PROJECT REPORT

ON WEATHER PREDICTION

Submitted in part of the Curriculum of

Degree of Bachelor of Technology
In
COMPUTER SCIENCE AND ENGINEERING
Submitted by:

G. LAHARI REDDY (21AG1A0586)

CH. YEDUKONDALA SRINIVAS (21AG1A0580)

M.SAI CHARAN (21AG1A05A0)

D. MANASA(21AG1A0585)

MD. TABREZ (21AG1A05A1)

CH. SHIRISHA (21AG1A0579)

M.SRI VAISHNAVI (21AG1A05A2)

R. POOJITHA (22AG5A0512)

Y. VIVEK REDDY (21AG1A05C9)

D. TANISHQ (21AG1A0584)

Under the guidance of
Mr. D. KRISHNA (Associative professor)

Department of Computer Science and Engineering

ACE Engineering College

(an Autonomus institution)

NBA ACCREDITED B. TECH COURSES: EEE, ECE, CSE & MECH
Ankushapur (V), Ghatkesar (M), Medchal. Dist. – 501 301
(Affiliated to Jawaharlal Nehru Technological UniversityHyderabad)

(2021-2025)

Ankushapur (V), Ghatkesar (M), Medchal. Dist. – 501 301

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Website: www.aceec.ac.in E-mail: info@aceec.ac.in

CERTIFICATE

This is to certify that the project work entitled "**WEATHER PREDICTION**" is being submitted by **G. LAHARI REDDY (21AG1A0586), M.SAI CHARAN (21AG1A05A0), MD. TABREZ (21AG1A05A1) M.SRI VAISHNAVI (21AG1A05A2,) Y. VIVEK REDDY (21AG1A05C9), CH. YEDUKONDALA SRINIVAS (21AG1A0580), D. MANASA(21AG1A0585), CH. SHIRISHA (21AG1A0579), R. POOJITHA (22AG5A0512), D. TANISHQ (21AG1A0584)** as a part of Curriculum of Degree of Bachelor of Technology in Computer Science and Engineering to the **ACE Engineering College** during the academic year 2021- 2025 is a record of Bonafide work carried out by them under our guidance and supervision.

Guide
Mr. KRISHNA
Associate Professor

Head of the Department
Dr.M.V.VIJAYA SARADHI
Professor and Head of the Dept.CSE

ACKNOWLEDGEMENT

We would like to express our gratitude to all the people behind the screen who have helped us to transform an idea into a real time application. We would like to express our heart-felt gratitude to our parents without whom we would not have been privileged to achieve and fulfill our dreams.

A special thanks to our Secretary, **Prof. Y. V. GOPALA KRISHNA MURTHY**, for having founded such an esteemed institution. We are also grateful to our beloved principal, **Dr. B. L. RAJU** for permitting us to carry out this project. We profoundly thank **Dr.M.V.VIJAYA SARADHI**, Head of the Department of Computer Science & Engineering.

We are very thankful to our guide **D. KRISHNA**, **Associate Professor,** who has been excellent and given continuous support for the completion of our project work.

The satisfaction and euphoria the accompany the successful completion of the task would be great, but incomplete without the mention of the people who made it possible, whose guidance and encouragement crown all the efforts with success. In this context, we would like to thank all the other staff members, both teaching and non-teaching, which have extended their timely help and easier our task.

G. LAHARI REDDY (21AG1A0586)

M.SAI CHARAN (21AG1A05A0)

MD. TABREZ (21AG1A05A1)

M.SRI VAISHNAVI (21AG1A05A2)

Y. VIVEK REDDY (21AG1A05C9)

CH. YEDUKONDALA SRINIVAS (21AG1A0580)

D. MANASA(21AG1A0585)

CH. SHIRISHA (21AG1A0579)

R. POOJITHA (22AG5A0512)

D. TANISHQ (21AG1A0584)

DECLARATION

We hereby declare that the project entitled "**WEATHER PREDICTION**" was submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering. This dissertation is our original work, and the project has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles and no part of it has been published or sent for publication at the time of submission.

