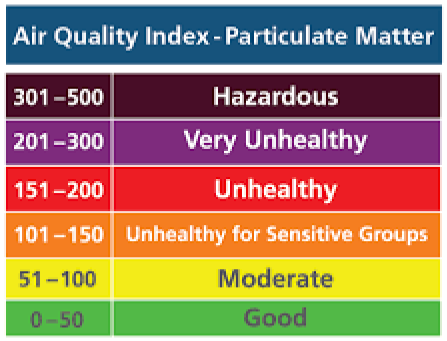
Houston 3-Day AQI (Air Quality Index) Forecast: Technical Overview

Introduction

Houston, Texas is one of the biggest cities in the US with heavy air pollution levels. There are two major causes of air pollution in this city: oil refineries and commuting. Houston has a large oil industry as well as some of the busiest interstates and roads in the country.

We can measure air quality with AQI (Air Quality Index), an index calculated daily by the Texas Commission of Environmental Quality (TCEQ) on over sixty monitoring sites across the Houston-Galveston-Brazoria counties (known as Region 12 by the TCEQ).



Air Quality Index - Particulate Matter	
301 – 500	Hazardous
201 – 300	Very Unhealthy
151 – 200	Unhealthy
101 – 150	Unhealthy for Sensitive Groups
51 – 100	Moderate
0 – 50	Good

Figure 1: AQI values and their respective categories.

Not everyone is affected equally in this region, however. Houston is a massive city and unfortunately, one of the biggest issues is that low-income African-American and Hispanic communities are hit harder than anyone else. These areas have massive swings in air quality from day to day making it hard for at-risk people to predict when it's safe to go outside. Forecasting this "unpredictability" is one of the issues I decided to tackle.

This write-up outlines the process of a website and deep-learning model that determines a three day AQI forecast indicating if it would be safe for certain groups to go outside. The target audience are people who are more sensitive to pollution to see if it would be safe to go out. People more susceptible would include children, senior citizens, pregnant women, people in low income areas, and people with medical conditions like asthma.

Usually with forecasts, meteorologists analyze trends on a map because it is much easier for a human to visualize. But machines can easily interpret rows of data much better than any human can. I wanted to see if weather (temperature, dew point humidity, pressure, wind speed, & visibility) had a correlation with air pollution, and make a forecast for AQI based on that.

I advertised the website by submitting a video to the Houston Teens Care About Clean Air Video Contest led by the Environmental Youth Council in 2021 and won an Honorable Mention for the number of views my video got. With enough support I deployed my app on Streamlit Cloud. You can still look at the video and the project's source code (which can be run locally following instructions found in the GitHub repository) with the following links:

GitHub repository: https://github.com/gallo-json/air-pollution-forecast
YouTube video: https://www.youtube.com/watch?v=m9oTBC5vhqk

I stopped hosting the website after one year because I was intermittently reaching the free Community Tier resource limits for Streamlit Cloud. I also got very busy with school and couldn’t keep up with updating the monthly data and retraining the models. Furthermore, widely used software like Windows 10 and Google started to include AQI forecasting features and my website declined in traffic since then.

Tech Stack

- Python 3
- Jupyter Notebook
  - Used for testing purposes and creating simple prototypes
- OpenCV
  - Used for creating the categories for the website
- Pandas
  - Used for cleaning and filtering the CSV data
- Tensorflow Keras API
  - Used for making the forecasts and the time-series prediction
- NumPy
  - Useful for manipulating arrays
- ScikitLearn
  - Used for scaling the data between 0 and 1 with `MinMaxScaler`
- SciPy
  - Used for k-d to find the nearest weather station from a list of coordinates
- MatplotLib
  - Used for making graphs and visualizing the CSV data
- Streamlit
  - Used for hosting and creating interactive frontend
- Node.js & TypeScript
  - Used to create the interactive map Streamlit component

Program Overview

I chose to build an LSTM (Long-Short Term Memory) neural network because, as the name implies, LSTM models take into account both long and short term trends. The short-term trend in this scenario would be the fluctuation in AQI levels on the daily. For example, AQI levels during the week may be higher due to car pollution commuting to work, compared to AQI levels during the weekends. The long-term trend would be the gradual, steady increase in AQI due to global warming; LSTMs can model this trend because they process sequences of data holistically. A regression model that takes into consideration both trends would be ideal.

