```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

dataset=pd.read_csv('subdataset(1000data1).csv')
#print 1st 20 details
print(dataset.head(20))
```

```
      datetime_utc  _conds  _dewptm  ...   _rain   _snow   _tempm
0   19961101-11:00   Smoke      9.0  ...       0       0     30.0
1   19961101-12:00   Smoke     10.0  ...       0       0     28.0
2   19961101-13:00   Smoke     11.0  ...       0       0     24.0
3   19961101-14:00   Smoke     10.0  ...       0       0     24.0
4   19961101-16:00   Smoke     11.0  ...       0       0     23.0
5   19961101-17:00   Smoke     12.0  ...       0       0     21.0
6   19961101-18:00   Smoke     13.0  ...       0       0     21.0
7   19961101-19:00   Smoke     13.0  ...       0       0     21.0
8   19961101-20:00   Smoke     13.0  ...       0       0     19.0
9   19961101-21:00   Smoke     13.0  ...       0       0     19.0
10  19961101-22:00   Smoke     13.0  ...       0       0     19.0
11  19961101-23:00   Smoke     12.0  ...       0       0     19.0
12  19961102-00:00   Smoke     11.0  ...       0       0     19.0
13  19961102-01:00   Smoke     11.0  ...       0       0     19.0
14  19961102-02:00   Smoke     10.0  ...       0       0     20.0
15  19961102-03:00   Smoke     10.0  ...       0       0     22.0
16  19961102-04:00   Smoke     10.0  ...       0       0     23.0
17  19961102-05:00   Smoke     11.0  ...       0       0     26.0
18  19961102-06:00   Clear     10.0  ...       0       0     28.0
19  19961102-07:00   Clear     10.0  ...       0       0     30.0

[20 rows x 12 columns]
```

```
print(dataset.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 999 entries, 0 to 998
Data columns (total 12 columns):
datetime_utc    999 non-null object
 _conds         999 non-null object
 _dewptm        988 non-null float64
 _fog           999 non-null int64
 _hail          999 non-null int64
 _heatindexm    4 non-null float64
 _hum           988 non-null float64
 _precipm       0 non-null float64
 _pressurem     999 non-null int64
 _rain          999 non-null int64
 _snow          999 non-null int64
 _tempm         988 non-null float64
dtypes: float64(5), int64(5), object(2)
memory usage: 93.8+ KB
None
```

```
print(dataset.describe())
```

```
              _dewptm        _fog   _hail  ...   _rain   _snow      _tempm
count   988.000000  999.000000   999.0  ...   999.0   999.0  988.000000
mean      5.356275    0.018018     0.0  ...     0.0     0.0   17.093117
std       4.143401    0.133083     0.0  ...     0.0     0.0    6.030607
min      -7.000000    0.000000     0.0  ...     0.0     0.0    4.000000
25%       3.000000    0.000000     0.0  ...     0.0     0.0   13.000000
50%       5.000000    0.000000     0.0  ...     0.0     0.0   17.000000
