FLOOD PREDICTION

Name : G B V K UDAY
Reg No : 39110325
College : Sathyabama Institute of Science and Technology
Branch : B.E , CSE. Project Title : Flood prediction using Machine Learning.

INDEX

b. UI output Screen-shots

INTRODUCTION :

Floods are inevitable, but with timely alerts, their effects can be minimized. There are many people who die every year due to devastating floods, the number of people become homeless and many people die due to lack of proper help after a flood. The lack of timely alerts has always been an issue concerning it. Delay in alerts in flood-prone areas is the biggest loophole of an economy. Conventional systems run a little low in forecasting floods at the right time so that proper actions could be taken before any disaster.
By using machine learning we can predict floods or forecast floods with better accuracy. This project aims at building predictive modeling based on the historical weather data of particular areas in order to predict the occurrence of floods. The predictive model is built on different machine learning algorithms. The concerned authority monitor this flood prediction system through a web application.

LITERATURE SURVEY :

For this project, we need to use classification algorithms as the output we will get is categorical values which is either floods will occur or not so, we need to build the most accurate classification algorithm to get the desired output
In this project,we will be using classification algorithms such as Decision tree, Random forest, KNN, and xgboost. We will train and test the data with these algorithms. From this best model is selected and saved in save format. We will be doing flask integration and cloud deployment.

THEORETICAL ANALYSIS:

1. Install Required Libraries.
2. Data collections

. Collect the dataset or Create the dataset

3. DataPreprocessing

- Import the Libraries.
- Importing the dataset.
- Understanding Data Type and Summary of features.
- Take care of missing data
- Data Visualization.
- Drop the column from DataFrame & replace the missing value.
- Splitting the Dataset into Dependent and Independent variables
- Splitting Data into Train and Test.

4. Model Building

- Training and testing the model
- Evaluation of Model
- Saving the Model

5. Application Building

- Create an HTML file
- Build a Python Code

6. Final UI

    - Dashboard Of the flask app.

HARDWARE / SOFTWARE REQUIREMENTS :

-> Processor : Intel(R) Core(TM) i3 - 6100U 2.30GHz / Equivalent processor
-> RAM : 4Gb or more
->Storage : A basic hard disk of storing files.

Software requirements :

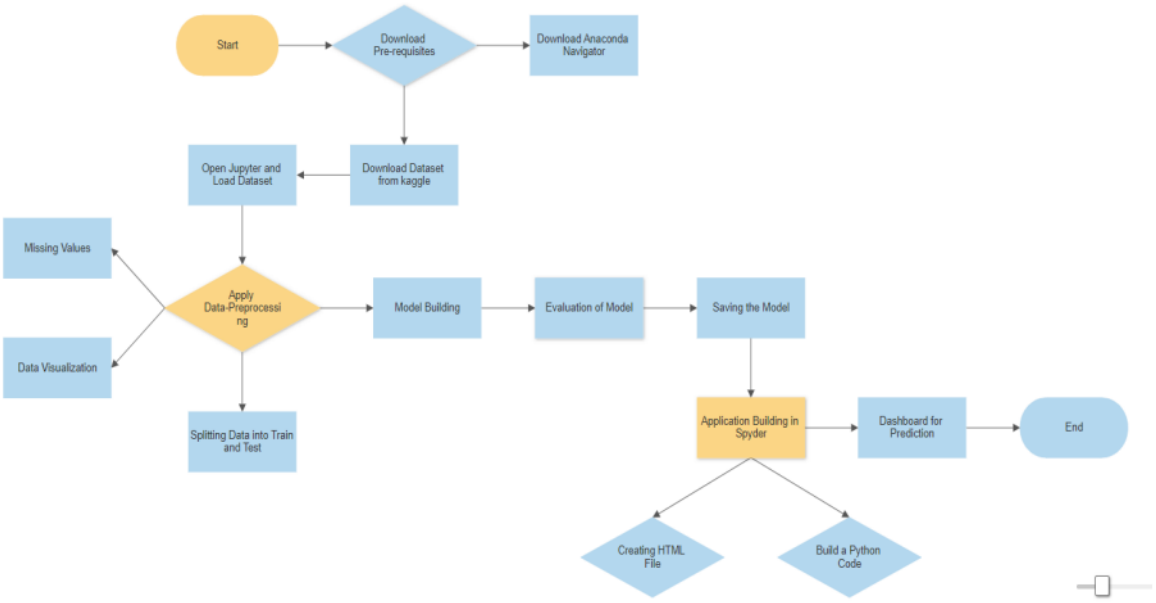
->Anaconda Software
->Visual Studio code / Spyder Software

EXPERIMENTAL INVESTIGATIONS:

