**SPRINT 4**

|  |  |
| --- | --- |
| Team ID | PNT2022TMID17878 |
| Project Name | Exploratory Analysis Of Rainfall Data In India For Agriculture |

**DEPLOYING ML MODELS WITH FLASK FRAMEWORK**

main.py

from flask import render\_template,Flask,request

import pickle

from sklearn.neighbors import KNeighborsClassifier

from sklearn.model\_selection import train\_test\_split

import pandas as pd

import numpy as np

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

from sklearn.tree import DecisionTreeClassifier

from sklearn.naive\_bayes import GaussianNB

dt = pd.read\_csv(r"C:/Users/NIVEDITHA/Downloads/Crop\_recommendation.csv")

# Create feature and target arrays

train=dt['rainfall']

target=dt['label']

train=np.array(train)

target=np.array(target)

# Split into training and test set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

train,target, test\_size = 0.3, random\_state=1)

knn = GaussianNB()

knn.fit(X\_train.reshape(-1,1), y\_train)

pred=knn.predict(X\_test.reshape(-1,1))

print(accuracy\_score(y\_test,pred))

appl=Flask(\_\_name\_\_)

file=open("model.pkl","rb")

file1=open("model1.pkl","rb")

file2=open("model2.pkl","rb")

file3=open("model3.pkl","rb")

file4=open("model4.pkl","rb")

file5=open("model5.pkl","rb")

random\_Forest=pickle.load(file)

file.close()

random\_Forest1=pickle.load(file1)

file1.close()

random\_Forest2=pickle.load(file2)

file2.close()

random\_Forest3=pickle.load(file3)

file3.close()

random\_Forest4=pickle.load(file4)

file4.close()

random\_Forest5=pickle.load(file5)

file5.close()

#random\_Forest=pickle.load(file)

#file.close()

@appl.route("/", methods=["GET","POST"])

def home():

if request.method=="POST":

myDict = request.form

Month = int(myDict["Month"])

state= (myDict["state"])

pred = [Month]

#stateCall(state)

#res=random\_Forest.predict([pred])[0]

if(state=="TAMILNADU"):

res=random\_Forest.predict([pred])[0]

elif state=="WEST BENGAL":

res=random\_Forest1.predict([pred])[0]

elif(state=="ORISSA"):

res=random\_Forest2.predict([pred])[0]

elif(state=="PUNJAB"):

res=random\_Forest3.predict([pred])[0]

elif(state=="UTTARAKHAND"):

res=random\_Forest4.predict([pred])[0]

else:

res=random\_Forest5.predict([pred])[0]

res=round(res,2)

ans=knn.predict([[res]])[0]

return render\_template('result.html',Month=Month,state=state,res=res,ans=ans)

return render\_template('index.html')

if \_\_name\_\_ == "\_\_main\_\_":

appl.run(debug=True)

Temp.py

import numpy as np

import pandas as pd

import pickle

from sklearn import metrics

data = pd.read\_csv(r"C:/Users/NIVEDITHA/Downloads/rainfall.csv")

# data.head()

data = data.fillna(data.mean())

group = data.groupby('SUBDIVISION')['YEAR','JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC']

dt=group.get\_group(('TAMIL NADU'))

# data.head()

df=dt.melt(['YEAR']).reset\_index()

# df.head()

df= df[['YEAR','variable','value']].reset\_index().sort\_values(by=['YEAR','index'])

# df.head()

df.columns=['Index','Year','Month','Avg\_Rainfall']

Month\_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,

'OCT':10,'NOV':11,'DEC':12}

df['Month']=df['Month'].map(Month\_map)

# df.head(12)

df.drop(columns="Index",inplace=True)

X=np.asanyarray(df[['Month']]).astype('int')

y=np.asanyarray(df['Avg\_Rainfall']).astype('int')

