Advanced Weather Forecasting

Ninad, Kengloong, Brientt, Lin Hong

Project Workflow

01	Data Scraping and Cleaning	Kengloong
02	Exploratory Data Analysis	Brientt
03	Data Manipulation and Preprocessing	Lin Hong
04	Model Creation and FineTuning	Ninad
05	Frontend App	Ninad

Context

- Weathers and the different environmental variables are constantly fluctuating nowadays due to global warming

- The weather is disruptive to the daily lives of people as the different environmental variables are constantly changing.

Objectives

We aim to create a more detailed Time Series Model with enhanced accuracy which help us predict essential environmental variables like humidity, temperature, and weather to enhance people's daily decision-making, from dressing appropriately to planning activities, by providing them with comprehensive information.

01

Data Scraping and Cleaning

Kengloong

Data Scraping

World Weather Online API

Weather API

Global daily, hourly, minute weather forecast, historical weather, marine weather, time zone, astronomy and much more. Over 300K+ Weather API users worldwide count on us.

Realtime	365 days ahead weather	Hourly, daily and 15 min interval	7 day Marine + Tide	Solar data	Historical weather	Air Quality Data	Astronomy, Time Zone

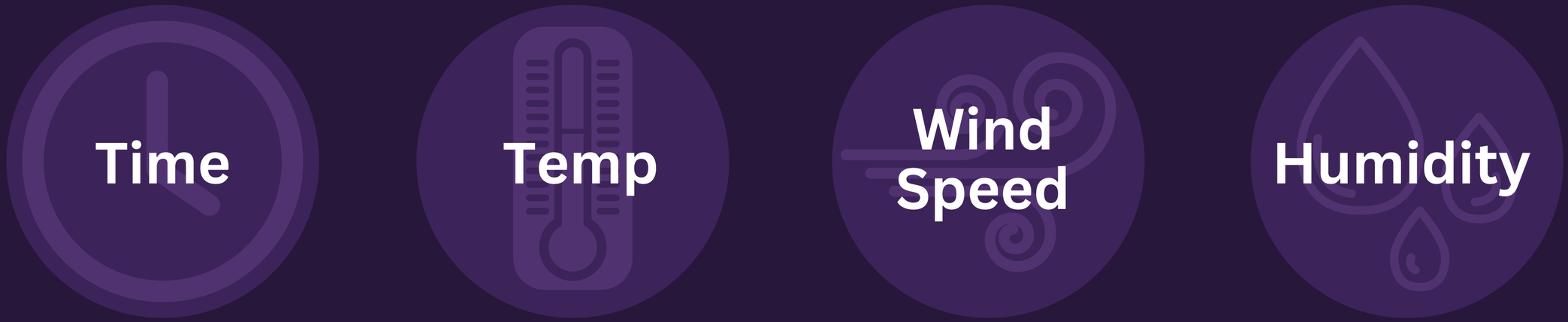
Weather API » | Weather Videos » | Sports Weather »

Global Weather

UK	USA	Australia	Canada
Blackpool	Anaheim	Adelaide	Abbotsford
Bolton	Bronx	Ballarat	Brampton
Bournemouth	Boston	Byron bay	Calgary
Bradford	Chicago	Brisbane	Edmonton
Brighton	Denver	Canberra	Hamilton
Coventry	Fort Wayne	Cairns	Mississauga
Derby	Houston	Geelong	Montreal
Hull	Las Vegas	Gold Coast	Ottawa