G. LAHARI REDDY (21AG1A0586)

CH. YEDUKONDALA SRINIVAS (21AG1A0580)

M.SAI CHARAN (21AG1A05A0)

D. MANASA(21AG1A0585)

MD. TABREZ (21AG1A05A1)

CH. SHIRISHA (21AG1A0579)

M.SRI VAISHNAVI (21AG1A05A2)

R. POOJITHA (22AG5A0512)

Y. VIVEK REDDY (21AG1A05C9)

D. TANISHQ (21AG1A0584)

INDEX

INTRODUCTION

Weather prediction is the task of predicting the atmosphere at a future time and a given area. This was done through physical equations in the early days in which the atmosphere is considered fluid. The current state of the environment is inspected, and the future state is predicted by solving those equations numerically, but we cannot determine very accurate weather for more than 10 days (about 1 and a half weeks) 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 many applications, including climate-related studies, energy, agricultural, medical, etc.

There are numerous kinds of machine learning calculations, which are Linear Regression, Polynomial Regression, Random Forest Regression, Artificial Neural Network, and Recurrent Neural Network. These models are prepared dependent on the authentic information gave of any area. Contribution to these models is given, for example, if anticipating temperature, least temperature, mean air weight, greatest temperature, mean dampness, and order for 2 days. Considering this Minimum Temperature and Maximum Temperature of 7 days will be accomplished.

1.1 MACHINE LEARNING

Machine Learning is relatively robust to perturbations and does not need any other physical variables for prediction. Therefore, machine learning is a much better opportunity in the evolution of weather forecasting. Before the advancement of Technology, weather forecasting was a hard nut to crack. Weather forecasters relied upon satellites, data model's atmospheric conditions with less accuracy. Weather prediction and analysis have vastly increased in terms of accuracy and predictability with the use of the Internet of Things, for the last 40 years. With the advancement of Data Science, Artificial Intelligence, Scientists now do weather forecasting with high accuracy and predictability.

1.2 USE OF ALGORITHMS:

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

METHODOLOGY

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

Importing Packages

```
import warnings

#provide a warning
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import preprocessing
%matplotlib inline
```

Fig 2.1 : Importing packages

Read the csv file into weather_df

```
weather_df = pd.read_csv("E:\Desktop\MachineLearning\datasets\weather_prediction_dataset.csv", parse_dates = ['DATE'], index_col=
weather_df.head()
```

DATE	MONTH	BASEL_cloud_cover	BASEL_humidity	BASEL_pressure	BASEL_global_radiation	BASEL_precipitation	BASEL_sunshine	BASEL_temp_mean	BA
2000-01-01	1	8	0.89	1.0286	0.20	0.03	0.0	2.9	
2000-01-02	1	8	0.87	1.0318	0.25	0.00	0.0	3.6	
2000-01-03	1	5	0.81	1.0314	0.50	0.00	3.7	2.2	
2000-01-04	1	7	0.79	1.0262	0.63	0.35	6.9	3.9	
2000-01-05	1	5	0.90	1.0246	0.51	0.07	3.7	6.0	

5 rows × 164 columns

Fig 2.2 : Raw Dataset

Checking Shape

```
weather_df.shape
```

```
(3654, 164)
```

Fig 2.3 : Shape of raw dataset

Checking for null values in the dataframe/dataset ¶

```
weather_df.isnull().any()
```

```
MONTH                    False
BASEL_cloud_cover        False
BASEL_humidity           False
BASEL_pressure           False
BASEL_global_radiation   False
                         ...
TOURS_global_radiation   False
TOURS_precipitation      False
TOURS_temp_mean          False
TOURS_temp_min           False
TOURS_temp_max           False
Length: 164, dtype: bool
```

Fig 2.4 : Checking for null values in raw dataset

Describing the dataframe

```
weather_df.describe()
```

	MONTH	BASEL_cloud_cover	BASEL_humidity	BASEL_pressure	BASEL_global_radiation	BASEL_precipitation	BASEL_sunshine	BASEL_temp_mean
count	3654.000000	3654.000000	3654.000000	3654.000000	3654.000000	3654.000000	3654.000000	3654.000000
mean	6.520799	5.418446	0.745107	1.017876	1.330380	0.234849	4.661193	11.022797
std	3.450083	2.325497	0.107788	0.007962	0.935348	0.536267	4.330112	7.414754
min	1.000000	0.000000	0.380000	0.985600	0.050000	0.000000	0.000000	-9.300000
25%	4.000000	4.000000	0.670000	1.013300	0.530000	0.000000	0.500000	5.300000
50%	7.000000	6.000000	0.760000	1.017700	1.110000	0.000000	3.600000	11.400000
75%	10.000000	7.000000	0.830000	1.022700	2.060000	0.210000	8.000000	16.900000
max	12.000000	8.000000	0.980000	1.040800	3.550000	7.570000	15.300000	29.000000

