A Project Report

on

Weather Forecasting with Machine Learning, using Python

Submitted in partial fulfillment of the
requirement for the award of the degree of

Batchelor of Technology in Computer Science and Engineering

Under The Supervision of
Mr. Dhruv Kumar
Assistant Professor

Submitted By

18SCSE1010211-Ajeet Kumar Choubey
18SCSE1010371-Awanish Kumar

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
MAY-2022

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"Weather Forecasting with Machine Learning, using Python"** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** submitted in the **School of Computing Science and Engineering of Galgotias University, Greater Noida,** is an original work carried out during the period of **JANUARY-2022 to MAY-2022**, under the supervision of **Mr.Dhruv Kumar, Assistant Professor**, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Ajeet Kumar Choubey-18SCSE1010211

Awanish Kumar-18SCSE1010371

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

(Mr. Dhruv Kumar,Assistant Professor)

CERTIFICATE

The Final Project Viva-Voce examination of **Ajeet Kumar Choubey -18SCSE1010211, Awanish Kumar -18SCSE1010371** has been held on_____and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: MAY 2022

Place: Greater Noida

Statement of Project Report Preparation

1. Thesis title: Weather Forecasting with Machine Learning
2. Degree for which the report is submitted: BACHELORS OF TECHNOLOGY.
3. Project Supervisor was referred to for preparing the report.
4. Specifications regarding thesis format have been closely followed.
5. The contents of the thesis have been organized based on the guidelines.
6. The report has been prepared without resorting to plagiarism.
7. All sources used have been cited appropriately.
8. The report has not been submitted elsewhere for a degree

Name: Ajeet Kumar Choubey

Roll No- 18SCSE1010211

Statement of Project Report Preparation

1.Thesis title: Weather Forecasting with Machine Learning

2.Degree for which the report is submitted: BACHELORS OF TECHNOLOGY.

3.Project Supervisor was referred to for preparing the report.

4.Specifications regarding thesis format have been closely followed.

5.The contents of the thesis have been organized based on the guidelines.

6.The report has been prepared without resorting to plagiarism.

7.All sources used have been cited appropriately.

8. The report has not been submitted elsewhere for a degree

Name: Awanish Kumar

Roll No- 18SCSE1010371

Abstract

Weather forecasting is the attempt by meteorologists to predict the weather conditions at some future time and the weather conditions that may be expected.
Earlier Forecasting was done on the basis of observed patterns of events, also known as pattern recognition. For ex- it was seen that on a particular day if the sunset was red it was considered to be fair weather. However not all of this predictions prove to be authentic.

Here in this Project we are making a prediction model using Machine Learning Algorithm and the algorithm used is supervised machine learning algorithm. In this we have collected previous data from Kaggle website and predicted the future temperature on the basis of temperature, humidity and Pressure

The application used in making this model is Jupyter Notebook,Kaggle website and the Framework used are Pandas, sklearn, Numpy, Matplotlib. These are the latest software and the applications used in weather Forecasting.

By using this model we will be easily be able to forecast the weather which is very essential in today's day to day life for planning for any vacation,in Air traffic, Agriculture, Military application,etc.

Table of Contents

List of Table

List of Figures

Acronyms

B.Tech.	Bachelor of Technology
SCSE	School of Computing Science and Engineering

CHAPTER-1

Introduction

1.1 Introduction

Weather forecasting is the attempt by meteorologists to predict the weather conditions at some future time and the weather conditions that may be expected. Earlier Forecasting was done on the basis of observed patterns of events, also known as pattern recognition. For ex- it was seen that on a particular day if the sunset was red it was considered to be fair weather. However not all of this predictions prove to be authentic.

Weather prediction is the task of prediction of the atmosphere at a future time and a given area. In early days, this has been done through physical equations in which the atmosphere is consider as fluid. The current state of the environment is inspected, and the future state is predicted by solving those equations numerically, but we can not determine a very accurate weather for more than 10 days and this can be improved with the help of science and technology. There are numerous of Machine Learning algorithms for Forecasting the weather in which we are using Linear Regression Algorithm and Polynomial Regression.

Machine learning, is relatively robust to perturbations and doesn't need any other physical variables for prediction. Therefore, machine learning is much better opportunity in 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 has vastly increased in terms of accuracy and predictability with the use of Internet of Things, since last 40 years. With the advancement of Data Science, Artificial Intelligence, Scientists now do weather forecasting with high accuracy and predictability.

1.2 Traditional Weather Forecasting

Weather forecasting is the application of science and technology to predict the state of the atmosphere for a given location. Weather forecasts are made by collecting quantitative data about the current state of the atmosphere and using scientific understanding of atmospheric processes to project how the atmosphere will evolve. There are a variety of end users to weather forecasts. Weather warnings are important forecasts because they are used to protect life and property.

In ancient times, forecasting was mostly based on weather pattern observation. Over the years, the study of weather patterns has resulted in various techniques for rainfall forecasting. Present rainfall forecasting embodies a combination of computer models, interpretation, and an acquaintance of weather patterns. The following technique was used for existing weather prediction.

