.

Introduction

Weather forecasting models are widely used by the general public, usually to help us make decisions on what to wear or to help plan future activities. However, these models have a much larger impact than we realize. They can often influence agricultural practices, public policy, and resource management. Because of this, it is important that we deploy models that can accurately represent how the weather evolves. The challenge is that weather systems are complex and exhibit chaotic behavior, meaning that small changes in initial conditions result in large differences over time. This makes implementing such models difficult. The goal of our work here is to explore ways to implement these models and to address some of the issues we may encounter in the real world. We will simplify our model by considering only temperature. We will predict the next day's maximum temperature given the previous days' maximum temperatures around Denver, CO.

Our work will consist of four sub-projects. Project 1 consists of collecting the data and performing any necessary pre-processing. This is also where to construct the narrative of our work and motivate the research question we wish to answer. This sets us up for the next two projects. In Project 2, we perform exploratory data analysis (EDA). The goal of this project is to gain understand the structure of our data. Here, we will examine how the variables in our dataset relate to one another and use this information to help us make modeling decisions. Project 3 is where we begin the modeling process. Finally, we will address real world issues of the model from this project in Project 4.

Project 1: Data Wrangling

In this project, we focus on collecting the data and cleaning it for future work. The data was gathered from NOAA's Climate Data Online Search Tool. We have collected data from four stations in Colorado: Denver Centennial Airport, Denver Central Park, Denver International Airport, and Denver Water Department. Each dataset contains data beginning on January 1, 2005 and ending in early April 2025 when the data was collected (end dates vary by availability). The data and documentation can be downloaded directly from NOAA or from `Project1/raw_data`.

The first step in this project is to clean each dataset. In order to do so, we will use the `pandas` library from Python. The data was pretty well organized in its raw state. The only major issue was missing data and how to impute it. `DEN_weather_raw.csv` contained roughly one year's worth of missing values from 2005-2006, so we decided to remove the initial entries in this dataset.

We will remove any columns with a lot of missing data ($\mathcal{O}(10^3)$). For columns with roughly 100 points of missing data, we will impute using the previous value. This is because we don't expect to see much of a

difference in value from the previous day's observation. In Project 4, we will further explore the impacts of missing data and imputation methods using Monte-Carlo simulation. Each of the cleaned datasets is stored in `Project1/clean_data`.

The next step is to split each of the four clean datasets into training, validation, and testing datasets. We will allocate 70% of the data for training, 15% for validation, and the 15% for testing. The data are stored in `Project1/train_data`, `Project1/validation_data`, and `Project1/test_data` respectively. Although we will only be using data from Denver Centennial Airport for the modeling process, we will still perform EDA using all four datasets for EDA. Therefore, will merge the training datasets. We are primarily focused on the temporal aspect of our data, so we will merge the data by date.

Project 2: EDA

In this project, we perform EDA on the data. Most variables did not seem to correlate much with maximum temperature when considering a single lag. The only two variables with high correlation were the the previous day's minimum temperature (`TMIN`) and maximum temperature (`TMAX`).

We show the correlation matrix for data from Denver Centennial Airport below in Figure 1. The lack of correlation between `TARGET` and all other variables suggests that we can most likely remove the other variables, as they won't be as useful. Furthermore, `TMAX` and `TMIN` themselves are highly correlated, suggesting that including both for our model would be redundant. Therefore, our model will likely perform well using only historical data of maximum temperature. Figure 2 shows the lag 1 plot between `TARGET` and `TMIN`.

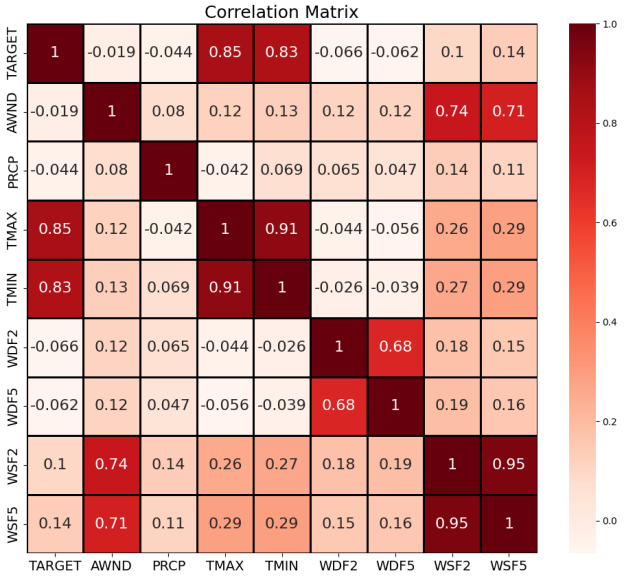


Figure 1: The plot shows the correlation matrix for the variables from the Denver Centennial Airport dataset. `TARGET` is the next day's maximum temperature, and all other variables come from the previous day. Only `TMIN` and `TMAX` have a correlation with `TARGET` whose magnitude is larger than 0.5.

There is a very strong positive relationship between the maximum temperature and its observed value from the previous day.

Project 3: Modeling

It is essential that we choose a model that handles sequential data well, given that we are working with time series data. The model we will be working with is a specialized recurrent neural network (RNN) known

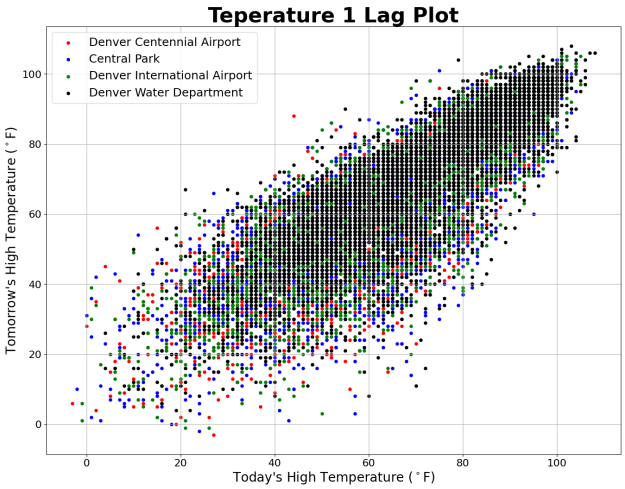


Figure 2: The plot shows the lag 1 plot for maximum temperature in degrees Fahrenheit. The x-axis shows the current day's maximum temperature (`TMAX`) and the y-axis shows the next day's maximum temperature (`TARGET`). There is a clear positive relationship between the two.

as long short term memory (LSTM). This model uses an input, output, and forget gate to select what information to retain and forget. By doing so, this model is able to better capture long term dependencies. To construct our model, we use `PyTorch`.

Our data will first be normalized using `MinMaxScaler` from `scikit-learn`. This normalizes our data between 0 and 1. Our model is a single layer LSTM with 16 hidden units. It will take a sequence of the previous days' maximum temperatures as input and output the next day's maximum temperature. The model is trained for 50 epochs using the MSE as the loss function with a learning rate of 10^{-3}. The training and validation loss are plotted in Figure 3.

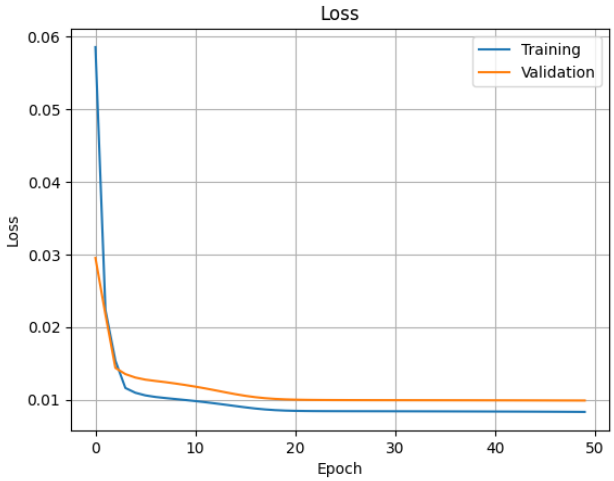
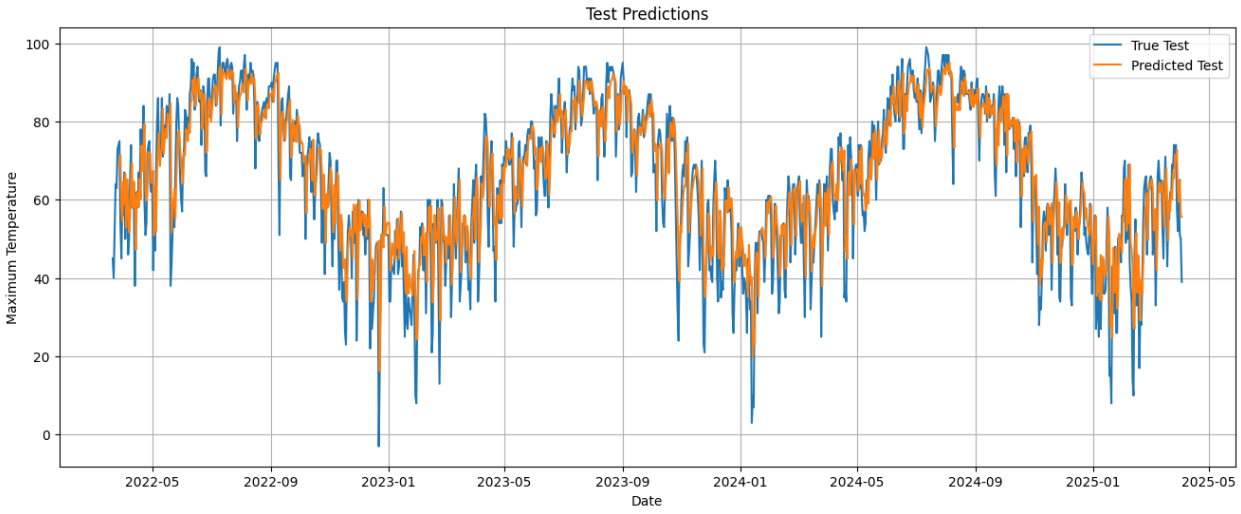


Figure 3: The plot shows the training and validation loss over 50 epochs. The loss converges after 20 epochs and does not seem to improve any further beyond that.

The validation loss is not much higher than the training loss, which is a good sign that we are not overfitting the model to the training data. Next we will test our model using the test dataset. The predictions and residual density plot are shown in Figure 4. While the model is able to capture the overall trend of the



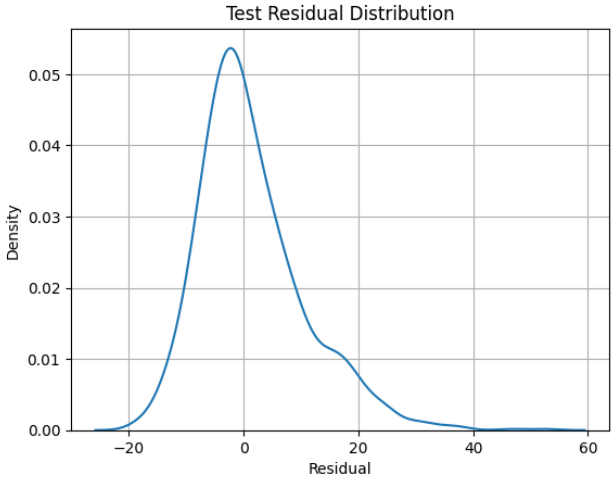


Figure 4: The test predictions and residual density are plotted in the figure. The model is able to find the general trend and shape of the temperature's evolution. However it is very conservative with it's predictions as it is unable to fully capture extremely high or extremely low temperatures. This is shown in the residual density plot. Our residuals are somewhat normally distributed around 0, but there is high variance.

data, it fails to capture outliers well. Our residual density plot confirms this. We have normally distributed residuals centered near 0, but high variance means that it is uncommon to find that the model predictions are off by 10°F or more. Predicting the temperature is already difficult enough with the complete data as shown in this example. However, the challenge of prediction will be intensified once we introduce missing data. This idea will be explored in the next project.

Project 4: Monte-Carlo Simulation

Under perfect conditions, we would have access to the full data. However, this is rarely ever the case. There are many reasons for not having access to complete data, such as routine maintenance of measuring devices or faulty equipment. In these scenarios, we have no choice but to resort to imputation if we want to be able to use the data. In this project, we explore how various amounts of missing data impact our model, with the main goal of finding imputation techniques to mitigate the error.

We will consider values of 10%, 25%, and 40% of missing data. We will also consider three different imputation methods: Kalman Smoother (KS), unconditional mean(UM), and linear interpolation (LI). This gives us a total of nine combinations to test. Given N data points (indexed from 0 to $N-1$), we will sample indices from the uniform distribution over integers in $[1,N-2]$ without replacement for each percentage of missing data. Then we will impute this data using the three techniques above. We will repeat this process five[1] times for each proportion of missing data and train the same model from Project 3. Each model will be initialized with the same weights to control for randomness.

We will compare the RMSE for each of the nine combinations. The mean and standard deviations are shown in Table 1. Although the sample size is small, it is still work taking a look at the statistics. Figure 5 shows the boxplot for the RMSE of the nine types of data. UM performs the worst at every level of missing data. This is likely because UM does not preserve any temporal structure, thus the model is learning from uninformative inputs. This leads to higher RMSE mean and variance as we increase the amount of missing data. Surprisingly, KS and LI perform quite similarly. However, their performance may differ if we introduce sequences of missing data (a week's worth of missing data to simulate routine maintenance). In this case LI will fit a straight line between the two end points, while KS may find something nonlinear. We can test for differences in RMSE between KS and LI using a t-test shown in Table 2. The null hypothesis is that the two means are the same, and we will reject the null hypothesis if we observe a p-value less than $\alpha = 0.05$.

Imputation Method	**KS**			**UM**			**LI**		
% Missing Data	**10%**	**25%**	**40%**	**10%**	**25%**	**40%**	**10%**	**25%**	**40%**
RMSE μ	9.478	9.499	9.661	9.707	10.357	11.086	9.495	9.462	9.583
RMSE σ	0.029	0.057	0.027	0.081	0.234	0.215	0.030	0.019	0.028

Table 1: RMSE mean and standard deviations are shown for each of the nine dataset configurations. Each dataset configuration was generated five times and used to train a model.

We do not reject the null hypothesis for 10% and 25% missing data. However, we do reject the null hypothesis and conclude that the two have different means for 40% missing data. Again, this test is not conclusive as there are a few issues with this design. The first, and most glaring issue is that we only complete five repetitions for each dataset configuration. This is nowhere near enough samples to draw any meaningful conclusions. Furthermore, the data generation process randomly samples indices without replacement. Because we chose low values of missing data, we are unlikely to get long sequences of missing data. Future studies should investigate how these two methods compare when imputing data with a sequence of missing data to simulate real world complications more closely.

[1]A sample size of five may be too small to draw any meaningful conclusions. Ideally, we would have access to more computational resources and perform more iterations.

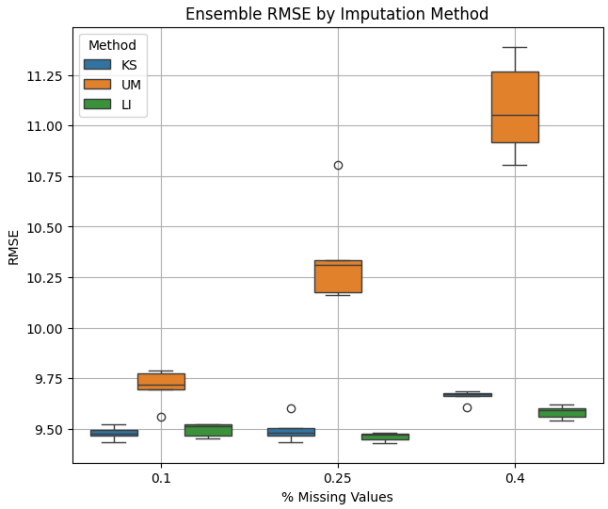


Figure 5: UM performs significantly worse than the other two imputation methods at every level of missing data. KS and LI perform pretty similarly at each level.

% Missing Data	t-score	p-value
10%	-0767	0.465
25%	1.229	0.275
40%	3.995	0.004

Table 2: T-test scores are shown comparing the differences between KS and LI at each level of missing data. The two are different for 40% missing data according to our test, but a larger sample is needed to verify the results.

Discussion

In general, weather models are much more complex. They often include more variables, make long term predictions, and sometimes include spatial dependencies. In our simple case, we were able to implement an LSTM model to predict tomorrow's high temperature given the previous week's daily highs. Although inaccurate at the upper and lower bounds, the model is able to capture the general trend and shape of the graph. The challenges of missing data and imputation methods were also explored. We have discovered that for sequential data, imputation methods which preserve the inherent temporal structure of the data generally works best. The differences between KS and LI seem to increase as we increase the amount of missing data, however nothing conclusive can be inferred from a sample size of five. More data and experimentation is needed to draw meaningful conclusions. Further developments include testing these imputation methods on data generated with sequences of missing data and incorporating spatial dependencies into the model.

Although seemingly inconsequential, weather models have ethical implications. As discussed in the introduction, these models have the power to influence many important things such as agriculture, resource management, and public policy. When dealing with such high stakes, it is important to rigorously test our models. Other issues include not being probabilistic or having an analytic solution. In its current state, our model outputs a single value. This is only the model's best guess, but it is usually interpreted as the definitive answer. The lack of an analytical form also makes it difficult to interpret how the model is making its predictions. The combination of these two factors makes it difficult for us to trust in the model fully.

These weather models play an important role in our daily lives, whether or not we realize it. Therefore, it is important that we perfect these models to accurately represent the truth. We have shown how we can model this system under the most simple circumstances, and looked into maintaining the model's accuracy in the presence of missing data. These methods are critical not only in weather modeling, but all problems requiring temporal data.