Data Cleaning

Extract relevant parameters such as



Data Workflow

Dataset Generator call the 2 functions

Returns the weather data and parses it

Writes the data onto a csv file

Run this file for about 20 cities such as Singapore, Paris & Dubai

02

Exploratory Data Analysis

Brientt

Parameters

Temperature

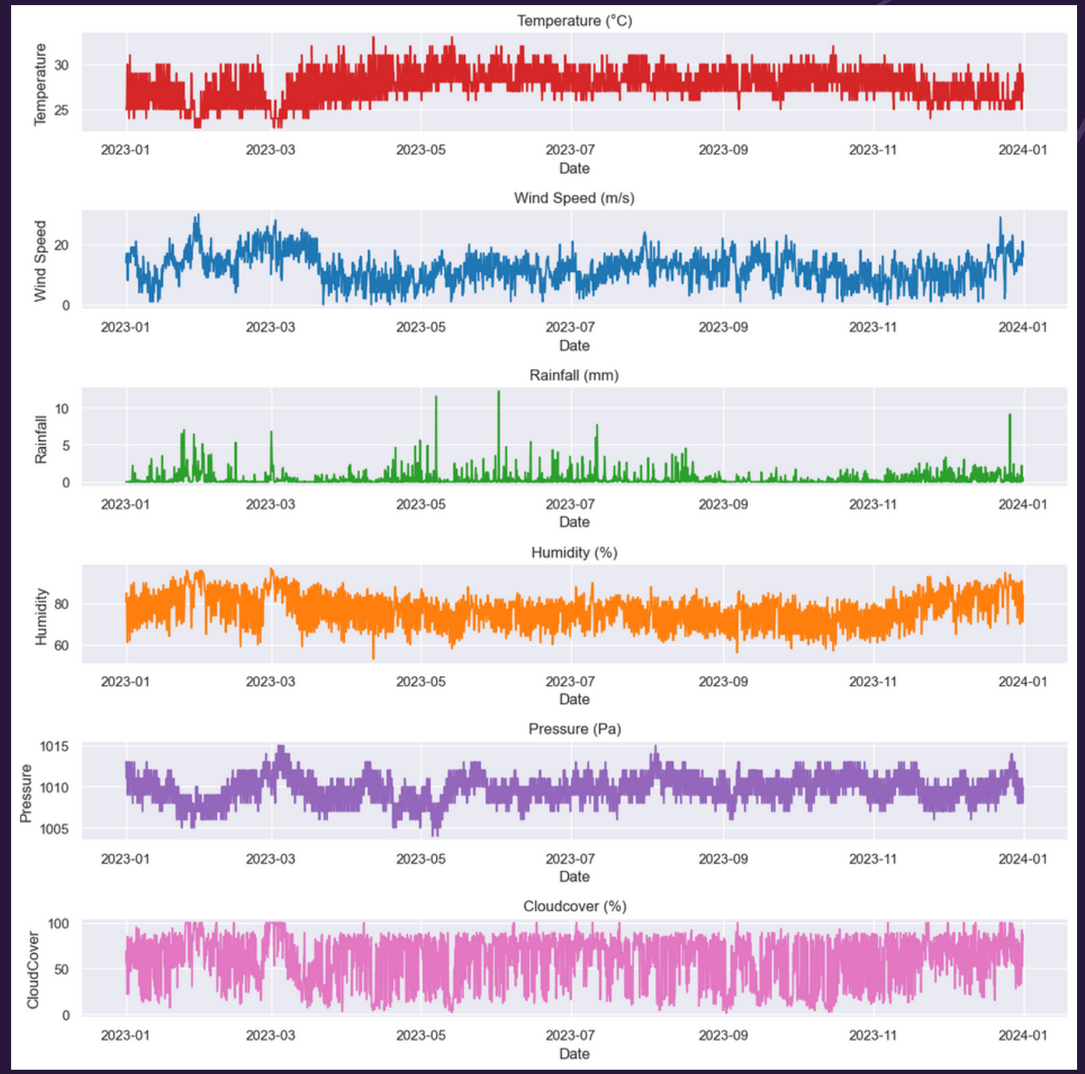
Wind Speed

Rainfall

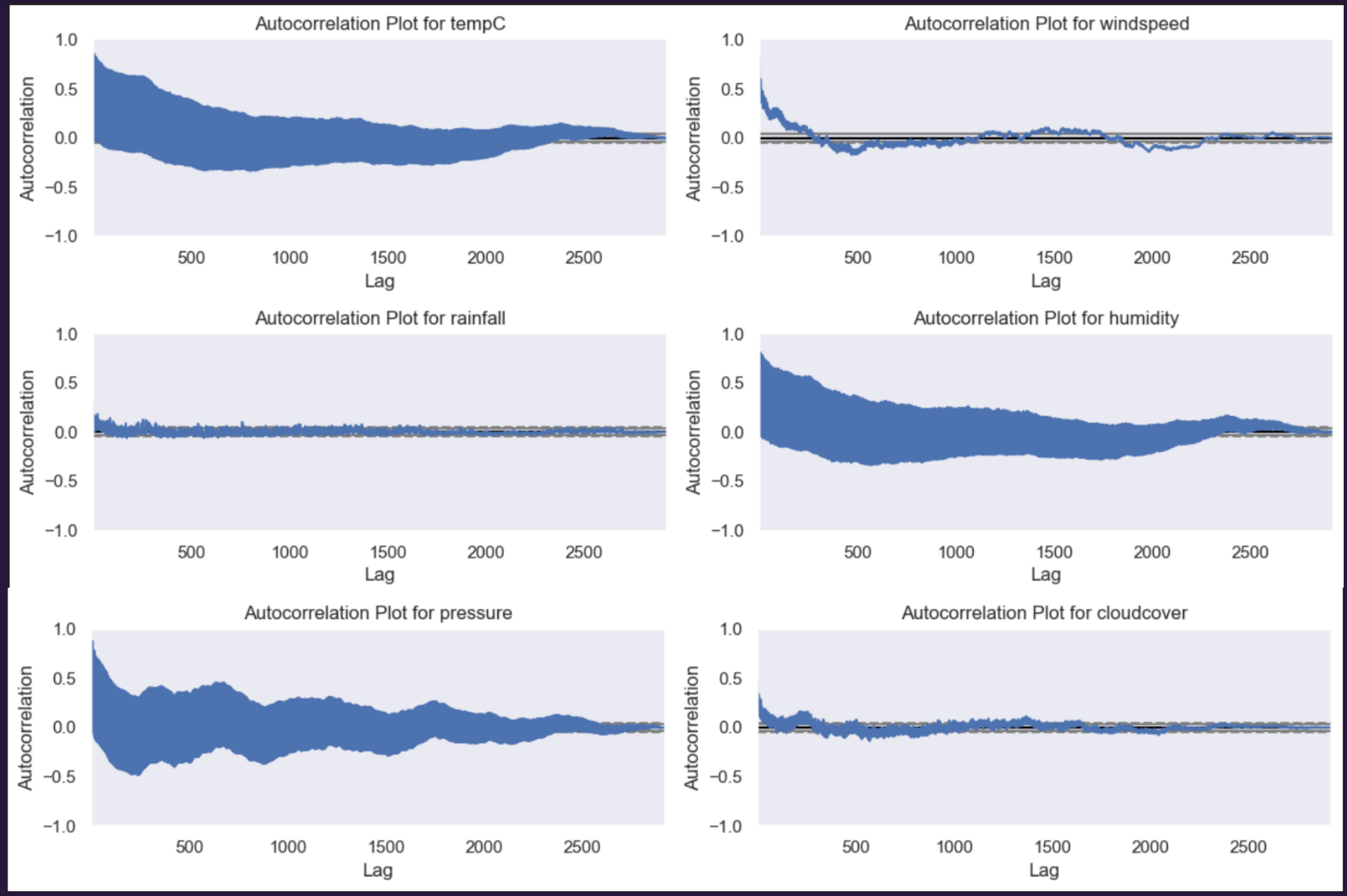
Humidity