75%       8.000000    0.000000     0.0  ...     0.0     0.0   21.000000
max      14.000000    1.000000     0.0  ...     0.0     0.0   34.000000

[8 rows x 10 columns]
```

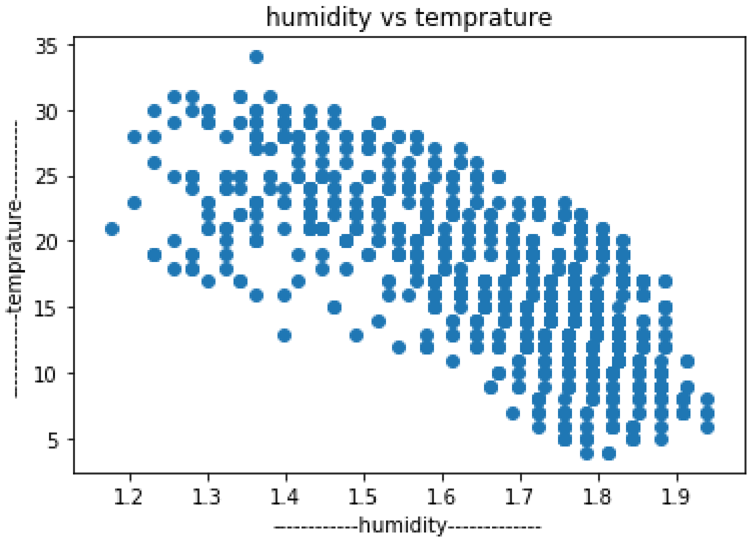
```
#no of rows
print(len(dataset))
#no of colmns
print(len(dataset.columns))
```

```
999
12
```

```
print(dataset.shape)
```

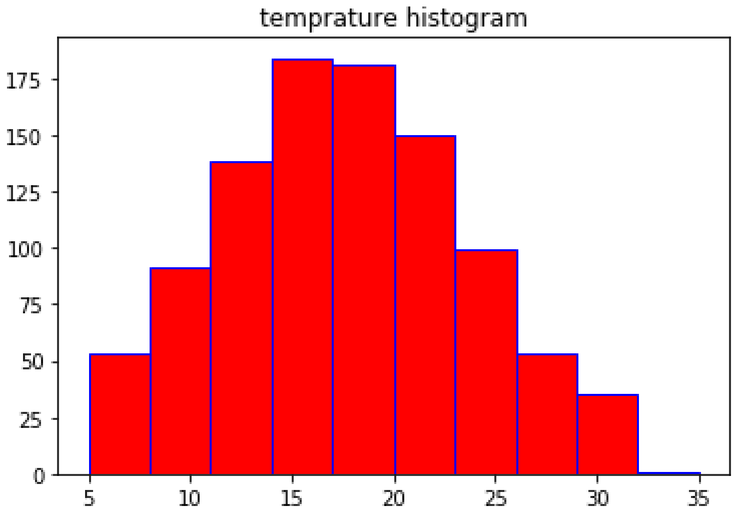
```
(999, 12)
```

```
#Analysis of data how they looks like on graphical representation
#graph himidity vs temparture
plt.scatter(np.log10(dataset[' _hum']),dataset[' _tempm'])
plt.title('humidity vs temprature')
plt.xlabel("------------humidity-------------")
plt.ylabel("------------temprature-----------")
plt.show()
```

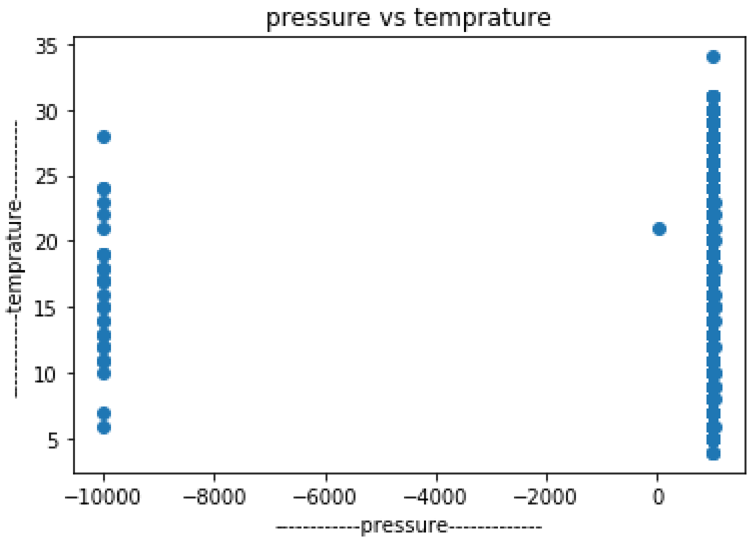


```
#histogram of data how they looks like on graphical represantation
plt.hist(dataset[' _tempm'],facecolor='red',edgecolor='blue',bins=10,range=(5,35))
plt.title("temprature histogram")
plt.show()
```

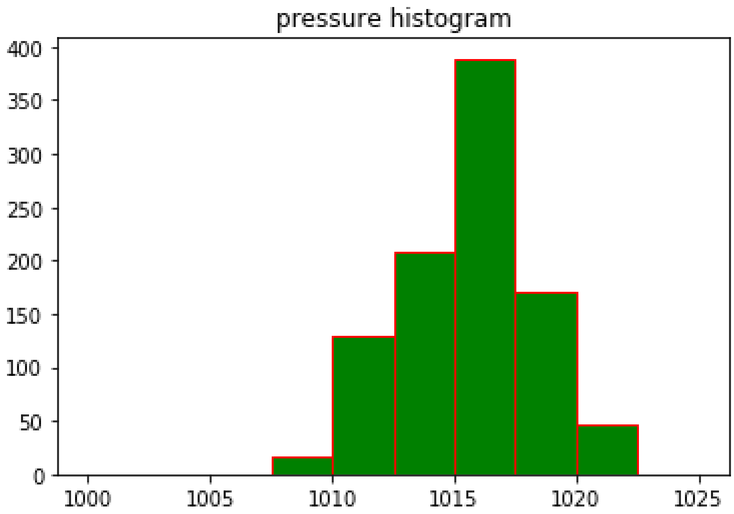
```
/usr/local/lib/python3.6/dist-packages/numpy/lib/histograms.py:829: RuntimeWarning: i
  keep = (tmp_a >= first_edge)
/usr/local/lib/python3.6/dist-packages/numpy/lib/histograms.py:830: RuntimeWarning: i
  keep &= (tmp_a <= last_edge)
```



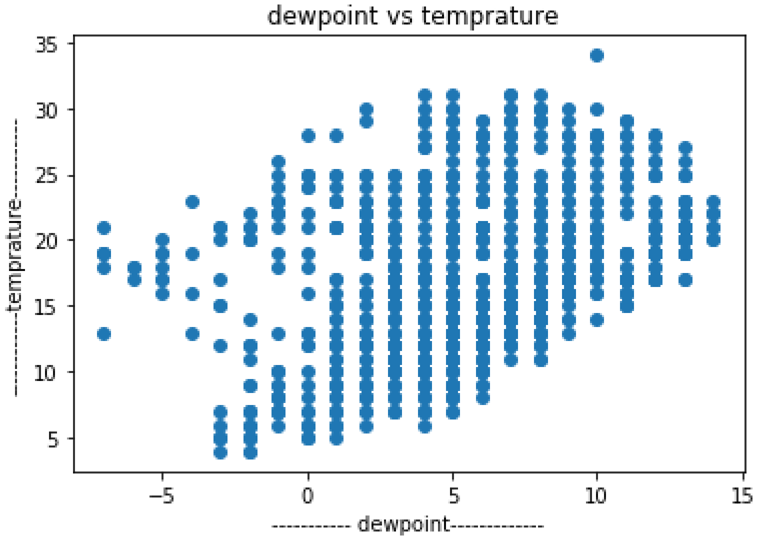
```
plt.scatter(dataset[' _pressurem'],dataset[' _tempm'])
plt.title('pressure vs temprature')
plt.xlabel("------------pressure-------------")
plt.ylabel("------------temprature-----------")
plt.show()
```



```
#histogram of data how they looks like on graphical represantation
plt.hist(dataset[' _pressurem'],facecolor='green',edgecolor='red',bins=10,range=(1000,1025
plt.title("pressure histogram")
plt.show()
```

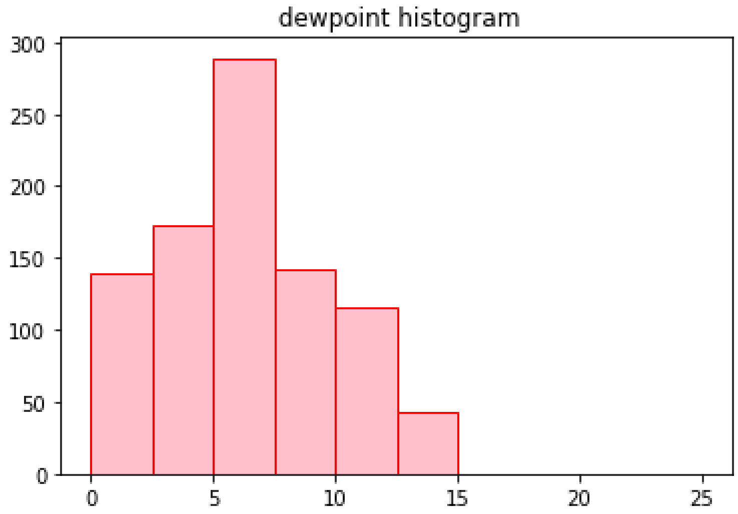


```
#graph dewpoint vs temparture
plt.scatter(dataset[' _dewptm'],dataset[' _tempm'])
plt.title(' dewpoint vs temprature')
plt.xlabel("----------- dewpoint-------------")
plt.ylabel("------------temprature-----------")
plt.show()
```



```
#histogram of data how they looks like on graphical represantation
plt.hist(dataset[' _dewptm'],facecolor='pink',edgecolor='red',bins=10,range=(0,25))
plt.title(" dewpoint histogram")
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/numpy/lib/histograms.py:829: RuntimeWarning: i
  keep = (tmp_a >= first_edge)
/usr/local/lib/python3.6/dist-packages/numpy/lib/histograms.py:830: RuntimeWarning: i
  keep &= (tmp_a <= last_edge)
```



```
#data wrangling
print(dataset.isnull())
```

```
     datetime_utc   _conds   _dewptm  ...   _rain   _snow   _tempm
0           False    False     False  ...   False   False    False
1           False    False     False  ...   False   False    False
2           False    False     False  ...   False   False    False
3           False    False     False  ...   False   False    False
4           False    False     False  ...   False   False    False
..            ...      ...       ...  ...     ...     ...      ...
994         False    False     False  ...   False   False    False
995         False    False     False  ...   False   False    False
996         False    False     False  ...   False   False    False
997         False    False     False  ...   False   False    False
998         False    False     False  ...   False   False    False

[999 rows x 12 columns]
```

```
#table of content in terms true and false
print(dataset.isnull().sum())
```

```
datetime_utc      0
 _conds           0
 _dewptm         11
 _fog             0
 _hail            0
 _heatindexm    995
 _hum            11
 _precipm       999
 _pressurem       0
 _rain            0
 _snow            0
 _tempm          11
dtype: int64
```

```
#droping all unuseful column
dataset.drop([" _heatindexm"],axis=1,inplace=True)
dataset.drop([" _precipm"],axis=1,inplace=True)
#output Delete all null values
print(dataset.isnull().sum())
```

```
datetime_utc     0
 _conds          0
 _dewptm        11
 _fog            0
 _hail           0
 _hum           11
 _pressurem      0
 _rain           0
 _snow           0
 _tempm         11
dtype: int64
```

```
dataset.dropna(inplace=True)
#check is there any null value
print(dataset.head(20))
```

```
          datetime_utc  _conds   _dewptm  ...   _rain   _snow   _tempm
0   19961101-11:00   Smoke       9.0  ...       0       0     30.0
1   19961101-12:00   Smoke      10.0  ...       0       0     28.0
2   19961101-13:00   Smoke      11.0  ...       0       0     24.0
3   19961101-14:00   Smoke      10.0  ...       0       0     24.0
4   19961101-16:00   Smoke      11.0  ...       0       0     23.0
5   19961101-17:00   Smoke      12.0  ...       0       0     21.0
6   19961101-18:00   Smoke      13.0  ...       0       0     21.0
7   19961101-19:00   Smoke      13.0  ...       0       0     21.0
8   19961101-20:00   Smoke      13.0  ...       0       0     19.0
9   19961101-21:00   Smoke      13.0  ...       0       0     19.0
10  19961101-22:00   Smoke      13.0  ...       0       0     19.0
11  19961101-23:00   Smoke      12.0  ...       0       0     19.0
12  19961102-00:00   Smoke      11.0  ...       0       0     19.0
13  19961102-01:00   Smoke      11.0  ...       0       0     19.0
14  19961102-02:00   Smoke      10.0  ...       0       0     20.0
15  19961102-03:00   Smoke      10.0  ...       0       0     22.0
16  19961102-04:00   Smoke      10.0  ...       0       0     23.0
17  19961102-05:00   Smoke      11.0  ...       0       0     26.0
