Task 8.1: Forecasting model

Data pre-processing:

```
from sklearn import preprocessing
le=preprocessing.LabelEncoder()
train_x.iloc[:,1]=le.fit_transform(train_x.iloc[:,1].values)
```

```
scaler = MinMaxScaler()
```

```
train_x.iloc[:,1:]=scaler.fit_transform(train_x.iloc[:,1:].values)
```

```
C:\Users\sagar\anaconda3\lib\site-packages\sklearn\utils\validation.py:590: DataConversionWarning: Data with input dtype int64
was converted to float64 by MinMaxScaler.
  warnings.warn(msg, DataConversionWarning)
```

```
train_size = int(len(dataset) * 0.67)
test_size = len(dataset) - train_size
print(train_size)
```

```
30897
```

```
test_size = len(dataset) - train_size
print(test_size)
```

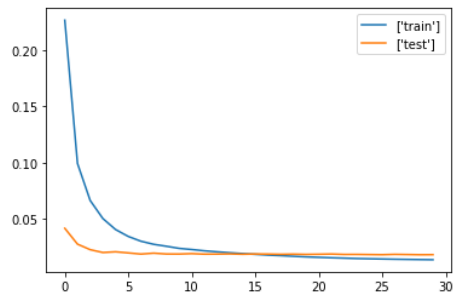
SimpleRNN:

```
reg.add(SimpleRNN(units=100, input_shape=(x_train.shape[1],x_train.shape[2]),name='simple_RNN'))
reg.add(Dropout(0.5,name = 'drp1'))
reg.add(Dense(units=10,name = 'fc1'))
reg.add(Dropout(0.2,name = 'drp2'))
reg.add(Dense(units=1,name='output'))

adm=Adam(lr=0.0001)
reg.compile(optimizer=adm, loss='mae')
reg.summary()
model=reg.fit(x_train,y_train,epochs=30,batch_size=100,validation_split=0.20)
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
simple_RNN (SimpleRNN)       (None, 100)               10300
_________________________________________________________________
drp1 (Dropout)               (None, 100)               0
_________________________________________________________________
fc1 (Dense)                  (None, 10)                1010
_________________________________________________________________
drp2 (Dropout)               (None, 10)                0
_________________________________________________________________
output (Dense)               (None, 1)                 11
=================================================================
Total params: 11,321
Trainable params: 11,321
Non-trainable params: 0
_________________________________________________________________
WARNING:tensorflow:From C:\Users\sagar\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:422: The name tf.global_
variables is deprecated. Please use tf.compat.v1.global_variables instead.

Train on 36868 samples, validate on 9218 samples
Epoch 1/30
36868/36868 [==============================] - 9s 248us/step - loss: 0.2270 - val_loss: 0.0414
Epoch 2/30
36868/36868 [==============================] - 9s 231us/step - loss: 0.0992 - val_loss: 0.0272
Epoch 3/30
36868/36868 [==============================] - 9s 231us/step - loss: 0.0662 - val_loss: 0.0223
Epoch 4/30
36868/36868 [==============================] - 9s 231us/step - loss: 0.0500 - val_loss: 0.0198
Epoch 5/30
36868/36868 [==============================] - 9s 233us/step - loss: 0.0403 - val_loss: 0.0204
```

```
pred=reg.predict(x_train)
err=np.mean(np.abs(y_train-pred))
print("MAE error for standard averaging:",err)
```

```
MAE error for standard averaging: 0.013902409068076065
```




```
pred=reg.predict(x_train)
err=np.mean(np.abs(y_train-pred))
print("MAE error for standard averaging:",err)
```

```
MAE error for standard averaging: 0.013902409068076065
```