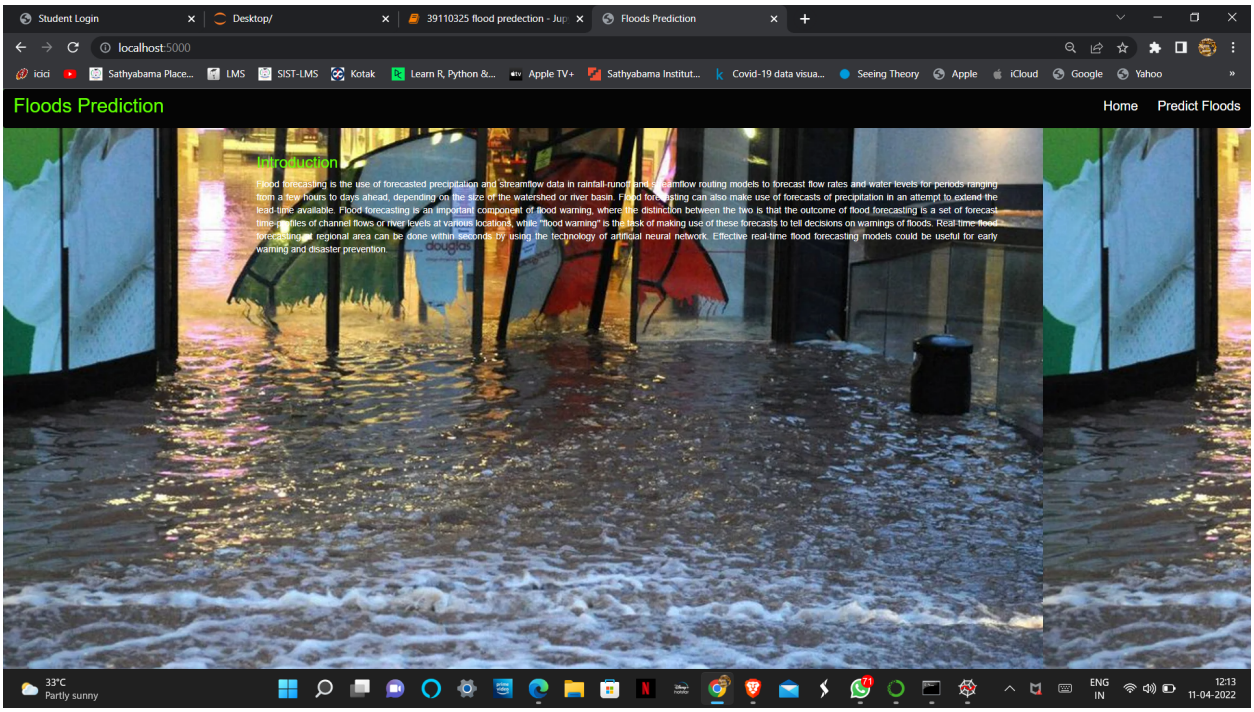
Category: Machine Learning

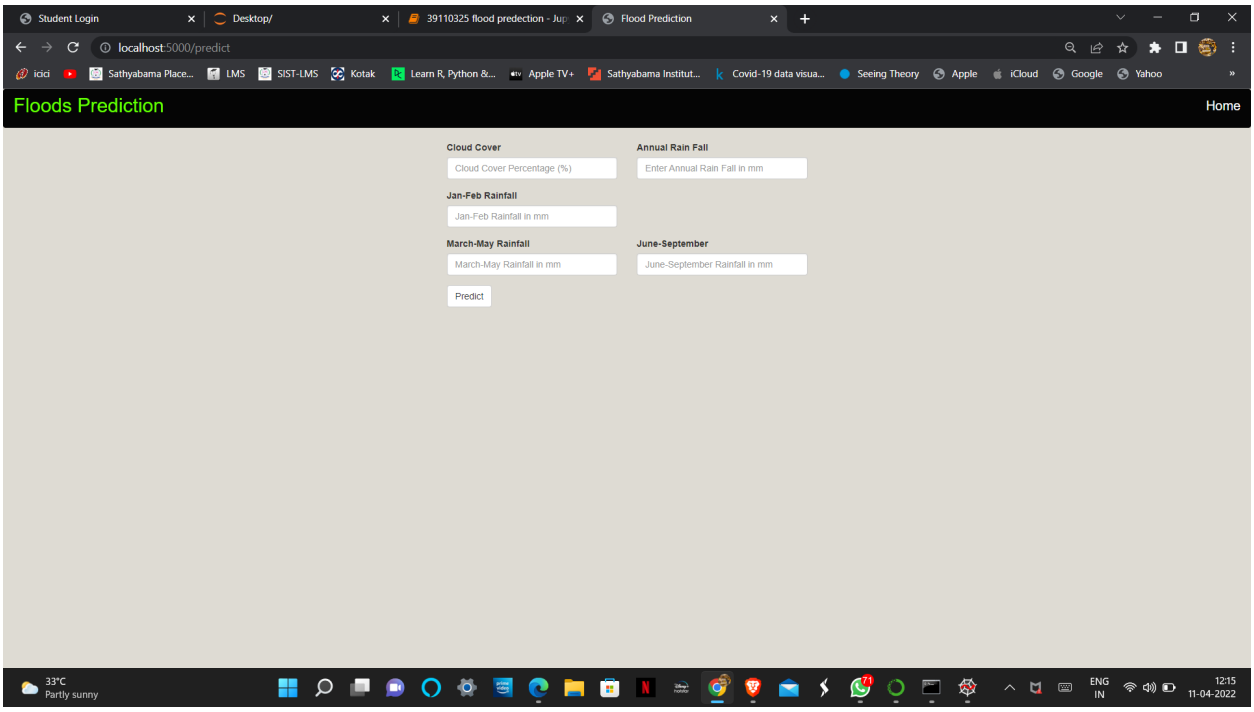
Skills Required:
Python,Python Web Frame Works,Python For Data Analysis,Exploratory Data Analysis,Data Preprocessing Techniques,Machine Learning
While working on the solution, I have seen the dataset of the floods and its quite frequent to see the whether forecast change drastically and cloud cover increases , rainfalls in the months April to June and lastly floods occur.There is interconnection of all the factors which leads to floods.

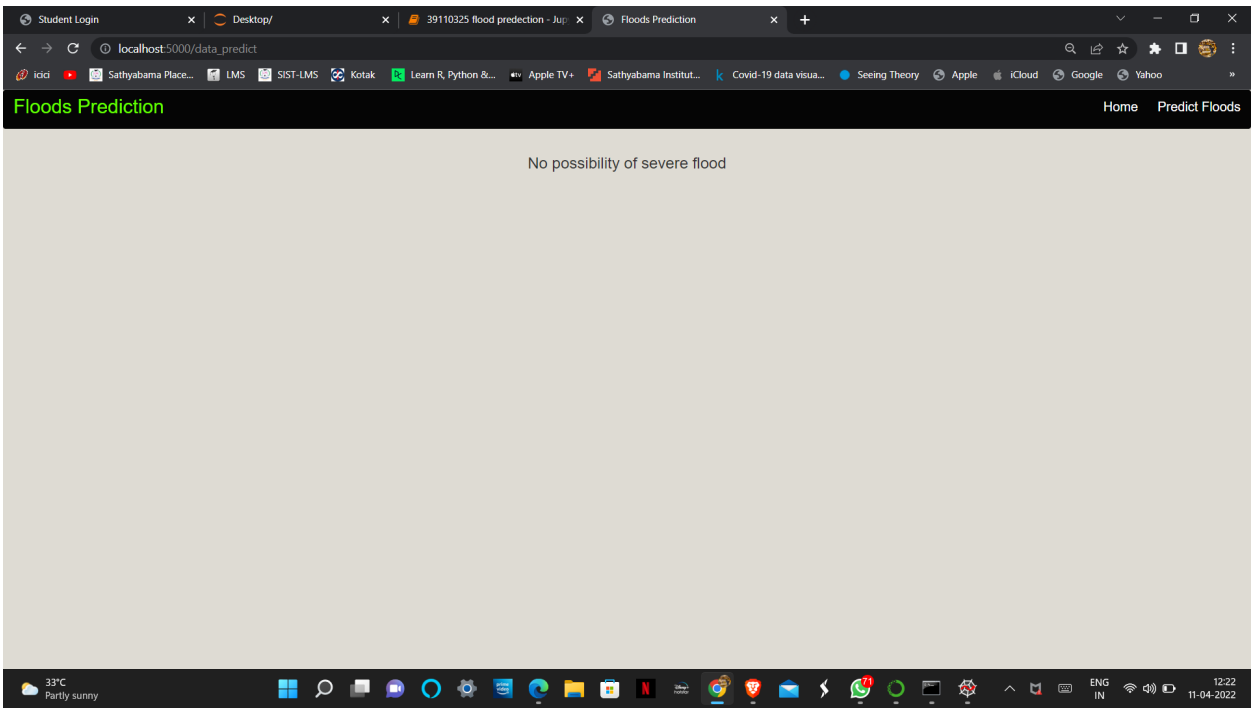
FLOW CHART :