Use of a barometer

Measurements of barometric pressure and the pressure tendency have been used in forecasting since the late 19th century. The larger the change in pressure, the larger the change in weather can be expected. If the pressure drop is rapid, a low pressure system is approaching, and there is a greater chance of rain

Looking at the sky

Along with pressure tendency, the condition of the sky is one of the most important parameters used to forecast weather in mountainous areas. Thickening of cloud cover or the invasion of a higher cloud deck is an indication of rain in the near future. At night, high thin clouds can lead to halos around the moon, which indicates the approach of a warm front and its associated rain. Morning fog portends fair conditions, as rainy conditions are preceded by wind or clouds which prevent fog formation

Nowcasting

The forecasting of the weather within the next six hours is often referred to as nowcasting. In this time range, it is possible to forecast smaller features such as individual showers and thunderstorms with reasonable accuracy, as well as other features too small to be resolved by a computer model. A human, given the latest radar, satellite and observational data will be able to make a better analysis of the small scale features present and so will be able to make a more accurate forecast for the following few hours

Analog technique

The analog technique is a complex way of making a forecast, requiring the forecaster to remember a previous weather event which is expected to be mimicked by an upcoming event. It remains a useful method of observing rainfall in places

such as oceans, as well as the forecasting of precipitation amounts and distribution in the future. A similar technique is used in medium range forecasting, which is known as teleconnections, when systems in other locations are used to help pin down the location of another system within the surrounding regime.

Radar

Radar stands for Radio Detection and Ranging. In radar, a transmitter sends out radio waves. The radio waves bounce off the nearest object and then return to a receiver. Weather radar can sense many characteristics of precipitation, its location, motion, intensity, and the likelihood of future precipitation. Most weather radar is Doppler radar, which can also track how fast the precipitation falls. Radar can outline the structure of a storm and in doing so estimates the possibility that it will produce severe weather condition

1.3 Objectives

Our project aims to predict the Weather and Atmosphere conditions using the previous dataset of the weather forecasting with a focus on improving the accuracy of prediction. This will increase the accuracy of the weather prediction and we will get accurate results than the traditional methods. Our dataset consists of max and min. temperature of everyday from the specific location.Classifications: When gathering datasets to give to the models there are sure parameters which are called as ordered information which incorporates: snow, rainstorm, rain, mist, cloudy, for the most part overcast, halfway shady, scattered mists, and clear.Thus our aim is to provide accurate result in order to provide correct prediction of weather for future so in critical conditions people can be aware of upcoming natural calamities.

CHAPTER-2

Literature Survey

Linear Regression Algorithm:

Before recognizing what is Linear Regression, let us get ourselves acclimated with regression. Regression is a technique for demonstrating an objective esteem dependent on free indicators. This strategy is for the most part utilized for estimating and discovering circumstances and end results connection between factors. Regression methods for the most part vary dependent on the quantity of autonomous factors and the kind of connection between the free and ward factors.

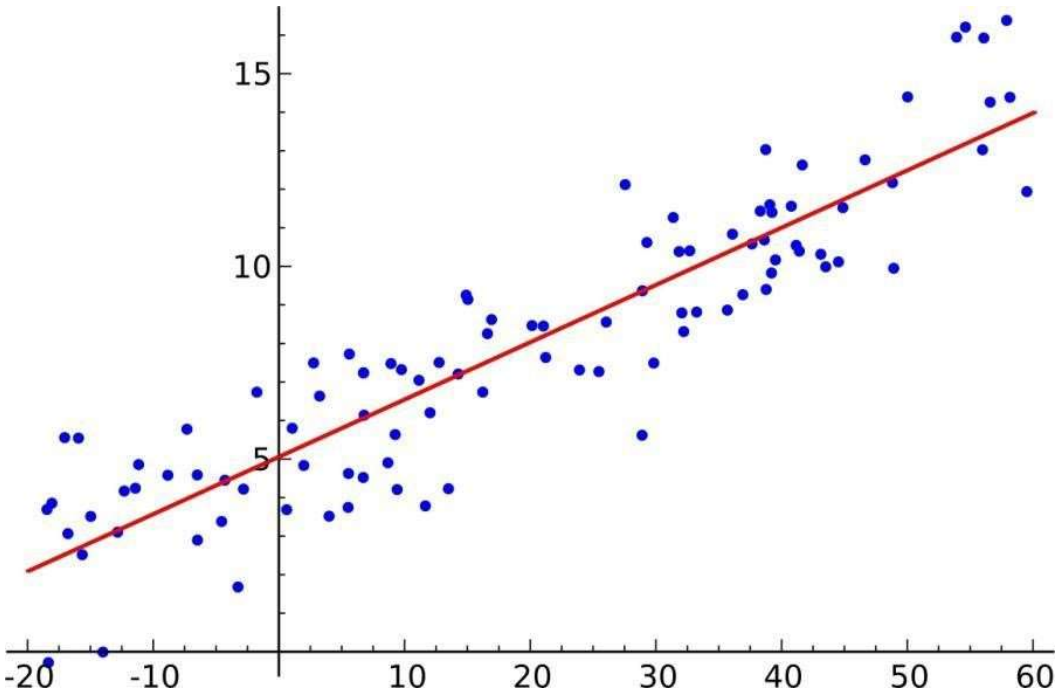
Linear Regression

Linear Regression is a machine learning algorithm used for the prediction of parameter which is in continuous nature. In this project, linear regression has been used for forecasting the minimum and maximum temperature and wind speed.

The major objectives of Linear Regression: Linear regression has been used for the following two objectives:

- In order to find the relationship among variables (here maximum temperature rainfall and minimum temperature, etc.).
- In order to estimate the values of some attributes so that new observations are entertained

Basic linear regression is a kind of regression examination where the quantity of autonomous factors is one and there is a straight connection between the independent(x) and dependent(y) variable. The red line in the above diagram is alluded to as the best fit straight line. In view of the given information focuses, we attempt to plot a line that models the focuses the best. The line can be displayed dependent on the straight condition demonstrated as follows



y = a0 * x + a1 ## Linear Equation

The intention of the linear regression calculation is to locate the best qualities for a0 and a1. Before proceeding onward to the calculation, we should view two critical ideas you should know to more readily comprehend linear regression.