The pipeline consists of four main parts: preprocessing the data, scaling the data, feeding the data into the model, and displaying the forecast.

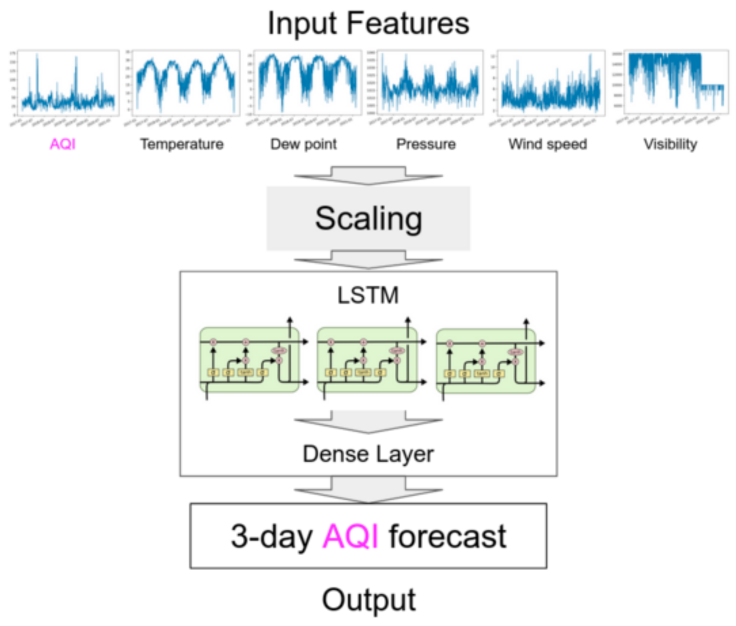


Figure 2: A flowchart showing the ML process for predicting AQIs.

Preprocessing: The input features (temperature, dew point, pressure, wind speed, and visibility) are stored in an array, with its label being the AQI value for that day. Each day going back to January 1, 2012[1] has an array and a label associated with it. Hence, the model was trained on about ten years worth of daily data. Like I mention in the Datasets section of this paper, the data comes from multiple sources, so I have to make sure that the units are consistent.

Scaling: Since each of the features have different ranges of values, I scale the data to fit between 0 and 1. This reduces the disparity between features and makes the training process faster.

Neural network: The reasoning behind choosing an LSTM model was already described, but in the end I include a densely connected layer with a sigmoid activation function to predict an AQI value. This is a regression problem and not a classification problem (although AQI can be classified by six different categories as shown in Figure 1, I wanted my forecasts to be more accurate and include a value). The predictions are then inversely transformed back into AQI values, since they were scaled to begin with.