18  19961102-06:00   Clear      10.0  ...       0       0     28.0
19  19961102-07:00   Clear      10.0  ...       0       0     30.0

[20 rows x 10 columns]
```

```
print(dataset.isnull().sum())
```

```
datetime_utc    0
 _conds         0
 _dewptm        0
 _fog           0
 _hail          0
 _hum           0
 _pressurem     0
 _rain          0
 _snow          0
 _tempm         0
dtype: int64
```

```
dataset.drop(["datetime_utc"],axis=1,inplace=True)
#delete all values from the pressure which has a value -9999
indexn=dataset[dataset[' _pressurem']==-9999].index
dataset.drop(indexn,inplace=True)
```

```
#taking all the features into x variable  and y for prediction
Y=dataset.iloc[:,len(dataset.columns)-1]
X=dataset.iloc[:,0:len(dataset.columns)-1]
```

```
print(Y)
print(X)
```

```
0      30.0
3      24.0
4      23.0
5      21.0
6      21.0
       ...
994     9.0
995    15.0
996    18.0
997    19.0
998    19.0
Name:  _tempm, Length: 955, dtype: float64
      _conds   _dewptm   _fog   _hail   _hum   _pressurem   _rain   _snow
0     Smoke       9.0      0       0   27.0         1010       0       0
3     Smoke      10.0      0       0   41.0         1010       0       0
4     Smoke      11.0      0       0   47.0         1011       0       0
5     Smoke      12.0      0       0   56.0         1011       0       0
6     Smoke      13.0      0       0   60.0         1010       0       0
..      ...       ...    ...     ...    ...          ...     ...     ...
994   Smoke       3.0      0       0   66.0         1021       0       0
995   Smoke       5.0      0       0   51.0         1021       0       0
996   Smoke       5.0      0       0   42.0         1021       0       0
997   Smoke       4.0      0       0   37.0         1019       0       0
998   Smoke       4.0      0       0   37.0         1018       0       0

[955 rows x 8 columns]
```

```
#set the dummies value as a level for the weather clacification
weather_condition=pd.get_dummies(X[' _conds'])
```

```
print(weather_condition)
```

```
     Clear  Haze  Mostly Cloudy  Scattered Clouds  Shallow Fog  Smoke  Unknown
0        0     0              0                 0            0      1        0
3        0     0              0                 0            0      1        0
4        0     0              0                 0            0      1        0
5        0     0              0                 0            0      1        0
6        0     0              0                 0            0      1        0
..     ...   ...            ...               ...          ...    ...      ...
994      0     0              0                 0            0      1        0
995      0     0              0                 0            0      1        0
