from sklearn.svm import SVC, LinearSVC

from sklearn.preprocessing import MinMaxScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.externals import joblib

import numpy as np

import pandas as pd

import pickle

import os

from sklearn.metrics import accuracy\_score

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

root = os.path.dirname(\_\_file\_\_)

path\_df = os.path.join(root, 'recons\_dataset/combined\_dataset.csv')

data = pd.read\_csv(path\_df)

scaler = MinMaxScaler()

train, test = train\_test\_split(data, test\_size=0.25)

X\_train = train.drop('num', axis=1)

Y\_train = train['num']

X\_test = test.drop('num', axis=1)

Y\_test = test['num']

# We don't scale targets: Y\_test, Y\_train as SVC returns the class labels not probability values

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.fit\_transform(X\_test)

clf = RandomForestClassifier()

# Training the classifier

clf.fit(X\_train, Y\_train)

# Testing model accuracy. Average is taken as test set is very small hence accuracy varies a lot everytime the model is trained

acc = 0

acc\_binary = 0

for i in range(0, 20):

Y\_hat = clf.predict(X\_test)

acc = acc + accuracy\_score(Y\_hat, Y\_test)

print("Average test Accuracy:{}".format(acc/20))

# Saving the trained model for inference

model\_path = os.path.join(root, 'models/rfc.sav')

joblib.dump(clf, model\_path)

# Saving the scaler object

scaler\_path = os.path.join(root, 'models/scaler.pkl')

with open(scaler\_path, 'wb') as scaler\_file:

pickle.dump(scaler, scaler\_file)