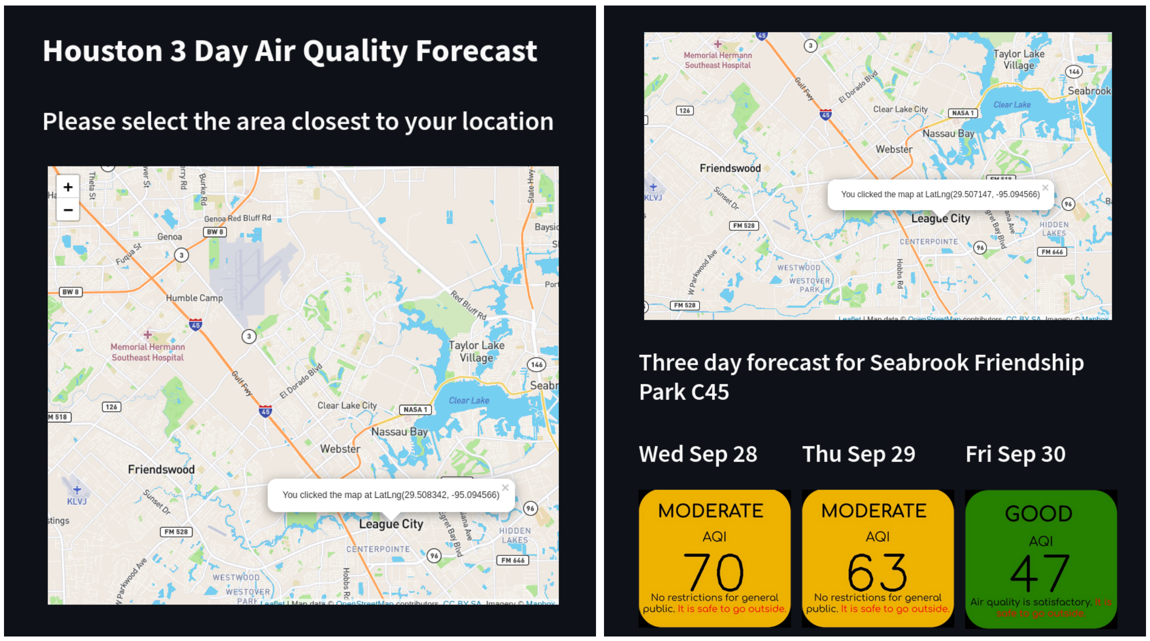
[1] January 1, 2012 was the latest date I could find for an API that served reliable records of Houston's meteorological data, even though the TCEQ AQI records went back to the 1990s.

There are some techniques that I implemented to improve the training process which I think are worth mentioning, inspired by open-source tutorials online (mentioned in the References section):

- The first one is a custom MSE (Mean Squared Error) loss function with a "warm up" period. The "warm up" period ignores the accuracy of those first five steps because, since the model relies on previous time-steps, that may cause bias in later predictions.

- The second one is speeding up the training process by using RMSprop (Root Mean Square Error Propagation) along with a method that reduces the learning rate if the MSE loss starts to plateau. That way RMSprop changes the learning rate to find the global minimum of the loss function faster. Speeding up the training process was really important because there are over 60 models, one tailored to each neighborhood in the region. The models were trained locally on my computer on its GPU.

Website: The frontend was made with Streamlit and community-made Streamlit components, such as the interactive Leaflet map API Streamlit component. The Leaflet Streamlit component is deployed on a separate port and sends the clicked coordinates to the main Python script. From there, the program performs a k-d tree with SciPy to find the monitoring site nearest to where the user clicked. Then the model for that specific site is queried and returns the three day forecast (Figure 3).


Houston 3 Day Air Quality Forecast

Please select the area closest to your location

You clicked the map at LatLng(29.508342, -95.094566)

You clicked the map at LatLng(29.507147, -95.094566)

Three day forecast for Seabrook Friendship Park C45

Wed Sep 28	Thu Sep 29	Fri Sep 30
MODERATE	MODERATE	GOOD
AQI 70	AQI 63	AQI 47
No restrictions for general public. It is safe to go outside.	No restrictions for general public. It is safe to go outside.	Air quality is satisfactory. It is safe to go outside.

Figure 3: Website working locally with Streamlit. The user clicks on an area in Houston and forecasted AQIs are posted below it with quick loading time. For instance, I clicked near League City, TX and it calculated the forecast for Sept 28-30.

New data is web scraped and put together every month and the model is retrained by myself. Feeding the model fresh data makes the model's predictions as accurate as possible instead of having to rely on past predictions for more than a month, which can cause a butterfly effect if the data becomes skewed.

Datasets

Most of the time spent on this project was not building the model, but rather cleaning up the data with Pandas. The data itself goes through a pipeline from web scraping all the way to feeding it into the model. I coalesced two different datasets provided by the NOAA (National Oceanic and Atmospheric Administration) and the TCEQ.