996      0     0              0                 0            0      1        0
997      0     0              0                 0            0      1        0
998      0     0              0                 0            0      1        0

[955 rows x 7 columns]
```

```
#delete last dummies value which is null
weather_condition.drop(["Unknown"],axis=1,inplace=True)
print(weather_condition.head(10))
```

```
    Clear  Haze  Mostly Cloudy  Scattered Clouds  Shallow Fog  Smoke
0       0     0              0                 0            0      1
3       0     0              0                 0            0      1
4       0     0              0                 0            0      1
5       0     0              0                 0            0      1
6       0     0              0                 0            0      1
9       0     0              0                 0            0      1
10      0     0              0                 0            0      1
11      0     0              0                 0            0      1
12      0     0              0                 0            0      1
13      0     0              0                 0            0      1
```

```
#concat the dummies value with the input feature X
X=pd.concat([X,weather_condition],axis=1)
```

```
print(X.head(10))
```

```
    _conds   _dewptm   _fog  ...  Scattered Clouds  Shallow Fog  Smoke
0    Smoke       9.0      0  ...                 0            0      1
3    Smoke      10.0      0  ...                 0            0      1
4    Smoke      11.0      0  ...                 0            0      1
5    Smoke      12.0      0  ...                 0            0      1
6    Smoke      13.0      0  ...                 0            0      1
9    Smoke      13.0      0  ...                 0            0      1
10   Smoke      13.0      0  ...                 0            0      1
11   Smoke      12.0      0  ...                 0            0      1
12   Smoke      11.0      0  ...                 0            0      1
13   Smoke      11.0      0  ...                 0            0      1

[10 rows x 14 columns]
```

```


X.drop([" _conds"],axis=1,inplace=True)
print(X.shape)
#now final data set has been created
print(X.head(10))
```

```
(955, 13)
      _dewptm  _fog  _hail  ...  Scattered Clouds  Shallow Fog  Smoke
0         9.0     0      0  ...                 0            0      1
3        10.0     0      0  ...                 0            0      1
4        11.0     0      0  ...                 0            0      1
5        12.0     0      0  ...                 0            0      1
6        13.0     0      0  ...                 0            0      1
9        13.0     0      0  ...                 0            0      1
