Task 4

Kosovan Ivan

In [17]:

```
import pandas as pd
import numpy as np
from itertools import repeat
```

In [18]:

```
features = pd.read_csv("features.txt", sep=' ', names=['id', 'cols'])
subject_train = np.loadtxt("subject_train.txt")
subject_test = np.loadtxt("subject_test.txt")
x_train = np.loadtxt("X_train.txt")
y_train = np.loadtxt("Y_train.txt")
x_test = np.loadtxt("X_test.txt")
y_test = np.loadtxt("Y_test.txt")
```

1. Merges the training and the test sets to create one data set.

In [19]:

```
x_train = np.append(x_train, subject_train.reshape(-1,1), axis=1)
x_train = np.append(x_train, y_train.reshape(-1,1), axis=1)

x_test = np.append(x_test, subject_test.reshape(-1,1), axis=1)
x_test = np.append(x_test, y_test.reshape(-1,1), axis=1)

X = pd.DataFrame(np.append(x_train, x_test, axis=0), columns=features.cols.values.
X
```

Out[19]:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAc mad()-
0	0.288585	-0.020294	-0.132905	-0.995279	-0.983111	-0.913526	-0.995112	-0.98318
1	0.278419	-0.016411	-0.123520	-0.998245	-0.975300	-0.960322	-0.998807	-0.97491
2	0.279653	-0.019467	-0.113462	-0.995380	-0.967187	-0.978944	-0.996520	-0.96366
3	0.279174	-0.026201	-0.123283	-0.996091	-0.983403	-0.990675	-0.997099	-0.98275
4	0.276629	-0.016570	-0.115362	-0.998139	-0.980817	-0.990482	-0.998321	-0.97967
...	...	...	...	...	...	...	...	
10294	0.310155	-0.053391	-0.099109	-0.287866	-0.140589	-0.215088	-0.356083	-0.14877
10295	0.363385	-0.039214	-0.105915	-0.305388	0.028148	-0.196373	-0.373540	-0.03003
10296	0.349966	0.030077	-0.115788	-0.329638	-0.042143	-0.250181	-0.388017	-0.13325
10297	0.237594	0.018467	-0.096499	-0.323114	-0.229775	-0.207574	-0.392380	-0.27961
10298	0.153627	-0.018437	-0.137018	-0.330046	-0.195253	-0.164339	-0.430974	-0.21829

10299 rows × 563 columns

2. Extracts only the measurements on the mean and standard deviation for each measurement.

In [20]:

```
X_mean_std = X[ features[features.cols.str.contains('mean') | features.cols.str.co
X_mean_std.head()
```

Out[20]:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tGravityAcc-mean()-X	tGravityAcc mean()-
0	0.288585	-0.020294	-0.132905	-0.995279	-0.983111	-0.913526	0.963396	-0.14084
1	0.278419	-0.016411	-0.123520	-0.998245	-0.975300	-0.960322	0.966561	-0.14155
2	0.279653	-0.019467	-0.113462	-0.995380	-0.967187	-0.978944	0.966878	-0.14201
3	0.279174	-0.026201	-0.123283	-0.996091	-0.983403	-0.990675	0.967615	-0.14397
4	0.276629	-0.016570	-0.115362	-0.998139	-0.980817	-0.990482	0.968224	-0.14875

5 rows × 79 columns

3. Uses descriptive activity names to name the activities in the data set

In [21]:

```
activities = pd.read_csv("activity_labels.txt",sep=' ', names=['activity', 'activi
activities
```

Out[21]:

	activity	activity_type
0	1	WALKING
1	2	WALKING_UPSTAIRS
2	3	WALKING_DOWNSTAIRS
3	4	SITTING
4	5	STANDING
5	6	LAYING

4. Appropriately labels the data set with descriptive variable names.

In [22]:

```
X = X.drop(columns=['activity']).rename(columns={'activity_type':'activity'})
X
```

Out[22]:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc mad()-
0	0.288585	-0.020294	-0.132905	-0.995279	-0.983111	-0.913526	-0.995112	-0.98318
1	0.278419	-0.016411	-0.123520	-0.998245	-0.975300	-0.960322	-0.998807	-0.97491
2	0.279653	-0.019467	-0.113462	-0.995380	-0.967187	-0.978944	-0.996520	-0.96366
3	0.279174	-0.026201	-0.123283	-0.996091	-0.983403	-0.990675	-0.997099	-0.98275
4	0.276629	-0.016570	-0.115362	-0.998139	-0.980817	-0.990482	-0.998321	-0.97967
...	...	...	...	...	...	...	...	
10294	0.310155	-0.053391	-0.099109	-0.287866	-0.140589	-0.215088	-0.356083	-0.14877
10295	0.363385	-0.039214	-0.105915	-0.305388	0.028148	-0.196373	-0.373540	-0.03003
10296	0.349966	0.030077	-0.115788	-0.329638	-0.042143	-0.250181	-0.388017	-0.13325
10297	0.237594	0.018467	-0.096499	-0.323114	-0.229775	-0.207574	-0.392380	-0.27961
10298	0.153627	-0.018437	-0.137018	-0.330046	-0.195253	-0.164339	-0.430974	-0.21829

10299 rows × 562 columns