A Project Report on

Weather prediction using Artificial Intelligence

submitted by

AYUSH PANDEY 1904331017

MUSHAB REHMAN 1904330135

VIPIN KUMAR 1904330165

under the guidance of

DR. ATUL KUMAR DWIVEDI

in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BUNDELKHAND INSTITUTE OF ENGINEERING AND TECHNOLOGY, JHANSI, INDIA

2022-2023

Declaration

We hereby proclaim that the work that is being presented in this report entitled **Weather prediction using Artificial Intelligence** is our own work and has not been reported by anyone else or submitted in any form for another degree or diploma to any other institution. Information derived from the other sources has been accredited in the text and a list of references has been given.

Ayush Pandey	**Mushab Rehman**	**Vipin Kumar**
Roll no.1904331017	Roll no. 1904331035	Roll no.1904331065

Certificate

This is to verify that the project entitled**Weather prediction using Artificial Intelligence** submitted by Ayush Pandey, Mushab Rehman, Vipin Kumar for the award of the degree of B.Tech,is carried out by them under my guidance.

Dr. Atul Kumar Dwivedi

Department of ECE, BIET Jhansi

Verification

This is to verify that the project entitled**"Weather prediction using Artificial Intelligence "** was submitted to the department of Electronics and communication engineering department of Bundelkhand Institute of Engineering and Technology, Jhansi.

Dr. Atul Kumar Dwivedi	**Prof. Deepak Nagariya**
Officer in-charge Projects(UG)	Head of Department

DEDICATION

To my beloved Parents and almighty

PROJECT OUTCOMES(Ps)

S. N.	**Project outcomes** After completing this project students will be able to	**Bloom's knowledge level**
P1	Apply the knowledge of Machine Learning,at various places depending upon the need.	KL3
P2	Design various Models to make predictions of different weather parameters by testing and training the model.	KL6
P3	Predict the weather parameters.	KL5

KL: Bloom's knowledge level, KL1: remember, KL2: Understand, KL3: Apply, KL4: Analyse, KL5: Evaluate, KL6: Create/Design

Mapping of project outcomes with Program Outcomes (POs)

S. N.	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
P1	3	3	2	3	2	3	3	3	-	3	3	2
P2	3	3	3	3	2	3	3	2	-	3	3	3
P3	3	3	2	3	2	3	3	3	-	3	3	2

Mapping rules (Rubrics)

1: poor 2: medium 3: best

Mapping of Program Outcomes(POs) with Project

S. N.	**Program outcomes** After completing this project students will be able to	**Map**
PO1	**Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	3
PO2	**Problem Analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences..	2
PO3	**Design/development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	3
PO4	**Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	2
PO5	**Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern Engineering and IT tools including prediction and modeling to complex Engineering activities with an understanding of the limitations	3
PO6	**The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	2
PO7	**Environment and Sustainability:** Understand the impact of the professional Engineering solutions in societal and Environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	2

PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the Engineering practice.	3
PO9	**Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	3
PO10	**Communication:** Communicate effectively on complex Engineering activities with the Engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	2
PO11	**Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	-
PO12	**Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change	2

Mapping rules (Rubrics)

1: poor   2: medium   3: best

ABSTRACT

KEYWORDS: Machine Learning,Models,Algorithms

Weather forecasting is the use of science and technology to predict the condition of the weather for a given area. It is one of the most difficult issues the world over. This project aims to estimate the weather by utilizing predictive analysis. For his reason, analysis of various data mining procedures is needed before applying. This introduces a classifier approach for prediction of weather conditions and shows how Regression and Neural Network algorithms can be utilized for classification purposes.

Users will enter some information such as current temperature,humidity and other parameters according to their need .This system will take this parameter and predict weather after analyzing the input information with the information in the database. Consequently two basic functions to be specific classification (training) and prediction (testing) will be performed. The outcomes demonstrated that these data mining procedures can be sufficient for weather forecasting

ACKNOWLEDGEMENT

We would like to express our deep sense of gratitude to our supervisor and guide Dr. Atul Kumar Dwivedi, Department of Communication Engineering, BIET, Jhansi, for his constant guidance and motivation during the project work from last one year. We really appreciate and value their guidance and constant motivation from the beginning of the project. Without his guidance and persistent help this project would not have been possible. We are also indebted to Professor Deepak Nagaria, Head Department of Electronics and Communications Engineering for his encouraging co-operation and continued interest in shaping this project. We express our sincere gratitude to Prof. P. M. Pandey, Director, BIET, Jhansi, who has been instrumental in providing an ideal environment for wholesome individual development and appreciating talents in both academic and extracurricular activities. We extend our sincere thanks to all faculties of the Department of Electronics and communication engineering, BIET, Jhansi, for their focused guidance and encouragement. We would like to thank our parents and friends and all those people who have supported me directly or indirectly to complete this project work.

Ayush Pandey	**Mushab Rehman**	**Vipin Kumar**
Roll no. 1904331017	Roll no. 1904331035	Roll no.1904331065

MAJOR CONTRIBUTIONS

The main objective of this project is to predict the weather(various parameters depending upon the needs) and display it to the user. The main contributions of the work are:

1 Machine Learning has been used for weather prediction.

2 The Prediction is done using various model training and testing in machine learning.

3 Prediction is being done here for the betterment of the experience of the person. ex.Farmer.

TABLE OF CONTENTS

Table of Contents (continued) **Page**

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ML	Machine Learning
ANN	Artificial Neural Network
NN	Neural Network
ADF	Augmented Dickey-Fuller
ARIMA	Autoregressive Integrated Moving Average
API	Application Programming Interface
MSE	Mean Square Error

CHAPTER 1

INTRODUCTION

1.1 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[1]. 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 depending on the authentic information given of any area. Contribution to these models is given, for example, if anticipating temperature, least temperature, greatest temperature, mean dampness, and order for 2 days. In light of this Minimum Temperature and Maximum Temperature of 7 days will be accomplished..

1.2 Purpose

Every Human is subject to adjusting themselves with respect to weather conditions for their dressing habits to strategic organizational planning activities, since the adverse weather conditions may cause considerable damage to lives and properties. We need to be on alert for these adverse weather conditions by taking some precautions and using prediction mechanisms to detect them and provide early warning of hazardous weather phenomena. Weather prediction is an indispensable requirement for all of us. Weather is important for most aspects of human life. Predicting weather is very useful. Humans have attempted to make predictions about the weather, many early religions used gods to explain the weather. Only relatively recently have humans developed reasonably accurate weather predictions. We decided to collect weather data and measure the accuracy of predictions made using linear regression.The Weather prediction model designed by us would be of great use to the farmers and for normal beings as well. In temperature forecasting one has to distinguish between the times the forecast goes ahead, for example temperature one hour ahead or minimum and maximum temperature of a given day. Several works has been done and different artificial neural networks (ANN) models have been tested[2]. The observations include:

Temperature - the measure of warmth or coldness

Humidity- the amount of moisture in the atmosphere

Precipitation - the amount of moisture(usually rain or snow) which falls in the ground

Pressure - the force atmosphere applies on the environment

1.3 Machine Learning

Machine Learning is the study of computer algorithms that improve automatically through experience. Applications range from data mining programs that discover general rules in large data sets, to information filtering systems that automatically learn users' interests.

Machine Learning is concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases[2]. A major focus of Machine Learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data; the difficulty lies in the fact that the set of all possible behaviors given all possible inputs is too complex to describe generally in programming languages, so that in effect programs must automatically describe programs.

The poor performance results produced by statistical estimation models have flooded the estimation area for over the last decade. Their inability to handle categorical data, cope with missing data points, spread of data points and most importantly lack of reasoning capabilities has triggered an increase in the number of studies using non-traditional methods like machine learning techniques. The area of machine learning draws on concepts from diverse fields such as statistics, artificial intelligence, philosophy, information theory, biology, cognitive science, computational complexity and control theory[3].

1.4 Types of Machine Learning

There are two main types of Machine Learning algorithms. In this project, supervised learning is adopted here to build models from raw data and perform regression and classification.

