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

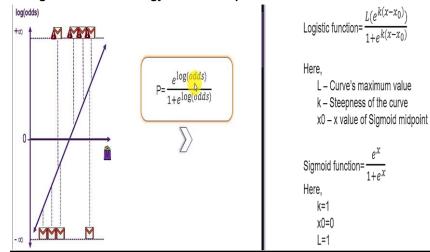
The theory and technique applied for different models are described herewith.

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



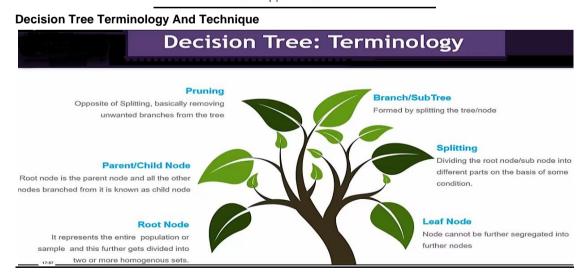
Sigmoid Function:

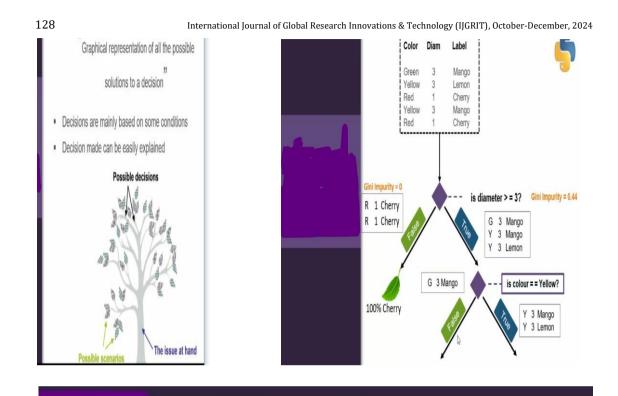
Sigmoid function= $\frac{e^x}{1+e^x}$ 

- S' shaped curve.
- Sigmoid curve has a finite limit of:

'0' as x approaches -∞

'1' as x approaches +∞





# 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

#### **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

2

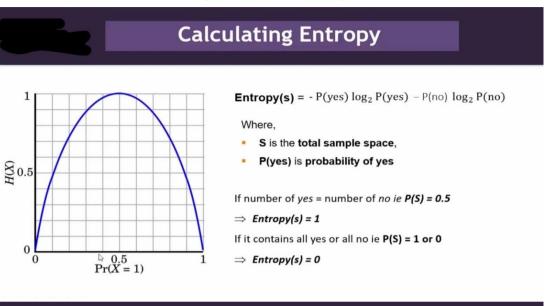


#### Information Gain

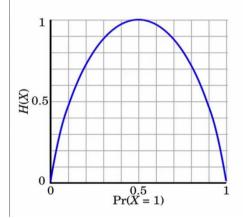
The information gain is the decrease in entropy after a dataset is split on the basis of an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain

#### **Gini Index**

The measure of impurity (or purity) used in building decision tree in CART is Gini Index

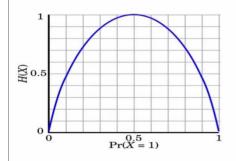


## **Calculating Entropy**



 $E(S) = -P(Yes) \log_2 P(Yes) - P(no) \log_2 P(no)$ 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

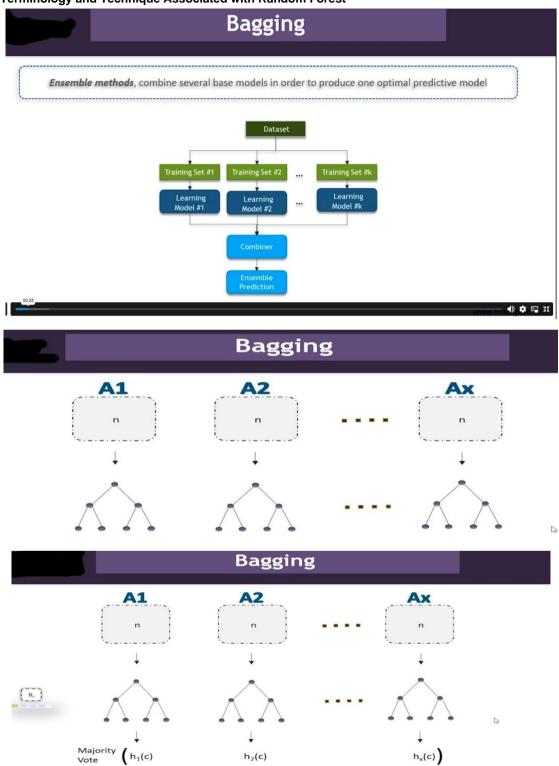
## **Calculating Entropy**



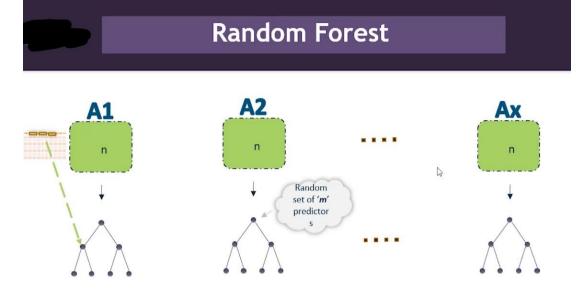
 $E(S) = -P(Yes) \log_2 P(Yes)$ When P(Yes) = 1 ie YES = Total Sample(S)  $E(S) = 1 \log_2 1$ E(S) = 0

 $E(S) = -P(No) \log_2 P(No)$ When P(No) = 1 ie No = Total Sample(S)  $E(S) = 1 \log_2 1$ E(S) = 0

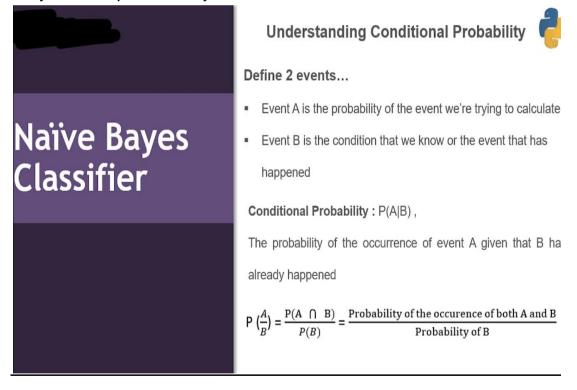
#### Terminology and Technique Associated with Random Forest

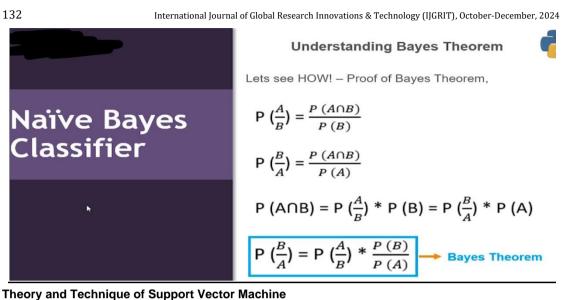


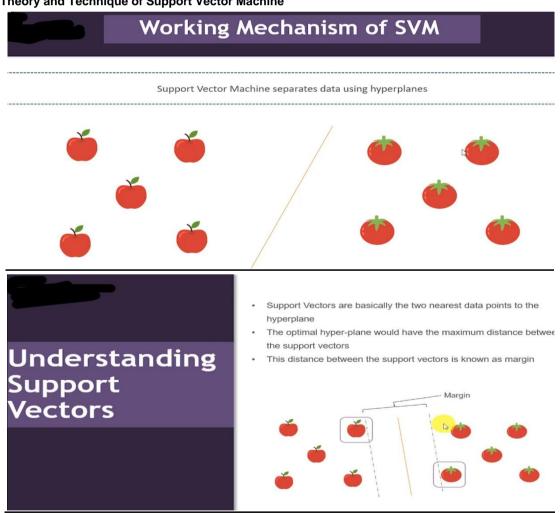
Random Forest Is Extension Of Bagging Where Selection Of Feature Columns At Each Split Is Randomly Choosen

