**THAKUR DEGREE COLLEGE OF SCIENCE AND COMMERCE**

KANDIVALI EAST

MUMBAI

A
PROJECT
ON

# Weather Prediction Using Machine Learning

For

Thakur College of Science and Commerce

By

Aman Singh

Submitted in partial fulfillment of
Bachelors of Science (Computer
Science)

**[UNIVERSITY OF
MUMBAI]**

Thakur Degree College of **Science** and
**Commerce Kandivali (East), Mumbai.**

**ACADEMIC YEAR 2022-2023**

**DATE:** ______________

THAKUR TRUSTS

Thakur Educational Trust's (Regd.)

THAKUR COLLEGE OF SCIENCE & COMMERCE

AUTONOMOUS COLLEGE, PERMANENTLY AFFILIATED TO UNIVERSITY OF MUMBAI

NAAC Accredited Grade 'A' (3rd Cycle) & ISO 9001: 2015 (Certified)

Azadi Ka Amrit Mahotsav Best College Award by University of Mumbai for the Year 2018-2019

tcsc CELEBRATING 25 YEARS OF GLORY

## A PROJECT ON Weather Prediction Using Machine Learning

A PROJECT SUBMITTED TO

THE UNIVERSITY OF MUMBAI FOR PARTIAL COMPLETION OF THE DEGREE

OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

UNDER THE FACULTY OF SCIENCE

BY

AMAN KAMLESH SINGH

UNDER THE GUIDANCE

OF MR GIRISH TERE

THAKUR COLLEGE OF SCIENCE AND COMMERCE, KANDIVALI

(EAST) MUMBAI, MAHARASHTRA 400101

THAKUR TRUSTS

Thakur Educational Trust's (Regd.)

THAKUR COLLEGE OF SCIENCE & COMMERCE

AUTONOMOUS COLLEGE, PERMANENTLY AFFILIATED TO UNIVERSITY OF MUMBAI

NAAC Accredited Grade 'A' (3rd Cycle) & ISO 9001: 2015 (Certified)

Azadi Ka Amrit Mahotsav **Best College Award by University of Mumbai for the Year 2018-2019**

tcsc CELEBRATING 25 YEARS OF GLORY

## DECLARATION BY LEARNER

I the undersigned **MR AMAN SINGH** hereby, declare that the work embodied in this project work titled **"WEATHER PREDICTION USING MACHINE LEARNING"** forms my own contribution to the research work carried out under the guidance of **Prof. GIRISH TERE** is a result of my own research work and has not been previouslysubmitted to any other University for any other Degree/Diploma to this or any other University.

Wherever reference has been made to previous works of others, it has been clearly indicated as such and included in the bibliography.

I, hereby further declare that all information of this document has been obtained and presented in accordance with academic rules and ethical conduct.

Name and signature of learner

**AMAN SINGH**

Certified By

Name and Signature of Guiding Teacher

**MR GIRISH TERE**

THAKUR TRUSTS

Thakur Educational Trust's (Regd.)

**THAKUR COLLEGE OF SCIENCE & COMMERCE**

AUTONOMOUS COLLEGE, PERMANENTLY AFFILIATED TO UNIVERSITY OF MUMBAI

NAAC Accredited Grade 'A' (3rd Cycle) & ISO 9001: 2015 (Certified)

**Best College Award by University of Mumbai for the Year 2018-2019**

Azadi Ka Amrit Mahotsav

tcsc

CELEBRATING 25 YEARS OF GLORY

## CERTIFICATE

This is to certify that **MR AMAN KAMLESH SINGH** has worked and duly completed his Project Work for the Degree of Bachelor in Computer Science under the Faculty of Science in the subject of Computer and the project is entitled, **"WEATHER PREDICTION USING MACHINE LEARNING"** under my supervision.

I further certify that the entire work has been done by the learner under my guidance and that no part of it has been submitted previously for any Degree or Diploma of any University.

It is her own work and facts reported by her personal findings and investigations.

| | | |
|---|---|---|
| **Dr. C.T. CHAKRABORTY** | **MR ASHISH TRIVEDI** | **PROF. GIRISH TERE** |
| Principal | Head Of Department | Project Guide |

INTERNAL EXAMINER EXTERNAL EXAMINER

Date of submission:

# INDEX

# **ACKNOWLEDGEMENT:**

Achievement is finding out what you would be doing rather thanwhat you have to do. It is not until you undertake such a project that you realize how much effort and hard work it really is, what are your capabilitiesand how well you can present yourself or other things. It gives me immense pleasure to present this report towards the fulfillment of my project.

It has been rightly said that we are built on the shoulder of others.For everything I have achieved, the credit goes to all those who had helped me to complete this project successfully.

I take this opportunity to express my profound gratitude tomanagement of Thakur Degree College of Science & Commerce for giving me this opportunity to accomplish this project work.

I am very much thankful to **Mrs. C. T. Chakraborty** - Principal of Thakur College for their kind co-operation in the completion ofmy project.

A special vote of thanks to our HOD **Mr ASHISH TRIVEDI**
and to our project guide **Mr GIRISH TERE**
Finally, I would like to thank all my friends & entire Computer Science department who directly or indirectly helped me in completion of thisproject & to my family without whose support, motivation & encouragementthis would not have been possible.

(AMAN SINGH)

# ORGANIZATION OVERVIEW:

The **Thakur College of Science and Commerce (TCSC)** is a college in Kandivali in Mumbai of Maharashtra, India running by Thakur  Education Trust

Thakur College was started in 1992 to serve the needs of students passing SSC examination from the schools around Kandivali area and Thakur Vidhya Mandir which has already established itself as one of the schools in the area. It offers courses at primarily the higher secondary and under-graduate levels. The courses at the undergraduate and post-graduate level are offered in affiliation with Mumbai University, Mumbai. An ISO 9001:2008 College with **A** grade as assessed by the National Assessment and Accreditation Council NAAC.

| | |
|---|---|
| Name: | Thakur College of Science and Commerce |
| Founded: | 1997 |
| Address: | Thakur College of Science and Commerce, Thakur Village<br>Kandivali(E), Mumbai – 400 001 |
| Motto: | Journey towards Excellence |
| Total Staff: | 200 |
| Number of Students: | 12500 |
| Email: | Helpdesk@tcsc.org.in |

## Software Requirements:

- Jupyter Notebook
- Python Packages

## Hardware Requirements:

- I5 Processor
- 4GB RAM

## Tools & Techniques:

- Jupyter Notebook
- Python module

  (Panda , Numpy , Matplotlib , sklearn)

## PYTHON

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0

# JUPYTER NOTEBOOK

JupyterLab: A Next-Generation Notebook Interface JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality.
Jupyter Notebook: The Classic Notebook Interface The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

# PANDA

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

# NUMPY

NumPy (**Numerical Python**) is an open  source Python library that's used in almost every field of science and engineering. It's the universal standard for working with numerical data in Python, and it's at the core of the scientific Python and PyData ecosystems. NumPy users include everyone from beginning coders to experienced researchers doing state-of-the-art scientific and industrial research and development. The NumPy API is used extensively in Pandas, SciPy, Matplotlib, scikit-learn, scikit-image and most other data science and scientific Python packages.

The NumPy library contains multidimensional array and matrix data structures (you'll find more information about this in later sections). It provides **ndarray**, a homogeneous n-dimensional array object, with methods to efficiently operate on it. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

# ABSTRACT

Traditionally, climate assessment has been performed reliably by treating the environment as a liquid. The current wind condition is being observed. The future state of the environment is recorded by understanding thermodynamics and the numerical position of the liquid elements. Nevertheless, this traditional arrangement of differential conditions as observed by physical models is at times unstable under oscillating effects and uncertainties when estimating the underlying states of air. This indicates an insufficient understanding of environmental variations, so it limits climate forecasts to 10-day periods because climate projections are essentially unreliable. But machine learning is moderately hearty for most barometric destabilizing effects compared to traditional techniques. Another favorable position of machine learning is that it does not depend on the physical laws of environmental processes.

# BACKGROUND

For the current situation, India observatory conducts traditional weather forecasting. There are four common methods to predict the weather. The first method is the climatology method that is reviewing weather statistics gathered over multiple years and calculating the averages. The second method is an analog method that is to find a day in the past with weather similar to the current forecast. The third method is the persistence and trends method that has no skill to predict the weather because it relies on past trends. The fourth method is numerical weather prediction the is making weather predictions based on multiple conditions in the atmosphere such as temperatures, wind speed, high- and low pressure systems, rainfall, snowfall, and other conditions. So, there are many limitations of these traditional methods. Not only it forecasts the temperature in the current month at most, but also it predicts without using machine learning algorithms. Therefore, my project is to increase the accuracy and predict the weather in the future for at least one month by applying machine learning techniques.

# OBJECTIVE (PURPOSE)

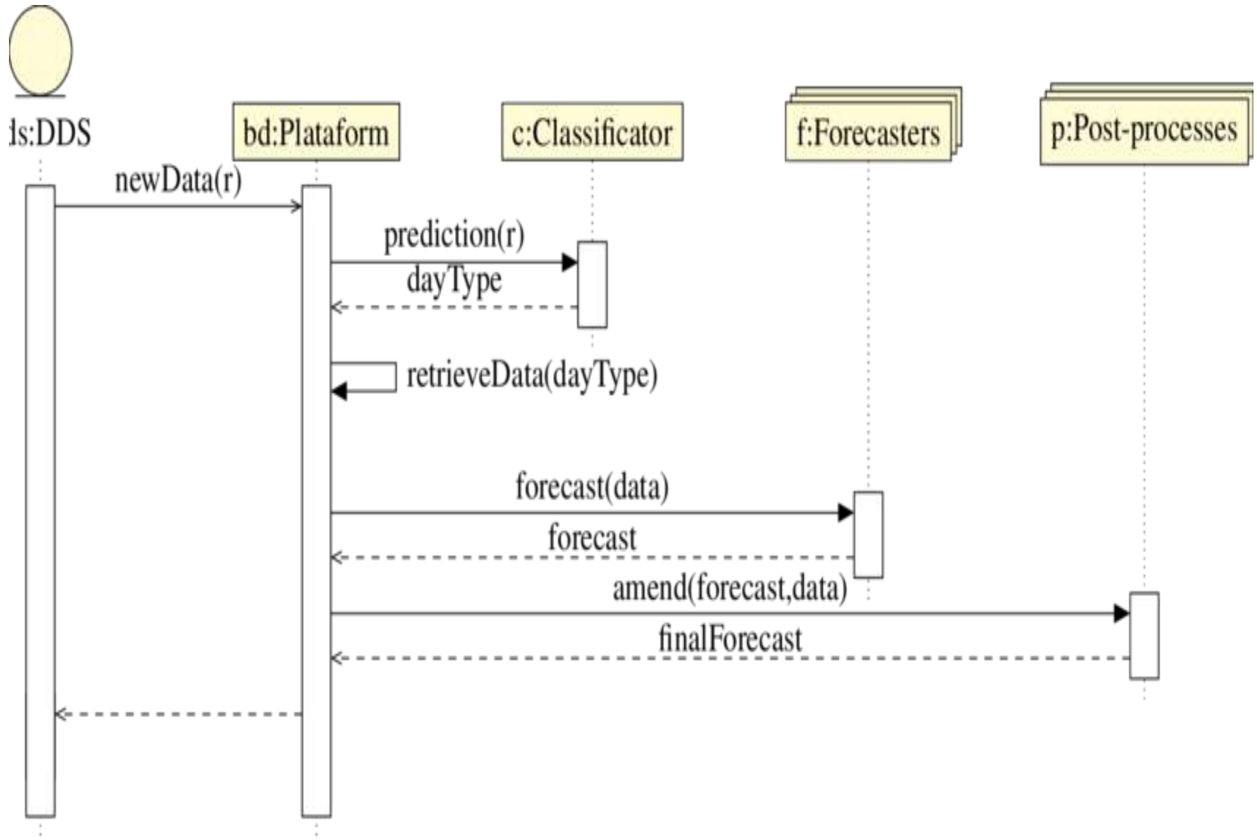
1.Purpose of this project is to predict the temperature using different algorithms like linear regression, random forest regression, and Decision tree regression.

