The Curse of Dimensionality

The "curse of dimensionality" is a fundamental challenge in machine learning that arises when dealing with data in high-dimensional spaces. Essentially, it describes the problems that occur when the number of features (dimensions) in a dataset increases significantly. Here's a breakdown of the key aspects:

What it means:

- **Increased sparsity:**
  - As the number of dimensions grows, the available data points become increasingly spread out or "sparse" in the high-dimensional space. This makes it harder to find meaningful patterns and relationships.
- **Increased complexity:**
  - With more dimensions, the complexity of machine learning models tends to increase. This can lead to overfitting, where a model fits the training data too closely and performs poorly on unseen data.
- **Increased computational cost:**
  - High-dimensional data requires more computational resources and time for processing and analysis.
- **Distance distortions:**
  - In high-dimensional spaces, the concept of "distance" between data points becomes less meaningful. Distances tend to become more uniform, making it difficult to distinguish between nearest and farthest neighbors.

Consequences:

- **Overfitting:** Models become overly tailored to the training data and fail to generalize well.
- **Increased computational burden:** Training and using models become much slower and more resource-intensive.
- **Reduced model performance:** The accuracy and effectiveness of machine learning algorithms can decrease.
- **Data sparsity:** It becomes difficult to find data points that are close to each other, which is crucial for many machine learning algorithms.

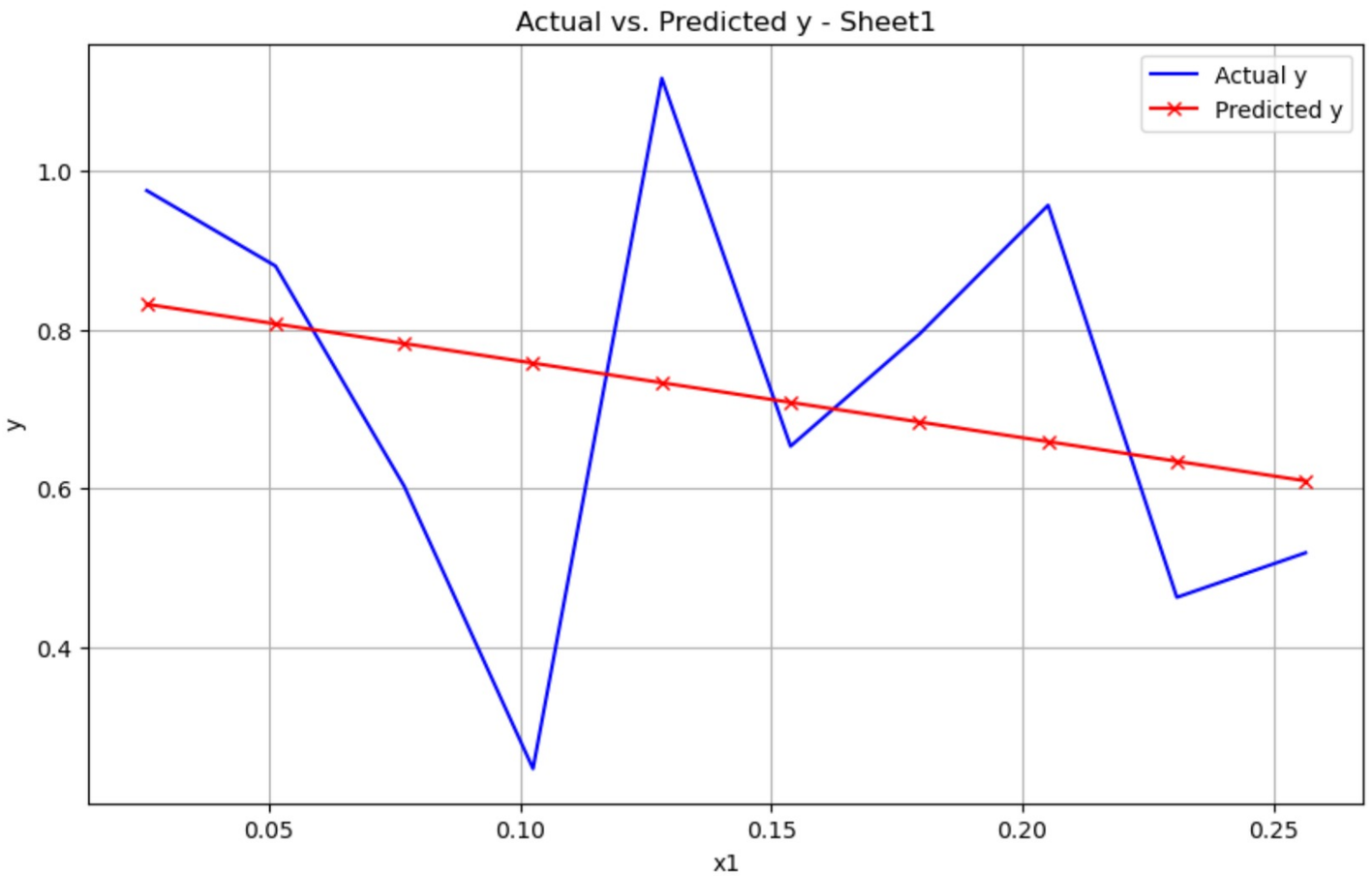
How to address the curse:

To mitigate the curse of dimensionality, several techniques are used:

- **Dimensionality reduction:**
  - Techniques like Principal Component Analysis (PCA) and t-SNE reduce the number of dimensions while preserving the most important information.
- **Feature selection:**
  - Identifying and selecting the most relevant features and discarding irrelevant or redundant ones.
- **Regularization:**
  - Techniques that penalize complex models and prevent overfitting.
- **Increase the amount of data:**
  - In some cases, having more data can help to mitigate the affects of high dimensionality.

