WEATHER PREDICTION USING MACHINE LEARNING A PROJECT REPORT

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of

KAMLESH GUPTA

Project Carried Out At

ARDENT COMPUTECH PVT. LTD.
High-End Technology Training and Project

Ardent Computech Pvt Ltd (An ISO 9001:2015 Certified)

CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064

Submitted By

GANGAJYOTI DHARA

PRONOY ROY

JLD ENGINEERING AND MANAGEMENT COLLEGE

JUNE – JULY 2018

In association with

ARDENT COMPUTECH PVT. LTD.
High-End Technology Training and Project

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

1. Title of the Project: **WEATHER PREDICTION USING MACHINE LEARNING**
2. Project Members: **1. GANGAJYOTI DHARA 2. PRONOY ROY**
3. Name and Address of the Guide: **KAMLESH GUPTA**

   Subject Matter Expert & Trainee Software Engineer (Machine Learning)
   Ardent Computech Pvt Ltd (An ISO 9001:2015 Certified) Module 132,
   SDF Building, Sector 5, Salt Lake, Kolkata - 64

	Ph.D*	M.Tech.*	B.E*/B.Tech.*	MCA*	M.Sc.*
4. Educational Qualification of the Guide:				Y	

5. Working / Training experience of the Guide:

6. Project Version Control History

Version	Primary Authors	Description of Version	Date Completed
Final	**GANGAJYOTI DHARA** **PRONOY ROY**	Project Report	

1.

2.

Signatures of Team Members

Date:

Signature of Approval

Date:

For Office Use Only

Mr. Kamlesh Gupta
Project Proposal Evaluator

Approved	**Not Approved**

PROJECT RESPONSIBILITY FORM

WEATHER PREDICTION USING MACHINE LEARNING

GROUP NO.	SL. NO.	NAME OF MEMBER	RESPONSIBILITY
2	1	**GANGAJYOTI DHARA**	Coding and testing
	2	**PRONOY ROY**	Coding and designing

Each group member must participate in project development and developing the ideas for the required elements. Individual group members will be responsible for completing tasks which help to finalize the project and the performance. All group members must be assigned a task.

Date:

Name of the Students

1. Gangajyoti Dhara
2. Pronoy Roy

Signatures of the students

a.

b.

DECLARATION

We hereby declare that the project work being presented in the project proposal entitled **"WEATHER PREDICTION USING MACHINE LEARNING"** in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** at **ARDENT COMPUTECH PVT. LTD, SALTLAKE, KOLKATA, WEST BENGAL,** is an authentic work carried out under the guidance of **MR. KAMLESH GUPTA**. The matter embodied in this project work has not been submitted elsewhere for the award of any degree of our knowledge and belief.

Date:

Name of the Students

1. Gangajyoti Dhara
2. Pronoy Roy

Signature of the students

a.

b.

Ardent Computech Pvt Ltd (An ISO 9001:2008 Certified)

CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064

CERTIFICATE

This is to certify that this proposal of minor project entitled **"WEATHER PREDICTION USING MACHINE LEARNING"** is a record of bona fide work, carried out by **Gangajyoti Dhara** and **Pronoy Roy** under my guidance at **ARDENT COMPUTECH PVT LTD**. In my opinion, the report in its present form is in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** and as per regulations of the **ARDENT®.** To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report.

Guide / Supervisor

_____

Mr. Kamlesh Gupta

Subject Matter Expert & Trainee Software Engineer (Machine Learning)

Ardent Computech Pvt Ltd (An ISO 9001:2015 Certified)

Module 132, SDF Building, Sector 5, Salt Lake, Kolkata - 64

ACKNOWLEDGEMENT

Success of any project depends largely on the encouragement and guidelines of many others. We take this sincere opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project work.

We would like to show our greatest appreciation to Mr. Kamlesh Gupta, Project Manager at Ardent, Kolkata. We always feel motivated and encouraged every time by his valuable advice and constant inspiration; without his encouragement and guidance this project would not have materialized.

Words are inadequate in offering our thanks to the other trainees, project assistants and other members at Ardent Computech Pvt. Ltd. for their encouragement and cooperation in carrying out this project work. The guidance and support received from all the members and who are contributing to this project, was vital for the success of this project.

SL NO.

TABLE OF CONTENTS

WEATHER PREDICTION USING MACHINE LEARNING WITH PYTHON

Introduction:-

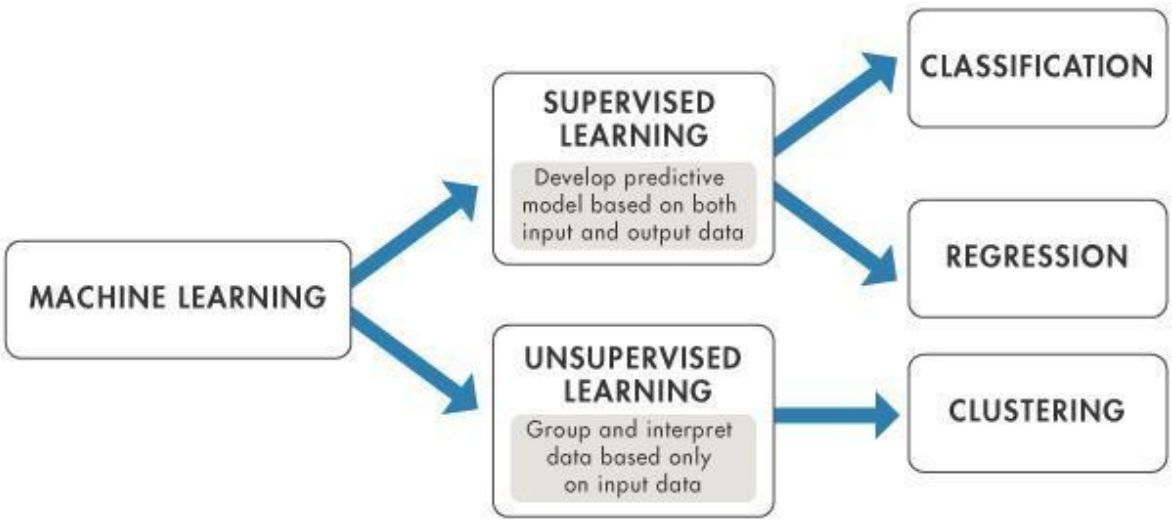
In the 1950s, there was a revolution in weather forecasting. Advances in technology made it possible to simulate the atmosphere using dynamical models, quickly and accurately enough to be used for operational forecasts. Dynamical models are now a central part of weather forecasting. Starting from basic physical laws, they make it possible to predict events such as storms before they have even begun to form.

A crucial challenge in the coming decade will be the integration of direct physical simulations on the one hand, and data-driven approaches on the other. Such a hybrid approach holds many opportunities for weather forecasting, as well as countless other fields.