10       13.0     0      0  ...                 0            0      1
11       12.0     0      0  ...                 0            0      1
12       11.0     0      0  ...                 0            0      1
13       11.0     0      0  ...                 0            0      1

[10 rows x 13 columns]
```

```
# train and testing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

#splitting Dataset into train set and test set
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,random_state=0)
model=LinearRegression()
model.fit(X_train,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
print(X_train)
```

```
      _dewptm  _fog  _hail  ...  Scattered Clouds  Shallow Fog  Smoke
432       5.0     0      0  ...                 0            0      1
903       1.0     0      0  ...                 0            0      1
66       10.0     0      0  ...                 0            0      1
83        9.0     0      0  ...                 0            0      1
682       4.0     0      0  ...                 0            0      1
..        ...   ...    ...  ...               ...          ...    ...
872       1.0     0      0  ...                 0            0      1
198       4.0     0      0  ...                 0            0      0
661       7.0     0      0  ...                 0            0      1
590       2.0     0      0  ...                 0            0      1
716       6.0     0      0  ...                 0            0      1

[764 rows x 13 columns]
```

```
print(y_train)
```

```
432    16.0
903     7.0
66     28.0
83     28.0
682    24.0
       ...
872    21.0
198    29.0
661    22.0
590     9.0
716    12.0
Name:  _tempm, Length: 764, dtype: float64
```

```
print(X_test)
```

```
       _dewptm   _fog   _hail  ...  Scattered Clouds  Shallow Fog  Smoke
917       3.0      0       0  ...                 0            0      1
942       1.0      0       0  ...                 0            0      1
18       10.0      0       0  ...                 0            0      0
735       4.0      0       0  ...                 0            0      0
59       13.0      0       0  ...                 0            0      1
..        ...    ...     ...  ...               ...          ...    ...
504       4.0      0       0  ...                 0            0      1
22        7.0      0       0  ...                 0            0      0
865       1.0      0       0  ...                 0            0      1
518       7.0      0       0  ...                 0            0      1
995       5.0      0       0  ...                 0            0      1

[191 rows x 13 columns]
```

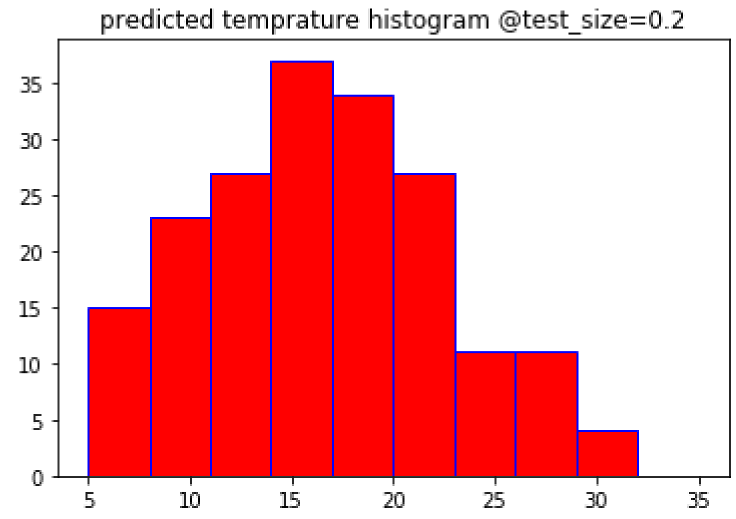
```
print(y_test)
```

```
917    12.0
942    12.0
18     28.0
735     8.0
59     17.0
       ...