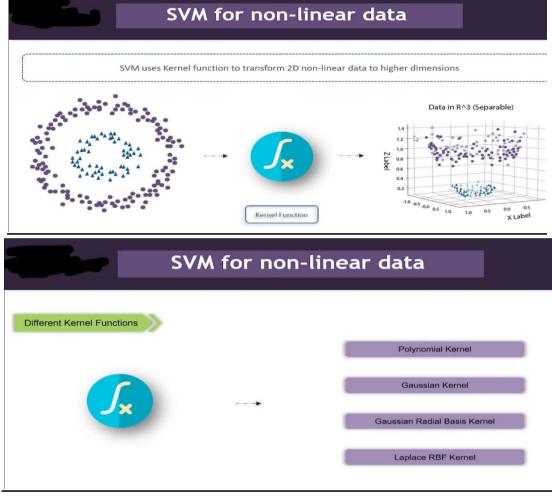


The term estimators are number of decision trees used in random forest model, where random forest is aggregate of decision for all decision trees with features choosen randomly at each split. **Theory and Technique of Naïve Bayes** 

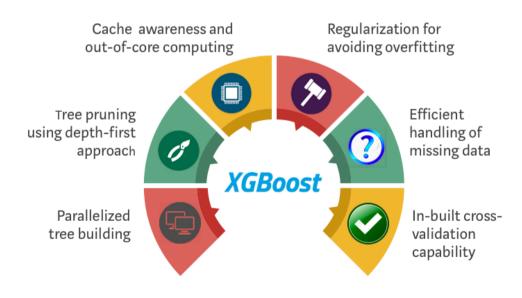








## Concept of Xgboost Technique



#### 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**

Type of weather in code

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

т

#### 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.

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	к	Dust fog
L	Line squall	м	Ground frost	N	Dew	0	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

		colab.res	-	2						·														☆			😁 Ern	ж
M Gm	nail ★	Bookmarks	G	M S	ent Ma	ail - su	mana	<u>*</u>	Tata	Photor	n Yahoo	· 0	PayUMor	ney 着	B Welco	ome to l	ndian	₿N	ationa	l Train En	qui S	3 Train Time Table	20	m BBD B	ag to Du	im D		×
		I ALIPO											☆										E Co	mment	2	Share	٥	1
+ Code	е + Те	xt																						Connect	•	+	Gemini	
																							$\uparrow$	↓œ		۵	. 0	J :
0	import	pandas	ac ad																				_					
	import	numpy a	as np																									
	imnort	seaborr	n as sr			-14																						
			tlib.py	plot	as	prt																						
	import import	matplot warning	gs				、 、																					
	import import	matplo1	gs				)																					
	import import warnin data=p	matplot warning	gs erwarn:	ings(	'ign	iore')		.e_2_	Dail	y_NDC	Q-20	24-06	i-192-CO	MBINE	D.csv'	) # s	ep=sepi	rator										
[]	import import warnin	matplot warning gs.filte	gs erwarn:	ings(	'ign	iore')		.e_2_	Dail	y_NDC	Q-28	24-06	-192-CO	MBINE	D.csv'	) # s	ep=sepi	rator										
	import import warnin data=p	matplot warning gs.filte	gs erwarni csv(r'į	ings( /cont	'ign ent/	iore') '42807	7_Tabl						i-192-CO MSLP	MBINE	<u>D.csv</u> ' WBT	) # 5 DPT	ep=sepi RH	rator VP	DD	vv	<b>c</b> 1	A						
[]	import import warnin data=p	matplot warning gs.filte d.read_c	gs erwarni csv(r'į	ings( /cont MN	'ign ent/ DT	iore') '42807	7_Tabl MIN	AW	RF					DBT	WBT	DPT	RH			<b>VV</b> 96.0	<b>C1</b> 0	A 0						
[]	import import warnin data=p data	matplot warning gs.filte d.read_c INDEX	gs erwarn: csv(r' <u>/</u> YEAR	ings( /cont MN	'ign ent/ DT 1	юге') (42807 МАХ	7_Tabl MIN 10.9	AW 3	RF 0.0	SSH 9.4	<b>T1</b> 1	••••	MSLP	DBT 17.6	WBT 14.0	DPT 11.0	RH 65.0		36									
[]	import import warnin data=p data 0	matplot warning gs.filte d.read_c INDEX 42807	gs erwarni csv(r' <u>/</u> YEAR 1969	ings( /cont MN 1	'ign ent/ DT 1 2	(42807 (42807 (MAX) (25.3) (26.7)	7_Tabl MIN 10.9 12.7	AW 3	RF 0.0 0.0	SSH 9.4 6.4	<b>T1</b> 1	···· 	MSLP 1014.2	DBT 17.6 16.8	WBT 14.0 15.0	DPT 11.0 13.6	RH 65.0 82.0		36 23	96.0	0	0						
[]	import import warnin data=p data 0 1	matplot warning gs.filte d.read_c INDEX 42807 42807	gs erwarni csv(r' <u>/</u> YEAR 1969 1969	ings( /cont MN 1 1	'ign ent/ DT 1 2 3	(42807 (42807 (MAX) (25.3) (26.7)	7_Tabl MIN 10.9 12.7 14.2	AW 3 2 1	RF 0.0 0.0	SSH 9.4 6.4	T1 1 4 4	••••  	MSLP 1014.2 1014.4	DBT 17.6 16.8 18.2	WBT 14.0 15.0 15.4	DPT 11.0 13.6 13.3	RH 65.0 82.0 73.0		36 23 34	96.0 93.0	0	0						
[]	import import warnin data=p data 0 1 2	matplot warning gs.filte d.read_c INDEX 42807 42807 42807	gs erwanni csv(r' <u>/</u> YEAR 1969 1969	ings( /cont MN 1 1 1	'ign ent/ DT 1 2 3 4	MAX 25.3 26.7 28.5 28.2	7_Tabl MIN 10.9 12.7 14.2	AW 3 2 1 2	RF 0.0 0.0 0.0	SSH 9.4 6.4 7	T1 1 4 4	····  	MSLP 1014.2 1014.4 1015.4	DBT 17.6 16.8 18.2 20.8	WBT 14.0 15.0 15.4 15.8	DPT 11.0 13.6 13.3	RH 65.0 82.0 73.0 57.0		36 23 34 23	96.0 93.0 95.0	0 0 0	0 0 0						

135

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.

