# Import Dependencies

%matplotlib inline

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

# Read CSV file

file = 'Global.csv'

df = pd.read\_csv(file, encoding="ISO-8859-1")

# Separate last digit from Latitude and Longitude to identify direction

df["Dir\_Lat"] = df["Latitude"].str[-1]

df["Dir\_Long"] = df["Longitude"].str[-1]

Latitude=df["Latitude"].copy()

df["Latitude"]=Latitude.str[:-1]

Longitude=df["Longitude"].copy()

df["Longitude"]=Longitude.str[:-1]

df

# Replace direction string to numeric that can be used in a model

df.loc[(df.Dir\_Lat == 'N'),'Dir\_Lat']=1

df.loc[(df.Dir\_Lat == 'S'),'Dir\_Lat']=2

df.loc[(df.Dir\_Long == 'W'),'Dir\_Long']=1

df.loc[(df.Dir\_Long == 'E'),'Dir\_Long']=2

df

# Select columns needed

df=df[["Date", "AverageTemperature","Latitude","Longitude","Dir\_Lat","Dir\_Long"]]

df

# Calculate Year, Month and Day based on date

df['Date'] = pd.to\_datetime(df['Date'])

df['Year'] = df['Date'].dt.year

df['Month'] = df['Date'].dt.month

df['Day'] = df['Date'].dt.day

df=df[["Year","Month","Day", "AverageTemperature","Latitude","Longitude","Dir\_Lat","Dir\_Long"]]

df

# Save csv file

df.to\_csv('Globaldata.csv')

# Assign the data to X and y

X = df[["Year", "Month", "Day","Latitude","Longitude","Dir\_Lat","Dir\_Long"]]

y = df[["AverageTemperature"]]

print(y)

# Import dependencies

from sklearn.datasets import make\_regression

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

# Use train\_test\_split to create training and testing data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=20)

print(X\_train.shape,y\_train.shape,X\_test.shape,y\_test.shape)

y.shape

# Import RandomForest

from sklearn.ensemble import RandomForestRegressor

regr = RandomForestRegressor(max\_depth=15, random\_state=20)

train=regr.fit(X\_train, y\_train)

#regr.score(X\_train, y\_train)

print(train)

# Result for Predict Train

res=train.predict([[1985, 11, 1, '12.05', '14.79', 1, 1]])

print(y\_test.loc[133940,:][0], res)

train.score(X\_test.values,y\_test.values)

train.score(X\_train.values,y\_train.values)

# Import pickle and open .sav file

import pickle

filename = 'Tempregression.sav'

pickle.dump(train, open(filename, 'wb'))

# load the model from disk

loaded\_model = pickle.load(open(filename, 'rb'))

result = loaded\_model.score(X\_test, y\_test)

print(result)

loaded\_model = pickle.load(open(filename, 'rb'))

res=loaded\_model.predict([[2030, 11, 1, '12.05', '14.79', 1, 1]])

print(y\_test.loc[133940,:][0], res)