Pressure

Cloudcover

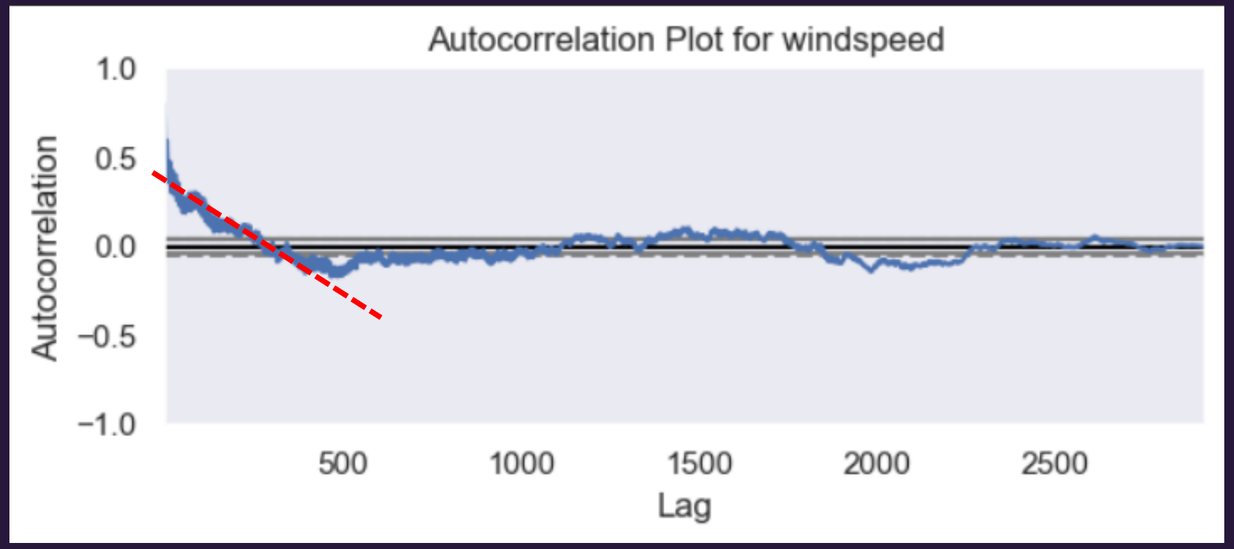


Autocorrelation



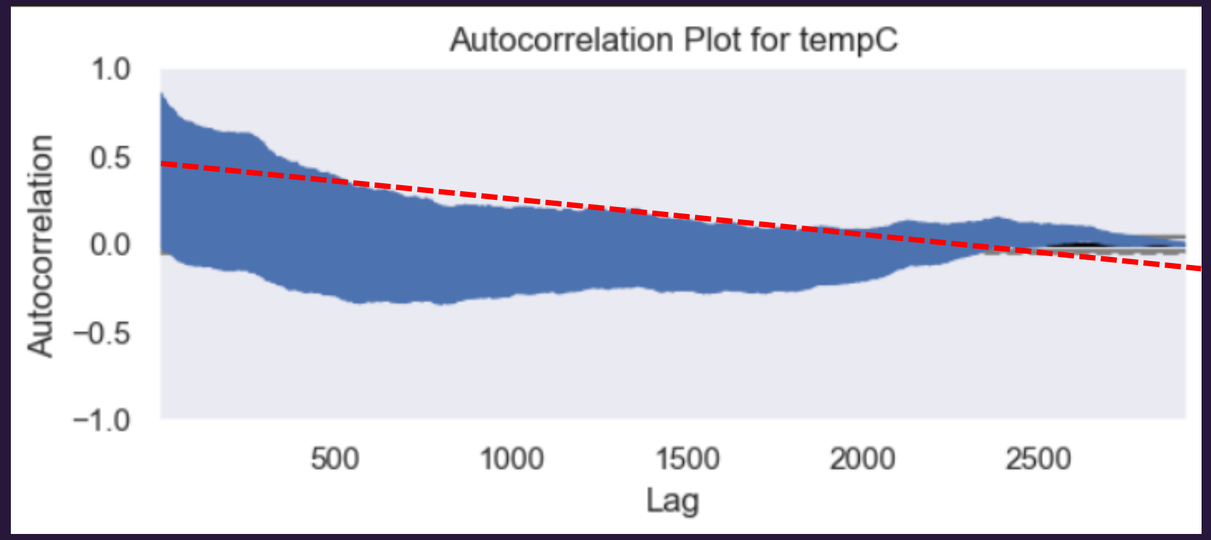
Decay Rate

Steep decay rate:
only the recent values of a parameter
is useful to predict the new predictions



Windspeed-time series has
a steep decay rate

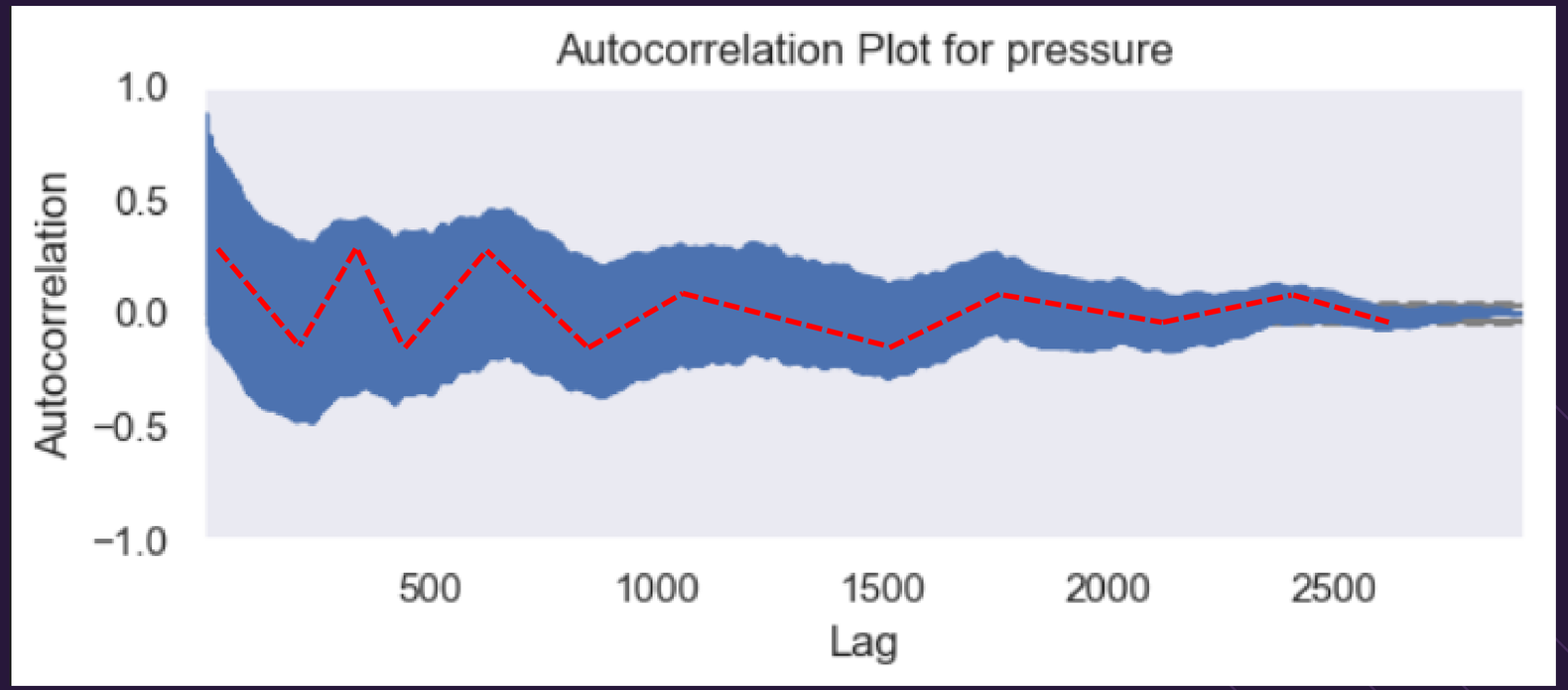
Gradual decay rate:
past observations maintain a relatively
stronger influence on the current value



