epochs = 2

batch\_size = 64

opt = 'adam'

model = Sequential()

model.add(Reshape((1,X\_train.shape[1],1)))

model.add(Conv2D(filters=32,kernel\_size=(1,5),padding='same',activation='relu',input\_shape=(1,X\_train.shape[1],1)))

#model.add(MaxPooling2D(pool\_size=(1,6),strides=(1,2)))

model.add(MaxPooling2D(pool\_size=(1,3),strides=(1,2)))

model.add(Flatten())

model.add(Dense(500,activation='relu'))

model.add(Dense(1,activation='linear'))

model.compile(loss='mean\_absolute\_percentage\_error',optimizer=opt,metrics=['accuracy'])

history = model.fit(X\_train,y\_train,validation\_data=(X\_test,y\_test),epochs=epochs, batch\_size=batch\_size)

SC=StandardScaler()

X\_train\_scale=SC.fit\_transform(X\_train)

X\_test\_scale=SC.transform(X\_test)

rf\_clf = kneighboursclassifiers()

rf\_clf.fit(X\_train\_scale,y\_train)

X\_pred=rf\_clf.predict(X\_test\_scale)

print(X\_pred)

rf\_clf.score(X\_test\_scale,y\_test)