504    23.0
22     31.0
865    10.0
518    16.0
995    15.0
Name:  _tempm, Length: 191, dtype: float64
```

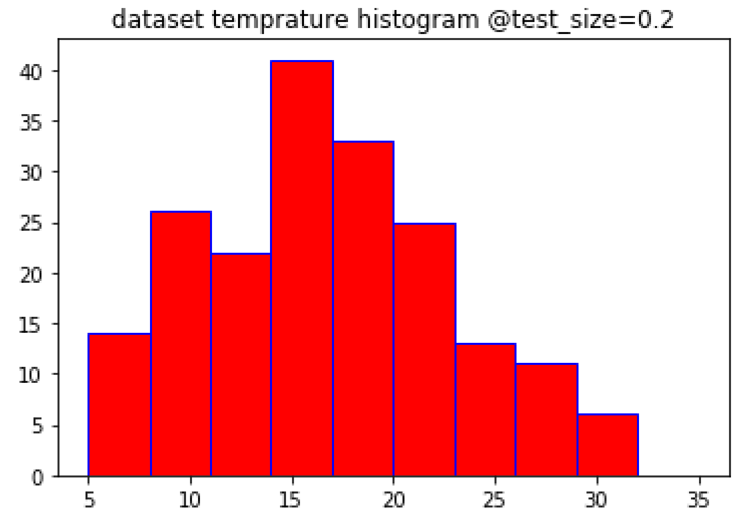
```
y_prediction=model.predict(X_test)
score=r2_score(y_test,y_prediction)
print("Temprature prediction Accuracy @test_size=0.2= ",score*100)
```

```
Temprature prediction Accuracy @test_size=0.2=  97.2146437617622
```

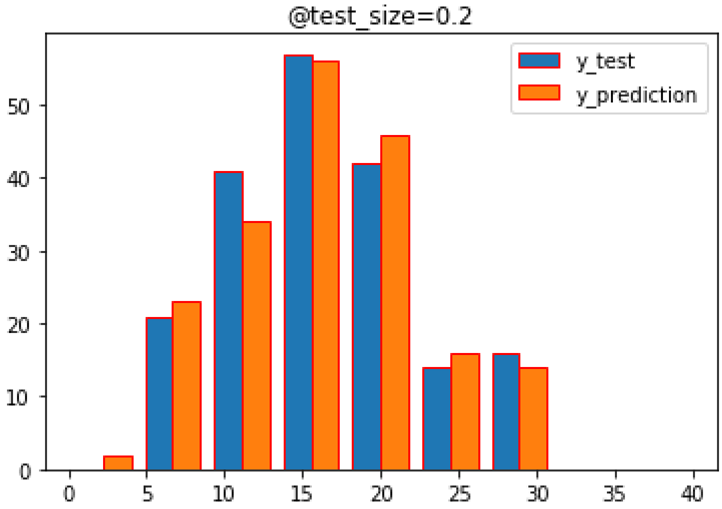
```
#histogram of data how they looks like on graphical represantation
plt.hist(y_prediction,facecolor='red',edgecolor='blue',bins=10,range=(5,35))
plt.title("predicted temprature histogram @test_size=0.2")
plt.show()
```



```
plt.hist(y_test,facecolor='red',edgecolor='blue',bins=10,range=(5,35))
plt.title("dataset temprature histogram @test_size=0.2")
plt.show()
```



```
bins1 = np.linspace(0, 40, 10)
plt.hist([y_test, y_prediction],bins1,edgecolor='red', label=['y_test', 'y_prediction'])
plt.legend(loc='upper right')
plt.title("@test_size=0.2")
plt.show()
```



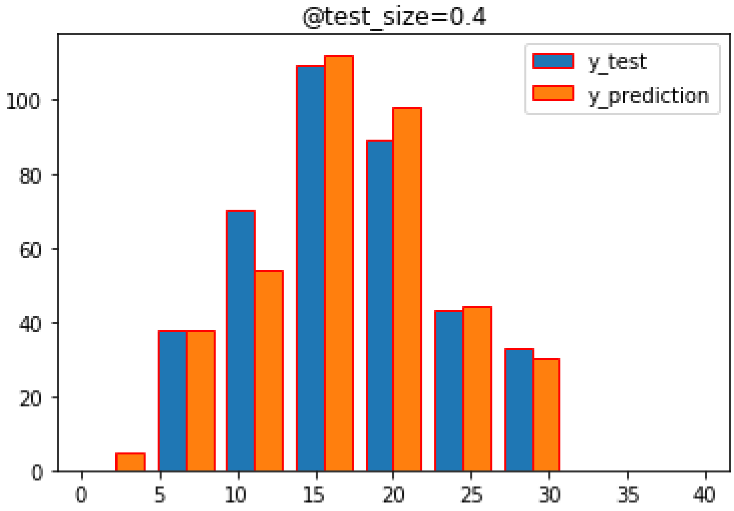
```
#splitting Dataset into train set and test set
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.4,random_state=0)
model=LinearRegression()