Temperature-time series has
a gradual decay rate

Cyclic Pattern

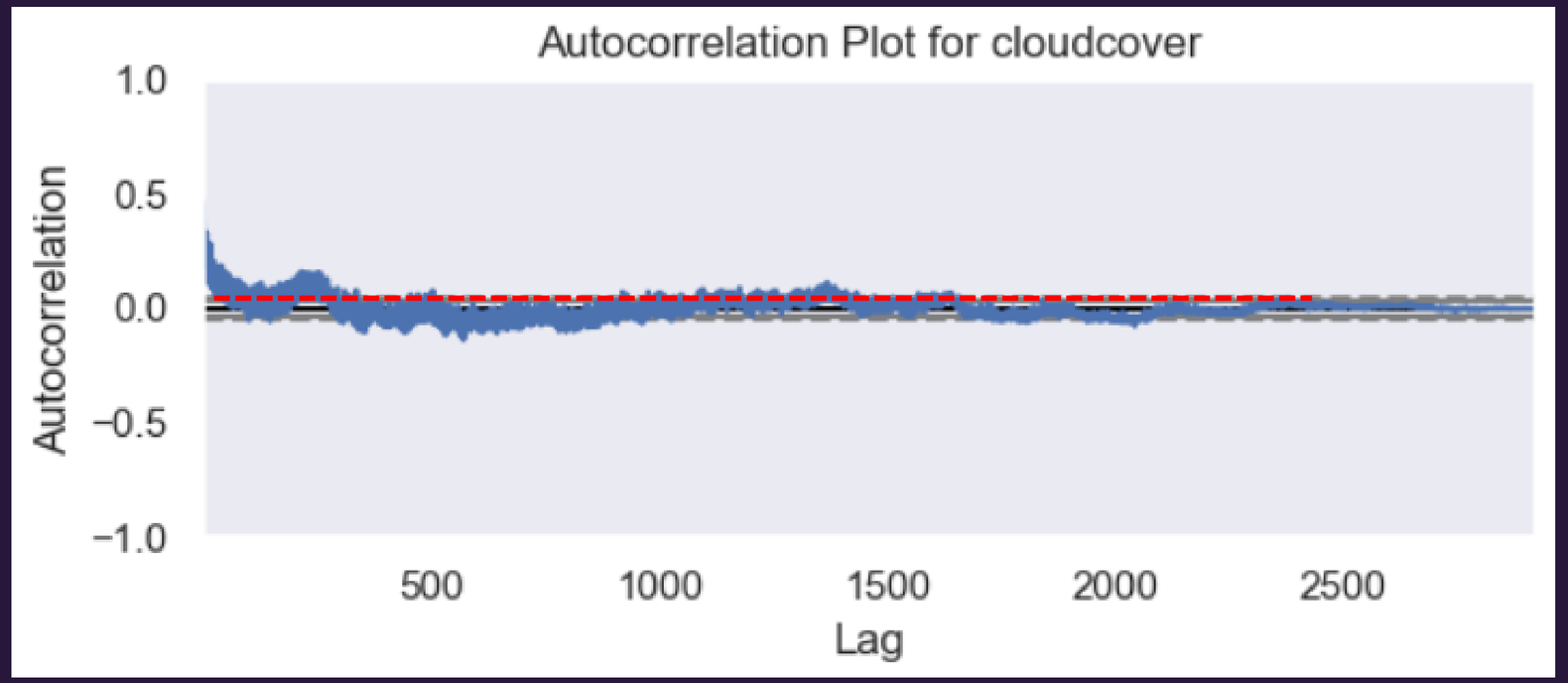
Cyclic pattern - regular and repeating pattern of changes which depicts seasonality in values



Pressure-time series

Randomness

If correlation is very low, the change is random and does not flow a predictable pattern.



03

Data Preprocessing and Manipulation

Lin Hong

Reasons for Data Manipulation

Typically to a range between -1~1 or 0~1

Imposes constraints on the model's weights

Accelerates the convergence of optimization algorithms

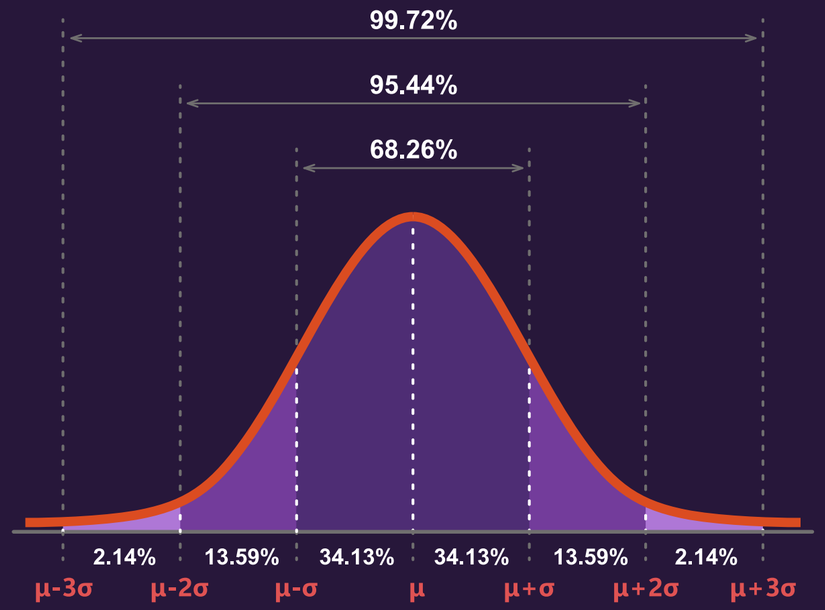
Reduces the risk of overfitting to train data

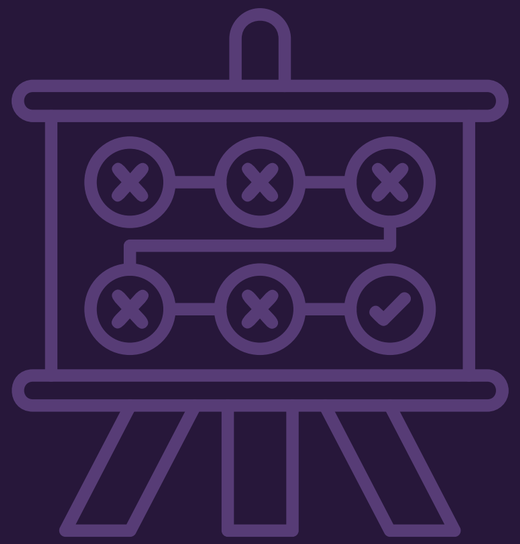
Normalization of Train data - How it works?

```python
def time(self, time):
    return time / 24
```

```python
def temp(self, temp):
    mean = 13.0
    stdev = 9.869784
    return ( (temp - mean) / stdev ) / 3

def windspeed(self, speed):
    return ( speed - 25) / 50

def pressure(self, pressure):
    return (pressure - 1000) / 50
```

```python
def rainfall(self, rainfall):
    return rainfall / 20

def humidity(self, humidity):
    return humidity / 120

def cloudcover(self, cover):
    return cover / 120
```





Renormalization of data

Why?

- Forecasted weather is for humans to read
- Predictions made are on normalized scale
- Not easily understandable
- Renormalizing improves readability

How?

- Reverse engineering / working backwards

AI Input, Get Nearest Time & Get Image

Job scopes of the functions

AI Input: the master function, recalls the relevant range of data from the dataset and sends them to the model for training.

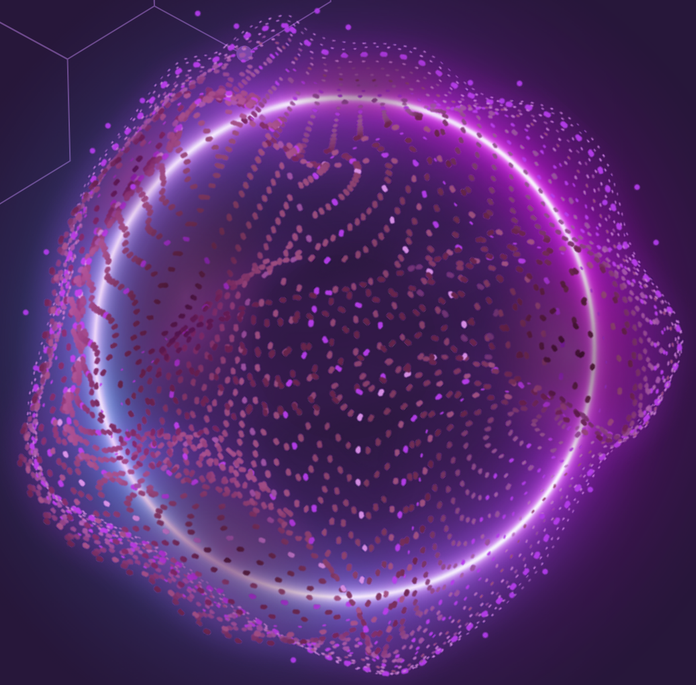
Get Nearest Time: adjust the input time to the nearest 3-hour increment as our dataset uses 3, 6, 9, 12 AM & PM data only.

Get image: interprets predicted parameters, mapping them to suitable image filenames for visualization.

04

Model Creation and Finetuning

Ninad



Long Short Term Memory Networks

An LSTM is a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data. LSTMs are equipped with special memory cells that can selectively remember or forget information over time.

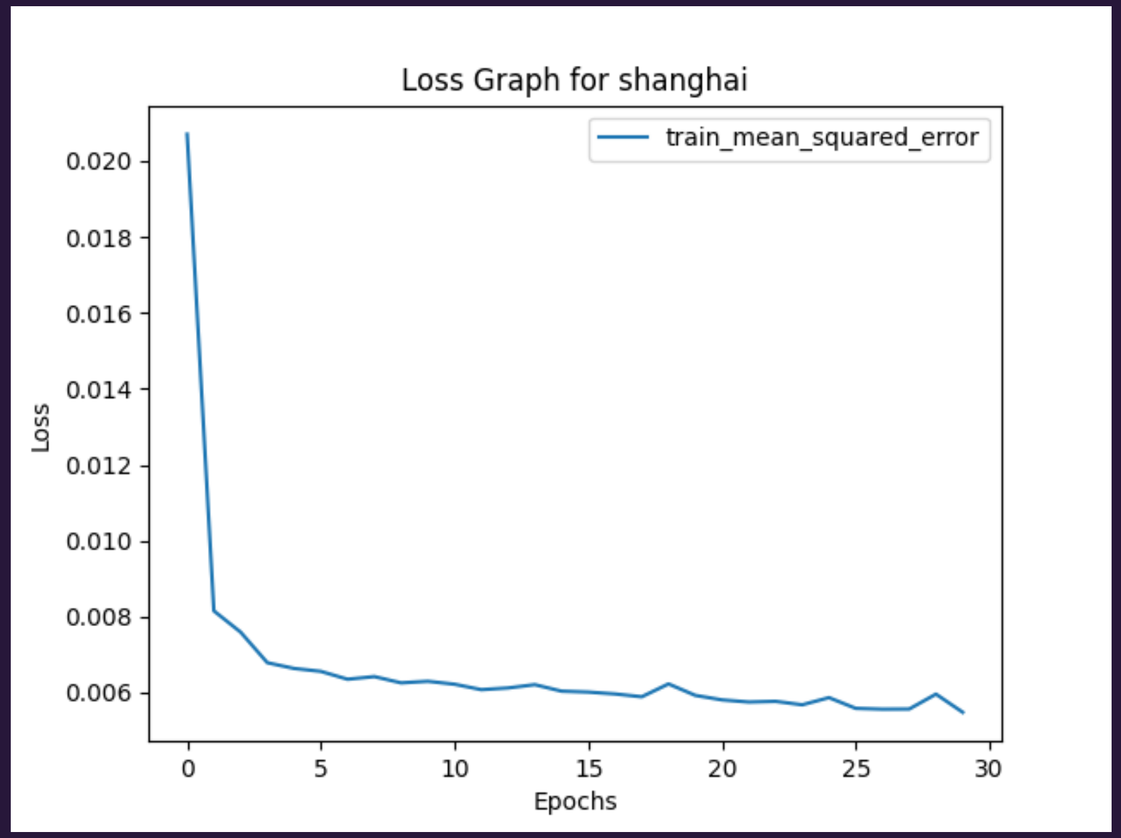
Why LSTM ??

They can capture complex temporal patterns, crucial for modeling weather dynamics accurately, outperforming traditional methods in handling long-term dependencies and non-linearities within time series data.

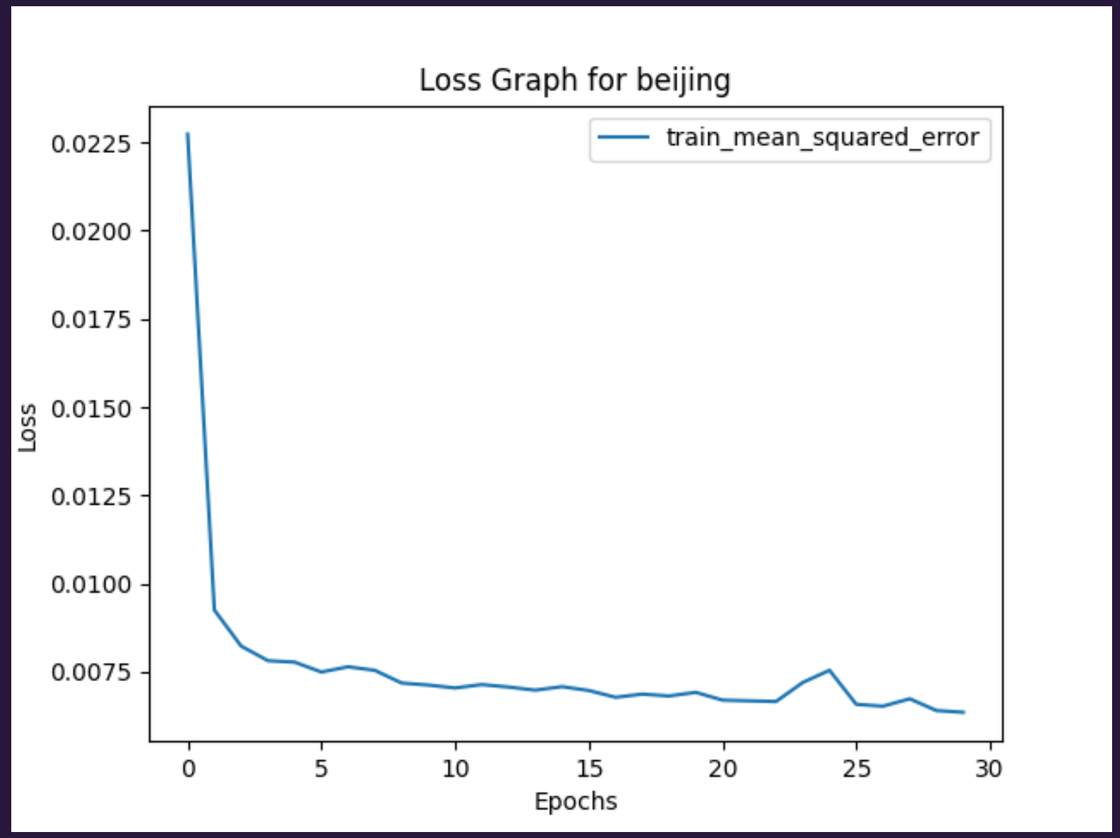
Model Architecture

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru (GRU)                   (None, 1, 128)            52608

 gru_1 (GRU)                 (None, 128)               99072

 dense (Dense)               (None, 512)               66048

 dense_1 (Dense)             (None, 7)                 3591

=================================================================
Total params: 221319 (864.53 KB)
Trainable params: 221319 (864.53 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Model Performance - Training Loss (MSE)



Shanghai



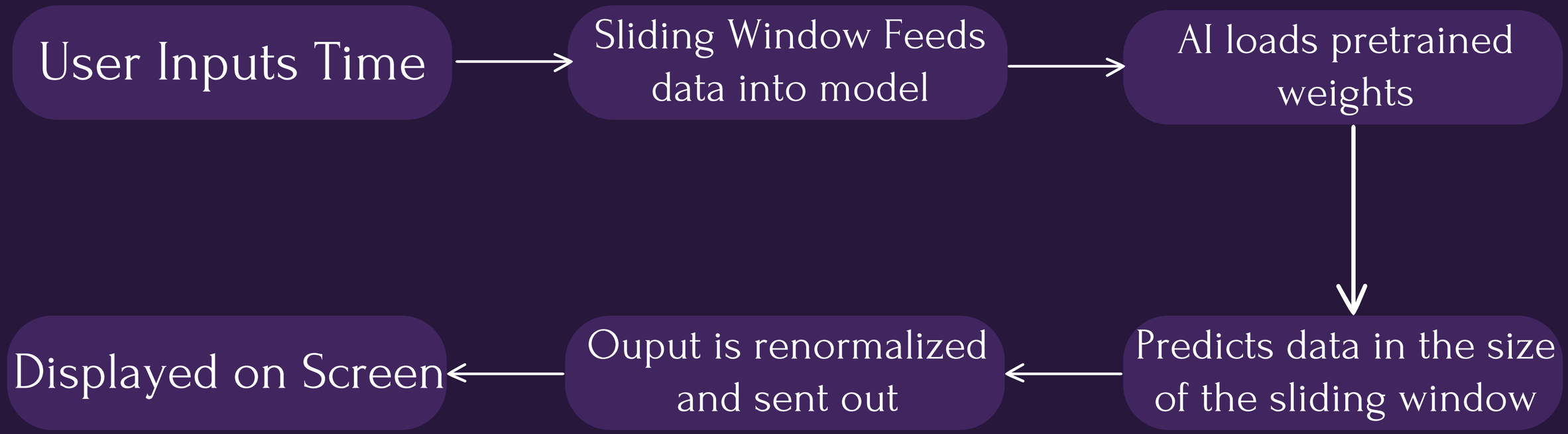
Beijing

Model Performance - Testing Loss (MSE)

```
27/27 [==============================] - 1s 6ms/step
Mean Square Error for shanghai: 0.010323245054993l6
Evaluating mean square error for beijing...
27/27 [==============================] - 1s 6ms/step
Mean Square Error for beijing: 0.008142937393277271
Evaluating mean square error for moscow...
18/18 [==============================] - 1s 6ms/step
Mean Square Error for moscow: 0.007517507995990033
Evaluating mean square error for karachi...
27/27 [==============================] - 1s 6ms/step
Mean Square Error for karachi: 0.019636702804828404
Evaluating mean square error for singapore...
27/27 [==============================] - 1s 6ms/step
Mean Square Error for singapore: 0.01695447906325258
Evaluating mean square error for london...
27/27 [==============================] - 1s 7ms/step
Mean Square Error for london: 0.011755865792817172
```

```
27/27 [==============================] - 1s 6ms/step
Mean Square Error for madrid: 0.013333511564898298
Evaluating mean square error for berlin...
27/27 [==============================] - 1s 5ms/step
Mean Square Error for berlin: 0.007963015993990354
Evaluating mean square error for paris...
27/27 [==============================] - 1s 6ms/step
Mean Square Error for paris: 0.011271321620767306
Evaluating mean square error for sydney...
27/27 [==============================] - 1s 6ms/step
Mean Square Error for sydney: 0.014500108344966761
Evaluating mean square error for rome...
27/27 [==============================] - 1s 6ms/step
Mean Square Error for rome: 0.010119664678l9526
Evaluating mean square error for toronto...
27/27 [==============================] - 1s 7ms/step
Mean Square Error for toronto: 0.010808070205799412
```

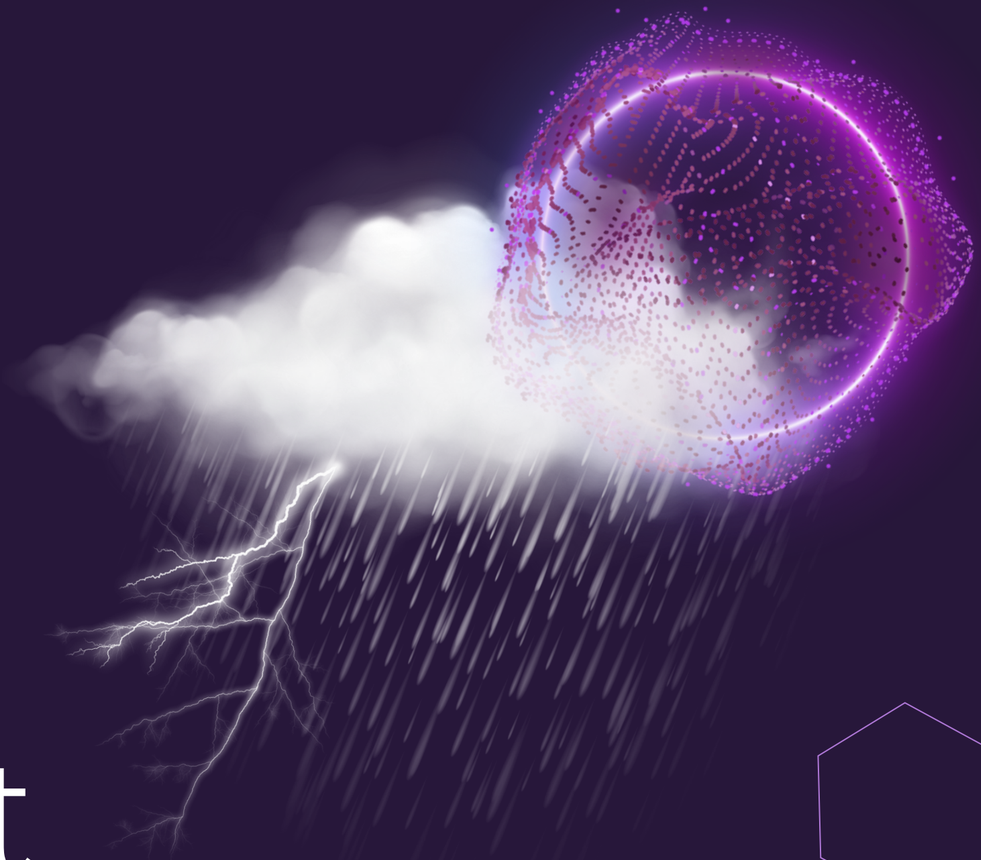
Model Workflow



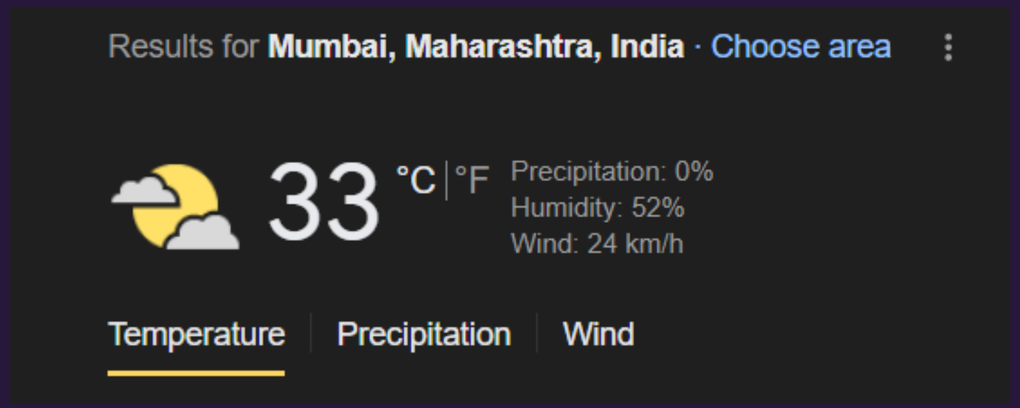
05

Weather Report Generation and Frontend

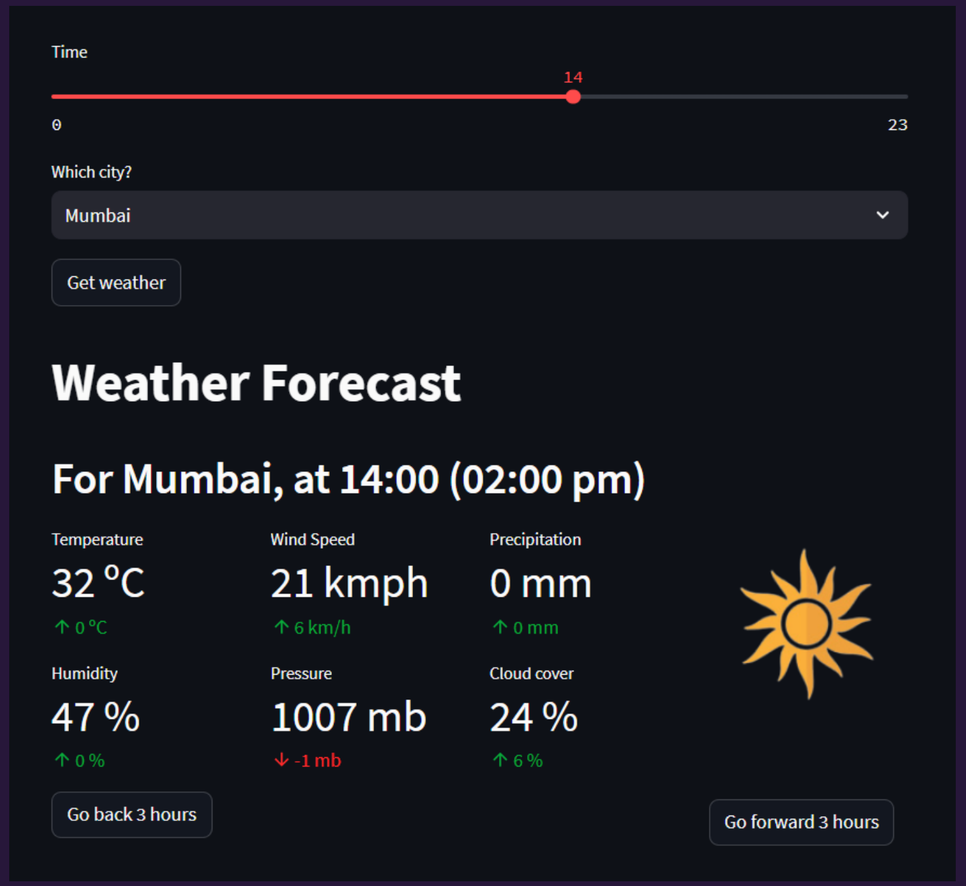
Ninad



Model Demo - Mumbai

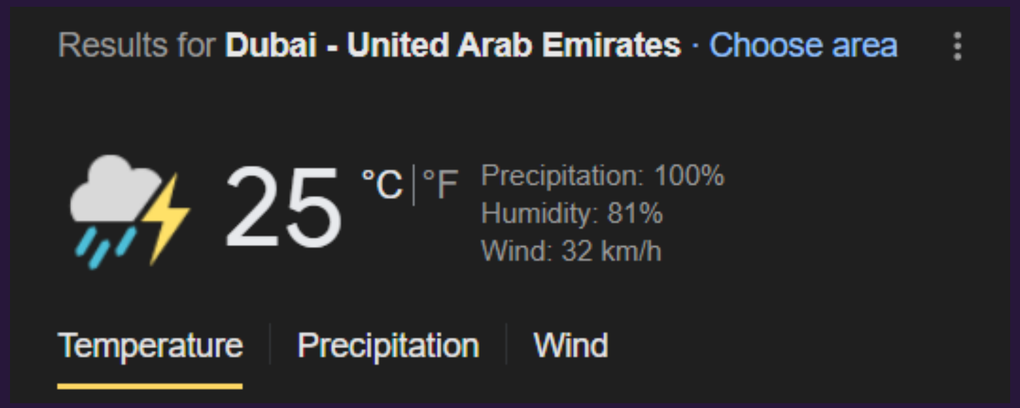


Actual Weather

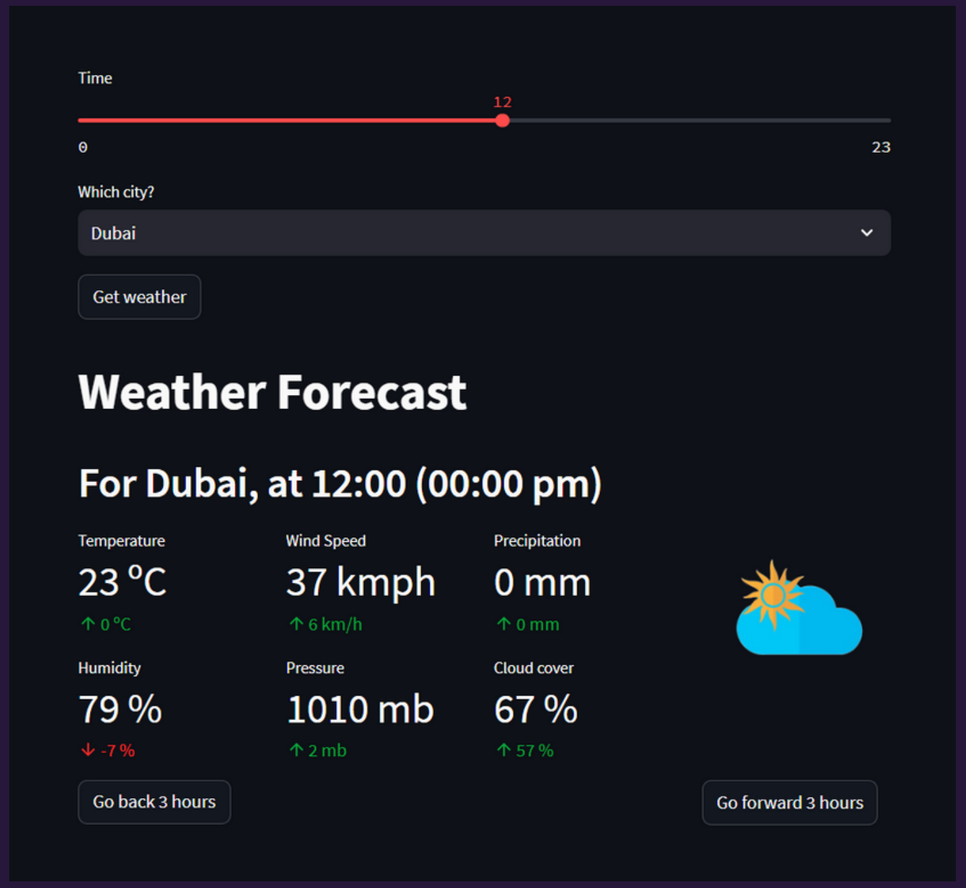


Predicted Weather

Model Demo - Dubai



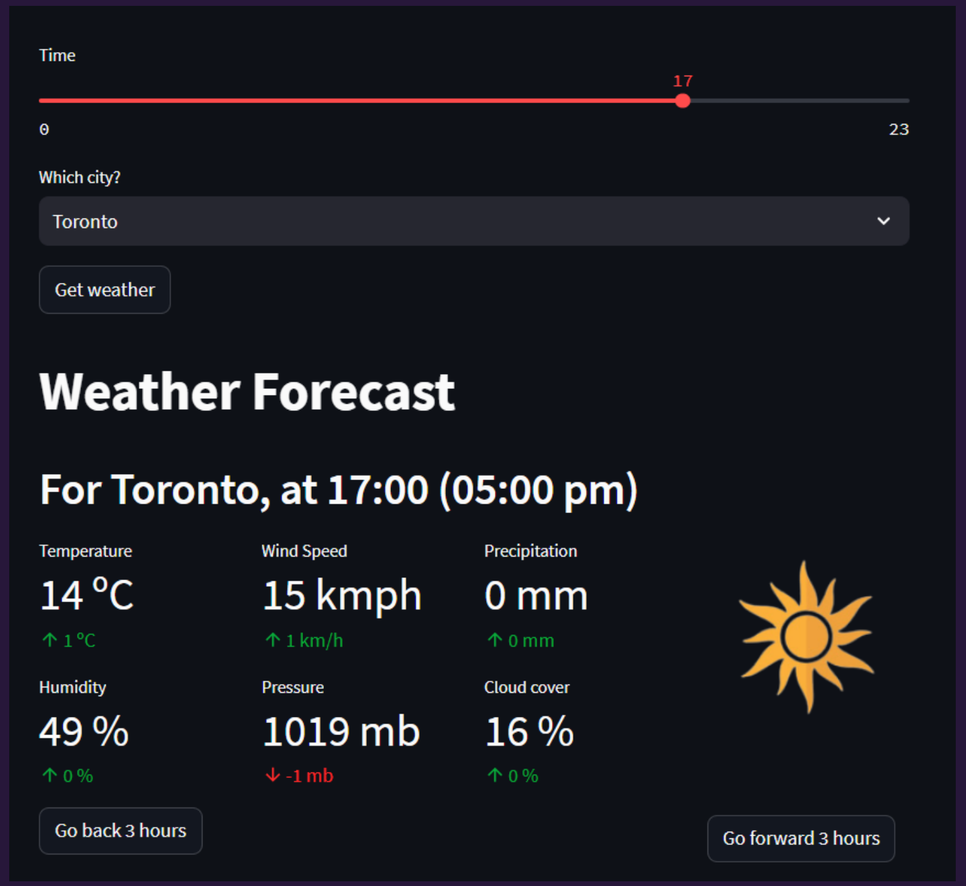
Actual Weather



Predicted Weather

Model Demo - Toronto