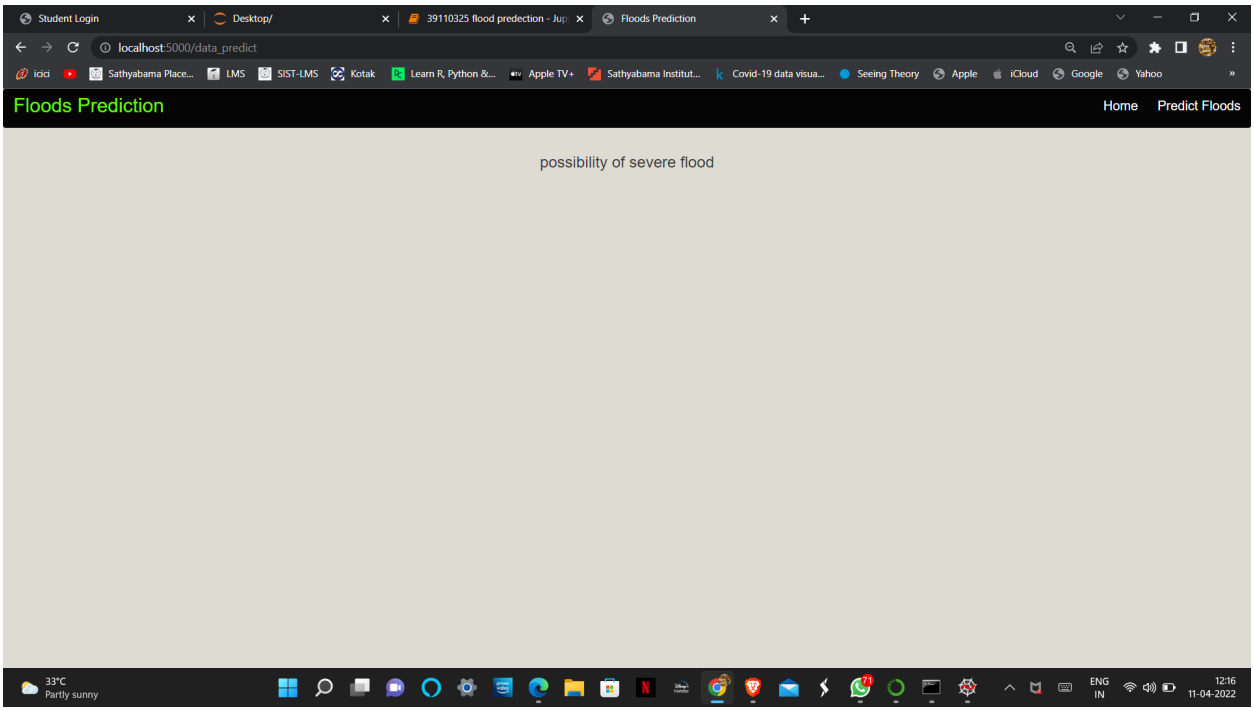


RESULT:




Student Login
Desktop/
39110325 flood predection - Jup
Flood Prediction
localhost:5000/predict
Floods Prediction
Home
Cloud Cover
Cloud Cover Percentage (%)
Annual Rain Fall
Enter Annual Rain Fall in mm
Jan-Feb Rainfall
Jan-Feb Rainfall in mm
March-May Rainfall
March-May Rainfall in mm
June-September
June-September Rainfall in mm
Predict


Student Login
Desktop/
39110325 flood predection - Jup
Floods Prediction
localhost:5000/data_predict
Floods Prediction
Home
Predict Floods
No possibility of severe flood



ADVANTAGES AND DISADVANTAGES ADVANTAGES :

->The main purpose of flood warning is to save life by allowing people, support and emergency services time to prepare for flooding.

->The model which I built is 96 % accurate so one can rely on the prediction done by the model. ->With the help of this model , you can determine and manage environmental and water resource systems.

->Knowledge about the characteristics of a river's drainage basin, such as soil-moisture conditions, ground temperature, snowpack, topography, vegetation cover, and impermeable land area, which can help to predict how extensive and damaging a flood might become.

DISADVANTAGES :

->One of the algorithm used in this model is Random Forest, where it can become a plot as it gets prediction accuracy on complex problems is usually inferior to Gradient - Boosted Trees.

APPLICATIONS :

The applications in flood prediction can be classified according to flood resource variables, i.e., water level, river flood, soil moisture, rainfall–discharge, precipitation, river inflow, peak flow, river flow, rainfall–runoff, flash flood, rainfall, streamflow, seasonal stream flow.

CONCLUSION :

The summary of the project is that even though Floods are inevitable, they can be prevented with accurate measures like predicting the outcomes, changes to be taken to be on the safe side like building dams, reservoirs etc..and especially the technology has taken huge turn so, it will be matter of time before we gain control on what we are challenged.

FUTURE SCOPE :

There can be future changes can be made to my model by adding the data like water level, soil moisture, average temperature of the area, rainfall - discharge , precipitation , drainage systems in the area etc.... in the dataset makes even more accurate and since the data will be enhanced, the analysis becomes the key part in the prediction.

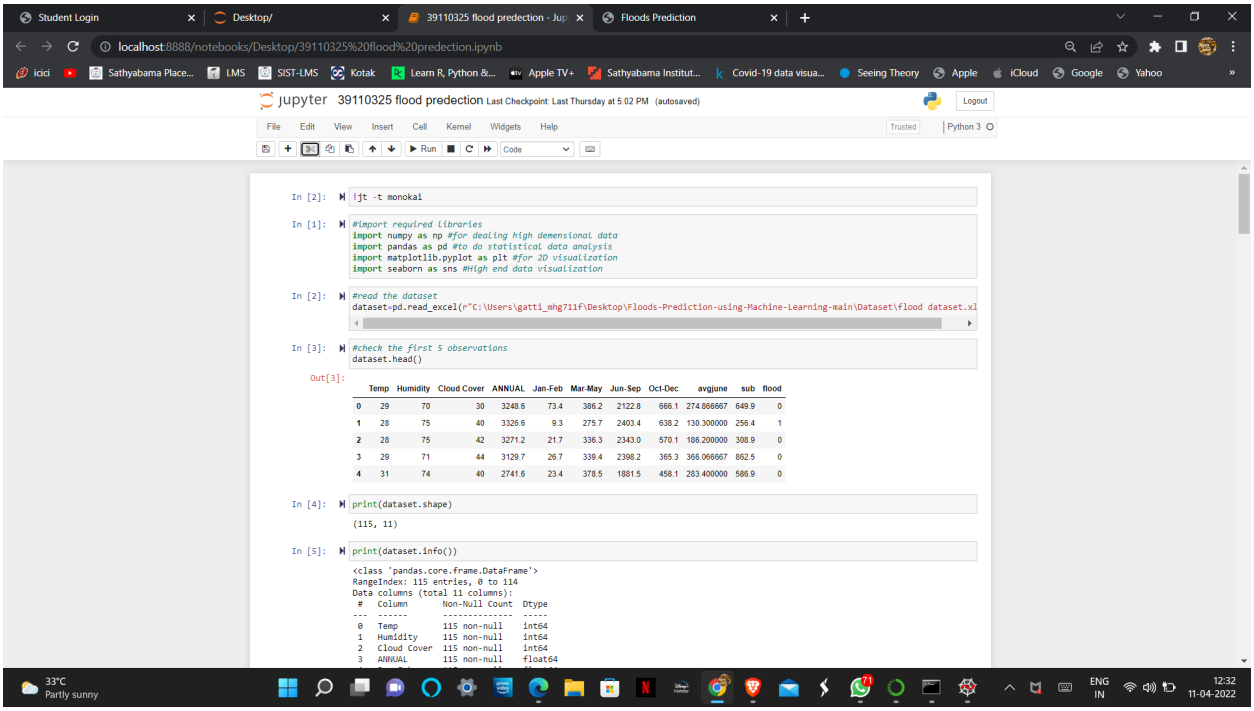
BIBILOGRAPHY:

