



# STATISTICAL ARBITRAGE MODEL

FOR NATIONAL STOCK EXCHANGE (NSE) DATA

# Data Pre-Processing

## TIME SERIES ANALYSIS:

```
#Time Series Analysis
start16 = datetime(2016, 1, 1)
end16 = datetime(2016, 12, 31)
stamp16 = pd.date_range(start16, end16)

start17 = datetime(2017, 1, 1)
end17 = datetime(2017, 12, 31)
stamp17 = pd.date_range(start17, end17)

stocks['Date'] = pd.to_datetime(stocks.TIMESTAMP, format='%Y-%m-%d')
stocks.index = stocks['Date']
```

# Data Pre-Processing

## SEPERATING TRAIN & TEST DATA:

```
#Seperating Train and test data
train = []
test = []
for index, rows in stocks.iterrows():
    if index in stamp16:
        train.append(list(rows))
    if index in stamp17:
        test.append(list(rows))

train = pd.DataFrame(train, columns = stocks.columns)
test = pd.DataFrame(test, columns = stocks.columns)
```

# Data Pre-Processing

## EXTRACTING DATA:

```
#Pre-Processing Train Data
X_train = train[['HL_PCT', 'OPEN', 'TOTTRDQTY', 'TOTTRDVAL', 'TOTALTRADES']]
x_train = X_train.to_dict(orient='records')
vec = DictVectorizer()
X = vec.fit_transform(x_train).toarray()
Y = np.asarray(train.CLOSE)
Y = Y.astype('int')

#Pre-Processing Test data
X_test = test[['HL_PCT', 'OPEN', 'TOTTRDQTY', 'TOTTRDVAL', 'TOTALTRADES']]
x_test = X_test.to_dict(orient='records')
vec = DictVectorizer()
x = vec.fit_transform(x_test).toarray()
y = np.asarray(test.CLOSE)
y = y.astype('int')
```

# Classification:

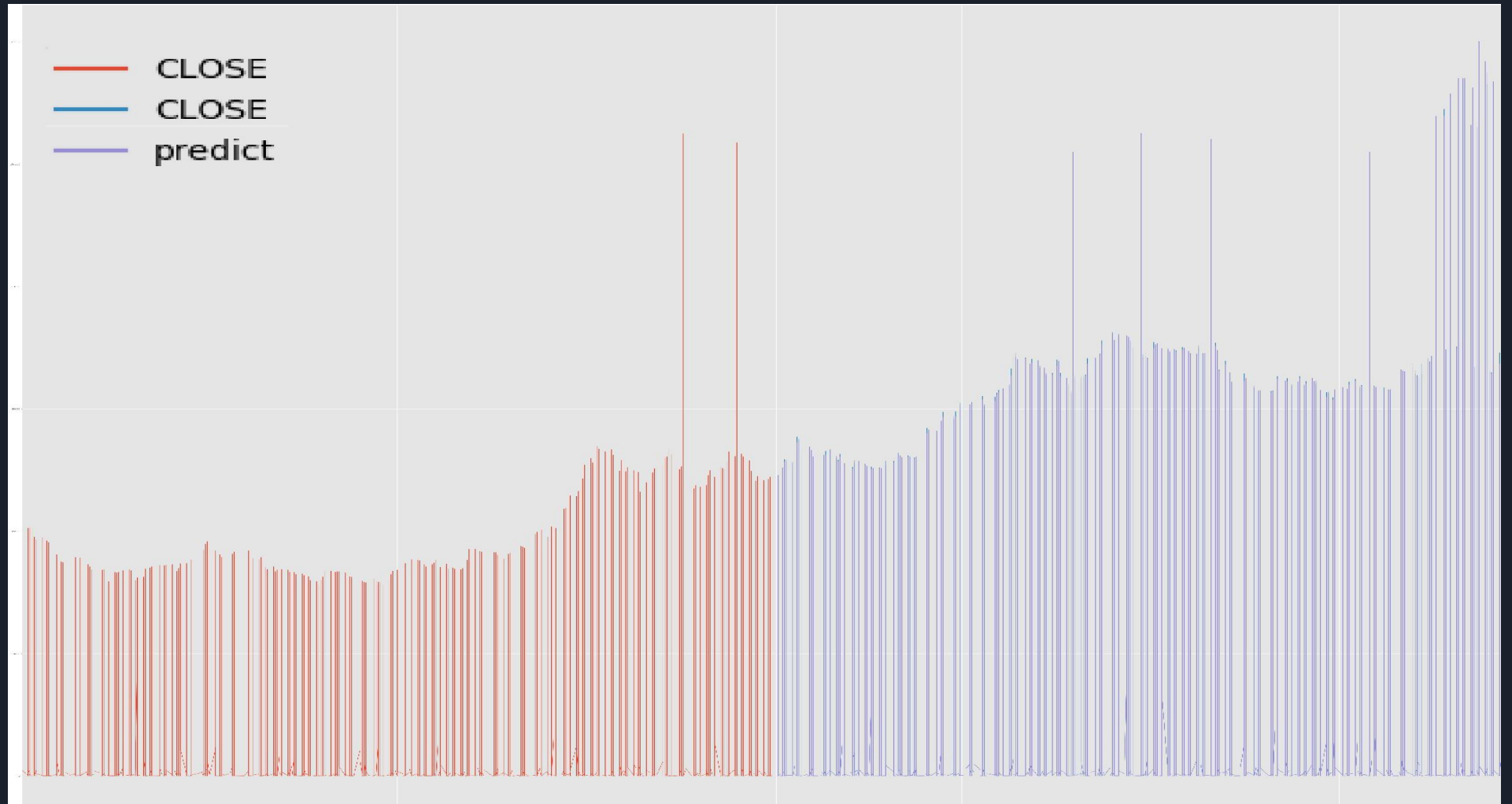
In simple linear regression, we predict scores on one variable from the scores on a second variable. The variable we are predicting is called the *criterion variable* and is referred to as Y

```
#Classifier
from sklearn.linear_model import LinearRegression
clf = LinearRegression()
clf.fit(X, Y)
print("Accuracy of this Statistical Arbitrage model is: ",clf.score(x,y))
predict = clf.predict(x)

test['predict'] = predict
```

# Plotting the prediction over 2017 dataset:

```
#Plotting
train.index = train.Date
test.index = test.Date
train['CLOSE'].plot()
test['CLOSE'].plot()
test['predict'].plot()
plt.legend(loc='best')
plt.xlabel('Date')
plt.ylabel('Price')
plt.show()
```



**Thank You!**

ABHISHEK KUMAR