Supervised Learning-Supervised Learning is a machine learning paradigm for acquiring the input output relationship information of a system based on a given set of paired input- output training samples. As the output is regarded as the label of the input data or the supervision, an input-output training sample is also called labeled training data, or supervised data. Learning from Labeled Data, or Inductive Machine Learning. The goal of supervised learning is to build an artificial system that can learn the mapping between the input and the output, and can predict the output of the system given new inputs. If the output takes a finite set of discrete values that indicate the class labels of the input, the learned mapping leads to the classification of the input data. If the output takes continuous values, it leads to a regression of the input. It deduces a function from training data that maps inputs to the expected outcomes. The output of the function can be a predicted continuous value (called regression), or a predicted class label from a discrete set for the input object (called classification). The goal of the supervised learner is to predict the value of the function for any valid input object from a number of training examples. The most widely used classifiers are the Neural Network (Multilayer perceptron), Support Vector Machines, Regression Analysis, Artificial neural networks and time series analysis[4].

Unsupervised Learning- Unsupervised learning studies how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of input patterns. By contrast with supervised learning or reinforcement learning, there are no explicit target outputs or environmental evaluations associated with each input; rather the unsupervised learner brings to bear prior biases as to what aspects of the structure of the input should be captured in the output.

1.5 Problem Identification

Weather prediction is a useful tool for informing populations of expected weather conditions. Weather prediction is a complex topic and poses significant variation in practice. We will attempt to understand and implement a weather prediction application.

A detailed study of the process should be done with a variety of techniques such as interviews, questionnaires etc. Data collected by these sources must be evaluated in order to reach a conclusion. The conclusion is to understand how the system works. This program is called the existing system. The current system is now being processed and the problem location is identified. The designer now acts as a problem solver and tries to solve the problems the business is facing[5]. Solutions are offered as suggestions. The proposal is then weighed against an existing system by analysis and selection is best. The suggestion is presented to the user for user approval.

Every Human is subject to adjusting themselves with respect to weather conditions for their dressing habits to strategic organizational planning activities, since the adverse weather conditions may cause considerable damage to lives and properties.

1.6 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 percent size of information is utilized and 20 percent size of information is named as a Test set.

For Example, if we need to anticipate the temperature of Jhansi, 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[6].

To finish up, Machine Learning has enormously changed the worldview of Weather estimating with high precision and predictivity. What's more, in the couple of years greater progression will be made utilizing these advances to precisely foresee the climate to avoid catastrophes like typhoons, Tornados, and Thunderstorms.

CHAPTER 2

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 "Jhansi City". The dataset was created by keeping in mind the necessity of such historical weather data in the community. The dataset was used with the help of worldweatheronline.com . The datasets contain weather data from 01-01-2019 to 01-02-2023. The data of the city is for more than 4 years. This data can be used to visualize the change in data or can be used to predict the weather.

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. This data can also be used to make visualizations 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 Jhansi city with the help of various machine learning algorithms and various regressions. By applying various regressions on the historical weather dataset of Jhansi's city we are predicting the temperature like first we are applying Multiple Linear regression, then Neural Network[7].

2.1 Plots(Various Parameters Present in the dataset)

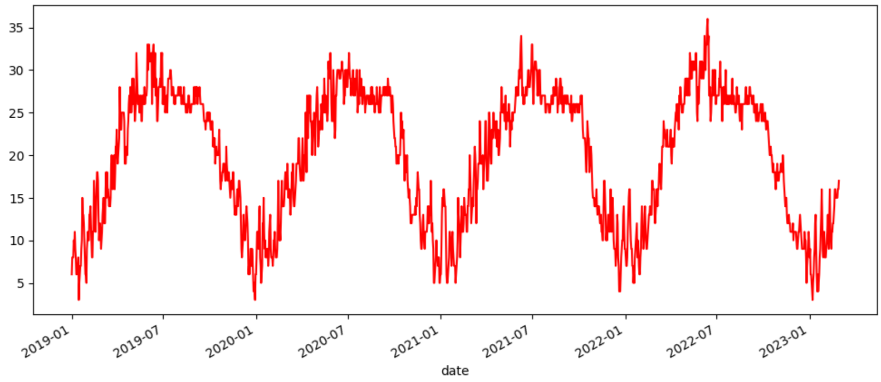


Fig. 2.1: Minimum Temperature Plot

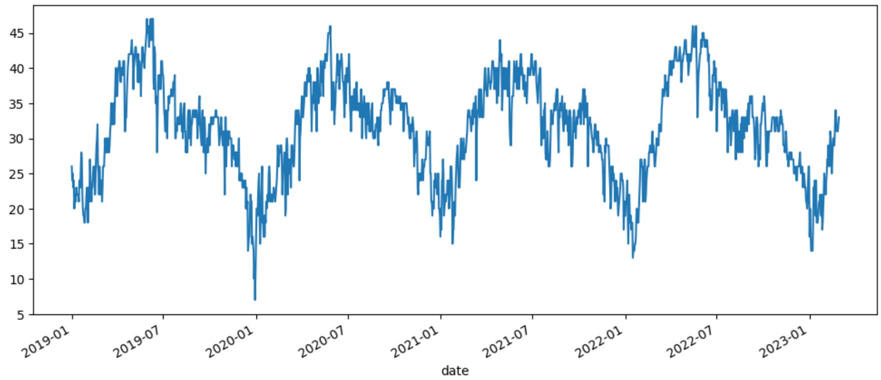


Fig. 2.2: Maximum Temperature Plot

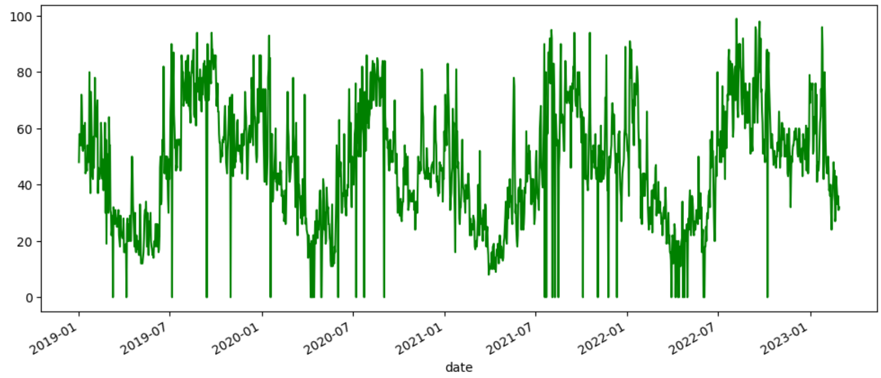


Fig. 2.3: Minimum Humidity Plot

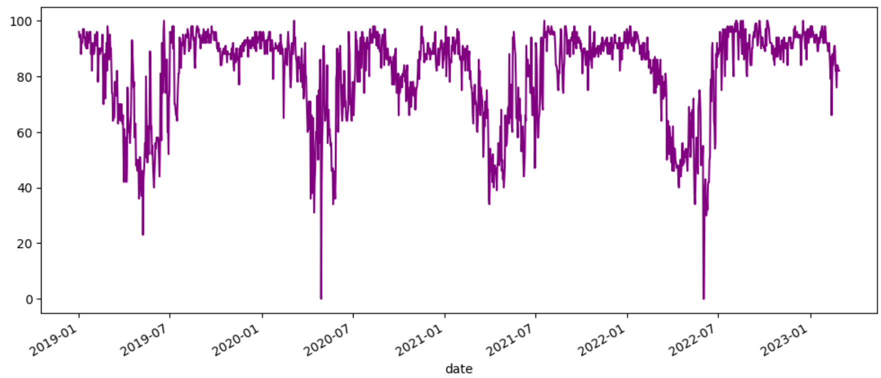


Fig. 2.4: Maximum Humidity Plot

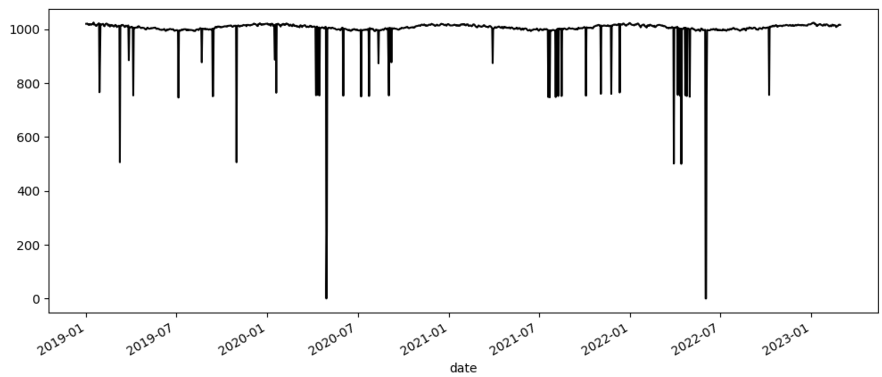


Fig. 2.5: Average Atmospheric Pressure Plot

CHAPTER 3

MODELS

3.1 Working Of ARIMA Model

Weather forecasting is performed on Time Series Data. A time series is a sequence where a metric is recorded over regular time intervals. We use this type of series to forecast any event in the future such as temperature, rainfall, humidity, budgets etc.

