

✓ Google stock price prediction using RNN

```
# import the library
import numpy as np
import pandas as pd

# load the train data
df = pd.read_csv('Google_Stock_Price_Train.csv')
train_data = df.iloc[:,1:2].values # to get open column in array form
train_data[:5]

array([[325.25],
       [331.27],
       [329.83],
       [328.34],
       [322.04]])

# Feature scaling
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range=(0,1))
train_data_scaled = sc.fit_transform(train_data)
train_data_scaled[:5]

array([[0.08581368],
       [0.09701243],
       [0.09433366],
       [0.09156187],
       [0.07984225]])

#shape of dataset
df.shape

(1258, 6)

# X(Feature) and y(label)
X_train = [] # create an empty list
y_train = [] # create an empty list
for i in range(60,1258): # 1258 is the number of rows
    X_train.append(train_data_scaled[i-60:i,0]) # first 60 timesteps as feature
    y_train.append(train_data_scaled[i,0]) # 61st timestep as target
X_train,y_train = np.array(X_train), np.array(y_train) # to convert to array for feeding in our LSTM model

# reshape the data
X_train = np.reshape(X_train,(X_train.shape[0],X_train.shape[1],1)) # for feeding into LSTM model
X_train.shape

(1198, 60, 1)

# Build the model
# Import the libraries
from keras.models import Sequential
from keras.layers import Dense , LSTM , Dropout

# intialise model
model = Sequential()
# first layer
model.add(LSTM(units=50,return_sequences=True , input_shape=(X_train.shape[1],1)))
model.add(Dropout(0.2))
# second layer
model.add(LSTM(units=50,return_sequences=True))
model.add(Dropout(0.2))
# third layer
model.add(LSTM(units=50,return_sequences=True))
model.add(Dropout(0.2))
# fourth layer
model.add(LSTM(units=50))
model.add(Dropout(0.2))
# output layer
model.add(Dense(units=1))
```