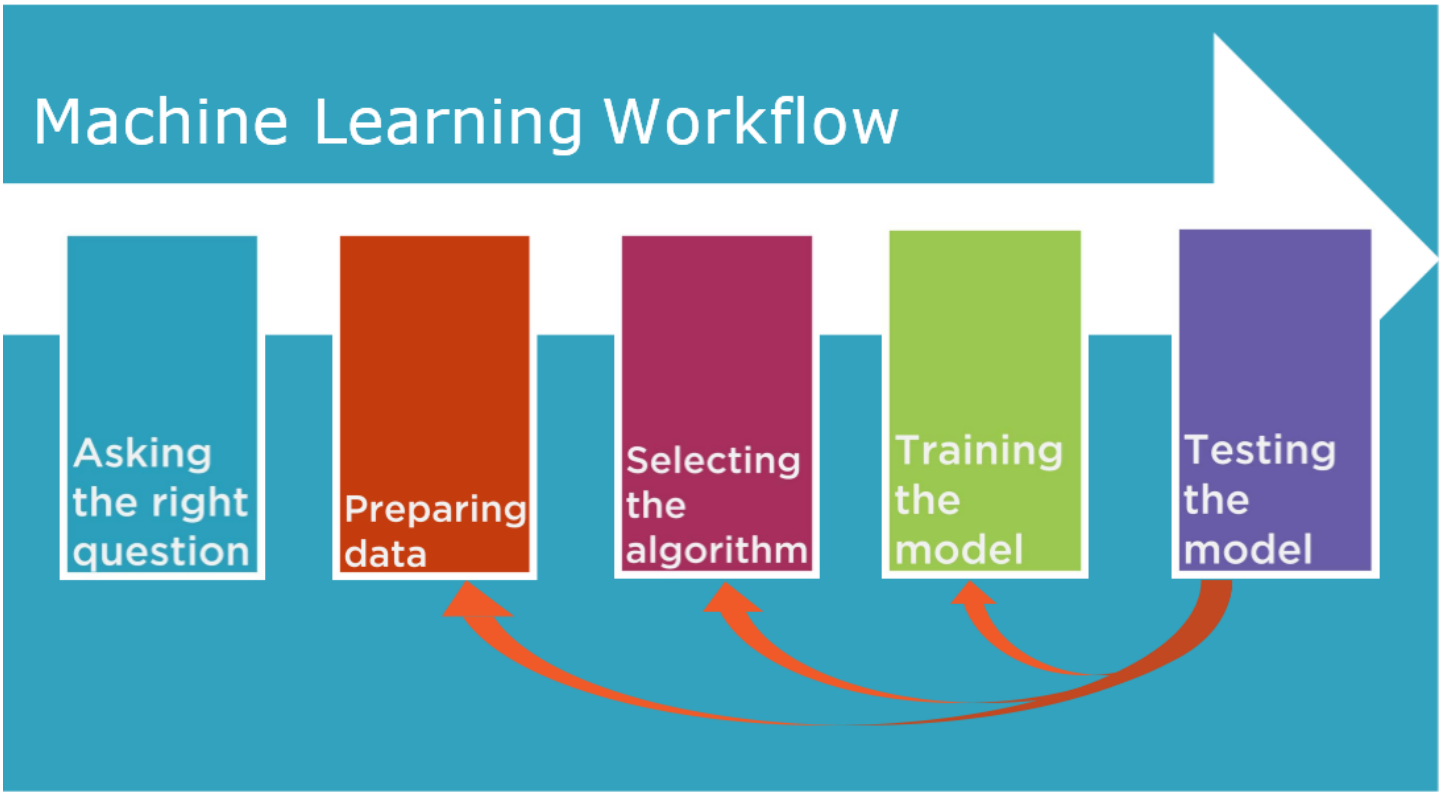
Machine learning, on the contrary, is relatively robust to perturbations and doesn’t require a complete understanding of the physical processes that govern the atmosphere. Therefore, machine learning may represent a viable alternative to physical models in weather forecasting.

What is Machine Learning?

Building a model from example inputs to make data-driven predictions vs. following strictly static program instructions.



Machine Learning Workflow



Project Description

- A cleaned dataset consisting of 40 columns and 997 rows which is preprocessed dataset, provided to us by our instructor for the weather prediction model.
- In order to approach we choose machine learning to predict the target variable for an efficient prediction of weather.
- We are using Jupyter notebook as our interactive working environment for its preinstalled packages and for its ease of use.
- Project is implemented using Python as it has large availability of scientific and statistical packages and it is easy to code and implement.

Problem Statement

Predicting the future temperature based off the past three days of weather measurements using machine learning.

Selecting the Algorithm:

Linear Regression Model Representation

Linear regression is an attractive model because the representation is simple.
The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

The linear equation assigns one scale factor to each input value or column, called a coefficient and represented by the capital Greek letter Beta (B). One additional coefficient is also added, giving the line an additional degree of freedom (e.g. moving up and down on a two-dimensional plot) and is often called the intercept or the bias coefficient.

For example, in a simple regression problem (a single x and a single y), the form of the model would be:

$$y = B0 + B1*x$$

In higher dimensions when we have more than one input (x), the line is called a plane or a hyper-plane. The representation therefore is the form of the equation and the specific values used for the coefficients (e.g. B0 and B1 in the above example).

It is common to talk about the complexity of a regression model like linear regression. This refers to the number of coefficients used in the model.

When a coefficient becomes zero, it effectively removes the influence of the input variable on the model and therefore from the prediction made from the model (0 * x = 0). This becomes relevant if you look at regularization methods that change the learning algorithm to reduce the complexity of regression models by putting pressure on the absolute size of the coefficients, driving some to zero.

Packages And Library Used:-

NumPy Package:-

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

a powerful N-dimensional array object

sophisticated (broadcasting) functions tools for

integrating C/C++ and Fortran code

useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of

databases. NumPy is licensed under the BSD license, enabling reuse with few restrictions.

Array Function:-

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the *rank* of the array; the *shape* of an array is a tuple of integers giving the size of the array along each dimension.

We can initialize numpy arrays from nested Python lists, and access elements using square brackets:

```
import numpy as np

a = np.array([1, 2, 3])      # Create a rank 1 array

print(type(a))           # Prints "<class 'numpy.ndarray'>"
```

Reshape Function:-

It is common to need to reshape a one-dimensional array into a two-dimensional array with one column and multiple arrays. NumPy provides the reshape() function on the NumPy array object that can be used to reshape the data. The reshape () function takes a single argument that specifies the new shape of the array.

Pandas:-

- Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name
- Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

- In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

- Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

- Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features Of Pandas:-

- Fast and efficient DataFrame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and sub-setting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.

Data Frame:-

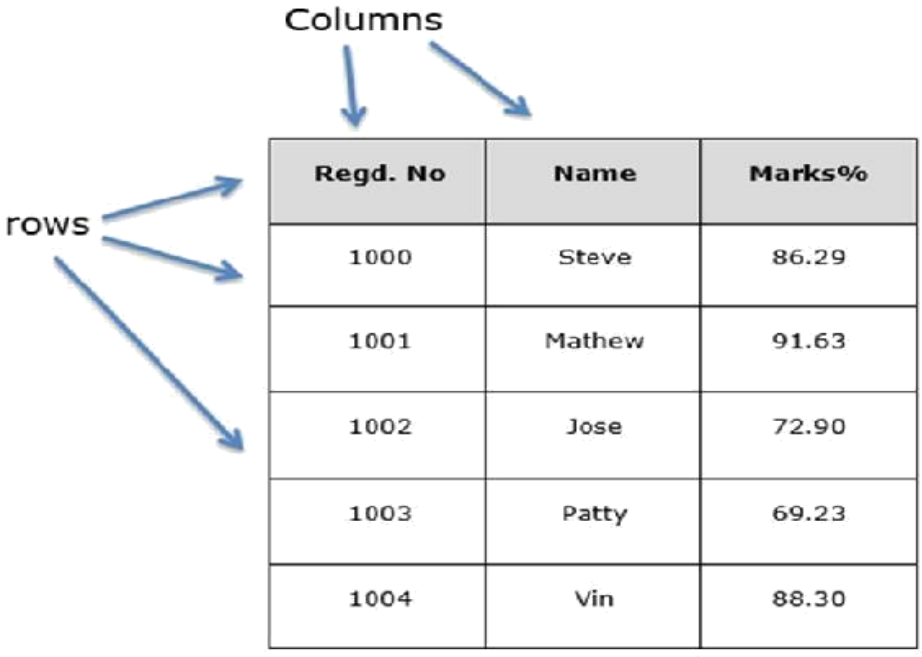
- A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.