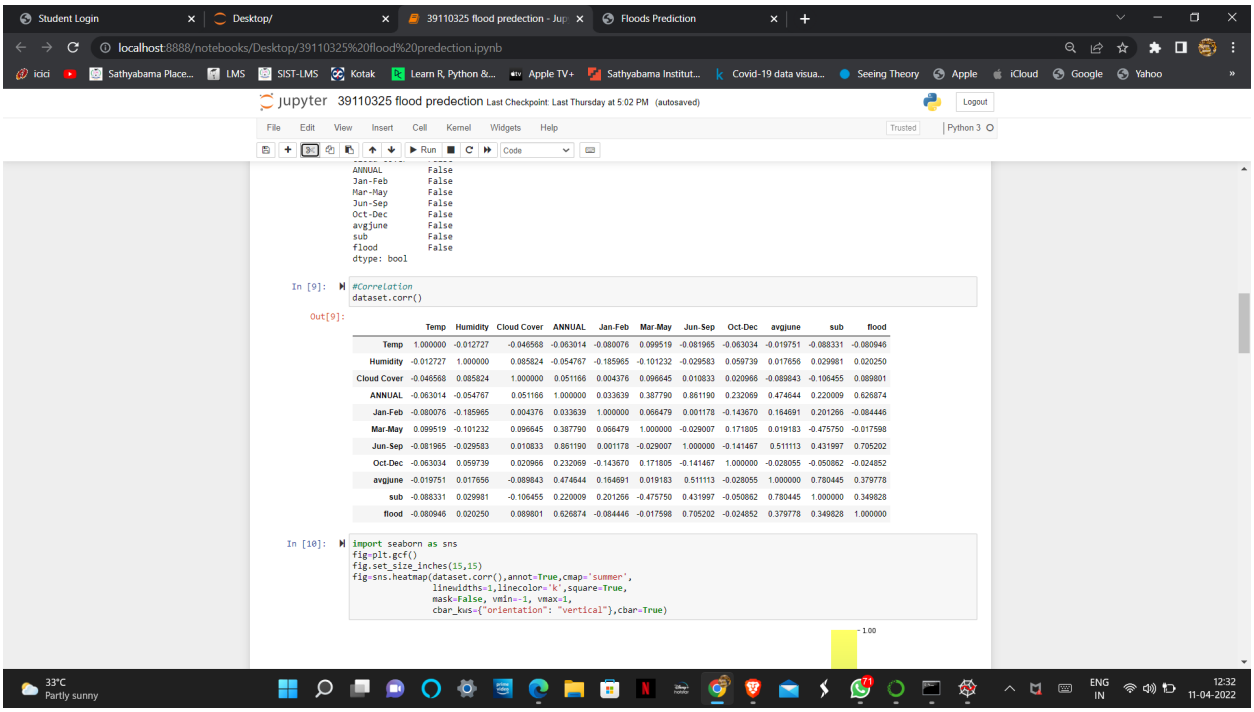
There are some references which are to be considered while making the project.

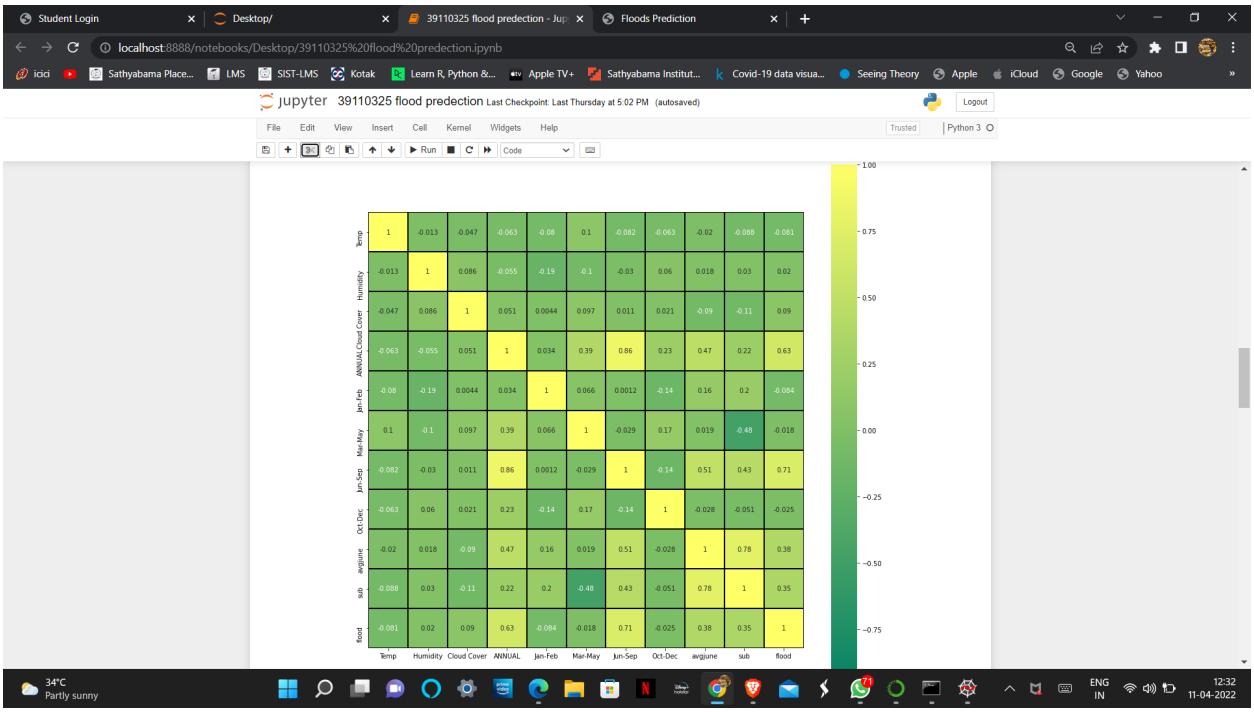
URL - 1 : Flood prediction - Oxford Reference

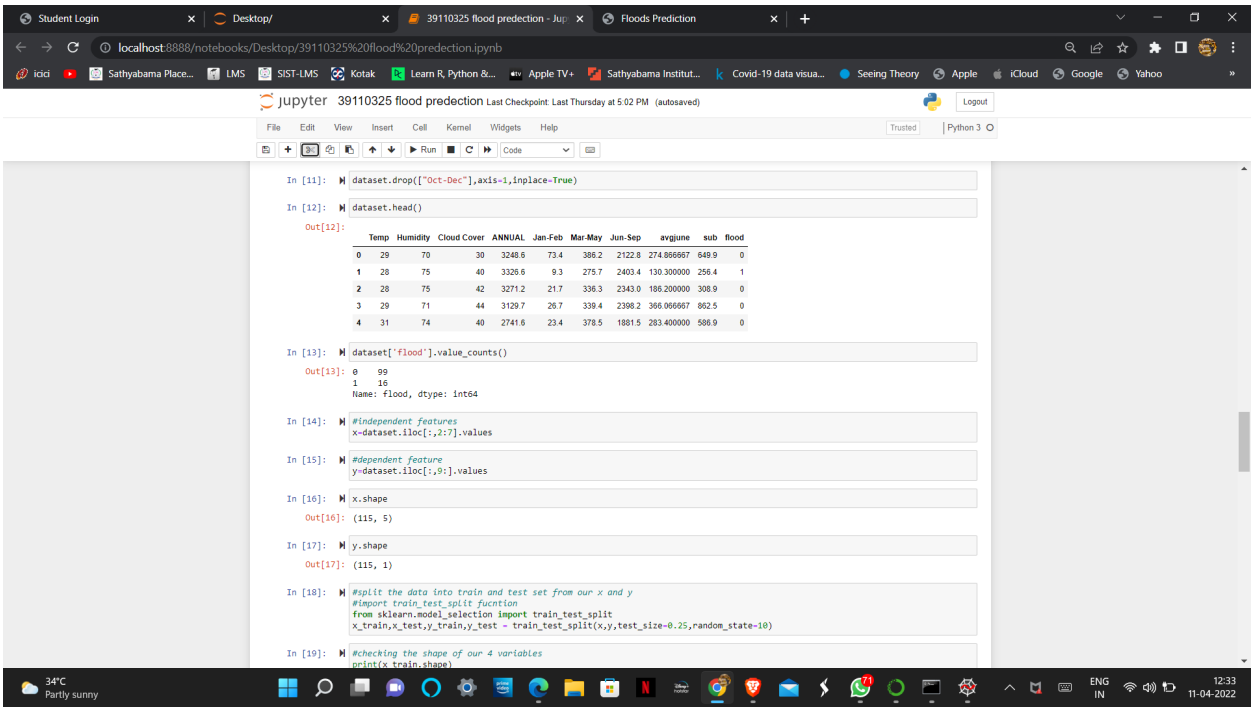
URL - 2 : Flood Forecasting - an overview | ScienceDirect Topics