Prediction model (Regression) created using a data set with one predictor variable, supported by only a few observations. The model seems to be fine

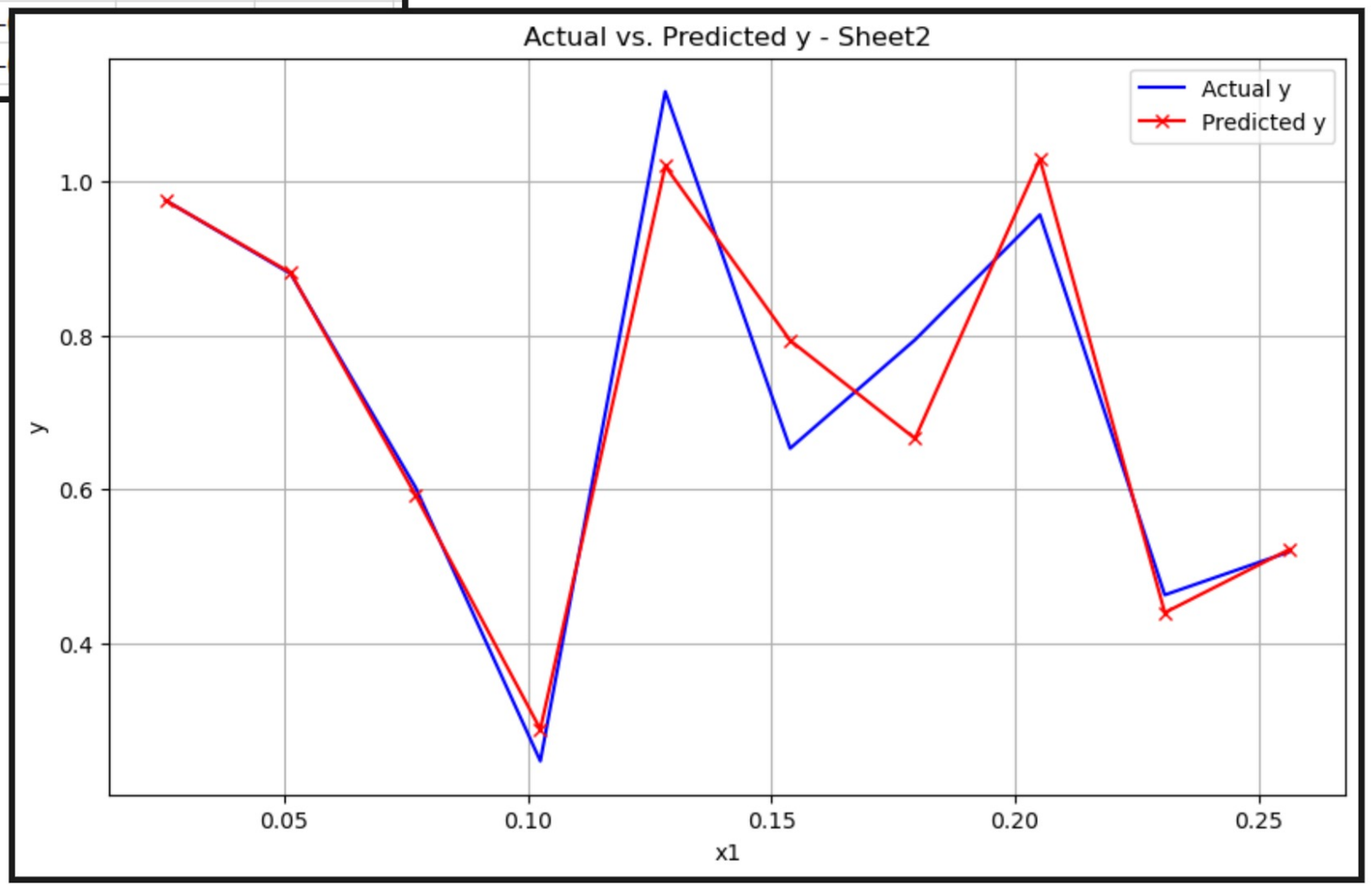
y	x1
0.974916288	0.025641
0.879902402	0.051282
0.60259774	0.076923
0.247087822	0.102564
1.116373478	0.128205
0.65306042	0.153846
0.794016193	0.179487
0.956404989	0.205128
0.462771865	0.230769
0.518899234	0.25641