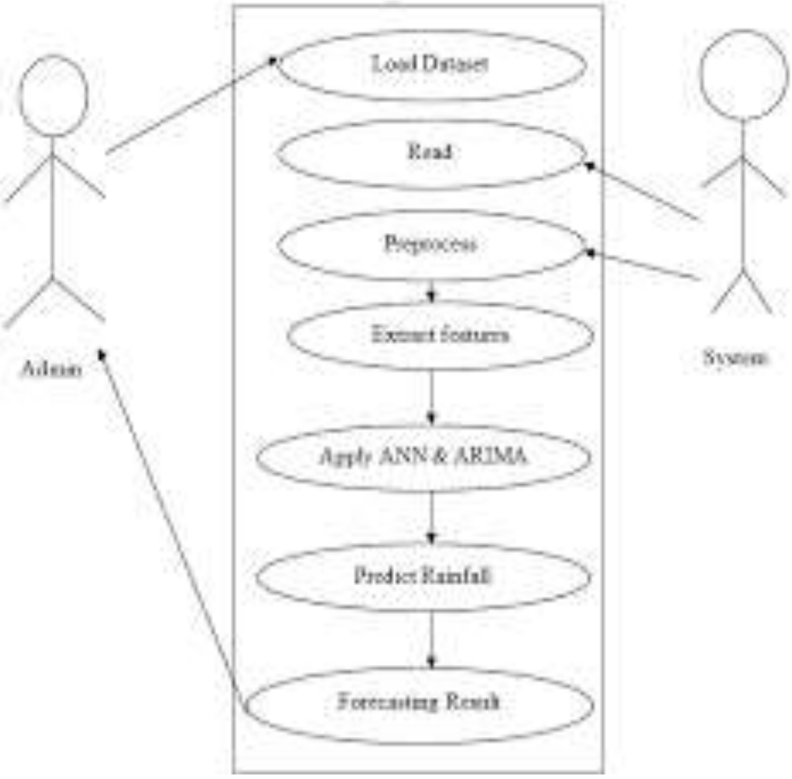
2. The output value should be numerically based on multiple extra factors like maximum temperature, minimum temperature, cloud cover, humidity, and sun hours in a day, precipitation, pressure and wind speed.

# SEQUENCE DIAGRAM:

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

Note that there are two types of sequence diagrams: UML diagrams and code-based diagrams. The latter is sourced from programming code and will not be covered in this guide. Lucidchart’s UML diagramming software is equipped with all the shapes and features you will need to model both.


ds:DDS
bd:Plataform
c:Classificator
f:Forecasters
p:Post-processes
newData(r)
prediction(r)
dayType
retrieveData(dayType)
forecast(data)
forecast
amend(forecast,data)
finalForecast


Load Dataset
Read
Preprocess
Extract features
Apply ANN & ARIMA
Predict Rainfall
Forecasting Result
Admin
System

# INTRODUCTION

Weather prediction is the task of predicting the atmosphere at a future time and a given area. This has been done through physical equations in the early days in which the atmosphere is considered fluid. The current state of the environment is inspected, and the future state is predicted by solving those equations numerically, but we cannot determine very accurate weather for more than 10 days and this can be improved with the help of science and technology. Machine learning can be used to process immediate comparisons between historical weather forecasts and observations. With the use of machine learning, weather models can better account for prediction inaccuracies, such as overestimated rainfall, and produce more accurate predictions. Temperature prediction is of major importance in a large number of applications, including climate-related studies, energy, agricultural, medical, or etc. There are numerous kinds of machine learning calculations, which are Linear Regression, Polynomial Regression, Random Forest Regression, Artificial Neural Network, and Recurrent Neural Network. These models are prepared dependent on the authentic information gave of any area. Contribution to these models is given, for example, if anticipating temperature, least temperature, mean air weight, greatest temperature, mean dampness, and order for 2 days. In light of this Minimum Temperature and Maximum Temperature of 7 days will be accomplished. Machine Learning Machine learning is relatively robust to perturbations and does not need any other physical variables for prediction. Therefore, machine learning is a much better opportunity in the evolution of weather forecasting. Before the advancement of Technology, weather forecasting was a hard nut to crack. Weather forecasters relied upon satellites, data model's atmospheric conditions with less accuracy. Weather prediction and analysis have vastly increased in terms of accuracy and predictability with the use of the Internet of Things, for the last 40 years. With the advancement of Data Science, Artificial Intelligence, Scientists now do weather forecasting with high accuracy and predictability.

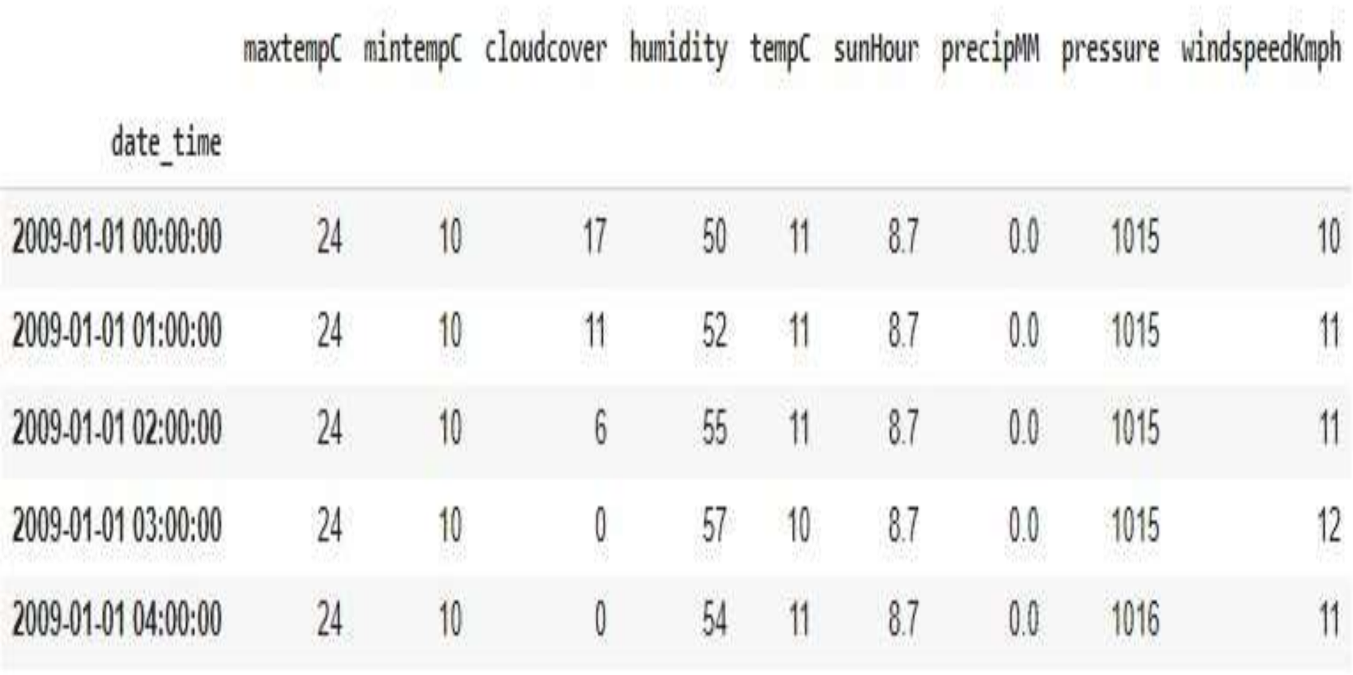
# USE OF ALGORITHMS:

There are different methods of foreseeing temperature utilizing Regression and a variety of Functional Regression, in which datasets are utilized to play out the counts and investigation. To Train, the calculations 80% size of information is utilized and 20% size of information is named as a Test set. For Example, if we need to anticipate the temperature of Kanpur, India utilizing these Machine Learning calculations, we will utilize 8 Years of information to prepare the calculations and 2 years of information as a Test dataset. The as opposed to Weather Forecasting utilizing Machine Learning Algorithms which depends essentially on reenactment dependent on Physics and Differential Equations, Artificial Intelligence is additionally utilized for foreseeing temperature: which incorporates models, for example, Linear regression, Decision tree regression, Random forest regression. To finish up, Machine Learning has enormously changed the worldview of Weather estimating with high precision and predictivity. What's more, in the following couple of years greater progression will be made utilizing these advances to precisely foresee the climate to avoid catastrophes like typhoons, Tornados, and Thunderstorms.

# METHODOLOGY

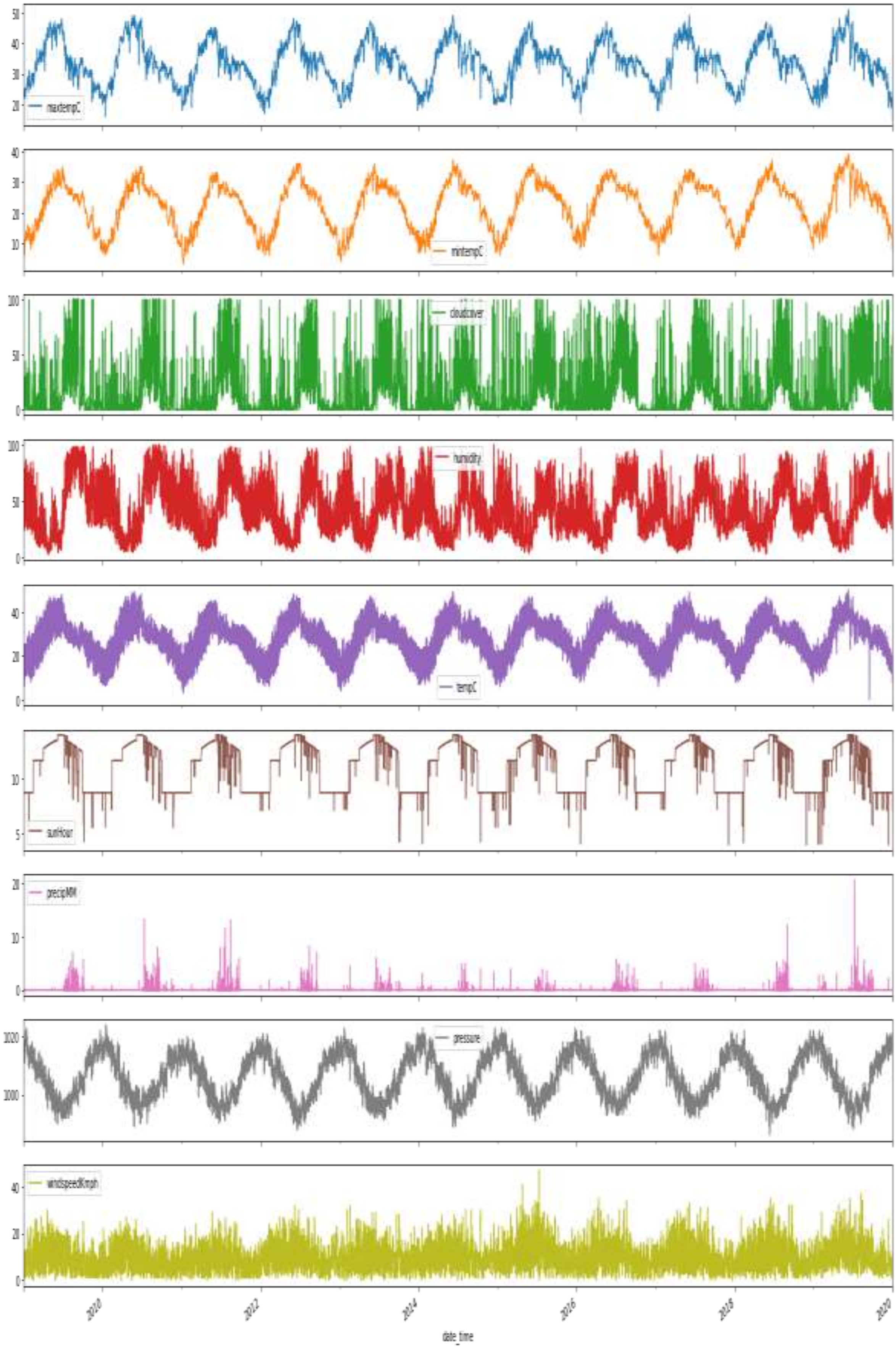
The dataset utilized in this arrangement has been gathered from Kaggle which is “Historical Weather Data for Indian Cities” from which we have chosen the data for “Kanpur City”. The dataset was created by keeping in mind the necessity of such historical weather data in the community. The datasets for the top 8 Indian cities as per the population. The dataset was used with the help of the worldweatheronline.com API and the wwo_hist package. The datasets contain hourly weather data from 01-01-2009 to 01-01-2020. The data of each city is for more than 10 years. This data can be used to visualize the change in data due to global warming or can be used to predict the weather for upcoming days, weeks, months, seasons, etc. Note: The data was extracted with the help of worldweatheronline.com API and we cannot guarantee the accuracy of the data. The main target of this dataset can be used to predict the weather for the next day or week with huge amounts of data provided in the dataset. Furthermore, this data can also be used to make visualization which would help to understand the impact of global warming over the various aspects of the weather like precipitation, humidity, temperature, etc. In this project, we are concentrating on the temperature prediction of Kanpur city with the help of various machine learning algorithms and various regressions. By applying various regressions on the historical weather dataset of Kanpur city we are predicting the temperature like first we are applying Multiple Linear regression, then Decision Tree regression, and after that, we are applying Random Forest Regression.