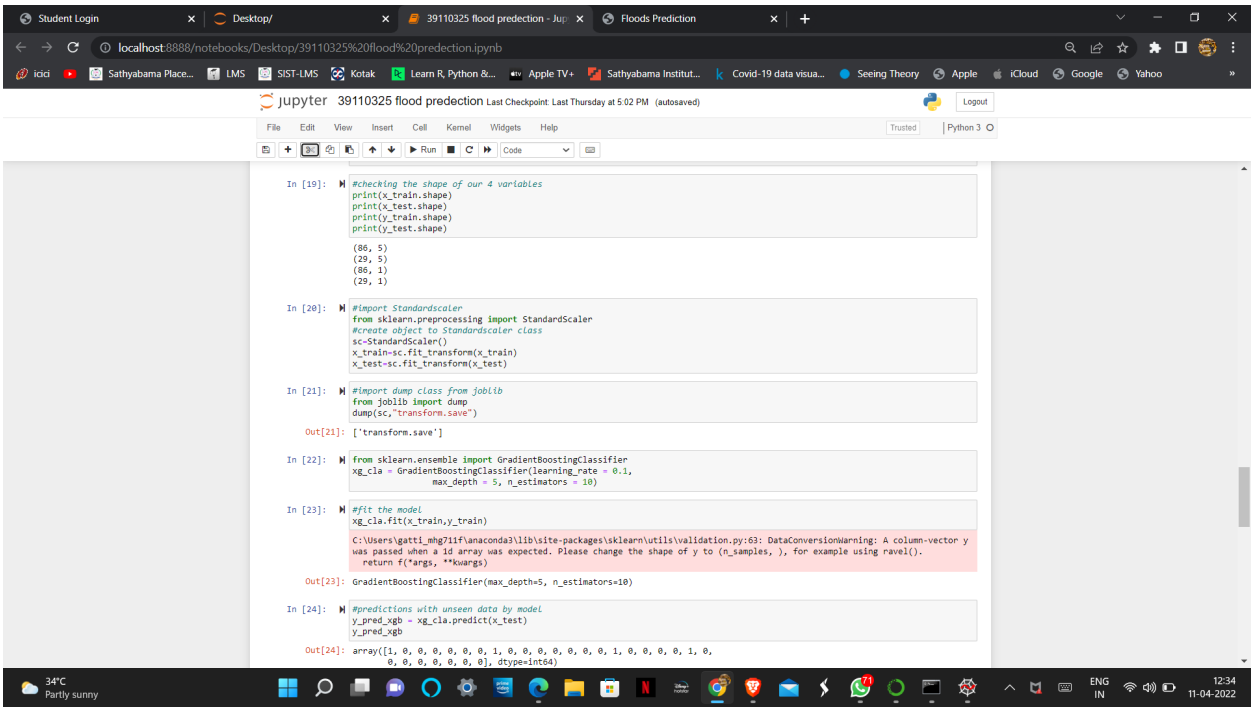
APPENDIX - SOURCE CODE :