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

Code + Text

] 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

5

**0** 1969-01-01

- 1 1969-01-02
- **2** 1969-01-03
- 3 1969-01-04

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

I	data.isn	ull <mark>()</mark> .su	um()
٢		0	
	INDEX	0	
	YEAR	0	
	MN	0	
	DT	0	
	MAX	31	
	MIN	8	
	AW	0	
	RF	7	
	SSH	0	
	T1	0	
	Т2	0	

## Figure 4: Checking null values

~ ୧	g gmail login - Yahoo India Se 🗴   M Inbox (37,722) - sumana.sp:: X   🛆 Search results - Google Driv X 🗰 ALIPORE data (1969-2023) 🕅 Sent Mail - sumana.sp:san 🖓 X   + - 🗇	×
$\leftarrow -$	🗢 😋 😂 colab.research.google.com/drive/14TmmS6qVAq_v_Irup98JnznLkMjGXRkT 🔍 🔍 🔂 🗌	:
CO	▲ ALIPORE data (1969-2023)logistic regression FLYER COPY ☆  File Edit View Insert Runtime Tools Help Last edited on 17 July	S
:≡	+ Code + Text Connect  Connect  Gemini	^
Q	00 507 VV 307 C1 307	
{ <i>x</i> }	A 307 Date 0	
(~) © <del></del>	dtype: int64	
	[ ] data.dropna(inplace=True)	
	[] data.shape	
	→ (19577, 25)	
	[ ] data['T1'] - data['T1'].apply(lambda x: "SIGNIFICANT WEATHER" if x in ['9', '6', '0', '5'] else "CLEAR")	
<>	<pre>[ ] data['T2'] = data['T2'].apply(lambda x: "SIGNIFICANT WEATHER" if x in ['9', '6', '0', '5'] else "CLEAR")</pre>	
=	[ ] data['T3'] = data['T3'].apply(lambda x: "SIGNIFICANT WEATHER" if x in ['9', '6', '0', '5'] else "CLEAR")	
>_	<pre>[ ] data['T4'] = data['T4'].apply(lambda x: "SIGNIFICANT WEATHER" if x in ['9', '6', '0', '5'] else "CLEAR")</pre>	
		• ×
4	🔎 Type here to search 🛛 👯 🚰 🛋 💆 📴 🔛 🦉 🔮 💆 🦉 🖉 🖉 🖉 🖉 🖉 🖉 🖉 🖓 31°C Mostly cloudy 🔨 🖗 🚱 🌾 🕬 31-08-2024	5
~	Figure 5: Encoding weather variable by lambda	
~	<pre>[ ] data['T']-data['T1']+data['T2']+data['T3']+data['T4']</pre>	
	<pre>[ ] data['T'] = data.apply(lambda row: "SIGNIFICANT WEATHER" if (row['T1']=="SIGNIFICANT WEATHER") or</pre>	")
	<pre>[ ] data = data.drop('T1',axis=1)</pre>	
< >	<pre>[ ] data = data.drop('T2',axis=1)</pre>	
	<pre>[ ] data = data.drop('T3',axis=1)</pre>	
>_	<pre>data - data.drop('T4',axis-1)</pre>	
	🔎 Type here to search 🛛 📢 👔 💽 🧮 💼 💼 🗊 🕨 🛞 🧑 🏧 😂	<u>28</u> °
	Figure 6: Encoding new weather variable 'T' by lambda and drop original weather	

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

dit Vie	ew Inse	n Ru	TUTTR		DOIS F	heip	Last	saved a	11.13	DAIVI										
+ Tex	đ																			Con
ata =	data.dr	rop <mark>('T</mark>	4',a	xis=	1)															
	∙data['1 new=dat	- C		NIFI	CANT I	WEATHE	R"													
ilter_	-		Lei ]																	
	INDEX	YFAR	MN	DT	ΜΔΧ	MIN	ΔW	RF	SSH	SLP	 WBT	DPT	RH	VP	DD	vv	c1	Δ	Date	т
13	42807					15.7		26.0	8.7			16.5			18	96.0	5			SIGNIFICANT WEATHER
	42807				34.7		7		0.7			17.6						-		SIGNIFICANT WEATHER
57			-					0.0		1014.8	 				25	96.0	0	0		
58	42807		2			22.4	-	0.0		1013.5	 	19.4			23	97.0	0	0		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		🗇 🧲 📄 tering new we	🔒 文 🔋 🕅 eather variable		~ \$
[ ] col=lis col	t(data.columns)				
<pre></pre>					

