International Journal of Global Research Innovations & Technology (IJGRIT)
ISSN : 2583-8717, Impact Factor: 6.382, Volume 02, No. 04, October-December, 2024, pp 126-152

WEATHER PREDICTION USING DIFFERENT MACHINE LEARNING MODELS

Sumana Chatterjee*

ABSTRACT

This is a paper based on different machine learning models and comparison between these models to predict weather for certain day of a place Alipore (station code 42807).Based on analysis of big data as well as understanding the trend of output ,we had the objective to predict the probable weather . As we get weather from observational data ,basically we get two types of weather ,one type is significant weather with weather phenomena like lightning (code 0),drizzle(code 5), rain(code 6), thunder storm with rain(code 9) and another type is no such significant weather that means clear weather .For analysis ,we created some new dependent variable 'T' and considered 'T' as '1' for significant weather while considered 'T' as '0' for clear weather. Ultimately we used different machine learning models to predict probable weather for certain day along with comparison of performance rate for all these models.

Keywords: *Logistic Regression, Decision Tree, Random Forest, Naïve Bayes, Support Vector Machine, XG Boost, Grid Search CV, Hyper Parameter Tuning (Random Forest).*

Introduction

Prediction of weather event by artificial intelligence and machine learning is a great challenge. In this case prediction is based on analysis by various machine learning techniques under supervised learning. Starting from analysis by logistic regression method, one by one other machine learning supervised learning process executed with the training and testing data to obtain the result of prediction, accuracy score, classification report, confusion matrix as well as updating score card each time after execution of each technique. Each time we obtained the output value either '0' or '1'. In our case, each time, execution after each model,we obtained the same result as'0'.

Literature Review

Mainly the study material along with python code compatible with google collaboratory and jupyter platform obtained from hands on training of the course 'advanced certification in data science and ai', offered by 'CCE CODE IIT MADRAS', organised by 'INTELLIPAAT'. Other sources are online websites 'geeks for geeks', 'medium', 'towards data science', 'analytics vidya', 'w3 school' etc.

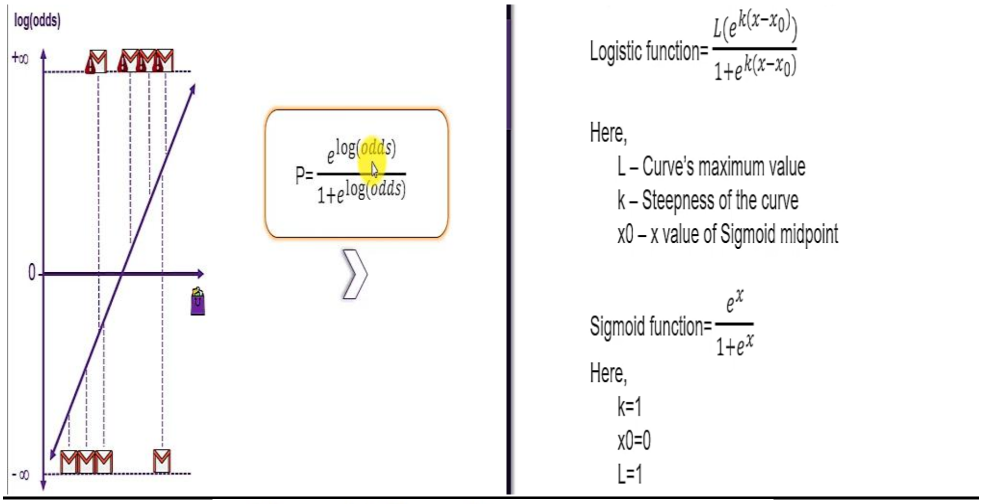
Theory and terminology used in this paper based on supervised machine learning technique

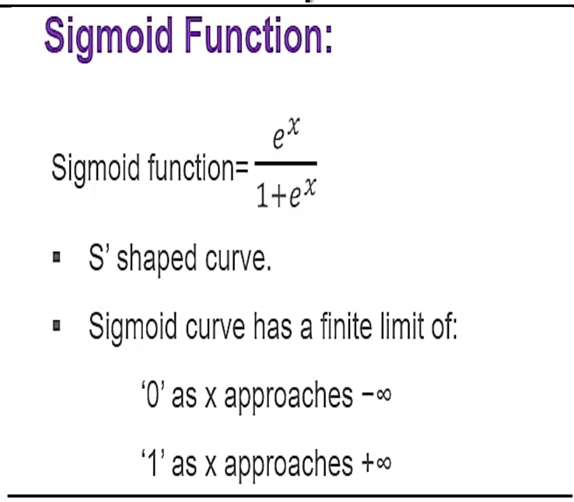
The theory and technique applied for different models are described herewith.

In this paper the prediction on test data set has been done by different machine learning models one by one and after execution of each model accuracy and updated score card is obtained to compare the efficiency. The theory and technique behind each model is described herewith.

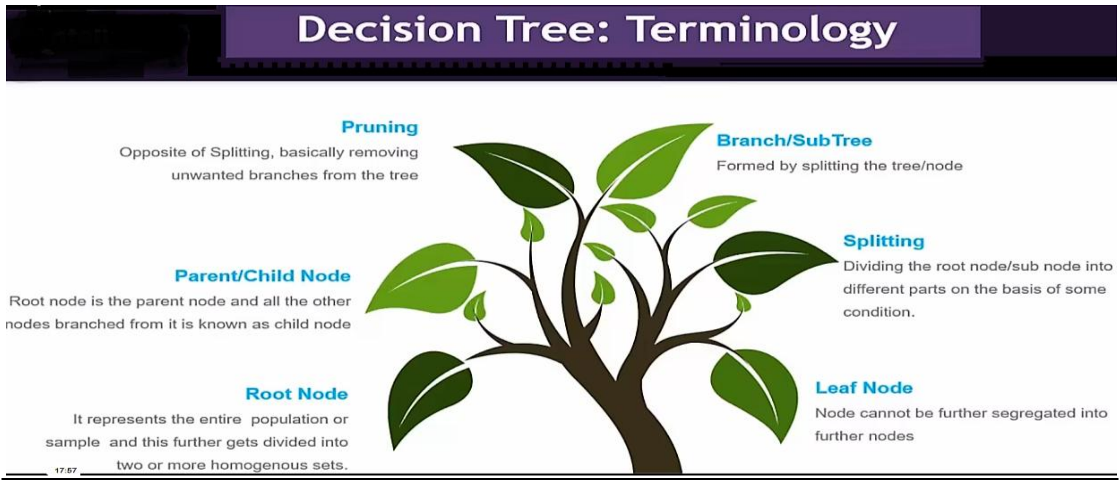
* India Meteorological Department, Nirwan University, Jaipur, Rajasthan, India.

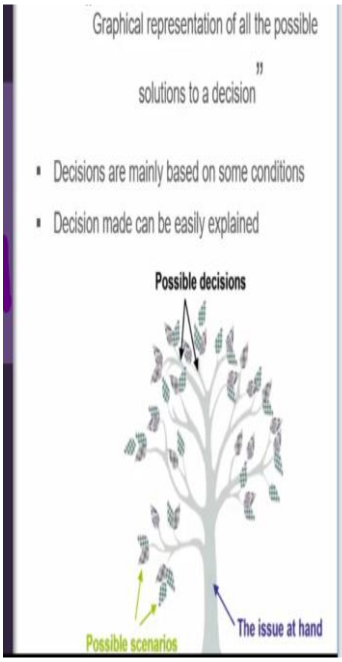
Logistic Regression Terminology and Technique

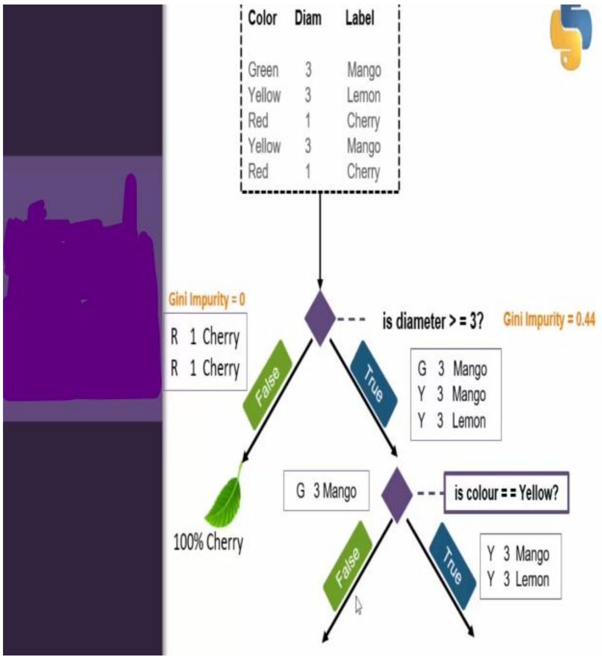




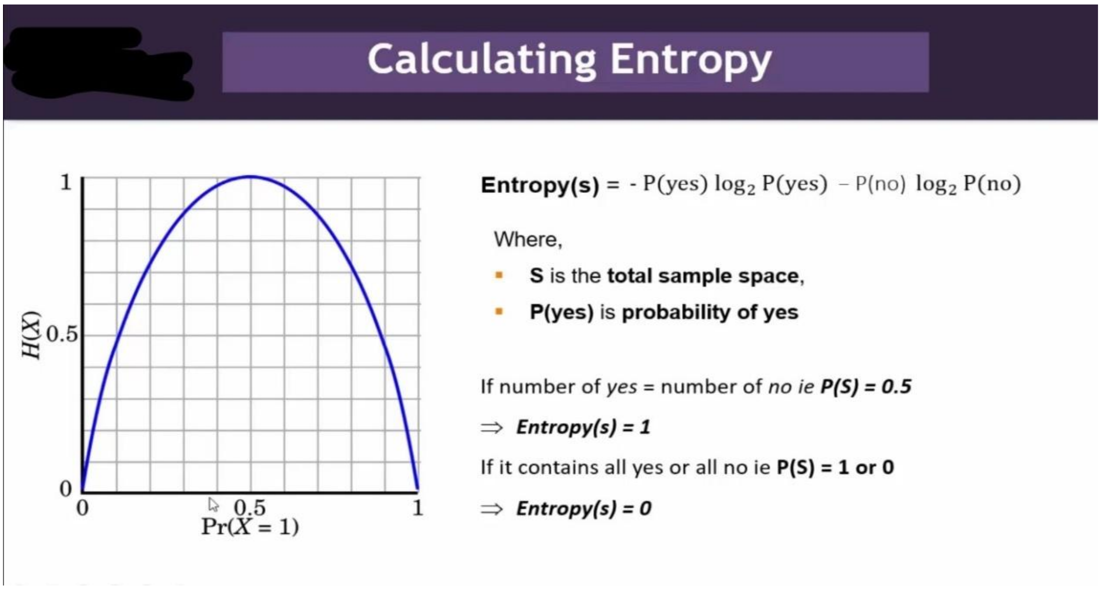
Decision Tree Terminology And Technique

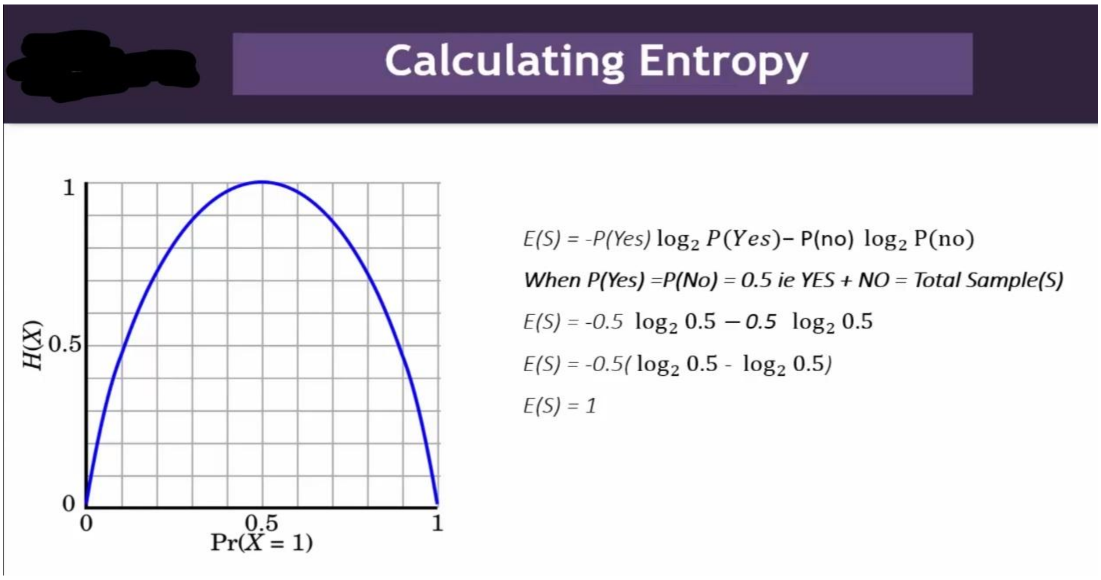


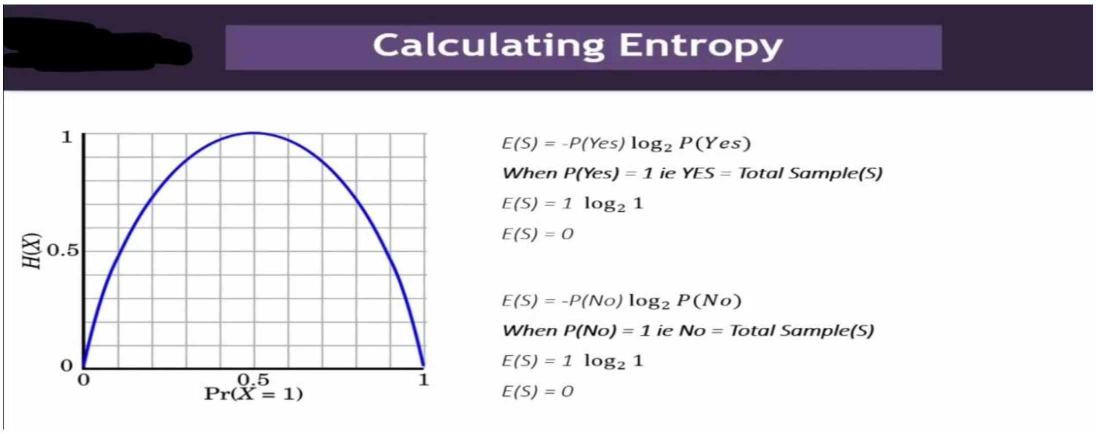

Graphical representation of all the possible
solutions to a decision"
Decisions are mainly based on some conditions
Decision made can be easily explained
Possible decisions
Possible scenarios
The issue at hand


