International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 09 | Sept - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Weather Prediction Using Logistic Regression (AI/ML Techniques)

Sumana Chatterjee, Ph.D. Scholar (Computer Science), Nirwan University, Jaipur, Rajasthan,
MET A, RMC Kolkata, India Meteorological Department
E-Mail Id: sumana.spssaha.chatterjee@gmail.com

ABSTRACT: Prediction of '**Weather or atmospheric condition**' by AI, machine learning techniques is a process of great challenge. Attempts had been made by Computer, Data-Scientists since long, how this condition can be performed successfully .The objective is to predict weather for a place for certain days, here 'ALIPORE (42807)'.We collected 'ALIPORE surface data' (CSV file) for the period, 1969-2023. After collecting this big data, completed process of 'data mining' and necessary 'feature engineering' steps along with choosing responsible dependent or independent parameters called as predictors to find results or outputs by various machine learning packages of Python like 'Pandas', 'SEABORN', 'STATS MODEL' etc. ,under 'SCIKIT LEARN' as well as various ML code and techniques like 'Shape', 'drop null values', 'Describe', 'Label-encoding', 'IV-method', 'VIF method' etc. ,some based on statistical theories . Ultimately equation of 'Logistic Regression' had been built with test-train split formula to predict future weather as 'SIGNIFICANT' or 'CLEAR' for certain test array. During analysis, all the weather phenomena as obtained from this big data set, were classified into two categories. No(1)--- 'Lightning (code 0)', 'Drizzle (Code 5)', 'Rain (Code 6)' and 'Thunderstorm with rain (Code 9)'---for occurrence of any of these weather phenomena ,data were considered as '1' or 'SIGNIFICANT' weather and No (2)---On the other hand , all weather except weather as mentioned above ,No (1),were considered as '0' or 'CLEAR' weather.

Keywords: Confusion-matrix, Heat-map, True-positive, True-negative, False-positive, False-negative, Accuracy-score, Classification-report, Precision, Recall, F1 score.

Introduction: With the advancement of technology and advent of computer, the process of **prediction of weather** or **determination of pattern of probable atmospheric events,** more accurately and efficiently ,with the help of AI (artificial intelligence) and ML (machine learning techniques) based on statistical theories has become a great point of interest .How the big data for some place ,since historical time ,can be analysed with machine learning software to find out probable weather .It is commonly known that weather is very crucial factor having direct impact on society ,including agriculture, transportation, emergency management and all other vital corners of our society and wellness. Different machine learning techniques are there to detect future weather , such as trend of weather pattern by TIME-SERIES etc. In this case we tried to detect probable weather criteria with the help of LOGISTIC REGRESSION METHOD, which falls under supervised learning method. This is based on data analysis with weather data with discrete output , either '1' or '0'. As we have considered , '1' for SIGNIFICANT weather , '0' for CLEAR weather ,from the predicted test data ,we can get array of predicted weather based on previous data and also giving input for previous data weather data, we can get next day weather pattern also.

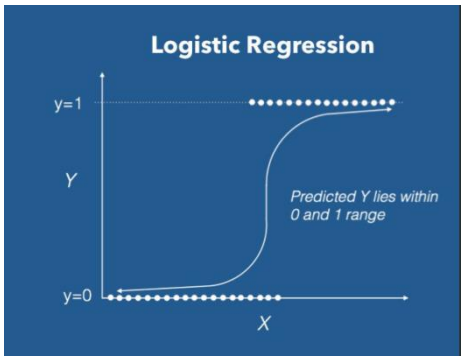
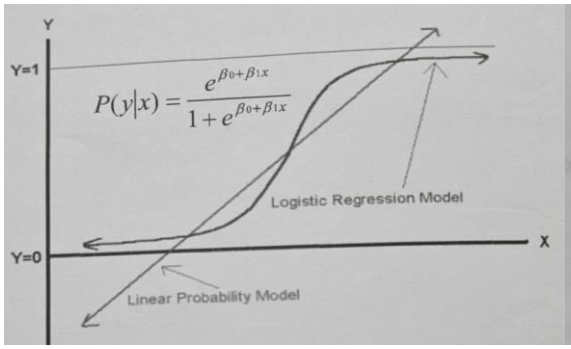


Image of logistic regression

Theory of logistic regression in the back ground of analytical procedure



e) The logit function

For any number π between 0 and 1 the logit function is defined by

$\text{logit}(\pi) = \log(\pi/(1-\pi))$

The logistic regression equation becomes

$logit(\pi) = \alpha + \beta_i X$

Estimation of Parameters

More formally, we define the logistic regression model for binary classification problems. We choose the hypothesis function to be the sigmoid function:

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

Here, theta denotes the parameter vector. For a model containing n features, we have $\theta = [\theta_0, \theta_1, ..., \theta_n]$ containing n + 1 parameters. The hypothesis function approximates the estimated probability of the actual output being equal to 1.

Cost Function

In this case, finding an optimal solution with the gradient descent method is not possible. **Instead, we use a logarithmic function to represent the cost of logistic regression.** It is guaranteed to be convex for all input values, containing only one minimum.

$$cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & , \text{if } y = 1 \\ -log(1 - h_\theta(x)) & , \text{if } y = 0 \end{cases}$$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

Citation : Srinivasan , Dr Babji ,Associate Professor ,Department of Applied Mechanics, IIT Madras,2023,note on logistic regression.

Literature Review:

Study materials ,notes of professors and data-scientists as well as hand on training on **python code** analysis on zupyter or google collaboratory platform ,study material collected from the course '**ADVANCED CERTIFICATION IN DATA SCIENCE AND AI ,CODE IIT MADRAS ,Digital skill academy's programme',** organised by INTELLIPAAT ,was the biggest source to gain this concept of data analysis ,machine learning and determination of output related to this work. Other than these, the sites of data analysis company, Kaggle , 'Analytics Vidya', 'Towards data science', 'geeks for geeks' 'W3 school' ,Medium etc. ,those open sources in google Search Engine were very much helpful to get the suitable codes and to execute the programme. Also for one python code , had taken a little help of AI chatbot, 'gemini', available in google collab platform.

