Feature Selection and Dimensionality Reduction of Country Data

Main Objectives

The main objective of this report is to reduce the number of features in this dataset to the most important as well as the one which accounts for the dataset itself.

PCA - Linear Method that finds the principle copnents of the data

Kernal PCA - Maps Data into a higher dimensionalit spaces allowing for capture of non linear relationships. Transformes the data into higher dimensionality to help identify and handle non linearly sperable data.

MDS - Aims to reduce dimensionality while highlighting and preserving the distances bewteeen points as it lower dimensionality.

Data Set

- country: Name of the country
- child_mort: Death of children under 5 years of age per 1000 live births
- exports: Exports of goods and services per capita. Given as %age of the GDP per capita
- health: Total health spending per capita. Given as %age of GDP per capita
- imports: Imports of goods and services per capita. Given as %age of the GDP per capita
- Income: Net income per person
- Inflation: The measurement of the annual growth rate of the Total GDP
- life_expec: The average number of years a new born child would live if the current mortality patterns are to remain the same
- total_fer: The number of children that would be born to each woman if the current age-fertility rates remain the same.
- gdpp: The GDP per capita. Calculated as the Total GDP divided by the total population.

This Dataset covers 167 Countrys and provides the information above. As well as this I've taken the time to update the dataset speicfically to change the Percentages for Exports, Health, and Imports to actual values using the GDPP

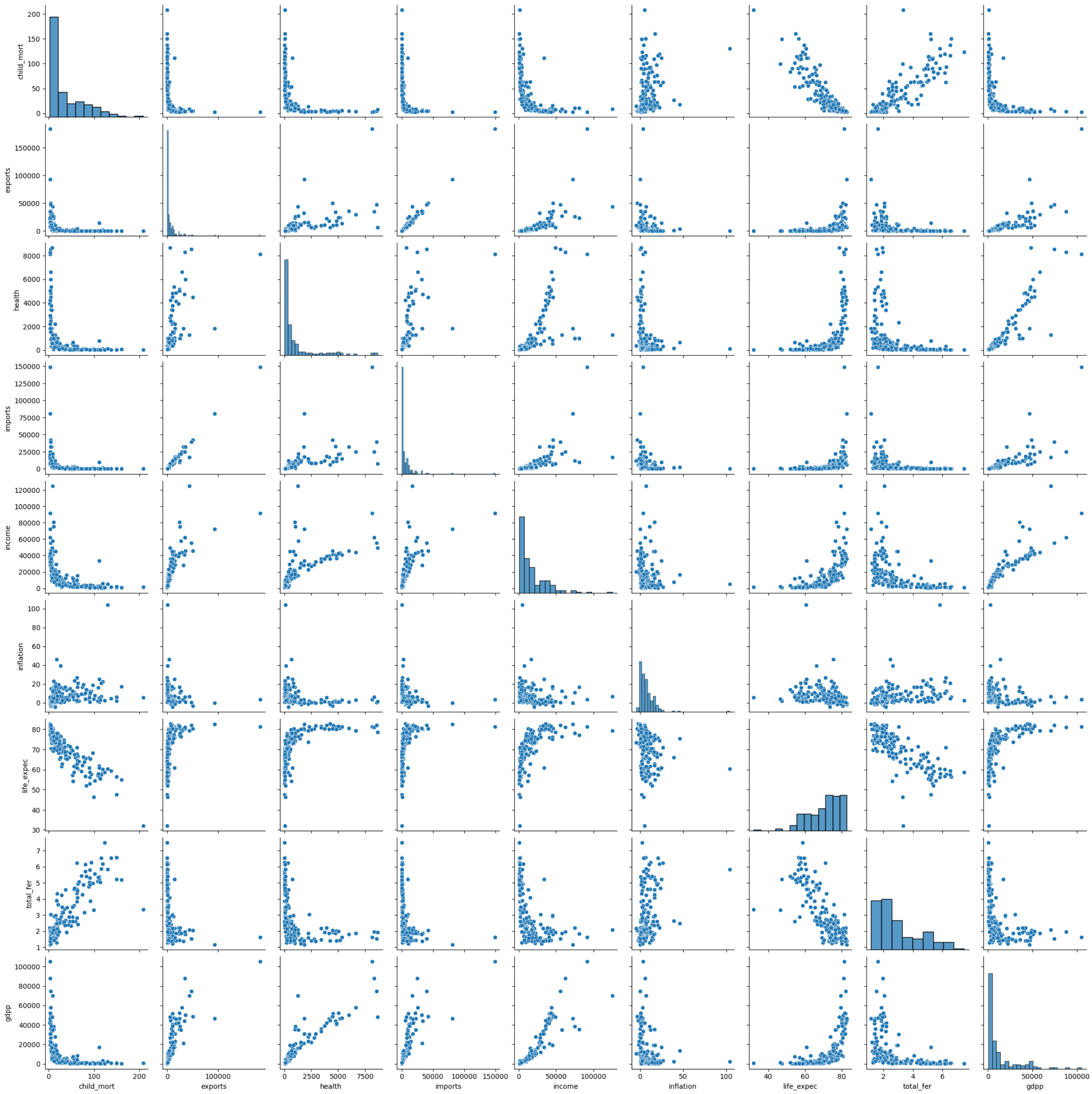
using this formula

```
Actual Value = Percentage Value * GDPP / 100
```