Figure 8: List columns

$\leftarrow \rightarrow$	G ==	colab.research.go	ogle.com/drive/10D	uiqieciam I ti-DPF8E	pivie4kbt/2r1s					☆ 23	S Error
1 Gmail	I ★ Bookma	rks 🕒 M Sent M	fail - sumana 📭 1	Fata Photon Yahoo 🛛 🚺	🏽 PayUMoney 🛛 🌆 W	elcome to Indian	😫 National Train Enqu	i 🔀 Train Time Tab	le 20 📶 BBD Bag	g to Dum D G Go	oogle »
CO			(1969-2023)log time Tools Help	-	a paper for flyer	curtain raise vi	sag ☆		E Com	iment 🛛 🏩 Share	<b>\$</b>
= +	+ Code + T	Text							C	onnect 👻 🔶	Gemini
2			10 24 31.8 22.9	3 2.3 0.8	1002.4 27.4	26.5 84.0 34.6	18 96.0 5 3	2022-10-24 SIG	NIFICANT WEATH	ER	
x}		ows × 22 columns									
7	[] data. → <clas< td=""><td></td><td>Frame.DataFrame'&gt;</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></clas<>		Frame.DataFrame'>								
>	Data # 0 1 2 3 4 5 6 7 8	MN         19577         nc           DT         19577         nc           MAX         19577         nc           MIN         19577         nc           AW         19577         nc           RF         19577         nc           SSH         19577         nc           SLP         19577         nc           MSLP         19577         nc	22 columns): L Count Dtype 								
	10 11 12 13 14	WBT         19577         nc           DPT         19577         nc           19577         nc         19577         nc           e to search         Figure		mation o	of data rea	-	-	s and lab	el encoc		11:50 08-09-2024
E (	10 11 12 13 14 O Type her ALIPORE iew Insert	NBT 19577 nc 19577 nc 19577 nc e to search Figure data (1969-2	on-null float64	mation o	of data re	quired fo	r analysi	s and lab	el encoc	ling	11:50 08-09-2024
E ( cual A tit Vi + Te	10 11 12 13 14 O Type her ALIPORE iew Insert	NBT 19577 nc 19577 nc 19577 nc e to search Figure data (1969-2	e 9: Infor 2023)logistic	mation o	of data re	quired fo	r analysi	s and lab	el encoc	ling	11:50 08-09-2024
E ( Lual A it Vi + Te	10 11 12 13 14 O Type her ALIPORE iew Insert ext	NBT 19577 nc 19577 nc 19577 nc e to search Figure data (1969-2	e 9: Infor 2023)logistic	mation o	of data re	quired fo	r analysi	s and lab	pel encoc	ling	11:50 08-09-2024
ual A ta.do	10 11 12 13 14 O Type her ALIPORE iew Insert ext escribe() INDEX	NBT 19577 nc DPT 19577 nc 19577 nc e to search Figure data (1969-2 Runtime Toc	m-null float64 phonenull float64 <b>2</b> <b>2</b> <b>9</b> : Infor 1023)logistic pls Help Last MN	regression pa	of data rea	quired fo curtain raise MIN	<b>r analysi</b> visag ☆	MSLP		Ling Commer Conne	11:50 08-09-2024
ual / t Vi + Te ta.do	10 11 12 13 14 O Type her ALIPORE iew Insert ext escribe() INDEX	NBT 19577 nc DPT 19577 nc 19577 nc e to search E to search Clata (1969-2 Runtime Toc YEAR	m-null float64 phonenull float64 <b>2</b> <b>2</b> <b>9</b> : Infor 1023)logistic pls Help Last MN	regression pa seaved at 1:15AM	of data red aper for flyer MAX	quired fo curtain raise MIN	r analysi visag ☆ RF	MSLP	DBT	Ling Commer Conne	11:50 08:09-2024
H ( ual A it Vi + Te ta.do	10 11 12 13 14 O Type her ALIPORE iew Insert ext escribe() INDEX 19577.0	NBT 19577 nc DPT 19577 nc 19577 nc eto search Eto search data (1969-2 Runtime Toc YEAR 19577.000000	m-null float64 <b>a 9: Infor</b> 2023)logistic bls Help Last MN 19577.000000	regression pa saved at 1:15AM DT 19577.000000	of data red aper for flyer MAX 19577.000000	quired fo curtain raise MIN 19577.000000	r analysi visag ☆ RF 19577.000000	MSLP 19577.000000	DBT 19577.000000	Commer Commer WBT 19577.000000	11:50 18:09-2024 □ nt <b>2:</b> S ect <b>-</b> 1 19577.000
E ( cual A + Te	10 11 12 13 14 O Type her ALIPORE iew Insert ext escribe() INDEX 19577.0 42807.0	NBT 19577 nc DPT 19577 nc 19577 nc eto search Gdata (1969-2 Runtime Toc YEAR 19577.000000 1995.596772	m-null float64 <b>a 9: Infor</b> 2023)logistic 2023)logistic 2038 Help Last MN 19577.000000 6.517189	regression pa saved at 1:15AM DT 19577.000000 15.722991	Aper for flyer MAX 19577.000000 31.691802	quired fo curtain raise MIN 19577.000000 22.618036	r analysi visag ☆ RF 19577.00000 4.961353	MSLP 19577.000000 1008.910390	DBT 19577.000000 26.283251	Ling Commer Conne WBT 19577.000000 23.256985	11:50 108:09-2022 □ nt ♣, S ect ♥   19577.000 21.600
) ual A t Vi + Te ta.de	10 11 12 13 14 O Type her ALIPORE iew Insert escribe() INDEX 19577.0 42807.0	NBT 19577 nc DPT 19577 nc 19577 nc eto search Eto search Cata (1969-2 Runtime Toc YEAR 19577.000000 1995.596772	m-null float64 <b>2 9: Infor</b> 2023)logistic 2023)logistic 2015 Help Last MN 19577.000000 6.517189 1.000000	rmation o regression pa saved at 1:15AM DT 19577.000000 15.722991 1.000000	MAX 19577.000000 31.691802 15.100000	quired fo curtain raise MIN 19577.000000 22.618036 7.400000	r analysi visag ☆ RF 19577.000000 4.961353 0.000000	MSLP 19577.000000 1008.910390 0.000000	DBT 19577.00000 26.283251 10.800000	Ling Commer Conne UBT 19577.00000 23.256985 8.800000	11:50 108:09-2024 □ 11
it Vi ta.do	10 11 12 13 14 <b>C</b> Type her ALIPORE iew Insert escribe() INDEX 19577.0 42807.0 42807.0	NBT 19577 nc DPT 19577 nc 19577 nc e to search Clata (1969-2 Runtime Toc YEAR 19577.000000 1995.596772 1969.000000	mnull float64         Imnull float64	regression pa saved at 1:15AM DT 19577.000000 15.722991 1.000000 8.000000	MAX 19577.00000 31.691802 15.100000 29.300000	quired fo curtain raise MIN 19577.000000 22.618036 7.400000 18.900000	r analysi visag ☆ 19577.000000 4.961353 0.000000 0.000000	MSLP 19577.000000 1008.910390 0.000000 1004.100000	DBT 19577.000000 26.283251 10.800000 22.800000	Ling Commer Conne UBT 19577.00000 23.256985 8.800000 19.400000	11:50 108:09-2022 □ 11
a ( uual A it Vi + Te ta.do ount hean min	10 11 12 13 14 O Type her ALIPORE iew Insert ext INDEX 19577.0 42807.0 42807.0 42807.0 42807.0	NBT 19577 nc DPT 19577 nc 19577 nc eto search data (1969-2 Runtime Toc YEAR 19577.000000 1995.596772 1969.000000 1982.000000	m-null float64           Im-null float64           Image: State	regression pa saved at 1:15AM DT 19577.000000 15.722991 1.000000 8.000000 16.000000	и data re aper for flyer мах 19577.000000 31.691802 15.100000 29.300000 32.200000	quired fo curtain raise MIN 19577.000000 22.618036 7.400000 18.900000 24.600000	r analysi visag ☆ 19577.000000 4.961353 0.000000 0.000000 0.000000	MSLP 19577.000000 1008.910390 0.000000 1004.100000	DBT 19577.000000 26.283251 10.800000 22.800000 27.800000	Ling Commer Conne Conne UBT 19577.000000 23.256985 8.800000 19.400000 25.400000	11:50 108:09-2024 □ 11 ▲ S 19577.000 21.600 -27.100 17.300 24.100

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.

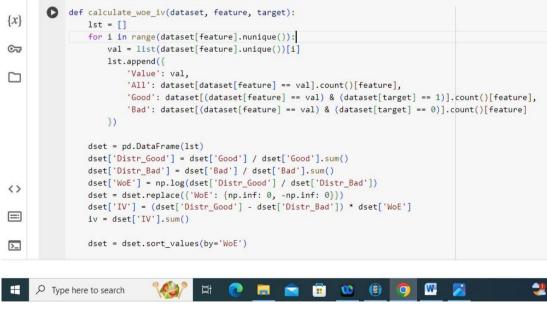


Figure 12: IV method

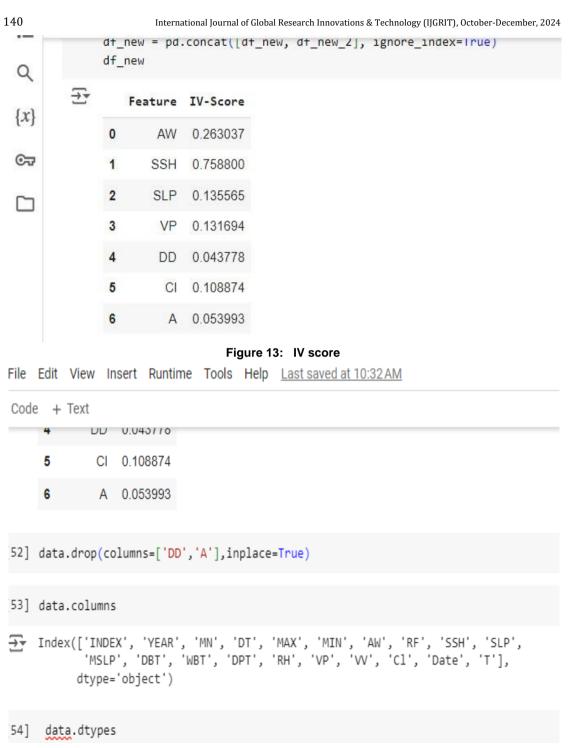
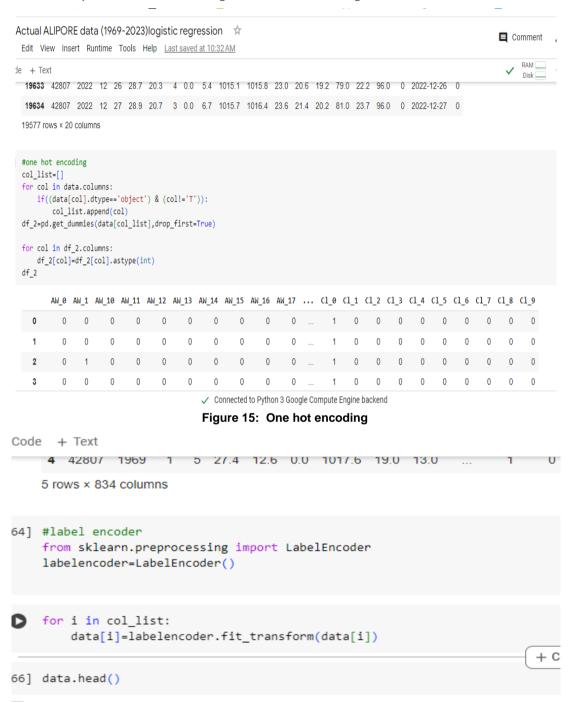


Figure 14: Drop the columns with very low IV scores

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.