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=10)

from sklearn.ensemble import RandomForestRegressor

random\_forest\_model = RandomForestRegressor(max\_depth=100, max\_features='sqrt', min\_samples\_leaf=4,

min\_samples\_split=10, n\_estimators=800)

random\_forest\_model.fit(X\_train, y\_train)

#--------------------------------WEST BENGAL----------------------------------#

dt1=group.get\_group(('WEST BENGAL'))

# data.head()

df1=dt1.melt(['YEAR']).reset\_index()

# df.head()

df1= df1[['YEAR','variable','value']].reset\_index().sort\_values(by=['YEAR','index'])

# df.head()

df1.columns=['Index','Year','Month','Avg\_Rainfall']

Month\_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,

'OCT':10,'NOV':11,'DEC':12}

df1['Month']=df1['Month'].map(Month\_map)

# df.head(12)

df1.drop(columns="Index",inplace=True)

X1=np.asanyarray(df1[['Month']]).astype('int')

y1=np.asanyarray(df1['Avg\_Rainfall']).astype('int')

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X1, y1, test\_size=0.3, random\_state=10)

random\_forest\_model1 = RandomForestRegressor(max\_depth=100, max\_features='sqrt', min\_samples\_leaf=4,

min\_samples\_split=10, n\_estimators=800)

random\_forest\_model1.fit(X\_train, y\_train)

#y\_predict = random\_forest\_model.predict(X\_test)

#print('MAE:', metrics.mean\_absolute\_error(y\_test,y\_predict))

# print('MSE:', metrics.mean\_squared\_error(y\_test, y\_predict))

#print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_predict)))

#------------------------------------ORISSA------------------------------------

dt2=group.get\_group(('ORISSA'))

# data.head()

df2=dt2.melt(['YEAR']).reset\_index()

# df.head()

df2= df2[['YEAR','variable','value']].reset\_index().sort\_values(by=['YEAR','index'])

# df.head()

df2.columns=['Index','Year','Month','Avg\_Rainfall']

Month\_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,

'OCT':10,'NOV':11,'DEC':12}

df2['Month']=df2['Month'].map(Month\_map)

# df.head(12)

df2.drop(columns="Index",inplace=True)

X2=np.asanyarray(df2[['Month']]).astype('int')

y2=np.asanyarray(df2['Avg\_Rainfall']).astype('int')

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X2, y2, test\_size=0.3, random\_state=10)

random\_forest\_model2 = RandomForestRegressor(max\_depth=100, max\_features='sqrt', min\_samples\_leaf=4,

min\_samples\_split=10, n\_estimators=800)

random\_forest\_model2.fit(X\_train, y\_train)

#-------------------------------------------PUNJAB----------------------------------------------

#group3= data.groupby('SUBDIVISION')['YEAR','JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC']

dt3=group.get\_group(("PUNJAB"))

# data.head()

df3=dt3.melt(['YEAR']).reset\_index()

# df.head()

df3= df3[['YEAR','variable','value']].reset\_index().sort\_values(by=['YEAR','index'])

# df.head()

df3.columns=['Index','Year','Month','Avg\_Rainfall']

Month\_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,

'OCT':10,'NOV':11,'DEC':12}

df3['Month']=df3['Month'].map(Month\_map)

# df.head(12)

df3.drop(columns="Index",inplace=True)

X3=np.asanyarray(df3[['Month']]).astype('int')

y3=np.asanyarray(df3['Avg\_Rainfall']).astype('int')

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X3, y3, test\_size=0.3, random\_state=10)

random\_forest\_model3 = RandomForestRegressor(max\_depth=100, max\_features='sqrt', min\_samples\_leaf=4,

min\_samples\_split=10, n\_estimators=800)

random\_forest\_model3.fit(X\_train, y\_train)

#--------------------------------------------------UTTARAKHAND---------------------

dt4=group.get\_group(('UTTARAKHAND'))

# data.head()

df4=dt4.melt(['YEAR']).reset\_index()