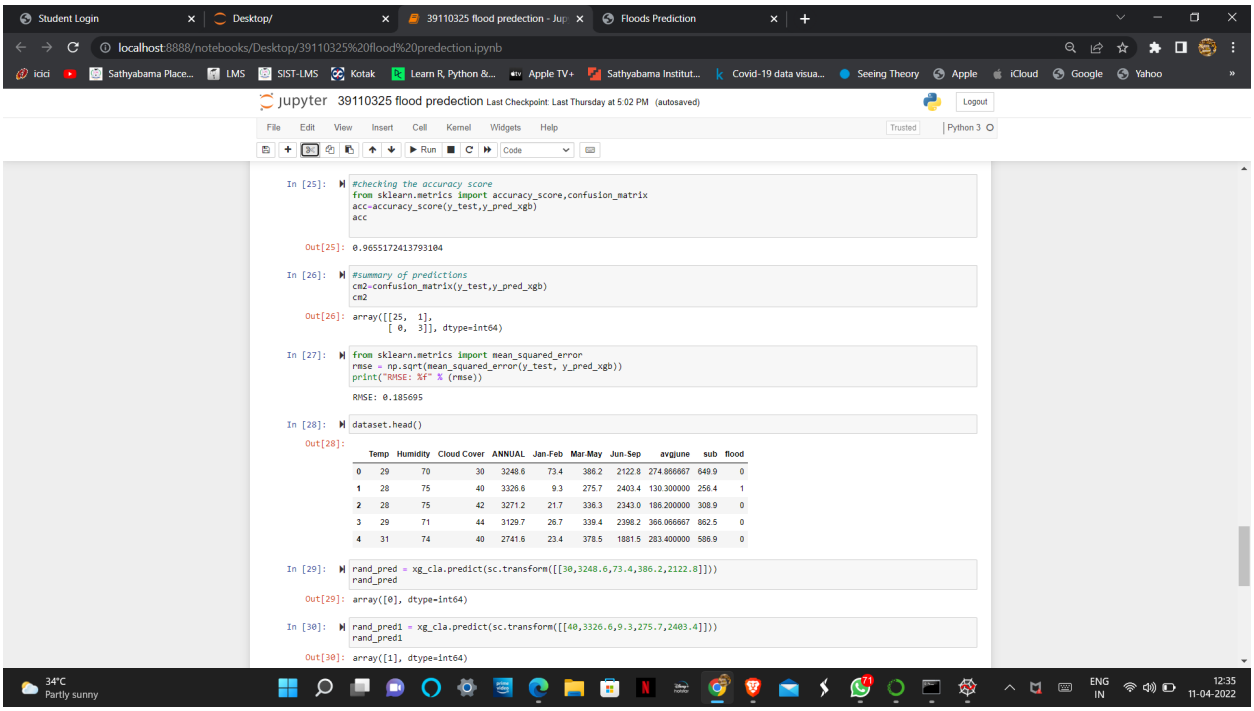



Student Login
Desktop/
39110325 flood predection - Jup
Floods Prediction
localhost:8888/notebooks/Desktop/39110325%20flood%20predection.ipynb
jupyter 39110325 flood predection Last Checkpoint: Last Thursday at 5:02 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
ANNUAL False
Jan-Feb False
Mar-May False
Jun-Sep False
Oct-Dec False
avgjune False
sub False
flood False
dtype: bool
In [9]:
#Correlation
dataset.corr()
Out[9]:
Temp Humidity Cloud Cover ANNUAL Jan-Feb Mar-May Jun-Sep Oct-Dec avgjune sub flood
Temp 1.000000 -0.012727 -0.046568 -0.063014 -0.080076 0.099519 -0.081965 -0.063034 -0.019751 -0.088331 -0.080946
Humidity -0.012727 1.000000 0.085824 -0.054767 -0.185965 -0.101232 -0.029583 0.059739 0.017656 0.029981 0.020250
Cloud Cover -0.046568 0.085824 1.000000 0.051166 0.004376 0.096645 0.010833 0.020966 -0.089843 -0.106455 0.089801
ANNUAL -0.063014 -0.054767 0.051166 1.000000 0.033639 0.387790 0.861190 0.232069 0.474644 0.220009 0.626874
Jan-Feb -0.080076 -0.185965 0.004376 0.033639 1.000000 0.066479 0.001178 -0.143670 0.164691 0.201266 -0.084446
Mar-May 0.099519 -0.101232 0.096645 0.387790 0.066479 1.000000 -0.029007 0.171805 0.019183 -0.475750 -0.017598
Jun-Sep -0.081965 -0.029583 0.010833 0.861190 0.001178 -0.029007 1.000000 -0.141467 0.511113 0.431997 0.705202
Oct-Dec -0.063034 0.059739 0.020966 0.232069 -0.143670 0.171805 -0.141467 1.000000 -0.028055 -0.050862 -0.024852
avgjune -0.019751 0.017656 -0.089843 0.474644 0.164691 0.019183 0.511113 -0.028055 1.000000 0.780445 0.379778
sub -0.088331 0.029981 -0.106455 0.220009 0.201266 -0.475750 0.431997 -0.050862 0.780445 1.000000 0.349828
flood -0.080946 0.020250 0.089801 0.626874 -0.084446 -0.017598 0.705202 -0.024852 0.379778 0.349828 1.000000
In [10]:
import seaborn as sns
fig=plt.gcf()
fig.set_size_inches(15,15)
fig=sns.heatmap(dataset.corr(),annot=True,cmap='summer',
linewidths=1,linecolor='k',square=True,
mask=False, vmin=-1, vmax=1,
cbar_kws={"orientation": "vertical"},cbar=True)
1.00
33°C
Partly sunny
ENG
IN
12:32
11-04-2022


Student Login
Desktop/
39110325 flood predection - Jup
Floods Prediction
localhost:8888/notebooks/Desktop/39110325%20flood%20predection.ipynb
jupyter 39110325 flood predection Last Checkpoint: Last Thursday at 5:02 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Temp Humidity Cloud Cover ANNUAL Jan-Feb Mar-May Jun-Sep Oct-Dec avgjune sub flood
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
34°C
Partly sunny
ENG
IN
12:32
11-04-2022


Student Login
Desktop/
39110325 flood predection - Jup
Floods Prediction
localhost:8888/notebooks/Desktop/39110325%20flood%20predection.ipynb
jupyter 39110325 flood predection Last Checkpoint: Last Thursday at 5:02 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [11]: dataset.drop(["Oct-Dec"],axis=1,inplace=True)
In [12]: dataset.head()
Out[12]:
Temp Humidity Cloud Cover ANNUAL Jan-Feb Mar-May Jun-Sep avgjune sub flood
0 29 70 30 3248.6 73.4 386.2 2122.8 274.866667 649.9 0
1 28 75 40 3326.6 9.3 275.7 2403.4 130.300000 256.4 1
2 28 75 42 3271.2 21.7 336.3 2343.0 186.200000 308.9 0
3 29 71 44 3129.7 26.7 339.4 2398.2 366.066667 862.5 0
4 31 74 40 2741.6 23.4 378.5 1881.5 283.400000 586.9 0
In [13]: dataset['flood'].value_counts()
Out[13]: 0 99
1 16
Name: flood, dtype: int64
In [14]: #independent features
x=dataset.iloc[:,2:7].values
In [15]: #dependent feature
y=dataset.iloc[:,9:].values
In [16]: x.shape
Out[16]: (115, 5)
In [17]: y.shape
Out[17]: (115, 1)
In [18]: #split the data into train and test set from our x and y
#import train_test_split fucntion
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state=10)
In [19]: #checking the shape of our 4 variables
print(x_train.shape)
34°C
Partly sunny
ENG
IN
12:33
11-04-2022


Student Login
Desktop/
39110325 flood predection - Jup
Floods Prediction
localhost:8888/notebooks/Desktop/39110325%20flood%20predection.ipynb
jupyter 39110325 flood predection Last Checkpoint: Last Thursday at 5:02 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [19]: #checking the shape of our 4 variables
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
(86, 5)
(29, 5)
(86, 1)
(29, 1)
In [20]: #import Standardscaler
from sklearn.preprocessing import StandardScaler
#create object to Standardscaler class
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
In [21]: #import dump class from joblib
from joblib import dump
dump(sc,"transform.save")
Out[21]: ['transform.save']
In [22]: from sklearn.ensemble import GradientBoostingClassifier
xg_cla = GradientBoostingClassifier(learning_rate = 0.1,
max_depth = 5, n_estimators = 10)
In [23]: #fit the model
xg_cla.fit(x_train,y_train)
C:\Users\gatti_mhg711f\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
return f(*args, **kwargs)
Out[23]: GradientBoostingClassifier(max_depth=5, n_estimators=10)
In [24]: #predictions with unseen data by model
y_pred_xgb = xg_cla.predict(x_test)
y_pred_xgb
Out[24]: array([1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0], dtype=int64)
34°C
Partly sunny
ENG
IN
12:34
11-04-2022



Student Login | Desktop/ | 39110325 flood predection - Jup | Floods Prediction

localhost:8888/notebooks/Desktop/39110325%20flood%20predection.ipynb

jupyter 39110325 flood predection Last Checkpoint: Last Thursday at 5:02 PM (autosaved)