#### 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

```
tor 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.

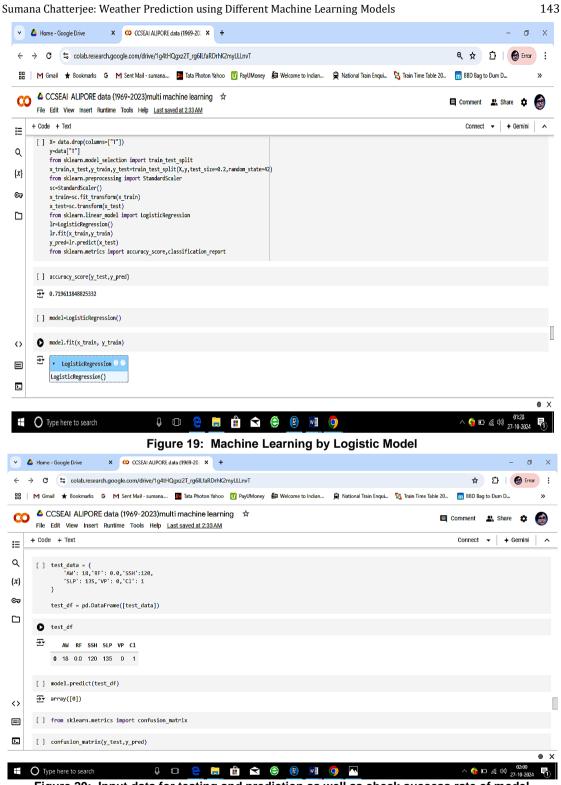


Figure 20: Input data for testing and prediction as well as check success rate of model

144	International Journal of Global Research Innovations & Technology (IJG	RIT), Octob	er-Decembe	r, 2024
👻 🝐 Home - Google Drive 🛛 🗙 🥨 CCSEA	I ALIPORE data (1969-20: × +		- 0	Х
← → C ≤ colab.research.google.com/dri	ve/1g4tHQgxz2T_rg6lLfaRDrhK2myLLLnvT	☆	다   😁 Erro	r :
🔡 🛛 M Gmail ★ Bookmarks 🛛 G M Sent Mail -	sumana 🚆 Tata Photon Yahoo  🗓 PayUMoney 🞄 Welcome to Indian 🚊 National Train Enqui 😫 Train Time Table	20 📊 BBD Ba	g to Dum D	»
CO CCSEAI ALIPORE data (1969-20 File Edit View Insert Runtime Tools	-	E Comment	🚉 Share 🔅	٢
E Files □ ×	+ Code + Text			^
م <u>۵</u> ۳۵۵۵	[] from sklearn.metrics import confusion_matrix			
{ <i>x</i> }	<pre>confusion_matrix(y_test,y_pred)</pre>			
ଫେ ▶ ■ sample_data	- 			
0	<pre>[ ] print(classification_report(y_test,y_pred))</pre>			
	→      precision recall f1-score support			
	0 0.72 0.85 0.78 2296			
	1 0.72 0.53 0.61 1620 accuracy 0.72 3916			
	macro avg 0.72 0.69 0.70 3916 weighted avg 0.72 0.72 0.71 3916			_
<>				
	[ ] score_card = pd.DataFrame(columns=['Wodel', 'Accuracy Score',])			
Disk 75.49 GB available	<pre>def update_score_card(model_name):</pre>			
- 0	Connected to Python 3 Google Compute Engine backend		00:43	• X
O Type here to search     Figure 2	📱 💷 🤮 👼 🛱 😒 😌 😰 🧕 21: Checking of classification report, confusion ma		ロ <i>信</i> (10) 28-10-2024	1
_	ALIPORE data (1969-20) × +		- 0	) X
← → C 😋 colab.research.google.com/driv	ve/1g4tHQgvz2T_rg6lLfaRDrhK2myLLLnvT	☆	다   🚱 Err	or :
🔡 🛛 M Gmail 🔺 Bookmarks 🛛 G M Sent Mail - :	sumana 🧧 Tata Photon Yahoo  👔 PayUMoney   ឆ្លែ Welcome to Indian 🚊 National Train Enqui 😫 Train Time Tabl	e 20 📶 BBD Ba	ag to Dum D	»
CO CCSEAI ALIPORE data (1969-20 File Edit View Insert Runtime Tools	-	Comment	🚉 Share 🏾 🏚	۲
⊨ + Code + Text		Connect		^
<pre>Q [ ] score_card = pd.DataFrame(colum </pre>	ns=['Model','Accuracy Score',])			
<pre>{X} def update_score_card(model_nam</pre>	e):			
C→ global score_card				
<pre>score_card = score_card.app return(score_card)</pre>	<pre>end({'Model': model_name,'Accuracy Score':accuracy_score(y_test, y_pred)},ignore_index = True</pre>	)		
[ ] import pandas as pd from sklearn.metrics import acc	uracy_score			
<pre>score_card = pd.DataFrame(colum</pre>	<pre>ns=['Model','Accuracy Score']) # Initialize score_card if it doesn't exist</pre>			
<pre>def update_score_card(model_nam</pre>	e, y_test, y_pred):			
Updates the score card with	the given model's accuracy.			
new_row = pd.DataFrame({'Mo	del': [model_name],'Accunacy Score': [accunacy_score(y_test, y_pred)]}) re_card, new_row], ignore_index=True)			
noture score cand				
				• ×

Figure 22: Score card for logistic regression

Sum	umana Chatterjee: Weather Prediction using Different Machine Learning Models 145								
•	A Home - Google Drive X CO CCSE	AI ALIPORE data (1969-20) × +	- 6 ×						
÷	→ C 😄 colab.research.google.com/d	ive/1g4tHQgxz2T_rg6lLfaRDrhK2myLL1nvT 5	국 🎦   🞯 Error 🗄						
88	🎛   M Gmail ★ Bookmarks 💪 M Sent Mail - sumana 🧧 Tata Photon Yahoo 👔 PayUMoney 🌆 Welcome to Indian 🔶 National Train Enqui 🗞 Train Time Table 20 💼 BBD Bag to Dum D 🔋								
C	CCSEAI ALIPORE data (1969-2023)multi machine learning ☆ 目 comment 🏖 Share 🌩 🤯								
≣	Files 🖸 🗙	+ Code + Text V RAM Disk							
۹		<pre># Assuming y_test and y_pred are defined update_score_card("Logistic Regression", y_test, y_pred) print(score_card)</pre>							
{x} ⊙⊽	<ul> <li>.config</li> <li>sample_data</li> </ul>	O         Logistic Regression         0.719612							
	, ■ sample_data	<pre>[ ] test_data = {</pre>							
		<pre>test_df = pd.DataFrame([test_data])</pre>							
		[] test_df							
<>		Empty         AW         RF         SSH         SLP         VP         C1           0         18         0.0         120         135         0         1							
≡									
۶.,	Disk 75.49 GB available	[] model.predict(test_df) → array([0])							
		Connected to Python 3 Google Compute Engine backend	• ×						
Ŧ	O Type here to search	U 🕒 😑 💼 🖨 🗢 🙂 🧕 🖼 <u>M</u> 🔨	00:58 ■ / (a d 3) 28-10-2024						

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.

- A Home - Google Drive × 🗴 🗙	AI ALIPORE data (1969-20) × +	- Ø ×
