```
%matplotlib inline
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

```
from pathlib import Path
path = Path('/content/drive/My Drive/assignment_2/exp_train.csv')
path_test = Path('/content/drive/My Drive/assignment_2/exp_test.csv')
```

```
import pandas as pd
names =[
't', # Time (secs)
'q1', 'q2', 'q3', # Joint angle
'dq1', 'dq2', 'dq3', # Joint velocity
'I1', 'I2', 'I3', # Motor current (A)
'eps21', 'eps22', 'eps31', 'eps32', # Strain measurements
'ddq1', 'ddq2', 'ddq3' # Joint accelerations
]
df = pd.read_csv(path, header=None,sep=',',
names=names, index_col=0)
df.head(6)
```

```
             q1      q2       q3           dq1            dq2            dq3         I1
   t
0.00  -0.000007  2.4958  -1.1345  -7.882100e-21  -4.940656e-321   3.913100e-29  -0.081623  -0
0.01  -0.000007  2.4958  -1.1345  -2.258200e-21  -4.940656e-321   2.626200e-31  -0.037411  -0
0.02  -0.000007  2.4958  -1.1345  -6.469800e-22  -4.940656e-321   1.762500e-33  -0.066319  -0
0.03  -0.000007  2.4958  -1.1345  -1.853600e-22  -4.940656e-321   1.182800e-35  -0.068020  -0
0.04  -0.000007  2.4958  -1.1345  -5.310600e-23  -4.940656e-321  -5.270900e-03  -0.052715  -0
0.05  -0.000007  2.4958  -1.1345  -1.521500e-23  -4.940656e-321   3.252600e-04  -0.088425  -0
```

```
y = df['I2'];y.shape
```

```
(8000,)
```

```
X = df[['q2', 'dq2', 'eps21', 'eps22', 'eps31', 'eps32', 'ddq2']];X.shape
```

```
(8000, 7)
```

```
from sklearn import metrics, linear_model
regr = linear_model.LinearRegression()
```

```
regr.fit(X, y)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
names =[
't', # Time (secs)
'q1', 'q2', 'q3', # Joint angle
'dq1', 'dq2', 'dq3', # Joint velocity
'I1', 'I2', 'I3', # Motor current (A)
'eps21', 'eps22', 'eps31', 'eps32', # Strain measurements
'ddq1', 'ddq2', 'ddq3' # Joint accelerations
]
df_test = pd.read_csv(path_test, header=None,sep=',',
names=names, index_col=0)
df_test.head(6)
```

	q1	q2	q3	dq1	dq2	dq3	I1	I2
t								
0.00	-0.000007	1.9024	0.26063	-0.000364	4.940656e-321	0.012596	-0.096928	-0.15134
0.01	0.000013	1.9024	0.26073	0.000739	4.940656e-321	0.012095	-0.028908	-0.11903
0.02	-0.000007	1.9024	0.26086	-0.000580	4.940656e-321	0.011596	-0.059517	-0.13944
0.03	0.000013	1.9024	0.26099	0.001409	4.940656e-321	0.013933	-0.079923	-0.15304
0.04	-0.000007	1.9024	0.26110	-0.001273	4.940656e-321	0.010793	-0.025507	-0.12924
0.05	-0.000007	1.9024	0.26124	0.001928	4.940656e-321	0.011915	-0.083324	-0.14964

```
y_test = df_test['I2']
X_test = df_test[['q2', 'dq2', 'eps21', 'eps22', 'eps31', 'eps32', 'ddq2']]
```

```
y_pred = regr.predict(X_test)
```

```
from sklearn.metrics import mean_squared_error, r2_score
print('Coefficients: \n', regr.coef_)
print('\n')
print('Mean squared error for testing dataset: %.2f'% mean_squared_error(y_test, y_pred))
print('\n')
print('R-Squared value: %.2f'% r2_score(y_test, y_pred))
```

```
Coefficients:
 [ 0.06255018  0.20584896  0.00118784  0.00044457 -0.0031362   0.00603298
  0.05487097]


Mean squared error for testing dataset: 0.01


R-Squared value: 0.87
```