model.fit(X_train,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
y_prediction=model.predict(X_test)
score=r2_score(y_test,y_prediction)
print("Temprature prediction Accuracy @test_size=0.4= ",score*100)
```

```
Temprature prediction Accuracy @test_size=0.4=  50.69512388708708
```

```
bins1 = np.linspace(0, 40, 10)
plt.hist([y_test, y_prediction],bins1,edgecolor='red', label=['y_test', 'y_prediction'])
plt.legend(loc='upper right')
plt.title("@test_size=0.4")
plt.show()
```



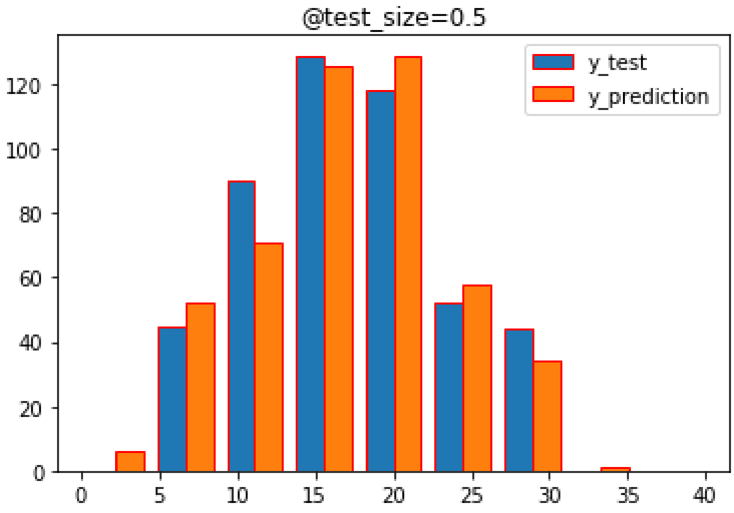
```
#splitting Dataset into train set and test set
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.5,random_state=0)
model=LinearRegression()
model.fit(X_train,y_train)
model.fit(X_train,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
y_prediction=model.predict(X_test)
score=r2_score(y_test,y_prediction)
print("Temprature prediction Accuracy @test_size=0.5= ",score*100)
```

```
Temprature prediction Accuracy @test_size=0.5=  52.592561782808666
```

```
bins1 = np.linspace(0, 40, 10)
plt.hist([y_test, y_prediction],bins1,edgecolor='red', label=['y_test', 'y_prediction'])
plt.legend(loc='upper right')
plt.title("@test_size=0.5")
plt.show()
```


@test_size=0.5
y_test
y_prediction