Color Diam Label
Green 3 Mango
Yellow 3 Lemon
Red 1 Cherry
Yellow 3 Mango
Red 1 Cherry
Gini Impurity = 0
R 1 Cherry
R 1 Cherry
is diameter > = 3?
Gini Impurity = 0.44
False
True
G 3 Mango
Y 3 Mango
Y 3 Lemon
100% Cherry
G 3 Mango
is colour = = Yellow?
False
True
Y 3 Mango
Y 3 Lemon

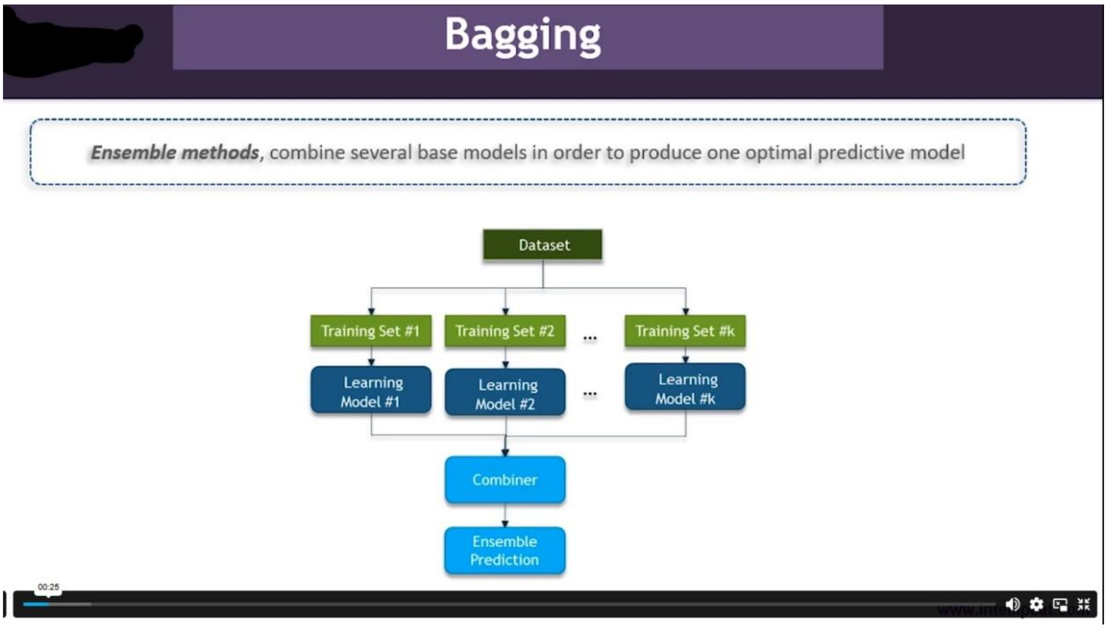

How do we split a Tree?
Entropy
Defines randomness in the data
It is a metric which measures the impurity
The first step to solve the problem of a decision tree
Information Gain
The information gain is the decrease in entropy after a dataset is split on the basis of an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain
1
2
3
4
Reduction in Variance
Reduction in variance is an algorithm used for continuous target variables (regression problems). The split with lower variance is selected as the criteria to split the population
Gini Index
The measure of impurity (or purity) used in building decision tree in CART is Gini Index

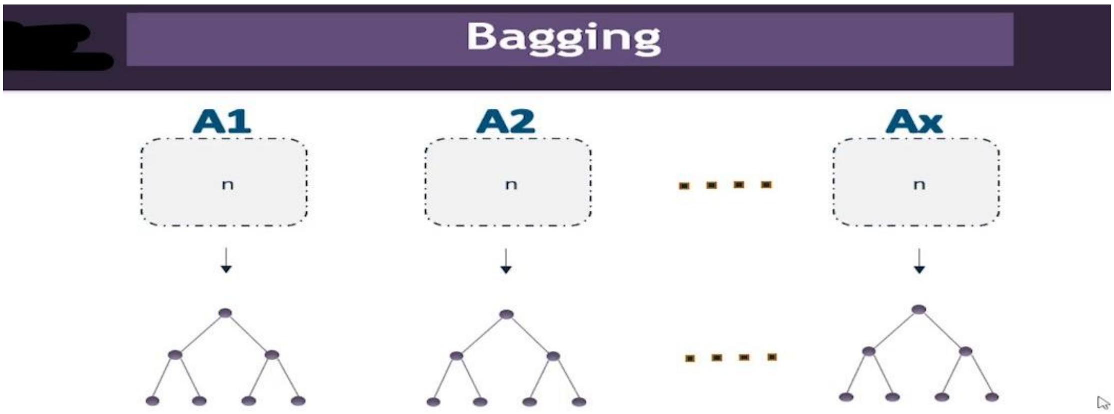

[redacted]
Calculating Entropy
Entropy(s) = - P(yes) log$_2$ P(yes) – P(no) log$_2$ P(no)
Where,
S is the total sample space,
P(yes) is probability of yes
If number of yes = number of no ie P(S) = 0.5
⇒ Entropy(s) = 1
If it contains all yes or all no ie P(S) = 1 or 0
⇒ Entropy(s) = 0
1
0.5
0
H(X)
0
0.5
1
Pr(X = 1)

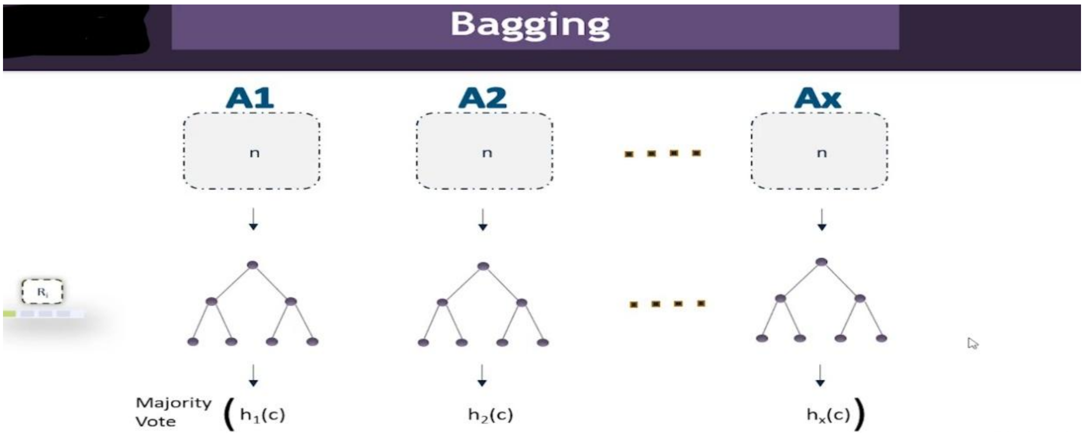

[redacted]
Calculating Entropy
E(S) = -P(Yes) log$_2$ P(Yes)– P(no) log$_2$ P(no)
When P(Yes) =P(No) = 0.5 ie YES + NO = Total Sample(S)
E(S) = -0.5 log$_2$ 0.5 – 0.5 log$_2$ 0.5
E(S) = -0.5(log$_2$ 0.5 - log$_2$ 0.5)
E(S) = 1
1
0.5
0
H(X)
0
0.5
1
Pr(X = 1)


[redacted]
Calculating Entropy
E(S) = -P(Yes) log$_2$ P(Yes)
When P(Yes) = 1 ie YES = Total Sample(S)
E(S) = 1 log$_2$ 1
E(S) = 0
E(S) = -P(No) log$_2$ P(No)
When P(No) = 1 ie No = Total Sample(S)
E(S) = 1 log$_2$ 1
E(S) = 0
1
0.5
0
H(X)
0
0.5
1
Pr(X = 1)

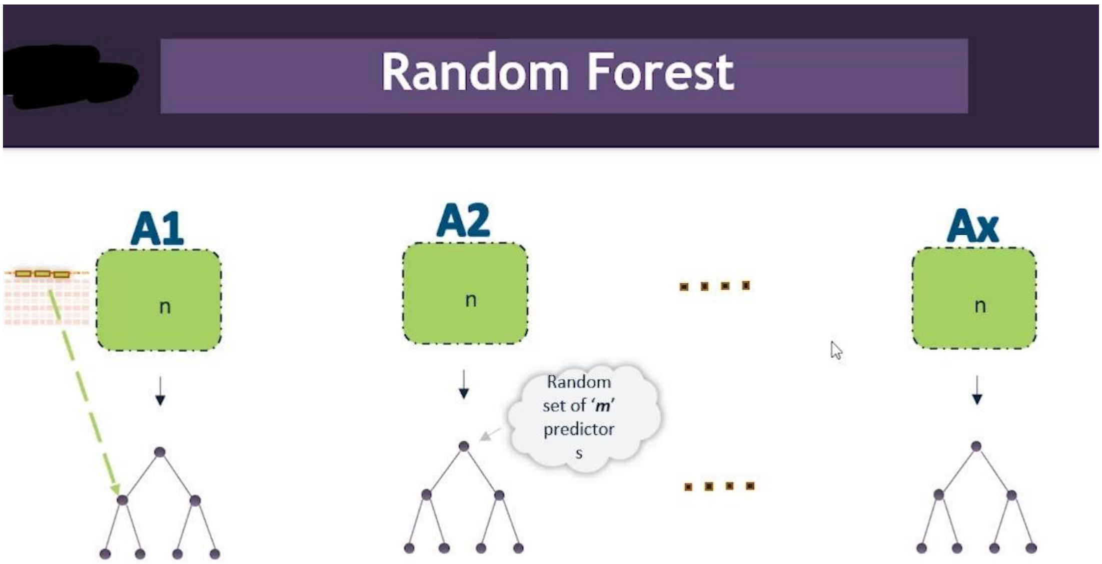
Terminology and Technique Associated with Random Forest