Polynomial Regression

Polynomial regression is a special case of linear regression where we fit a polynomial equation on the data with a curvilinear relationship between the target variable and the independent variables.

In a curvilinear relationship, the value of the target variable changes in a non-uniform manner with respect to the predictor (s).

In Linear Regression, with a single predictor, we have the following equation:

$$Y = \theta_0 + \theta_1 x$$

where,

Y is the target,

x is the predictor,

is the bias,

and **1** is the weight in the regression equation

This linear equation can be used to represent a linear relationship. But, in polynomial regression, we have a polynomial equation of degree ***n*** represented as:

$$Y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_n x^n$$

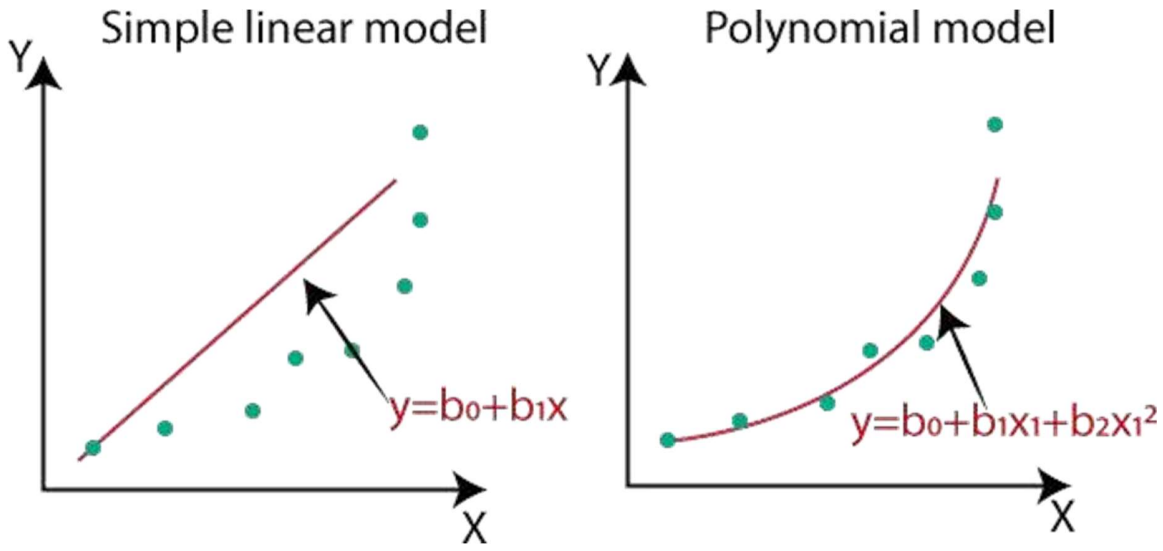
Here:

0 is the bias,

1, 2, …, n are the weights in the equation of the polynomial regression,

and ***n*** is the degree of the polynomial

The number of higher-order terms increases with the increasing value of ***n,*** and hence the equation becomes more complicated.



Chapter-3

Software Requirement Specification

3.1 Python

i)Python

3.2 Libraries

i) Numpy
ii) Scikit-learn
iii) Pandas
iv) Matplotlib

3.3 Operating System

i) Windows

Hardware Requirements Specification

I. Laptop with basic hardware.

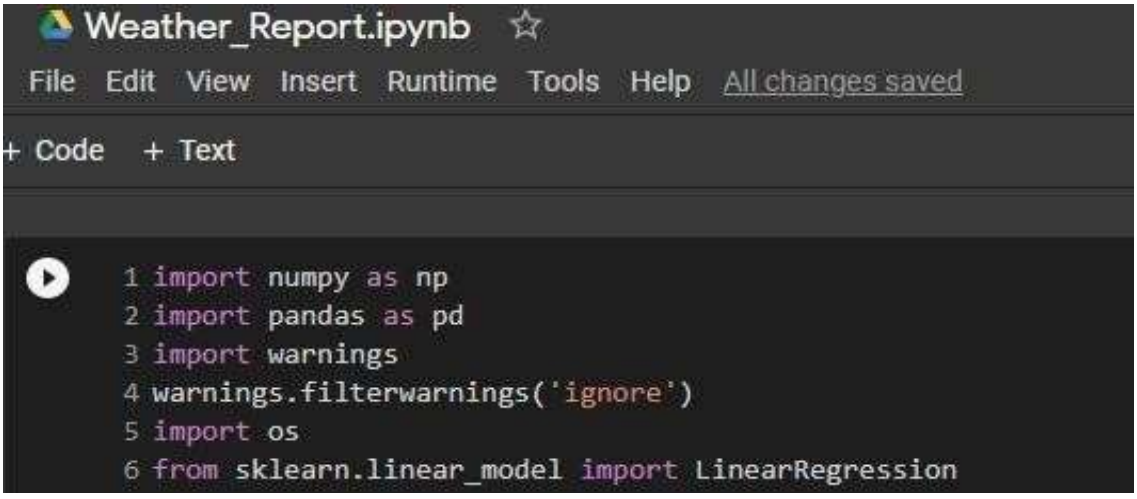
Chapter 4

Functionality / Working of Project

The dataset that I have chosen for this exercise is Weather dataset of Szeged City of Hungary. Its 10 years of data from 2006–2016 and it has hourly entries of the weather related features.