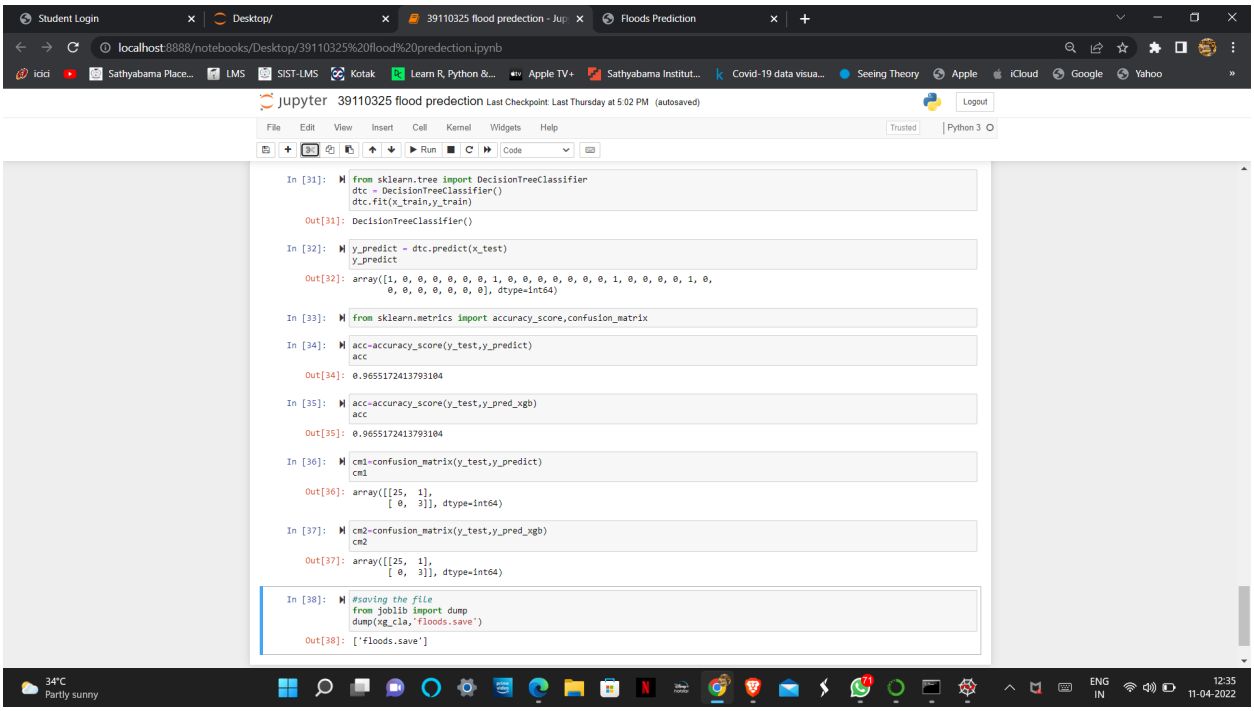
File Edit View Insert Cell Kernel Widgets Help

```
In [25]: #checking the accuracy score
         from sklearn.metrics import accuracy_score,confusion_matrix
         acc=accuracy_score(y_test,y_pred_xgb)
         acc

Out[25]: 0.9655172413793104

In [26]: #summary of predictions
         cm2=confusion_matrix(y_test,y_pred_xgb)
         cm2

Out[26]: array([[25,  1],
                [ 0,  3]], dtype=int64)

In [27]: from sklearn.metrics import mean_squared_error
         rmse = np.sqrt(mean_squared_error(y_test, y_pred_xgb))
         print("RMSE: %f" % (rmse))

         RMSE: 0.185695

In [28]: dataset.head()

Out[28]:
```

	Temp	Humidity	Cloud Cover	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	avgjune	sub	flood
0	29	70	30	3248.6	73.4	386.2	2122.8	274.866667	649.9	0
1	28	75	40	3326.6	9.3	275.7	2403.4	130.300000	256.4	1
2	28	75	42	3271.2	21.7	336.3	2343.0	186.200000	308.9	0
3	29	71	44	3129.7	26.7	339.4	2398.2	366.066667	862.5	0
4	31	74	40	2741.6	23.4	378.5	1881.5	283.400000	586.9	0

```
In [29]: rand_pred = xg_cla.predict(sc.transform([[30,3248.6,73.4,386.2,2122.8]]))
         rand_pred

Out[29]: array([0], dtype=int64)

In [30]: rand_pred1 = xg_cla.predict(sc.transform([[40,3326.6,9.3,275.7,2403.4]]))
         rand_pred1

Out[30]: array([1], dtype=int64)
```

34°C Partly sunny | ENG IN | 12:35 11-04-2022



Student Login | Desktop/ | 39110325 flood predection - Jup | Floods Prediction

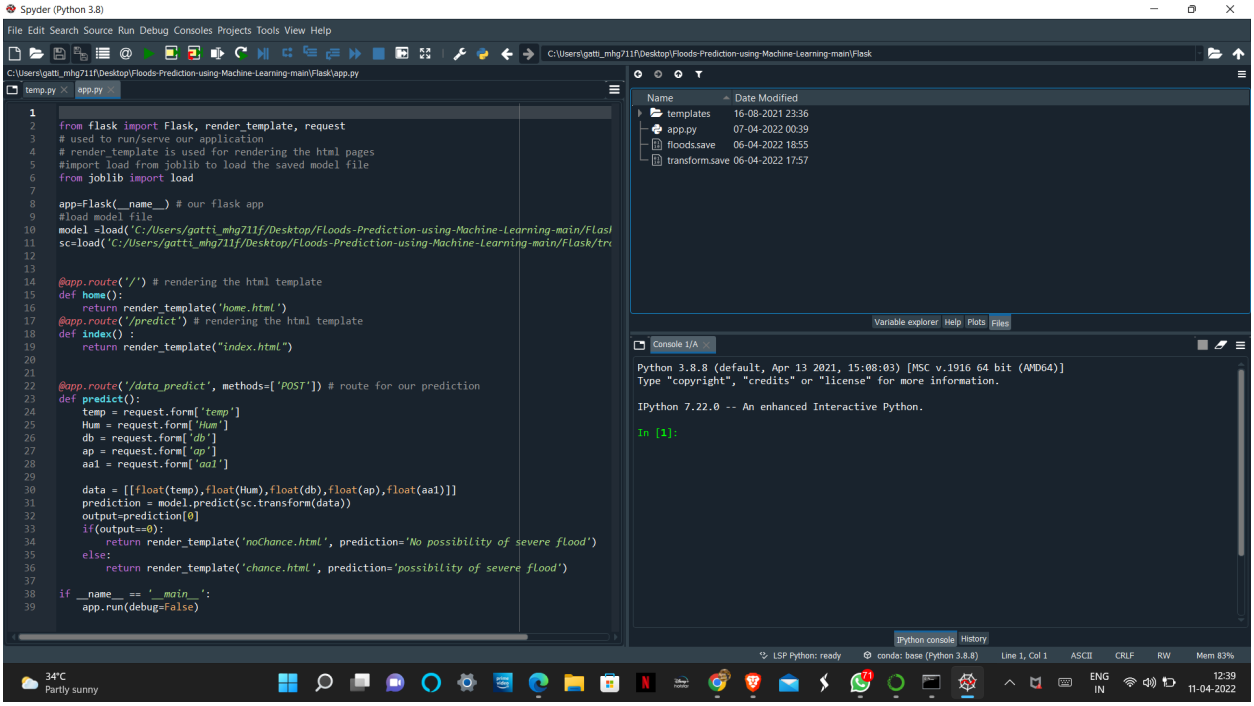
localhost:8888/notebooks/Desktop/39110325%20flood%20predection.ipynb

