Importing Packages

In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sea
```

In [49]:
```python
df=pd.read_csv('Friction Stir Journal dataset.csv')
```

In [51]:
```python
df
```

Out[51]:

	Exp. N o	Tool material [Type]	Rotational speed [RPM]	Welding speed [mm/min]	Axial force [KN]	UTS [MPa]
0	1	H13	900	25	2	251
1	2	H13	900	25	2	254
2	3	H13	900	25	2	257
3	4	H13	1200	35	3	264
4	5	H13	1200	35	3	260
5	6	H13	1200	35	3	268
6	7	H13	1500	45	4	284
7	8	H13	1500	45	4	284
8	9	H13	1500	45	4	281
9	10	H13	900	35	4	242
10	11	H13	900	35	4	244
11	12	H13	900	35	4	241
12	13	H13	1200	45	2	264
13	14	H13	1200	45	2	264
14	15	H13	1200	45	2	260
15	16	H13	1500	25	3	288
16	17	H13	1500	25	3	288
17	18	C40	1500	25	3	286
18	19	C40	900	45	3	238
19	20	C40	900	45	3	231
20	21	C40	900	45	3	236
21	22	C40	1200	25	4	271
22	23	C40	1200	25	4	268
23	24	C40	1200	25	4	273
24	25	C40	1500	35	2	281
25	26	C40	1500	35	2	278
26	27	C40	1500	35	2	280
27	28	C40	900	25	2	248
28	29	C40	900	25	2	248
29	30	C40	900	25	2	245
30	31	C40	1200	35	3	258
31	32	C40	1200	35	3	257
32	33	C40	1200	35	3	254
33	34	C40	1500	45	4	281
34	35	HSS	1500	45	4	286
35	36	HSS	1500	45	4	285
36	37	HSS	900	35	4	248
37	38	HSS	900	35	4	246
38	39	HSS	900	35	4	247
39	40	HSS	1200	45	2	266
40	41	HSS	1200	45	2	264
41	42	HSS	1200	45	2	269
42	43	HSS	1500	25	3	291
43	44	HSS	1500	25	3	292
44	45	HSS	1500	25	3	291
45	46	HSS	900	45	3	239
46	47	HSS	900	45	3	242
47	48	HSS	1200	25	4	276
48	49	HSS	1200	25	4	274
49	50	HSS	1500	35	2	286
50	51	HSS	1500	35	2	285
51	52	HSS	1500	35	2	285

In [ ]:
```python
df_toolmaterial = pd.get_dummies(df['Tool material [Type]'],axis=1)
```

In [54]:
```python
df.drop('Tool material [Type]',axis=1,inplace=True)
```

In [62]:
```python
new_df=pd.concat([df,df_toolmaterial],axis=1)
```