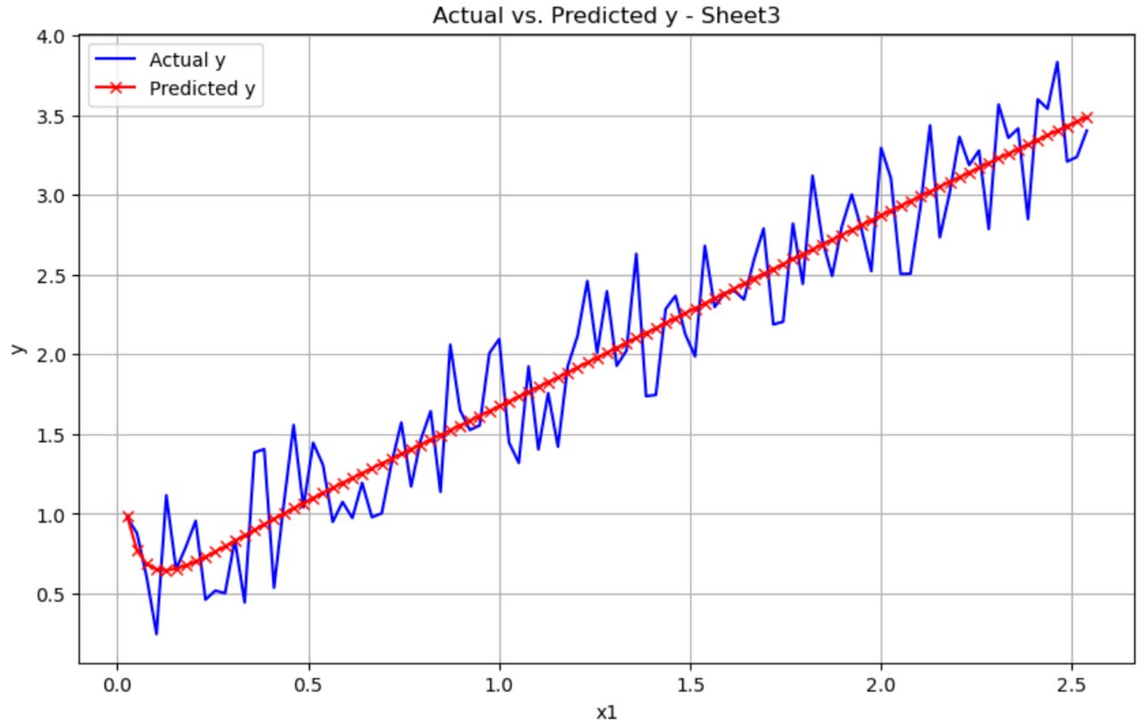
y	x1	x2	x3	x4	x5	x6	x7	x8
0.974916288	0.025641	0.000657	1.6858E-05	0.025638	0.025635	-1.59106	0.160128	0.29488
0.879902402	0.051282	0.00263	0.00013486	0.05126	0.051237	-1.29003	0.226455	0.371525
0.60259774	0.076923	0.005917	0.00045517	0.076847	0.076772	-1.11394	0.27735	0.42529
0.247087822	0.102564	0.010519	0.00107891	0.102384	0.102206	-0.989	0.320256	0.468093
1.116373478	0.128205	0.016437	0.00210725	0.127854	0.127507	-0.89209	0.358057	0.504237
0.65306042	0.153846	0.023669	0.00364133	0.15324	0.152644	-0.81291	0.392232	0.535832
0.794016193	0.179487	0.032216	0.0057823	0.178525	0.177584	-0.74597	0.423659	0.564085
0.956404989	0.205128	0.042078	0.0086313	0.203693	0.202299	-0.68797	0.452911	0.58976
0.462771865	0.230769	0.053254	0.01228949	0.228726	0.226758	-		
0.518899234	0.25641	0.065746	0.016858	0.25361	0.250935	-		

In this case, more predictor variables are existing in the data set, but the number of observations have not increased - leading to too many 'features' chasing too few observations.