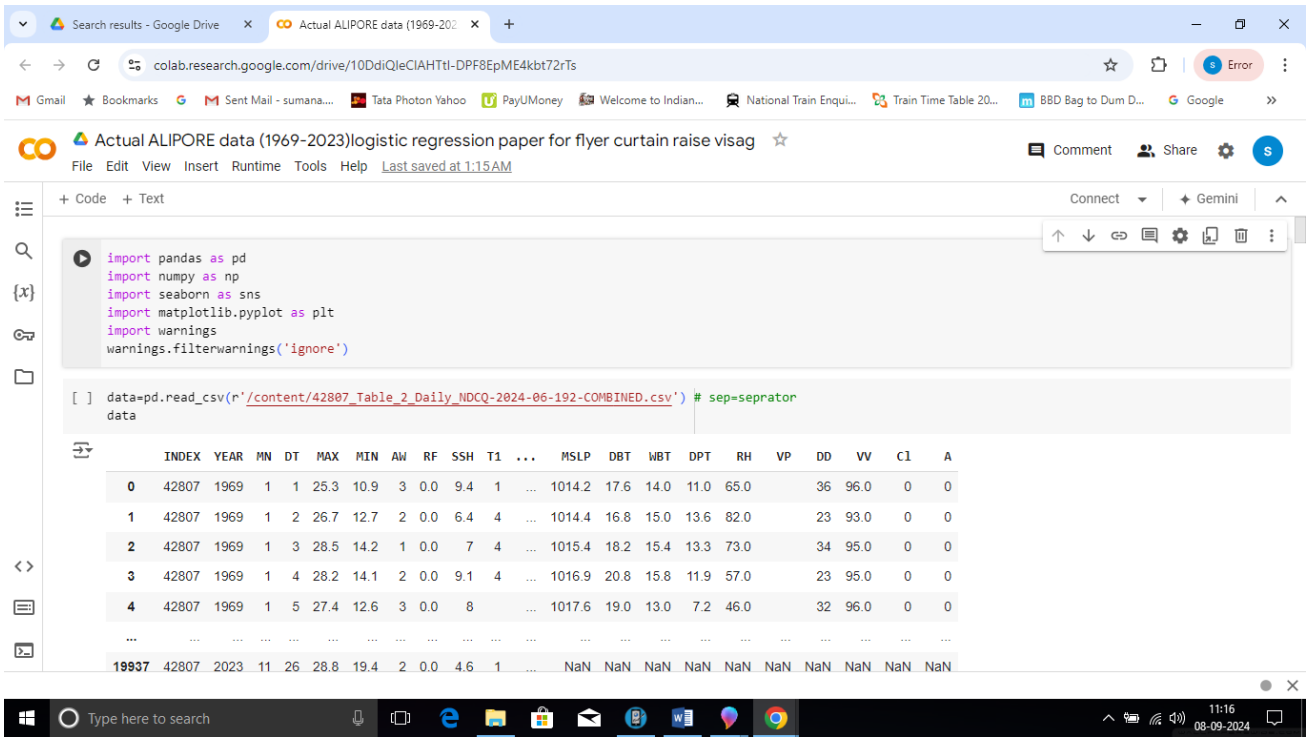
Research gap- Several research papers are there , based on data analysis /machine learning related to weather prediction ,with the help of trained data set, to determine predicted weather for next-day with the help of previous day input data ,but basically these were obtained in some different manner, not by such analysis ,where with the help of logistic regression model analysis had been done and where data set used was big data ,historical data starting from 1969 till 2023(year 2023 was absent for table 3). So naturally the analysis was more dependable to obtain future prediction more accurately.

RESEARCH QUESTIONS /HYPOTHESIS

In this research , basically we wanted to determine whether significant that means weather with rain or thunder or drizzle would occur or not. Depending on the output we had to reach that. Based on the model, with the help of trend of this previous data pattern ,we could get array of probable weather pattern ,lastly with the help of input of previous day data ,we can find next day weather pattern ,whether clear weather will be there or not.

METHODS:

Data-science ,machine learning, statistical theory ,data analysis ,scientific methods all together run hand to hand . In our case for weather analysis , we first of all collected weather data from the data supply portal of IMD PUNE (India Meteorological Department –Pune, data supply centre).From this portal, online collection of Alipore (station code 42807) data in CSV (comma separated)file format was collected for the purpose of weather analysis and to get weather prediction as model output . Scikit learn is an open source library for data analysis built on Numpy, Scipy and matplotlib , seaborn etc. In our case ,this research design is based upon qualitative analysis ,where we want to predict the qualitative output, whether the outcome is 'significant weather' or 'clear weather' .To get the output ,data analysis acts a vital role in this research design. The first step is data engineering. It consists of data download, to understand the volume of data (shape of data),conversion of data in proper format, deleting (drop command) of duplicate data ,null data to prepare the compatibility of successful execution of python code. Also to label encoding different types of weather criteria into numeric value by 'apply lambda' function. Based on this transformation and also various codes of data analysis ultimately reached the output of prediction. Some screenshots of procedure of data analysis had been given herewith sequentially.