NOAA's dataset came formatted as a SQL database from DoltHub, which I downloaded and converted to CSV. It contains meteorological weather data from over thirteen different weather stations strategically placed around the Houston area. The daily meteorological data were the features I fed into the model.

TCEQ's dataset is an online tabular dataset containing daily AQI data. I created methods to web-scrape the values into a CSV file so I could parse and feed into the model. Like I mentioned earlier, it provides data from over 60 monitoring sites in Houston. The AQI value is what the model is trying to predict three days into the future.

Notice that the NOAA dataset has only thirteen weather stations, while the TCEQ dataset has over 60 monitoring sites. To coalesce the datasets, I created an algorithm to match each of the 60 monitoring sites to the nearest NOAA station to make sure the meteorological data lined with each neighborhood. Because each monitoring site has unique coordinates, I performed a k-d tree (with SciPy) to find the nearest NOAA station and mapped monitoring sites to stations that way. I use a k-d tree because it is faster than computing all the euclidean distances between coordinates and sorting to find the smallest distance[2]. Because Houston is such a big city, I wanted to keep all the monitoring sites TCEQ provides in order to create models tailored to each neighborhood in the region. This is important because each area is affected differently by air pollution.

I also matched the AQI (y_train and y_test) and meteorological data (X_train and X_test) based on dates. Furthermore, I had to filter out all the data that was messed up (e.g. null values). If many null values were alongside each other, I would have to perform linear interpolation to have at least some numbers to work with, but this of course would skew the data if a lot was missing.

From there, I train/test split the data in chronological order to train the model and then test it on data never it had never seen before (Figure 4). Once the loss was satisfactory, the models were ready to deploy for forecasting.

[2] The average search k-d tree has a time complexity of $O(\log n)$

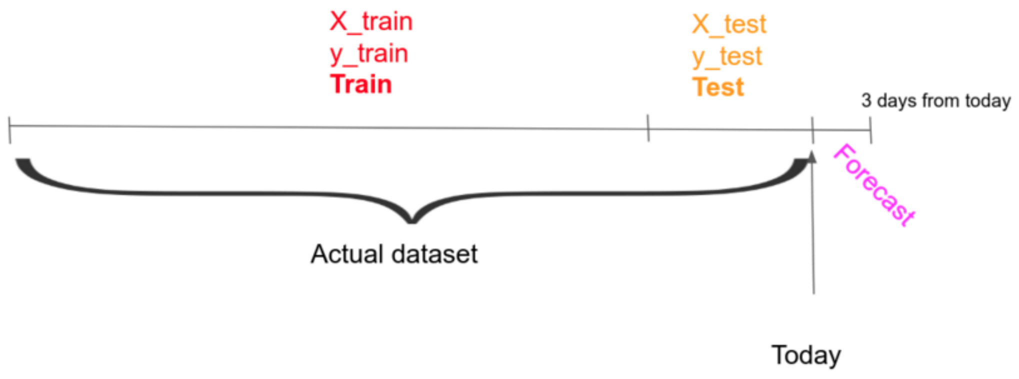
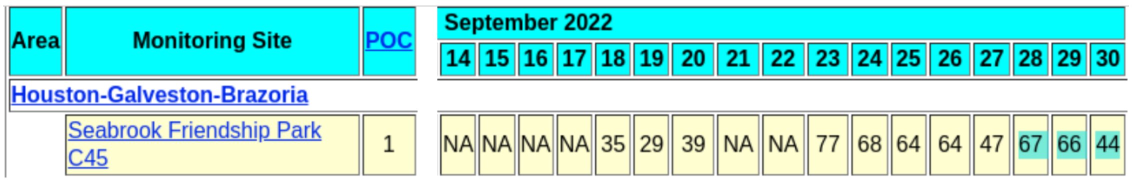


Figure 4: Diagram showing how the dataset was split in order to train the model.