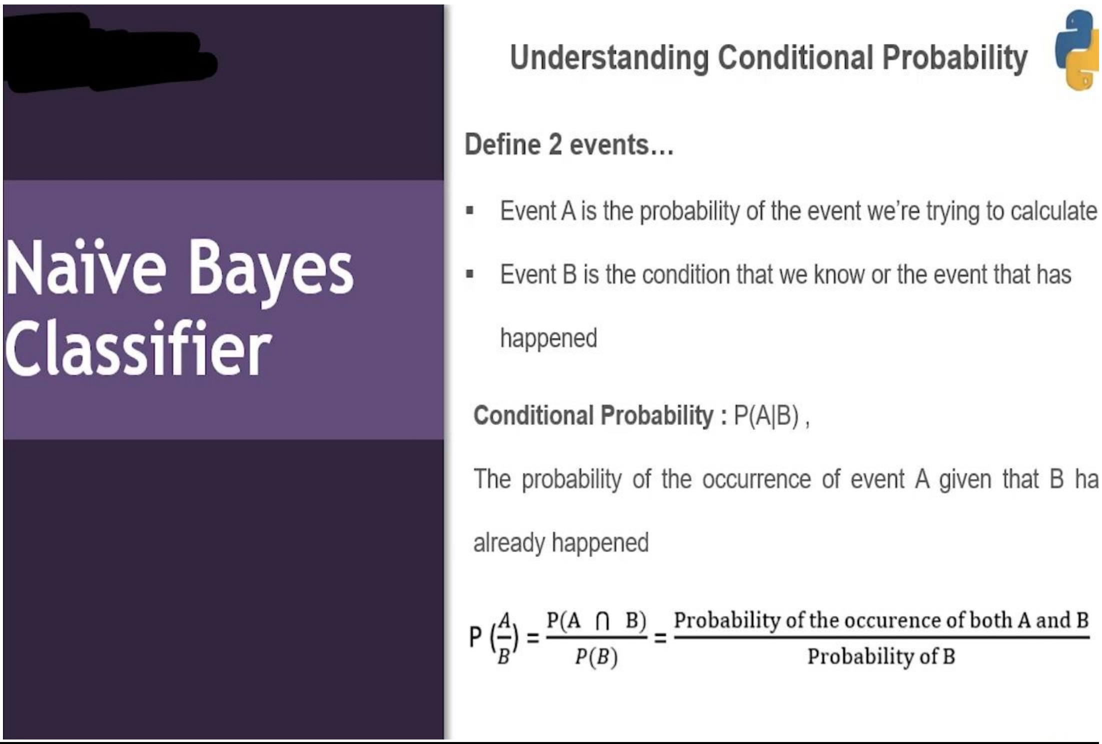


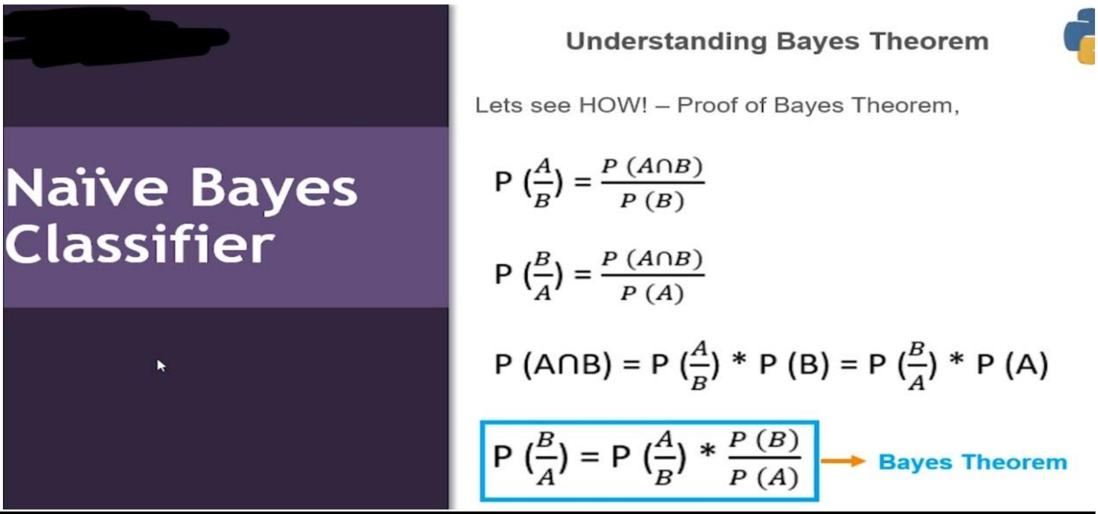
Random Forest Is Extension Of Bagging Where Selection Of Feature Columns At Each Split Is Randomly Choosen



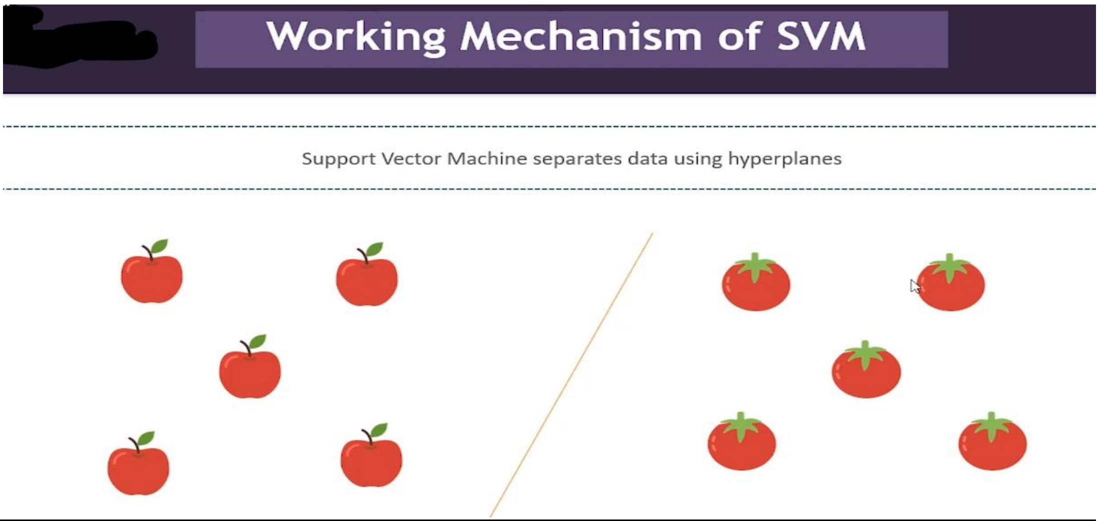
The term estimators are number of decision trees used in random forest model, where random forest is aggregate of decision for all decision trees with features choosen randomly at each split.

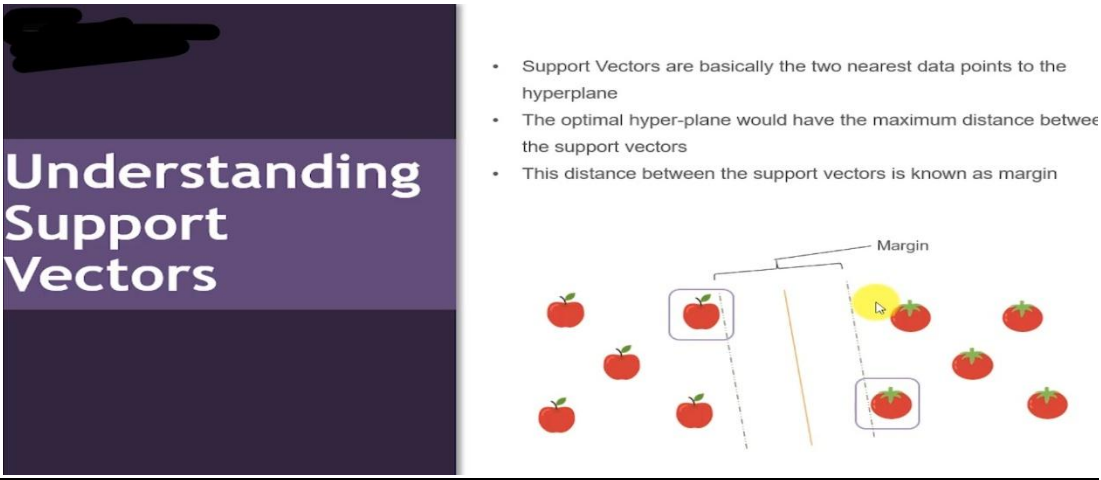
Theory and Technique of Naïve Bayes

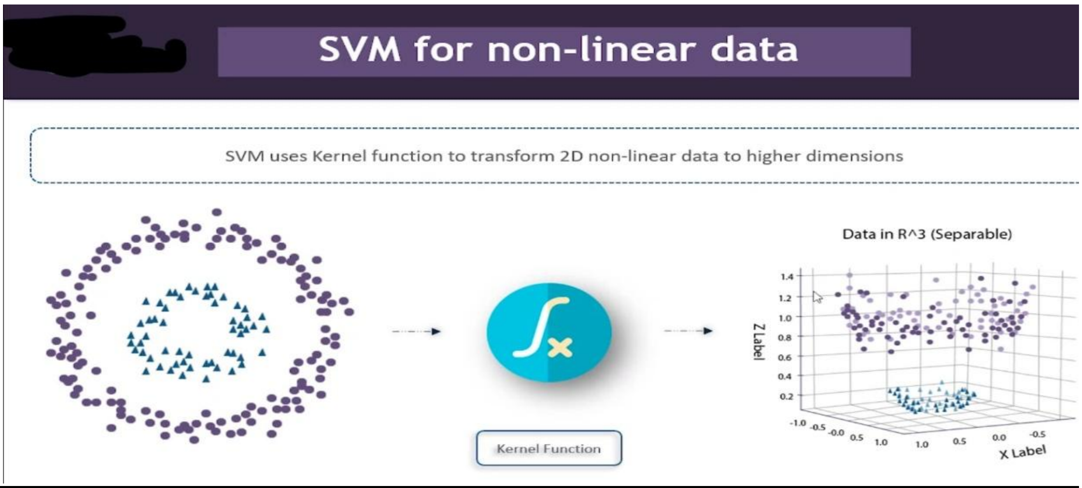


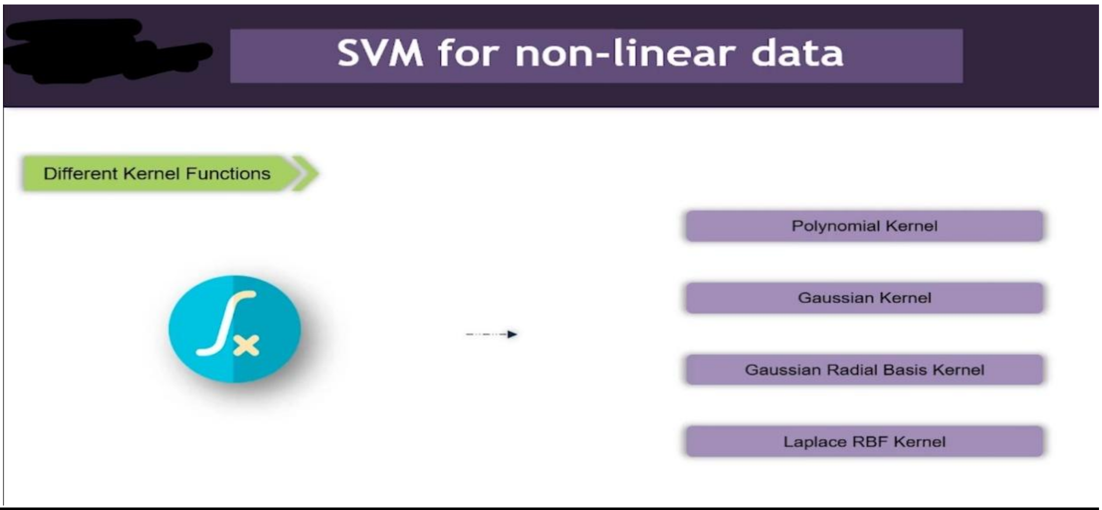


Theory and Technique of Support Vector Machine

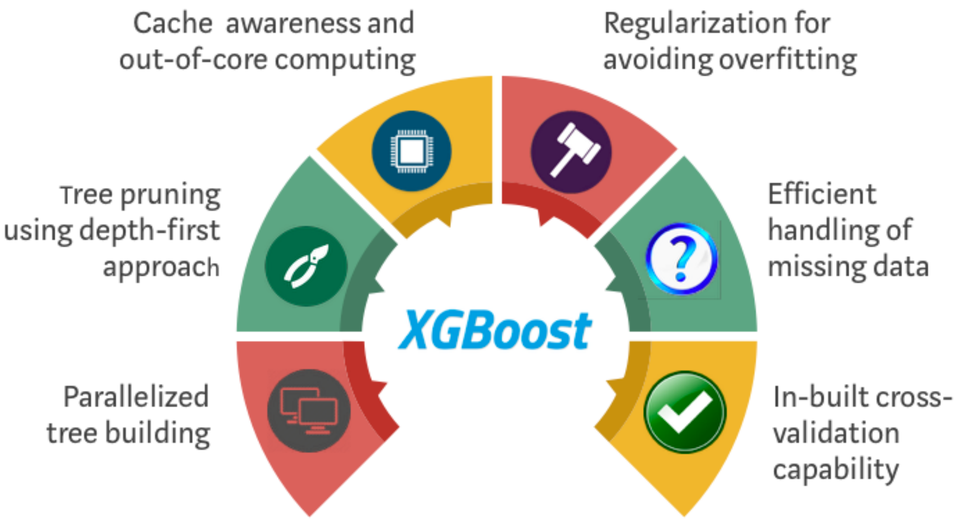








Concept of Xgboost Technique



Hyper Parameter Tuning (Grid Search Cv In Random Forest Classifier)

In this paper, lastly we used hyper parameter tuning in random forest classifier to get more accurate result. The hyper parameter used here is “Grid Search CV”.

RESEARCH GAP

This procedure is based on analytical study ,simultaneously by most of machine learning models under supervised learning with a comparison between these as well as obtaining updated score card ,to observe performance report at each step which is at a time getting output as well as cross verification .Generally this is somehow different from conventional machine learning technique by a particular suitable model evaluation .Moreover dealing with historical big data also helped to understand the trend for prediction of output .

Research Questions/Hypothesis

During building each model, using train-test split formula ,fit the train data and while obtaining prediction for test data , we assumed that the predicted weather will be clear, which is null hypothesis .After obtaining result by each model, if we could not prove the alternative, then we could not reject the null hypothesis. In this case always we failed to reject the null hypothesis.

Methods

- **Importing Adequate Machine Learning Libraries under Machine Learning Scikit Learn**

Imported numpy, panda, seaborn, matplotlib, followed by extraction of big csv data,two csv files as obtained for parameters as on TAB2,TAB3 consisting with surface observational data ,collected from online data collection platform of IMD PUNE. Merged these two csv files into one with most relevant observational data necessary for analysis, then uploaded this file on the content folder of google collab and extracted this by python code.

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.

