```
import pandas as pd
import numpy as np
import pandas_datareader as web
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
df1=web.DataReader('SBIN.NS',data_source='yahoo', start='2012-01-01',end='2020-01-27')
ndf1=df1[['Close']]
df=ndf1
# forecast column 'Y'
forecast_out=2
df['Prediction']=df[['Close',]].shift(-forecast_out)
print (df.tail(6))
# Create just the independent column x
x=np.array(df[['Close']])
#remove forecast_out num of rows
x=x[:-forecast_out]
#x2=np.array(df[['Closing Crude Price']]) .Later
# Create the dependent column Y
Y=np.array(df[['Prediction']])
#remove forecast_out num of rows
Y=Y[:-forecast_out]
# split the data into 80%train and test 20%
x_train, x_test,Y_train, Y_test=train_test_split(x,Y,test_size=0.2)
print(Y_train.shape)
# Craete and train the Support Vector Machine Regressor
svr_rbf=SVR(kernel='rbf', C=1e3, gamma=0.1)
svr_rbf.fit(x_train, Y_train)
#Testing model ,score  prints the coeffeicient of determination R^2
# best possible score is 1.0
svm_confidence= svr_rbf.score(x_test, Y_test)
print ("SVR confidence level" ,svm_confidence)
# Create and train the linear regression model
lr=LinearRegression()
lr=lr.fit(x_train,Y_train)
#Testing model ,score  prints the coeffeicient of determination R^2
# best possible score is 1.0
lr_confidence=lr.score(x_test,Y_test)
print ("lr confidence level :", lr_confidence)

# Predict for the forecast out days for now set 30
#set a new variable x_forecast and get n days of x data
x_forecast=np.array(df.drop(['Prediction'],1))[-forecast_out:]
print ("Printing x forecast variable")
print (x_forecast)
#Print the predictions for the next 30 days using Linear regression
lr_predicted=lr.predict(x_forecast)
```

```
print ("predictions by linear regressor ",lr_predicted)
#Print the predictions for the next n days using SVM Regression
svr_predicted=svr_rbf.predict(x_forecast)
print ("Predictions by SVM Regressor" ,svr_predicted)
```