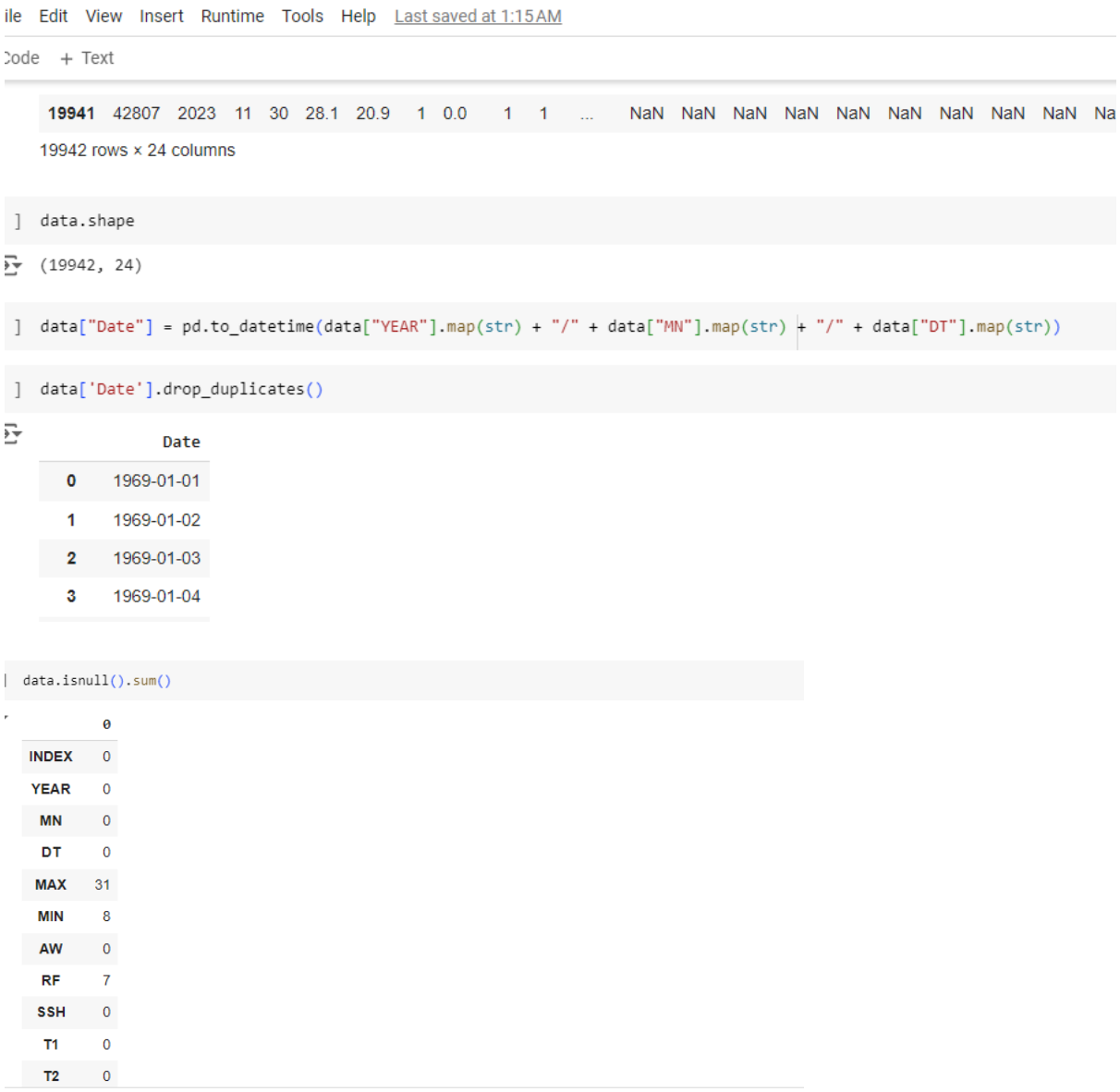


	INDEX	YEAR	MN	DT	MAX	MIN	AW	RF	SSH	T1	...	MSLP	DBT	WBT	DPT	RH	VP	DD	VV	Cl	A
0	42807	1969	1	1	25.3	10.9	3	0.0	9.4	1	...	1014.2	17.6	14.0	11.0	65.0		36	96.0	0	0
1	42807	1969	1	2	26.7	12.7	2	0.0	6.4	4	...	1014.4	16.8	15.0	13.6	82.0		23	93.0	0	0
2	42807	1969	1	3	28.5	14.2	1	0.0	7	4	...	1015.4	18.2	15.4	13.3	73.0		34	95.0	0	0
3	42807	1969	1	4	28.2	14.1	2	0.0	9.1	4	...	1016.9	20.8	15.8	11.9	57.0		23	95.0	0	0
4	42807	1969	1	5	27.4	12.6	3	0.0	8		...	1017.6	19.0	13.0	7.2	46.0		32	96.0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19937	42807	2023	11	26	28.8	19.4	2	0.0	4.6	1	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Collected two csv files table 2 and table 3 from IMD PUNE DATA SUPPLY PORTAL ,merged as new excel file '42807_Table_2_Daily_NDCQ-2024_06_192_COMBINED-EXCEL file' ,then converted this merged file into csv file type for data analysis. Received this data file in content folder of google collab for analysis.Checked the shape /volume of data at initial situation before editing,merged the three columns of date related fields 'year','month','date' as one single field for ease of data engineering and then cheked whether any duplicate value was there to avoid duplication error.

	Date
0	1969-01-01
1	1969-01-02
2	1969-01-03
3	1969-01-04

Then checked the data pattern as sample beginning from head of data with code as herewith below, also checked for null values (data not present) to avoid accuracy problem.



ile Edit View Insert Runtime Tools Help Last saved at 1:15 AM

Code + Text

19941	42807	2023	11	30	28.1	20.9	1	0.0	1	1	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

19942 rows × 24 columns

```
data.shape
```

```
(19942, 24)
```

```
data["Date"] = pd.to_datetime(data["YEAR"].map(str) + "/" + data["MN"].map(str) + "/" + data["DT"].map(str))
```

```
data['Date'].drop_duplicates()
```

	Date
0	1969-01-01
1	1969-01-02
2	1969-01-03
3	1969-01-04

```
data.isnull().sum()
```

	0
INDEX	0
YEAR	0
MN	0
DT	0
MAX	31
MIN	8
AW	0
RF	7
SSH	0
T1	0
T2	0

From this source of data, Alipore, weather data since 1969 to 2023 as per availability ,obtained from IMD PUNE ,DATA SUPPLY PORTAL.As checked there ,whether the dataset had any null value for any feature there. The required code was as above where in our dataset there were some null values as in one of the two csv datasets ,table 2 and table 3. In table 2 ,data were absent from 27th december 2022 , as obtained,which considered as null values and had to delete to avoid erronius results. As obtained from table 2, data for 'maximum temperature','minimum temperature','average wind','rainfall','sunshine hour' and 'weather phenomenon' in four quarters as mentioned there for a day,24 hours .On the other hand, from table 3 ,We obtained 'station level pressure','Mean sea level pressure','dry bulb temperature','wet bulb temperature','dew point temperature','relative humidity','vapourpressure','wind direction','visibility','low cloud''amount of low cloud' . Deleting duplicate data ,dropped null values ,rows with null values and in this way reshaped the whole data set to make compatible for analysis. Moreover Instead of four weather phenomena ,as representation of weather parameter criteria in four quarters of a day,we considered a new variable T ,to represent T1,T2,T3,T4 with the help of lambda function . If any significant weather was present for any quarter ,that means if any of the representing weather T1,T2,T3,T4 was within '9','6','0' ,'5' (where '9' represented there as thunder-storm,'6' represented there as 'rain','0' represented there as lightning and '5' represented there as drizzle)

