SPRINT 4

Team ID	PNT2022TMID30456
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()'''
```