Features of Dataframe:-

- Potentially columns are of different types
- Size – Mutable
- Labeled axes (rows and columns)
- Can Perform Arithmetic operations on rows and columns

Structure:-

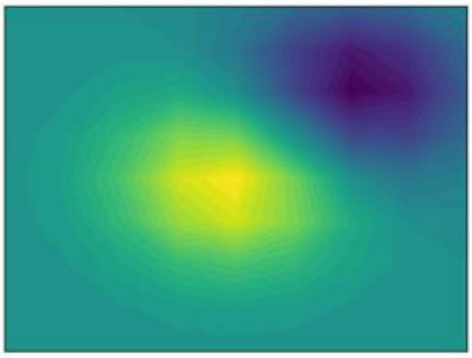
Let us assume that we are creating a data frame with student's data.

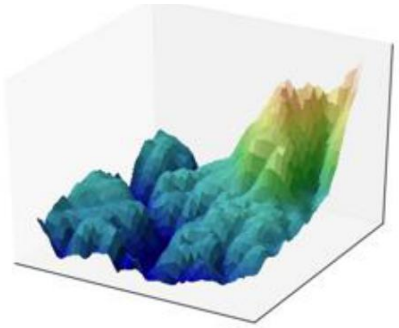


Regd. No	Name	Marks%
1000	Steve	86.29
1001	Mathew	91.63
1002	Jose	72.90
1003	Patty	69.23
1004	Vin	88.30

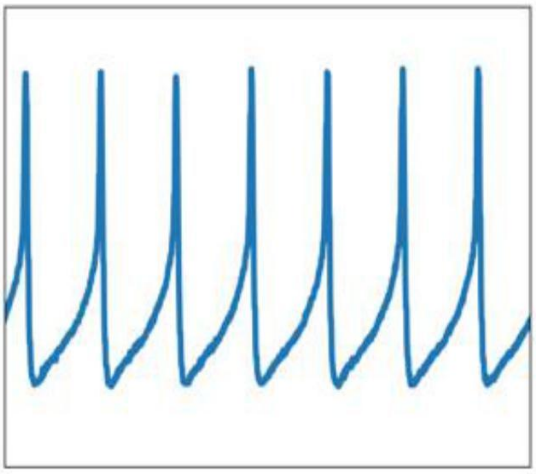
A pandas DataFrame can be created using the following constructor −
pandas.DataFrame(data, index, columns, dtype, copy)

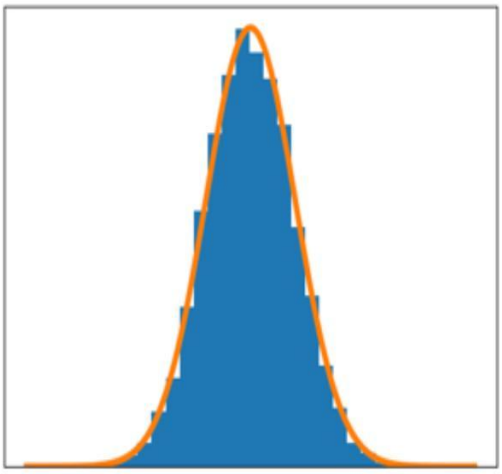
Matplotlib And Pyplot:-





Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.





Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc., via an object oriented interface or via a set of functions familiar to MATLAB users.

Show:-

The show function will display the plotted graph.

Sklearn:-

Scikit-learn (formerly **scikits.learn**) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

The scikit-learn project started as scikits.learn, a Google Summer of Code project by David Cournapeau. Its name stems from the notion that it is a "SciKit" (SciPy Toolkit), a separately-developed and distributed third-party

extension to SciPy.[4] The original codebase was later rewritten by other developers. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel, all from INRIA took leadership of the project and made the first public release on February the 1st 2010.[5] Of the various scikits, scikit-learn as well as scikit-image were described as "well-maintained and popular" in November 2012.

Code Description:

Provided dataset is assigned to a object (dataset) using the library pandas

In [15]:
```
import pandas as pd      #importing the pandas module  as pd

dataset=pd.read_csv('weather-prediction-dataset.csv')
```

[DESCR]:the CSV file is read from the file and copied to dataset object

In [16]:
```
type(dataset)  #print type of your dataset
```

Out[16]:
```
pandas.core.frame.DataFrame
```

In [17]:
```
dataset.head(10)
```

Out[17]:

	date	meantempm	maxtempm	mintempm	meantempm_1	meantempm_2	meantempm_3	mear
0	2015-01-04	-14	-12	-18	-4.0	-6.0	-6.0	
1	2015-01-05	-9	-3	-14	-14.0	-4.0	-6.0	
2	2015-01-06	-10	-6	-14	-9.0	-14.0	-4.0	
3	2015-01-07	-16	-12	-19	-10.0	-9.0	-14.0	
4	2015-01-08	-7	2	-16	-16.0	-10.0	-9.0	
5	2015-01-09	-11	-7	-16	-7.0	-16.0	-10.0	
6	2015-01-10	-6	6	-17	-11.0	-7.0	-16.0	
7	2015-01-11	-5	3	-13	-6.0	-11.0	-7.0	
8	2015-01-12	-13	-7	-19	-5.0	-6.0	-11.0	
9	2015-01-13	-12	-4	-20	-13.0	-5.0	-6.0	

10 rows × 40 columns

As the provided data set is preprocessed . so we simply set our target feature

In [18]:
```
target=pd.DataFrame(dataset['meantempm'])
print(target)
```

```
     meantempm
0          -14
1           -9
2          -10
3          -16
4           -7
5          -11
6           -6
7           -5
8          -13
9          -12
10          -2
11           1
12           4
13           7
14           4
15           5
16           4
17           2
18          -1
19           1
20           6
21           3
22           7
23           4
24           9
25          -1
26          -3
27           2
28          -8
29         -14
..         ...
967         19
968         20
969         22
970         22
971         24
972         24
973         22
974         16
975         14
976         18
977         21
978         22
979         24
980         23
981         22
982         23
983         27
984         27
985         19
986         16
987         21
988         26
```

```
989         22
990         25
991         30
992         28
993         24
994         16
995         15
996         16

[997 rows x 1 columns]
```

[DESCR]:the 'meantempm' i.e our target column is copied to 'target' object

Find the correlation of the features

In [19]:
```
corr=dataset.corr()[['meantempm']]#saving the correlated values
corr.sort_values(by='meantempm',ascending=False)
                    #sorting the correlated values by target features:
```

Out[19]:

	meantempm
meantempm	1.000000
maxtempm	0.976328
mintempm	0.973122
meantempm_1	0.937563
maxtempm_1	0.923787
mintempm_1	0.905423
mindewptm_1	0.899000
meandewptm_1	0.896681
maxdewptm_1	0.887235
meantempm_2	0.881221
maxtempm_2	0.863906
meantempm_3	0.855662
mintempm_2	0.854320
mindewptm_2	0.852760
meandewptm_2	0.848907
maxdewptm_2	0.839893
mintempm_3	0.836340
meandewptm_3	0.834251
mindewptm_3	0.833546
maxtempm_3	0.832974
maxdewptm_3	0.829230
maxhumidity_3	0.167035
maxhumidity_2	0.151358
maxhumidity_1	0.132466
precipm_3	0.098684
precipm_1	0.086617
precipm_2	0.084394
minpressurem_3	-0.102955
minpressurem_2	-0.104455
minhumidity_3	-0.118564
minhumidity_2	-0.143211
minhumidity_1	-0.148602
minpressurem_1	-0.201003

	meantempm
meanpressurem_3	-0.263008
meanpressurem_2	-0.269896
meanpressurem_1	-0.365682
maxpressurem_3	-0.408902
maxpressurem_2	-0.425666
maxpressurem_1	-0.519699