As a result this is the output from the dataset itself (summarized)

	count	mean	std	min	25%	50%	75%	max
child_mort	167.0	38.270060	40.328931	2.60	8.250	19.30	62.10	208.00
exports	167.0	7420.618862	17973.885789	1.08	447.140	1777.44	7278.00	183750.00
health	167.0	1056.733174	1801.408921	12.82	78.535	321.89	976.94	8663.60
imports	167.0	6588.352096	14710.810423	0.65	640.215	2045.58	7719.60	149100.00
income	167.0	17144.688623	19278.067698	609.00	3355.000	9960.00	22800.00	125000.00
inflation	167.0	7.781737	10.570770	-4.21	1.810	5.39	10.75	104.00
life_expec	167.0	70.555689	8.893172	32.10	65.300	73.10	76.80	82.80
total_fer	167.0	2.947964	1.513848	1.15	1.795	2.41	3.88	7.49
gdpp	167.0	12964.155689	18328.704809	231.00	1330.000	4660.00	14050.00	105000.00

This is also the relationships between each coloumn itself.



	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
child_mort	0.000000	-0.297230	-0.430438	-0.319138	-0.524315	0.288275	-0.886676	0.848478	-0.483032
exports	-0.297230	0.000000	0.612919	0.987686	0.725351	-0.141559	0.377694	-0.291096	0.768894
health	-0.430438	0.612919	0.000000	0.638581	0.690857	-0.253951	0.545626	-0.407984	0.916593
imports	-0.319138	0.987686	0.638581	0.000000	0.672056	-0.179466	0.397515	-0.317061	0.755114
income	-0.524315	0.725351	0.690857	0.672056	0.000000	-0.147759	0.611962	-0.501840	0.895571
inflation	0.288275	-0.141559	-0.253951	-0.179466	-0.147759	0.000000	-0.239707	0.316921	-0.221629
life_expec	-0.886676	0.377694	0.545626	0.397515	0.611962	-0.239707	0.000000	-0.760875	0.600089
total_fer	0.848478	-0.291096	-0.407984	-0.317061	-0.501840	0.316921	-0.760875	0.000000	-0.454910
gdpp	-0.483032	0.768894	0.916593	0.755114	0.895571	-0.221629	0.600089	-0.454910	0.000000

The Data itself is not missing any values, but does have values in the negatives. Each column having a high overall absolute value correlation in total. Not only this but each has a high Skew as well (This isn't suprising based upon how different countries are at either different stages in both Health and Economic Boom)