← → C  colab.research.google.com/c	rive/1g4tHQgxz2T_rg6lLfaRDrhK2myLLLnvT	☆ 🖸   🞯 Error 🗄
🔡 📔 M Gmail ★ Bookmarks 🕒 M Sent Mail	- sumana 🧕 Tata Photon Yahoo  🗊 PayUMoney 🛛 🕼 Welcome to Indian 🚊 National Train Enqui 😫 Train Time Table	20 m BBD Bag to Dum D »
CO CCSEAI ALIPORE data (1969-: File Edit View Insert Runtime Tool:		🖪 Comment 🔐 Share 🌣 🚱
⊞ Files □ ×	+ Code + Text	✓ RAM 🔤 👻 🕇 Gemini 🛛 🔨
ଦ୍ 🖻 ୯ 🖻 ⊚	Ξ array([θ])	
{x} → □ .config corr → □ sample_data	<pre>[ ] from sklearn.tree import DecisionTreeClassifier dt=DecisionTreeClassifier() dt.fit(x_train,y_train) y_pred-dt.predict(x_test) print(classification_report(y_test,y_pred))</pre>	
	Precision         recall f1-score         support	
	0 0.76 0.76 0.76 2295 1 0.66 0.67 0.66 1620	
	accuracy 0.72 3916	
	macro avg 0.71 0.71 0.71 3916 weighted avg 0.72 0.72 0.72 3916	
$\diamond$	<pre>accuracy_score(y_test,y_pred)</pre>	
	₹ 0.7198672114402451	
Disk 75.49 GB available	[ ] update_score_card(model_name="DecisionTreeClassifier", y_test=y_test, y_pred=y_pred)	
	<ul> <li>Connected to Python 3 Google Compute Engine backend</li> </ul>	• ×
Type here to search	4 🗆 🗐 🥥 😂 🚖 🏦 🚍	^ 🍓 🖬 🧖 (⊅) 01:06 28-10-2024 🖣

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

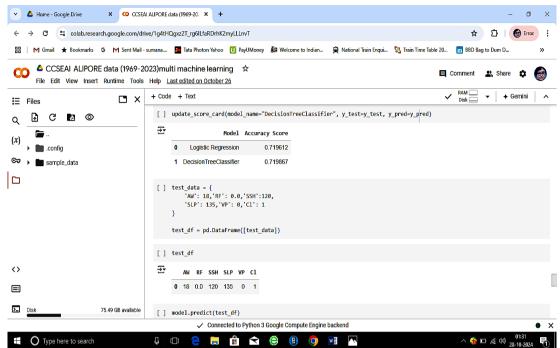


Figure 25: Update score card after execution of decision tree model with input test data and predict weather for this model

## **Random Forest Classifier**

← → C	ve/1g4tHQgxz2T_rg6lLfaRDrhK2myLLLnvT	☆ 🎦   🙆 Error 🗄
🎛 🛛 🕅 Gmail ★ Bookmarks 🕒 M Sent Mail	sumana 🧱 Tata Photon Yahoo 👖 PayUMoney 🕼 Welcome to Indian 侯 National Train Enqui 🖏 Train Time Tabi	le 20 m BBD Bag to Dum D »
CCO CCSEAI ALIPORE data (1969-2 File Edit View Insert Runtime Tools	-	🗏 🎲 🕺 Share 🔗
E Files	+ Code + Text	✓ RAM → ← Gemini ∧
	[ ] model.predict(test_df)	
Q	→ array([0])	
Em .config      F    Sample_data	<pre>[ ] from sklearn.ensemble import RandomForestClassifier rfc= RandomForestClassifier(n_estimators=1000) rfc.fit(x_train,y_train) y_pred=ffc.predict(x_test) accuracy_score(y_test,y_pred)</pre>	
	.79341164453524	
	<pre>print(classification_report(y_test,y_pred))</pre>	
	→ precision recall f1-score support	
·>	0 0.80 0.86 0.83 2296 1 0.78 0.70 0.74 1620	
. /	accuracy 0.79 3916	
	macro avg 0.79 0.78 0.78 3916 weighted avg 0.79 0.79 0.79 3916	
Disk 75.42 GB available		
	✓ Connected to Python 3 Google Compute Engine backend	• :

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

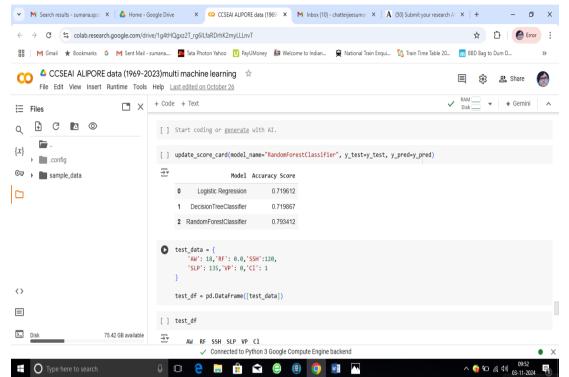


Figure 27: Update score card after execution of Random forest classifier model with input test data and predict weather for this model

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

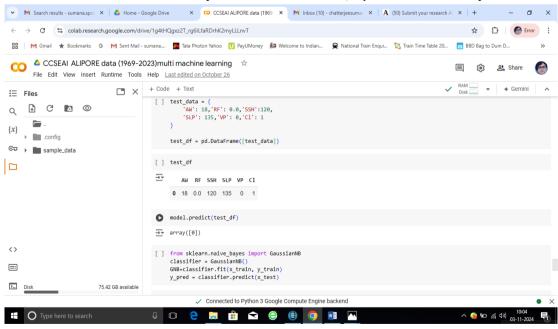


Figure 28: Import Naïve Bayes, fit the train data

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

•	▲ Home - Google Drive × CO CCSEAI ALIPORE data (1969-20: × +	-	o ×
÷	→ C S colab.research.google.com/drive/1g4tHQgrc2T_rg6lLfaRDrhK2myLLLnvT	☆ ひ ⊻	Error :
00	M Gmail ★ Bookmarks G M Sent Mail - sumana 🖪 Tata Photon Yahoo 🕡 PayUMoney 🕼 Welcome to Indian 💂 National Train Enqui 🖏 Train Time Table 20.	. 📊 BBD Bag to Dum D	»
C	CCSEAI ALIPORE data (1969-2023)multi machine learning 🖈 File Edit View Insert Runtime Tools Help Last edited on October 26	🗏 🎲 🖧 St	nare 👩
Ξ	+ Code + Text	Connect 👻 🔶 G	Gemini 🔨
Q	[ ] from sklearn.naive_bayes import GaussianNB		
Q	classifier = GaussianNB() GNB=classifier.fit(x_train, y_train)		
{ <i>x</i> }	<pre>y_pred = classifier.predict(x_test)</pre>		
©⊋	[ ] accuracy_score(y_test,y_pred)		
	<pre>print(classification_report(y_test,y_pred))</pre>		
	0 0.67 0.96 0.79 2296 1 0.85 0.34 0.48 1620		
	accuracy 0.70 3916		
$\langle \rangle$	marco avg 0.76 0.65 0.64 3016 weighted avg 0.74 0.70 0.66 3016		
	ucrelice and on one and		
≕	<pre>[ ] update_score_card(model_name="Naive Bayes", y_test=y_test, y_pred=y_pred)</pre>		
>_			
			• ×
Ŧ	🖸 🗘 Type here to search 🛛 🖟 💷 🤮 📑 💼 😭 😭 📵 📑 🥥	^ 礗 ⊡ 🦟 ଐ) <sub>00</sub>	14:32 3-11-2024 1
	Figure 29: Find accuracy score, classification report with respect to N	laïve Bayes	
Inpu	ut Test Data and Predict Weather by Naïve Bayes		
~	A Home - Google Drive X CO CCSEAI ALIPORE data (1969-20: X +	-	o ×
÷	← → C S colab.research.google.com/drive/1g4tHQgxz2T_rg6iLfaRDrtk2myLLLnvT	* 1 7 6	Error