Data Set — https://www.kaggle.com/budincsevity/szeged-weather

We have first import the required libraries for the data pre - processing for the models.




```
 9 from matplotlib import pyplot as plt
10 %matplotlib inline
11 import sklearn
12 from sklearn.model_selection import train_test_split
13 from sklearn.metrics import accuracy_score
14 from sklearn import preprocessing
```

Before feeding the data into our model, we first need to make sure , Dataset is appropriate for our model or not. The data might be having some missing or null values or some not required values which need to be handles properly and replace that wrong data.

We need to understand the data well, it's really helps us for processing the data.

So our data, consists of 12 columns and many rows. The entries our made on hourly basis every day. So for each day there are 24 entries.

```
1 weather_df = pd.read_csv('weatherHistory.csv')
2 weather_df.head(10)
```

1 to

index	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibili
0	2006-04-01 00:00:00.000 +0200	Partly Cloudy	rain	9.472222222222221	7.388888888888887	0.89	14.1197	251.0	15.826300
1	2006-04-01 01:00:00.000 +0200	Partly Cloudy	rain	9.355555555555558	7.227777777777777	0.86	14.2646	259.0	15.826300
2	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	rain	9.377777777777778	9.377777777777778	0.89	3.9284	204.0	
3	2006-04-01 03:00:00.000 +0200	Partly Cloudy	rain	8.28888888888889	5.944444444444446	0.83	14.1036	269.0	15.826300
4	2006-04-01 04:00:00.000 +0200	Mostly Cloudy	rain	8.755555555555553	6.977777777777778	0.83	11.0446	259.0	15.826300
5	2006-04-01 05:00:00.000 +0200	Partly Cloudy	rain	9.222222222222221	7.111111111111111	0.85	13.9587	258.0	

```
1 weather_df.columns
```

```
Index(['Formatted Date', 'Summary', 'Precip Type', 'Temperature (C)',
       'Apparent Temperature (C)', 'Humidity', 'Wind Speed (km/h)',
       'Wind Bearing (degrees)', 'Visibility (km)', 'Loud Cover',
       'Pressure (millibars)', 'Daily Summary'],
      dtype='object')
```

```
1 weather_df.shape
```

```
(96453, 12)
```

Total 96453 Rows and 12 Columns are Available in our Dataset.

```
1 weather_df.describe()
```

index	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)
count	96453.0	96453.0	96453.0	96453.0	96453.0
mean	11.932678437511868	10.855028874166726	0.7348989663358906	10.810640140793208	187.50923247592092
std	9.551546320656923	10.696847392119263	0.19547273906722662	6.9135710125921515	107.38342838070538
min	-21.822222222222226	-27.716666666666665	0.0	0.0	0.0
25%	4.688888888888886	2.311111111111109	0.6	5.828200000000002	116.0
50%	12.0	12.0	0.78	9.9659	180.0
75%	18.838888888888892	18.838888888888892	0.89	14.1358	290.0
max	39.90555555555555	39.344444444444434	1.0	63.8526	359.0

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

1 to 8 of 8 entries Filter

Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)
96453.0	96453.0	96453.0	96453.0
187.50923247592092	10.347324929237148	0.0	1003.2359558541606
107.38342838070538	4.192123191422925	0.0	116.96990568258147
0.0	0.0	0.0	0.0
116.0	8.3398	0.0	1011.9
180.0	10.0464	0.0	1016.45
290.0	14.812000000000001	0.0	1021.09
359.0	16.1	0.0	1046.38

Now, I'm checking, Any missing values present in our dataset or not. True represents the Missing Data.

```
]   1 weather_df.isnull().any()

Formatted Date              False
Summary                     False
Precip Type                  True
Temperature (C)             False
Apparent Temperature (C)    False
Humidity                    False
Wind Speed (km/h)           False
Wind Bearing (degrees)      False
Visibility (km)             False
Loud Cover                  False
Pressure (millibars)        False
Daily Summary               False
dtype: bool
```

```
3]   1 weather_df.value_counts

<bound method DataFrame.value_counts of                       Formatted Date  ...
0      2006-04-01 00:00:00.000 +0200  ...       Partly cloudy throughout the day.
1      2006-04-01 01:00:00.000 +0200  ...       Partly cloudy throughout the day.
2      2006-04-01 02:00:00.000 +0200  ...       Partly cloudy throughout the day.
3      2006-04-01 03:00:00.000 +0200  ...       Partly cloudy throughout the day.
4      2006-04-01 04:00:00.000 +0200  ...       Partly cloudy throughout the day.
...                              ...  ...                                     ...
96448  2016-09-09 19:00:00.000 +0200  ...  Partly cloudy starting in the morning.
96449  2016-09-09 20:00:00.000 +0200  ...  Partly cloudy starting in the morning.
96450  2016-09-09 21:00:00.000 +0200  ...  Partly cloudy starting in the morning.
96451  2016-09-09 22:00:00.000 +0200  ...  Partly cloudy starting in the morning.
96452  2016-09-09 23:00:00.000 +0200  ...  Partly cloudy starting in the morning.

[96453 rows x 12 columns]>
```

Total 85224 rainy days and 10712 days with sbow.

```
1 weather_df['Precip Type'].value_counts()
```

```
rain    85224
snow    10712
Name: Precip Type, dtype: int64
```

```
weather_df.loc[weather_df['Precip Type'] == 'rain','Precip Type'] = 1

weather_df.loc[weather_df['Precip Type'] == 'snow','Precip Type'] = 0

weather_df.head(10)
```