For prediction we are going to use one of the most popular models for time series, Autoregressive Integrated Moving Average (ARIMA) which is a standard statistical model for time series forecast and analysis[8]. An ARIMA model can be understood by outlining each of its components as follows: Autoregression (AR) - refers to a model that shows a changing variable that regresses on its own lagged, or prior, values. The notation AR (p) indicates an autoregressive model of order p.

$$Yt = a + b1Yt - 1 + b2Yt - 2 + ... + bpYt - p + e1 \tag{3.1}$$

Integrated (I) - represents the differencing of raw observations to allow for the time series to become stationary, i.e., data values are replaced by the difference between the data values and the previous values **Moving average (MA)** - incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations. The notation MA(q) refers to the moving average model of order q.

$$Y = a + et + et - 1 + et - 2 + ... + qet - q \tag{3.2}$$

Equation of the ARIMA model- Combination of AR and MA model

$$Yt = a + b1Yt - 1 + b2Yt - 2 + ... + bpYt - pet + et - 1 + et - 2 + ... + qet - q \tag{3.3}$$

Before using the ARIMA model, we need to check whether the dataset is stationary or not. Check for below necessary conditions:
–> Constant mean
–> Constant variance
–> An auto covariance that does not depend on time

If we have constant Mean and Variance, and our Test statistic is less than Critical Values, we already have a stationary Time series. So our 'd' value will become 0 in the ARIMA Model. And if it was non-stationary, in that case we would use below techniques to make it stationary by using any of the below techniques:
–> Decomposing
–> Differencing

Auto ARIMA is a variance of ARIMA that is particularly useful for non-stationary datasets. Auto ARIMA saves the task of differencing and computing p, q, d values of ARIMA. Forecasting is done directly by fitting the Auto ARIMA model on the univariate time series data

3.2 Dataset Description

time series weather dataset is used to implement the ARIMA model of forecasting. The dataset contains weather data for Jhansi, India from year 2019 to 2023.This weather dataset includes several attributes such as temperature , humidity, atmospheric pressure etc. We apply the ARIMA model on various univariate time series from the Jhansi weather dataset.

Univariate time series is a time series that consists of only single observations recorded sequentially over equal time increments. Here, the ARIMA model of weather forecasting is applied to the temperature data from the weather dataset. The Auto ARIMA model of weather forecasting is applied to the temperature data from the weather dataset. Following are some data values from temperature and all the parameters in the dataset respectively.

Table 3.1: Weather Dataset of Jhansi City

Date	Temp-max	Temp-min	Atm-avg	Hum-max	Hum-min	Location
2019-01-01	26	6	1020	96	48	-Jhansi
2019-01-02	24	8	1020	94	58	Jhansi
2019-01-03	25	8	1021	94	54	Jhansi
2019-01-04	23	8	1020	95	57	Jhansi
2019-01-05	24	10	1018	88	54	Jhansi

3.3 Materials and Methods

The following process is used to implement the ARIMA model of weather forecasting on temperature data of Jhansi.

Step-1: Import statsmodels and pmdarima Python module for loading the ARIMA model. Import numpy, pandas, matplotlib, seaborn Python libraries for implementation and load the temperature dataset for ARIMA forecasting..

Step-2: Import the required libraries

```
# Install required packages
install Jupyter notebook
install numpy
install pandas
install matplotlib
install seaborn
install statsmodels
install pmdarima
```

```python
# Imported required libraries

import numpy as np
from scipy import stats
import pandas as pd
import matplotlib.pyplot as plt

import statsmodels.api as sm
```

Step-3: Load the data

```python
# Loading the data

df=pd.read_csv('dataset.csv',index_col='date',parse_dates=True)
df=df.dropna()
df=df.drop("s.no",axis=1)
print('Shape of data',df.shape)
df.head()
```

```
Shape of data (1519, 9)

            day  month  year  temp_max  temp_min  atm_avg  hum_max  hum_min  \
date
2019-01-01    1      1  2019        26         6     1020       96       48
2019-01-02    2      1  2019        24         8     1020       94       58
2019-01-03    3      1  2019        25         8     1021       94       54
2019-01-04    4      1  2019        23         8     1020       95       57
2019-01-05    5      1  2019        24        10     1018       88       54

          Location
```

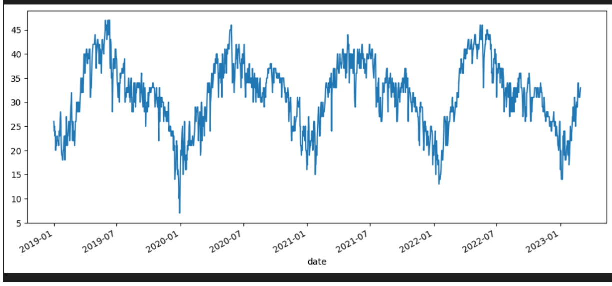
Step-4: Plot the data to check the trend and seasonality of any parameter in the dataset

```python
# Plotting the data to check the trend and seasonality of the particular attribute

df['temp_max'].plot(figsize=(12,5))
```

Python

```
<AxesSubplot: xlabel='date'>
```



Step-5: Perform the ad-fuller test to check the stationarity of the data

When we make a model for forecasting purposes in time series analysis,we require a stationary time series for better prediction.So the first step to work on modeling is to make a time series stationary.Testing for stationarity is a frequently used activity in autoregressive modeling.

ADF (Augmented Dickey-Fuller) test is a statistical significance test which means the test will give results in hypothesis tests with null and alternative hypotheses. As a result, we will have a p-value from which we will need to make inferences about the time series, whether it is stationary or not.

```python
# For stationary data, get parameter values for the ARIMA
# model.

from statsmodels.tsa.stattools import adfuller

def adf_test(dataset):
  dftest = adfuller(dataset, autolag = 'AIC')
  print("1. ADF : ",dftest[0])
  print("2. P-Value : ", dftest[1])
  print("3. Num Of Lags : ", dftest[2])
  print("4. Num Of Observations Used For ADF Regression and Critical Values Calculation :", dftest[3])
  print("5. Critical Values :")
  for key, val in dftest[4].items():
      print("\t",key, ": ", val)
```

Python

```python
adf_test(df['temp_max'])
```

Python

```
1. ADF :  -3.1023391279877295
2. P-Value :  0.02638056613489309
3. Num Of Lags :  9
4. Num Of Observations Used For ADF Regression and Critical Values Calculation : 1509
5. Critical Values :
	1% :  -3.434690926976026
	5% :  -2.863457245470903
	10% :  -2.5677907166982643
```

Here in the results, we can see that the p-value for time series is greater than 0.05, and we can say we fail to reject the null hypothesis and the time series is non-stationary.

Step-6: For forecasting non-stationary data, the Auto ARIMA model is used

```python
# Apply ARIMA model on the stationary training data and perform model fitting. Plot the model's residual errors

from pmdarima import auto_arima
import warnings
warnings.filterwarnings("ignore")
```

```python
stepwise_fit = auto_arima(df['temp_max'],trace=True,
                          suppress_warnings=True)

stepwise_fit.summary()
```

```
Performing stepwise search to minimize aic
 ARIMA(2,0,2)(0,0,0)[0] intercept   : AIC=6689.182, Time=1.41 sec
 ARIMA(0,0,0)(0,0,0)[0] intercept   : AIC=10233.644, Time=0.03 sec
 ARIMA(1,0,0)(0,0,0)[0] intercept   : AIC=6785.267, Time=0.12 sec
 ARIMA(0,0,1)(0,0,0)[0] intercept   : AIC=8811.163, Time=0.20 sec
 ARIMA(0,0,0)(0,0,0)[0]             : AIC=14862.969, Time=0.02 sec
 ARIMA(1,0,2)(0,0,0)[0] intercept   : AIC=6714.755, Time=0.41 sec
 ARIMA(2,0,1)(0,0,0)[0] intercept   : AIC=6688.107, Time=0.76 sec
 ARIMA(1,0,1)(0,0,0)[0] intercept   : AIC=6748.562, Time=0.26 sec
 ARIMA(2,0,0)(0,0,0)[0] intercept   : AIC=6758.825, Time=0.15 sec
 ARIMA(3,0,1)(0,0,0)[0] intercept   : AIC=6689.081, Time=1.33 sec
 ARIMA(3,0,0)(0,0,0)[0] intercept   : AIC=6740.247, Time=0.20 sec
 ARIMA(3,0,2)(0,0,0)[0] intercept   : AIC=6689.687, Time=1.17 sec
 ARIMA(2,0,1)(0,0,0)[0]             : AIC=6697.456, Time=0.09 sec

Best model:  ARIMA(2,0,1)(0,0,0)[0] intercept
Total fit time: 6.170 seconds
```

The best model which came out to be used is(2,0,1) the A non seasonal ARIMA model is classified as an "ARIMA(p,d,q)" model, where:

p is the number of autoregressive terms,

d is the number of nonseasonal differences needed for stationarity, and

q is the number of lagged forecast errors in the prediction equation.

Summary of the model

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1519
Model:               SARIMAX(2, 0, 1)   Log Likelihood               -3339.053
Date:                Thu, 23 Mar 2023   AIC                           6688.107
Time:                        21:05:45   BIC                           6714.736
Sample:                             0   HQIC                          6698.021
                               - 1519
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept      0.1285      0.056      2.307      0.021       0.019       0.238
ar.L1          1.5706      0.046     34.368      0.000       1.481       1.660
ar.L2         -0.5748      0.045    -12.879      0.000      -0.662      -0.487
ma.L1         -0.8229      0.033    -24.591      0.000      -0.888      -0.757
sigma2         4.7438      0.135     35.147      0.000       4.479       5.008
===================================================================================
Ljung-Box (L1) (Q):                   0.29   Jarque-Bera (JB):               534.85
Prob(Q):                              0.59   Prob(JB):                         0.00
Heteroskedasticity (H):               0.68   Skew:                            -0.85
Prob(H) (two-sided):                  0.00   Kurtosis:                         5.36
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
"""
```

Step-7: Set the testing and training part

```
💡Performing the training and testing on the dataset
from statsmodels.tsa.arima_model import ARIMA
```

```
print(df.shape)
train=df.iloc[:-30]
test=df.iloc[-30:]
print(train.shape,test.shape)
print(test.iloc[0],test.iloc[-1])
```

```
(1519, 9)
(1489, 9) (30, 9)
day             30
month            1
year          2023
temp_max        25
temp_min        11
atm_avg       1017
hum_max         96
hum_min         80
Location    Jhansi
Name: 2023-01-30 00:00:00, dtype: object day             28
month            2
year          2023
temp_max        33
temp_min        17
atm_avg       1016
hum_max         82
hum_min         32
Location    Jhansi
Name: 2023-02-28 00:00:00, dtype: object
```

Step-8: Testing our model on the remaining datasets and plotting to see the difference, what was the temperature according to dataset and what the model has predicted

```
date
2023-01-30    21.675166
2023-01-31    24.013524
2023-02-01    22.717895
2023-02-02    22.801285
2023-02-03    22.122150
2023-02-04    23.633579
2023-02-05    24.475516
2023-02-06    26.089064
2023-02-07    26.267278
2023-02-08    27.909430
2023-02-09    27.363429
2023-02-10    25.993397
2023-02-11    27.511069
2023-02-12    29.853649
2023-02-13    29.309247
2023-02-14    27.192778
2023-02-15    25.679980
2023-02-16    27.080176
2023-02-17    28.578335
2023-02-18    29.409354
2023-02-19    28.770658
2023-02-20    28.819856
2023-02-21    30.355846
2023-02-22    32.713499
2023-02-23    32.929253
2023-02-24    31.611283
2023-02-25    30.928537
2023-02-26    30.941488
2023-02-27    30.952145
2023-02-28    31.708667
Name:  predictions, dtype: float64
```

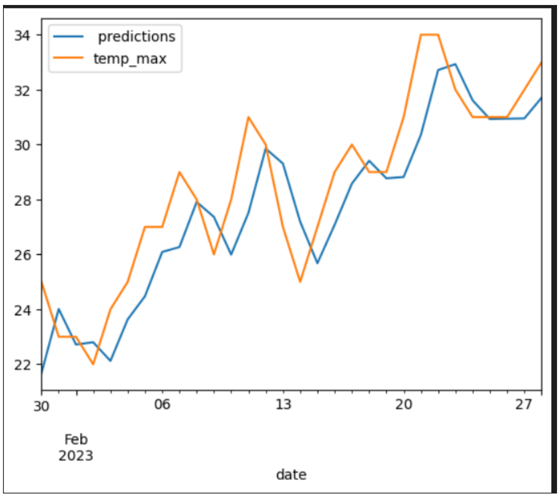


Fig. 3.1: Training and Testing Output

Step-9: Now the most important part comes where we have to predict that particular parameter for the future dates as per the requirements

```
# Prediction of the further dates as per the requirements

index_future_dates=pd.date_range(start='2023-02-28',end='2023-04-29')
print(index_future_dates)
pred=model.predict(start=len(df),end=len(df)+60,typ='levels').rename('ARIMA Predictions')
pred.index=index_future_dates
print(pred)
✓ 0.0s
```

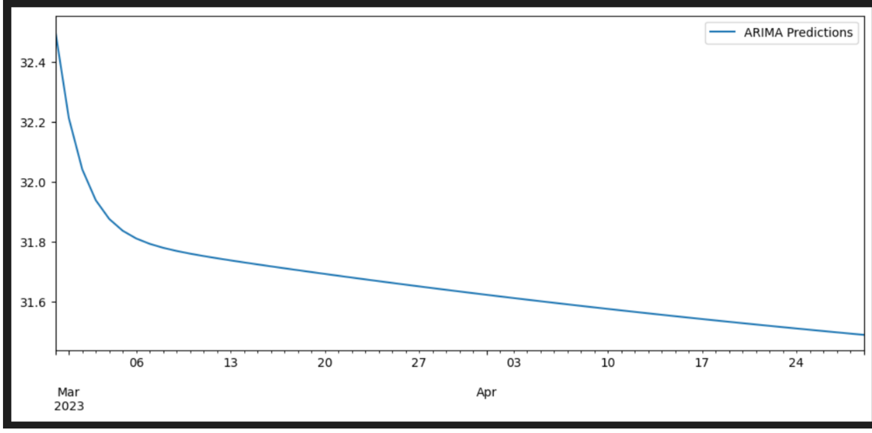
```
Output exceeds the size limit. Open the full output data in a text editor
DatetimeIndex(['2023-02-28', '2023-03-01', '2023-03-02', '2023-03-03',
               '2023-03-04', '2023-03-05', '2023-03-06', '2023-03-07',
               '2023-03-08', '2023-03-09', '2023-03-10', '2023-03-11',
               '2023-03-12', '2023-03-13', '2023-03-14', '2023-03-15',
               '2023-03-16', '2023-03-17', '2023-03-18', '2023-03-19',
               '2023-03-20', '2023-03-21', '2023-03-22', '2023-03-23',
               '2023-03-24', '2023-03-25', '2023-03-26', '2023-03-27',
               '2023-03-28', '2023-03-29', '2023-03-30', '2023-03-31',
               '2023-04-01', '2023-04-02', '2023-04-03', '2023-04-04',
               '2023-04-05', '2023-04-06', '2023-04-07', '2023-04-08',
               '2023-04-09', '2023-04-10', '2023-04-11', '2023-04-12',
               '2023-04-13', '2023-04-14', '2023-04-15', '2023-04-16',
               '2023-04-17', '2023-04-18', '2023-04-19', '2023-04-20',
               '2023-04-21', '2023-04-22', '2023-04-23', '2023-04-24',
               '2023-04-25', '2023-04-26', '2023-04-27', '2023-04-28',
               '2023-04-29'],
              dtype='datetime64[ns]', freq='D')
2023-02-28    32.504177
2023-03-01    32.213261
2023-03-02    32.041324
2023-03-03    31.938481
2023-03-04    31.875778
                ...
2023-04-25    31.505503
2023-04-26    31.501210
```

Step-10: Plot of the graph and the table of the values are as follows:

```
# Plotting the prediction

pred.plot(figsize=(12,5),legend=True)
✓ 0.3s
```

```
&lt;AxesSubplot: &gt;
```



3.4 Weather Forecasting using Neural Network

Artificial neural networks (ANNs) have emerged as a result of the discovery of the computer, the advancement of technology, the ability to store data regularly, and the ability of computers to think, problem-solve, remember and learn[10]. We are presenting weather predictions using Artificial Neural Network and Back Propagation Algorithm. We are implementing a data intensive model using data mining techniques. Weather is a dynamic and non-linear process and an artificial neural network (ANN) can deal with such a Process. We are using ANN which is based on smart analyzing the trend from historical data. The other models are accurate in calculation but not in predictions as they are not able to adapt the irregular patterns of data which can neither be written in the form of function or deducted as formula[11]. Use of ANN will give more accurate results. Here, the error may or may not reduce completely. But, the accuracy will improve as compared to previous forecasts.

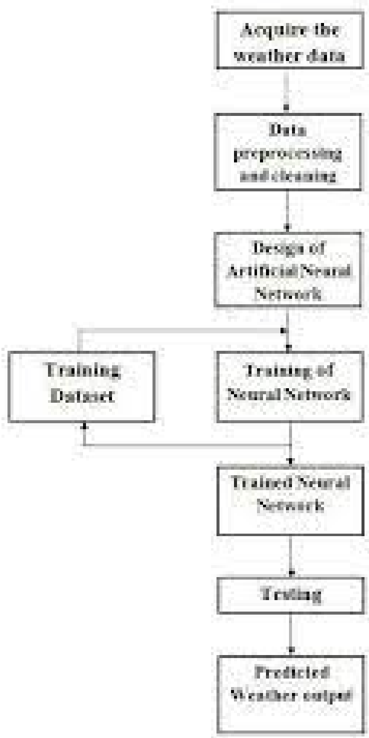


Fig. 3.2: Flow chart Of Neural Network Model Working

3.5 Procedure to develop model

To develop an ANN model for weather forecasting, region selection for input data and parameters is necessary. The input data is to be taken from a specific area on which the model is trained and tested so that the model is able to generate accurate results[12]. The number of input data given to the model also helps to improve accuracy of the model by giving the results with a high degree of similarity between predicted and actual output data. The available data may be noisy , data should be cleaned.

Similarly, it has to be normalized because all the parameters are of different units and normalization will help the input and output parameters to correlate with each other . The data should be divided in training and testing samples in proper proportion so that the results can be predicted, tested and validated properly. Structure of the NN model also has a great impact on the generation of accurate results[13].

As the weather data is nonlinear, Artificial Neural Network (ANN) has become an effective way of predicting weather data precisely and accurately[14]. Neural Network is a system that can be trained with certain input and output. It creates its own structure based upon how it is trained.

3.6 Back-Propagation Approach

The back propagation algorithm is used in layered feedforward ANNs. It uses supervised learning, which means the model trains itself with the use of target output. For every set of input data the target output is provided. The neural network model processes the input data with random values for weights and suitable activation function using one or more hidden layers in between and then produces the predicted output[15]. This predicted output is then compared with the target output provided for the same input dataset. Thus, error is calculated by subtracting predicted output from target output. Using this error, the weights are adjusted and again the entire process is repeated for multiple epochs until the error is minimal or in acceptable range[16] .

We start the training with random weights, and the goal is to adjust them so that the error will be minimal. The area for input data can be any one of a meteorological station area in which all the data is limited to a certain region. The different input parameters are taken viz. temperature, relative humidity, rainfall,etc.Input data is then pre-processed and cleaned. That means it is checked with any outlier and that is missing values are entered, and data is checked if it is in the given range for the given parameter.

Later ANN is designed with a number of input and output nodes, hidden layers, activation function, and maximum number of epochs, weights, bias, goal and learning function. Neural network is trained with seventy percent of the input data. Where the model is trained using this observed data to forecast the weather. Then the mean squared error and accuracy is calculated for the model by comparing the output of testing with target output. This model generates output in terms of minimum and maximum temperature of the day, relative humidity[17]. The back propagation (BP) neural network algorithm is a multi-layer feedforward network trained according to error back propagation algorithm and is one of the most widely applied neural network models. BP network can be used to learn and store a great deal of mapping relations of input-output model, and no need to disclose in advance the mathematical equation that describes these mapping relations. Its learning rule is to adopt the steepest descent method in which the back propagation is used to regulate the weight value and threshold value of the network to achieve the minimum error sum of square[18].

3.7 Methods

The following process is used to implement the Neural Network model of weather forecasting on temperature data of Jhansi.

Step-1: Import the required libraries.

```
import pandas as pd
from neuralprophet import NeuralProphet
from matplotlib import pyplot as plt
import pickle
```

Step-2: Read the dataset

```
df = pd.read_csv('dataset.csv')
df.head()
```

```
   s.no  day  month  year  temp_max  temp_min  atm_avg  hum_max  hum_min  \
0     0    1      1  2019        26         6     1020       96       48
1     1    2      1  2019        24         8     1020       94       58
2     2    3      1  2019        25         8     1021       94       54
3     3    4      1  2019        23         8     1020       95       57
4     4    5      1  2019        24        10     1018       88       54

       date Location
0  1/1/2019   Jhansi
1  1/2/2019   Jhansi
2  1/3/2019   Jhansi
3  1/4/2019   Jhansi
4  1/5/2019   Jhansi
```

Step-3: Setting the location as unique as there would be no problems if any other city is present in the dataset

```
df.Location.unique()
```

```
array(['Jhansi'], dtype=object)
```

```
df.columns
```

```
Index(['s.no', 'day', 'month', 'year', 'temp_max', 'temp_min', 'atm_avg',
       'hum_max', 'hum_min', 'date', 'Location'],
      dtype='object')
```

```
df.dtypes
```

```
s.no         int64
day          int64
month        int64
year         int64
temp_max     int64
temp_min     int64
atm_avg      int64
hum_max      int64
hum_min      int64
date        object
Location    object
dtype: object
```

```
df.index
```

Step-4: Importing and fitting and training the model

```
# TRAINING
```
✓ 0.2s Python

```
m = NeuralProphet()
```
✓ 0.2s Python

\+ Code \+ Markdown

```
model = m.fit(data, freq='D')
```
✓ 23.0s Python

```
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.868% of the data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency - D
INFO - (NP.config.init_data_params) - Setting normalization to global as only one dataframe provided for training.
INFO - (NP.utils.set_auto_seasonalities) - Disabling daily seasonality. Run NeuralProphet with daily_seasonality=True to override this.
INFO - (NP.config.set_auto_batch_epoch) - Auto-set batch_size to 32
INFO - (NP.config.set_auto_batch_epoch) - Auto-set epochs to 171

100% 130/130 [00:00<00:00, 251.02it/s]

INFO - (NP.utils_torch.lr_range_test) - lr-range-test results: steep: 1.00E-01, min: 2.23E-01

100% 130/130 [00:00<00:00, 314.89it/s]

INFO - (NP.utils_torch.lr_range_test) - lr-range-test results: steep: 1.00E-01, min: 2.23E-01
INFO - (NP.forecaster._init_train_loader) - lr-range-test selected learning rate: 1.00E-01
Epoch[171/171]: 100%|██████████| 171/171 [00:20<00:00,  8.32it/s, SmoothL1Loss=0.00328, MAE=2.19, RMSE=2.8, Loss=0.00236, RegLoss=0]
```

Step-5: .Making the Future predictions and plotting it

```
# FORECAST  AWAY
```
✓ 0.0s Python

```
future = m.make_future_dataframe(data, periods=60)
forecast = m.predict(future)
forecast.head()
```
✓ 0.1s Python

```
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.868% of the data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency - D
INFO - (NP.df_utils.return_df_in_original_format) - Returning df with no ID column
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 98.333% of the data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency - D
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 98.333% of the data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency - D
INFO - (NP.df_utils.return_df_in_original_format) - Returning df with no ID column

          ds     y residual1      yhat1      trend  season_yearly  \
0 2023-03-01  None       NaN  31.542973  33.145584      -1.803085
1 2023-03-02  None       NaN  31.586842  33.149689      -1.578331
2 2023-03-03  None       NaN  31.501572  33.153793      -1.353288
3 2023-03-04  None       NaN  31.893810  33.157898      -1.127531
4 2023-03-05  None       NaN  32.124893  33.162003      -0.900648

   season_weekly
0       0.200474
1       0.015485
2      -0.298932
3      -0.136553
4      -0.136461
```

Step-6: Forecast Plotting

```
plot1 = m.plot(forecast)
```
✓ 0.2s Python

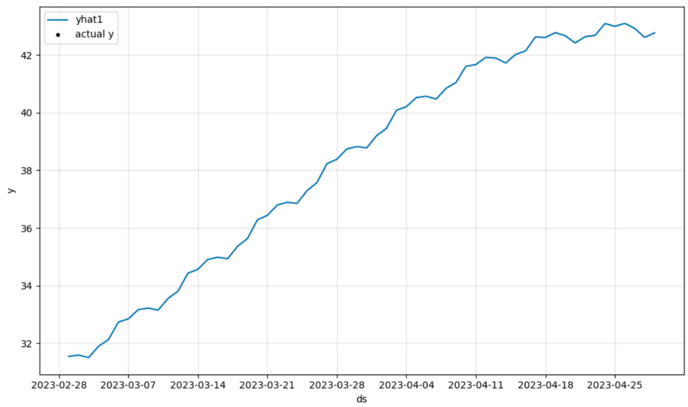


Fig. 3.3: Forecast Of Maximum Temperature for time span of 2 Months of NN Mode

Step-7: Component Plotting

```python
plt2 = m.plot_components(forecast)
# trend shows the overall repeating of the data
# second one shows the yearly seasonality
# weekly seasonality
```

✓ 0.4s Python

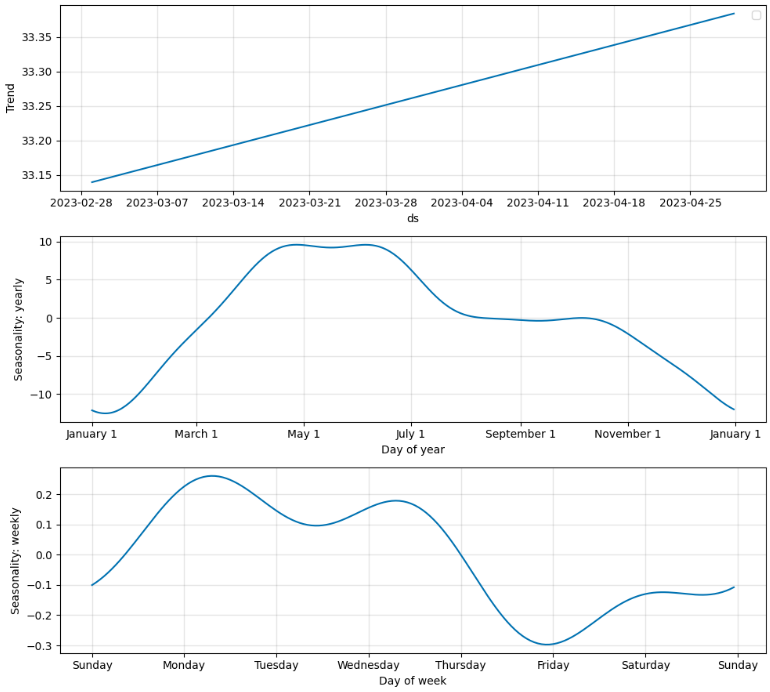


Fig. 3.4: Components Plotting of Maximum Temperature

3.8 Predictions of various parameters through neural networks:

1. Minimum Temperature

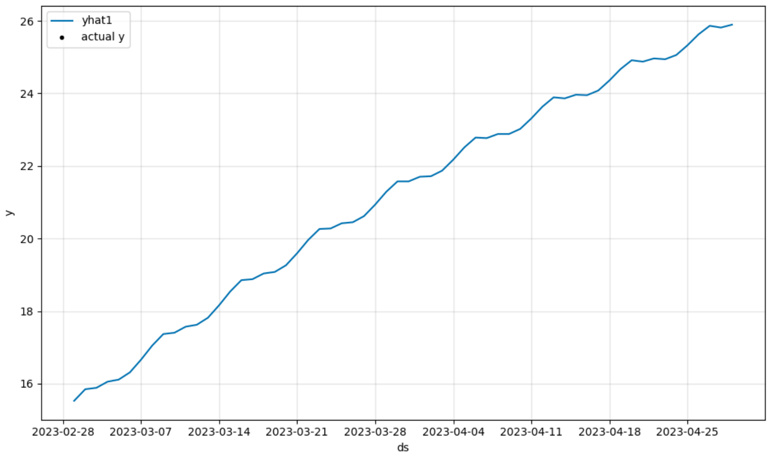


Fig. 3.5: Plot of Minimum Temperature by NN Model

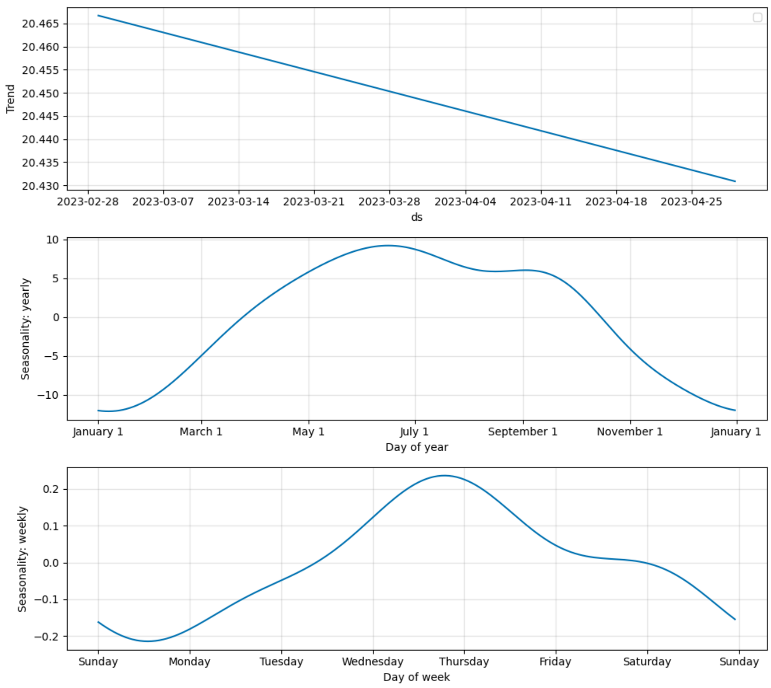


Fig. 3.6: Components Plotting of Minimum Temperature

2. Minimum Humidity

3. Maximum Humidity

Fig. 3.7: Plot of Minimum Humidity by NN Model

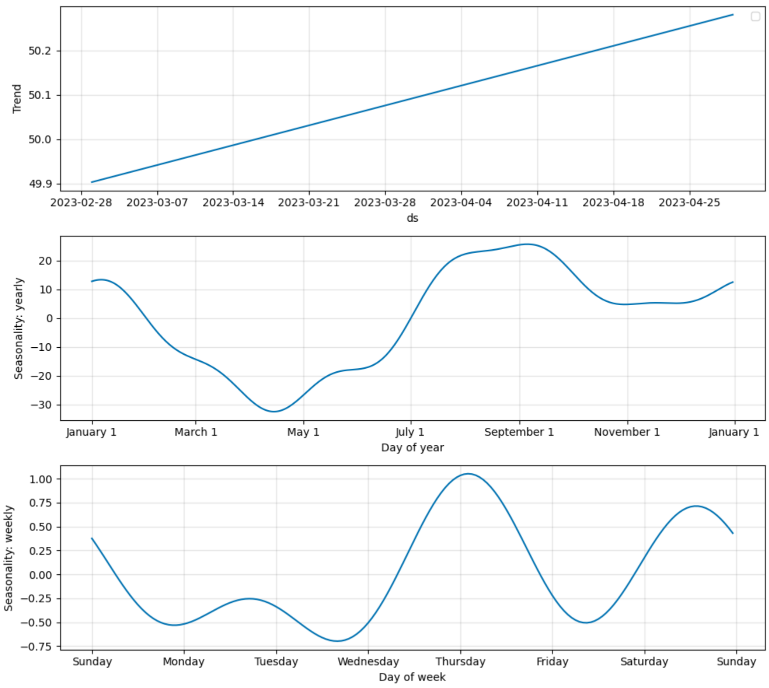


Fig. 3.8: Components Plotting of Minimum Humidity

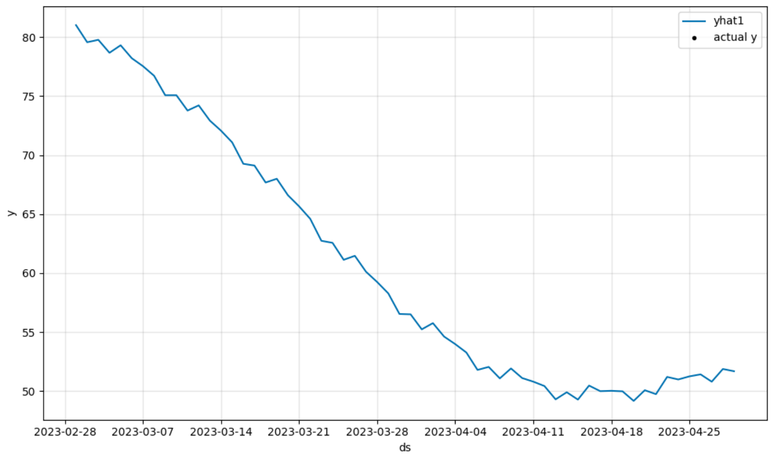


Fig. 3.9: Plot of Maximum Humidity by NN Model

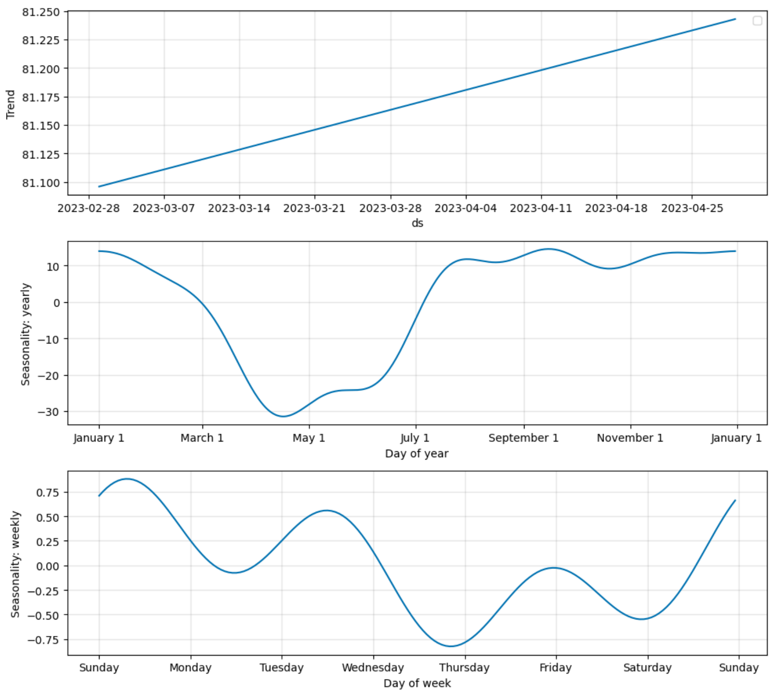


Fig. 3.10: Components Plotting of Maximum Humidity

CHAPTER 4

OUTPUTS (PREDICTIONS vs ACTUAL)

4.1 Maximum Temperature Comparison

Table 4.1: Comparison of Maximum Temperature

Dates	**ARIMA** ($°C$)	**Neural Network** ($°C$)	**Actual** ($°C$)
01/03/2023	32.7	31.5	30
02/03/2023	32.2	31.6	32
03/03/2023	32.7	31.5	33
04/03/2023	31.9	32	28
05/03/2023	31.8	32.7	31
15/03/2023	32.7	35	34
16/03/2023	32.7	35	28
17/03/2023	31.7	35.5	31
18/03/2023	32.9	34	30
19/03/2023	34.8	36.3	27
20/03/2023	33.7	36.5	25
01/04/2023	31.6	39.3	30
02/04/2023	31.6	39.5	32
03/04/2023	36.9	40.3	35
04/04/2023	37.7	40.3	36
05/04/2023	37.7	40.6	35
15/04/2023	31.5	42	40
16/04/2023	31.5	42.3	40
17/04/2023	32.9	42.8	41
18/04/2023	38.4	42.7	38
19/04/2023	39.4	42.8	39
20/04/2023	40	42.7	40

In the above comparison the ARIMA models proves to be accurate among the both.

4.2 Minimum Temperature Comparison

Table 4.2: Comparison of Minimum Temperature

Dates	**ARIMA** (°C)	**Neural Network** (°C)	**Actual** (°C)
01/03/2023	17	15.5	17
02/03/2023	17	15.7	16
03/03/2023	17.2	15.8	16
04/03/2023	17.2	16	19
05/03/2023	17.3	16.2	17
15/03/2023	18	18.4	18
16/03/2023	17.9	18.7	12
17/03/2023	17.9	18.7	18
18/03/2023	18	18.9	18
19/03/2023	18	19.2	19
20/03/2023	18.5	19.4	18
01/04/2023	18.8	21.5	17
02/04/2023	18.8	21.5	18
03/04/2023	18.4	21.7	17
04/04/2023	18.5	22	20
05/04/2023	18.5	22.3	19
15/04/2023	18.7	23.8	22
16/04/2023	18.8	23.8	22
17/04/2023	19	24	23
18/04/2023	19.9	24.2	27
19/04/2023	20	24.5	24
20/04/2023	22.4	24.7	24

In the above comparison the ARIMA models proves to be accurate among the both.

4.3 Maximum Humidity Comparison

Table 4.3: Comparison of Maximum Humidity

Dates	ARIMA	Neural Network	Actual
01/03/2023	82.3	81.4	82
02/03/2023	80	80	86
03/03/2023	81.5	80.1	82
04/03/2023	82	79	81
05/03/2023	82.6	79.8	74
15/03/2023	72.7	71.6	86
16/03/2023	72.4	69.8	80
17/03/2023	71.8	69.5	80
18/03/2023	71.5	68.1	72
19/03/2023	72.8	68.6	74
20/03/2023	72	66.2	74
01/04/2023	68.4	55.6	72
02/04/2023	68.6	56.3	65
03/04/2023	68.3	55.1	61
04/04/2023	68.5	54.5	62
05/04/2023	65	53.7	64
15/04/2023	59.4	49.6	60
16/04/2023	57.1	51	61
17/04/2023	55.4	50.5	52
18/04/2023	57.4	50.5	55
19/04/2023	54.3	50.4	57
20/04/2023	50.8	49.6	49

In the above comparison the ARIMA models proves to be accurate among the both.

4.4 Minimum Humidity Comparison

In the above comparison the ARIMA models proves to be accurate among the both.

Table 4.4: Comparison of Minimum Humidity

Dates	ARIMA	Neural Network	Actual
01/03/2023	34.9	34.9	40
02/03/2023	36	36.2	35
03/03/2023	36.3	34.7	38
04/03/2023	35.5	34.8	38
05/03/2023	35.7	34.7	34
15/03/2023	38.2	30.7	35
16/03/2023	38.4	31.9	35
17/03/2023	36.5	30.2	42
18/03/2023	36.7	29.7	41
19/03/2023	35.5	28.5	50
20/03/2023	34.8	28.1	35
01/04/2023	30.4	22.2	41
02/04/2023	28.5	21.8	35
03/04/2023	25.4	20.7	25
04/04/2023	25.4	20.3	28
05/04/2023	24.7	19.3	24
15/04/2023	22.4	19.9	31
16/04/2023	22.8	18.1	17
17/04/2023	20.4	17.5	19
18/04/2023	20.4	17.6	18
19/04/2023	20.4	17.6	22
20/04/2023	20	19.4	22

CHAPTER 5

CONCLUSION

These methods are extremely easy to adopt as they don't require any specific deep technical concepts to be clear in.Nonetheless,**predictions perfectly fit in the error range designed by the dataset itself** . It is important to consider that we only have examined monthly average values while it may be interesting to consider daily values too and have daily predictions.

We have implemented the ARIMA model of weather forecasting on Jhansi's weather dataset. The implementation uses the Forward Fill method in the data cleaning process to fill missing values. We proposed a modification in the data cleaning process- to fill missing values using the mean of the observations. Auto ARIMA is then applied on the modified temperature Mean Squared Error(MSE) is used to evaluate the model's performance for a certain data cleaning method. For Non-stationary data, Mean of the Observations Data Cleaning process in Auto ARIMA forecasting is a better approach than Forward fill to fill missing values. All the machine learning models: 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, our models may beat genius professional ones[19].

In the second model which we made was the Neural Network Model.Neural network is trained with seventy percent of the input data. Where the model is trained using this observed data to forecast the weather, followed by testing done using remaining thirty percentages of input data. Then the mean squared error and accuracy is calculated for the model by comparing the output of testing with target output. This model generates output in terms of minimum and maximum of various parameters.other machine-learning methods have been applied to various needs for targeted weather forecasts. Such applications include forecasting for agricultural decision support , forecasting road weather to enhance the safety of surface transportation[20].

APPENDIX A

Important Codes/Analysis(1)

A.1 ARIMA Model Code