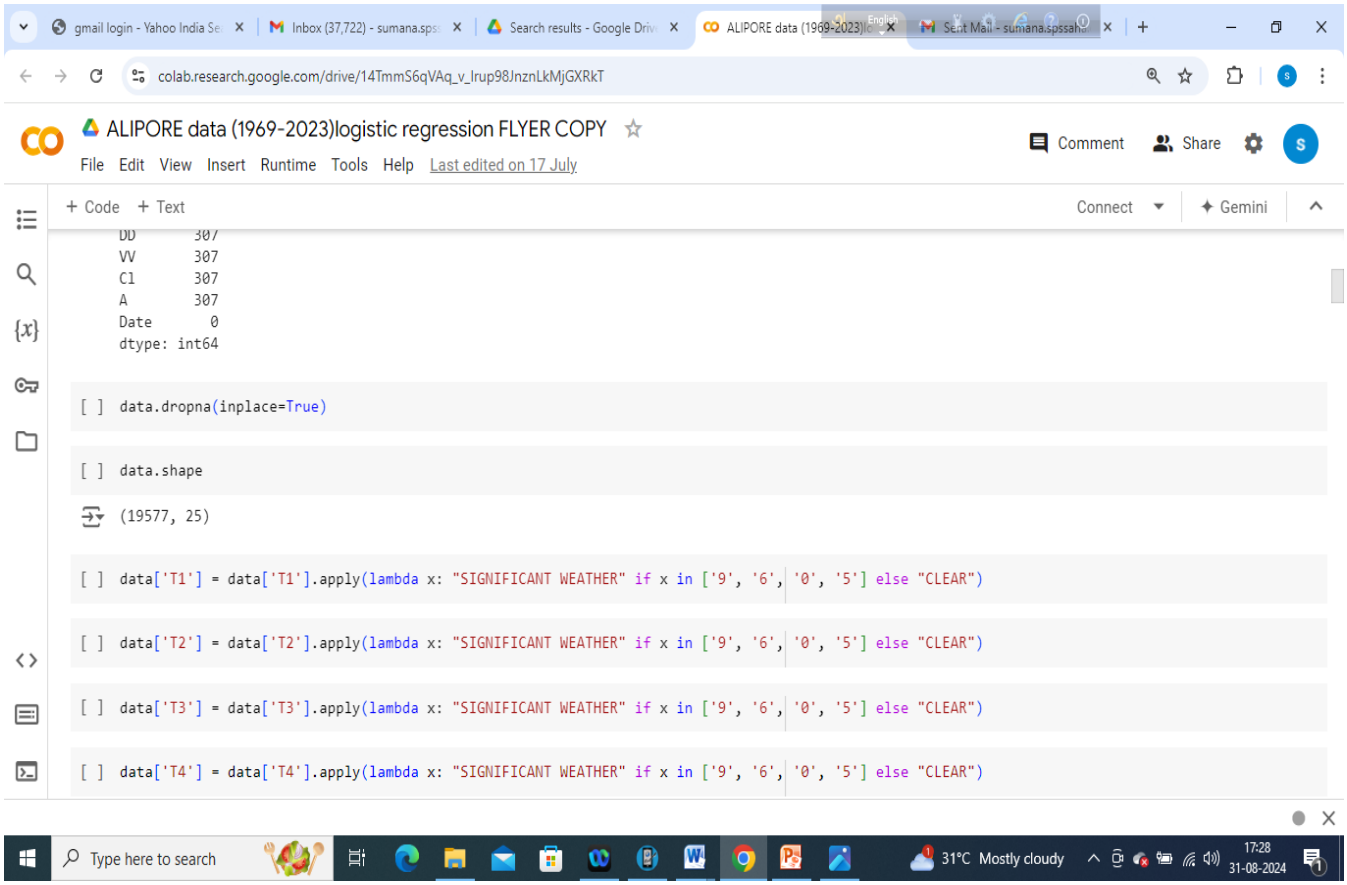
T Type of weather in code

Code	Weather	Code	Weather	Code	Weather	Code	Weather
0	Lightning	1	Haze	2	Mist	3	Sand/ Dust storm
4	Fog	5	Drizzle	6	Rain	7	Squall
8	Gale	9	Thunder storm	J	Hail storm	K	Dust fog
L	Line squall	M	Ground frost	N	Dew	O	Snow/sleet

G Time of commencement of weather (If one weather phenomenon has occured more than once in a day then first time of its commencement is reported).

Code	Time between hours IST	Code	Time between hours IST
1	0001 to 0300	5	1201 to 1500
2	0301 to 0600	6	1501 to 1800
3	0601 to 0900	7	1801 to 2100
4	0901 to 1200	8	2101 to 2400

DUR Duration in minutes upto 804 minutes. For duration more than 804, value is rounded off to the nearest tens of minutes and 800 is added and the resulting value is entered. e.g. if duration is 947 minutes then it is entered as : 895 i.e. 95 + 800.



```
[ ] data['T']=data['T1']+data['T2']+data['T3']+data['T4']

[ ] data['T'] = data.apply(lambda row: "SIGNIFICANT WEATHER" if (row['T1']=="SIGNIFICANT WEATHER") or
                                   (row['T2']=="SIGNIFICANT WEATHER") or (row['T3']=="SIGNIFICANT WEATHER")
                                   or (row['T4']=="SIGNIFICANT WEATHER") else "CLEAR", axis=1)

[ ] data = data.drop('T1',axis=1)

[ ] data = data.drop('T2',axis=1)

[ ] data = data.drop('T3',axis=1)

    data = data.drop('T4',axis=1)
```

With the help of label encoding ,thus represented the weather for each day by the new variable 'T' ,as image 6.1 above and then dropped (deleted) the existing predictors T1,T2,T3,T4 to avoid error of duplication.Then understanding the effect of this transformation by filtering data for significant weather by data.

Edit View Insert Runtime Tools Help Last saved at 1:15 AM

+ Text Connect

```
data = data.drop('T4',axis=1)
```

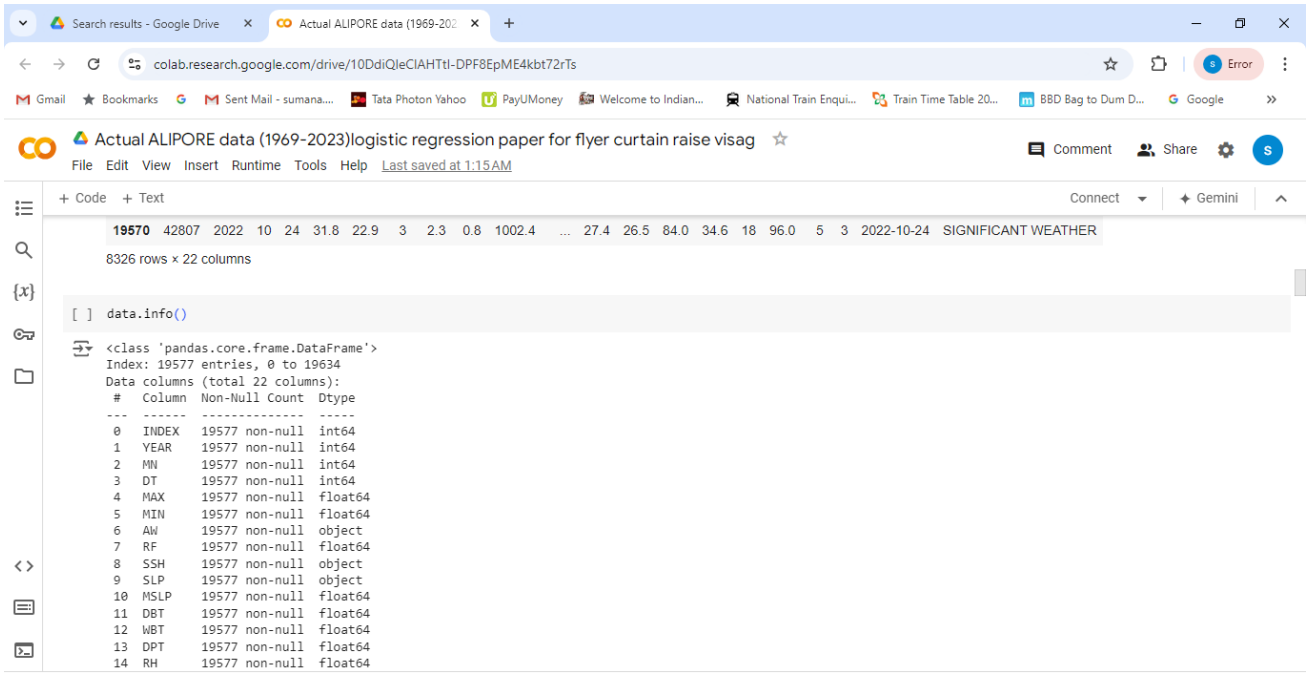
```
filter=data['T']== "SIGNIFICANT WEATHER"
filter_new=data[filter]
filter_new
```