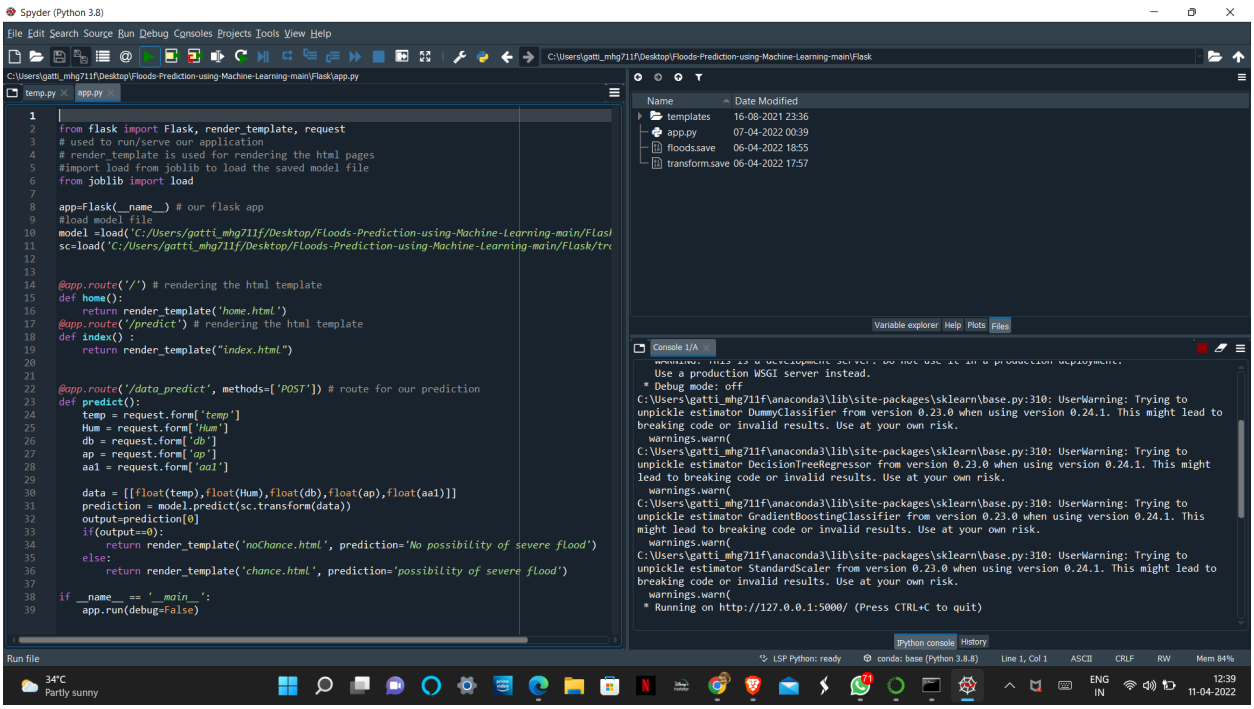
jupyter 39110325 flood predection Last Checkpoint: Last Thursday at 5:02 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

```
In [31]: from sklearn.tree import DecisionTreeClassifier
         dtc = DecisionTreeClassifier()
         dtc.fit(x_train,y_train)

Out[31]: DecisionTreeClassifier()

In [32]: y_predict = dtc.predict(x_test)
         y_predict

Out[32]: array([1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
                0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)

In [33]: from sklearn.metrics import accuracy_score,confusion_matrix

In [34]: acc=accuracy_score(y_test,y_predict)
         acc

Out[34]: 0.9655172413793104

In [35]: acc=accuracy_score(y_test,y_pred_xgb)
         acc

Out[35]: 0.9655172413793104

In [36]: cm1=confusion_matrix(y_test,y_predict)
         cm1

Out[36]: array([[25,  1],
                [ 0,  3]], dtype=int64)

In [37]: cm2=confusion_matrix(y_test,y_pred_xgb)
         cm2

Out[37]: array([[25,  1],
                [ 0,  3]], dtype=int64)

In [38]: #saving the file
         from joblib import dump
         dump(xg_cla,'floods.save')

Out[38]: ['floods.save']
```

34°C Partly sunny | ENG IN | 12:35 11-04-2022


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\gatti_mhg711f\Desktop\Floods-Prediction-using-Machine-Learning-main\Flask\app.py
temp.py
app.py
from flask import Flask, render_template, request
# used to run/serve our application
# render_template is used for rendering the html pages
#import load from joblib to load the saved model file
from joblib import load
app=Flask(__name__) # our flask app
#load model file
model =load('C:/Users/gatti_mhg711f/Desktop/Floods-Prediction-using-Machine-Learning-main/Flas
sc=load('C:/Users/gatti_mhg711f/Desktop/Floods-Prediction-using-Machine-Learning-main/Flask/tr
@app.route('/') # rendering the html template
def home():
return render_template('home.html')
@app.route('/predict') # rendering the html template
def index() :
return render_template("index.html")
@app.route('/data_predict', methods=['POST']) # route for our prediction
def predict():
temp = request.form['temp']
Hum = request.form['Hum']
db = request.form['db']
ap = request.form['ap']
aa1 = request.form['aa1']
data = [[float(temp),float(Hum),float(db),float(ap),float(aa1)]]
prediction = model.predict(sc.transform(data))
output=prediction[0]
if(output==0):
return render_template('noChance.html', prediction='No possibility of severe flood')
else:
return render_template('chance.html', prediction='possibility of severe flood')
if __name__ == '__main__':
app.run(debug=False)
Name
Date Modified
templates
16-08-2021 23:36
app.py
07-04-2022 00:39
floods.save
06-04-2022 18:55
transform.save
06-04-2022 17:57
Variable explorer
Help
Plots
Files
Console 1/A
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.22.0 -- An enhanced Interactive Python.
In [1]:
IPython console
History
LSP Python: ready
conda: base (Python 3.8.8)
Line 1, Col 1
ASCII
CRLF
RW
Mem 83%
34°C
Partly sunny
ENG
IN
12:39
11-04-2022


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\gatti_mhg711f\Desktop\Floods-Prediction-using-Machine-Learning-main\Flask\app.py
temp.py
app.py
from flask import Flask, render_template, request
# used to run/serve our application
# render_template is used for rendering the html pages
#import load from joblib to load the saved model file
from joblib import load
app=Flask(__name__) # our flask app
#load model file
model =load('C:/Users/gatti_mhg711f/Desktop/Floods-Prediction-using-Machine-Learning-main/Flas
sc=load('C:/Users/gatti_mhg711f/Desktop/Floods-Prediction-using-Machine-Learning-main/Flask/tr
@app.route('/') # rendering the html template
def home():
return render_template('home.html')
@app.route('/predict') # rendering the html template
def index() :
return render_template("index.html")
@app.route('/data_predict', methods=['POST']) # route for our prediction
def predict():
temp = request.form['temp']
Hum = request.form['Hum']
db = request.form['db']
ap = request.form['ap']
aa1 = request.form['aa1']
data = [[float(temp),float(Hum),float(db),float(ap),float(aa1)]]
prediction = model.predict(sc.transform(data))
output=prediction[0]
if(output==0):
return render_template('noChance.html', prediction='No possibility of severe flood')
else:
return render_template('chance.html', prediction='possibility of severe flood')
if __name__ == '__main__':
app.run(debug=False)
Name
Date Modified
templates
16-08-2021 23:36
app.py
07-04-2022 00:39
floods.save
06-04-2022 18:55
transform.save
06-04-2022 17:57
Variable explorer
Help
Plots
Files
Console 1/A
Use a production WSGI server instead.
* Debug mode: off
C:\Users\gatti_mhg711f\anaconda3\lib\site-packages\sklearn\base.py:310: UserWarning: Trying to unpickle estimator DummyClassifier from version 0.23.0 when using version 0.24.1. This might lead to breaking code or invalid results. Use at your own risk.
warnings.warn(
C:\Users\gatti_mhg711f\anaconda3\lib\site-packages\sklearn\base.py:310: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 0.23.0 when using version 0.24.1. This might lead to breaking code or invalid results. Use at your own risk.
warnings.warn(
C:\Users\gatti_mhg711f\anaconda3\lib\site-packages\sklearn\base.py:310: UserWarning: Trying to unpickle estimator GradientBoostingClassifier from version 0.23.0 when using version 0.24.1. This might lead to breaking code or invalid results. Use at your own risk.
warnings.warn(
C:\Users\gatti_mhg711f\anaconda3\lib\site-packages\sklearn\base.py:310: UserWarning: Trying to unpickle estimator StandardScaler from version 0.23.0 when using version 0.24.1. This might lead to breaking code or invalid results. Use at your own risk.
warnings.warn(
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
IPython console
History
Run file
LSP Python: ready
conda: base (Python 3.8.8)
Line 1, Col 1
ASCII
CRLF
RW
Mem 84%
34°C
Partly sunny
ENG
IN
12:39
11-04-2022