Figure 1: TAB2,TAB3

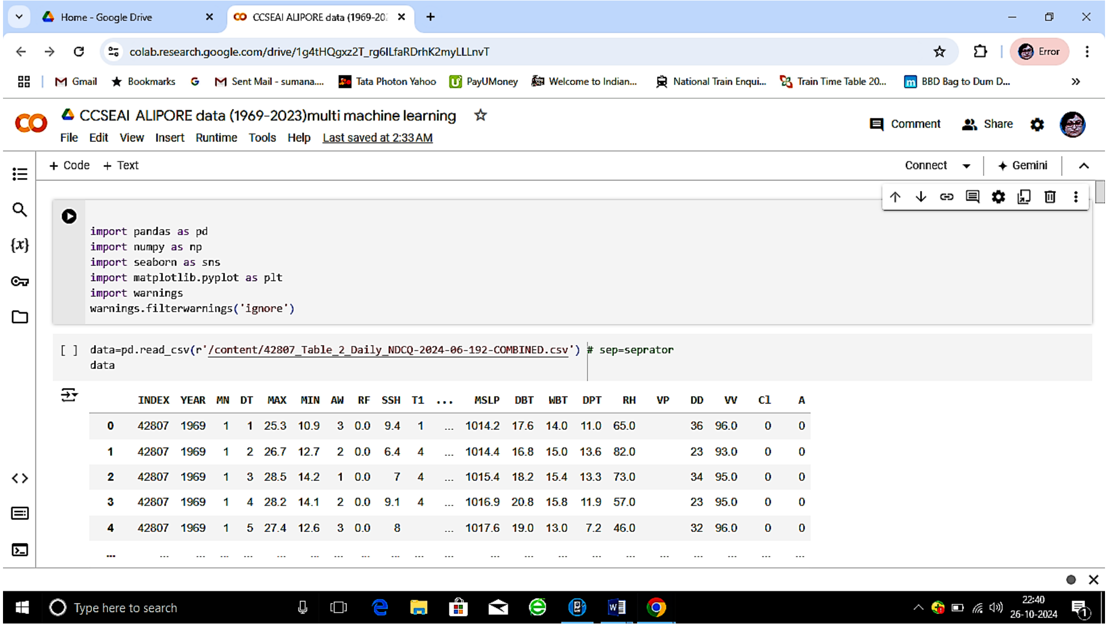


Figure 2: Import library and upload csv weather data

Understanding Data and Feature Engineering

Understanding shape of data, merged year, month, date column as well as made the date column into date-time format ,checked whether there was any null values or duplicate values to avoid bias in output. Deleted null and duplicate records .Encoded a new dependent output variable 'T',which represented the weather event ,significant as '1' and clear as '0'. Dropped the original weather columns. Understanding type of data. Listing of columns, Understanding statistics of data, value count of 'T' variable.

Figure 3: Understanding shape of data, formatting date field, drop duplicate records

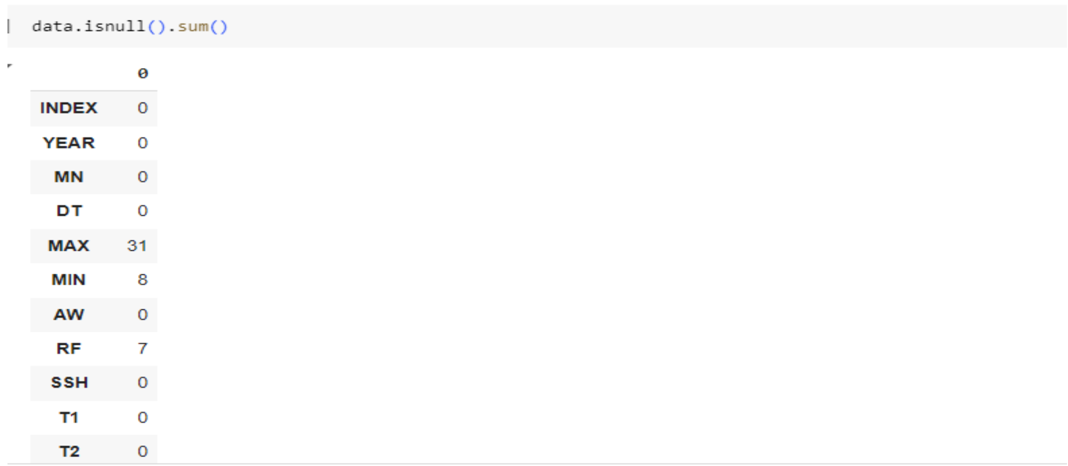


Figure 4: Checking null values

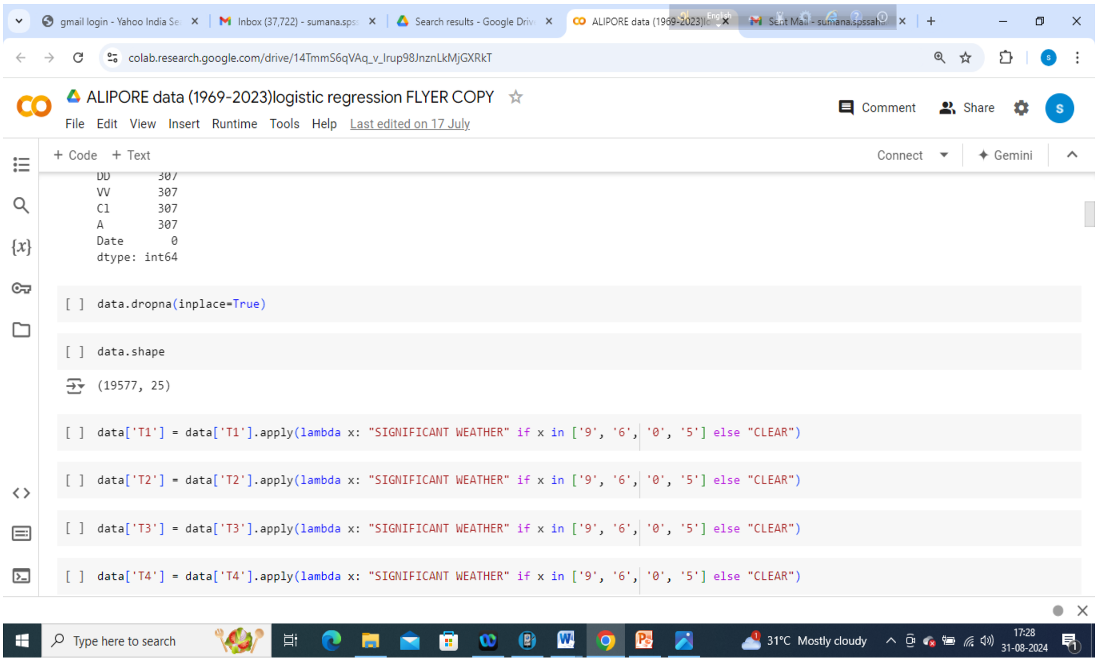
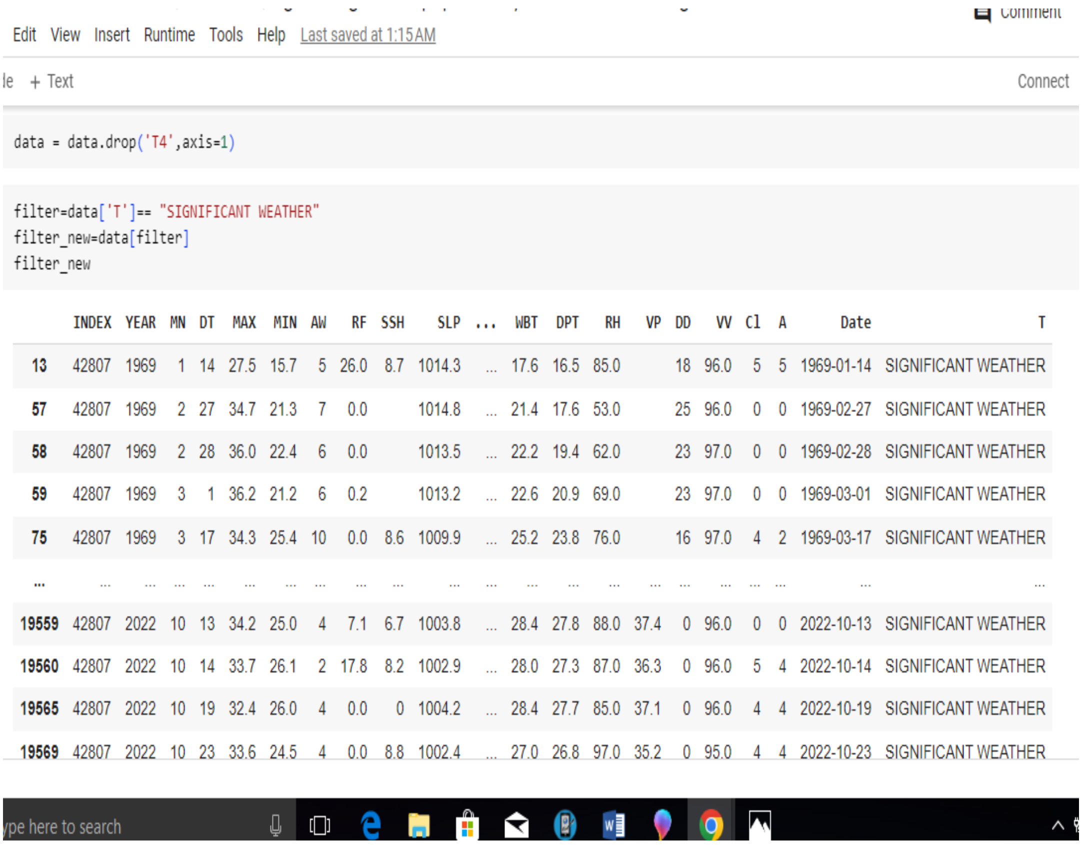


Figure 5: Encoding weather variable by lambda

Figure 6: Encoding new weather variable 'T' by lambda and drop original weather



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

Comment

+ 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

ype here to search

Figure 7: Filtering new weather variable 'T' for significant weather

```
[ ] 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']
```

Figure 8: List columns

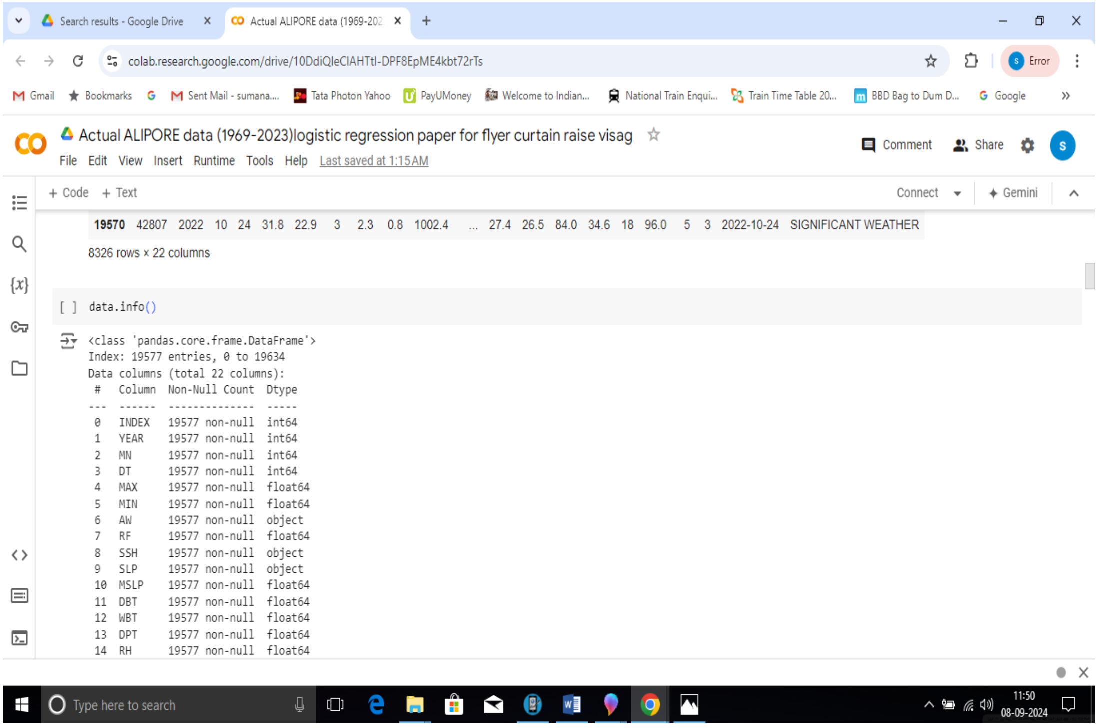
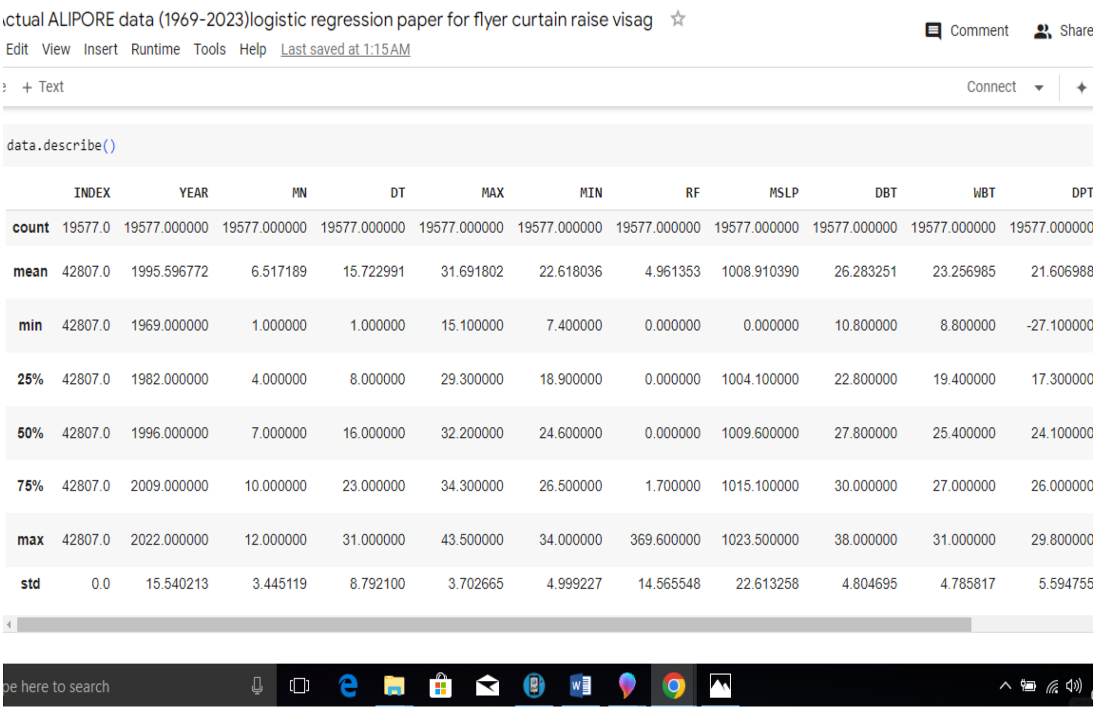


Figure 9: Information of data required for analysis and label encoding



	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

Figure 10: data describe to understand statistical significance of the data

Figure 11: value count of how many records having significant weather and how many clear

IV Method

Information value method to find out the score of each object variable and to discard that object type variables with iv score less than 0.02. Thus obtained the columns with high scores ,and then these subjected to analysis . The range is as follows : less than (<) **0.02 is 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.** In our case we after IV analysis ,discarded the column "DD" and 'A' where 'DD' is wind direction and 'A' is amount of low cloud in octas.