Table 2.1: Historical Weather Dataset of Kanpur City



| | maxtempC | mintempC | cloudcover | humidity | tempC | sunHour | precipMM | pressure | windspeedKmph |
|---|---|---|---|---|---|---|---|---|---|
| date_time | | | | | | | | | |
| 2009-01-01 00:00:00 | 24 | 10 | 17 | 50 | 11 | 8.7 | 0.0 | 1015 | 10 |
| 2009-01-01 01:00:00 | 24 | 10 | 11 | 52 | 11 | 8.7 | 0.0 | 1015 | 11 |
| 2009-01-01 02:00:00 | 24 | 10 | 6 | 55 | 11 | 8.7 | 0.0 | 1015 | 11 |
| 2009-01-01 03:00:00 | 24 | 10 | 0 | 57 | 10 | 8.7 | 0.0 | 1015 | 12 |
| 2009-01-01 04:00:00 | 24 | 10 | 0 | 54 | 11 | 8.7 | 0.0 | 1016 | 11 |

Figure 2.1: Plot for each factor for 10 years



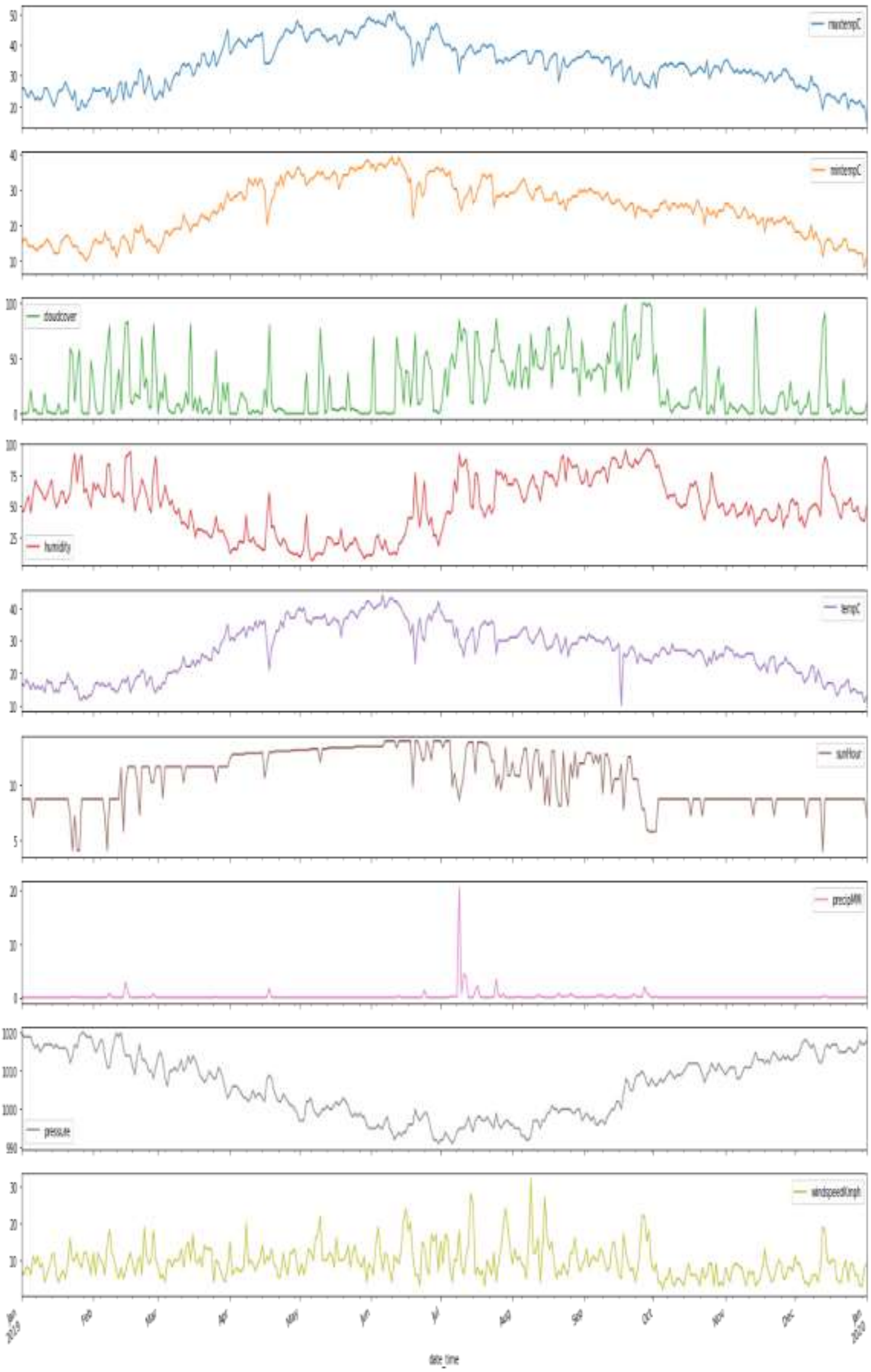
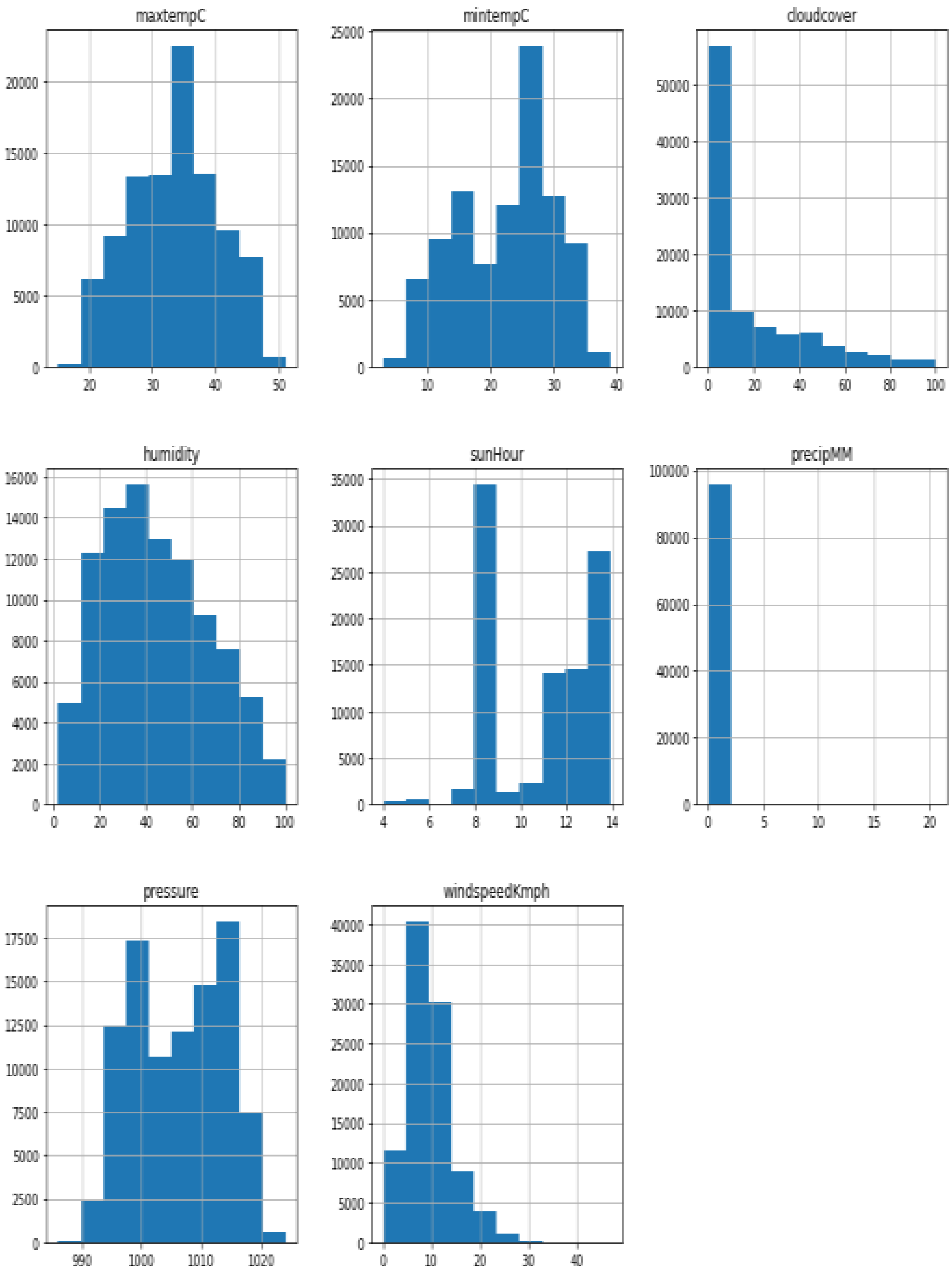


Figure 2.2: Plot for each factor for 1 year

# EXPERIMENTATION

The record has just been separated into a train set and a test set. Each information has just been labeled. First, we take the trainset organizer. We will train our model with the help of histograms and plots. The feature so extracted is stored in a histogram. This process is done for every data in the train set. Now we will build the model of our classifiers. The classifiers which we will take into account are Linear Regression, Decision Tree Regression, and Random Forest Regression. With the help of our histogram, we will train our model. The most important thing in this process is to tune these parameters accordingly, such that we get the most accurate results.  Once the training is complete, we will take the test set. Now for each data variable of the test set, we will extract the features using feature extraction techniques and then compare its values with the values present in the histogram formed by the train set. The output is then predicted for each test day. Now in order to calculate accuracy, we will compare the predicted value with the labeled value. The different metrics that we will use confusion matrix, R2 score, etc.



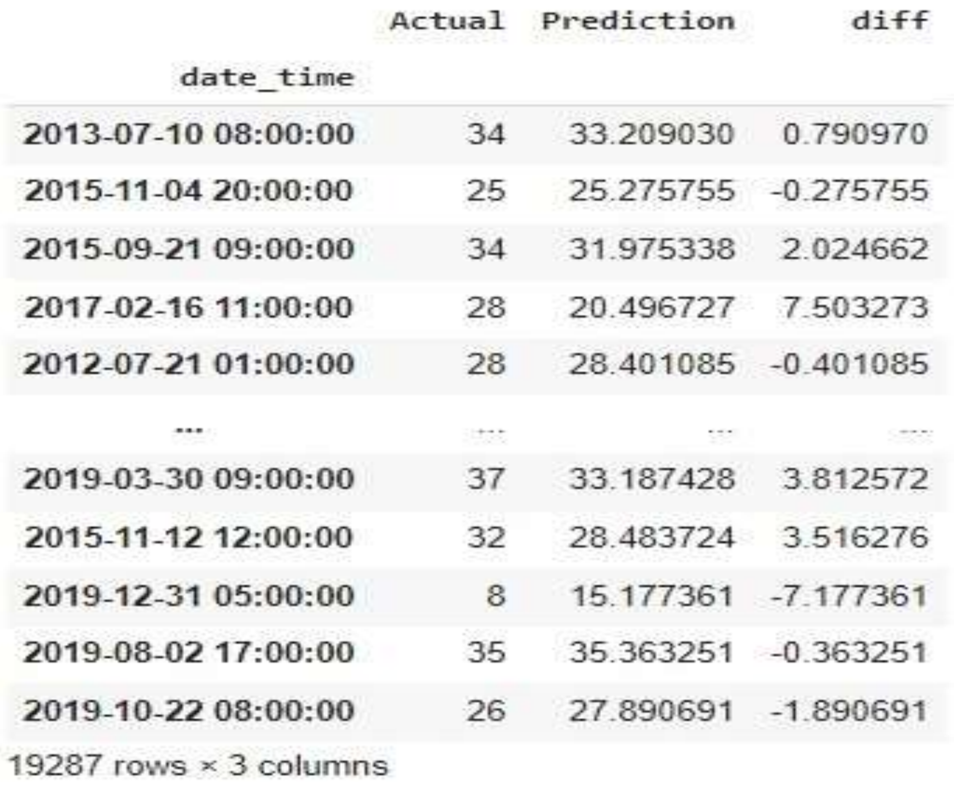
# RESULT AND DISCUSSION

The results of the implementation of the project are demonstrated below.

## Multiple Linear Regression:

This regression model has high mean absolute error, hence turned out to be the least accurate model. Given below is a snapshot of the actual result from the project implementation of multiple linear regression.

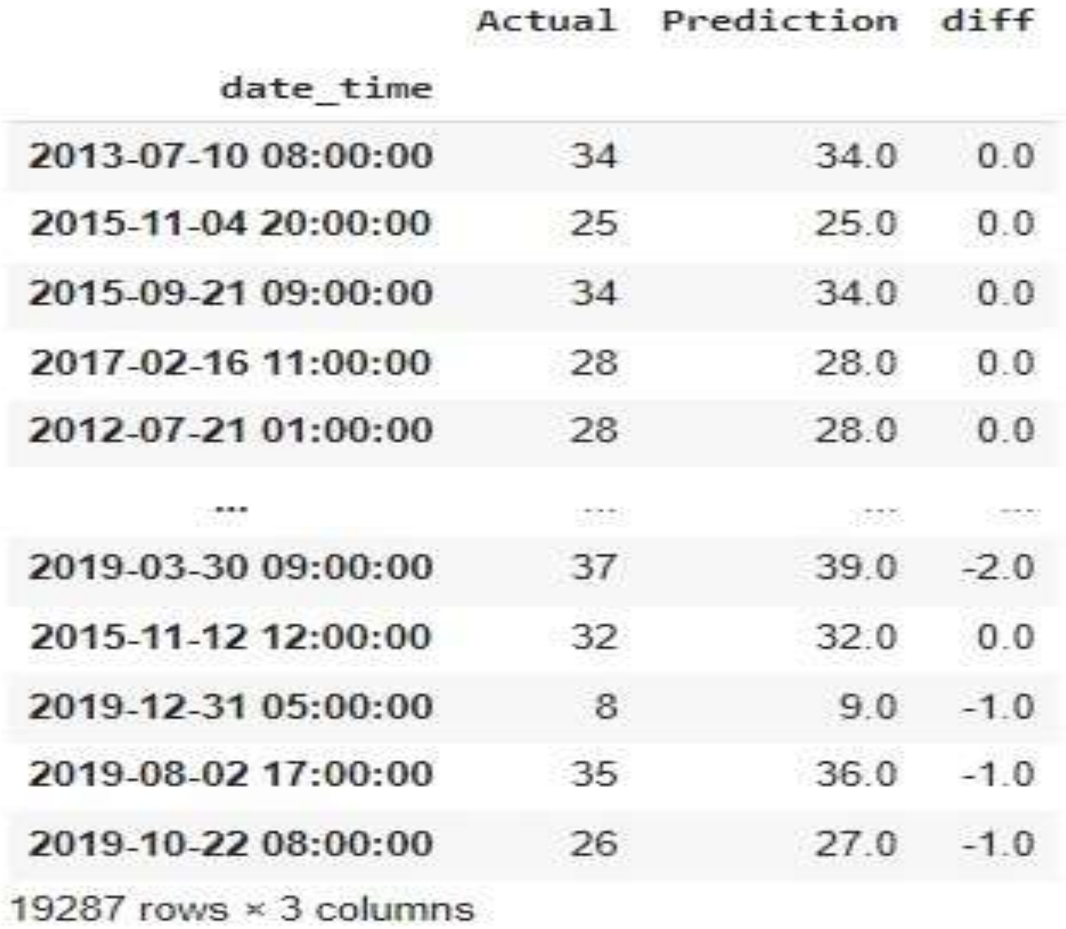
Multiple Linear Regression attempts to model the relationship between two or more features and a response by fitting a linear equation to observed data. The steps to perform multiple linear Regression are almost similar to that of simple linear Regression. The Difference Lies in the evaluation. We can use it to find out which factor has the highest impact on the predicted output and how different variables relate to each other. Here: Y = b0 + b1 * x1 + b2 * x2 + b3 * x3 + …… bn * xn Y = Dependent variable and x1, x2, x3, …… xn = multiple independent variables Assumption of Regression Model: Linearity: The relationship between dependent and independent variables should be linear. Homoscedasticity: Constant variance of the errors should be maintained. Multivariate normality: Multiple Regression assumes that the residuals are normally distributed. Lack of Multicollinearity: It is assumed that there is little or no multicollinearity in the data.



```
                     Actual  Prediction      diff
date_time
2013-07-10 08:00:00      34   33.209030  0.790970
2015-11-04 20:00:00      25   25.275755 -0.275755
2015-09-21 09:00:00      34   31.975338  2.024662
2017-02-16 11:00:00      28   20.496727  7.503273
2012-07-21 01:00:00      28   28.401085 -0.401085
        ...             ...         ...       ...
2019-03-30 09:00:00      37   33.187428  3.812572
2015-11-12 12:00:00      32   28.483724  3.516276
2019-12-31 05:00:00       8   15.177361 -7.177361
2019-08-02 17:00:00      35   35.363251 -0.363251
2019-10-22 08:00:00      26   27.890691 -1.890691
19287 rows × 3 columns
```

## Decision Tree Regression:

This regression model has medium mean absolute error, hence turned out to be the little accurate model. Given below is a snapshot of the actual result from the project implementation of multiple linear regression. Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.

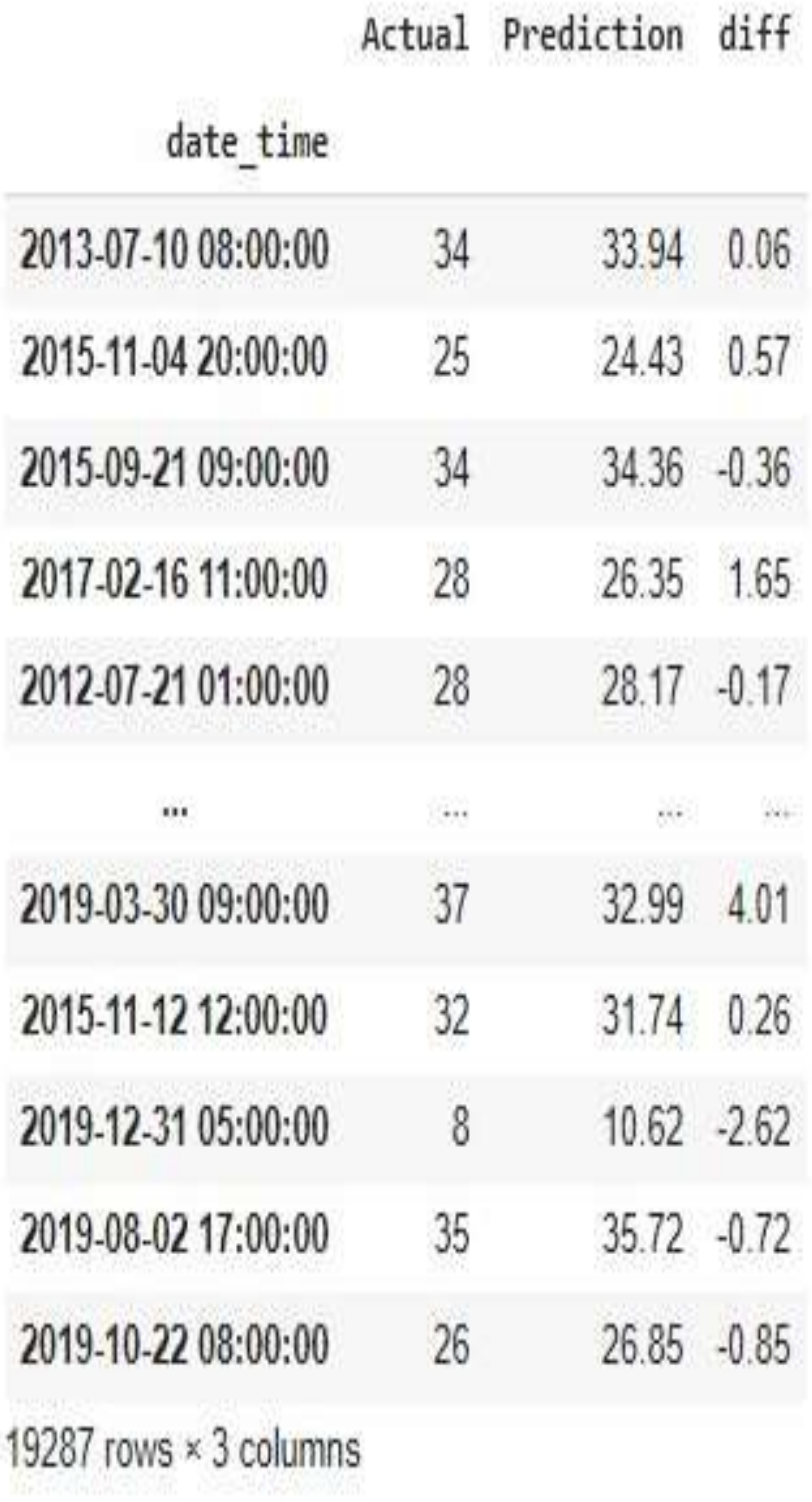


| | Actual | Prediction | diff |
|---|---|---|---|
| date_time | | | |
| 2013-07-10 08:00:00 | 34 | 34.0 | 0.0 |
| 2015-11-04 20:00:00 | 25 | 25.0 | 0.0 |
| 2015-09-21 09:00:00 | 34 | 34.0 | 0.0 |
| 2017-02-16 11:00:00 | 28 | 28.0 | 0.0 |
| 2012-07-21 01:00:00 | 28 | 28.0 | 0.0 |
| ... | ... | ... | ... |
| 2019-03-30 09:00:00 | 37 | 39.0 | -2.0 |
| 2015-11-12 12:00:00 | 32 | 32.0 | 0.0 |
| 2019-12-31 05:00:00 | 8 | 9.0 | -1.0 |
| 2019-08-02 17:00:00 | 35 | 36.0 | -1.0 |
| 2019-10-22 08:00:00 | 26 | 27.0 | -1.0 |

19287 rows × 3 columns

## Random Forest Regression:

This regression model has low mean absolute error, hence turned out to be the more accurate model. Given below is a snapshot of the actual result from the project implementation of multiple linear regression.



| | Actual | Prediction | diff |
|---|---|---|---|
| date_time | | | |
| 2013-07-10 08:00:00 | 34 | 33.94 | 0.06 |
| 2015-11-04 20:00:00 | 25 | 24.43 | 0.57 |
| 2015-09-21 09:00:00 | 34 | 34.36 | -0.36 |
| 2017-02-16 11:00:00 | 28 | 26.35 | 1.65 |
| 2012-07-21 01:00:00 | 28 | 28.17 | -0.17 |
| ... | ... | ... | ... |
| 2019-03-30 09:00:00 | 37 | 32.99 | 4.01 |
| 2015-11-12 12:00:00 | 32 | 31.74 | 0.26 |
| 2019-12-31 05:00:00 | 8 | 10.62 | -2.62 |
| 2019-08-02 17:00:00 | 35 | 35.72 | -0.72 |
| 2019-10-22 08:00:00 | 26 | 26.85 | -0.85 |

19287 rows × 3 columns

# CONCLUSION

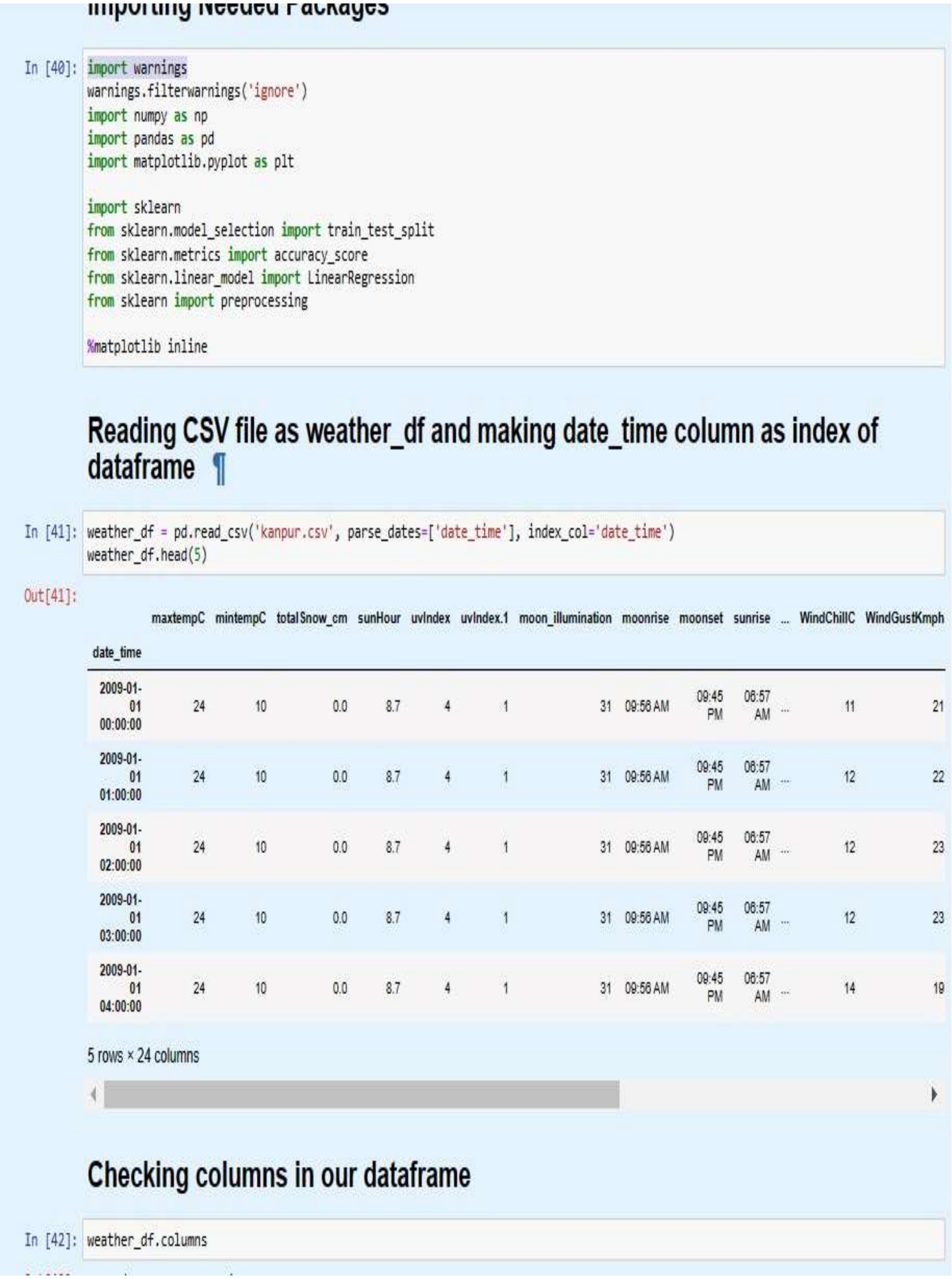
All the machine learning models: linear regression, various linear regression, decision tree regression, random forest regression were beaten by expert climate determining apparatuses, even though the error in their execution reduced significantly for later days, demonstrating that over longer timeframes, our models may beat genius professional ones.

Linear regression demonstrated to be a low predisposition, high fluctuation model though polynomial regression demonstrated to be a high predisposition, low difference model. Linear regression is naturally a high difference model as it is unsteady to outliers, so one approach to improve the linear regression model is by gathering more information. Practical regression, however, was high predisposition, demonstrating that the decision of the model was poor and that its predictions can't be improved by the further accumulation of information. This predisposition could be expected to the structure decision to estimate temperature dependent on the climate of the previous two days, which might be too short to even think about capturing slants in a climate that practical regression requires. On the off chance that the figure was rather founded on the climate of the past four or five days, the predisposition of the practical regression model could probably be decreased. In any case, this would require significantly more calculation time alongside retraining of the weight vector w, so this will be conceded to future work.

Talking about Random Forest Regression, it proves to be the most accurate regression model. Likely so, it is the most popular regression model used, since it is highly accurate and versatile. Below is a snapshot of the implementation of Random Forest in the project.

Weather Forecasting has a major test of foreseeing the precise outcomes which are utilized in numerous ongoing frameworks like power offices, air terminals, the travel industry focuses, and so forth. The trouble of this determining is the mind-boggling nature of parameters. Every parameter has an alternate arrangement of scope qualities.

# IMPORTANT CODING AND OUTPUT:



## Importing Needed Packages

In [40]:
```
import warnings
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import sklearn
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LinearRegression
from sklearn import preprocessing

%matplotlib inline
```

## Reading CSV file as weather_df and making date_time column as index of dataframe ¶

In [41]:
```
weather_df = pd.read_csv('kanpur.csv', parse_dates=['date_time'], index_col='date_time')
weather_df.head(5)
```

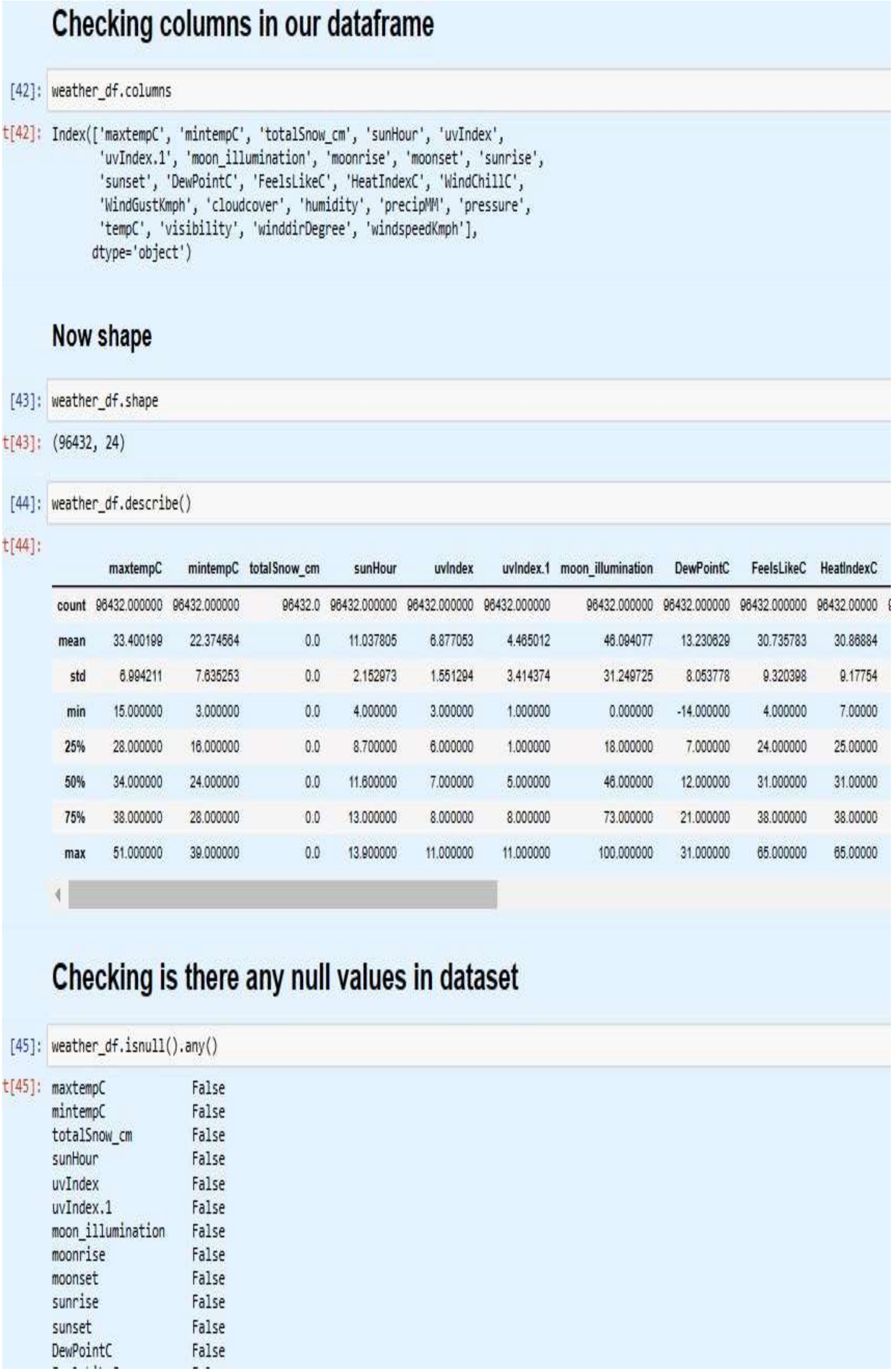
Out[41]:

| | maxtempC | mintempC | totalSnow_cm | sunHour | uvIndex | uvIndex.1 | moon_illumination | moonrise | moonset | sunrise | ... | WindChillC | WindGustKmph |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| date_time | | | | | | | | | | | | | |
| 2009-01-01 00:00:00 | 24 | 10 | 0.0 | 8.7 | 4 | 1 | 31 | 09:56 AM | 09:45 PM | 06:57 AM | ... | 11 | 21 |
| 2009-01-01 01:00:00 | 24 | 10 | 0.0 | 8.7 | 4 | 1 | 31 | 09:56 AM | 09:45 PM | 06:57 AM | ... | 12 | 22 |
| 2009-01-01 02:00:00 | 24 | 10 | 0.0 | 8.7 | 4 | 1 | 31 | 09:56 AM | 09:45 PM | 06:57 AM | ... | 12 | 23 |
| 2009-01-01 03:00:00 | 24 | 10 | 0.0 | 8.7 | 4 | 1 | 31 | 09:56 AM | 09:45 PM | 06:57 AM | ... | 12 | 23 |
| 2009-01-01 04:00:00 | 24 | 10 | 0.0 | 8.7 | 4 | 1 | 31 | 09:56 AM | 09:45 PM | 06:57 AM | ... | 14 | 19 |

5 rows × 24 columns

## Checking columns in our dataframe

In [42]:
```
weather_df.columns
```



## Checking columns in our dataframe

```
[42]: weather_df.columns
```

```
t[42]: Index(['maxtempC', 'mintempC', 'totalSnow_cm', 'sunHour', 'uvIndex',
              'uvIndex.1', 'moon_illumination', 'moonrise', 'moonset', 'sunrise',
              'sunset', 'DewPointC', 'FeelsLikeC', 'HeatIndexC', 'WindChillC',
              'WindGustKmph', 'cloudcover', 'humidity', 'precipMM', 'pressure',
              'tempC', 'visibility', 'winddirDegree', 'windspeedKmph'],
             dtype='object')
```

## Now shape

```
[43]: weather_df.shape
```

```
t[43]: (96432, 24)
```

```
[44]: weather_df.describe()
```

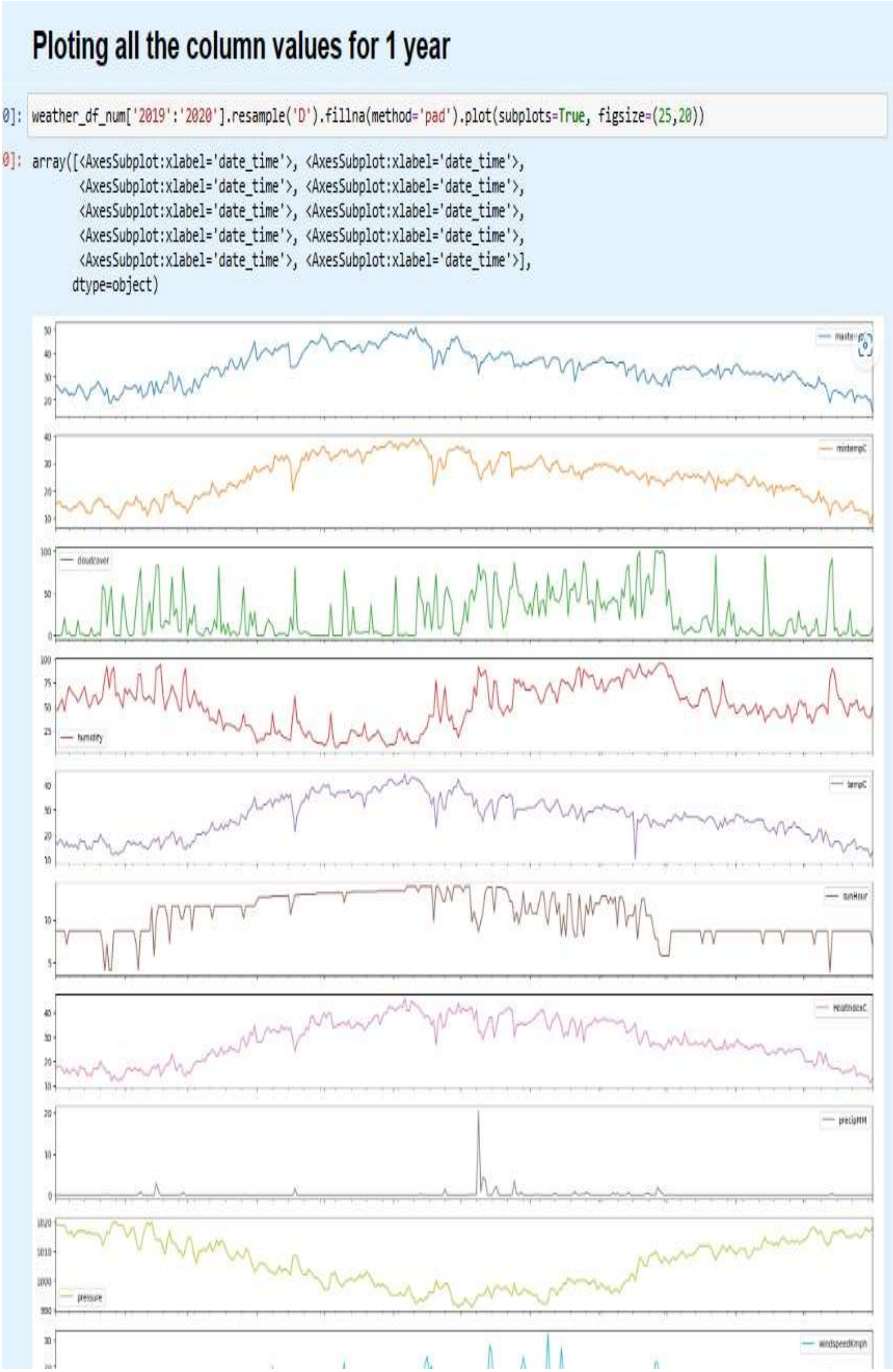
t[44]:

| | maxtempC | mintempC | totalSnow_cm | sunHour | uvIndex | uvIndex.1 | moon_illumination | DewPointC | FeelsLikeC | HeatIndexC |
|---|---|---|---|---|---|---|---|---|---|---|
| count | 96432.000000 | 96432.000000 | 96432.0 | 96432.000000 | 96432.000000 | 96432.000000 | 96432.000000 | 96432.000000 | 96432.000000 | 96432.00000 |
| mean | 33.400199 | 22.374564 | 0.0 | 11.037805 | 6.877053 | 4.465012 | 46.094077 | 13.230629 | 30.735783 | 30.86884 |
| std | 6.994211 | 7.635253 | 0.0 | 2.152973 | 1.551294 | 3.414374 | 31.249725 | 8.053778 | 9.320398 | 9.17754 |
| min | 15.000000 | 3.000000 | 0.0 | 4.000000 | 3.000000 | 1.000000 | 0.000000 | -14.000000 | 4.000000 | 7.00000 |
| 25% | 28.000000 | 16.000000 | 0.0 | 8.700000 | 6.000000 | 1.000000 | 18.000000 | 7.000000 | 24.000000 | 25.00000 |
| 50% | 34.000000 | 24.000000 | 0.0 | 11.600000 | 7.000000 | 5.000000 | 46.000000 | 12.000000 | 31.000000 | 31.00000 |
| 75% | 38.000000 | 28.000000 | 0.0 | 13.000000 | 8.000000 | 8.000000 | 73.000000 | 21.000000 | 38.000000 | 38.00000 |
| max | 51.000000 | 39.000000 | 0.0 | 13.900000 | 11.000000 | 11.000000 | 100.000000 | 31.000000 | 65.000000 | 65.00000 |

## Checking is there any null values in dataset

```
[45]: weather_df.isnull().any()
```

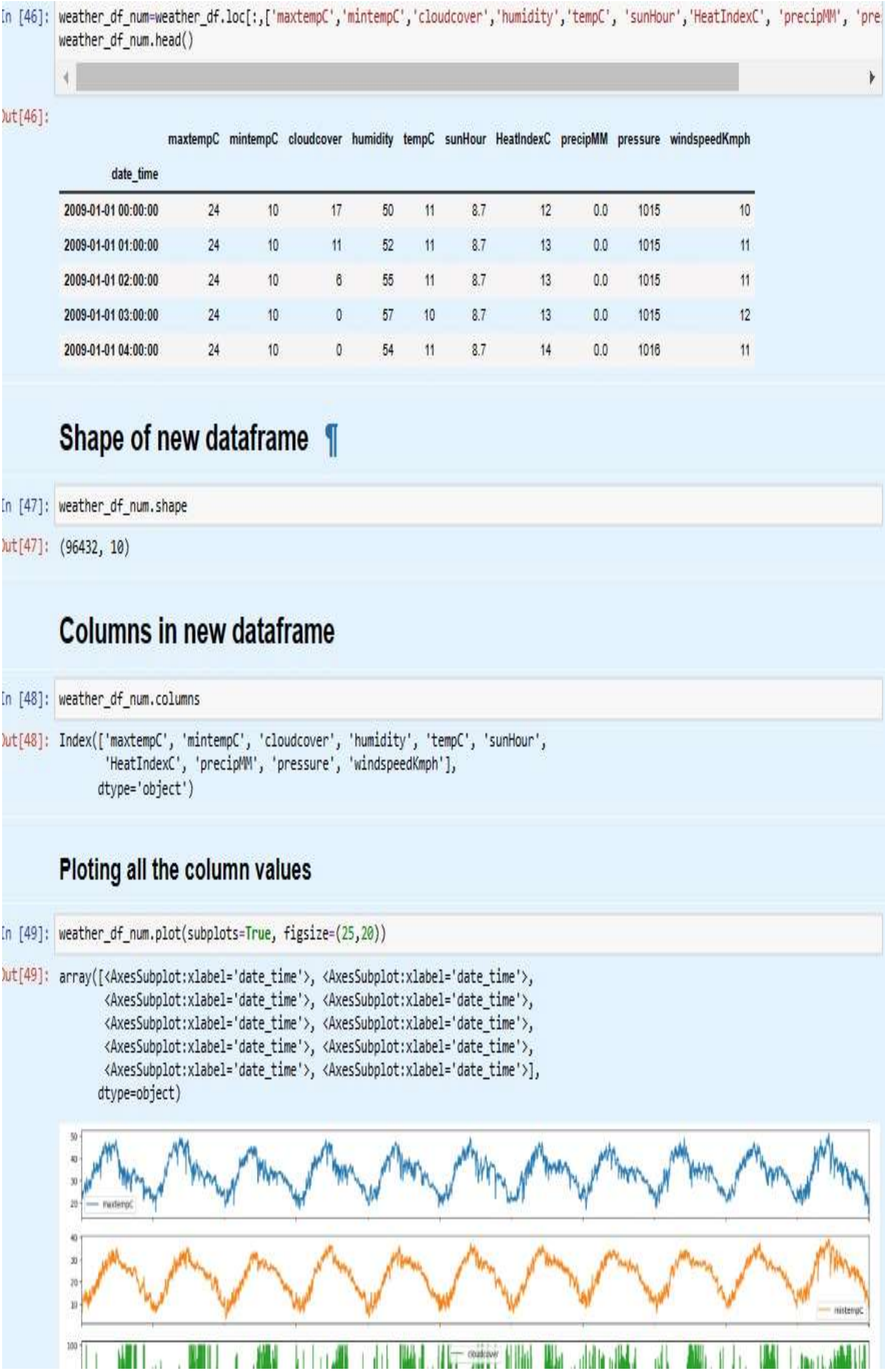
```
t[45]: maxtempC             False
       mintempC             False
       totalSnow_cm         False
       sunHour              False
       uvIndex              False
       uvIndex.1            False
       moon_illumination    False
       moonrise             False
       moonset              False
       sunrise              False
       sunset               False
       DewPointC            False
```



# Ploting all the column values for 1 year

```
0]: weather_df_num['2019':'2020'].resample('D').fillna(method='pad').plot(subplots=True, figsize=(25,20))
```

```
0]: array([<AxesSubplot:xlabel='date_time'>, <AxesSubplot:xlabel='date_time'>,
           <AxesSubplot:xlabel='date_time'>, <AxesSubplot:xlabel='date_time'>,
           <AxesSubplot:xlabel='date_time'>, <AxesSubplot:xlabel='date_time'>,
           <AxesSubplot:xlabel='date_time'>, <AxesSubplot:xlabel='date_time'>,
           <AxesSubplot:xlabel='date_time'>, <AxesSubplot:xlabel='date_time'>],
          dtype=object)
```



```
In [46]: weather_df_num=weather_df.loc[:,['maxtempC','mintempC','cloudcover','humidity','tempC', 'sunHour','HeatIndexC', 'precipMM', 'pre
         weather_df_num.head()
```

Out[46]:

| | maxtempC | mintempC | cloudcover | humidity | tempC | sunHour | HeatIndexC | precipMM | pressure | windspeedKmph |
|---|---|---|---|---|---|---|---|---|---|---|
| date_time | | | | | | | | | | |
| 2009-01-01 00:00:00 | 24 | 10 | 17 | 50 | 11 | 8.7 | 12 | 0.0 | 1015 | 10 |
| 2009-01-01 01:00:00 | 24 | 10 | 11 | 52 | 11 | 8.7 | 13 | 0.0 | 1015 | 11 |
| 2009-01-01 02:00:00 | 24 | 10 | 6 | 55 | 11 | 8.7 | 13 | 0.0 | 1015 | 11 |
| 2009-01-01 03:00:00 | 24 | 10 | 0 | 57 | 10 | 8.7 | 13 | 0.0 | 1015 | 12 |
| 2009-01-01 04:00:00 | 24 | 10 | 0 | 54 | 11 | 8.7 | 14 | 0.0 | 1016 | 11 |

## Shape of new dataframe ¶

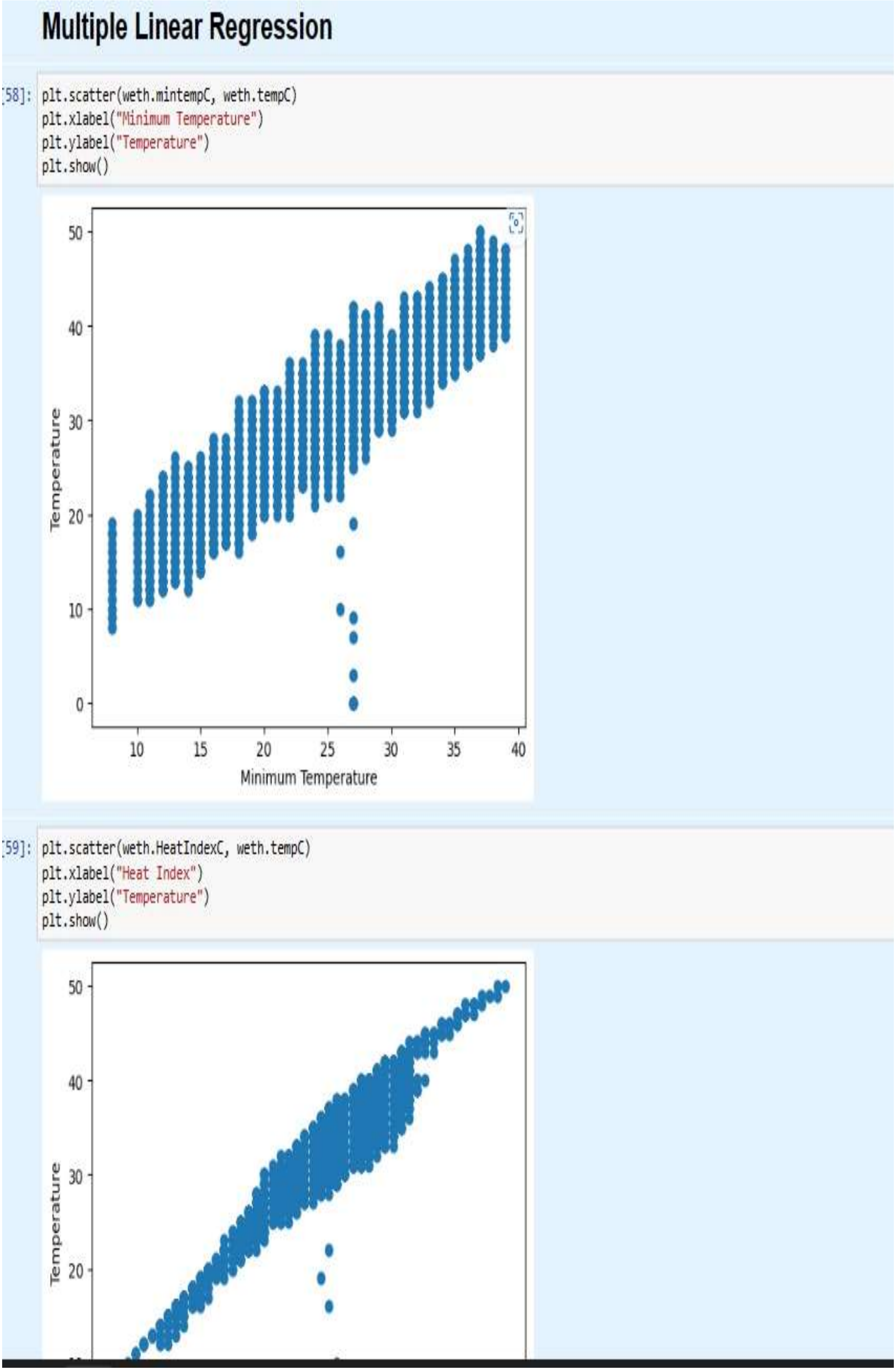
```
In [47]: weather_df_num.shape

Out[47]: (96432, 10)
```

## Columns in new dataframe

```
In [48]: weather_df_num.columns

Out[48]: Index(['maxtempC', 'mintempC', 'cloudcover', 'humidity', 'tempC', 'sunHour',
                'HeatIndexC', 'precipMM', 'pressure', 'windspeedKmph'],
               dtype='object')
```

### Ploting all the column values

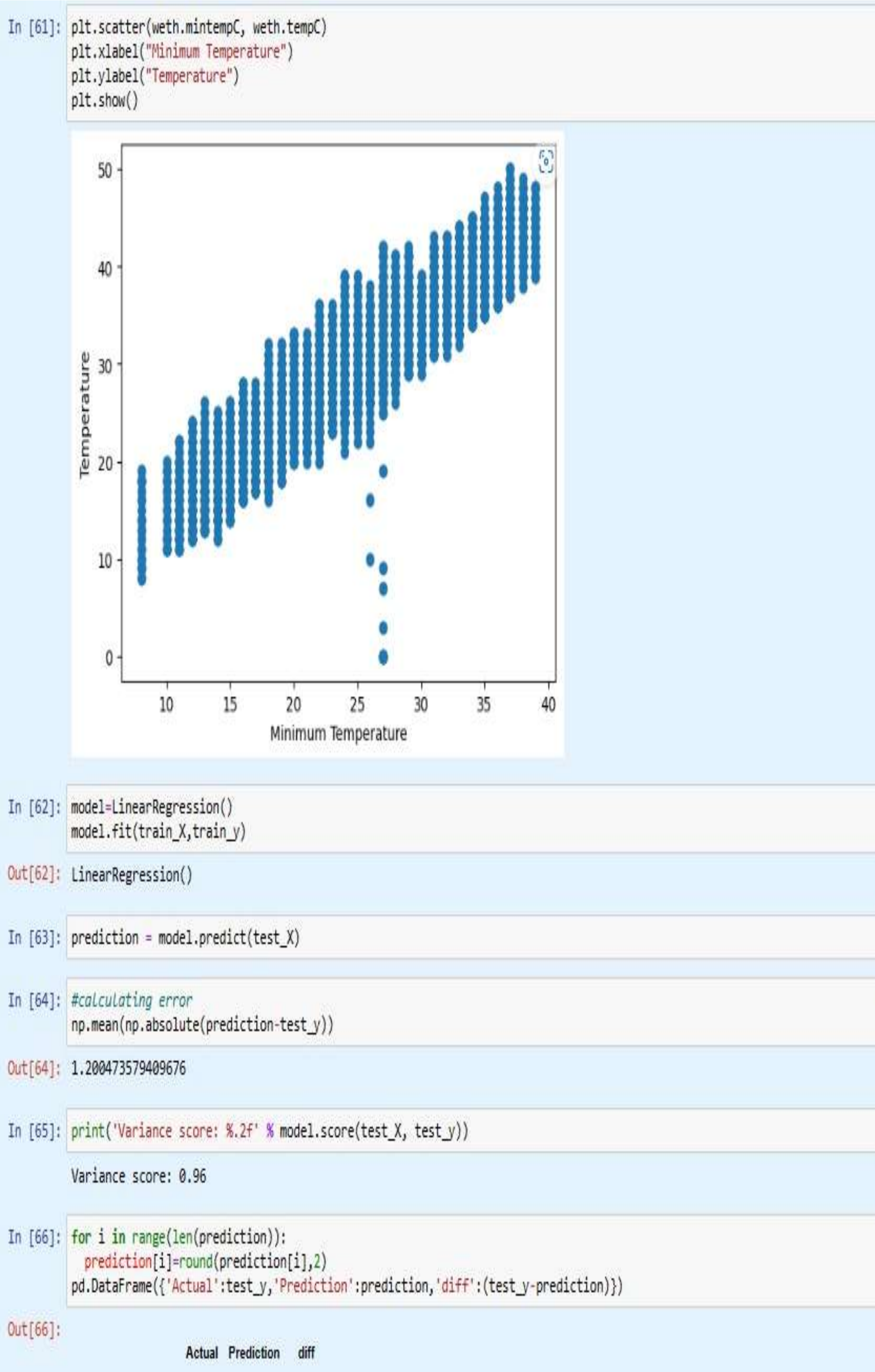
```
In [49]: weather_df_num.plot(subplots=True, figsize=(25,20))

Out[49]: array([<AxesSubplot:xlabel='date_time'>, <AxesSubplot:xlabel='date_time'>,
                <AxesSubplot:xlabel='date_time'>, <AxesSubplot:xlabel='date_time'>,
                <AxesSubplot:xlabel='date_time'>, <AxesSubplot:xlabel='date_time'>,
                <AxesSubplot:xlabel='date_time'>, <AxesSubplot:xlabel='date_time'>,
                <AxesSubplot:xlabel='date_time'>, <AxesSubplot:xlabel='date_time'>],
               dtype=object)
```



## Multiple Linear Regression

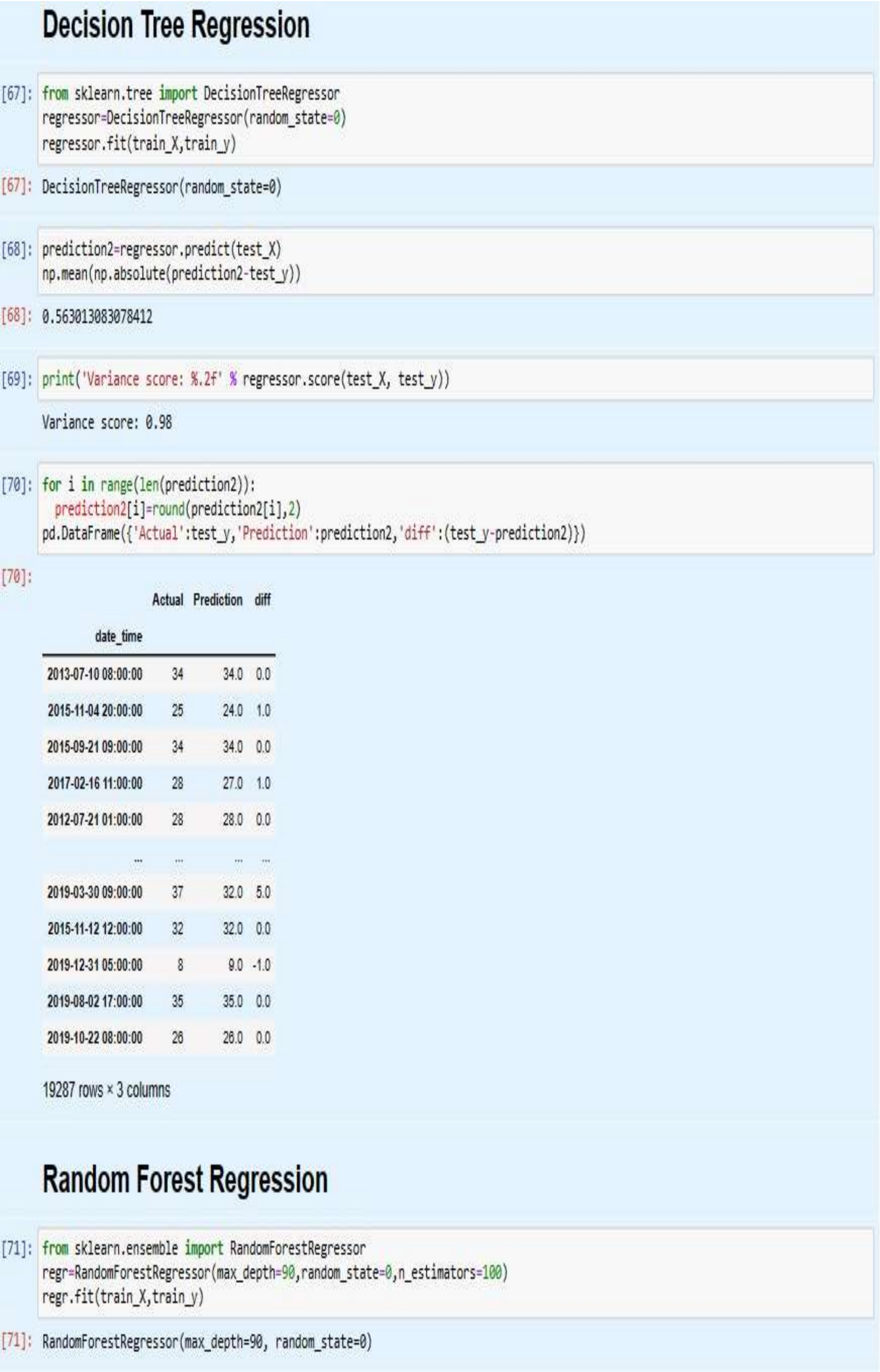
```
[58]: plt.scatter(weth.mintempC, weth.tempC)
      plt.xlabel("Minimum Temperature")
      plt.ylabel("Temperature")
      plt.show()
```

```
[59]: plt.scatter(weth.HeatIndexC, weth.tempC)
      plt.xlabel("Heat Index")
      plt.ylabel("Temperature")
      plt.show()
```



```
In [61]: plt.scatter(weth.mintempC, weth.tempC)
         plt.xlabel("Minimum Temperature")
         plt.ylabel("Temperature")
         plt.show()
```

```
In [62]: model=LinearRegression()
         model.fit(train_X,train_y)

Out[62]: LinearRegression()

In [63]: prediction = model.predict(test_X)

In [64]: #calculating error
         np.mean(np.absolute(prediction-test_y))

Out[64]: 1.200473579409676

In [65]: print('Variance score: %.2f' % model.score(test_X, test_y))

         Variance score: 0.96

In [66]: for i in range(len(prediction)):
           prediction[i]=round(prediction[i],2)
         pd.DataFrame({'Actual':test_y,'Prediction':prediction,'diff':(test_y-prediction)})

Out[66]:
                    Actual  Prediction   diff
```



## Decision Tree Regression

```
[67]: from sklearn.tree import DecisionTreeRegressor
      regressor=DecisionTreeRegressor(random_state=0)
      regressor.fit(train_X,train_y)

[67]: DecisionTreeRegressor(random_state=0)
```

```
[68]: prediction2=regressor.predict(test_X)
      np.mean(np.absolute(prediction2-test_y))

[68]: 0.563013083078412
```

```
[69]: print('Variance score: %.2f' % regressor.score(test_X, test_y))

      Variance score: 0.98
```

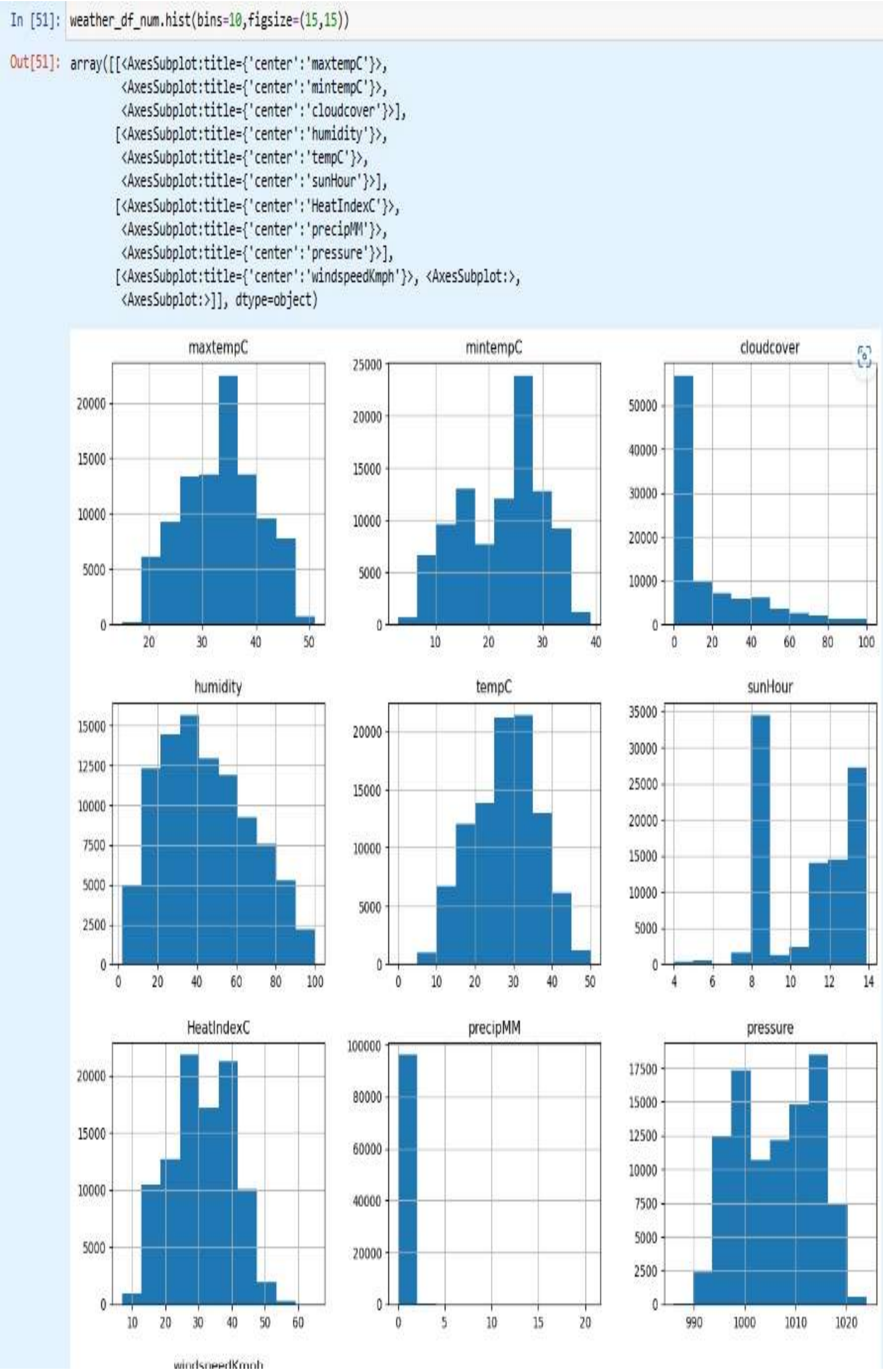
```
[70]: for i in range(len(prediction2)):
        prediction2[i]=round(prediction2[i],2)
      pd.DataFrame({'Actual':test_y,'Prediction':prediction2,'diff':(test_y-prediction2)})

[70]:
```

|  | Actual | Prediction | diff |
|---|---|---|---|
| date_time |  |  |  |
| 2013-07-10 08:00:00 | 34 | 34.0 | 0.0 |
| 2015-11-04 20:00:00 | 25 | 24.0 | 1.0 |
| 2015-09-21 09:00:00 | 34 | 34.0 | 0.0 |
| 2017-02-16 11:00:00 | 28 | 27.0 | 1.0 |
| 2012-07-21 01:00:00 | 28 | 28.0 | 0.0 |
| ... | ... | ... | ... |
| 2019-03-30 09:00:00 | 37 | 32.0 | 5.0 |
| 2015-11-12 12:00:00 | 32 | 32.0 | 0.0 |
| 2019-12-31 05:00:00 | 8 | 9.0 | -1.0 |
| 2019-08-02 17:00:00 | 35 | 35.0 | 0.0 |
| 2019-10-22 08:00:00 | 26 | 26.0 | 0.0 |

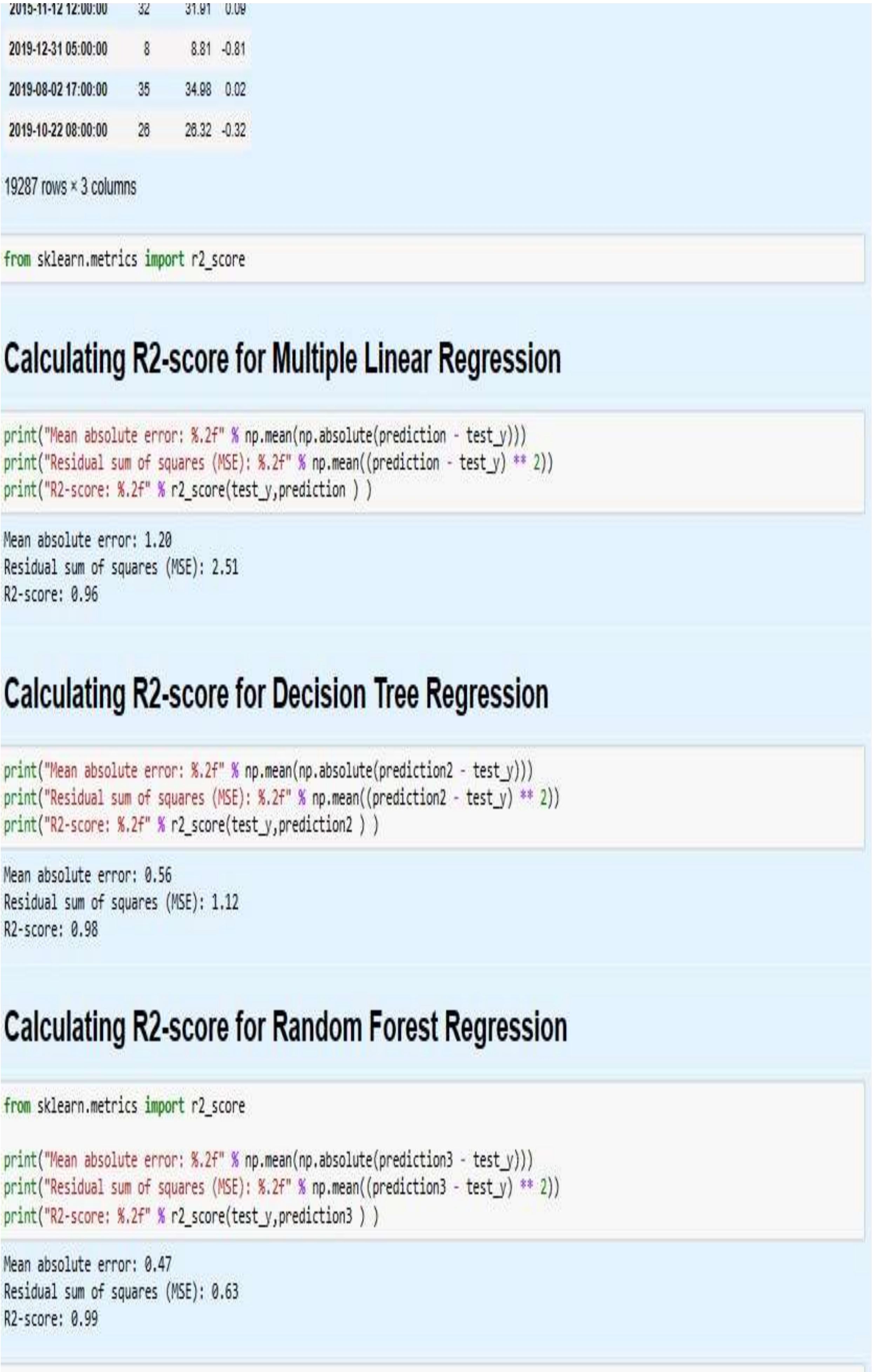
19287 rows × 3 columns

## Random Forest Regression

```
[71]: from sklearn.ensemble import RandomForestRegressor
      regr=RandomForestRegressor(max_depth=90,random_state=0,n_estimators=100)
      regr.fit(train_X,train_y)

[71]: RandomForestRegressor(max_depth=90, random_state=0)
```



```
In [51]: weather_df_num.hist(bins=10,figsize=(15,15))

Out[51]: array([[<AxesSubplot:title={'center':'maxtempC'}>,
                 <AxesSubplot:title={'center':'mintempC'}>,
                 <AxesSubplot:title={'center':'cloudcover'}>],
                [<AxesSubplot:title={'center':'humidity'}>,
                 <AxesSubplot:title={'center':'tempC'}>,
                 <AxesSubplot:title={'center':'sunHour'}>],
                [<AxesSubplot:title={'center':'HeatIndexC'}>,
                 <AxesSubplot:title={'center':'precipMM'}>,
                 <AxesSubplot:title={'center':'pressure'}>],
                [<AxesSubplot:title={'center':'windspeedKmph'}>, <AxesSubplot:>,
                 <AxesSubplot:>]], dtype=object)
```



| | | | |
|---|---|---|---|
| 2019-11-12 12:00:00 | 32 | 31.91 | 0.09 |
| 2019-12-31 05:00:00 | 8 | 8.81 | -0.81 |
| 2019-08-02 17:00:00 | 35 | 34.98 | 0.02 |
| 2019-10-22 08:00:00 | 26 | 26.32 | -0.32 |

19287 rows × 3 columns

```
from sklearn.metrics import r2_score
```

## Calculating R2-score for Multiple Linear Regression

```
print("Mean absolute error: %.2f" % np.mean(np.absolute(prediction - test_y)))
print("Residual sum of squares (MSE): %.2f" % np.mean((prediction - test_y) ** 2))
print("R2-score: %.2f" % r2_score(test_y,prediction ) )
```

```
Mean absolute error: 1.20
Residual sum of squares (MSE): 2.51
R2-score: 0.96
```

## Calculating R2-score for Decision Tree Regression

```
print("Mean absolute error: %.2f" % np.mean(np.absolute(prediction2 - test_y)))
print("Residual sum of squares (MSE): %.2f" % np.mean((prediction2 - test_y) ** 2))
print("R2-score: %.2f" % r2_score(test_y,prediction2 ) )
```

```
Mean absolute error: 0.56
Residual sum of squares (MSE): 1.12
R2-score: 0.98
```

## Calculating R2-score for Random Forest Regression

```
from sklearn.metrics import r2_score

print("Mean absolute error: %.2f" % np.mean(np.absolute(prediction3 - test_y)))
print("Residual sum of squares (MSE): %.2f" % np.mean((prediction3 - test_y) ** 2))
print("R2-score: %.2f" % r2_score(test_y,prediction3 ) )
```

```
Mean absolute error: 0.47
Residual sum of squares (MSE): 0.63
R2-score: 0.99
```

# FUTURE ENHANCEMENTS:

We can estimate the weather events using a machine learning model that takes into account the different weather parameters. In this paper, we presented different machine learning models which can be used for prediction of weather with much simpler and easier way than the physical models. The accuracy evaluation of the models shows that the machine learning models perform better than the traditional Smart Weather Prediction Using Machine Learning models. These models made use of the dataset collected from predefined recourses in which the maximum accuracy is observed upto 81.67%. In future, is is planned to use the different IoT devices to collect the accurate data so that the data set to be used in the model will be more exact and accordingly the performance of the model will be more correct.

# REFERENCE AND BIBLIOGRAPHY:

- **YOU TUBE (https://youtu.be/baqxBO4PhI8)**
- **GOOGLE**