Preparing the features to proceed further:

basic idea: take the features having high correlated values(in the range 0.8-1 as a strong correlated values) as our features set:

In [20]:
```
feat_list=['maxtempm','mintempm','meantempm_1','maxtempm_1','mintempm_1','mindewp
'maxtempm_2',
'meantempm_3',
'mintempm_2',
'mindewptm_2',
'meandewptm_2',
'maxdewptm_2',
'mintempm_3',
'meandewptm_3',
'mindewptm_3',
'maxtempm_3',
'maxdewptm_3']
feat_selected=pd.DataFrame()
```

In [21]:
```
for i in feat_list:
    feat_selected[i]=dataset[i]
```

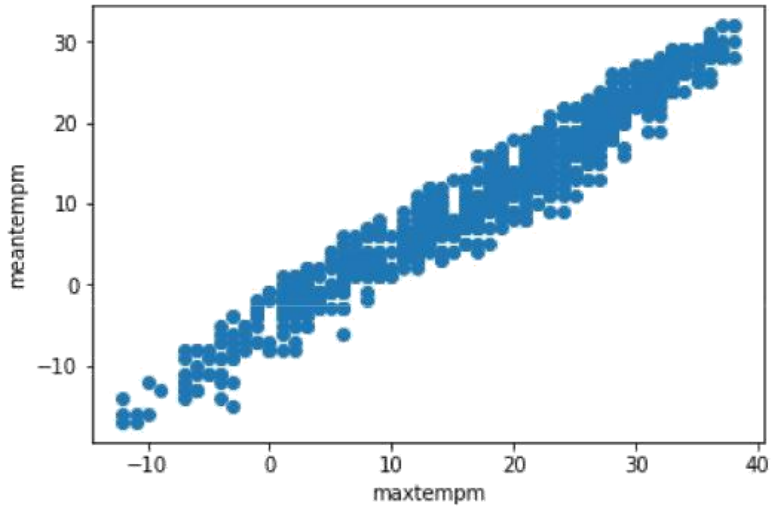
[DESCR]:loop storing the selected features from feat_list in feat_selected

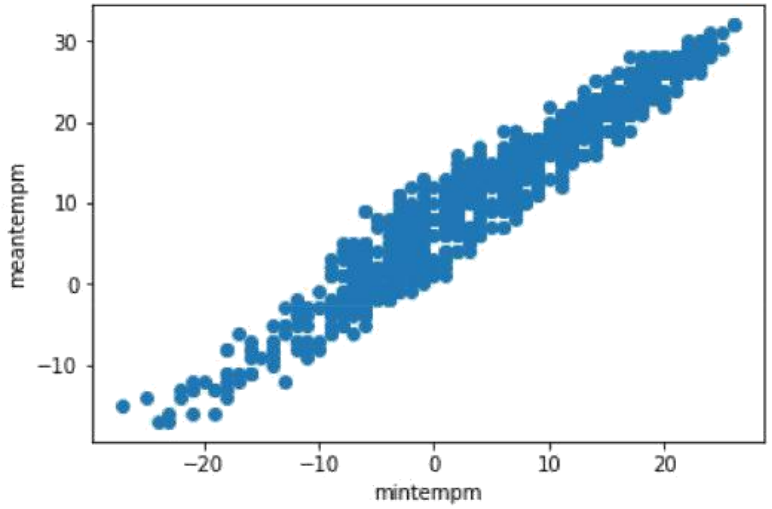
In [22]:
```
print(feat_selected.head(10))
```

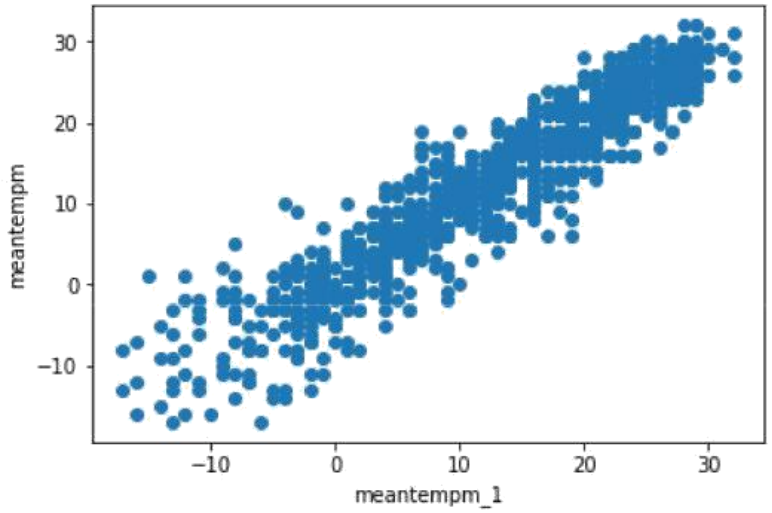
```
   maxtempm  mintempm  meantempm_1  maxtempm_1  mintempm_1  mindewptm_1  \
0       -12       -18         -4.0         3.0       -13.0        -16.0
1        -3       -14        -14.0       -12.0       -18.0        -23.0
2        -6       -14         -9.0        -3.0       -14.0        -17.0
3       -12       -19        -10.0        -6.0       -14.0        -18.0
4         2       -16        -16.0       -12.0       -19.0        -24.0
5        -7       -16         -7.0         2.0       -16.0        -21.0
6         6       -17        -11.0        -7.0       -16.0        -21.0
7         3       -13         -6.0         6.0       -17.0        -19.0
8        -7       -19         -5.0         3.0       -13.0        -14.0
9        -4       -20        -13.0        -7.0       -19.0        -21.0

   meandewptm_1  maxdewptm_1  meantempm_2  maxtempm_2  meantempm_3  \
0         -11.0         -4.0         -6.0         1.0         -6.0
1         -19.0        -16.0         -4.0         3.0         -6.0
2         -14.0         -8.0        -14.0       -12.0         -4.0
3         -15.0        -12.0         -9.0        -3.0        -14.0
4         -22.0        -18.0        -10.0        -6.0         -9.0
5         -12.0         -4.0        -16.0       -12.0        -10.0
6         -19.0        -13.0         -7.0         2.0        -16.0
7         -12.0         -6.0        -11.0        -7.0         -7.0
8         -11.0         -7.0         -6.0         6.0        -11.0
9         -17.0        -13.0         -5.0         3.0         -6.0

   mintempm_2  mindewptm_2  meandewptm_2  maxdewptm_2  mintempm_3  \
0       -12.0        -13.0          -9.0         -6.0       -13.0
1       -13.0        -16.0         -11.0         -4.0       -12.0
2       -18.0        -23.0         -19.0        -16.0       -13.0
3       -14.0        -17.0         -14.0         -8.0       -18.0
4       -14.0        -18.0         -15.0        -12.0       -14.0
5       -19.0        -24.0         -22.0        -18.0       -14.0
6       -16.0        -21.0         -12.0         -4.0       -19.0
7       -16.0        -21.0         -19.0        -13.0       -16.0
8       -17.0        -19.0         -12.0         -6.0       -16.0
9       -13.0        -14.0         -11.0         -7.0       -17.0

   meandewptm_3  mindewptm_3  maxtempm_3  maxdewptm_3
0         -12.0        -18.0         2.0         -6.0
1          -9.0        -13.0         1.0         -6.0
2         -11.0        -16.0         3.0         -4.0
3         -19.0        -23.0       -12.0        -16.0
4         -14.0        -17.0        -3.0         -8.0
5         -15.0        -18.0        -6.0        -12.0
6         -22.0        -24.0       -12.0        -18.0
7         -12.0        -21.0         2.0         -4.0
8         -19.0        -21.0        -7.0        -13.0
9         -12.0        -19.0         6.0         -6.0
```