# df.head()

df4= df4[['YEAR','variable','value']].reset\_index().sort\_values(by=['YEAR','index'])

# df.head()

df4.columns=['Index','Year','Month','Avg\_Rainfall']

Month\_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,

'OCT':10,'NOV':11,'DEC':12}

df4['Month']=df4['Month'].map(Month\_map)

# df.head(12)

df4.drop(columns="Index",inplace=True)

X4=np.asanyarray(df4[['Month']]).astype('int')

y4=np.asanyarray(df4['Avg\_Rainfall']).astype('int')

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X4,y4, test\_size=0.3, random\_state=10)

random\_forest\_model4 = RandomForestRegressor(max\_depth=100, max\_features='sqrt', min\_samples\_leaf=4,

min\_samples\_split=10, n\_estimators=800)

random\_forest\_model4.fit(X\_train, y\_train)

#-------------------------------------JAMMU & KASHMIR--------------------------------------

dt5=group.get\_group(('JAMMU & KASHMIR'))

# data.head()

df5=dt5.melt(['YEAR']).reset\_index()

# df.head()

df5= df5[['YEAR','variable','value']].reset\_index().sort\_values(by=['YEAR','index'])

# df.head()

df5.columns=['Index','Year','Month','Avg\_Rainfall']

Month\_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,

'OCT':10,'NOV':11,'DEC':12}

df5['Month']=df5['Month'].map(Month\_map)

# df.head(12)

df5.drop(columns="Index",inplace=True)

X5=np.asanyarray(df5[['Month']]).astype('int')

y5=np.asanyarray(df5['Avg\_Rainfall']).astype('int')

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X5, y5, test\_size=0.3, random\_state=10)

random\_forest\_model5 = RandomForestRegressor(max\_depth=100, max\_features='sqrt', min\_samples\_leaf=4,

min\_samples\_split=10, n\_estimators=800)

random\_forest\_model5.fit(X\_train, y\_train)

#----------------------------------------------------------------------------------

file = open("model.pkl","wb")

file1=open("model1.pkl","wb")

pickle.dump(random\_forest\_model,file)

pickle.dump(random\_forest\_model1,file1)

file.close()

file1.close()

file2 = open("model2.pkl","wb")

file3=open("model3.pkl","wb")

pickle.dump(random\_forest\_model2,file2)

pickle.dump(random\_forest\_model3,file3)

file2.close()

file3.close()

file4 = open("model4.pkl","wb")

file5=open("model5.pkl","wb")

pickle.dump(random\_forest\_model4,file4)

pickle.dump(random\_forest\_model5,file5)

file4.close()

file5.close()

# print(y\_predict)

'''def stateCall(state):

dt=group.get\_group((state))# data.head()

df=dt.melt(['YEAR']).reset\_index()

# df.head()

df= df[['YEAR','variable','value']].reset\_index().sort\_values(by=['YEAR','index'])

# df.head()

df.columns=['Index','Year','Month','Avg\_Rainfall']

Month\_map={'JAN':1,'FEB':2,'MAR' :3,'APR':4,'MAY':5,'JUN':6,'JUL':7,'AUG':8,'SEP':9,

'OCT':10,'NOV':11,'DEC':12}

df['Month']=df['Month'].map(Month\_map)

# df.head(12)

df.drop(columns="Index",inplace=True)

X=np.asanyarray(df[['Month']]).astype('int')

y=np.asanyarray(df['Avg\_Rainfall']).astype('int')

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=10)

from sklearn.ensemble import RandomForestRegressor

random\_forest\_model = RandomForestRegressor(max\_depth=100, max\_features='sqrt', min\_samples\_leaf=4,

min\_samples\_split=10, n\_estimators=800)

random\_forest\_model.fit(X\_train, y\_train)

file = open("model.pkl","wb")

pickle.dump(random\_forest\_model,file)

file.close()'''