This results in overfit models - as evident from the plot alongside



y	x1	x2	x3	x4	x5	x6	x7	x8
0.97491629	0.02564	0.00066	1.686E-05	0.02564	0.02564	-1.5911	0.16013	0.00855
0.8799024	0.05128	0.00263	0.0001349	0.05126	0.05124	-1.29	0.22646	0.01709
0.60259774	0.07692	0.00592	0.0004552	0.07685	0.07677	-1.1139	0.27735	0.02564
0.24708782	0.10256	0.01052	0.0010789	0.10238	0.10221	-0.989	0.32026	0.03419
1.11637348	0.12821	0.01644	0.0021073	0.12785	0.12751	-0.8921	0.35806	0.04274
0.65306042	0.15385	0.02367	0.0036413	0.15324	0.15264	-0.8129	0.39223	0.05128
0.79401619	0.17949	0.03222	0.0057823	0.17853	0.17758	-0.746	0.42366	0.05983
0.95640499	0.20513	0.04208	0.0086313	0.20369	0.2023	-0.688	0.45291	0.06838
0.46277187	0.23077	0.05325	0.0122895	0.22873	0.22676	-0.6368	0.48038	0.07692
0.51889923	0.25641	0.06575	0.016858	0.25361	0.25093	-0.5911	0.50637	0.08547
0.50315032	0.28205	0.07955	0.022438	0.27833	0.2748	-0.5497	0.53109	0.09402
0.83801445	0.30769	0.09467	0.0291306	0.30286	0.29834	-0.5119	0.5547	0.10256
0.44545653	0.33333	0.11111	0.037037	0.32719	0.32151	-0.4771	0.57735	0.11111
1.38512875	0.35897	0.12886	0.0462584	0.35131	0.34431	-0.4449	0.59914	0.11966
1.40570838	0.38462	0.14793	0.0568958	0.3752	0.36671	-0.415	0.62017	0.12821
0.53738393	0.41026	0.16831	0.0690504	0.39884	0.38869	-0.3869	0.64051	0.13675
1.06174709	0.4359	0.19001	0.0828234	0.42222	0.41024	-0.3606	0.66023	0.1453
1.5566868	0.46154	0.21302	0.0983159	0.44533	0.43134	-0.3358	0.67937	0.15385
1.04135657	0.48718	0.23734	0.1156291	0.46814	0.45197	-0.3123	0.69798	0.16239
1.44560911	0.51282	0.26298	0.134864	0.49064	0.47214	-0.29	0.71611	0.17094
1.30681257	0.53846	0.28994	0.156122	0.51282	0.49182	-0.2688	0.7338	0.17949
0.95000481	0.5641	0.31821	0.179504	0.53466	0.51102	-0.2486	0.75107	0.18803
1.07455259	0.58974	0.3478	0.2051113	0.55615	0.52971	-0.2293	0.76795	0.19658
0.97434017	0.61538	0.3787	0.2330451	0.57727	0.54791	-0.2109	0.78446	0.20513
1.19370358	0.64103	0.41091	0.2634063	0.59802	0.5656	-0.1931	0.80064	0.21368
0.97941216	0.66667	0.44444	0.2962963	0.61837	0.58278	-0.1761	0.8165	0.22222
1.00370854	0.69231	0.47929	0.3318161	0.63832	0.59946	-0.1597	0.83205	0.23077
1.31604894	0.71795	0.51545	0.3700669	0.65784	0.61564	-0.1439	0.84732	0.23932
1.5724815	0.74359	0.55293	0.4111499	0.67693	0.63131	-0.1287	0.86232	0.24786
1.17174728	0.76923	0.59172	0.4551661	0.69558	0.64648	-0.1139	0.87706	0.25641
1.46714903	0.79487	0.63182	0.5022168	0.71377	0.66116	-0.0997	0.89156	0.26496
1.64377495	0.82051	0.67324	0.5524031	0.7315	0.67535	-0.0859	0.90582	0.2735
1.13845302	0.84615	0.71598	0.6058261	0.74874	0.68905	-0.0726	0.91987	0.28205
2.05991159	0.87179	0.76003	0.662587	0.76549	0.70228	-0.0596	0.9337	0.2906
1.65027631	0.89744	0.80539	0.722787	0.78173	0.71505	-0.047	0.94733	0.29915
1.52620682	0.92308	0.85207	0.7865271	0.79746	0.72735	-0.0348	0.96077	0.30769
1.55432429	0.94872	0.90007	0.8539085	0.81267	0.7392	-0.0229	0.97402	0.31624
2.00784693	0.97436	0.94938	0.9250325	0.82734	0.75061	-0.0113	0.9871	0.32479
2.09549805	1	1	1	0.84147	0.76159	0	1	0.33333
1.44728778	1.02564	1.05194	1.0789123	0.85505	0.77215	0.011	1.01274	0.34188
1.31984868	1.05128	1.10519	1.1618706	0.86806	0.7823	0.02172	1.02532	0.35043
1.92415384	1.07692	1.15976	1.2489759	0.8805	0.79206	0.03218	1.03775	0.35897
1.40346032	1.10256	1.21565	1.3403294	0.89237	0.80142	0.0424	1.05003	0.36752
1.75623679	1.12821	1.27285	1.4360323	0.90364	0.8104	0.05239	1.06217	0.37607
1.42134939	1.15385	1.33136	1.5361857	0.91433	0.81902	0.06215	1.07417	0.38462
1.92544735	1.17949	1.39119	1.6408908	0.92441	0.82729	0.07169	1.08604	0.39316
2.1162841	1.20513	1.45233	1.7502487	0.93389	0.83521	0.08103	1.09778	0.40171



In this case, the larger set of features are supported by a larger number of observations. The resulting model has not overfit the training data

The Curse of Dimensionality: These three Figures illustrate how 'higher dimensionality', by way of more features, can be a 'curse' - unless supported by an equally large number of observations.

EDA / Data Preparation : Features (ie. Columns)

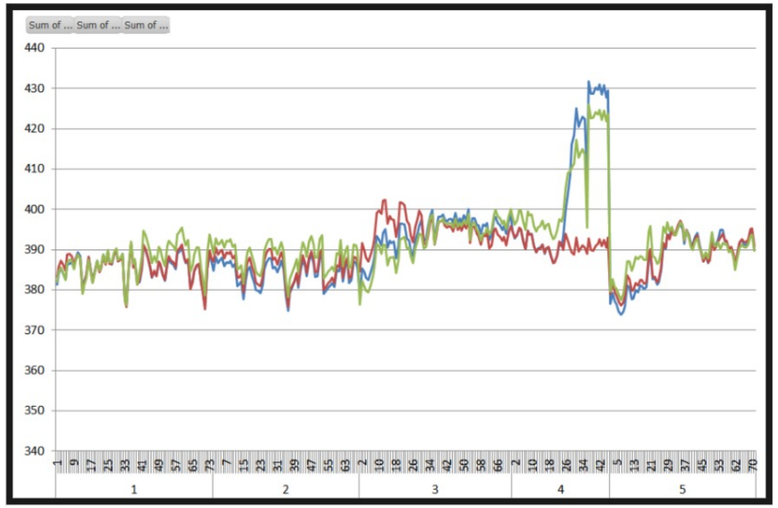
- Number of features and their impact on models: Reduce the number of features to improve model quality
- Identification / elimination of interdependent features
    - Feature Correlation : Heat Maps
    - Multicollinearity detection : VIF - Variance Inflation
    - PCA : Principal Component Analysis
- Multidimensional visualization:
    - PCA
    - t-SNE : t-distributed Stochastic Neighbour Embedding

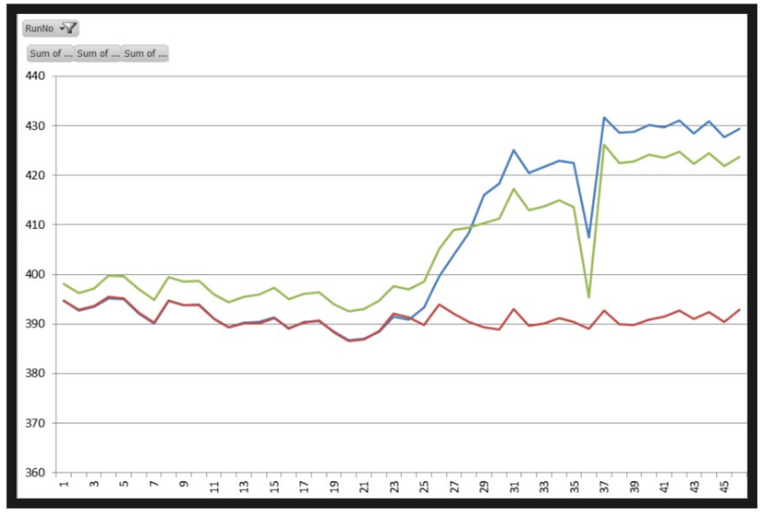
DIMENSIONALITY REDUCTION.