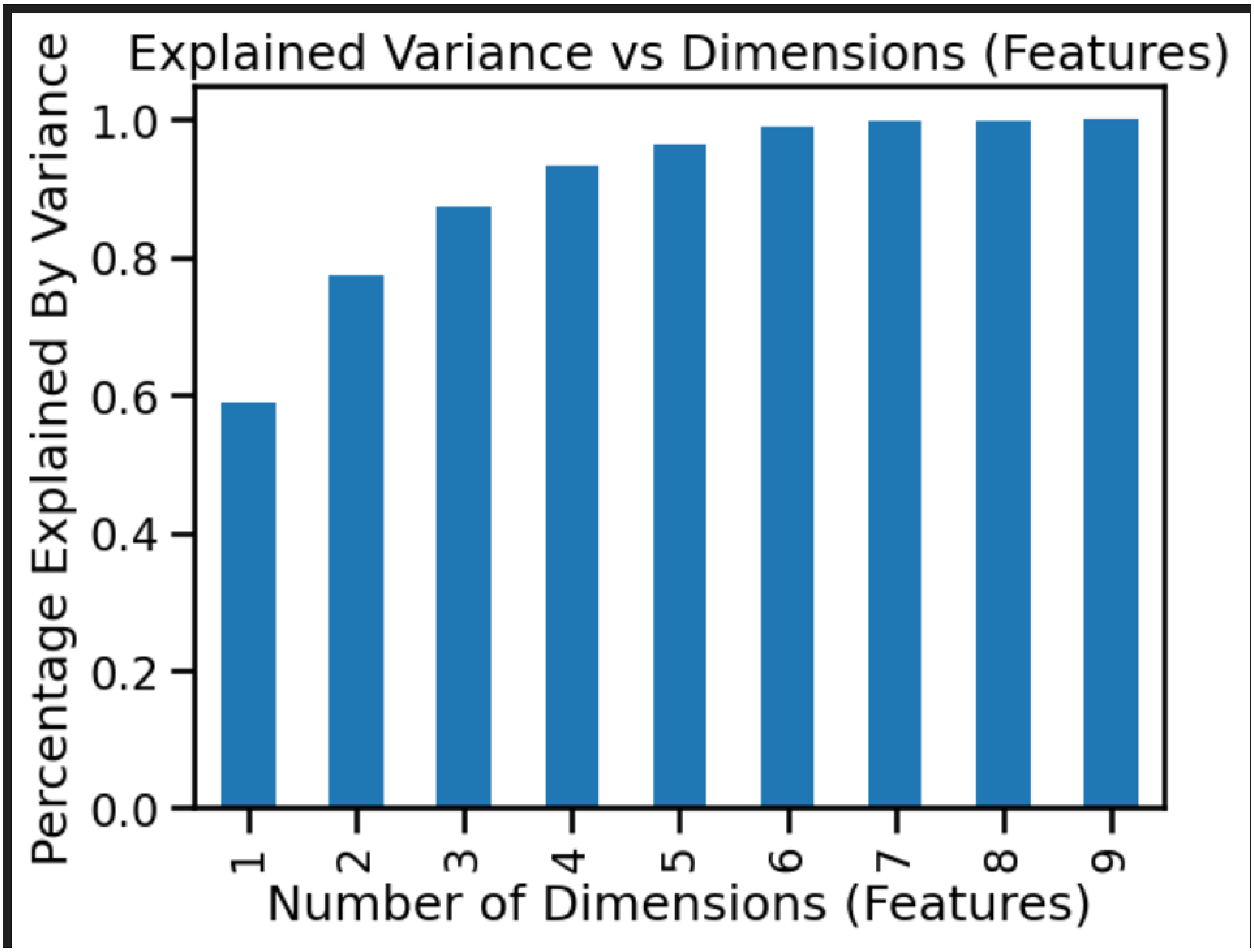
Decided Against correcting for Skew because it minimilizes the impact of the dataset as well as causes issues later on with feature scaling. In terms of choices I will be scaling the data since this provides both a better outcome for feature scaling as well as if I wanted to use it for KMeans as well as Agglomorative Custering (ward)