8 rows × 164 columns

Fig 2.5 : Describing the raw dataframe

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

3.1 PREPARING NEW DATAFRAME :

Separate the features that predict the weather from the rest of the features and create a new dataframe

```
weather_df_new = weather_df.loc[:,['BASEL_cloud_cover', 'BASEL_humidity', 'BASEL_pressure',
       'BASEL_global_radiation', 'BASEL_precipitation', 'BASEL_sunshine',
       'BASEL_temp_mean', 'BASEL_temp_min', 'BASEL_temp_max',
       'STOCKHOLM_temp_min', 'STOCKHOLM_temp_max', 'TOURS_wind_speed',
       'TOURS_humidity', 'TOURS_pressure', 'TOURS_global_radiation',
       'TOURS_precipitation', 'TOURS_temp_mean', 'TOURS_temp_min',
       'TOURS_temp_max']]
weather_df_new.head()
```

DATE	BASEL_cloud_cover	BASEL_humidity	BASEL_pressure	BASEL_global_radiation	BASEL_precipitation	BASEL_sunshine	BASEL_temp_mean	BASEL_tem
2000-01-01	8	0.89	1.0286	0.20	0.03	0.0	2.9	
2000-01-02	8	0.87	1.0318	0.25	0.00	0.0	3.6	
2000-01-03	5	0.81	1.0314	0.50	0.00	3.7	2.2	
2000-01-04	7	0.79	1.0262	0.63	0.35	6.9	3.9	
2000-01-05	5	0.90	1.0246	0.51	0.07	3.7	6.0	

Fig 3.1 : New Dataframe created after separating features

Checking shape of new dataframe

```
weather_df_new.shape
```

```
(3654, 19)
```

Checking columns in new dataframe

```
weather_df_new.columns
```

```
Index(['BASEL_cloud_cover', 'BASEL_humidity', 'BASEL_pressure',
       'BASEL_global_radiation', 'BASEL_precipitation', 'BASEL_sunshine',
       'BASEL_temp_mean', 'BASEL_temp_min', 'BASEL_temp_max',
       'STOCKHOLM_temp_min', 'STOCKHOLM_temp_max', 'TOURS_wind_speed',
       'TOURS_humidity', 'TOURS_pressure', 'TOURS_global_radiation',
       'TOURS_precipitation', 'TOURS_temp_mean', 'TOURS_temp_min',
       'TOURS_temp_max'],
      dtype='object')
```

Fig 3.2 : Shape and columns of new dataframe

Plotting all the column values w.r.t DATE column

```
10]: weather_df_new.plot(subplots=True, figsize=(25,20))
```

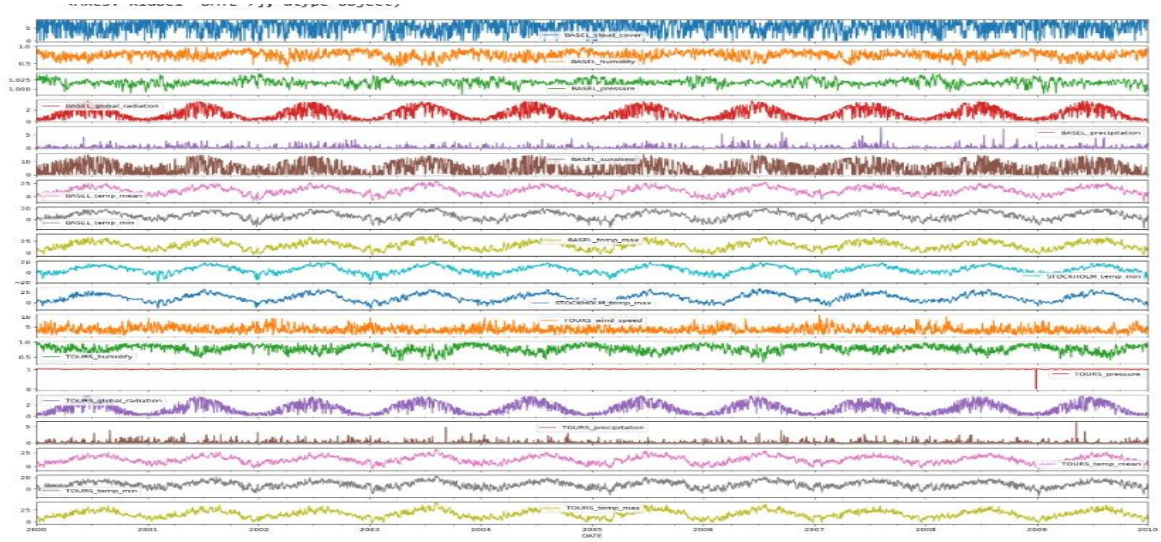

Fig 3.3 : Plot for each factor for 10 years

Plotting column values for 1 year

```
weather_df_new['2000':'2001'].resample('D').fillna(method='pad').plot(subplots=True, figsize=(25,20))
```