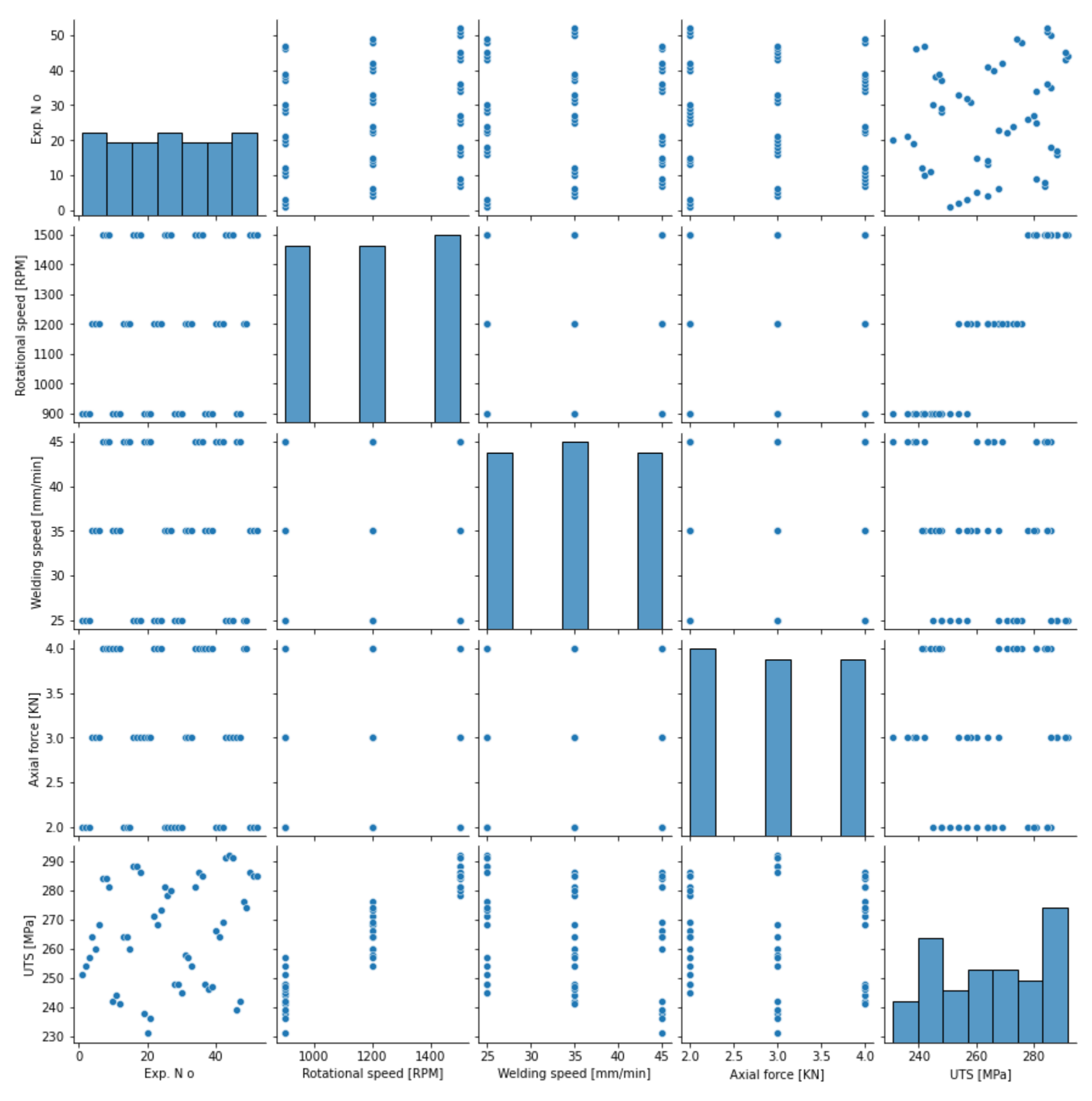
In [93]:
```python
new_df
```

Out[93]:

	Rotational speed [RPM]	Welding speed [mm/min]	Axial force [KN]	UTS [MPa]	C40	H13	HSS
0	900	25	2	251	0	1	0
1	900	25	2	254	0	1	0
2	900	25	2	257	0	1	0
3	1200	35	3	264	0	1	0
4	1200	35	3	260	0	1	0
5	1200	35	3	268	0	1	0
6	1500	45	4	284	0	1	0
7	1500	45	4	284	0	1	0
8	1500	45	4	281	0	1	0
9	900	35	4	242	0	1	0
10	900	35	4	244	0	1	0
11	900	35	4	241	0	1	0
12	1200	45	2	264	0	1	0
13	1200	45	2	264	0	1	0
14	1200	45	2	260	0	1	0
15	1500	25	3	288	0	1	0
16	1500	25	3	288	0	1	0
17	1500	25	3	286	1	0	0
18	900	45	3	238	1	0	0
19	900	45	3	231	1	0	0
20	900	45	3	236	1	0	0
21	1200	25	4	271	1	0	0
22	1200	25	4	268	1	0	0
23	1200	25	4	273	1	0	0
24	1500	35	2	281	1	0	0
25	1500	35	2	278	1	0	0
26	1500	35	2	280	1	0	0
27	900	25	2	248	1	0	0
28	900	25	2	248	1	0	0
29	900	25	2	245	1	0	0
30	1200	35	3	258	1	0	0
31	1200	35	3	257	1	0	0
32	1200	35	3	254	1	0	0
33	1500	45	4	281	1	0	0
34	1500	45	4	286	0	0	1
35	1500	45	4	285	0	0	1
36	900	35	4	248	0	0	1
37	900	35	4	246	0	0	1
38	900	35	4	247	0	0	1
39	1200	45	2	266	0	0	1
40	1200	45	2	264	0	0	1
41	1200	45	2	269	0	0	1
42	1500	25	3	291	0	0	1
43	1500	25	3	292	0	0	1
44	1500	25	3	291	0	0	1
45	900	45	3	239	0	0	1
46	900	45	3	242	0	0	1
47	1200	25	4	276	0	0	1
48	1200	25	4	274	0	0	1
49	1500	35	2	286	0	0	1
50	1500	35	2	285	0	0	1
51	1500	35	2	285	0	0	1

In [6]:
```python
sea.pairplot(df)
```

Out[6]: <seaborn.axisgrid.PairGrid at 0x2c4c8818bb0>



Define x and y

In [73]:
```python
x= new_df.drop("UTS [MPa]",axis=1).values
y= new_df["UTS [MPa]"].values
```

In [138…
```python
print(x)
```

```
[[ 900   25    2    0    1    0]
 [ 900   25    2    0    1    0]
 [ 900   25    2    0    1    0]
 [1200   35    3    0    1    0]
 [1200   35    3    0    1    0]
 [1200   35    3    0    1    0]