Results

The average scaled MSE loss across all the models was 0.1. This translates into roughly 4 AQI points after an inverse transformation. In other worlds, the models had an average accuracy of ±4 AQI points.



Area	Monitoring Site	POC	September 2022																
			14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Houston-Galveston-Brazoria																			
	Seabrook Friendship Park C45	1	NA	NA	NA	NA	35	29	39	NA	NA	77	68	64	64	47	67	66	44

Figure 5: Double-checking the predictions from Figure 3 three days later after visiting the website. Notice how the Seabrook Friendship Park station did not record data for many days at the beginning of the month, yet the predictions for Sept. 28-30 were accurate to ±3 AQI points.

In Figure 6, the blue line shows the real AQI values, whereas the orange line shows the predicted AQI values. Notice how although this is testing data that the model has never seen before, the model predicts peaks and troughs in the AQI very well.

For example, notice how the model predicted the massive spike in AQI right before the hundredth day. Perhaps that day the interstate next to the neighborhood experienced heavy traffic with extreme heat. The AQI that day was around 150 (unhealthy for sensitive groups), and it was able to predict that well in advance. But it also predicted the period of low AQI (i.e. clean air) starting from around the 50th day. Predicting these swings in advance is important because the website notifies at-risk people days in advance in case they need to stay at home.

This shows that the model accomplished my goal for this project, which was to predict the wild swings in AQI from day to day.

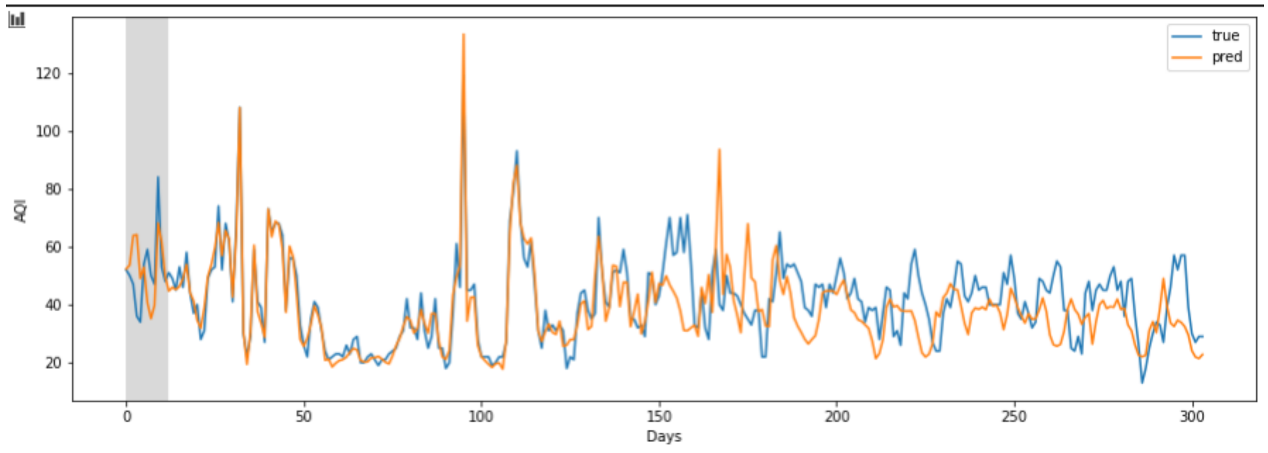


Figure 6: Graph showing the relationship between days past and AQI from the testing dataset. This data is from the Channelview C15-AH115 TCEQ monitoring site from Channelview, TX.

References

This project was heavily based on other open-source software. Apart from all the library documentations, these are two projects that I took code from and modified for my project.

Tensorflow Time-Series Prediction Tutorial:
https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/23_Time-Series-Prediction.ipynb

Interactive map Streamlit component:
https://github.com/andfanilo/streamlit-light-leaflet