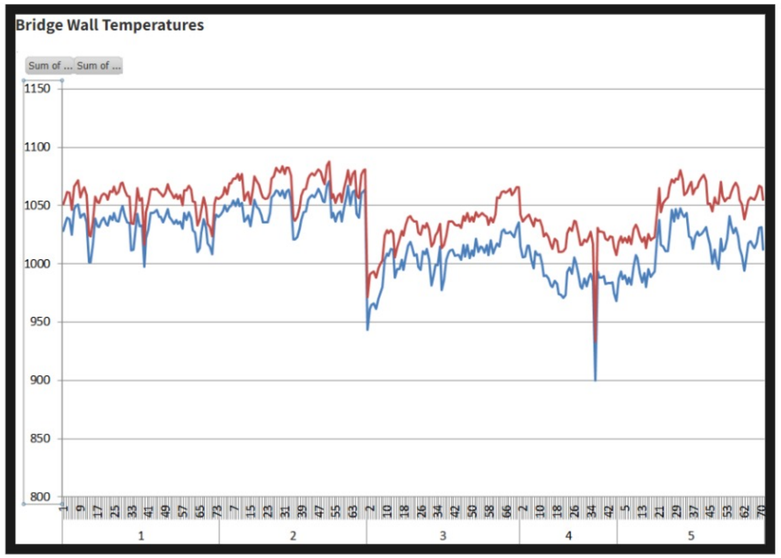
⇒ WITHOUT LOSING MUCH INFORMATION.

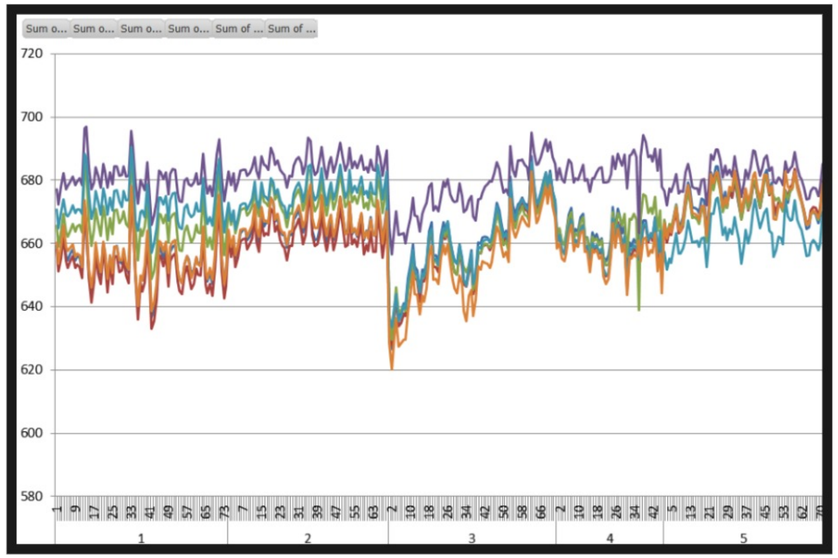
Pair-wise Correlation of Features

"230 columns"