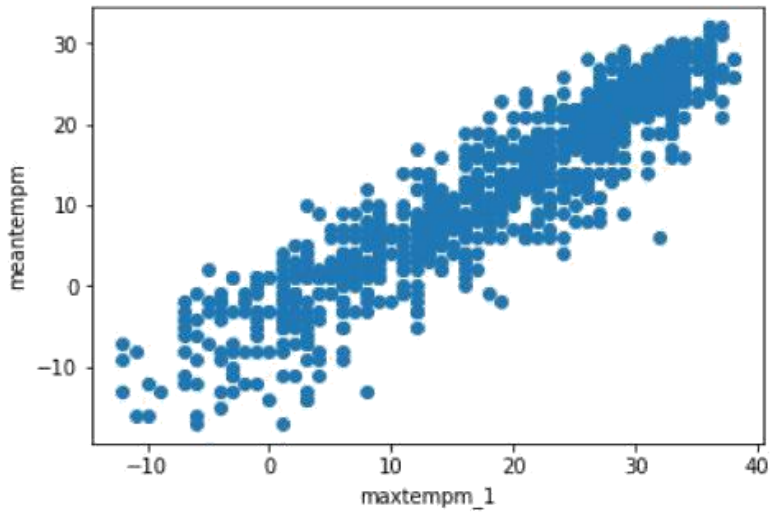
[DESCR]:printing the first 10 rows of the selected features dataframe

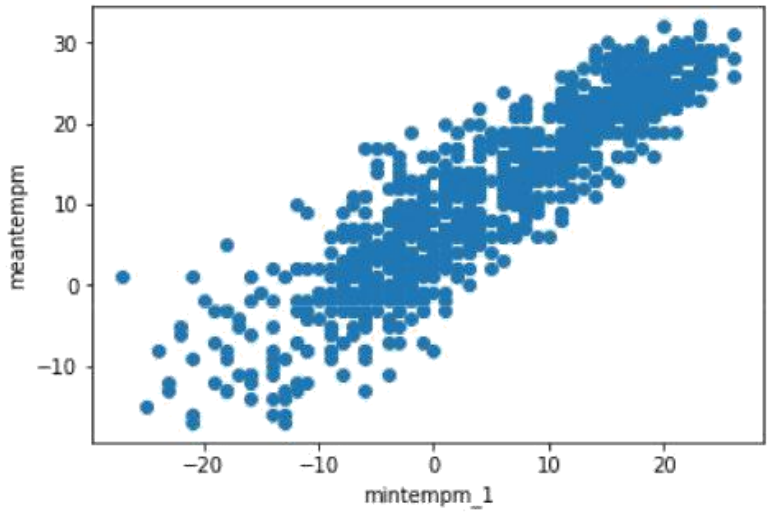
In [23]:
```
for i in feat_selected:
    plt.plot(feat_selected[i],target,'o')
    plt.xlabel(i)
    plt.ylabel('meantempm')
    plt.show()
```