 [1500   45    4    0    1    0]
 [1500   45    4    0    1    0]
 [1500   45    4    0    1    0]
 [ 900   35    4    0    1    0]
 [ 900   35    4    0    1    0]
 [ 900   35    4    0    1    0]
 [1200   45    2    0    1    0]
 [1200   45    2    0    1    0]
 [1200   45    2    0    1    0]
 [1500   25    3    0    1    0]
 [1500   25    3    0    1    0]
 [1500   25    3    1    0    0]
 [ 900   45    3    1    0    0]
 [ 900   45    3    1    0    0]
 [ 900   45    3    1    0    0]
 [1200   25    4    1    0    0]
 [1200   25    4    1    0    0]
 [1200   25    4    1    0    0]
 [1500   35    2    1    0    0]
 [1500   35    2    1    0    0]
 [1500   35    2    1    0    0]
 [ 900   25    2    1    0    0]
 [ 900   25    2    1    0    0]
 [ 900   25    2    1    0    0]
 [1200   35    3    1    0    0]
 [1200   35    3    1    0    0]
 [1200   35    3    1    0    0]
 [1500   45    4    1    0    0]
 [1500   45    4    0    0    1]
 [1500   45    4    0    0    1]
 [ 900   35    4    0    0    1]
 [ 900   35    4    0    0    1]
 [ 900   35    4    0    0    1]
 [1200   45    2    0    0    1]
 [1200   45    2    0    0    1]
 [1200   45    2    0    0    1]
 [1500   25    3    0    0    1]
 [1500   25    3    0    0    1]
 [1500   25    3    0    0    1]
 [ 900   45    3    0    0    1]
 [ 900   45    3    0    0    1]
 [1200   25    4    0    0    1]
 [1200   25    4    0    0    1]
 [1500   35    2    0    0    1]
 [1500   35    2    0    0    1]
 [1500   35    2    0    0    1]]
```

Split the dataset into Training and Test Set

In [80]:
```python
from sklearn.model_selection import train_test_split #importing ML package
```

In [128…
```python
X_train,X_test,y_train,y_test = train_test_split(x,y,test_size = 0.3,random_state=0) #splitting 70% training and 30% for testing