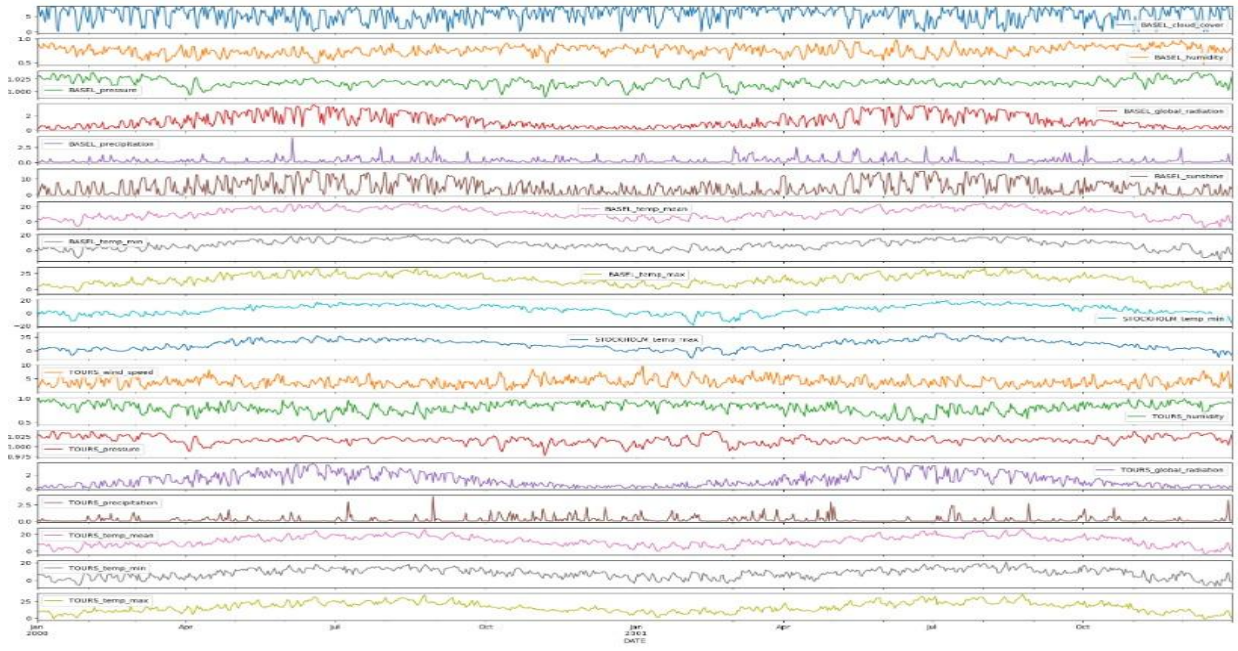

Fig 3.3 : Plot for each factor for 1 year

Hist plots

```
weather_df_new.hist(bins=10,figsize=(15,15))
```

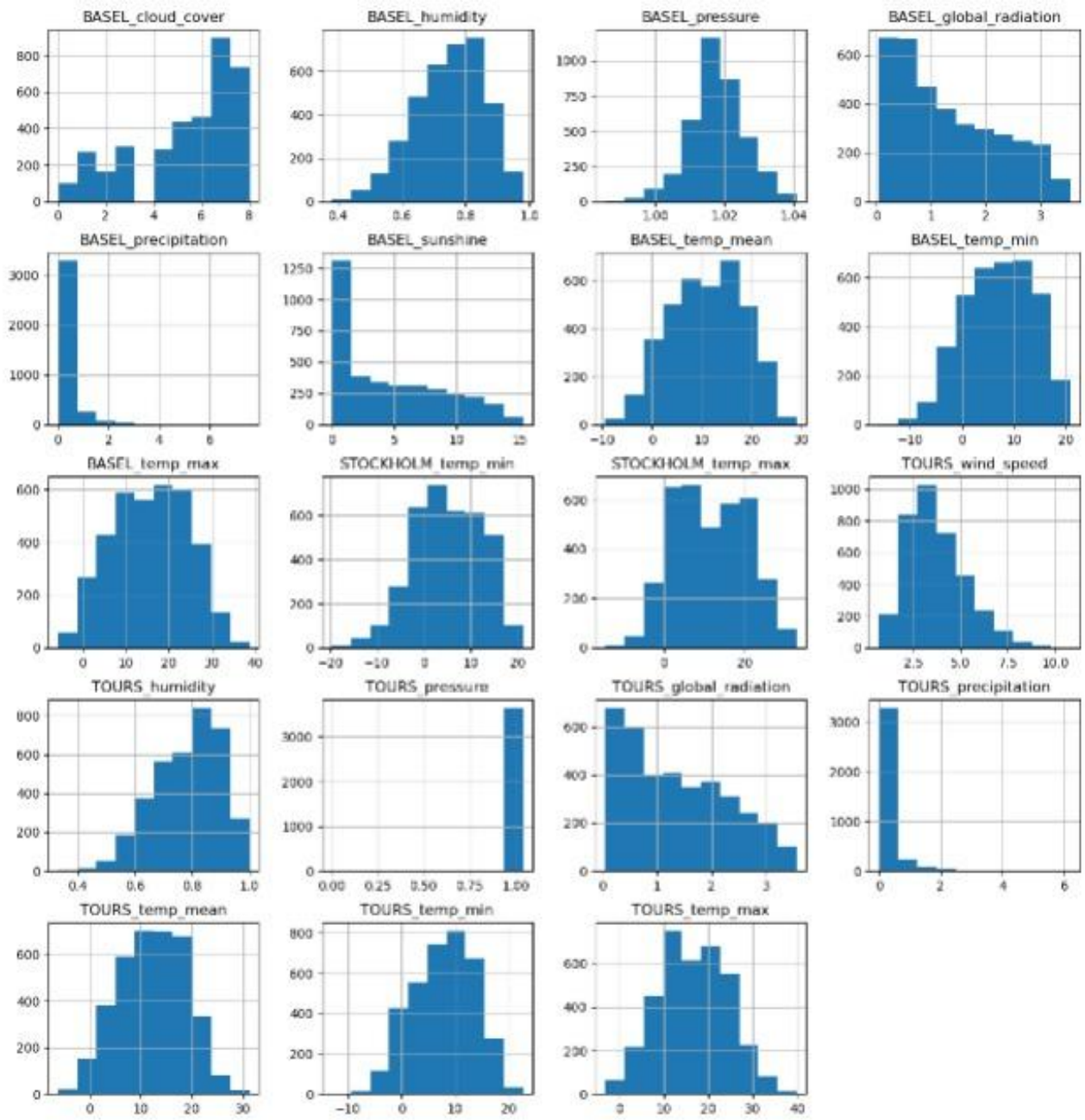

Fig 3.4 : Hist Plot

```
weather_2000_2001 = weather_df_new['2000':'2001']
weather_2000_2001.head()
```

DATE	BASEL_cloud_cover	BASEL_humidity	BASEL_pressure	BASEL_global_radiation	BASEL_precipitation	BASEL_sunshine	BASEL_temp_mean	BASEL_tem
2000-01-01	8	0.89	1.0286	0.20	0.03	0.0	2.9	
2000-01-02	8	0.87	1.0318	0.25	0.00	0.0	3.6	
2000-01-03	5	0.81	1.0314	0.50	0.00	3.7	2.2	
2000-01-04	7	0.79	1.0262	0.63	0.35	6.9	3.9	
2000-01-05	5	0.90	1.0246	0.51	0.07	3.7	6.0	

Fig 3.5 : Data of 1 year is taken from the dataframe

```
weather_y = weather_df_new.pop("TOURS_temp_min")
weather_x = weather_df_new
```

Fig 3.6 : Single feature is taken

3.2 TRAINING THE MODEL :

```
train_X,test_X,train_y,test_y=train_test_split(weather_x,weather_y,test_size=0.2,random_state=4)
```

```
train_X.shape
```

```
(2923, 18)
```

```
train_y.shape
```

```
(2923,)
```

```
train_y.head()
```

```
DATE
2003-11-06     4.5
2004-03-19     7.8
2000-09-25    10.2
2004-12-22     1.2
2006-09-05    16.7
Name: TOURS_temp_min, dtype: float64
```

