[CM1] Data Cleaning: [TODO: this should go first] deal with any missing values in the data (use any of the methods discussed in class: dropping data, interpolating, replacing with approximations,. . .). You can also remove any noise from the data by applying smoothing on some features. Report any changes you make and justify them. You can make comparisons of any of these approaches have an impact on classification performance using your validation set. Normalization: Normalize the data using min-max and zscore and compare to unnormalized version of the data. Explain any differences that you see. You may want to do some of the visualization in [CM2] to see the impact of Data Cleaning on the distribution of the data.

IRIS DATASET

Replacing the null values with Mean in iris dataset

```
In [4]: for col in ['sepal_length','sepal_width','petal_length','petal_width']:
            df_iris[col].fillna(df_iris[col].mean(),inplace=True)
        df_iris
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.045070	2.508203	3.018024	1.164924	Iris-versicolor
1	6.325517	2.115481	4.542052	1.413651	Iris-versicolor
2	5.257497	3.814303	1.470660	0.395348	Iris-setosa
3	6.675168	3.201700	5.785461	2.362764	Iris-virginica
4	5.595237	2.678166	4.077750	1.369266	Iris-versicolor
5	6.707485	3.093846	5.048317	2.373470	Iris-virginica
6	4.811740	3.037915	1.494268	-0.042428	Iris-setosa
7	5.205868	3.059083	1.675654	0.112269	Iris-setosa
8	4.436832	2.867772	1.428415	0.385249	Iris-setosa
9	6.847619	3.132270	5.878479	2.166297	Iris-virginica
10	6.478961	3.149863	5.078239	1.785111	Iris-virginica

```
In [5]: df_irisnew = df_iris.copy()
```

Normalization : MinMax and Zscore (iris)

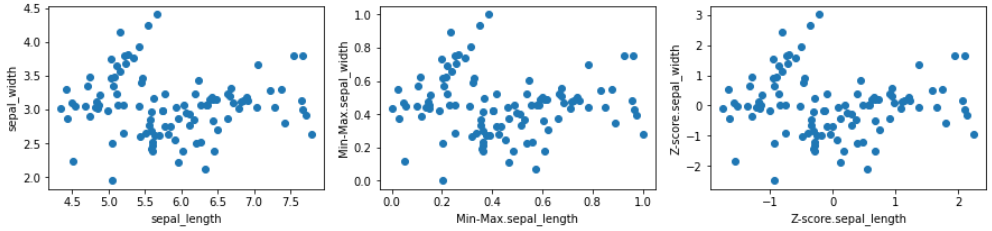
```
df_iris['Min-Max.sepal_length'] = (df_iris.sepal_length - df_iris.sepal_length.min())/(df_iris.sepal_length.max() - df_iris.sepal
df_iris['Min-Max.sepal_width'] = (df_iris.sepal_width - df_iris.sepal_width.min())/(df_iris.sepal_width.max() - df_iris.sepal_wi
df_iris['Min-Max.petal_length'] = (df_iris.petal_length - df_iris.petal_length.min())/(df_iris.petal_length.max() - df_iris.petal
df_iris['Min-Max.petal_width'] = (df_iris.petal_width - df_iris.petal_width.min())/(df_iris.petal_width.max() - df_iris.petal_wi
df_iris['Z-score.sepal_length'] = (df_iris.sepal_length - df_iris.sepal_length.mean())/df_iris.sepal_length.std()
df_iris['Z-score.sepal_width'] = (df_iris.sepal_width - df_iris.sepal_width.mean())/df_iris.sepal_width.std()
df_iris['Z-score.petal_length'] = (df_iris.petal_length - df_iris.petal_length.mean())/df_iris.petal_length.std()
df_iris['Z-score.petal_width'] = (df_iris.petal_width - df_iris.petal_width.mean())/df_iris.petal_width.std()
```

Min-Max.sepal_length	Min-Max.sepal_width	Min-Max.petal_length	Min-Max.petal_width	Z-score.sepal_length	Z-score.sepal_width	Z-score.petal_length	Z-score.petal_width
0.203115	0.228204	0.346084	0.462421	-0.944525	-1.234386e+00	-4.609905e-01	-0.044188
0.574092	0.068791	0.611799	0.555392	0.541536	-2.114380e+00	4.234637e-01	0.271779
0.264660	0.758373	0.076301	0.174764	-0.697987	1.692264e+00	-1.358987e+00	-1.021809
0.675395	0.509707	0.828588	0.910157	0.947335	3.195704e-01	1.145063e+00	1.477473
0.362512	0.297195	0.530848	0.538801	-0.306013	-8.535406e-01	1.540108e-01	0.215395
0.684758	0.465927	0.700066	0.914159	0.984841	7.789623e-02	7.172694e-01	1.491072
0.135514	0.443223	0.080417	0.011129	-1.215323	-4.743312e-02	-1.345287e+00	-1.577931

In [10]:
```
plt.figure(figsize=(15,3))
plt.subplot(1,3,1)
plt.scatter(df_iris['sepal_length'],df_iris['sepal_width'])
plt.xlabel('sepal_length')
plt.ylabel('sepal_width')

plt.subplot(1,3,2)
plt.scatter(df_iris['Min-Max.sepal_length'],df_iris['Min-Max.sepal_width'])
plt.xlabel('Min-Max.sepal_length')
plt.ylabel('Min-Max.sepal_width')

plt.subplot(1,3,3)
plt.scatter(df_iris['Z-score.sepal_length'],df_iris['Z-score.sepal_width'])
plt.xlabel('Z-score.sepal_length')
plt.ylabel('Z-score.sepal_width')
plt.show()
```



HEART DISEASE DATASET

Replacing the null values with Mean, Mode and Median in heart disease dataset

In [6]:
```
for col in ['restecg','slope']:
    df_hd[col].fillna(df_hd[col].mode()[0],inplace=True)
for col in ['trestbps','chol','thalach','thal']:
    df_hd[col].fillna(df_hd[col].mean(),inplace=True)
df_hd['oldpeak'] = df_hd['oldpeak'].fillna(df_hd['oldpeak'].median())
df_hd
```

Out[6]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	76	0	2	140.102822	197.105970	0	2.0	115.952071	0	1.284822	1.0	0	2.175904	1
1	43	0	0	132.079599	341.049462	1	0.0	135.970028	1	3.110483	1.0	0	3.082071	0
2	47	1	2	107.899290	242.822816	0	1.0	152.210039	0	-0.023723	2.0	0	2.020827	0
3	51	1	2	99.934001	244.133256	0	1.0	143.049207	1	1.195082	1.0	0	2.100312	1
4	57	1	0	110.103508	334.952353	0	1.0	143.099327	1	3.082052	1.0	1	2.831509	0
5	58	0	2	119.900334	339.874109	0	1.0	149.647978	0	-0.080278	2.0	0	2.035667	1
6	63	0	2	134.926653	252.023252	0	0.0	171.840974	0	0.106792	2.0	0	1.903701	1
7	58	1	1	119.924599	284.008194	0	0.0	159.850624	0	1.718823	1.0	0	2.060892	0
8	57	1	0	140.080577	192.215185	0	1.0	148.025188	0	0.436121	1.0	0	0.931512	1
9	62	1	0	119.963981	267.062360	0	1.0	98.844968	1	1.740426	1.0	2	3.042729	0
10	71	0	2	110.176447	264.876099	1	0.0	130.008925	0	0.163255	2.0	1	2.050483	1

In [7]:
```
df_hdnew = df_hd.copy()
```

In [11]:
```
df_hd['MinMax_age'] = (df_hd.age - df_hd.age.min())/(df_hd.age.max() - df_hd.age.min())
df_hd['MinMax_sex'] = (df_hd.sex - df_hd.sex.min())/(df_hd.sex.max() - df_hd.sex.min())
df_hd['MinMax_cp'] = (df_hd.cp - df_hd.cp.min())/(df_hd.cp.max() - df_hd.cp.min())
df_hd['MinMax_trestbps'] = (df_hd.trestbps - df_hd.trestbps.min())/(df_hd.trestbps.max() - df_hd.trestbps.min())
df_hd['MinMax_chol'] = (df_hd.chol - df_hd.chol.min())/(df_hd.chol.max() - df_hd.chol.min())
df_hd['MinMax_fbs'] = (df_hd.fbs - df_hd.fbs.min())/(df_hd.fbs.max() - df_hd.fbs.min())
df_hd['MinMax_restecg'] = (df_hd.restecg - df_hd.restecg.min())/(df_hd.restecg.max() - df_hd.restecg.min())
df_hd['MinMax_thalach'] = (df_hd.thalach - df_hd.thalach.min())/(df_hd.thalach.max() - df_hd.thalach.min())
df_hd['MinMax_exang'] = (df_hd.exang - df_hd.exang.min())/(df_hd.exang.max() - df_hd.exang.min())
df_hd['MinMax_oldpeak'] = (df_hd.oldpeak - df_hd.oldpeak.min())/(df_hd.oldpeak.max() - df_hd.oldpeak.min())
df_hd['MinMax_slope'] = (df_hd.slope - df_hd.slope.min())/(df_hd.slope.max() - df_hd.slope.min())
df_hd['MinMax_ca'] = (df_hd.ca - df_hd.ca.min())/(df_hd.ca.max() - df_hd.ca.min())
df_hd['MinMax_thal'] = (df_hd.thal - df_hd.thal.min())/(df_hd.thal.max() - df_hd.thal.min())
df_hd['MinMax_target'] = (df_hd.target - df_hd.target.min())/(df_hd.target.max() - df_hd.target.min())

df_hd['zscore_age'] = (df_hd.age - df_hd.age.mean())/df_hd.age.std()
df_hd['zscore_sex'] = (df_hd.sex - df_hd.sex.mean())/df_hd.sex.std()
df_hd['zscore_cp'] = (df_hd.cp - df_hd.cp.mean())/df_hd.cp.std()
df_hd['zscore_trestbps'] = (df_hd.trestbps - df_hd.trestbps.mean())/df_hd.trestbps.std()
df_hd['zscore_chol'] = (df_hd.chol - df_hd.chol.mean())/df_hd.chol.std()
df_hd['zscore_fbs'] = (df_hd.fbs - df_hd.fbs.mean())/df_hd.fbs.std()
df_hd['zscore_restecg'] = (df_hd.restecg - df_hd.restecg.mean())/df_hd.restecg.std()
df_hd['zscore_thalach'] = (df_hd.thalach - df_hd.thalach.mean())/df_hd.thalach.std()
df_hd['zscore_exang'] = (df_hd.exang - df_hd.exang.mean())/df_hd.exang.std()
df_hd['zscore_oldpeak'] = (df_hd.oldpeak - df_hd.oldpeak.mean())/df_hd.oldpeak.std()
df_hd['zscore_slope'] = (df_hd.slope - df_hd.slope.mean())/df_hd.slope.std()
df_hd['zscore_ca'] = (df_hd.ca - df_hd.ca.mean())/df_hd.ca.std()
df_hd['zscore_thal'] = (df_hd.thal - df_hd.thal.mean())/df_hd.thal.std()
df_hd['zscore_target'] = (df_hd.target - df_hd.target.mean())/df_hd.target.std()

pd.set_option("display.max_columns", None)
df_hd
```

MinMax_age	MinMax_sex	MinMax_cp	MinMax_trestbps	MinMax_chol	MinMax_fbs	MinMax_restecg	MinMax_thalach	MinMax_exang	MinMax_oldpeak	MinMax_slope
0.979167	0.0	0.666667	0.470641	0.252879	0.0	1.0	0.244681	0.0	0.231837	0.5
0.291667	0.0	0.000000	0.388835	0.765412	1.0	0.0	0.420115	1.0	0.519670	0.5
0.375000	1.0	0.666667	0.142289	0.415661	0.0	0.5	0.562440	0.0	0.025532	1.0
0.458333	1.0	0.666667	0.061073	0.420327	0.0	0.5	0.482156	1.0	0.217688	0.5
0.583333	1.0	0.000000	0.164763	0.743703	0.0	0.5	0.482595	1.0	0.515187	0.5
0.604167	0.0	0.666667	0.264653	0.761227	0.0	0.5	0.539986	0.0	0.016616	1.0
0.708333	0.0	0.666667	0.417864	0.448420	0.0	0.0	0.734482	0.0	0.046109	1.0
0.604167	1.0	0.333333	0.264901	0.562308	0.0	0.0	0.629400	0.0	0.300261	0.5
0.583333	1.0	0.000000	0.470415	0.235464	0.0	0.5	0.525764	0.0	0.098031	0.5
0.687500	1.0	0.000000	0.265302	0.501969	0.0	0.5	0.094758	1.0	0.303667	0.5
0.875000	0.0	0.666667	0.165507	0.494185	1.0	0.0	0.367873	0.0	0.055011	1.0

MinMax_ca	MinMax_thal	MinMax_target	zscore_age	zscore_sex	zscore_cp	zscore_trestbps	zscore_chol	zscore_fbs	zscore_restecg	zscore_thalach	zscore_exang
0.00	0.544604	1.0	2.371556	-1.483808	1.019477	0.468495	-1.037435e+00	-0.389174	2.681603	-1.541022	-0.722982
0.00	0.919222	0.0	-1.236840	-1.483808	-0.936442	0.016614	2.137999e+00	2.557427	-1.070871	-0.625536	1.376636
0.00	0.480494	0.0	-0.799459	0.670763	1.019477	-1.345260	-2.890868e-02	-0.389174	0.805366	0.117171	-0.722982
0.00	0.513354	1.0	-0.362077	0.670763	1.019477	-1.793878	6.269911e-16	-0.389174	0.805366	-0.301783	1.376636
0.25	0.815637	0.0	0.293995	0.670763	-0.936442	-1.221115	2.003495e+00	-0.389174	0.805366	-0.299491	1.376636
0.00	0.486629	1.0	0.403340	-1.483808	1.019477	-0.669342	2.112071e+00	-0.389174	0.805366	0.000000	-0.722982
0.00	0.432073	1.0	0.950066	-1.483808	1.019477	0.176965	1.740556e-01	-0.389174	-1.070871	1.014957	-0.722982
0.00	0.497057	0.0	0.403340	0.670763	0.041517	-0.667975	8.796526e-01	-0.389174	-1.070871	0.466600	-0.722982
0.00	0.030162	1.0	0.293995	0.670763	-0.936442	0.467242	-1.145328e+00	-0.389174	0.805366	-0.074215	-0.722982
0.50	0.902957	0.0	0.840721	0.670763	-0.936442	-0.665757	5.058226e-01	-0.389174	0.805366	-2.323384	1.376636
0.25	0.492754	1.0	1.824829	-1.483808	1.019477	-1.217007	4.575931e-01	2.557427	-1.070871	-0.898157	-0.722982

zscore_oldpeak	zscore_slope	zscore_ca	zscore_thal	zscore_target
0.158324	-0.688814	-0.703850	-0.288348	0.916242
1.651142	-0.688814	-0.703850	1.220199	-1.086266
-0.911656	0.915896	-0.703850	-0.546514	-1.086266
0.084945	-0.688814	-0.703850	-0.414190	0.916242
1.627895	-0.688814	0.258835	0.803073	-1.086266
-0.957900	0.915896	-0.703850	-0.521808	0.916242
-0.804936	0.915896	-0.703850	-0.741499	0.916242
0.513201	-0.688814	-0.703850	-0.479816	-1.086266
-0.535648	-0.688814	-0.703850	-2.359956	0.916242
0.530866	-0.688814	1.221520	1.154704	-1.086266
-0.758766	0.915896	0.258835	-0.497144	0.916242

MinMax Normalization:

- General formula is used to calculate the MinMax Normalization i.e. (Value-Min/Max-Min)
- It bounds the data between 0 & 1 but the squashing which occurs in the un-normalized data it fixes the squashing problem.
- It does not deals with the outliers.

Z-Score Normalization:

- General formula is used to calculate the Z-Score Normalization i.e. (Value-Mean/Std)
- It takes the points roughly in the same scale on both features and handles the squashing.
- Handles the outliers more effectively as compared to other normalization methods.