Now, “Rain” value is replaced by binary value 1 and “Snow” value is replaced by 0.

index	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)
0	2006-04-01 00:00:00.000 +0200	Partly Cloudy	1	9.472222222222221	7.388888888888887	0.89	14.1197
1	2006-04-01 01:00:00.000 +0200	Partly Cloudy	1	9.355555555555558	7.227777777777777	0.86	14.2646
2	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	1	9.377777777777778	9.377777777777778	0.89	3.9284
3	2006-04-01 03:00:00.000 +0200	Partly Cloudy	1	8.28888888888889	5.944444444444446	0.83	14.1036
4	2006-04-01 04:00:00.000 +0200	Mostly Cloudy	1	8.755555555555553	6.977777777777778	0.83	11.0446
5	2006-04-01 05:00:00.000 +0200	Partly Cloudy	1	9.222222222222221	7.111111111111111	0.85	13.9587
6	2006-04-01 06:00:00.000 +0200	Partly Cloudy	1	7.733333333333333	5.522222222222222	0.95	12.3648
7	2006-04-01 07:00:00.000 +0200	Partly Cloudy	1	8.77222222222222	6.527777777777778	0.89	14.1519

```
[23]  1 weather_df.loc[weather_df['Precip Type'] == 0][:5]
```

index	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	W
1562	2006-12-13 02:00:00.000 +0100	Foggy	0	-0.4833333333333339	-4.15	1.0	11.0929	
1563	2006-12-13 03:00:00.000 +0100	Foggy	0	-0.4833333333333339	-4.061111111111111	0.96	10.7387	
1564	2006-12-13 04:00:00.000 +0100	Foggy	0	-0.9222222222222224	-3.4777777777777787	1.0	7.0679	
1565	2006-12-13 05:00:00.000 +0100	Foggy	0	-1.0388888888888894	-4.400000000000001	1.0	9.499	
1566	2006-12-13 06:00:00.000 +0100	Foggy	0	-1.0888888888888897	-4.438888888888886	1.0	9.4346	

```
1 weather_df_num = weather_df[list(weather_df.dtypes[weather_df.dtypes != 'object'].index)]
```

```
1 weather_df_num[:15]
```

index	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)
0	9.472222222222221	7.388888888888887	0.89	14.1197	251.0
1	9.355555555555558	7.227777777777777	0.86	14.2646	259.0
2	9.377777777777778	9.377777777777778	0.89	3.9284	204.0
3	8.28888888888889	5.944444444444446	0.83	14.1036	269.0
4	8.755555555555553	6.977777777777778	0.83	11.0446	259.0
5	9.222222222222221	7.111111111111111	0.85	13.9587	258.0
6	7.733333333333333	5.522222222222222	0.95	12.3648	259.0
7	8.77222222222222	6.527777777777778	0.89	14.1519	260.0
8	10.822222222222221	10.822222222222221	0.82	11.3183	259.0
9	13.77222222222222	13.77222222222222	0.72	12.525800000000002	279.0
10	16.016666666666666	16.016666666666666	0.67	17.5651	290.0
11	17.144444444444446	17.144444444444446	0.54	19.7869	316.0
12	17.800000000000004	17.800000000000004	0.55	21.9443	281.0
13	17.333333333333332	17.333333333333332	0.51	20.6885	289.0
14	18.87777777777778	18.87777777777778	0.47	15.375500000000002	262.0

Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)
251.0	15.826300000000002	0.0	1015.13
259.0	15.826300000000002	0.0	1015.63
204.0	14.9569	0.0	1015.94
269.0	15.826300000000002	0.0	1016.41
259.0	15.826300000000002	0.0	1016.51
258.0	14.9569	0.0	1016.66
259.0	9.982	0.0	1016.72
260.0	9.982	0.0	1016.84
259.0	9.982	0.0	1017.37
279.0	9.982	0.0	1017.22
290.0	11.2056	0.0	1017.42
316.0	11.4471	0.0	1017.74
281.0	11.27	0.0	1017.59
289.0	11.27	0.0	1017.48
262.0	11.4471	0.0	1017.17

```
]  1 weather_df.head()
```

index	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)
0	2006-04-01 00:00:00.000 +0200	Partly Cloudy	1	9.472222222222221	7.388888888888887	0.89	14.1197
1	2006-04-01 01:00:00.000 +0200	Partly Cloudy	1	9.355555555555558	7.227777777777777	0.86	14.2646
2	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	1	9.377777777777778	9.377777777777778	0.89	3.9284
3	2006-04-01 03:00:00.000 +0200	Partly Cloudy	1	8.28888888888889	5.944444444444446	0.83	14.1036
4	2006-04-01 04:00:00.000 +0200	Mostly Cloudy	1	8.755555555555553	6.977777777777778	0.83	11.0446

```
[31]  1 weather_y = weather_df_num.pop('Temperature (C)')
      2 weather_x = weather_df_num

[32]  1 train_x,test_x,train_y,test_y = train_test_split(weather_x,weather_y,test_size = 0.2, random_state=4)
```

Training set for Model Training. And Testing Set used for Prediction.

```
1 train_x.head()
```

index	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)
70626	21.06111111111111	0.31	12.558	110.0	16.1
52457	25.016666666666666	0.36	18.498900000000006	352.0	10.3523
90690	0.7388888888888879	0.89	17.1304	270.0	15.826300000000002
69528	13.77222222222222	0.78	14.49	300.0	15.826300000000002
92419	23.288888888888888	0.82	6.391700000000001	357.0	16.1

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

1 to 5 of 5 entries Filter

Visibility (km)	Loud Cover	Pressure (millibars)
16.1	0.0	1005.87
10.3523	0.0	1025.36
15.826300000000002	0.0	1014.75
15.826300000000002	0.0	1014.56
16.1	0.0	1022.05

```
[ ]  1 train_y.head()

70626    21.061111
52457    25.016667
90690     4.422222
69528    13.772222
92419    23.288889
Name: Temperature (C), dtype: float64
```

Now, Linear Regression Model applied for our Model.