```

Train the model on Training Set

In [129…
```python
from sklearn.linear_model import LinearRegression
ml = LinearRegression()
ml.fit(X_train,y_train)
```

Out[129… LinearRegression()

Predict the test set result

In [130…
```python
y_predict = ml.predict(X_test)
print(y_predict)
```

```
[246.32380658 248.47257606 246.90276404 283.85632575 250.33412689
 262.3        262.46221865 262.3        281.70755627 266.31032031
 246.90276404 267.71791368 262.3        246.32380658 292.70560228
 292.70560228]
```

In [131…
```python
ml.predict([[1200,25,4,0,0,1]])
```

Out[131… array([273.29804601])

Evaluate model

In [132…
```python
from sklearn.metrics import r2_score
```

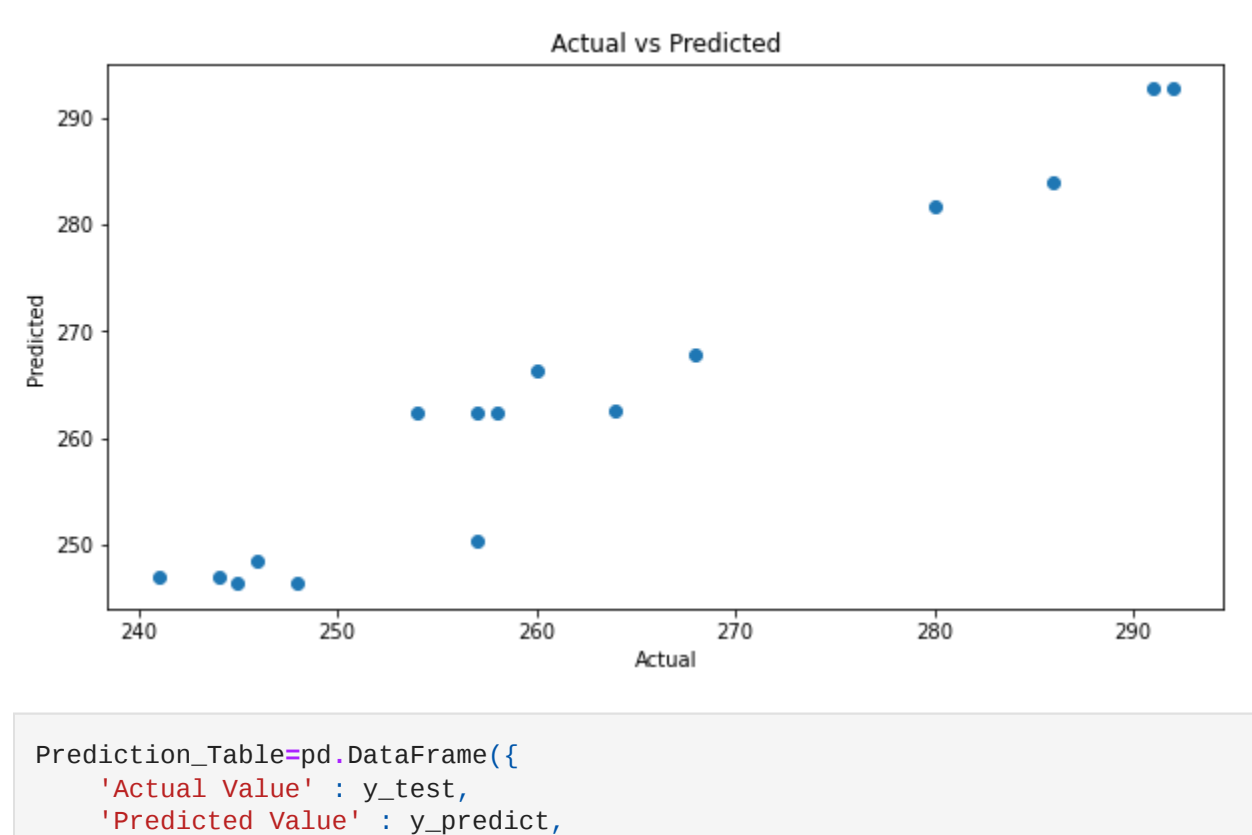
In [133…
```python
r2_score(y_test,y_predict)
```

Out[133… 0.9382007318425754

Plot the results

In [134…
```python
plt.figure(figsize=(10,5))
plt.scatter(y_test,y_predict)
plt.xlabel("Actual")
plt.ylabel("Predicted")
plt.title("Actual vs Predicted")
```

Out[134… Text(0.5, 1.0, 'Actual vs Predicted')



In [135…
```python
Prediction_Table=pd.DataFrame({
    'Actual Value' : y_test,
    'Predicted Value' : y_predict,
    'Difference' : y_test-y_predict
})
```

In [136…
```python
Prediction_Table
```

Out[136…

	Actual Value	Predicted Value	Difference
0	248	246.323807	1.676193
1	246	248.472576	-2.472576
2	241	246.902764	-5.902764
3	286	283.856326	2.143674
4	257	250.334127	6.665873
5	258	262.300000	-4.300000
6	264	262.462219	1.537781
7	254	262.300000	-8.300000
8	280	281.707556	-1.707556
9	260	266.310320	-6.310320
10	244	246.902764	-2.902764
11	268	267.717914	0.282086
12	257	262.300000	-5.300000
13	245	246.323807	-1.323807
14	291	292.705602	-1.705602
15	292	292.705602	-0.705602

In [ ]: