Weather: Predicting the Unpredictable

Whitney Anderson, Bryce Hepner, Rockford Lines, Carter Landon, Caelan Osman

September 27, 2022

Abstract

The goal of this paper is find a method for predicting the weather in Los Angeles. This includes the temperature, wind, precipitation, and CO_2 levels. Current prediction methods are only highly accurate for about a week. We use methods including Bayesian modeling, ARIMA models, particle filters, gradient/Newton boosted trees, and random forests. Our hypothesis is that by using these different statistical prediction methods we will produce results that match or exceed professional weather prediction methods.

1 Problem Statement and Motivation

Weather prediction is important for many community concerns including water management, traffic control, and natural disaster avoidance. For example, knowing that there will be seven days of consecutive rain allows a community to manage water more effectively, anticipate an increase in traffic accidents, and be ready to react to a flash flood. Unfortunately, current weather predictive models are accurate for only a few days because of high variance in weather. [3] The ability to predict weather data past a few days is of great interest to community development and improvement. Current weather prediction involves a variety of methods due to the chaos of the atmosphere. [7] Different combinations of global and local models are used which are trained on a variety of data. Our goal is to create one single model that provides accurate weather prediction.

2 Data

The atmospheric CO_2 data came from the Government Database Website [2]. The data source was created as part of a government project to reduce the CO_2 emissions by 80%. Regrettably, the only data we have is from the year 2018. Obviously a larger amount of data would provide our models with more accurate predictions. The few websites that we did find with sufficient amounts of data charged a high amount of money for the data. We were disappointed that although carbon dioxide is one of the more controversial issues in the world today, data collection for CO_2 data is either nonexistent or expensive.

Weather data was scraped from Weather Underground [1], which gave the temperature, dew point, humidity, wind speed, and other features. The data comes from The Weather Channel. According to Weather Underground's website, their information may be used for non-commercial use. We asked for the information directly and did not receive a response, nor was there a robot.txt, so we felt justified in scraping their website using `Selenium` so we could look at the ads as the page loaded. The missing information was filled in manually with data surrounding the hours that were missing.

3 Methods and Results

3.1 ARIMA/VARMAX

We wanted to use the ARIMA method to see if we could train on one month of weather data and predict the next several days or weeks of weather data. Because we're using vectors of temperature, dew point, humidity, wind speed, wind gust, pressure, precipitation, and CO_2, we chose `statsmodels`' implementation of VARMAX.

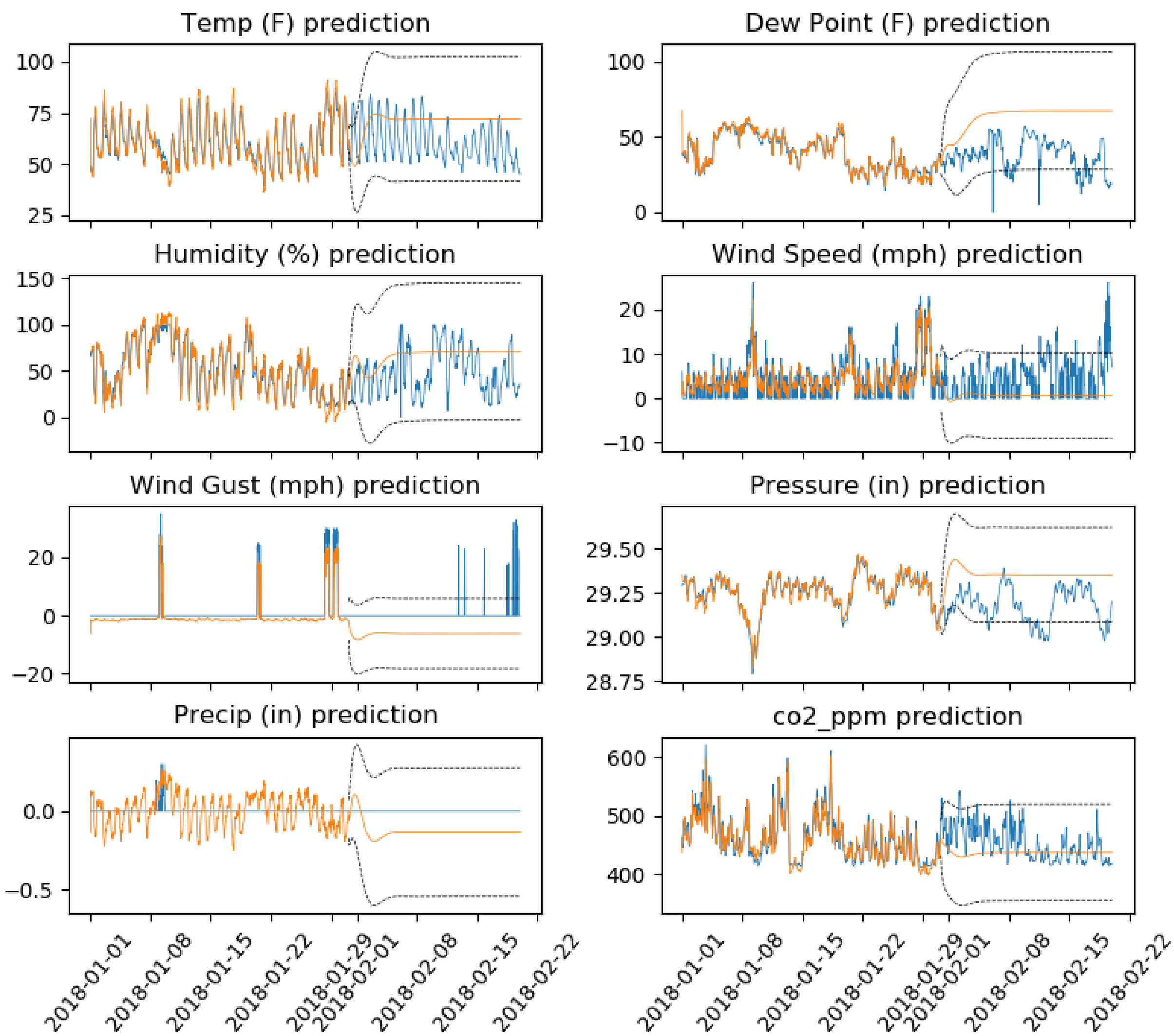

Figure 1: Training VARMAX on January Data to Predict February. Actual data in blue, predicted data in orange, and a 95% confidence interval is given by the dotted black lines.

When we trained on January and predicted February, it performed rather poorly, as shown in Figure 1. The temperature, dew point, humidity, wind speed, pressure, and CO_2 predictions were all very mediocre. Our VARMAX model didn't capture the ups and downs that the true data showed, and many of the predictions had 95% confidence intervals that were poorer at containing the actual data than a 95% level of confidence would suggest. The wind gust and precipitation predictions were nonsensical, both predicting negative values.

3.1.1 Differencing Data for ARIMA

Because the data does not seem stationary, we tried differencing the hourly data, so that we had the changes in temperature, humidity, etc. instead of the actual values to see if this would de-trend the data enough for it to fit with an ARIMA framework. We also took off the wind gust and precipitation columns because those seemed particularly problematic.

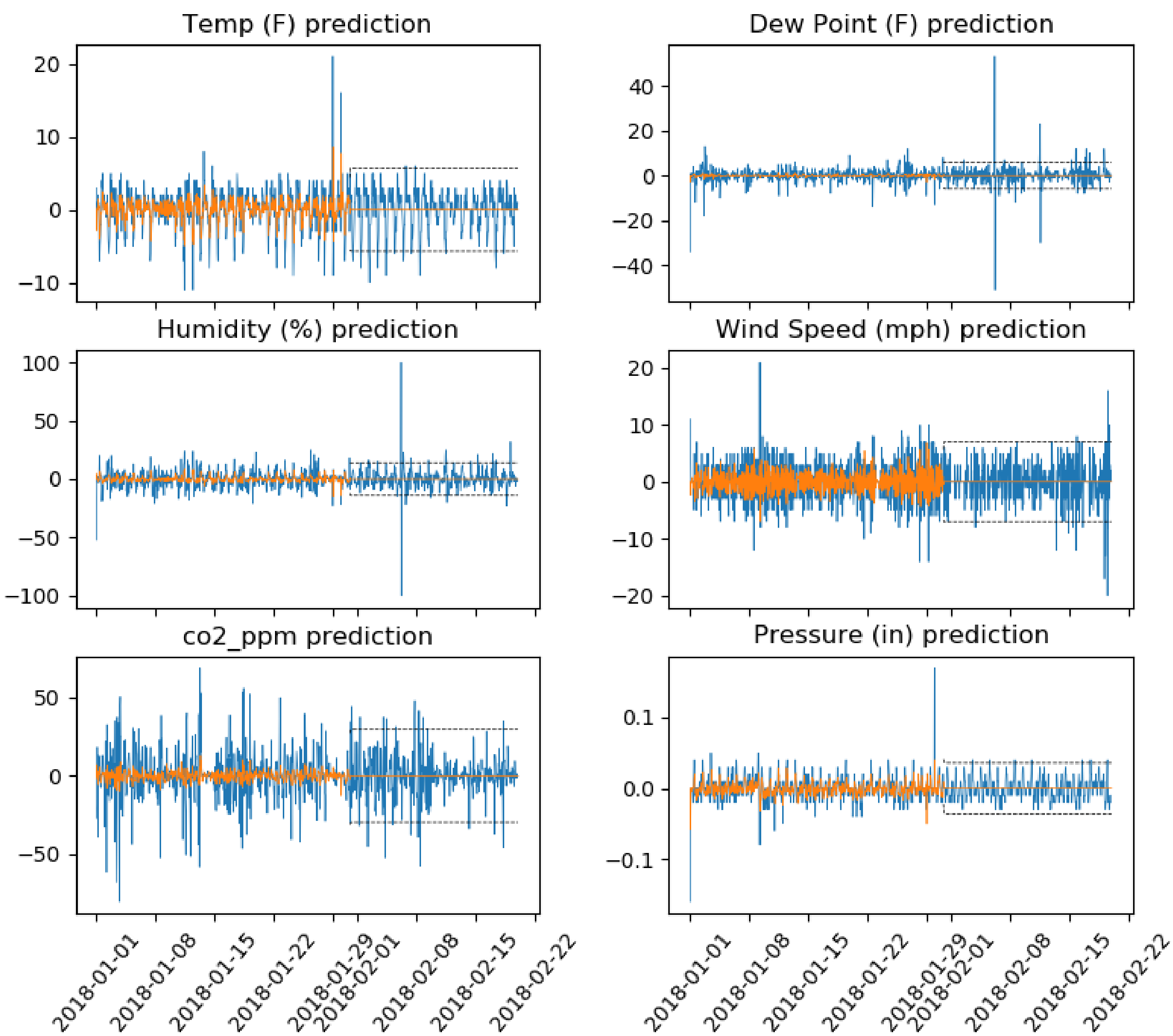

Figure 2: Training VARMAX on differenced January data to predict February. Actual data in blue, predicted data in orange, and a 95% confidence interval is given by the dotted black lines.

As we can see in fig. 2, the model predicts 0 for the hourly change in all of the features. The 95% confidence interval does a better job with the differenced data versus the original data, but the predictions are still not very useful and don't contain the daily fluctuations that most columns have. Having tried both of these, we decided that ARIMA and VARMAX would not be good methods for predicting the weather.

3.2 Trees

We decided to also use regression trees to predict the temperature. This was done by lagging the data and using it to predict future temperature values. Our data understandably has daily, and yearly seasonal patterns. However, since our goal was to predict at most the next week, we decided that it would only be necessary to lag the data by 24 hours to capture the daily seasonality.

3.2.1 Gradient-Boosted Trees

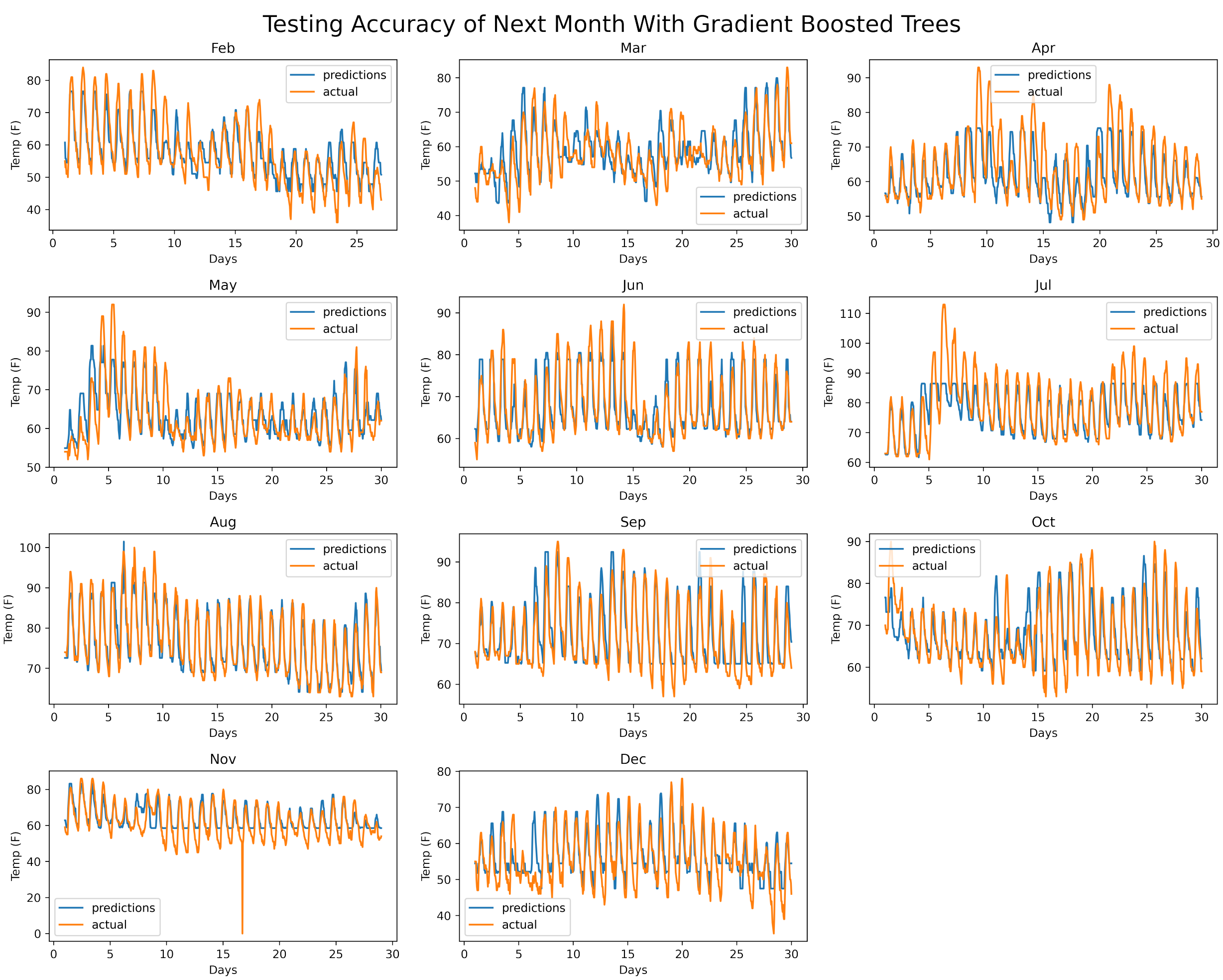

Figure 3: Using the lagged data, we predict the consecutive month. The consecutive months with the least change are predicted the most accurately.

Similarly to how the current weather prediction models use a variety of models, we used gradient boosted trees with ensembling in order to produce more accurate predictive methods. We trained on lagged data from each month to predict the temperature change of the following month. As shown in fig. 3, our model predicts future months with small amounts of change from day to day fairly accurately in the winter and summer but does poorly in consecutive months with high temperature change which occurs mostly in the spring and fall months.

3.2.2 Newton-Boosted Trees

Because of the quadratic approximation of the loss function, we hoped that a Newton-boosted tree would lead to better results than gradient-boosted trees. Additionally, we hoped this would also help overfitting the model. The data was prepared the same way as in section 3.2. We chose to use the `XGBoost` implementation to find the best hyperparameters of a predictive model. The results from `XGBoost` regression are very similar if not slightly better to the results from gradient-boosted regression, as shown in fig. 4.

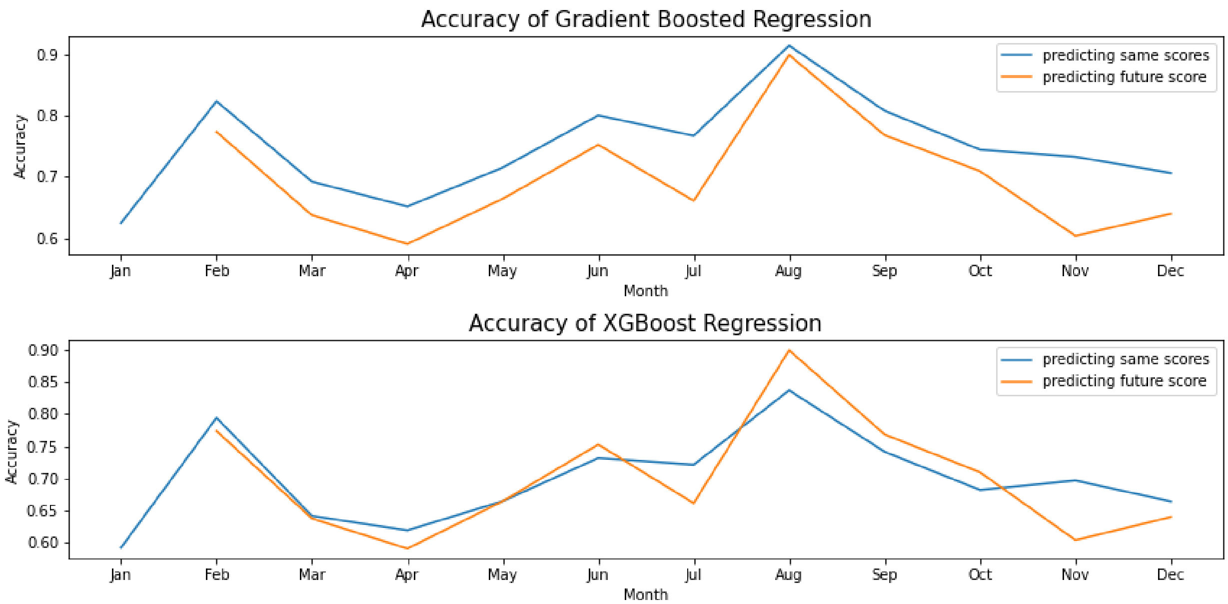

Figure 4: We compare the accuracy of testing on the current month and predicting the future month for both `XGBoost` and gradient-boosted regression.

3.2.3 Random Forest Regressor

In contrast with gradient boosting and Newton boosting, where each new tree is created to address existing error while fitting the model, trees for a random forest regressor are built randomly. This helps avoid over-fitting and is better equipped to deal with noisy data, which is needed for our weather data. Using 100 trees on subsets of our sample, we were able to observe what is shown in fig. 5.

Even though a random forest model tends to be generally better at working with noisy data, this method still had difficulty predicting temperatures beyond the rise and fall of individual days. Like the other methods, it especially struggles when predicting around a seasonal change, such as June to July. Though we included many features into the prediction, such as the previous month's temperature, dew point, and humidity, we failed to account here for the more general seasonal changes that we see year to year rather than just the changes of day to day.

3.3 Initial Analysis

Our decision tree models performed decently as expected, since data prediction with random forests or boosted trees generally produces reasonable output. On the other hand, our ARMA models

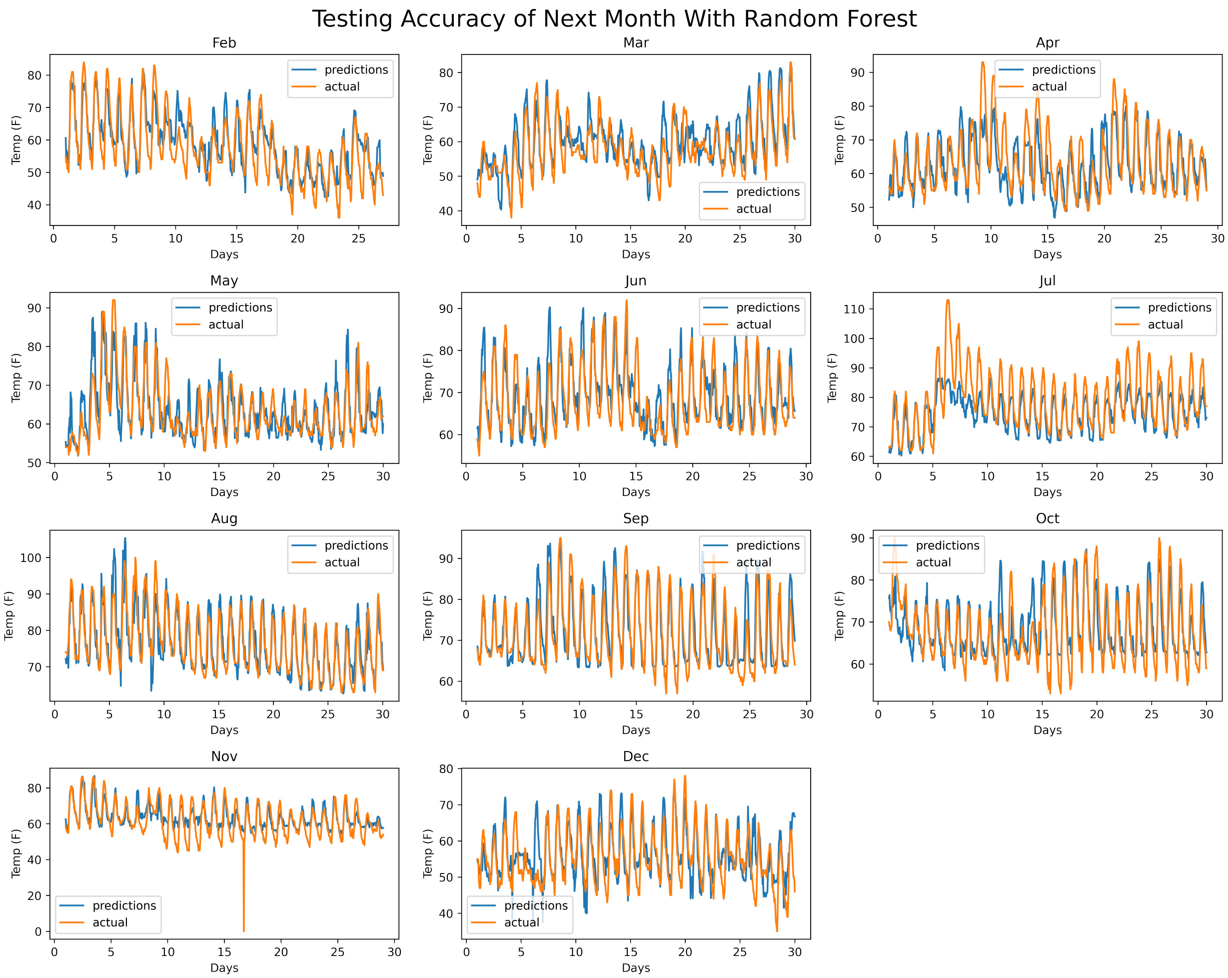

Figure 5: Similarly to the other decision tree methods, we use the lagged data to predict the next month.

performed rather poorly and gave mediocre predictions of the weather. The problem is that these (ARMA \ ARIMA) methods assume the time series to be of a certain form. More specifically, the time series Z_t is assumed to be linear in the previous states and in the previous error terms. It is known that every ARMA model can be written as a linear state space model. Furthermore, integrated (ARIMA) models will only make the resultant differenced time series stationary if the trend can be expressed as a polynomial. As [6] explains, "ARIMA forecasts are prone to large trend errors when there is a change in trend near the cutoff period and they fail to capture any seasonality." This can help explain why our ARIMA models in particular struggled making forecasts when it came to change in the seasons. Our initial models using VARMA and trees produced moderate to bad results. Some challenges to be considered were a lack of covariance stationary data and a small dataset.

3.3.1 Covariance Stationary Data

As shown in fig. 1, our multi-variate ARMA methods gave terrible to mediocre predictions. One challenge is that without our time series being covariance stationary we are not guaranteed that an

ARMA model can accurately predict weather. Unfortunately, our weather data does not appear to be covariance stationary. We mathematically examine how stationary the data is by looking at the mean and variance of the first 15 days compared to the second 15 days:

Feature	Mean 1-15	Mean 16-30	Variance 1-15	Variance 16-30
Temp (F)	61.122222	60.202778	71.868028	111.939268
Dew Point (F)	45.833333	33.10833	84.401114	92.988231
Humidity (%)	63.386111	41.525000	643.257188	430.478482
Wind Speed (mph)	3.363889	5.313889	14.766937	29.352453
Pressure (in)	29.237556	29.274583	0.012083	0.007575
CO_2 (ppm)	470.784889	452.999694	1596.721029	1205.692629

(1)

Seeing as this is only January 1-15 compared to January 16-30, the means and variances are different enough that we can assume that the series is not covariance stationary. We also looked at the means and variances of the first hundred days compared to the second hundred days, and the differences were more extreme. This contributed to our lack of adequate predictions with our ARMA models.

3.3.2 Lagging Monthly Data

One of the biggest challenges is that we only have data from 2018. Since weather is seasonal in 24-hour increments as well as yearly increments, this means that only using the data to predict on the next month produces results that don't take into account the changes in temperature per month. No matter what regression technique we used, whether it was gradient-boosted trees, XGBoost or random forests, predicting weather changes from one month to another on 24-hour lagged data isn't a good technique. As an example, using February's lagged data to predict March's temperature with gradient-boosted regression produced a coefficient of determination ($\mathbb{R}^2$ of .591, whereas using July's data to predict August's with gradient-boosted regression gives an $\mathbb{R}^2$ value of .899. Therefore, our model is only accurate for weather prediction of consecutive months with little temperature change. This is a deficiency of the lack of yearly data. Future research will include yearly data prediction.

3.4 Prophet

The most successful model used to predict the weather was `Prophet`. Before we present our findings it is important to know how this model works so we can analyze the outcome.

3.4.1 Mathematical Preliminaries

`Prophet` was designed by Facebook to provide a better model for time series forecasting. Facebook realized that the problem with most time series predictions is that unless the time series data has low variance, data prediction models require a lot of parameter adjusting [6]. Here, they use a decomposable time series model where $g(t)$ represents the non-periodic changes (growth), $s(t)$ represents the periods (seasonality), $h(t)$ is the effects of holidays and ϵ_t is the error term (assumed to be normally distributed). The different terms are simply summed and the model becomes:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t. \tag{2}$$

While this is generally used for business forecasting, we hypothesized that the seasonality effects the researchers at Facebook built into their modeling software, both short term and long term, would lead to a more accurate prediction model for weather. These same researchers further realized, that a nonlinear model was needed in order to better predict nonlinear behavior. This is accomplished particularly well using a logistic regression model for growth.

$$g(t) = \frac{C(t)}{1 + \exp(-k(t)(t - m))} \tag{3}$$

Where $C(t)$ is the (nonconstant) carrying capacity, $k(t)$ is the (nonconstant) growth rate, and m is an offset parameter (usually the mean of the data). This change was implemented after noticing that modeling growth in terms of population in business was similar to natural (logistic) growth. The points where the growth rate changes are called change points s_j and with them there is a corresponding rate of adjustment δ_j that is the change in the growth rate. For example, after one change point the growth rate would be $k(t)+\delta_1$. and after two change points the growth rate would be $k(t) + \delta_1 + \delta_2$. Additional hyperparameters usually indicates longer model set-up times for the analyst. Fortunately, the change points be chosen automatically using Bayesian methods and the adjustment rates can be found using standard optimization methods.

When making an actual forecast, [6] explains that future predictions are chosen so that the frequency of change points is the same as the training data. Uncertainty in the forecast will have the same average magnitude and frequency of growth rate changes that was seen in the training data.

The final piece that makes this method so valuable for weather forecasting is the seasonality component. Here, seasonality is approximated using a Fourier series. This choice was made because Fourier series are incredibly well suited for capturing periodic events (like those seen in weather data). The additional parameters (coefficients on the sine and cosine functions as well as the period) can be chosen by examining the AIC of a given model.

The only significant difference between the business forecast use and our weather prediction model is inclusion of $h(t)$ or the holiday term. It seems unlikely that a cold holiday such as Christmas would be a useful addition to predict cold weather. With all this in mind, it seems that `Prophet` is very capable of reaching our prediction goals.

3.4.2 Results

Figure 6 shows the results given by our prophet model. The prediction seems to follow the overall trend of the data. Almost all data points are within the confidence interval and many of them lie directly on the prediction line. When the forecasting actually begins in February the prediction does a great job of capturing the high temperatures of the day, but the model does have some trouble capturing the low temperatures.

4 Final Analysis

In contrast to the forests and ARMA methods, Facebook's `Prophet` prediction model performed very accurately. This was our most successful model in weather prediction.

In terms of our time series, a carrying capacity could be the minimum or the maximum that the weather could be in Los Angeles (determined by historical records for that change point region). The carrying capacity would then change based on the (daily, weekly, yearly) season that LA is in. The growth rate would change for similar reasons, we expect that the rate of change of temperature is likely different midday than at dusk. This further explains why our prediction had a hard time capturing some of the extremes seen in the testing data. This might be mitigated by training on

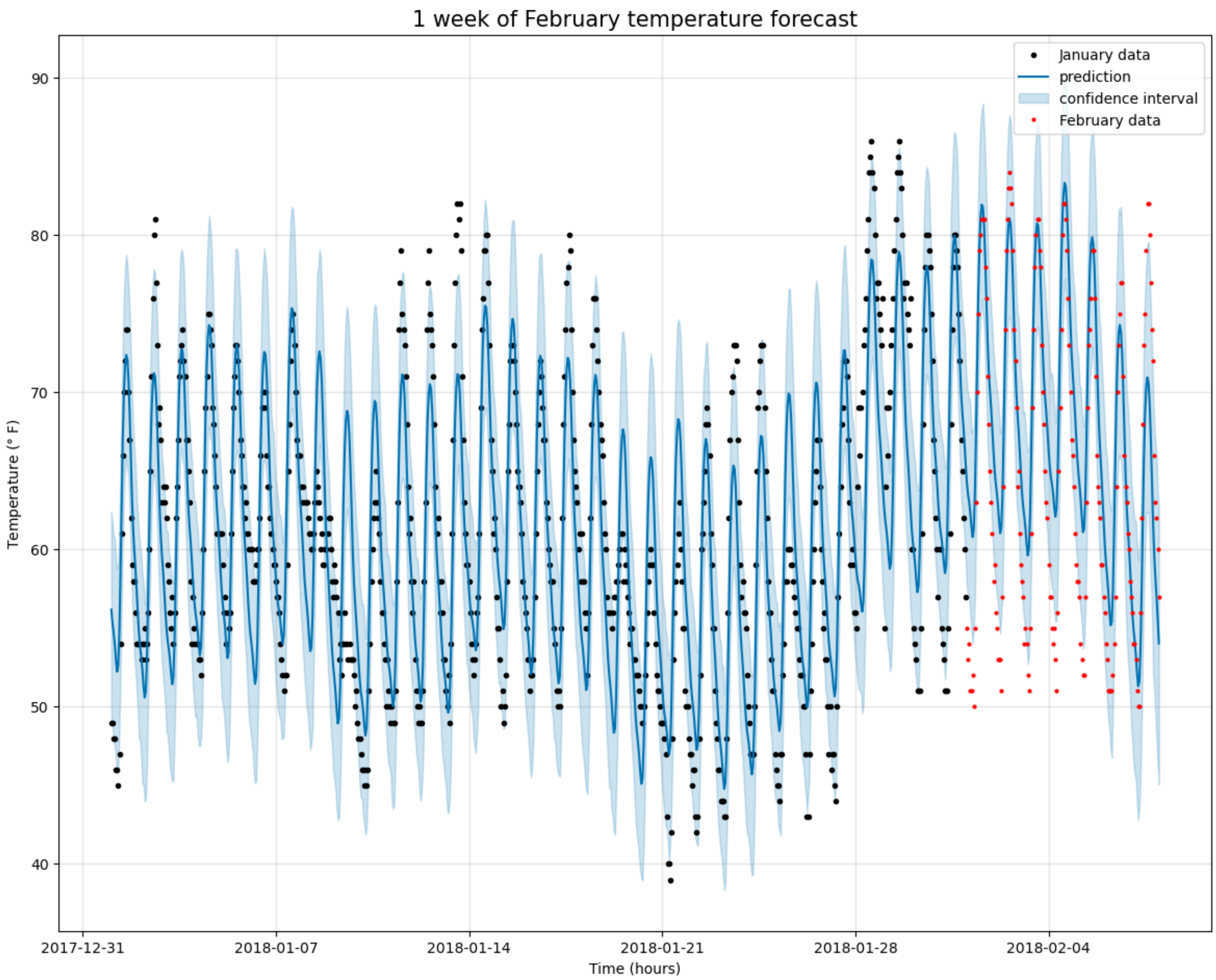

Figure 6: The black dots are the hourly temperature data in Los Angeles for the entire month of January. Using `Prophet` we predicted the future data (blue line) as well as a confidence interval (blue shaded region) that closely matched the training data in January. We then used this to forecast the first week of February. The actual February data is given by red points. As we can see for the first two to three days, the prediction does a pretty good job of forecasting the temperature.

other passed relevant data. The difficulty is in most places, the weather in August has little to no impact on the weather in the following January or February, in other words, we expect the frequencies and magnitudes of change points and growth rates (respectively) to be different. This might be handled by adding a decay rate α so that when forecasting, more distance frequencies and magnitudes will have less influence than more recent values. This would hopefully allow training on much more data without the worry of under fitting.

Putting all these things together makes a well suited method for approximating a weather time series that has variable growth, trends, and seasonality.

5 Ethical Considerations

For the most part, we see little ethical concern with this paper. The data collected and used was public and impersonal, as was the analysis. Privacy is not an issue in regards to weather data.

That said, results of this paper could give rise to concerns if misused or misinterpreted. One potential use of these methods could be to make climate change predictions based on green house gas and weather data collected over the past century. The concern is that misuse of these methods could cause people to underreact (or overreact) to an incorrect model. Furthermore, the limitations of the data made it so we were not able to successfully parse out cyclical trends such as daily and yearly cycles to make clean, accurate predictions in the near future that account for those. Thus, the inaccurate or incomplete weather forecasts of our model could be dangerous, as they don't account for heat waves, blizzards, precipitation, or other safety concerns. It could affect economies and lives, as farmers need to plan to take care of their crops, and shipping companies and travellers need to account for climate conditions.

We can mitigate the potential abuse of the results of our paper by being very clear and deliberate about the scope and purpose of our project. By being communicative about project limitations, we can avoid others placing false confidence the in the results of our paper. We keep our algorithms open to noise and correction, and we can emphasize that we are only capable of providing an estimate, rather than an exact prediction.

6 Conclusion

We have determined that data prediction using modern time-series methods such as VARMA, decision trees, and `Prophet` is definitely something that can be improved through a thorough hyperparameters search and through high mathematical prowess. We also hope to use data from longer periods of time in future research. Our models were very limited by only having data from one year, 2018. We currently have a low understanding of weather trends and yet with simple statistical prediction models we were able to reasonably predict future weather forecasts. Our `Prophet` model predicted especially well with little parameter adjustment. Obviously with more fine tuning and expertise in weather modelling, these methods could be greatly improved. Our `Prophet` model showed accuracy of about a week and an eventual goal is to be able to forecast a month's temperature in advance. We would hope that weather forecasting companies around the world would experiment more with time series prediction models in order to have better data and help their societies to be more successful and safer.

References

[1] Burbank, ca weather history. https://www.wunderground.com/history/daily/us/ca/burbank/KBUR, December 2018.

[2] Amount of co2 generated by ladwp. https://catalog.data.gov/dataset/amount-of-co2-generated-by-ladwp, November 2020.

[3] Britannica. https://www.britannica.com/science/weather-forecasting/Principles-and-methodology-of-weather-forecasting. Accessed: 2022-03-21.

[4] Jason Brownlee. Random forest for time series forecasting. https://machinelearningmastery.com/random-forest-for-time-series-forecasting/, November 2020. Accessed: 2022-04-01.

[5] Jason Brownlee. Time series forecasting. https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/, August 2021. Accessed: 2022-03-21.

[6] Benjamin Letham Sean J. Taylor. Forecasting at scale. *PeerJ Preprints 5:e3190v2*, 2017.

[7] Weather US. https://weather.us/model-chartsg. Accessed: 2022-03-21.