Linear Regression Model

```
[36]  1 model = LinearRegression()
      2 model.fit(train_x,train_y)

      LinearRegression()
```

```
[37]  1 prediction = model.predict(test_x)
```

```
[44]  1 np.mean((prediction-test_y)**2)

      0.902274371188337
```

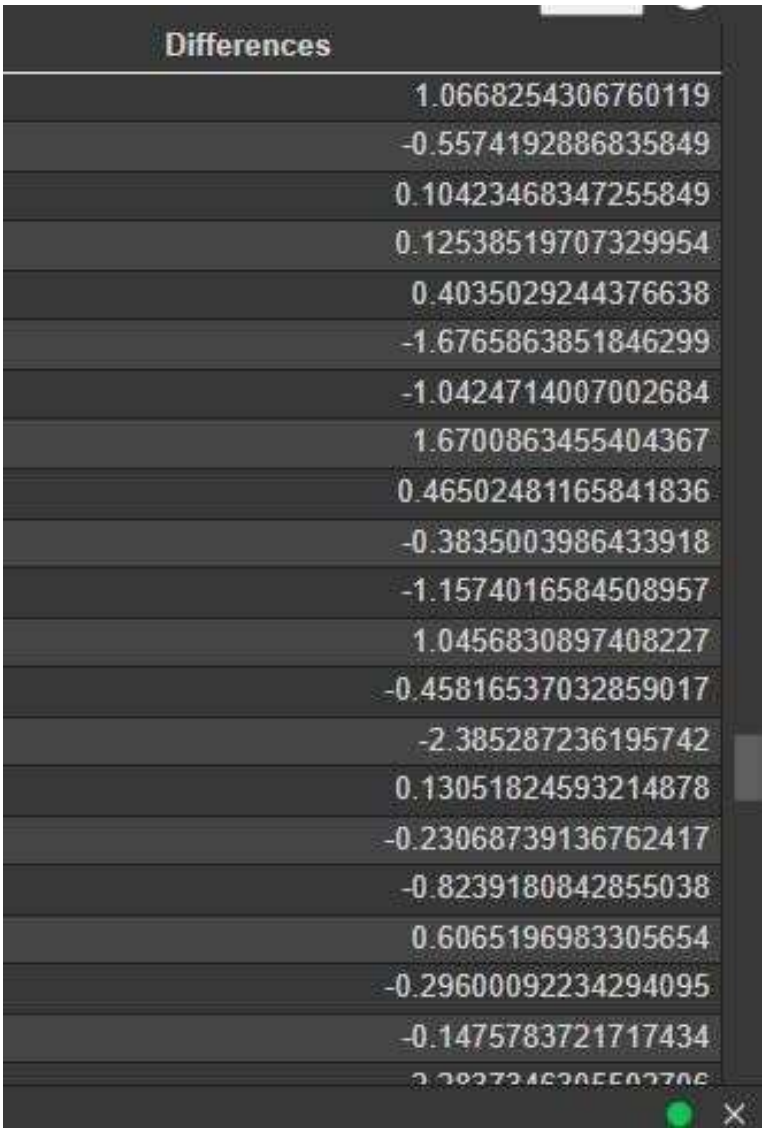
```
[72]  1 print(f"Model Error is  {str(round(np.mean((prediction-test_y)**2) * 100,2))} %")

      Model Error is  90.23 %

[89]  1 model_efficiency = 100 - np.mean((prediction-test_y)**2)*100
      2 print(f"Model Efficiency is   {model_efficiency} %")

      Model Efficiency is   9.772562881166309 %
```

```
1 pd.DataFrame({'Actual_Value' : test_y,
2               'Prediction' : prediction,
3               'Differences' : (test_y - prediction)})
```

This is the Actual and Predicted value by our Linear Regression Model.

index	Actual_Value	Prediction
37443	-2.2888888888888896	-3.3557143195649015
86534	8.861111111111112	9.418530399794697
2082	9.805555555555557	9.701320872082999
53130	27.222222222222218	27.09683702514892
45196	17.705555555555556	17.302052631117892
57822	3.888888888888889	5.565475274073519
26754	17.777777777777782	18.82024917847805
53177	28.977777777777767	27.30769143223733
7855	7.727777777777777	7.262752966119359
34256	9.949999999999998	10.33350039864339
95437	12.716666666666667	13.874068325117562
45440	3.7722222222222217	2.726539132481399
5591	9.522222222222222	9.980387592550812
73484	-7.3277777777777775	-4.9424905415820355
59641	5.505555555555554	5.375037309623405
56484	23.93333333333333	24.164020724700954
17287	13.783333333333335	14.607251417618839
63335	4.816666666666666	4.210146968336101
69767	12.488888888888885	12.784889811231826
84937	12.416666666666664	12.564245038838408
26784	12.222222222222221	14.595956852772402