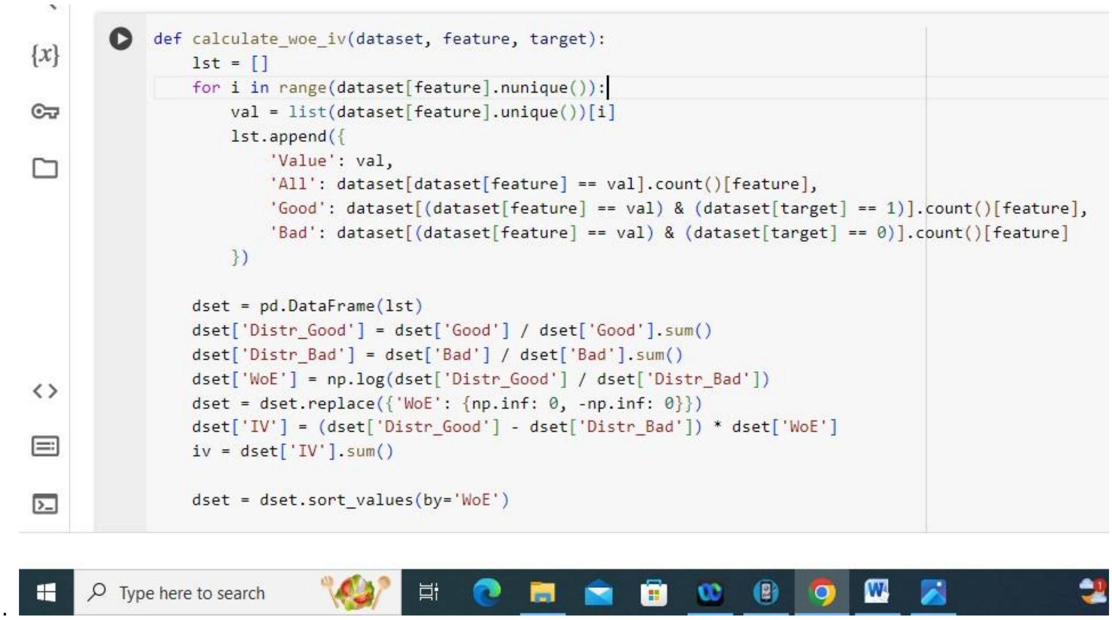


Figure 12: IV method

```
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

Figure 13: IV score

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

Code + Text

4	DD	0.043778
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
```

Figure 14: Drop the columns with very low IV scores

One Hot Encoding Method and Label Encoding and VIF Method

Now started method one hot encoding and label encoding to convert the string type variable as numeric for compatibility of execution. Then VIF method to remove multi collinearity factor.

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

Figure 15: One hot encoding

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])
```

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

Figure 16: Label encoding

As date and time does not have any role now, already duplicate record have been deleted ,these columns can be omitted now .VIF process done repeatedly to remove multi collinearity factor with dropping of column having highest VIF value after each execution. Ultimately obtained the columns with VIF value 6 or less than that.