8	🔠   M Gmail ★ Bookmarks 🛛 M Sent Mail - sumana 📓 Tata Photon Yahoo 😈 PayUMoney 🕼 Welcome to Indian 🔶 National Train Enqui 💱 Train Time Table 20	m BBD Bag to Dum D	>>
•	CCSEAI ALIPORE data (1969-2023)multi machine learning     File Edit View Insert Runtime Tools Help Last edited on October 26	🗏 🎲 🖧 Shar	e 🎯
E	= + Code + Text	Connect 👻 🔶 Ger	mini ^
a	weighted avg 0.74 0.70 0.66 3916		
{ <i>x</i> }	<pre>x} [ ] update_score_card(model_name="Naive Bayes", y_test=y_test, y_pred-y_pred)</pre>		
©:	Hodel Accuracy Score		
C	0 Logistic Regression 0.719612		
	1     DecisionTreeClassifier     0.719867       2     RandomForestClassifier     0.793412		
	3 Naive Bayes 0.700970		
	[] test_data = {		
<	'AW': 18, 'RF': 0.0, 'SSH':120, 'SLP': 135, 'VP': 0, 'Cl': 1 > )		
	test_df = pd.DataFrame([test_data])		

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

∧ 🍓 🖂 1/4% re

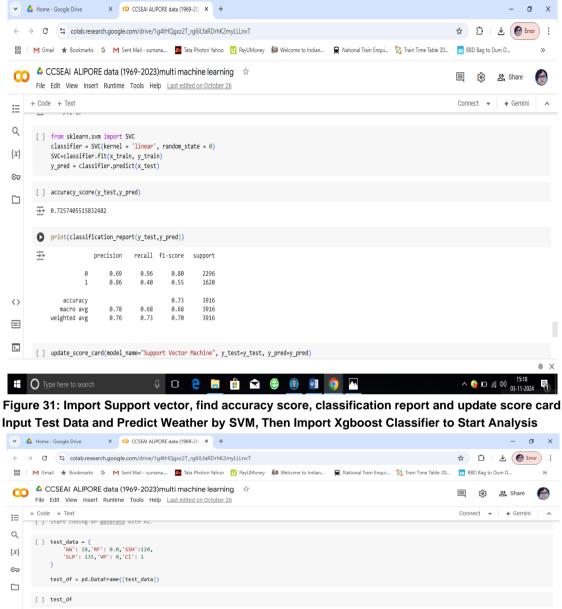
ning 15:08 03-11-2024

📲 🔿 Type here to search 🛛 📮 🗇 🤮 📄 🟦 😒 🥯 📳 👰 🏧

148

▶\_ [] test\_df

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



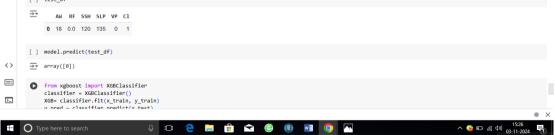


Figure 32: Input test data in support vector machine, predict model accuracy, import XGB00ST classifier

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

101																	
<ul> <li>4</li> </ul>	Home	- Google D	Irive	× α	CCSEAI ALIPO	ORE data (196	<b>9-2</b> 02 ×	+								-	0 ×
>	e c		olab.resear	ch.google.	:om/drive/1g	4tHQgxz2T	_rg6lLfaRE	orhK2myLL	LnvT					☆	Ð	± 🎯	rror
	M Gm	ail ★ Bi	pokmarks	G M Se	nt Mail - suman	a 🎦 Ta	ita Photon Y	ahoo 😈	PayUMoney	🕼 Welcome to India	an 🚊	National Train Enqui	🔀 Train Time Tab	le 20 📶 l	BD Bag to [	Dum D	>>
0					769-2023) Tools Help			-	\$						ŝ	<u>න</u> Share	0
			assifier.		in, y_train ct(x_test)	1)								sumar	e Account 1a chatterj 1a.spssaha	ee 🔸 Gemin a.chatterjee@g	mail.com
	[]	accurac	y_score(y	_test,y_p	red)												
	₹	0.799540	34729315	53													
	[]	print <mark>(c</mark>	lassifica	tion_repo	rt(y_test,	/_pred))											
	∋*		pro	ecision	recall f	f1-score	support	t									
			0 1	0.81 0.79	0.86 0.71	0.83 0.74	2290 1620										
			uracy o avg d avg	0.80 0.80	0.79 0.80	0.80 0.79 0.80	3910 3910 3910	5									
	[]	undate	score car	d(model r	ame="XGBoo:	st" v te	stev test	t v nred	=v nred)								
	÷	upuuce_	score_car		Accuracy Se		50-9_005	c, <u>j</u> _pres	-)_p.cu/								
		0	Logistic Re		0.719												
s://acc	O Typ	pe here to	o search	ns?hl=en&c	ontinue=https:/	/colab.resear	<b>e</b> 1		Ŷ	rg6lLfaRDrhK2myLLLnvT	9		update s			(の) 15:30 03-11-2 10r SV	024
s://acc	O Tyr Figu	pe here to Jre 3	o search 3: Pr	ns?hl=en&c	ontinue=https:/	el acc	e curac	📜 📋 cy, pi	rint c	-	ion i	report ,	-	core c	ard f		024
F Pe	O ™ Figu erpa	pe here to Jre 3	o search 3: Pr eter 1	edict	ontinue=https:/	el acc Rand	e curac om (	📜 📋 cy, pi	rint c	e 🔋 🗐	ion i	report ,	-	core c	ard f	ior SV	024 🛃
F De	O ™ Figu erpa	pe here to ure 3 ram - Google D	o search 3: Pr eter 7 Irive	edict Funir × ∝	minue=https://	Colab.resear	e curac om ( <sup>59-20:</sup> ×	y, pi Class	rint c	e 🔋 🗐	ion i	report ,	-	core c	ard f ter	ior SV	)24 <b>T</b> M
s://acc F pe	O Tyr Figu rpa Home C	pe here to JIRE 3 rame - Google D S c	o search 3: Pr eter 7 Irive olab.resear	edict Funir × α	ccseal ALIPC	(colab.resear () el acco Rand DRE data (196 4tHQgxz2T	Curac Curac Om ( 59-20: ×	y, pi Class +	rint c sifier	e 🔋 🗐	tion i idse	report , r arch CV	-	core c arame	ard f ter	for SV	)24 <b>T</b> M
F De	O Tyr Figu rpa Home O C M Gm	pe here to II TE 3 rame - Google D Sociel & ail & Br CCSEAI	a search <b>3: Pr</b> <b>eter</b> trive olab.resear pokmarks ALIPORE	edict Funir × « G M Se data (19	ccseal ALIPC	(colab.resear c	Curac om ( 99-20: × _rg6lLfaRE tta Photon Y chine le	y, pr Class + DrhK2myLLL ahoo 0 arning	rint c sifier	lassificat	tion i idse	report , r arch CV	Hyperpa	core c arame	ard f ter ☆	for SV	224 <b>T</b>
	O Tyr Figu Frpa Home C M Gm G File	pe here to II TE 3 rame - Google D Sociel & ail & Br CCSEAI	a search 3: Pr eter 7 rive olab.resear pokmarks ALIPORE w Insert	edict Funir × « G M Se data (19	entinue=https:/	(colab.resear c	Curac om ( 99-20: × _rg6lLfaRE tta Photon Y chine le	y, pr Class + DrhK2myLLL ahoo 0 arning	rint c sifier	lassificat	tion i idse	report , r arch CV	Hyperpa	core c arame <sup>ie 20</sup> n E	eard f ter	for SV _ 고 ( @ E Dum D	