Differences Between Actual value and Predicted

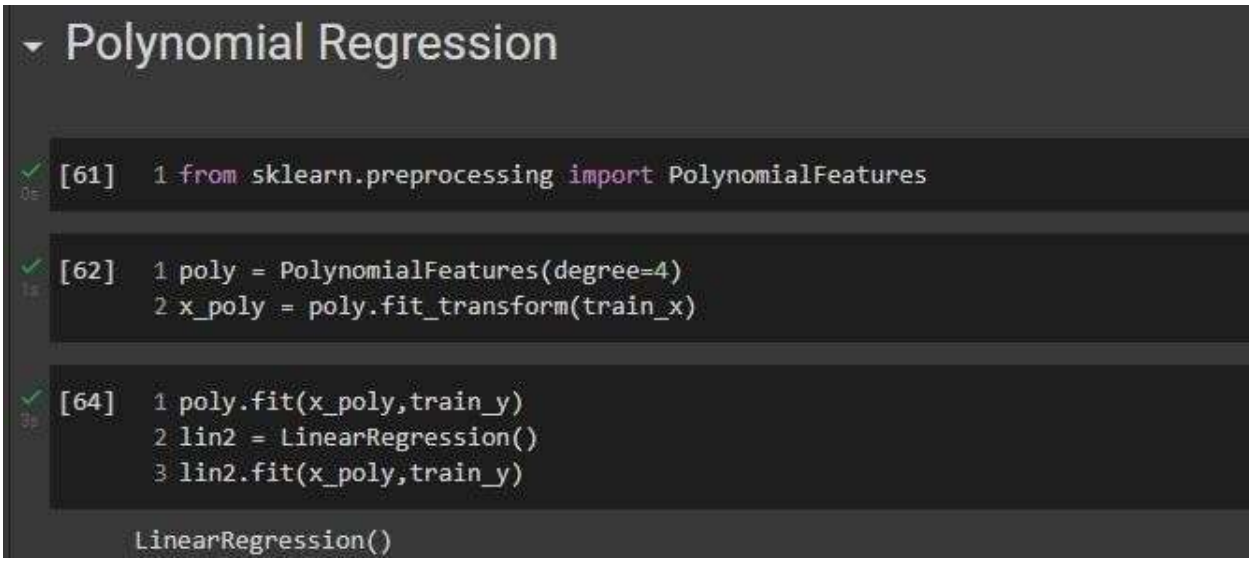


Differences
1.0668254306760119
-0.5574192886835849
0.10423468347255849
0.12538519707329954
0.4035029244376638
-1.6765863851846299
-1.0424714007002684
1.6700863455404367
0.46502481165841836
-0.3835003986433918
-1.1574016584508957
1.0456830897408227
-0.45816537032859017
-2.385287236195742
0.13051824593214878
-0.230687391367624 17
-0.8239180842855038
0.6065196983305654
-0.29600092234294095
-0.1475783721717434

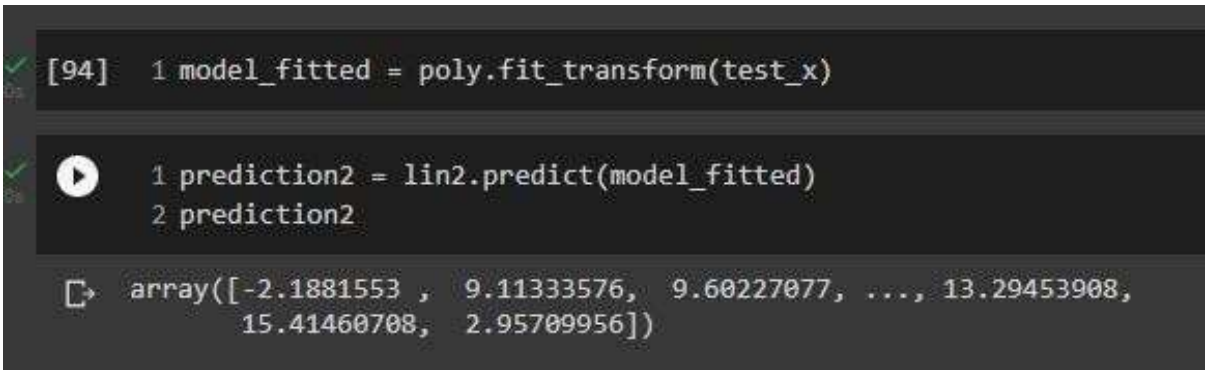
Now, Our Linear Regression model Efficiency is very Bad, Efficiency come out to be only 10 %.

So, We are going to use **Polynomial Regression Algorithm.**

Polynomial Regression Class has been imported from Sklearn Library. And we are going to first create our model object and then we will fit our model tothe Regression Object.