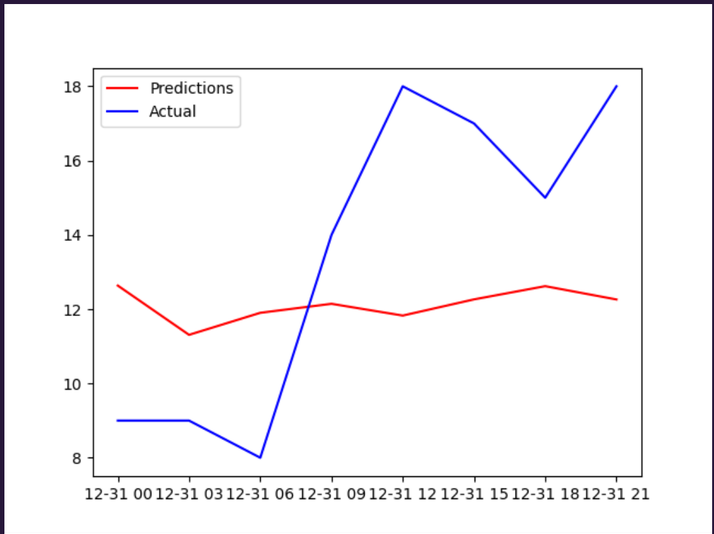
Actual Weather



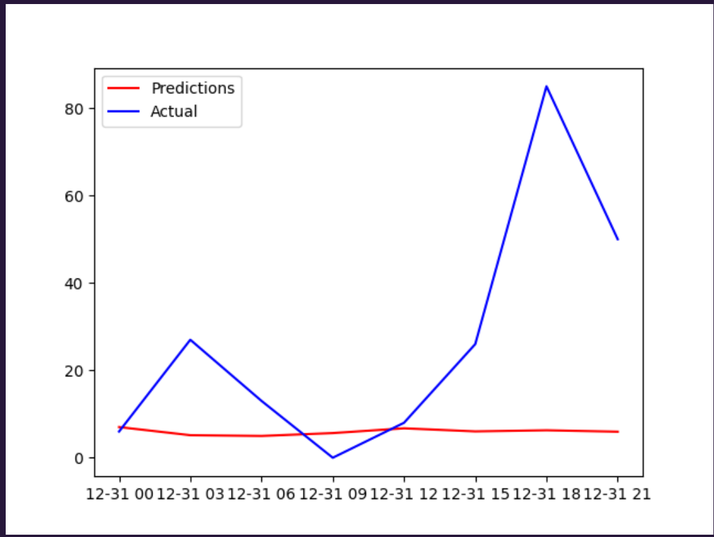
Predicted Weather

Appendix

ARIMA (Autoregressive Integrated Moving Average) Model MSE



MSE : 17.0440222
Windspeed



MSE : 1138.839316
Cloudcover

Appendix

01	Data Scraping and Cleaning	Kengloong
02	Exploratory Data Analysis	Brientt
03	Data Manipulation and Preprocessing	Lin Hong
04	Model Creation and FineTuning	Ninad
05	Frontend App	Ninad