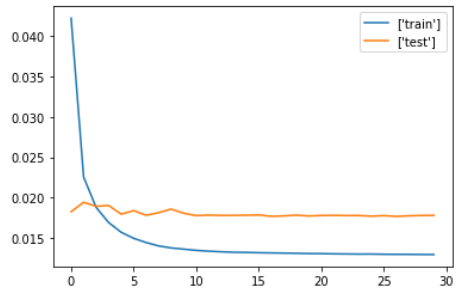
LSTM:

```
reg=Sequential()
print("x=",x_train.shape[1])
reg.add(LSTM(units=100, input_shape=(x_train.shape[1],x_train.shape[2]),name='LSTM'))
reg.add(Dropout(0.5,name = 'drp1'))
reg.add(Dense(units=10,name = 'fc1'))
reg.add(Dropout(0.2,name = 'drp2'))
reg.add(Dense(units=1,name='fc2'))
adm=Adam(lr=0.0001)
reg.compile(optimizer=adm, loss='mae')
reg.summary()
model=reg.fit(x_train,y_train,epochs=30,batch_size=100,validation_split=0.20)
```

```
x= 30
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
LSTM (LSTM)                  (None, 100)               41200
_________________________________________________________________
drp1 (Dropout)               (None, 100)               0
_________________________________________________________________
fc1 (Dense)                  (None, 10)                1010
_________________________________________________________________
drp2 (Dropout)               (None, 10)                0
_________________________________________________________________
fc2 (Dense)                  (None, 1)                 11
=================================================================
Total params: 42,221
Trainable params: 42,221
Non-trainable params: 0
_________________________________________________________________
Train on 36868 samples, validate on 9218 samples
Epoch 1/30
36868/36868 [==============================] - 13s 347us/step - loss: 0.0422 - val_loss: 0.0182
Epoch 2/30
36868/36868 [==============================] - 12s 337us/step - loss: 0.0225 - val_loss: 0.0194
Epoch 3/30
36868/36868 [==============================] - 12s 335us/step - loss: 0.0188 - val_loss: 0.0189
Epoch 4/30
36868/36868 [==============================] - 12s 335us/step - loss: 0.0169 - val_loss: 0.0190
Epoch 5/30
36868/36868 [==============================] - 12s 337us/step - loss: 0.0157 - val_loss: 0.0179
Epoch 6/30
```

```
pred=reg.predict(x_train)
err=np.mean(np.abs(y_train-pred))
print("MAE error for standard averaging:",err)
```

```
MAE error for standard averaging: 0.013870752687459686
```




```
pred=reg.predict(x_train)
err=np.mean(np.abs(y_train-pred))
print("MAE error for standard averaging:",err)
```

```
MAE error for standard averaging: 0.013870752687459686
```

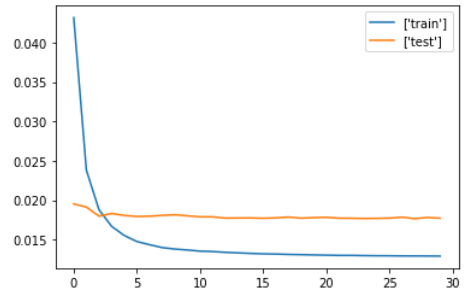
GRU:

```
reg=Sequential()
print("x=",x_train.shape[1])
reg.add(GRU(units=100, input_shape=(x_train.shape[1],x_train.shape[2]),name='GRU'))
reg.add(Dropout(0.5,name = 'drp1'))
reg.add(Dense(units=10,name = 'fc1'))
reg.add(Dropout(0.2,name = 'drp2'))
reg.add(Dense(units=1,name='fc2'))
adm=Adam(lr=0.0001)
reg.compile(optimizer=adm, loss='mae')
reg.summary()
model=reg.fit(x_train,y_train,epochs=30,batch_size=100,validation_split=0.20)
```

```
x= 30
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
GRU (GRU)                    (None, 100)               30900
_________________________________________________________________
drp1 (Dropout)               (None, 100)               0
_________________________________________________________________
fc1 (Dense)                  (None, 10)                1010
_________________________________________________________________
drp2 (Dropout)               (None, 10)                0
_________________________________________________________________
fc2 (Dense)                  (None, 1)                 11
=================================================================
Total params: 31,921
Trainable params: 31,921
Non-trainable params: 0
_________________________________________________________________
Train on 36868 samples, validate on 9218 samples
Epoch 1/30
36868/36868 [==============================] - 15s 401us/step - loss: 0.0431 - val_loss: 0.0195
Epoch 2/30
36868/36868 [==============================] - 14s 392us/step - loss: 0.0238 - val_loss: 0.0191
Epoch 3/30
```

```
pred=reg.predict(x_train)
err=np.mean(np.abs(y_train-pred))
print("MAE error for standard averaging:",err)
```

```
MAE error for standard averaging: 0.013842379968173548
```




```
pred=reg.predict(x_train)
err=np.mean(np.abs(y_train-pred))
print("MAE error for standard averaging:",err)
```

```
MAE error for standard averaging: 0.013842379968173548
```

All the above plots are based on the validation size of 0.20.