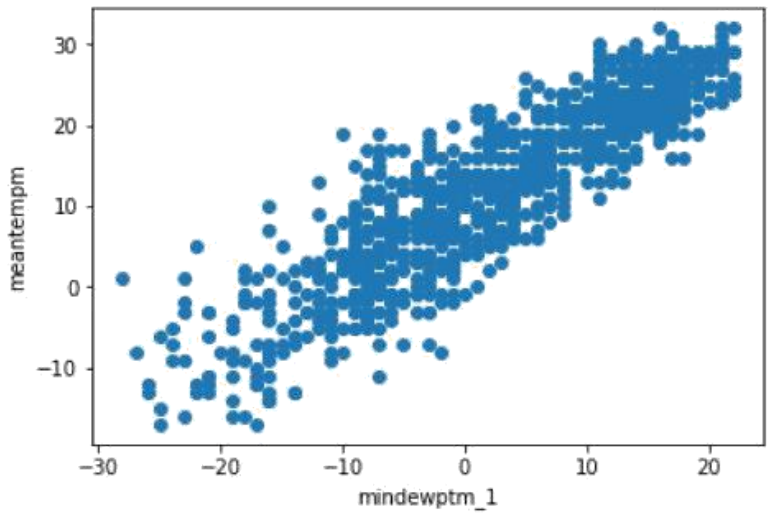


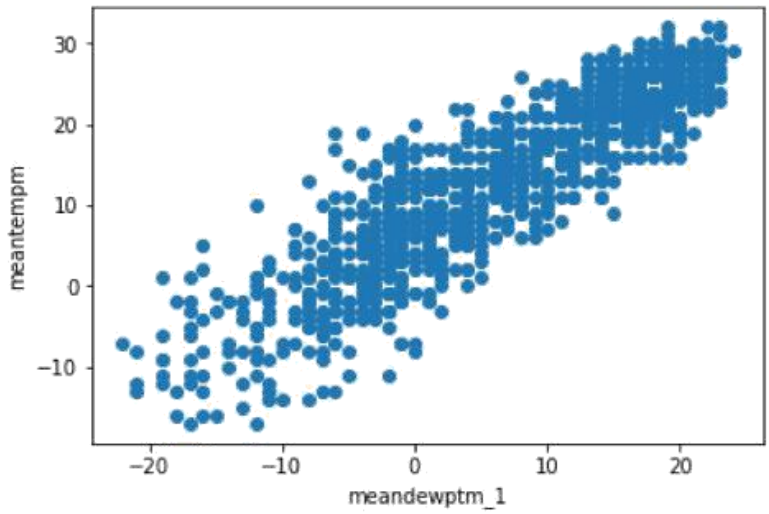


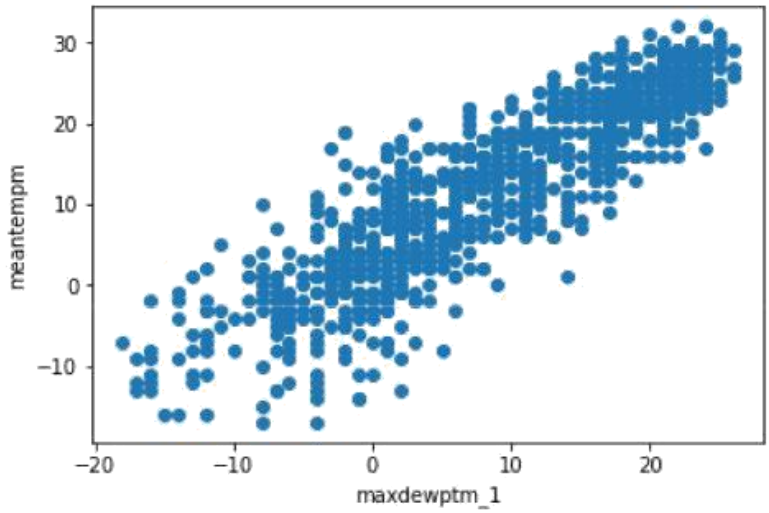


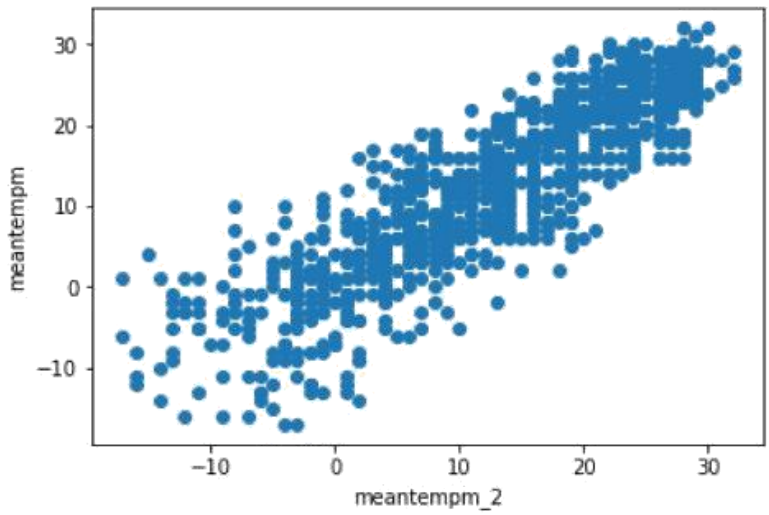

meantempm
maxtempm_1
30
20
10
0
−10
−10
0
10
20
30
40

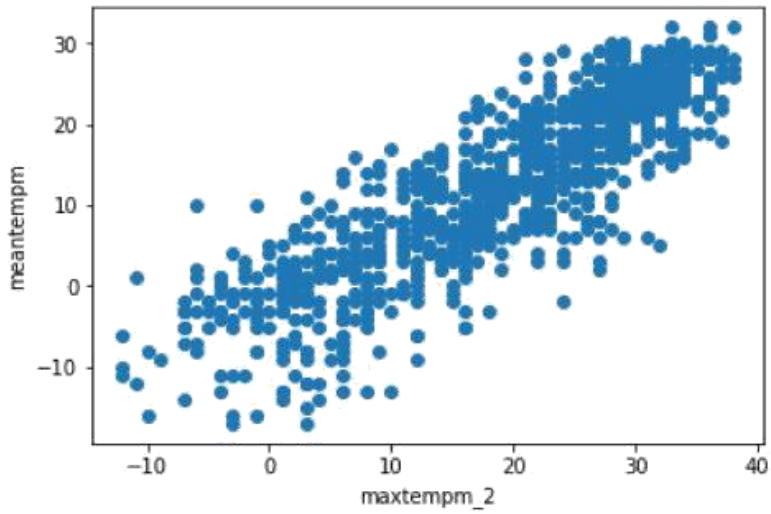

meantempm
mintempm_1
30
20
10
0
−10
−20
−10
0
10
20

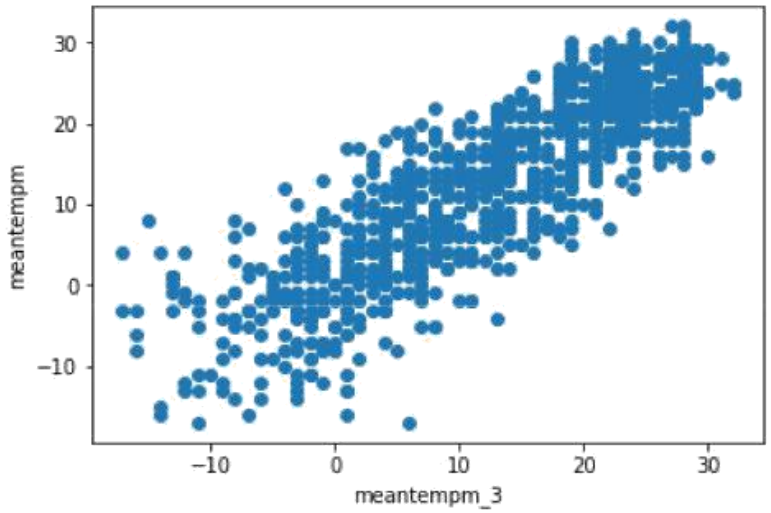

meantempm
mindewptm_1
30
20
10
0
−10
−30
−20
−10
0
10
20

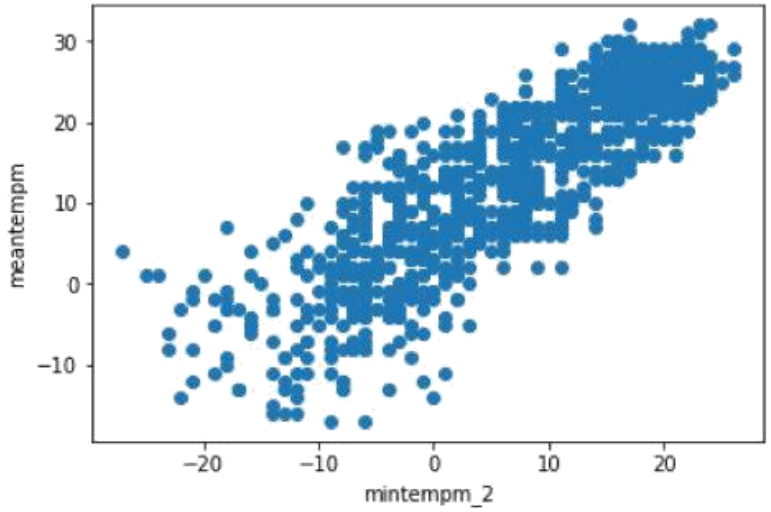

meantempm
meandewptm_1
30
20
10
0
−10
−20
−10
0
10
20

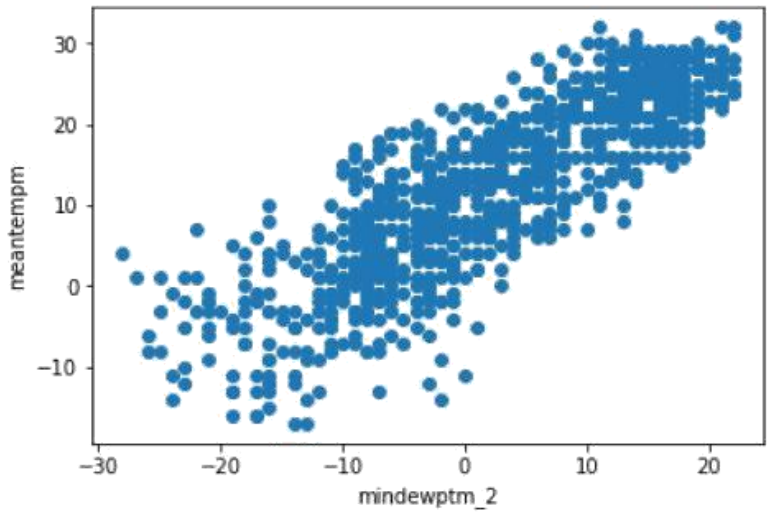

meantempm
maxdewptm_1
30
20
10
0
−10
−20
−10
0
10
20

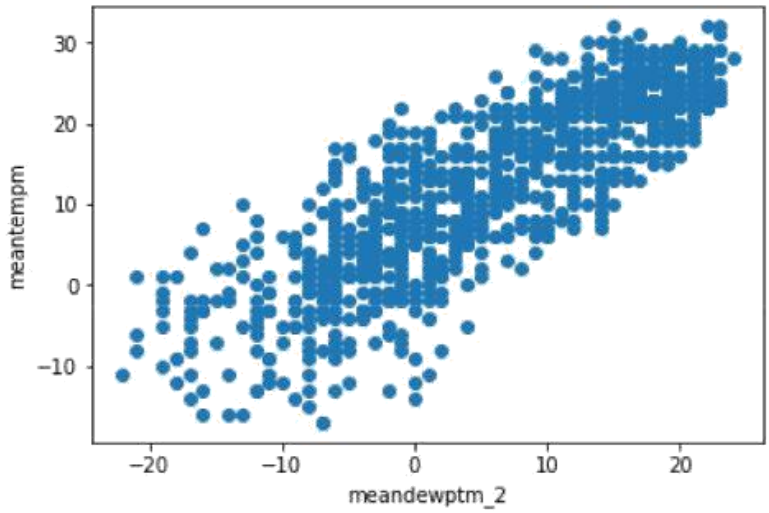

meantempm
meantempm_2
30
20
10
0
−10
−10
0
10
20
30

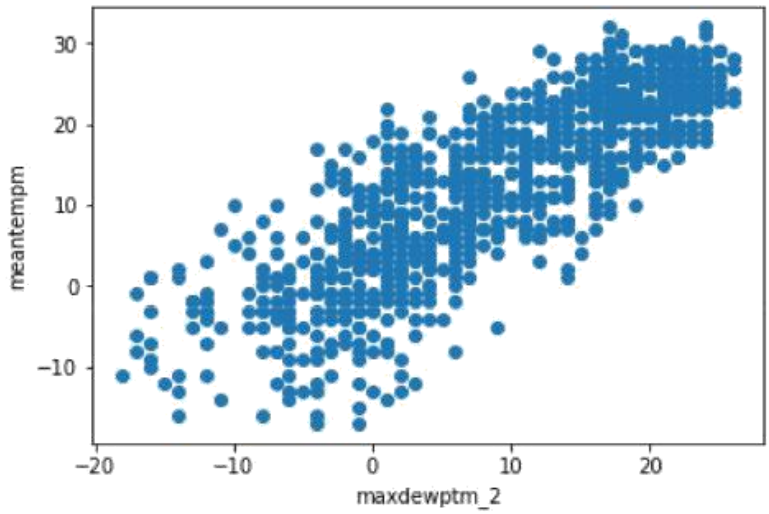

meantempm
maxtempm_2
30
20
10
0
−10
−10
0
10
20
30
40

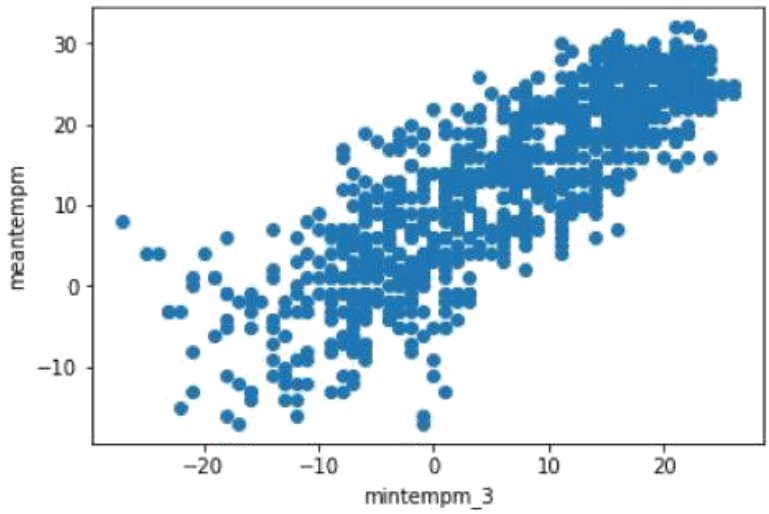

meantempm
meantempm_3
30
20
10
0
−10
−10
0
10
20
30

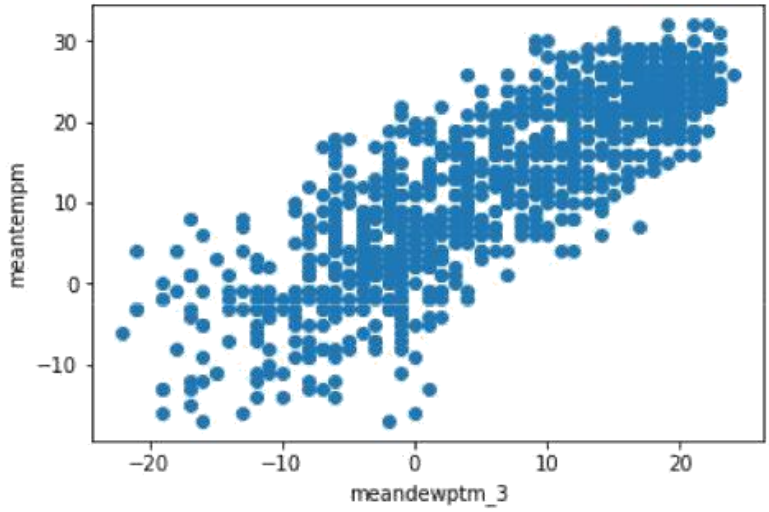

meantempm
mintempm_2
30
20
10
0
−10
−20
−10
0
10
20

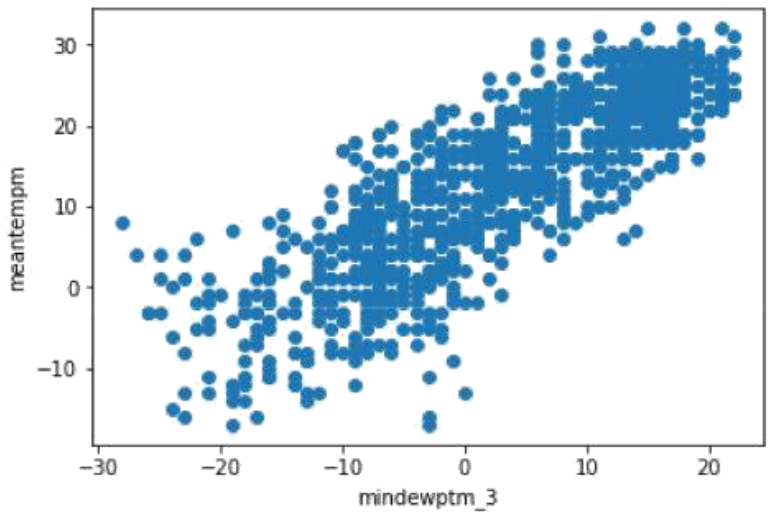

meantempm
mindewptm_2
30
20
10
0
−10
−30
−20
−10
0
10
20

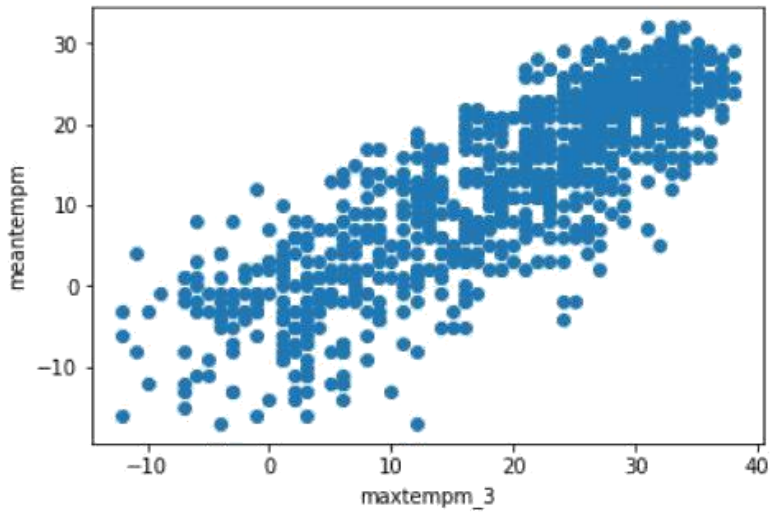

meantempm
meandewptm_2
30
20
10
0
−10
−20
−10
0
10
20

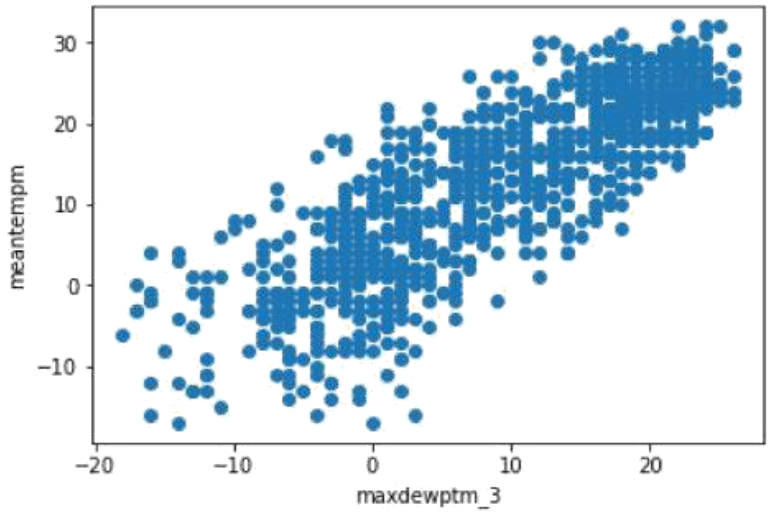

meantempm
maxdewptm_2
30
20
10
0
−10
−20
−10
0
10
20


meantempm
mintempm_3
30
20
10
0
−10
−20
−10
0
10
20


meantempm
meandewptm_3
30
20
10
0
−10
−20
−10
0
10
20


meantempm
mindewptm_3
30
20
10
0
−10
−30
−20
−10
0
10
20





[DESCR]:plotting the features with the target("meantempm")

as our predicted output or target will represent a continuous value. so we select Linear_regression model as our algorithm. and this Linear_regression model is present in scikit_learn library. so we import sklearn first and then import the models linear_model,model_selection,metrics

In [34]:
```
import sklearn
from sklearn import linear_model as ln
from sklearn import model_selection as ms
from sklearn import metrics as m
import matplotlib.pyplot as plt   # importing the matpoltlib to plot & show our g