Now, Model is fitted into our Regression Instance.



```
[96]  1 np.mean((prediction2-test_y)**2)

      0.14602317508410093

      1 print(f"Model Error is {round(np.mean((prediction2-test_y)**2),2) * 100} %")

      Model Error is 15.0 %
```

Now, Our Polynomial Regression Model has only 15% Error Prediction.

```
[87]  1 model_efficiency = 100 - round(np.mean((prediction2-test_y)**2)*100,3)
      2 print(f"Model Efficiency is   {model_efficiency} %")

      Model Efficiency is   85.398 %
```

Model Efficiency is 85% and it is the Efficient Model but not the most efficient for our Dataset.

```
[67]  1 pd.DataFrame({'Actual_Value' : test_y,
      2                'Prediction' : prediction2,
      3                'Differences' : (test_y - prediction2)})
```

Actual, Predicited and Difference between Actual and Predicted values are :

[67]

index	Actual_Value
37443	-2.2888888888888896
86534	8.861111111111112
2082	9.805555555555557
53130	27.222222222222218
45196	17.705555555555556
57822	3.888888888888889
26754	17.777777777777782
53177	28.977777777777767
7855	7.727777777777777
34256	9.949999999999998
95437	12.716666666666667
45440	3.7722222222222217
5591	9.522222222222222
73484	-7.3277777777777775
59641	5.505555555555554
56484	23.93333333333333
17287	13.783333333333335
63335	4.816666666666666
69767	12.488888888888885
84937	12.416666666666664
26781	12.222222222222221
55424	-12.222222222222221

Predicted values by our Polynomial Regression Model

Prediction
-2.188155298368418
9.113335764759096
9.60227077369542
27.130414359458666
17.770751431830842
4.267411908143924
17.75757524578513
28.706245846157582
6.969903562102228
10.10144412810224
13.467815262026317
3.7133855228150985
9.429887300608254
-6.747399715736481
5.2564073045059665
23.920775005316965
14.240158078551996
4.630587569870169
12.649292847734625
12.638699798842264
12.888067374345393
-12.710136834539515

Differences Between Actual and Predicted values by our Model are :

Differences
-0.1007335905204716
-0.25222465364798374
0.20328478186013754
0.09180786276355235
-0.06519587627528622
-0.37852301925503484
0.020202531992651984
0.27153193162018496
0.7578742156755487
-0.15144412810224317
-0.7511485953596502
0.058836699407123216
0.09233492161396839
-0.5803780620412962
0.24914825104958727
0.012558328016364584
-0.4568247452186611
0.18607909679649737
-0.16040395884573933
-0.22203313217559995
-0.6658451521231719
0.4879146123172937

Now, Our Polynomial Regression model Efficiency is good but not the best, Efficiency come out to be only 85 %.

So, We are going to use **Random Forest Algorithm.**

Chapter-5

Result and Discussion

In this project we have used three regression model for weather Forecasting. In which in Linear Regression model Efficiency was very Bad as there was more error prediction percentage and that result in low Efficiency of only 10 %.

So, We have used Polynomial Regression Algorithm**.** In Polynomial Regression Model there was only 15% Error Prediction and Model Efficiency is 85% and thus it was far better than Linear Regression model but not the most efficient.

So, we used Random Forest Regression Model. In this there was 0.2% error prediction and the Model Efficiency is 99% and thus it was the best regression model for our data set.

There are also other machine learning algorithms through which we can more efficient results.In which we have used above three algorithms and shown their efficiency rate in weather forecasting.

```
[72]  1 print(f"Model Error is  {str(round(np.mean((prediction-test_y)**2) * 100,2))} %")

      Model Error is  90.23 %

[89]  1 model_efficiency = 100 - np.mean((prediction-test_y)**2)*100
      2 print(f"Model Efficiency is   {model_efficiency} %")

      Model Efficiency is   9.772562881166309 %
```

Linear Regression model efficiency

```
[87]  1 model_efficiency = 100 - round(np.mean((prediction2-test_y)**2)*100,3)
      2 print(f"Model Efficiency is   {model_efficiency} %")

Model Efficiency is   85.398 %
```

Polynomial Regression Model Efficiency

```
[46] model_efficiency = 100 - round(np.mean((y_test_pred_3-test_y)**2)*100,3)
     print(f"Model Efficiency is   {model_efficiency} %")

     Model Efficiency is   99.773 %

[47] test_pred_flat_array3 = np.ravel(y_test_pred_3) # converted 2 dimensional array into 1 dimensional array
     test_pred_flat_array3[:5]

     array([-2.27666667,  8.85944444,  9.845     , 27.20777778, 17.70555556])
```

Random Forest Regression Model Efficiency

We can easily notice the big difference between the three regression model and can conclude Random Fregression Model give more accurate result in weather Forecasting.

Chapter-6

Conclusion and Future Scope

In this paper, we presented a technology to utilize machine learning techniques to provide weather forecasts. Machine learning technology can provide intelligent models, which are much simpler than traditional physical models. They are less resource-hungry and can easily be run on almost any computer including mobile devices. Our evaluation results show that these machine learning models can predict weather features accurately enough to compete with traditional models.

The most scientific and technical challenging problem around the world is forecasting the weather. Weather Prediction relies on two correct things 1) First the collection of the data from the meteorological department and 2) the appropriate selection of the data mining techniques for predicting the weather conditions. The major concerns of Weather prediction are the Accuracy of the model and its Timely output. The Problem domain of Weather Forecasting is very vast and therefore it is very feasible to use data mining techniques which can perform in a thorough manner with the complex problem domain of weather forecasting and give some accurate results. However more than one data mining technique is applied in parallel for better and accurate results for the weather prediction. The proposed work is an attempt to forecast different weather conditions using a fusion of different forecasting and data mining techniques. Even though the rainfall is dependent on many parameters, the proposed model was able to get an impressive classification accuracy using limited parameters.

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 of more information.

Chapter-7

Refrences

- https://numpy.org/doc/
- https://pandas.pydata.org/docs/
- https://matplotlib.org/
- https://scikit-learn.org/stable/
- https://realpython.com/linear-regression-in-python/
- https://www.geeksforgeeks.org/python-implementation-of-polynomial-regression/