Fig 3.7 : Training the model

3.2 DECISION TREE REGRESSION :

```
In [19]: from sklearn.tree import DecisionTreeRegressor
         regressor=DecisionTreeRegressor(random_state=0)
         regressor.fit(train_X,train_y)

Out[19]: DecisionTreeRegressor(random_state=0)
         In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
         On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

Fig 3.8 : Decision tree regression

```
#variance
print('Variance score: %.2f' % regressor.score(test_X, test_y))
```

```
Variance score: 0.98
```

Fig 3.9 : Variance score using decision tree regression

3.3 RANDOM FOREST REGRESSION :

```
In [26]: from sklearn.ensemble import RandomForestRegressor
         reg = RandomForestRegressor(max_depth=90,random_state=0,n_estimators=100)
         reg.fit(train_X,train_y)

Out[26]: RandomForestRegressor(max_depth=90, random_state=0)
         In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
         On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

Fig 3.10 : Random forest regression

```
print('Variance score: %.2f' % reg.score(test_X, test_y))
```

```
Variance score: 0.99
```

Fig 3.11 : Variance score using Random forest regression

RESULT AND DISCUSSION

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

Decision Tree Regression:

This regression model has medium mean absolute error, hence turned out to be the little accurate model. Given below is a snapshot of the actual result from the project implementation of Decision tree regression.

Calculating R2-score for Decision Tree Regression

```
print("Mean absolute error: %.2f" % np.mean(np.absolute(prediction1 - test_y)))
print("Residual sum of squares (MSE): %.2f" % np.mean((prediction1 - test_y) ** 2))
print("R2-score: %.2f" % r2_score(test_y,prediction1 ) )
```

```
Mean absolute error: 0.58
Residual sum of squares (MSE): 0.65
R2-score: 0.98
```

Fig 4.1 : R2-score using Decision tree regression

DATE	Actual	Prediction	diff
2007-09-02	10.2	9.8	0.4
2008-12-11	-1.5	-1.3	-0.2
2000-11-27	6.6	6.8	-0.2
2003-06-03	15.3	14.8	0.5
2001-02-13	6.7	6.4	0.3
...	...	...	...
2007-06-07	14.9	14.3	0.6
2008-10-11	7.9	8.1	-0.2
2003-02-10	3.0	2.9	0.1
2006-05-07	8.6	9.6	-1.0
2008-02-28	8.4	8.3	0.1

731 rows × 3 columns

Fig 4.2: Table for prediction for Decision tree regression

```
print(f"the score of the model for weather prediction is: {score*100:.2f}%")
```

```
the score of the model for weather prediction is: 98.00%
```

Fig 4.3: The final score of the model using Decision Tree Regression

Random Forest Regression:

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

Calculating R2-score for Random Forest Regression

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

```
Mean absolute error: 0.31
Residual sum of squares (MSE): 0.21
R2-score: 0.99
```

Fig 4.4 : R2-score using Random forest regression

DATE	actual	predicted	differences
2007-09-02	10.2	10.285	0.085
2008-12-11	-1.5	-1.480	0.020
2000-11-27	6.6	6.967	0.367
2003-06-03	15.3	15.066	-0.234
2001-02-13	6.7	6.259	-0.441
...	...	...	...
2007-06-07	14.9	14.538	-0.362
2008-10-11	7.9	8.624	0.724
2003-02-10	3.0	2.814	-0.186
2006-05-07	8.6	8.391	-0.209
2008-02-28	8.4	8.245	-0.155

731 rows × 3 columns

Fig 4.4 : Table for prediction for Random forest regression

```
print(f"the score of the model for weather prediction is: {score*100:.2f}%")
```

```
the score of the model for weather prediction is: 99.34%
```

Fig 4.5: The final score of the model using Random Forest Regression

CONCLUSION

All the machine learning models: linear regression, various linear regression, decision tree regression, and random forest regression were beaten by expert climate determining apparatuses, even though the error in their execution reduced significantly for later days, demonstrating that over longer timeframes, our models may beat genius professional ones. Linear regression demonstrated to be a low predisposition, high fluctuation model though polynomial regression demonstrated to be a high predisposition, low difference model.

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

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

REFERENCE

DATASET:

weather_prediction_dataset.csv

DECISION TREE IN MACHINE LEARNING:

https://youtu.be/RmajweUFKvM?si=EdaxATB01b0CMQkO

RANDOM FOREST ALGORITHM:

https://youtu.be/3LQI-w7-FuE?si=6IfxElLTK_tTOHhr