```

[DESCR]: importing important libraries

In [47]:
```
train_feat,test_feat,train_target,test_target=ms.train_test_split(feat_selected,t
```

[DESCR]: splitting the dataset into training and testing part we take 75% of shuffled data(feature set) from our dataset to train train_set(train_feat and train_target) and take 25 % of shuffled data(target set) from the dataset for test set (test_feat and test_target) taking random_state with a default value

In [26]:
```
len(train_feat)
```

Out[26]: 747

In [27]:
```
len(train_target)
```

Out[27]: 747

In [28]:
```
len(test_feat)
```

Out[28]: 250

In [29]:
```
len(test_target)
```

Out[29]: 250

In [30]:
```
predictor=ln.LinearRegression()
```

[DESCR]:loading the linear regression algorithm

predictor.fit(train_feat,train_target)

[DESCR]:training the data set

In [49]:
```
pred_y=predictor.predict(test_feat)
```

[DESCR]:testing our model by providing test-feat

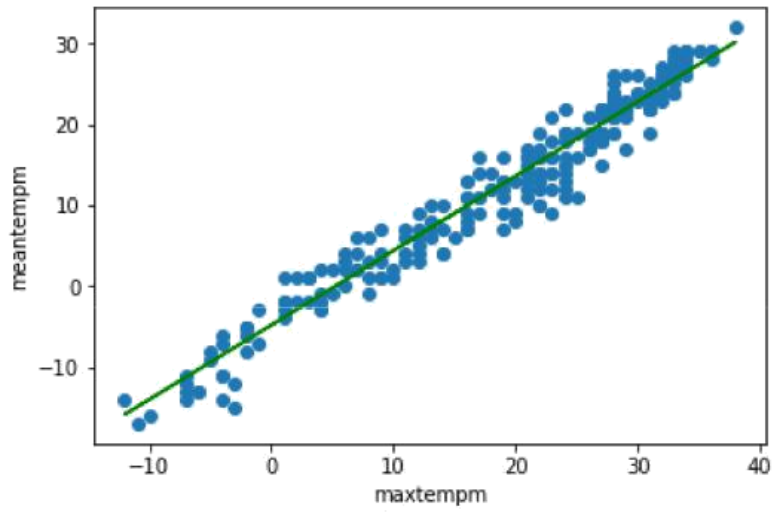
In [37]:
```
import numpy as np
```

[DESCR]: Importing the numpy library as np

In [38]:
```
pred_demo=ln.LinearRegression()
pred_demo.fit(np.array(train_feat['maxtempm']).reshape(747,1),train_target)
pred_y=pred_demo.predict(np.array(test_feat['maxtempm']).reshape(250,1))
```

[DESCR]:To test our model we select a feature 'maxtemp' and then fitting to a new model 'pred_demo' then predict the target.just to visualise the linear graph

In [39]:
```
plt.plot(test_feat['maxtempm'],test_target,'o')
plt.plot(test_feat['maxtempm'],pred_y,'g')
plt.xlabel('maxtempm')
plt.ylabel('meantempm')
plt.show()
```



[DESCR]:plotting the test_feat['maxtempm'] and test_target

In [40]:
```
score=m.r2_score(test_target,pred_y)
print(score)
```

Out[40]:
```
0.9563966908042949
```

[DESCR]:printing the predicted score

In [41]:
```
mse=m.mean_squared_error(test_target,pred_y)
print(mse)
```

Out[41]:
```
5.916253963586376
```