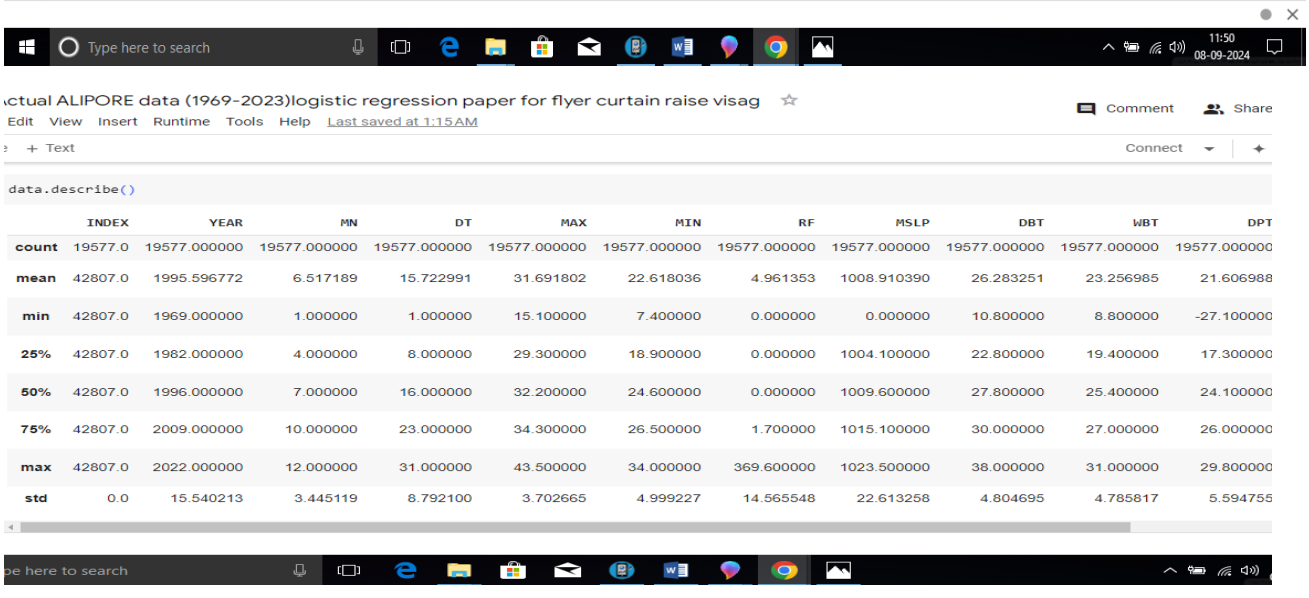
	INDEX	YEAR	MN	DT	MAX	MIN	AW	RF	SSH	SLP	...	WBT	DPT	RH	VP	DD	VV	Cl	A	Date	T
13	42807	1969	1	14	27.5	15.7	5	26.0	8.7	1014.3	...	17.6	16.5	85.0		18	96.0	5	5	1969-01-14	SIGNIFICANT WEATHER
57	42807	1969	2	27	34.7	21.3	7	0.0		1014.8	...	21.4	17.6	53.0		25	96.0	0	0	1969-02-27	SIGNIFICANT WEATHER
58	42807	1969	2	28	36.0	22.4	6	0.0		1013.5	...	22.2	19.4	62.0		23	97.0	0	0	1969-02-28	SIGNIFICANT WEATHER
59	42807	1969	3	1	36.2	21.2	6	0.2		1013.2	...	22.6	20.9	69.0		23	97.0	0	0	1969-03-01	SIGNIFICANT WEATHER
75	42807	1969	3	17	34.3	25.4	10	0.0	8.6	1009.9	...	25.2	23.8	76.0		16	97.0	4	2	1969-03-17	SIGNIFICANT WEATHER
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19559	42807	2022	10	13	34.2	25.0	4	7.1	6.7	1003.8	...	28.4	27.8	88.0	37.4	0	96.0	0	0	2022-10-13	SIGNIFICANT WEATHER
19560	42807	2022	10	14	33.7	26.1	2	17.8	8.2	1002.9	...	28.0	27.3	87.0	36.3	0	96.0	5	4	2022-10-14	SIGNIFICANT WEATHER
19565	42807	2022	10	19	32.4	26.0	4	0.0	0	1004.2	...	28.4	27.7	85.0	37.1	0	96.0	4	4	2022-10-19	SIGNIFICANT WEATHER
19569	42807	2022	10	23	33.6	24.5	4	0.0	8.8	1002.4	...	27.0	26.8	97.0	35.2	0	95.0	4	4	2022-10-23	SIGNIFICANT WEATHER

Then other necessary feature engineering such as data.info() to understand type of data,list of columns of data presently existing then ,data.describe() to check statistical result ,value counts for significant and clear weather in the whole data set.

```
[ ] col=list(data.columns)
    col
```

```
['INDEX',
 'YEAR',
 'MN',
 'DT',
 'MAX',
 'MIN',
 'AW',
 'RF',
 'SSH',
 'SLP',
 'MSLP',
 'DBT',
 'WBT',
 'DPT',
 'RH',
 'VP',
 'DD',
 'VV',
 'Cl',
 'A',
 'Date',
 'T']
```





	INDEX	YEAR	MN	DT	MAX	MIN	RF	MSLP	DBT	WBT	DPT
count	19577.0	19577.000000	19577.000000	19577.000000	19577.000000	19577.000000	19577.000000	19577.000000	19577.000000	19577.000000	19577.000000
mean	42807.0	1995.596772	6.517189	15.722991	31.691802	22.618036	4.961353	1008.910390	26.283251	23.256985	21.606988
min	42807.0	1969.000000	1.000000	1.000000	15.100000	7.400000	0.000000	0.000000	10.800000	8.800000	-27.100000
25%	42807.0	1982.000000	4.000000	8.000000	29.300000	18.900000	0.000000	1004.100000	22.800000	19.400000	17.300000
50%	42807.0	1996.000000	7.000000	16.000000	32.200000	24.600000	0.000000	1009.600000	27.800000	25.400000	24.100000
75%	42807.0	2009.000000	10.000000	23.000000	34.300000	26.500000	1.700000	1015.100000	30.000000	27.000000	26.000000
max	42807.0	2022.000000	12.000000	31.000000	43.500000	34.000000	369.600000	1023.500000	38.000000	31.000000	29.800000
std	0.0	15.540213	3.445119	8.792100	3.702665	4.999227	14.565548	22.613258	4.804695	4.785817	5.594755

File Edit View Insert Runtime Tools Help Last saved at 1:15 AM

Code + Text

	T
CLEAR	11251
SIGNIFICANT WEATHER	8326

dtype: int64

```
[ ] data['T']=np.where(data['T']=='SIGNIFICANT WEATHER',1,0)
    data['T'].dtype
```

```
dtype('int64')
```

```
data['T'].value_counts()
```

T	count
0	11251
1	8326

dtype: int64

Then understanding the effect of this transformation by filtering data for significant weather by data. describe , value_counts etc. As this is a case of qualitative data analysis, and there are object type data also ,so here we used IV (Information Value)method to determine strong ,medium and weak predictors. For this method score value as follows:

The process of data analysis by IV method and the output of scores are as follows:

```
def calculate_woe_iv(dataset, feature, target):
    lst = []
    for i in range(dataset[feature].nunique()):
        val = list(dataset[feature].unique())[i]
        lst.append({
            'Value': val,
            'All': dataset[dataset[feature] == val].count()[feature],
            'Good': dataset[(dataset[feature] == val) & (dataset[target] == 1)].count()[feature],
            'Bad': dataset[(dataset[feature] == val) & (dataset[target] == 0)].count()[feature]
        })

    dset = pd.DataFrame(lst)
    dset['Distr_Good'] = dset['Good'] / dset['Good'].sum()
    dset['Distr_Bad'] = dset['Bad'] / dset['Bad'].sum()
    dset['WoE'] = np.log(dset['Distr_Good'] / dset['Distr_Bad'])
    dset = dset.replace({'WoE': {np.inf: 0, -np.inf: 0}})
    dset['IV'] = (dset['Distr_Good'] - dset['Distr_Bad']) * dset['WoE']
    iv = dset['IV'].sum()

    dset = dset.sort_values(by='WoE')
```

Type here to search

```
df_new = pd.concat([df_new, df_new_2], ignore_index=True)
df_new
```

	Feature	IV-Score
0	AW	0.263037
1	SSH	0.758800
2	SLP	0.135565
3	VP	0.131694
4	DD	0.043778
5	Cl	0.108874
6	A	0.053993

Result of analysis by IV method as above. Depending on the score value we had to take decision of screening the variables according as the score value and according as the priority of predictor indicated there. The importance was decided as : **<0.02 useless ,0.02 to 0.1 weak predictors, 0.1 to 0.3 medium predictors, 0.3 to 0.5 strong predictors.0.5 suspicious .**

File Edit View Insert Runtime Tools Help Last saved at 10:32 AM

Code + Text

5	Cl	0.108874
6	A	0.053993

```
[52] data.drop(columns=['DD','A'],inplace=True)

[53] data.columns

Index(['INDEX', 'YEAR', 'MN', 'DT', 'MAX', 'MIN', 'AW', 'RF', 'SSH', 'SLP',
       'MSLP', 'DBT', 'WBT', 'DPT', 'RH', 'VP', 'VV', 'Cl', 'Date', 'T'],
      dtype='object')

[54] data.dtypes
```

According to the score of IV method , we dropped the columns ‘DD’ and ’A’ as these are considered as weak predictors according as the IV method. After IV method, the data file subjected to the process of ‘one hot encoding’ and label encoding to transform all the data into numeric type and compatible.

Actual ALIPORE data (1969-2023)logistic regression

Edit View Insert Runtime Tools Help Last saved at 10:32 AM

Comment

+ Text

RAM Disk

19633	42807	2022	12	26	28.7	20.3	4	0.0	5.4	1015.1	1015.8	23.0	20.6	19.2	79.0	22.2	96.0	0	2022-12-26	0
19634	42807	2022	12	27	28.9	20.7	3	0.0	6.7	1015.7	1016.4	23.6	21.4	20.2	81.0	23.7	96.0	0	2022-12-27	0

19577 rows × 20 columns

```
#one hot encoding
col_list=[]
for col in data.columns:
    if((data[col].dtype=='object') & (col!='T')):
        col_list.append(col)
df_2=pd.get_dummies(data[col_list],drop_first=True)

for col in df_2.columns:
    df_2[col]=df_2[col].astype(int)
df_2
```

	AW_0	AW_1	AW_10	AW_11	AW_12	AW_13	AW_14	AW_15	AW_16	AW_17	...	Cl_0	Cl_1	Cl_2	Cl_3	Cl_4	Cl_5	Cl_6	Cl_7	Cl_8	Cl_9
0	0	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0	0	0	0	0

Connected to Python 3 Google Compute Engine backend

Code + Text

4	42807	1969	1	5	27.4	12.6	0.0	1017.6	19.0	13.0	...	1	0

5 rows × 834 columns

```
64] #label encoder
    from sklearn.preprocessing import LabelEncoder
    labelencoder=LabelEncoder()
```

```
    for i in col_list:
        data[i]=labelencoder.fit_transform(data[i])
```

+ C

```
66] data.head()
```

This time, the columns DATE, INDEX , YEAR, MONTH were discarded, as these parameters would not matter on this type of data analysis of qualitative type. Then to remove the factor of multi-collinearity ,we executed VIF (variation inflation factor) method repeatedly one after another checking output with VIF value and dropping the column with highest VIF value ,each time.

```
data.drop(columns=['Date','INDEX','YEAR','MN','DT'],inplace=True)
```

```
#vif
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

```
col_list=[]
for col in data.columns:
    if ((data[col].dtype!='object')&(col!='T')):
        col_list.append(col)
X=data[col_list]
vif_data=pd.DataFrame()
vif_data['Feature']=X.columns
vif_data['VIF']=[variance_inflation_factor(X.values,i) for i in range(len(X.columns))]
vif_data
```

	Feature	VIF
0	MAX	231.067895
1	MIN	69.765153

Thus reached the position with predictors having VIF factor 6 and less than 6 and thus ascertained the revised data set.

```
for col in data.columns:
    if ((data[col].dtype!='object')&(col!='T')):
        col_list.append(col)
X=data[col_list]
vif_data=pd.DataFrame()
vif_data['Feature']=X.columns
vif_data['VIF']=[variance_inflation_factor(X.values,i) for i in range(len(X.columns))]
vif_data
```

	Feature	VIF
0	AW	3.613193
1	RF	1.174193
2	SSH	2.335474
3	SLP	2.528825
4	VP	6.272426
5	CI	4.445478

Now with the training and testing data set , with ratio 80% ,20% ,formed the test-train split and built logistic regression model and prediction data set.

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.80,random_state=0
```

```
pd.DataFrame(x_train)
```

	AW	RF	SSH	SLP	VP	Cl
580	22	11.0	71	18	216	8
19058	22	0.0	95	101	201	6
17722	2	0.0	76	29	231	5
2096	21	0.0	116	95	79	1
804	22	22.6	16	78	151	5

The accuracy score, classification report, confusion matrix (with true positive, true negative, false positive, false negative), precision, recall ,heatmap were determined accordingly to understand the success rate of the model.

```
] accuracy_score(y_test,test_pred)
```

```
0.7221654749744637
```

```
] print(classification_report(y_test,test_pred))
```

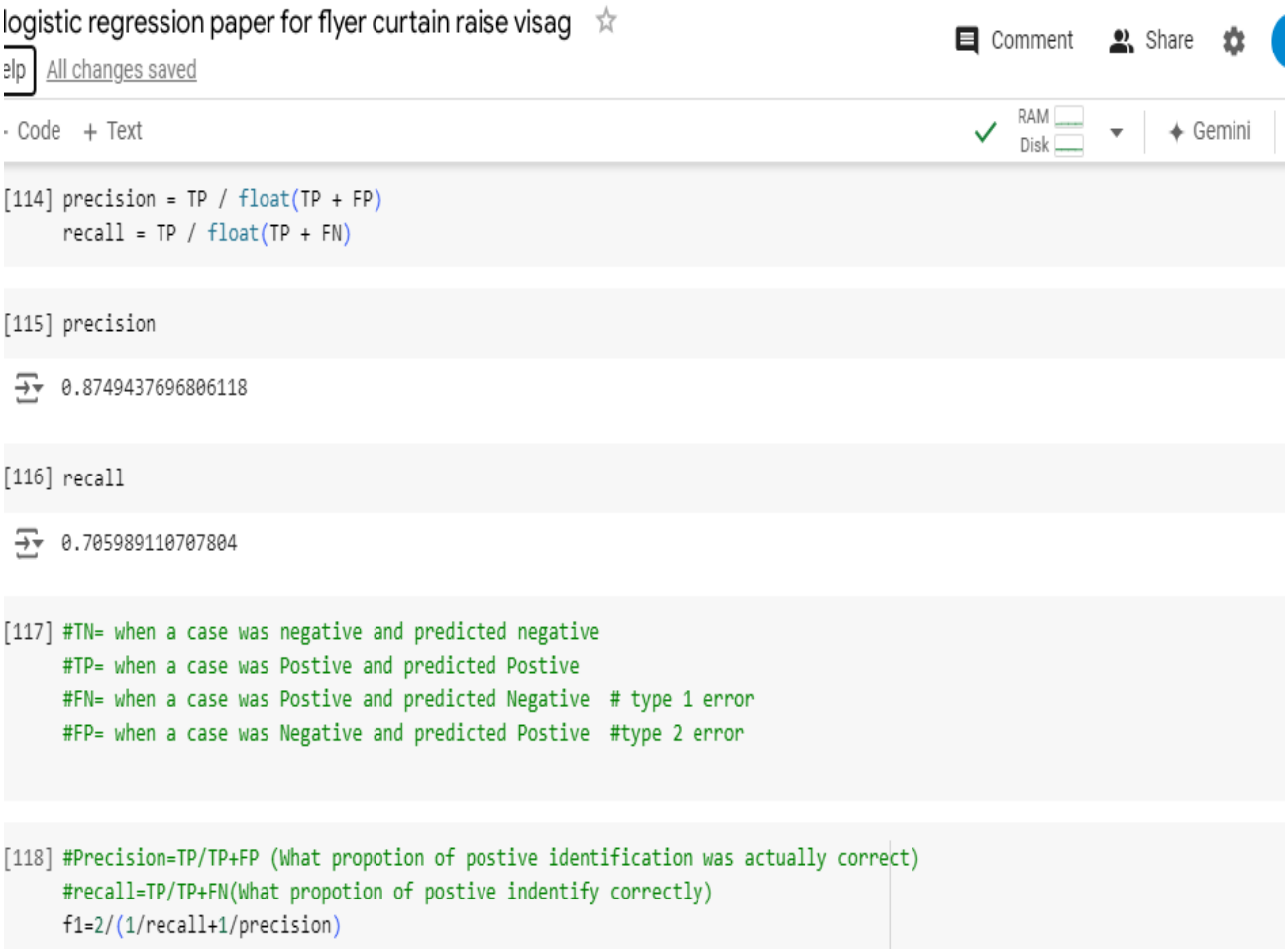
```
              precision    recall  f1-score   support

           0       0.71      0.87      0.78      2223
           1       0.76      0.52      0.62      1693

    accuracy                           0.72      3916
   macro avg       0.73      0.70      0.70      3916
weighted avg       0.73      0.72      0.71      3916
```

```
] precision = TP / float(TP + FP)
  recall = TP / float(TP + FN)
```

```
] #TN= when a case was negative and predicted negative
```

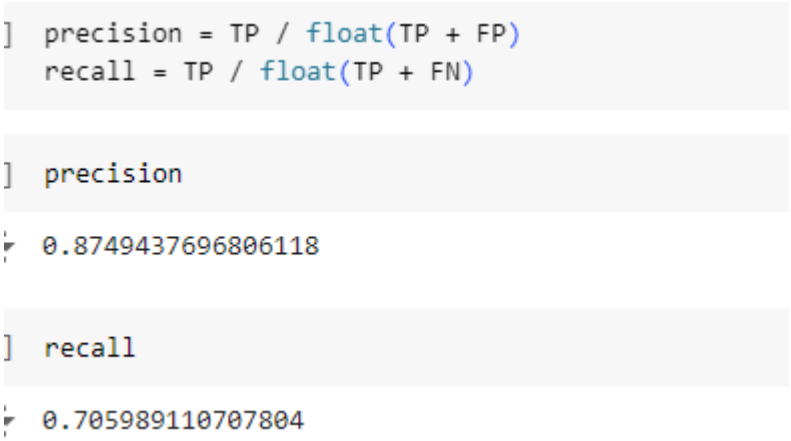
```
[ ] sns.heatmap(cf_matrix, annot=True)
```

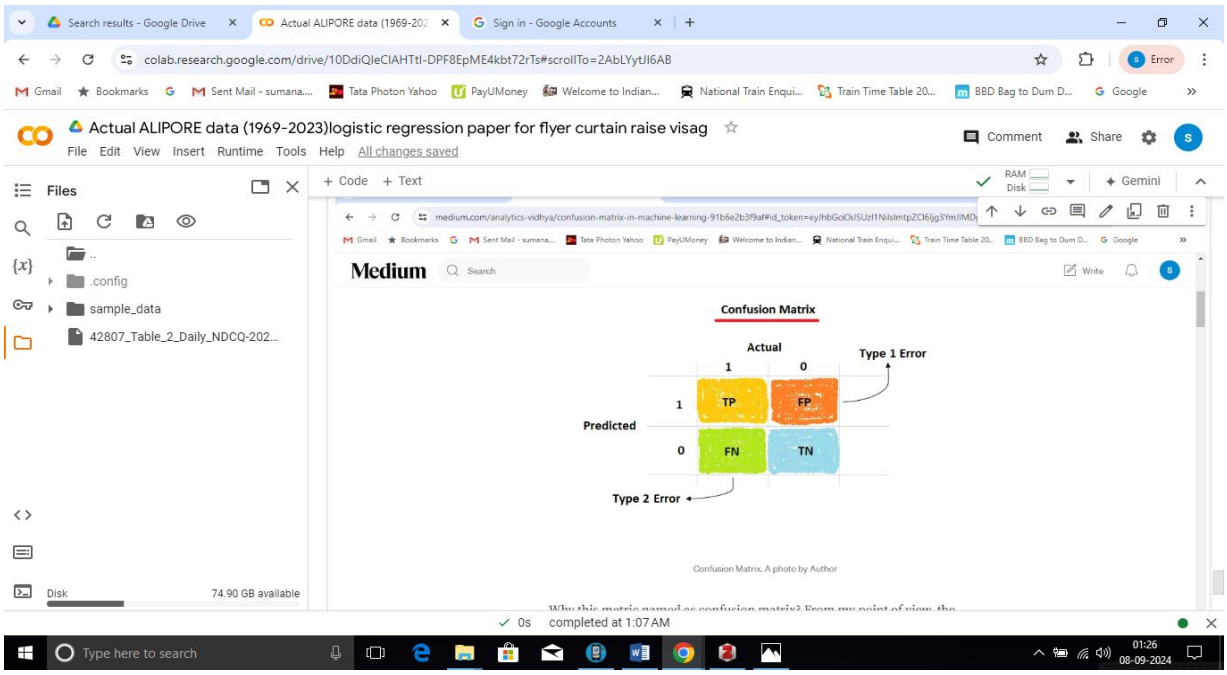
```
<Axes: >
```



The F1 score is : F1 = 2 * precision * recall / (precision + recall). It ranges from 0 to 1, 1 representing perfect precision and recall, and 0 indicates poor performance. Here value of f1 is 0.78 , so not so bad performance of model in this research.

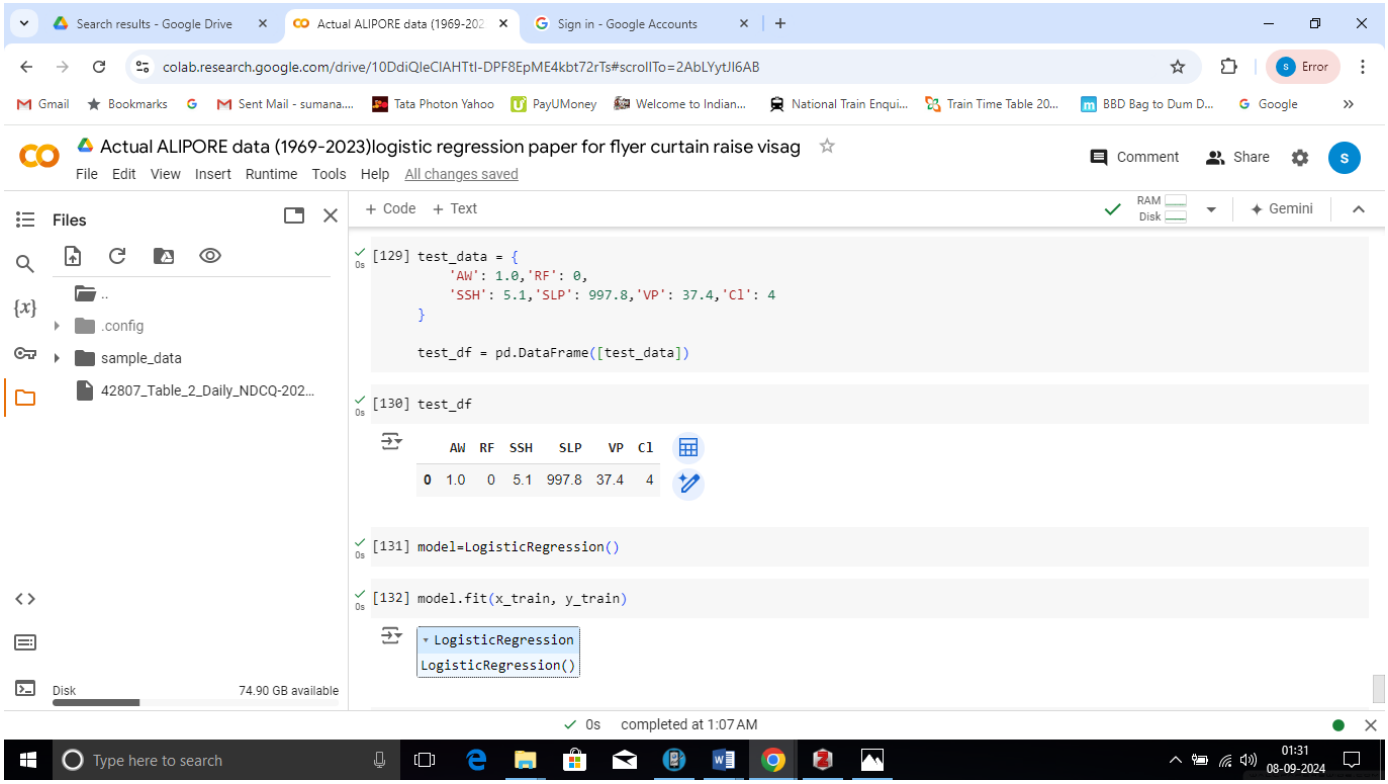
Precision and recall value as image shown below:





Citation: Chauhan,Amit/"Confusion Matrix In Machine Learning",*Analytics Vidya* ,2021.

https://medium.com/analytics-vidhya/confusion-matrix-in-machine-learning-91b6e2b3f9af



Prediction of next day weather as below on the basis of input of previous day weather data for the weather parameter as given for test data series as picture shown just above.

```
model.predict(test_df)
```

```
array([0])
```

As prediction here ,the output was '0' so the predicted weather as derived here as 'clear' weather.

SIGNIFICANCE OF THE STUDY

If with the help of machine learning, the trend of upcoming weather pattern, somehow predicted based on this research study with the help of logistic regression model, then it will be helpful.
Similar as this research study following the research and data analysis pattern, updating data till date , we can get probable weather for few upcoming days. Obviously this would be beneficial for all fields.

TIMELINE Regarding this research study , it must be mentioned the great contribution of the advanced course of 'CCE IIT MADRAS' ,conducted by 'INTELLIPAAT' completed within one year, consisting of hands-on of 'python data analysis' on zupyter as well as google collaboratory platform ,the statistical theory based on these data analysis ,and course also with other software like my-sql and power-bi.From this data analysis consisting of different types of analysis pattern , such as pandas ,linear regression ,logistic regression, decision tree, random forest ,time series analysis, tensor flow ,natural language processing, KNN(K-Nearest Neighbour) ,image recognition etc. This research study is a try of data analysis for prediction of output with logistic regression , the idea obtained from data -analysis with the help of predictors ,where predictors are various weather parameters and this is a type of categorical analysis with output as '1' and '0' ,where '1' considered as all type of 'significant' weather and '0' considered as 'clear' weather. The total time taken is about two years considering course of data-analysis, acquiring ability to understand as well as execute the programme with compatible data to get correct output and the rest is own idea how to form a suitable analysis to predict outcome on the basis of previous knowledge.

Conclusion and future work: In this research design, this was dealt with weather prediction with the help of logistic regression method. The accuracy score, F1 score and the confusion matrix all proved that the performance of the model was good enough . In future this may be applied with the addition of latest appended data to get further more accurate result .The data analysis may be done also with the help of other machine learning tools to compare between ,to get the more accurate output. In future , we can try this by other advanced AI and ML technique with the help of pretrained model and fine tuning.

References:-

1) Singh, H.S.(2023,January 1). The python code file , shared in google drive ,by the hands-on faculty Mr Hitesh ,from Intellipaat, Training centre in Bengaluru, Karnataka.
https://colab.research.google.com/drive/1wg8IzcaQeLj_7FLpq23ishdJ3XnQKfbl

2) Srinivasan,Dr Babji. "Note on logistic regression. *PDF file received as download in the LMS portal of data analysis training site INTELLIPAAT* ,11/12/2022"
Doi 11/12/2022

3) ,Collection of data from "*Data supply portal India Meteorological Department,Pune* ,https://dsp.imdpune.gov.in/"

Format: Author's last name, first name. "Title of the Article." Magazine. Month and year of publication: page numbers.

Bibilography :1) Jayasingh,S.K. et.al "Smart Weather Prediction Using Machine Learning". ResearchGate. May 2022: Pg.571-583

4) Mathur, Saksham .et.al "Prognostication of weather patterns using meteorological data and ML techniques".ResearchArticle . Volume 11,2024.