```
import numpy as np
from scipy import stats
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm



df=pd.read_csv('dataset.csv',index_col='date',parse_dates=True)
df=df.dropna()
df=df.drop("s.no",axis=1)
print('Shape␣of␣data',df.shape)
df.head()
df['hum_max'].plot(figsize=(12,5))

from statsmodels.tsa.stattools import adfuller

def adf_test(dataset):
  dftest = adfuller(dataset, autolag = 'AIC')
  print("1.ADF:",dftest[0])
  print("2.P-Value:", dftest[1])
  print("3.Num␣Of␣Lag:", dftest[2])
  print("4.Num␣Of␣Observations␣Used␣For␣ADF␣Regression␣and␣Critical␣Values␣Calculatio
  print("5.␣Critical␣Values␣:")
  for key, val in dftest[4].items()
      print("\t",key, ":␣", val)

adf_test(df['hum_max'])
```

```python
from pmdarima import auto_arima
import warnings
warnings.filterwarnings("ignore")

stepwise_fit = auto_arima(df['hum_max'],trace=True,suppress_warnings=True)
stepwise_fit.summary()

from statsmodels.tsa.arima_model import ARIMA

print(df.shape)
train=df.iloc[:-30]
test=df.iloc[-30:]
print(train.shape,test.shape)
print(test.iloc[0],test.iloc[-1])

import statsmodels.api as sm

from statsmodels.tsa.arima_model import ARIMA
model=sm.tsa.arima.ARIMA(train['hum_max'],order=(2, 0, 1))
model=model.fit()
model.summary()

start=len(train)
end=len(train)+len(test)-1
# index_future_dates=pd.date_range(start='2023-03-01',end='2023-03-30')
pred=model.predict(start=start,end=end,typ='levels').rename('␣predictions')
print(pred)
# pred.index=index_future_dates
pred.plot(legend=True)
test['hum_max'].plot(legend=True)


test['hum_max'].mean()

from sklearn.metrics import mean_squared_error
```

```python
from math import sqrt
rmse=sqrt(mean_squared_error(pred,test['hum_max']))
print(rmse)



model=sm.tsa.arima.ARIMA(df['hum_max'],order=(2, 0, 1))
model=model.fit()
df.tail()

index_future_dates=pd.date_range(start='2023-02-28',end='2023-04-29')

# print(index_future_dates)
pred=model.predict(start=len(df),
end=len(df)+60,typ='levels').rename('ARIMAPredictions')
pred.index=index_future_dates
print(pred)

pred.plot(figsize=(12,5),legend=True)
```

APPENDIX B

Important Codes/Analysis(2)

B.1 Neural Network Model Code

```
!pip install neuralprophet

import pandas as pd
from neuralprophet import NeuralProphet
from matplotlib import pyplot as plt
import pickle

df = pd.read_csv('dataset.csv')
df.head()

df.Location.unique()

df.columns

df.dtypes

df.index

jhansi = df[df['Location']=='Jhansi']
jhansi['date'] = pd.to_datetime(jhansi['date'])

jhansi.dtypes
jhansi.head()

plt.plot(jhansi['date'], jhansi['hum_max'])
plt.show()

data = jhansi[['date', 'hum_max']]
data.dropna(inplace=True)
```

```
data.columns = ['ds', 'y']
data.head()


m = NeuralProphet()


model = m.fit(data, freq='D')

future = m.make_future_dataframe(data, periods=60)
forecast = m.predict(future)
forecast.head()

future

plot1 = m.plot(forecast)

plt2 = m.plot_components(forecast)

To save the model

with open('saved_model.pkl', "wb") as f:
    pickle.dump(m, f)

del m

with open('saved_model.pkl', "rb") as f:
    m = pickle.load(f)
```

APPENDIX C

Weekly Report

W. N.	Week Dates	Work Done
1	15-07-22 to 20-07-22	We all did the research work
2	01-08-22 to 15-08-22.	We all collected the base paper
3	22-08-22	We all give our introduction presentation to the Faculty member
4	04-09-22 to 22-09-22	We extracted the required dataset from various websites for the use of our project
5	11-12-22 to 12-12-22	We all together made our required models and tested it through our dataset
6	15-01-23 to 22-01-23	We all together completed the project
7	08-02-23 to 23-03-23	We prepared a brief report of the project

REFERENCES

[1] Holmstrom Mark, Dylan Liu and Christopher Vo, "Machine learning applied to weather forecasting", Meteorol. Appl, pp. 1-5, 2016.

[2] Sunita B. Aher and L. M. R. J. Lobo, "Comparative study of classification algorithms", International Journal of Information Technology, vol. 5.2, pp. 239-243, 2012.

[3] Sharma, S.; Sharma, S. Activation functions in neural networks. Towards Data Sci.no. 6,pp.310–316,2017

[4] Nalluri Sravani, Ramasubbareddy Somula and G Kannayaram, "Weather Prediction Using Clustering Strategies in Machine Learning", Journal of Computational and Theoretical Nanoscience, vol. 16, no. 5, pp. 1977-1981, 2019.

[5] S. D Drobot, M. Chapman, P. A. Pisano and B. B. McKeever, "Using vehicles as mobile weather platforms" in Data and Mobility: Transforming Information into Intelligent Traffic and Transportation Services, Springer, pp. 203-214, 2010.

[6]S. B. Lajevardi and B. Minaei-Bidgoli, "Combination of Time Series Decision Tree and Clustering : A Case Study in Aerology Event Prediction", International Conference on Computer and Electrical Engineering, pp. 111-115, 2008.

[7] D. B. Percival and A.T. Walden, "Methods for time series analysis" in , Cambridge, MA:Cambridge University Press, 2000.

[8] S. Shaminder, B. Pankaj and G. Jasmeen, "Time Series based Temperature Prediction using Back Propagation with Genetic Algorithm Technique", International Journal of Computer Science, vol. 8, pp. 28-32, 2011.

[9] Imran Maqsood, Ajith Muhammad Riaz Khan, Abraham An ensemble of neural network for weather forecasting, volume 13, pp 112 - 122,2004.

[10] Amanpreet Kaur, Sharma Artificial neural networks in forecasting maximum and minimum relative humidity International Journal of Computer Science and Network Security, volume 11, issue 5, p. 197 - 199,2011.

[11] G Vamsi Krishna An integrated approach for weather forecasting based on data mining and forecasting analysis International Journal of Computer Applications, volume 120, issue 11, p. 975 - 8887, 2015.

[12] Neeraj Kumar, Govind Kumar Jha A time series ANN approach for weather forecasting International Journal of Control Theory and Computer Modeling (IJCTCM), volume 3, 2013.

[13] N. Hasan, M. T. Uddin and N. K. Chowdhury, "Automated weather event analysis with machine learning", Proc. IEEE 2016 International Conference on Innovations in Science Engineering and Technology (ICISET), pp. 1-5, 2016.

[14] A. G. Salman, B. Kanigoro and Y. Heryadi, "Weather forecasting using deep learning techniques", Proc. IEEE 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 281-285, 2015.

[15]L. L. Lai, H. Braun, Q. P. Zhang, Q. Wu, Y. N. Ma, W. C. Sun, et al., "Intelligent weather forecast", Proc. IEEE 2004 International Conference on Machine Learning and Cybernetics, pp. 4216-4221, 2004.

[16]Bashar Muneer Yahya and Dursun Zafer Seker, "Designing Weather Forecasting Model Using Computational Intelligence Tools", Applied Artificial Intelligence, vol. 33:2, pp. 137-151, 2019.

[17]Vladimir M. Krasnopolsky and Michael S. FoxRabinovitz, "Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction", Neural Networks19, vol. 2, pp. 122-134, 2006.

[18] S. D Drobot, M. Chapman, P. A. Pisano and B. B. McKeever, "Using vehicles as mobile weather platforms" in Data and Mobility: Transforming Information into Intelligent Traffic and Transportation Services, Springer, pp. 203-214, 2010.

[19] Predicting Wind Speed, Energies, 14, 20, (M. Wiston and M. KM, "Weather Forecasting: From the Early Weather Wizards to Modern-day Weather Predictions", Journal of Climatology Weather Forecasting, vol. 6, pp. 1-9, 2018.

[20]M. Biswas, T. Dhoom and S. Barna, "Weather Forecast Prediction: An Integrated

Approach for Analyzing and Measuring Weather", Data International Journal of Computer Applications, vol. 182, no. 4, pp. 20-24, 2018.