[DESCR]:the mean squared prediction error or mean squared error of the predictions of a smoothing or curve fitting procedure is the expected value of the squared difference between the fitted values implied by the predictive function and the values of the (unobservable) function (that is our target set).

sklearn.metrics.mean_squared_error(y_true, y_pred, sample_weight=None, multioutput='uniform_average')

In [43]:
```
score_list=[]
error_list=[]
```

In [44]:
```
for i in range(10):
    train_feat,test_feat,train_target,test_target=ms.train_test_split(feat_select
    predictor.fit(train_feat,train_target)
    pred_y=predictor.predict(test_feat)
    score=m.r2_score(test_target,pred_y)
    mse=m.mean_squared_error(test_target,pred_y)
    score_list.append(score)
    error_list.append(mse)
```

[DESCR]:train the predictor with train_feat,train_target then predicting the target and calculating the score and mean squared error, and adding those values to the newly created lists score_list and error_list

In [45]:
```
np.array(score_list).mean()
```

Out[45]:
```
0.998643913179891
```

[DESCR]:calculating the mean of the score_list

In [46]:
```
np.array(error_list).mean()
```

Out[46]:
```
0.1607126628036864
```

[DESCR]:calculating the mean of mean_squared_error

WEATHER PREDICTION USING MACHINE LEARNING WITH PYTHON

Conclusion:

We predict the Mean temperature using the selected features from the dataset .

We predict the result using 250 data sets of test using LinearRegression() model.

The amount of error has been calculated by Mean squared error and r^2 score as given in the code description.

References:-

- https://en.wikipedia.org/wiki/Scikit-learn
- https://en.wikipedia.org/wiki/Linear_regression
- https://matplotlib.org/
- http://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html
- http://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html
- http://scikit-learn.org/stable/modules/generated/sklearn.metrics.median_absolute_error.html
- http://scikit-learn.org/stable/modules/generated/sklearn.metrics.median_absolute_error.html