Findings

PCA

When running through 1 - 9 different types of Dinensions, these are the results.

feature	child_mort	exports	gdpp	health	imports	income	inflation	life_expec	total_fer
n									
1	0.107892	0.116927	0.136399	0.122263	0.117601	0.129597	0.048793	0.117257	0.103271
2	0.122412	0.122881	0.121710	0.107057	0.121218	0.110518	0.055843	0.120874	0.117486
3	0.119985	0.115510	0.113589	0.102652	0.115675	0.108752	0.090923	0.120477	0.112437
4	0.116610	0.119549	0.115728	0.110998	0.119997	0.106182	0.085882	0.113994	0.111060
5	0.114987	0.117121	0.113958	0.114032	0.120036	0.112811	0.085395	0.113035	0.108625
6	0.114721	0.115363	0.112317	0.113637	0.117979	0.111124	0.084851	0.116395	0.113612
7	0.116658	0.114680	0.111759	0.113143	0.117327	0.110500	0.084391	0.117692	0.113849
8	0.116536	0.114589	0.112102	0.113271	0.117276	0.110610	0.084304	0.117567	0.113746
9	0.116499	0.114683	0.112088	0.113257	0.117360	0.110588	0.084278	0.117534	0.113713



*According to the graph, the amount of features, that account for the most significant amount of the dataset is 5.

	model	var
n		
1	PCA(n_components=1)	0.589373
2	PCA(n_components=2)	0.773825
3	PCA(n_components=3)	0.872939
4	PCA(n_components=4)	0.933662
5	PCA(n_components=5)	0.963954
6	PCA(n_components=6)	0.988552
7	PCA(n_components=7)	0.99795
8	PCA(n_components=8)	0.999506
9	PCA(n_components=9)	1.0

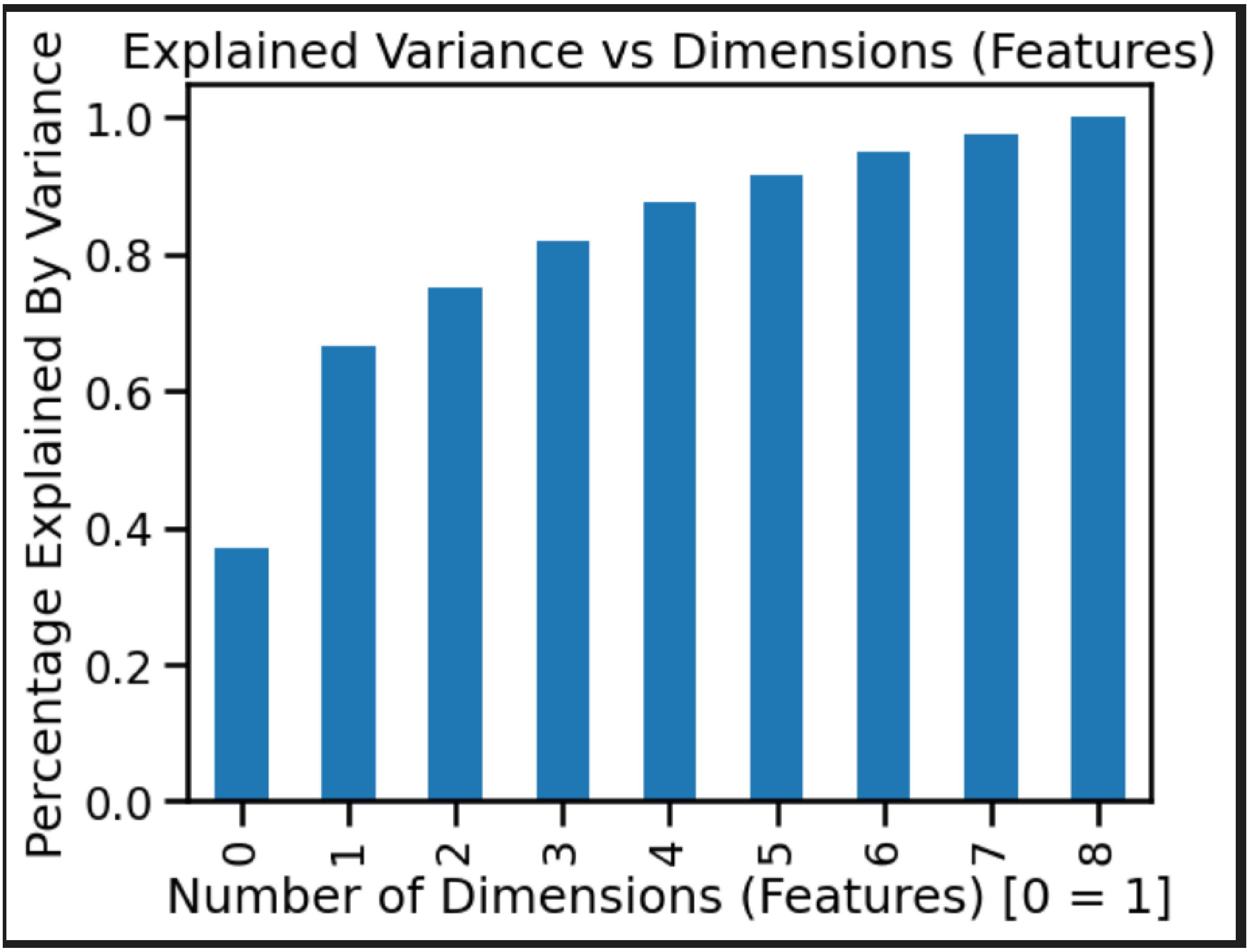
Kernel PCA

When Running and optimizing the Kernel PCA, with

```
param_grid =
    'gamma': [0.001,0.01,0.1,0.5,1.0],
    'n_components': [2,3,4, 5,6,7,8,9]
```

The best result was gamma of 0.5 and a number of components of 9. After re running and looking for the Explained variance and ratios, the results displayed that similar to PCA, around 5 was the number of features that would explain the dataset.

	Explained_Variance	Explained_Variance_Ratio	Sum_Explained_Variance_Ratio
0	0.172169	0.371690	0.371690
1	0.137001	0.295768	0.667458
2	0.038518	0.083155	0.750613
3	0.032359	0.069858	0.820471
4	0.025546	0.055151	0.875623
5	0.018810	0.040609	0.916232
6	0.015595	0.033668	0.949900
7	0.011849	0.025580	0.975480
8	0.011358	0.024520	1.000000



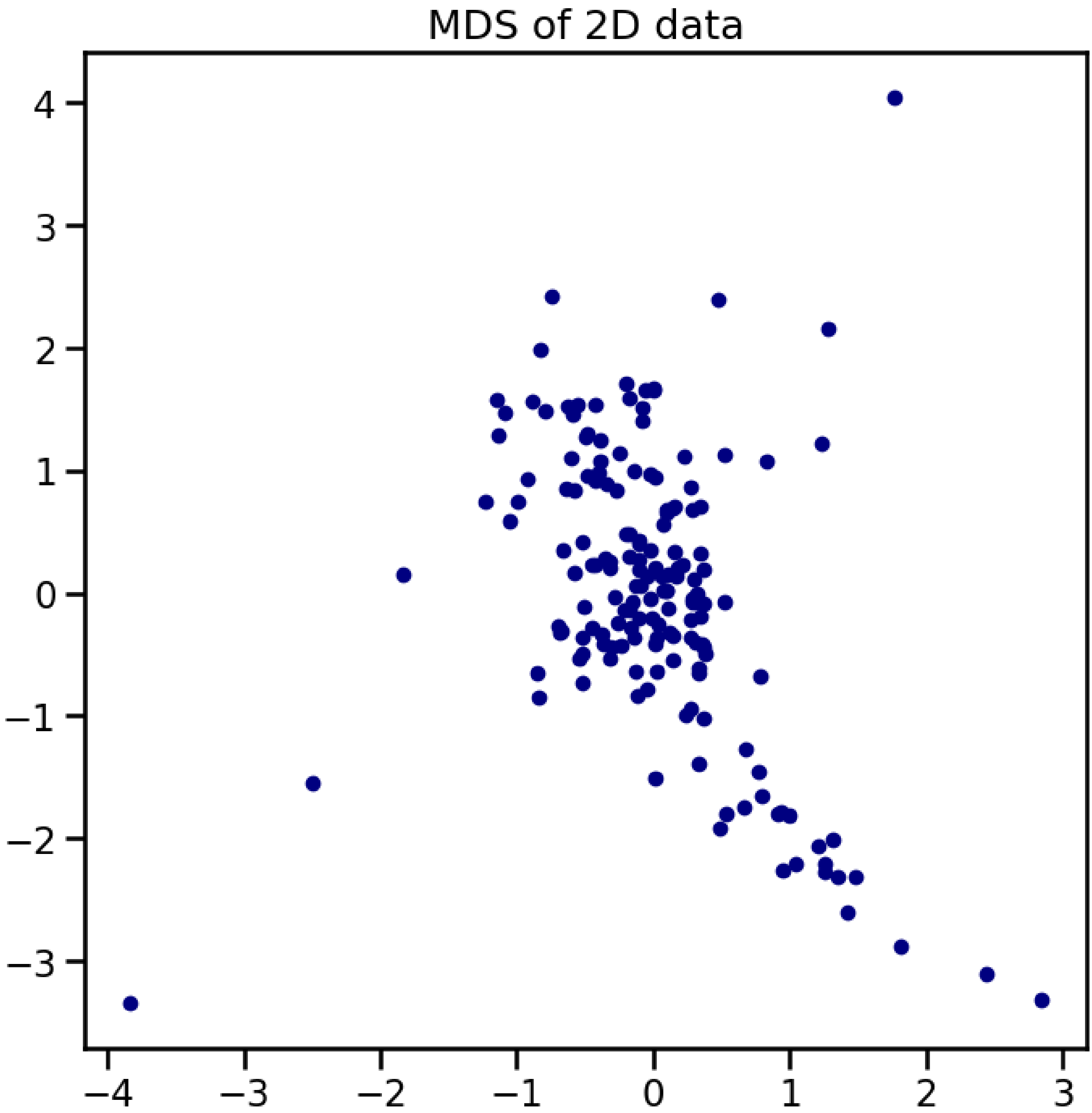
MDS Multi Dimensional Scaling

To Reduce the Number of Dimensions down.

In Comparision to PCA and Kernel the Number of Compnoents that was used was 9 then chosen the final, and then from that you would utilize the best captured columns based upon variance. Unfortuently MDS doesn't work the same way. It's goal is to protect the distances between values and features. So We'll be using Stress as a good indicator of results.

Below shows a slight divergance from PCA and Kernal, showing that around 7 components produce the best resultss, and additional values only resudce the stress minimally.

	Compnents	Stress
0	2.0	2633.920157
1	3.0	642.140828
2	4.0	196.328534
3	5.0	80.436881
4	6.0	50.252286
5	7.0	30.654464
6	8.0	31.818366
7	9.0	28.096330



Results / Conclusion

PCA and Kernal PCA both produced an outcome where when feature selecting I found that 5 features post PCA (Kernal and Other) produced enough of data sets variance and Feature weights. While when using MDS, it showed that 7 features (reduction) was the most optimal, in regards to reducing the features. I believe that in terms of Feature Reduction I would chose PCA (non kernal). MDS and Kernel both produced useful results, but PCA was able to account for more of the data with fewer actual columns itself. Such as at 4 Features, PCA produced 93 variance, while for Kernal it was 0.82. And for MDS there was a 196 stress, which is way to high for me to accept as a valaid reduction.