```
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

Figure 17: Drop date, index, year, month,dt column and VIF method

```
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

Figure 18: VIF last level score

Machine Learning Models

In this section starting of different machine learning models to predict probable weather with the same process of train-test split, fit the model for training set, input value from outside in test data and predict output weather for this particular model.

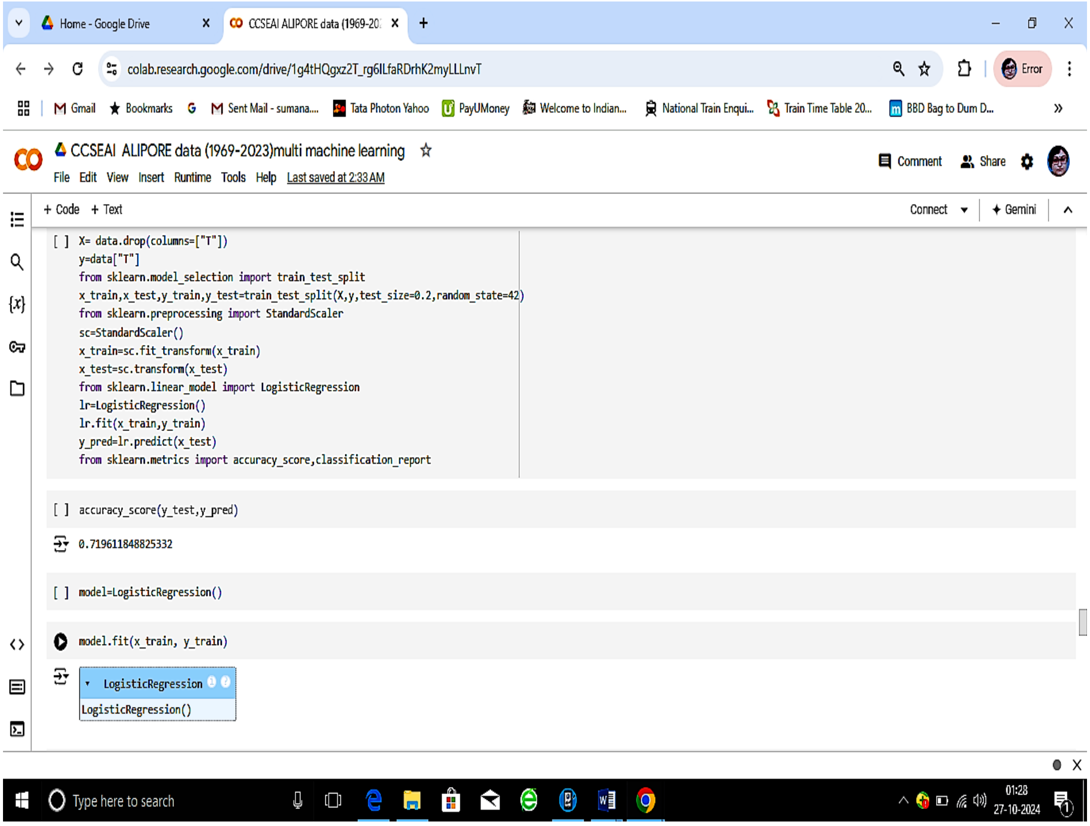


Figure 19: Machine Learning by Logistic Model

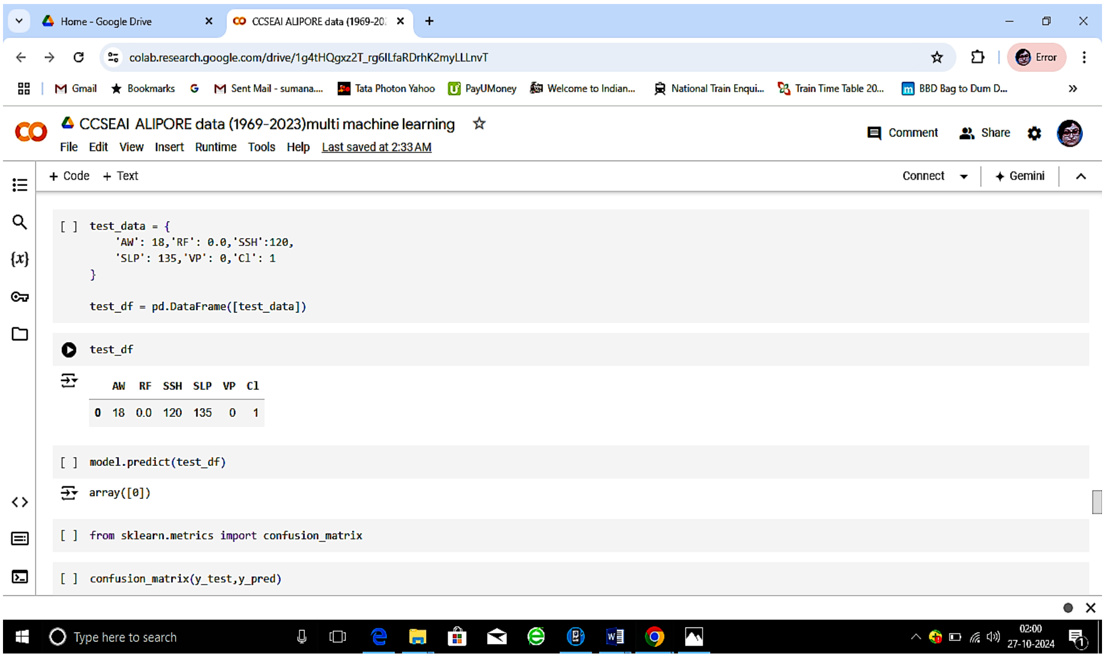


Figure 20: Input data for testing and prediction as well as check success rate of model

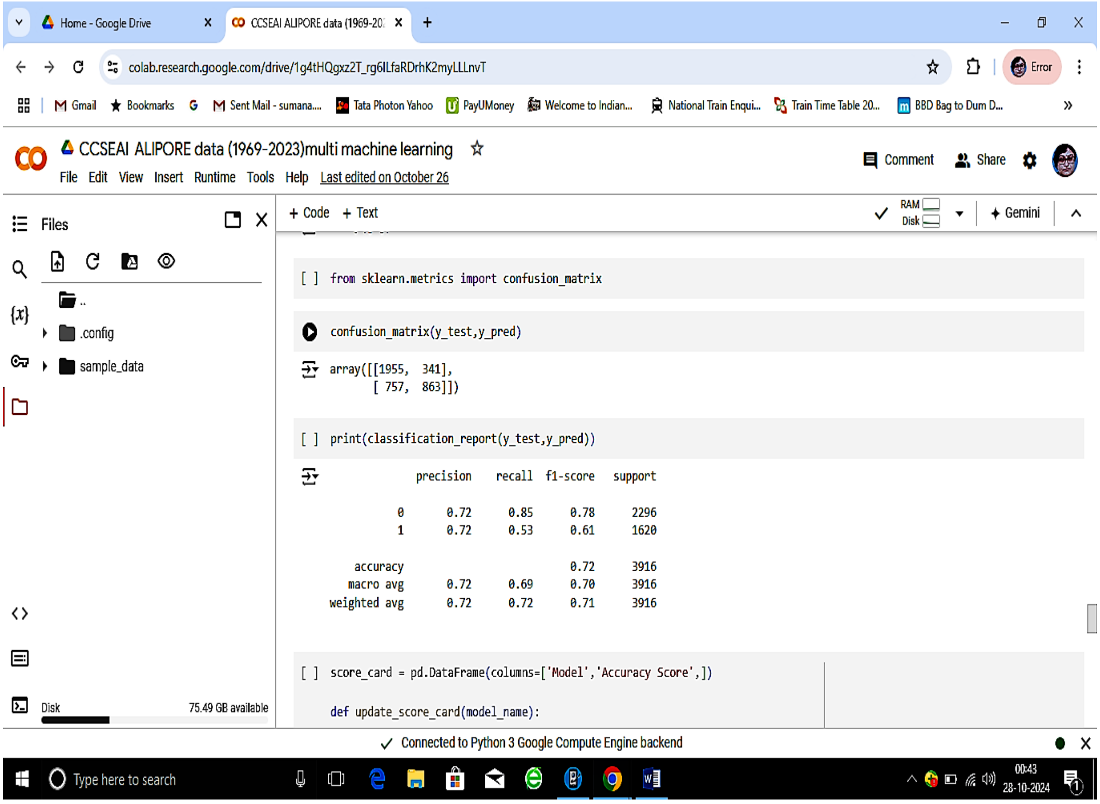


Figure 21: Checking of classification report, confusion matrix

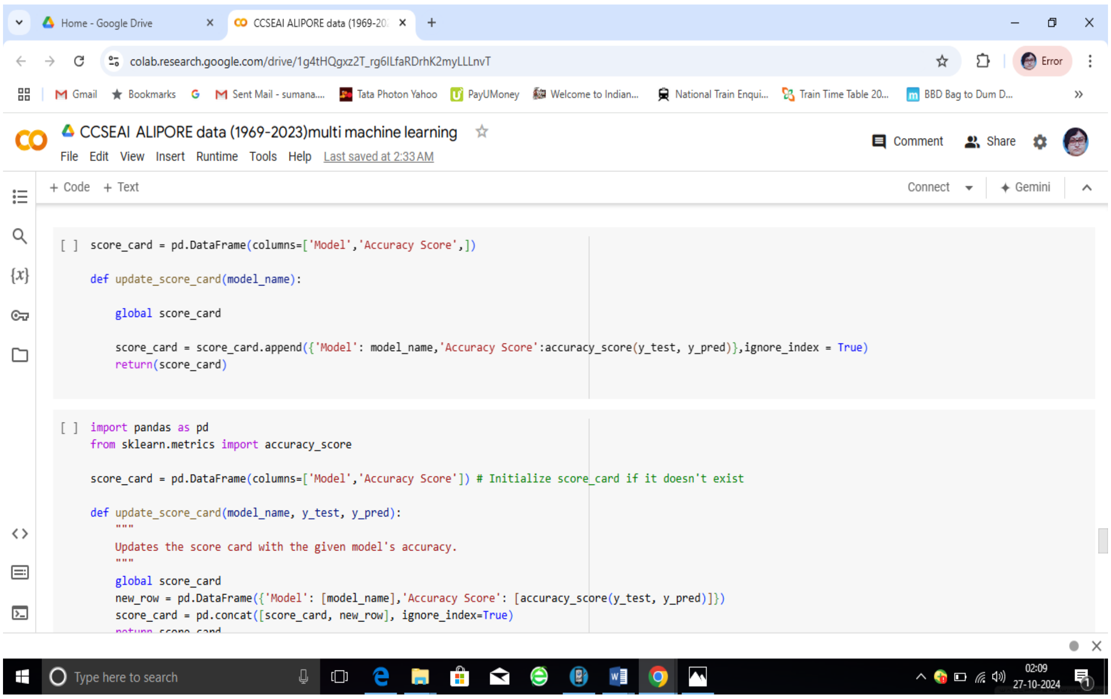


Figure 22: Score card for logistic regression

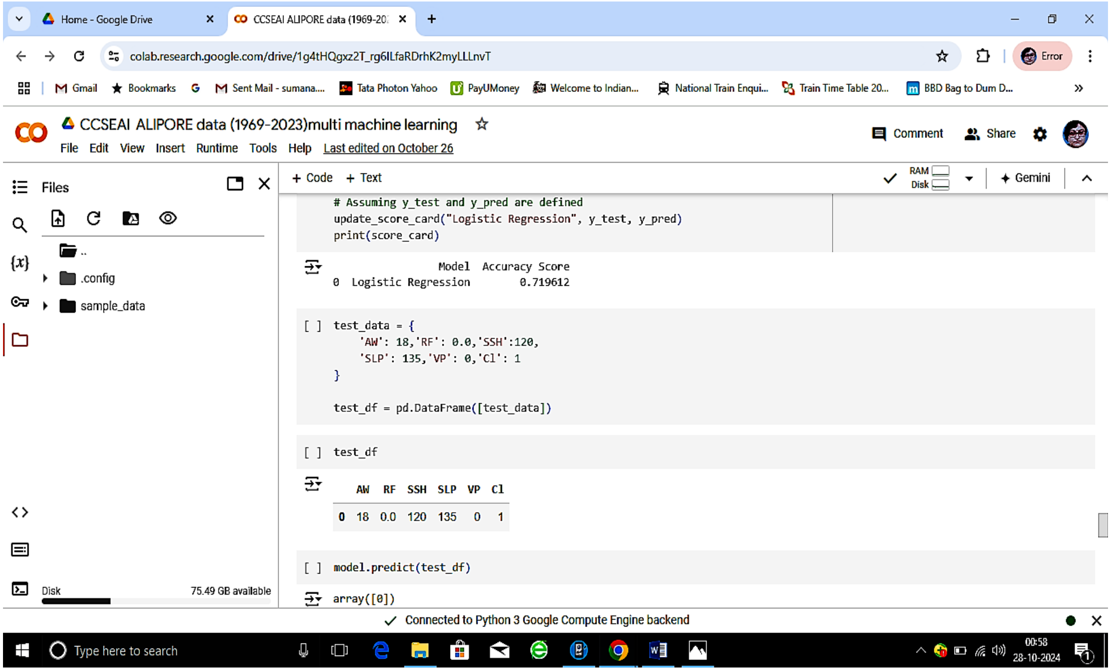


Figure 23: Update score card for logistic regression, input test data and predict weather for logistic regression model

Same procedure as for logistic model, continued for each model, one by one along with update score card for execution of each model. After logistic regression, we started execution by decision tree classifier with the same method, input test data, update score card for decision tree and predict weather for this model of decision tree classifier.

Decision Tree Classifier

Same procedure as for logistic model, continued for each model, one by one along with update score card for execution of each model.

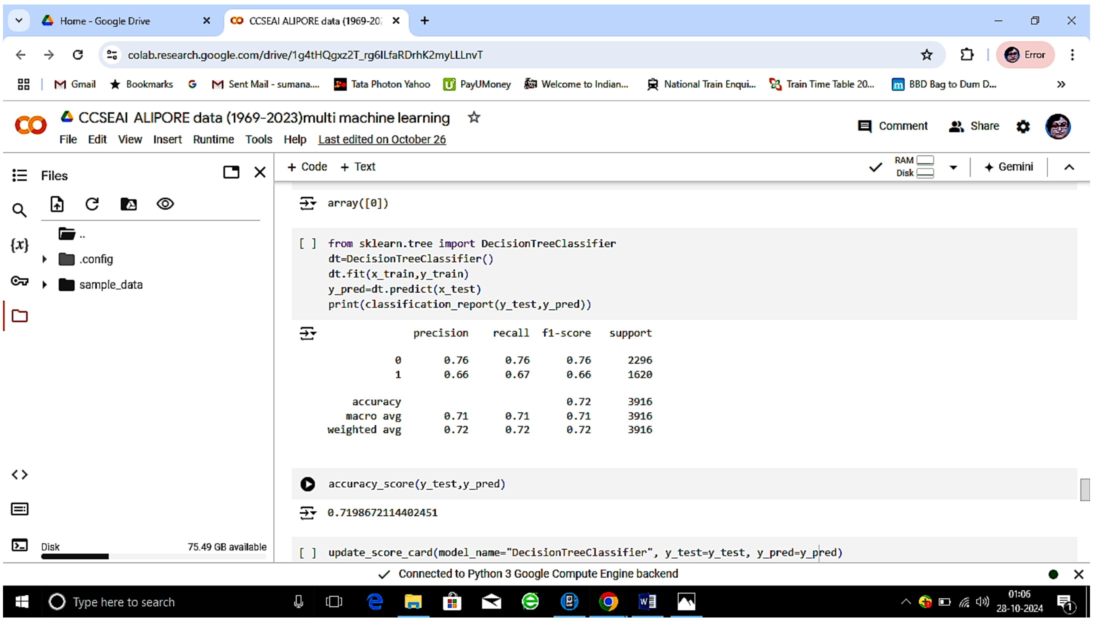


Figure 24: Import decision tree classifier, fit train data, print classification report, accuracy score

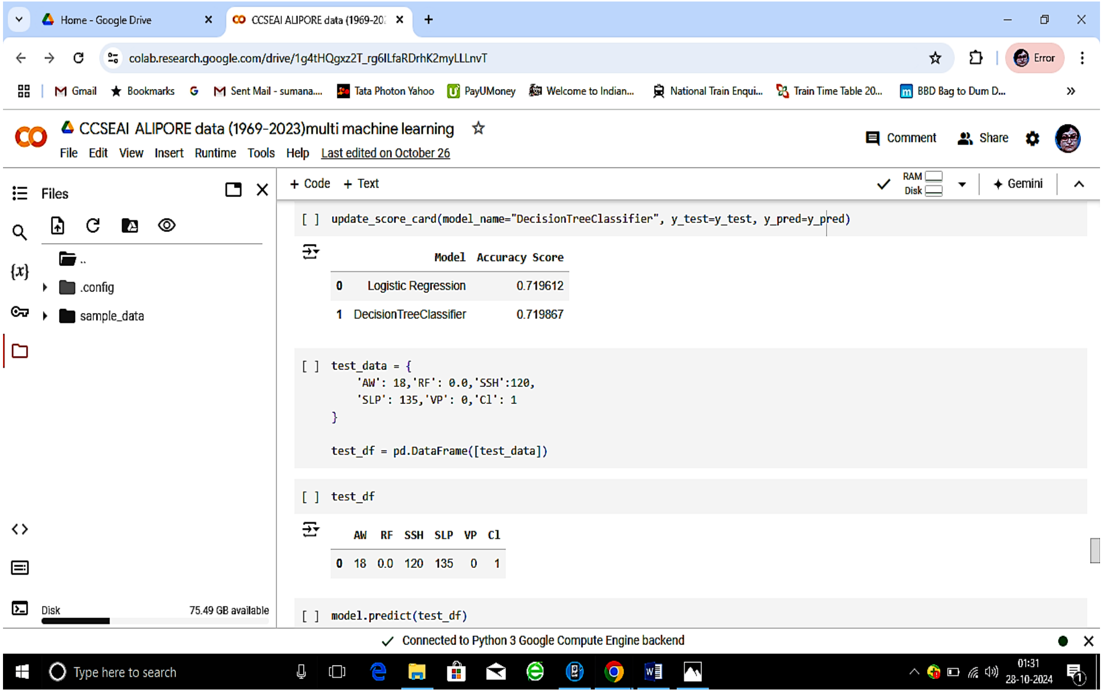


Figure 25: Update score card after execution of decision tree model with input test data and predict weather for this model

Random Forest Classifier

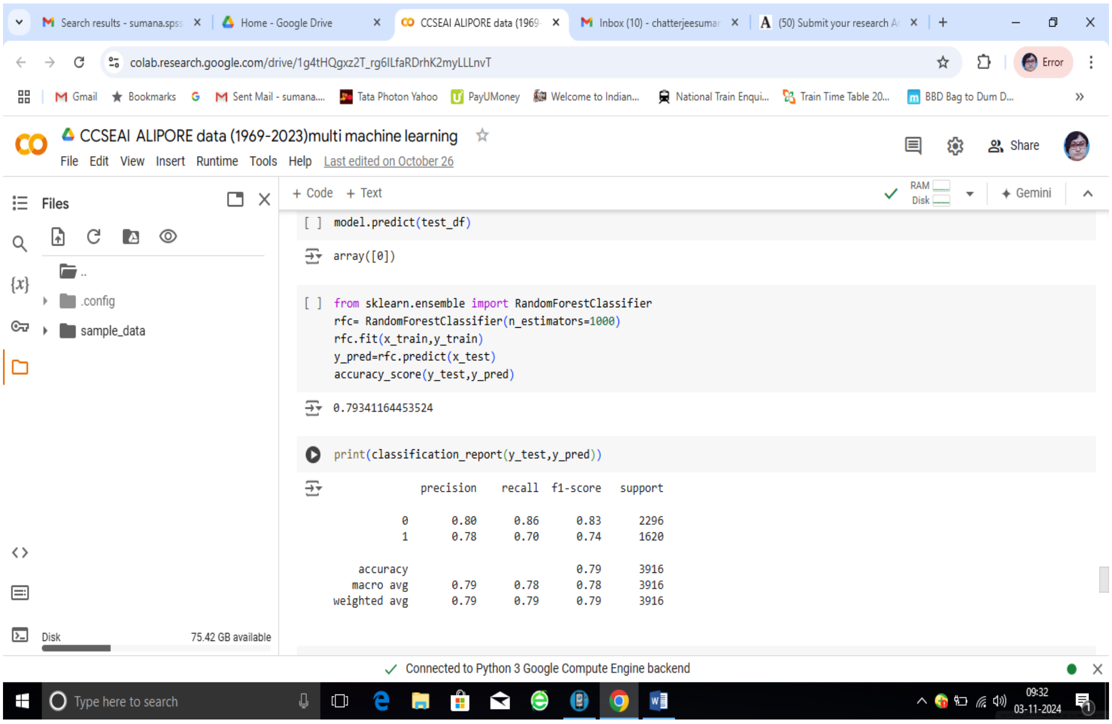


Figure 26: Import Random forest classifier, fit train data, print classification report, accuracy score

Update Score Card By Random Forest Classifier, Input Test Data And Predict Weather After Execution Of Random Forest Classifier

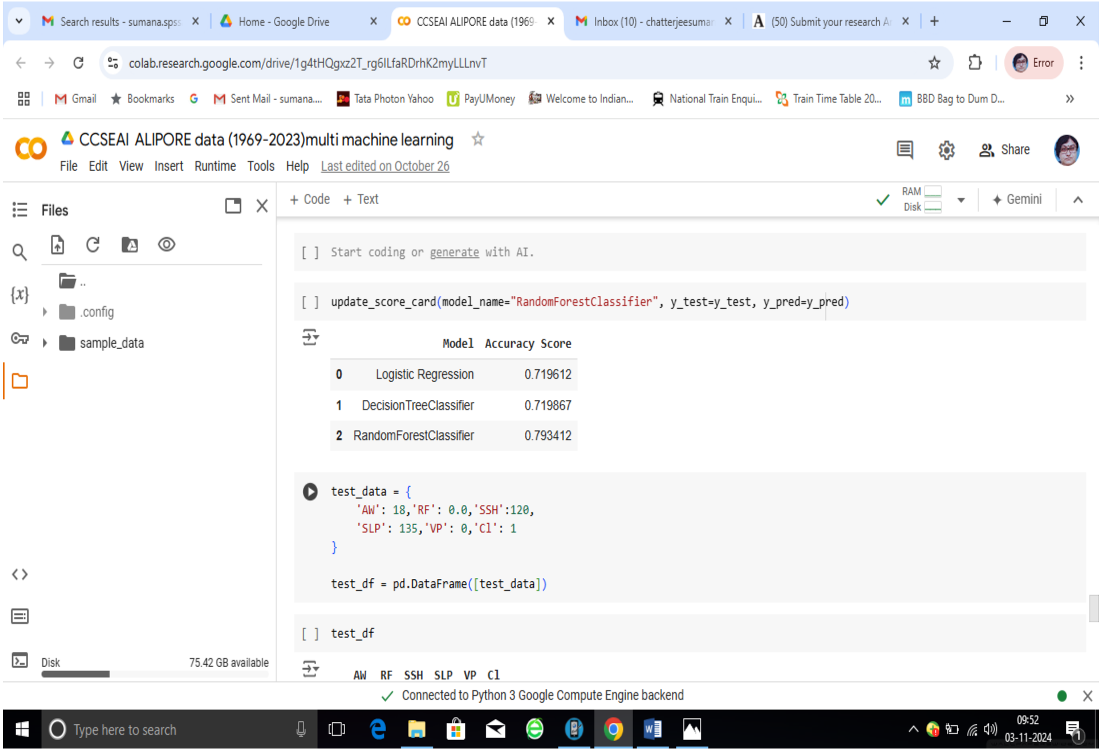


Figure 27: Update score card after execution of Random forest classifier model with input test data and predict weather for this model

After Prediction Probable Weather by Random Forest Model ,Import Model Naïve Bayes

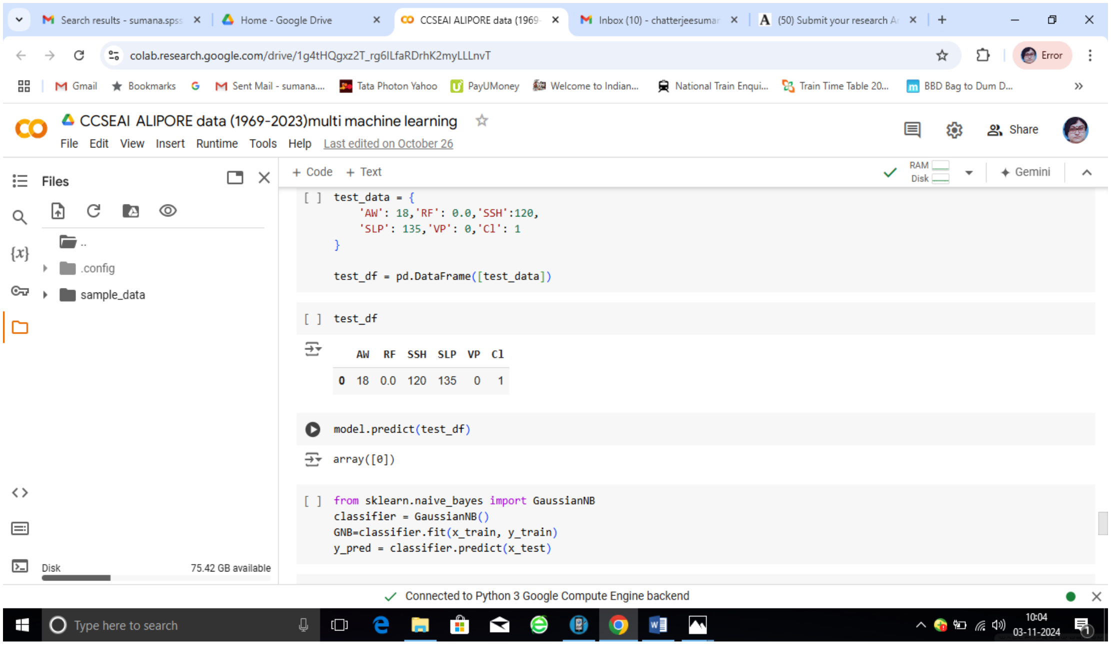


Figure 28: Import Naïve Bayes, fit the train data

Find Accuracy Score, Classification Report, Update Score Card with Naïve Bayes

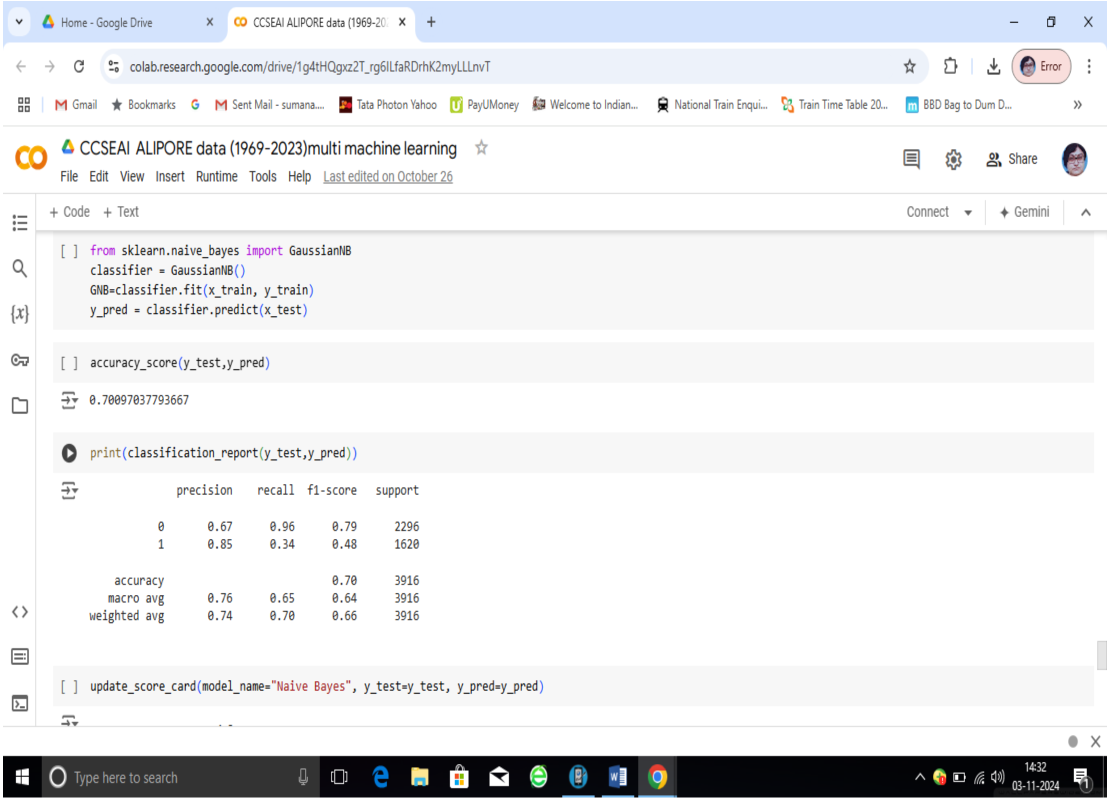


Figure 29: Find accuracy score, classification report with respect to Naïve Bayes

Input Test Data and Predict Weather by Naïve Bayes

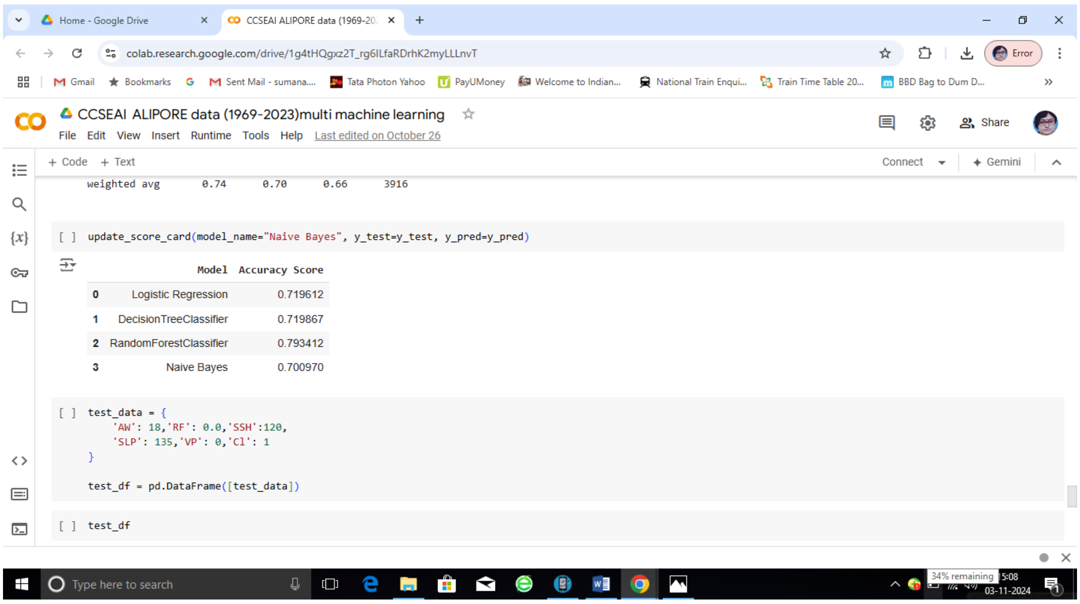


Figure 30: Input test data for prediction on test data for Naïve Bayes as well as print updated score card with respect to Naïve Bayes

Import support vector machine, find accuracy sore,print classification report ,update score card with support vector machine(Svm)

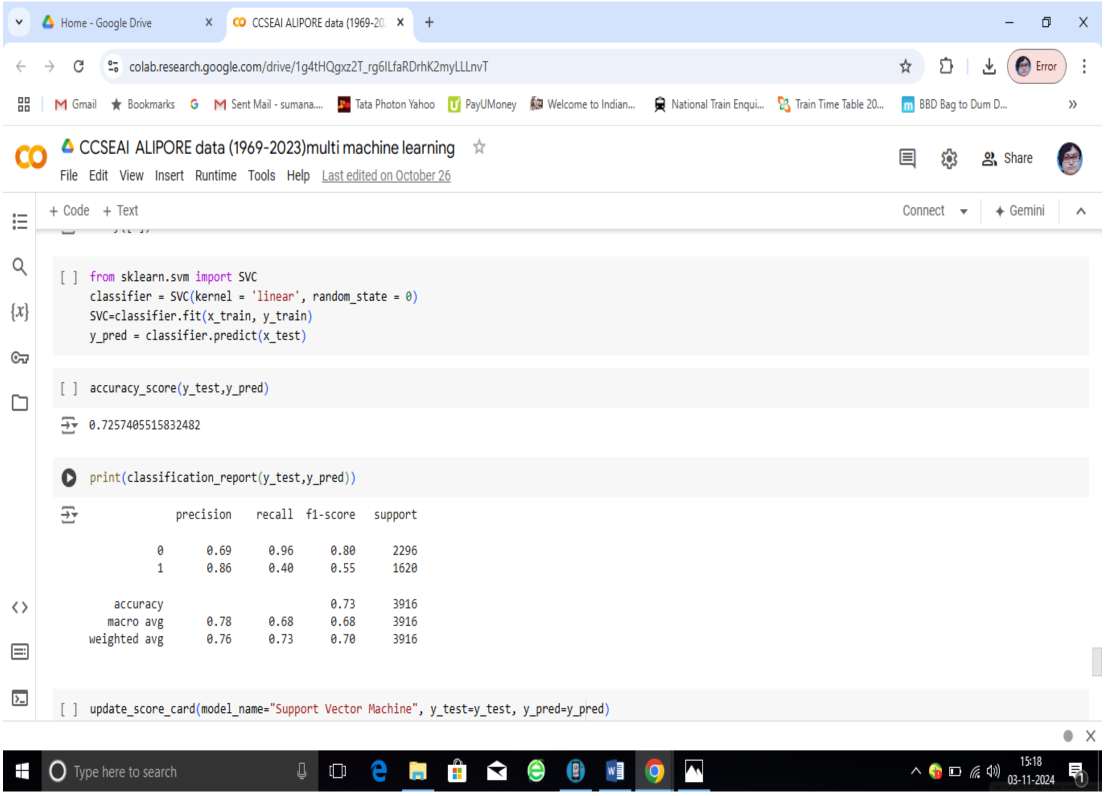


Figure 31: Import Support vector, find accuracy score, classification report and update score card

Input Test Data and Predict Weather by SVM, Then Import Xgboost Classifier to Start Analysis

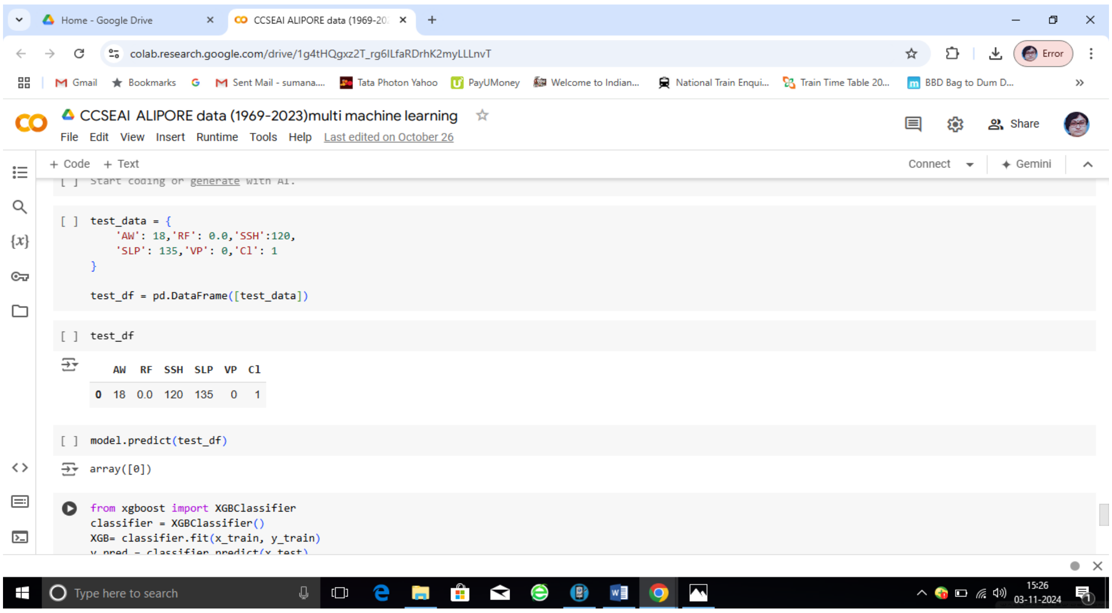


Figure 32: Input test data in support vector machine, predict model accuracy, import XGBOOST classifier

Find Accuracy Score, Classification Report for Xgboost and Update Score Card

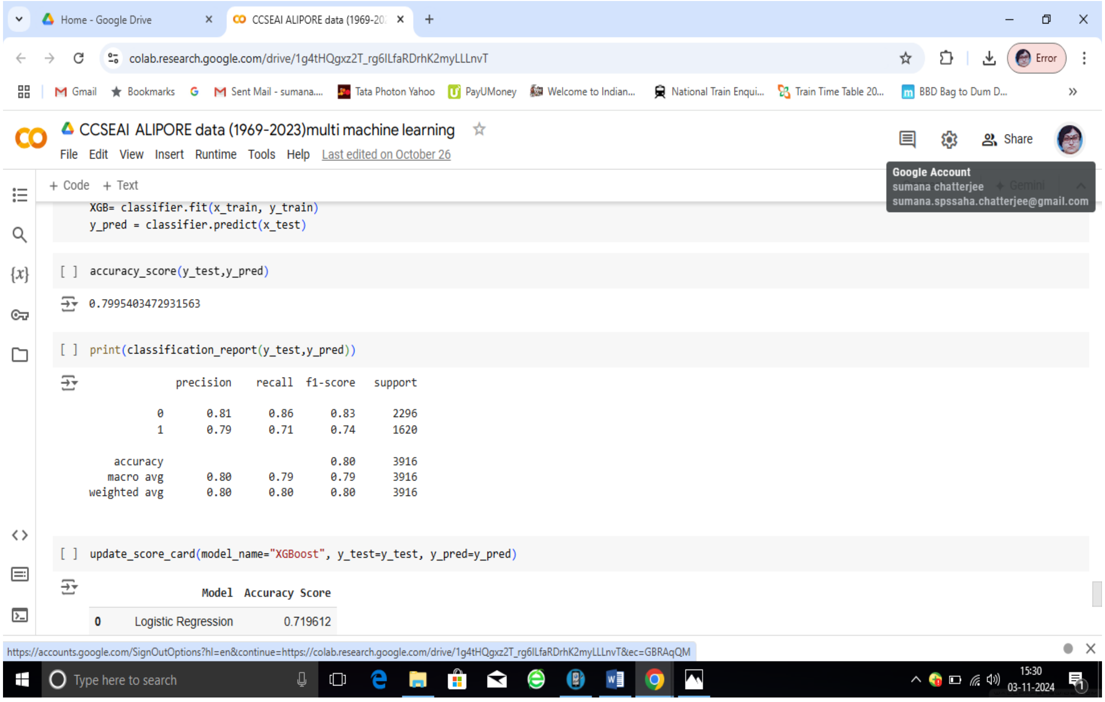


Figure 33: Predict model accuracy, print classification report , update score card for SVM

Hyperparameter Tuning on Random Classifier using Gridsearch CV Hyperparameter

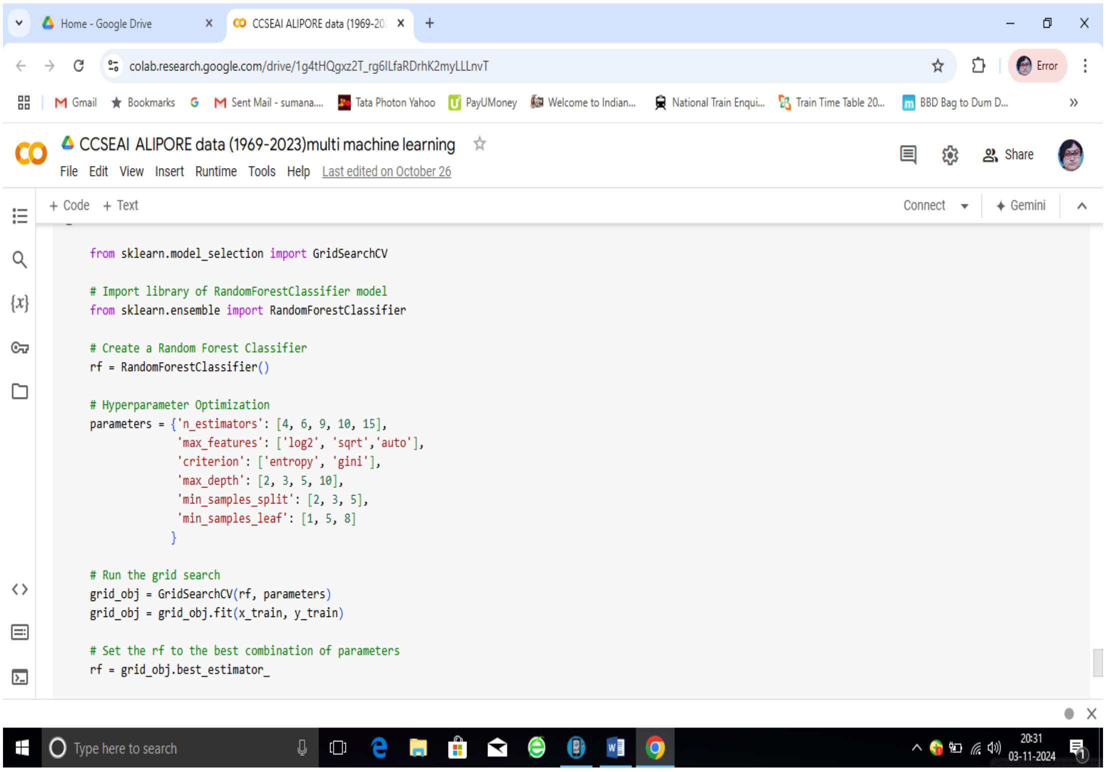


Figure 34: Hyper Parameter tuning using Grid search CV

Model Fit on Train Data Set, Find Round Off Accuracy of Model, Prediction on Test Data Set

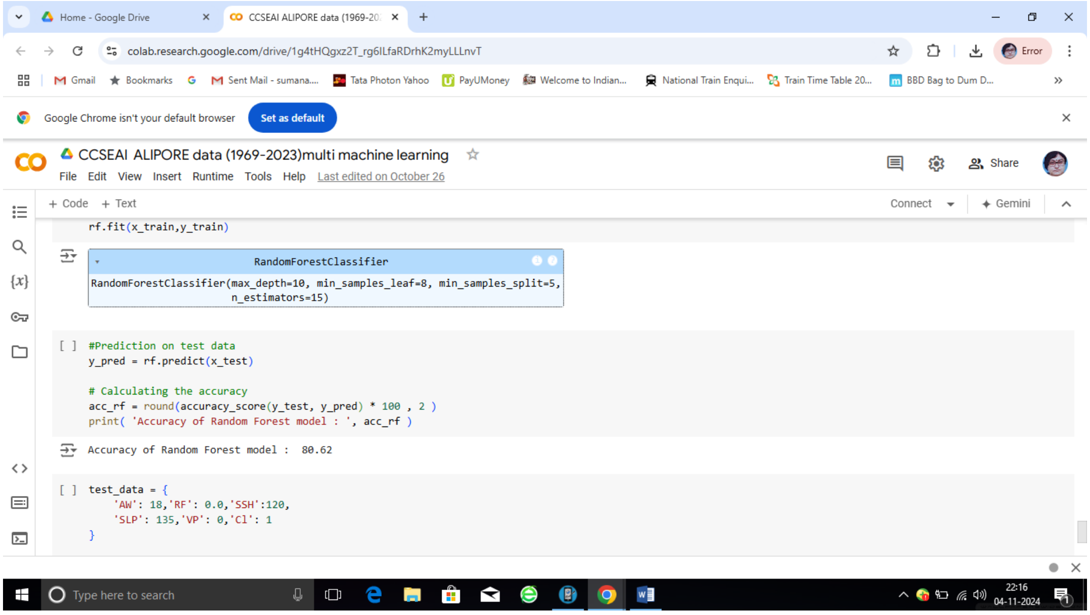


Figure 35: Train data set, print accuracy and prediction on test data on the basis of hyper parameter tuning

Update Score Card for Hyper Parameter Tuning of Random Forest

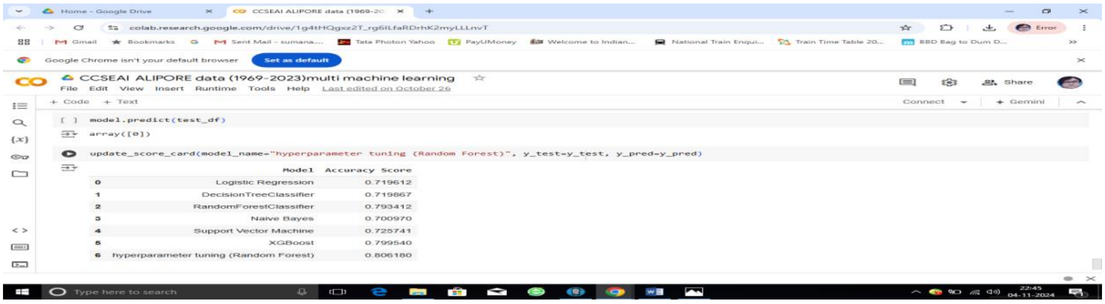


Figure 36: Predict model with random score with hyper parameter tuning and check score card

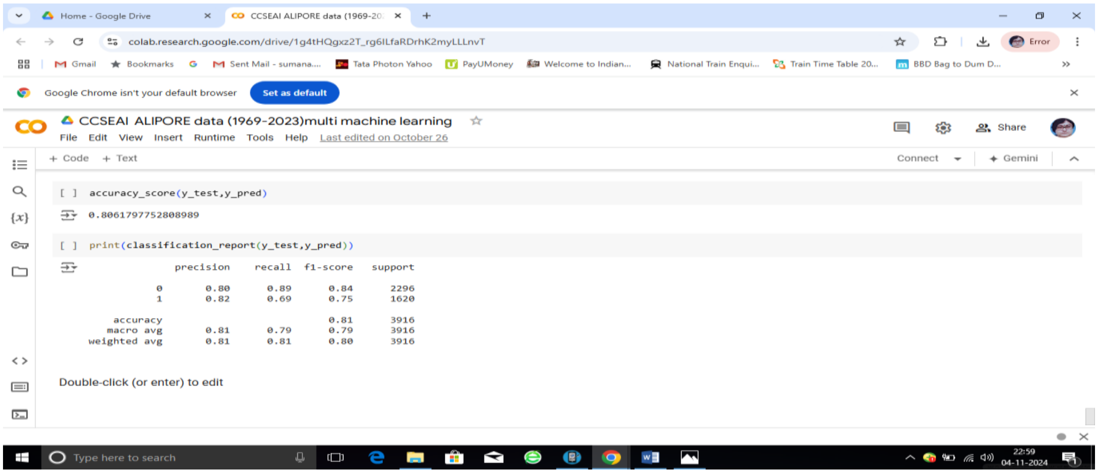


Figure 37: Print accuracy score, classification report with random forest hyper tuning

Conclusion

This paper is done by machine learning method with comparison between several supervised models. In future other ML techniques such as weather prediction by neural network etc. may be tried also.

Acknowledgements

I hereby would like to thank everyone, just everyone!

References

1. Collection of data from "*Data supply portal India Meteorological Department, Pune*, https://dsp.imdpune.gov.in/"
2. Study material of INTELLIPAAT along with python code
3. https://www.linkedin.com/pulse/xgboost-classifier-algorithm-machine-learning-kavya-kumar/ june17/2021.