	O Tyr Figu Trpa Mome Mome Mome File	e here to ure 3 rame - Google D = c ail * Be CCSEAI Edit Vie + Tex	a search 3: Pr eter 7 rive olab.resear pokmarks ALIPORE w Insert t	edict Funir × cc G M Se cdata (19 Runtime	entinue=https:/	(colab.resear	Curac om ( 59-20: × _rg6lLfaRE chine le ted on Oct	y, pr Class + DrhK2myLLL ahoo 0 arning	rint c sifier	lassificat	tion i idse	report , r arch CV	Hyperpa	core c arame <sup>ie 20</sup> n E	ard f ter	for SV	
	O Tyr Figu rrpa → C M Gm Gm File + Code	e here to JIE 3 JIE 3 JIE 3 - Google D - Google D	a) search 3: Pr eter 1 brive olab.resear pokmarks ALIPORE w Insert t learn.mod t library	edict Funir × cc ch.google. G M Se cdata (19 Runtime el_select of Rando	entinue=https:/ 	(colab.resear c	Curac Jom ( 19-20: × _rg6lLfaRC tta Photon Y chine le ted on Oct	y, pr Class + DrhK2myLLL ahoo 0 arning	rint c sifier	lassificat	tion i idse	report , r arch CV	Hyperpa	core c arame Me 20 n E	ard f ter	for SV	
s://acc F pe 2 2 3 3 4 3 3 3 3	O Tyr Figu rrpa M Gm M Gm File + Code	De here to UTE 3 rame - Google D C- Goog	a search 3: Pr eter 1 brive olab,resear bokmarks ALIPORE w Insert t learn.mod t library learn.ens	edict Funir x a a chgoogle.d data (1% Runtime el_select of Randd	ontinue=https:// c mode ag on l c ccSEAI ALIPC com/drive/1g com/drive/1g th Mail - suman 269-2023)i/ Tools Help ion import mForestClai ort Random	(colab.resear c	Curac Jom ( 19-20: × _rg6lLfaRC tta Photon Y chine le ted on Oct	y, pr Class + DrhK2myLLL ahoo 0 arning	rint c sifier	lassificat	tion i idse	report , r arch CV	Hyperpa	core c arame Me 20 n E	ard f ter	for SV	
	O Tyr Figu rpa Home C M Gm G G G G G G G G G G G G G G G G G G	De here to to the set of the set	a search 3: Pr eter 1 brive olab,resear bokmarks ALIPORE w Insert t learn.mod t library learn.ens	edict Funir × α chgoogle. G M Se chgoogle. A data (15, chgoogle. Chgoogle. A data (15, chgoogle. Ch	ontinue=https:// ammodeling.com/drive/1g com/drive/1g com/drive/1g the fail-suman p69-2023);1 Tools Help ion import mForestCla: ort Random Classifier	(colab.resear c	Curac Jom ( 19-20: × _rg6lLfaRC tta Photon Y chine le ted on Oct	y, pr Class + DrhK2myLLL ahoo 0 arning	rint c sifier	lassificat	tion i idse	report , r arch CV	Hyperpa	core c arame Me 20 n E	ard f ter	for SV	nor »
	O Tyr Figu rpa → C M Gm File + Code	<pre>pe here t d ure 3 ramu -Google D -Google</pre>	a) search 3: Pr eter 1 trive colob.resear colob.resear colob.resear colob.resear colob.resear colob.resear t t learn.mod domrars a a Rando domrars a a Rando domrars a a Rando colob.resear colo	edict Funir x x x chgoogle.d chgoogle.	ontinue=https:// c mode ag on l c coseal aLIPC c com/drive/1g nt Mail - suman 269-2023)J Tools Help ion import mForestCla: oort Randomi Classifier er()	(colab resear Del acco Rand AttHQyz2T AttHQyz2T ar ar ar ar ar ar ar ar ar ar	Curaco Curaco (om ( )9-20 x x rg6lfaRC chron V chine le edd on Oct chron V chine le ssifier 15], ','auto'	Class + + arning ober 26	rint c sifier	lassificat	tion i idse	report , r arch CV	Hyperpa	core c arame Me 20 n E	ard f ter	for SV	D24
F vpe · · · ·	O Tyr Figurpa Home C C File	<pre>e here te ure 3 ramu - Google D - Google D - Google D - Google D - Google D - Google D - CCSEAI Vie Edit Vie Edit Vie Edit Vie Edit Vie Edit Vie From sk # Impor from sk # Unpor from sk</pre>	assearch assearch assearch assearch rive	edic1 Funir x cc ch.google.d G M Se edicta (19 Runtime el_select of Randd emble imp m Forest tclassiff Optimizz estimato archertour arc	entinue=https:// model ag on l ag on	(colab research of the second	Curaco Curaco (om ( )9-20 x x rg6lfaRC chron V chine le edd on Oct chron V chine le ssifier 15], ','auto'	Class + + arning ober 26	rint c sifier	lassificat	tion i idse	report , r arch CV	Hyperpa	core c arame Me 20 n E	ard f ter	for SV	N o > rror ; »

📲 🔿 Type here to search 📮 🖸 😋 📑 😭 😭 🚱 📳 🧕

# Set the rf to the best combination of parameters

rf = grid\_obj.best\_estimator\_

>\_

Figure 34: Hyper Parameter tuning using Grid search CV

• x

P1

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

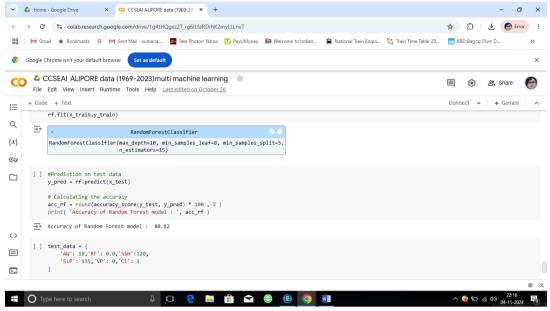


Figure 35: Train data set, print accuracy and prediction on test data on the basis of hyper parameter tuning

#### Update Score Card for Hyper Parameter Tuning of Random Forest

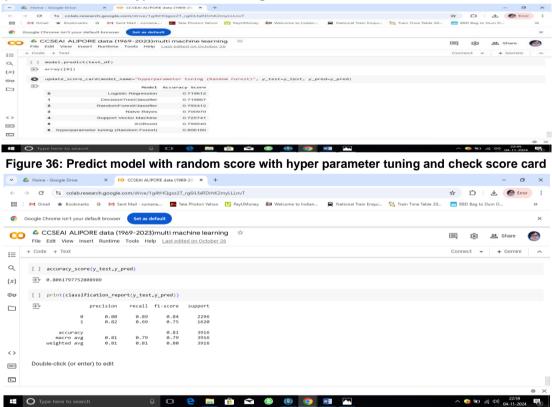


Figure 37: Print accuracy score, classification report with random forest hyper tuning

#### 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.

## 000