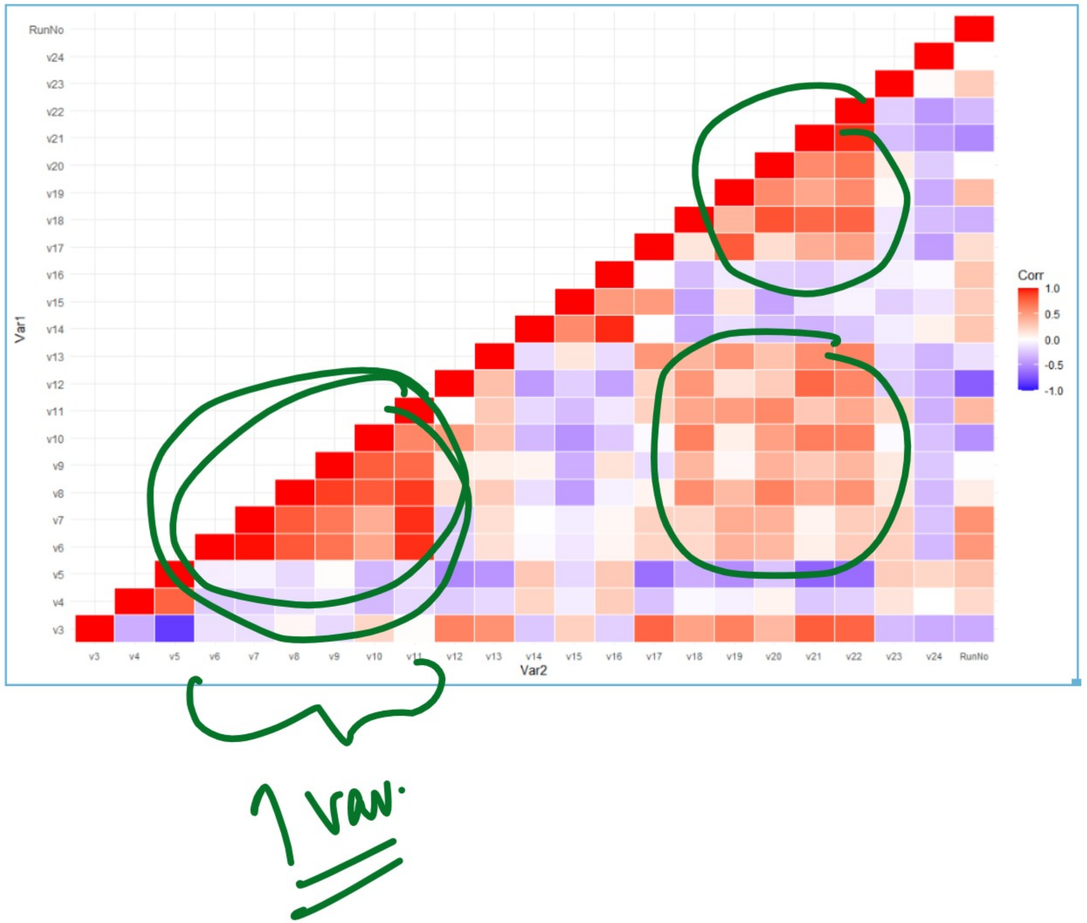


This method of pair-wise or group-wise plotting of features are convenient to detect feature dependencies if there are only a handful of features.

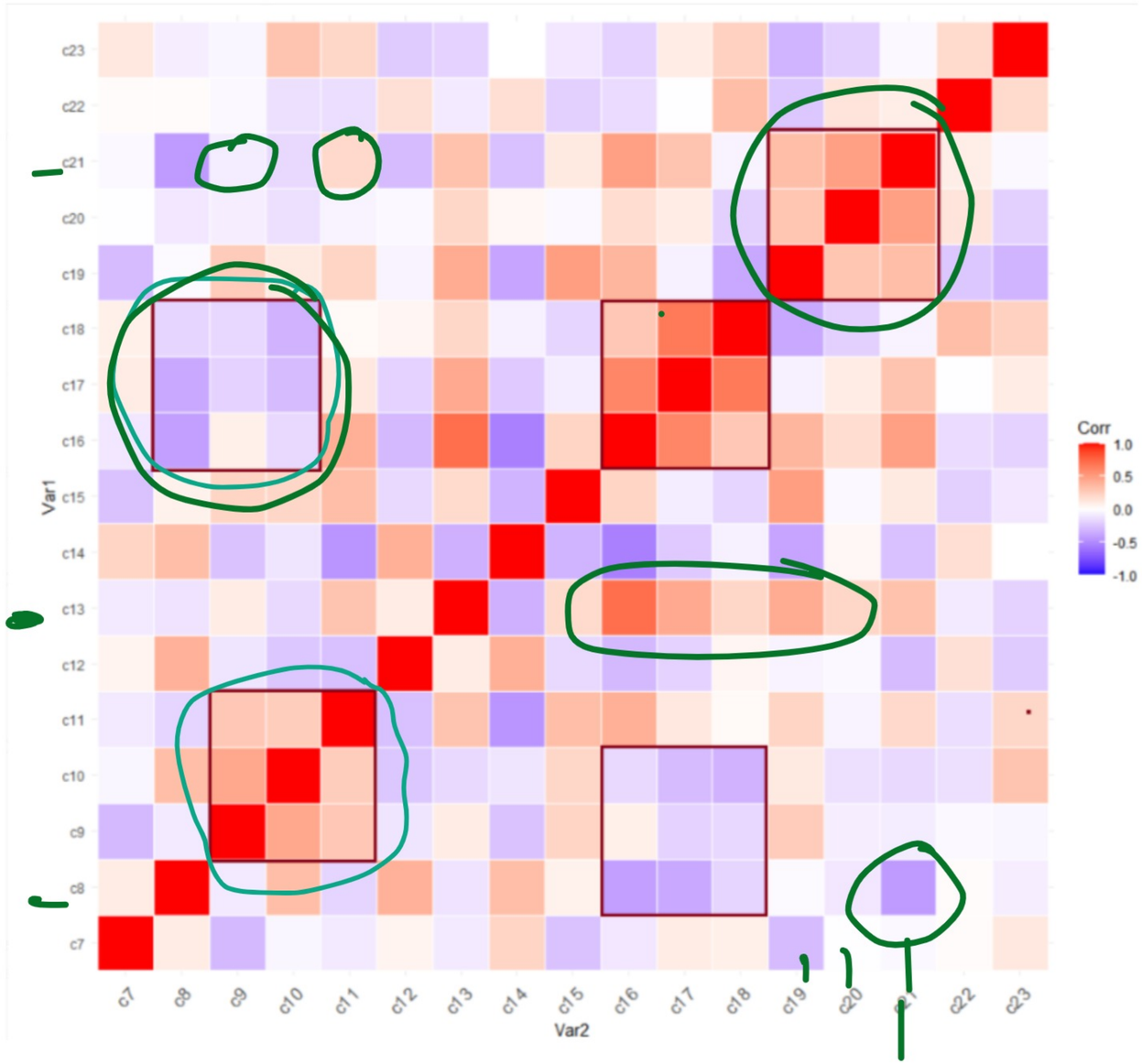
For a very large feature set, correlation **heat maps** are used. They can be automatically, and exhaustively created, and they provide a consolidated view of **feature correlations**.

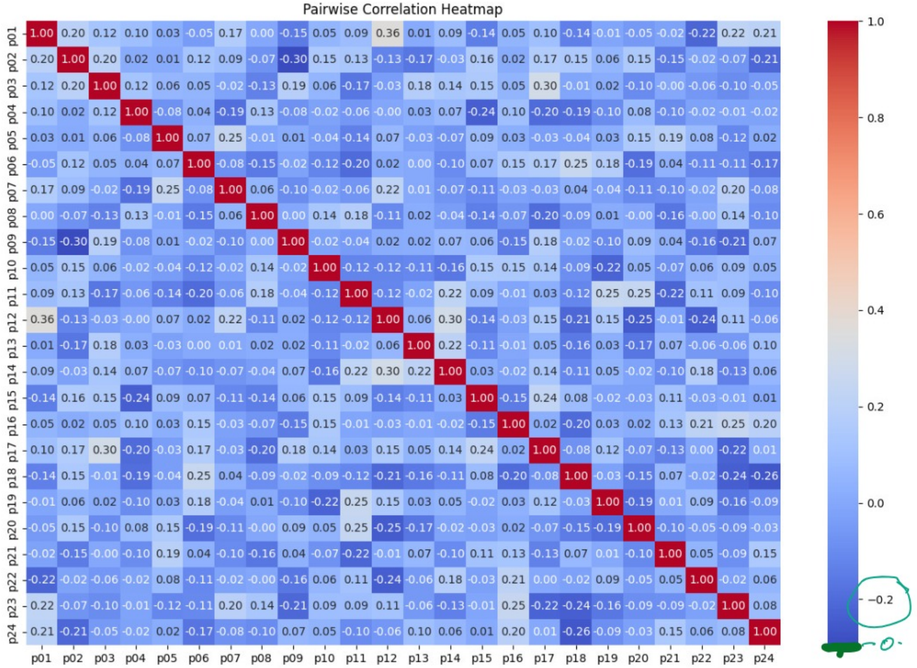
Correlation Heat Maps

(most simple analysis).



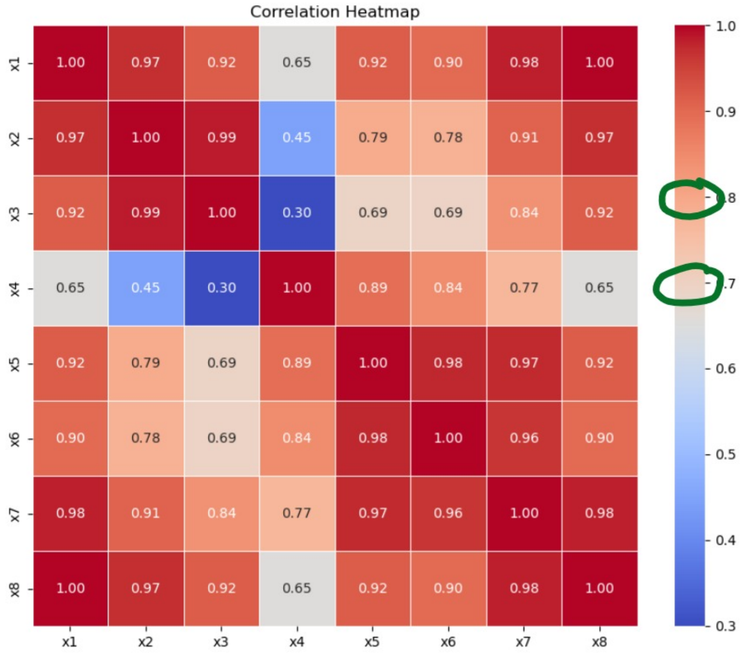
1 var.





Correlation heatmap of the 24 blood test parameters

No significant pairwise correlation can be detected based on this heatmap



Correlation heatmap of the 8 features of the 'curse of dimensionality' dataset

It is seen that multiple groups of features (eg. x1, x2, x3 / x5, x6, x7 / etc.) have high correlation - hence many in these groups are potential candidates for 'dropping' based on further analysis, like VIF

Variance Inflation Factor (VIF)

Multicollinearity Detection
Variance Inflation Factor (**VIF**)

The Variance Inflation Factor (VIF) quantifies how much the variance of a regression coefficient is inflated due to multicollinearity in the model. Mathematically, the VIF for a particular feature is calculated by performing a linear regression of that feature on all the other features.

For a given feature X_i, the VIF is defined as:

$$\text{VIF}(X_i) = \frac{1}{1-R_i^2}$$

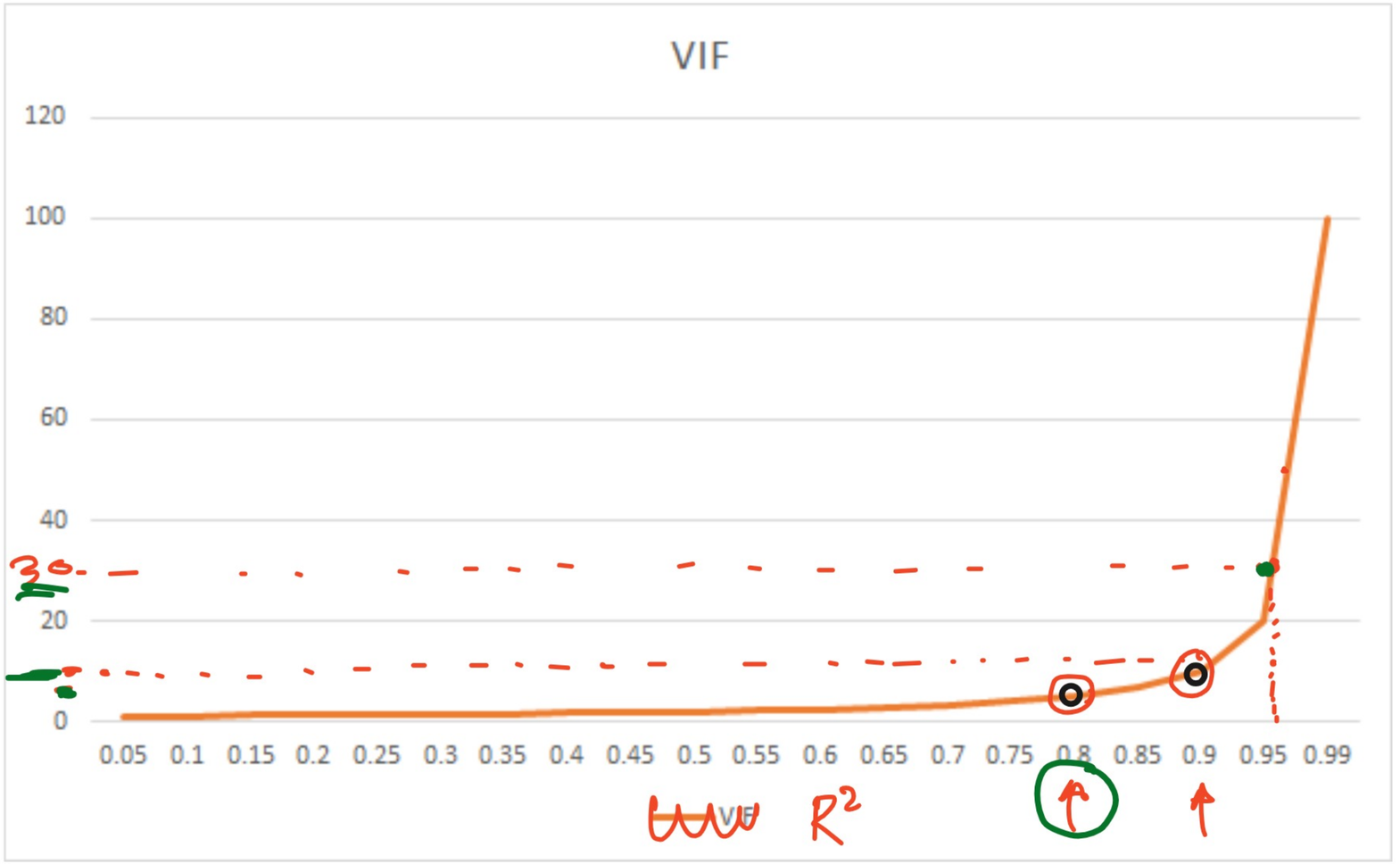
Where:

- R_i^2 is the R^2 value obtained by regressing the feature X_i against all other features.

1. For each feature X_i in your dataset, run a linear regression using X_i as the dependent variable and all other features as independent variables.
2. Obtain the R^2 value from this regression, denoted as R_i^2.
3. Calculate the VIF for X_i using the formula above.

A VIF of 1 indicates that there's no multicollinearity, whereas a VIF above 1 suggests that the feature is correlated with other features. A common rule of thumb is that a VIF above 10 indicates high multicollinearity, but this threshold can vary depending on the context.

Note: VIF analysis is done only on the independent variables (x)



VIF v/s R-square

R-square	VIF
0.05	1.052631579
0.1	1.111111111
0.15	1.176470588
0.2	1.25
0.25	1.333333333
0.3	1.428571429
0.35	1.538461538
0.4	1.666666667
0.45	1.818181818
0.5	2
0.55	2.222222222
0.6	2.5
0.65	2.857142857
0.7	3.333333333
0.75	4
0.8	5
0.85	6.666666667
0.9	10
0.95	20
0.99	100

The following strategy can be used to remove features:

1. Identify the features with high VIF values: The first step is to identify the features with high VIF values, typically above 5 or 10, which indicate that the feature is highly correlated with other features in the model.
2. Evaluate the importance of the features: The next step is to evaluate the importance of the features based on their relevance to the research question, their interpretability, and their contribution to the model's performance.
3. Remove the least important features: Based on the evaluation, the least important features can be removed from the model to reduce multicollinearity and improve the model's performance.
4. Re-evaluate the model: After removing the features, it is important to re-evaluate the model's performance using appropriate metrics and cross-validation techniques to ensure that the model is still accurate and reliable.

How to 'drop' features based on VIF analysis?

CAVEAT

It is important to note that **removing features can result in loss of information and may affect the interpretability of the model**. Therefore, it is important to carefully consider the trade-offs between multicollinearity and model performance and to choose the appropriate strategy based on the research question, the data, and the assumptions of the model

```
VIF Analysis Results:
    Feature         VIF
0       p01   49.636385
1       p02  102.484270
2       p03  326.095532
3       p04   16.112721
4       p05   21.841927
5       p06  274.166036
6       p07  160.435727
7       p08  288.760284
8       p09  329.408597
9       p10  867.641924
10      p11   14.915806
11      p12  297.850825
12      p13  180.653012
13      p14  280.544036
14      p15   18.899792
15      p16  115.238058
16      p17   60.474670
17      p18  115.420677
18      p19   12.779019
19      p20   16.554402
20      p21   66.312282
21      p22   48.778575
22      p23    7.145011
23      p24    5.966092
```

```
Removed 'p10' with VIF: 867.64
Removed 'p03' with VIF: 323.12
Removed 'p06' with VIF: 269.48
Removed 'p12' with VIF: 264.66
Removed 'p09' with VIF: 248.58
Removed 'p08' with VIF: 198.00
Removed 'p14' with VIF: 186.65
Removed 'p07' with VIF: 132.80
Removed 'p13' with VIF: 109.28
Removed 'p16' with VIF: 95.60
Removed 'p02' with VIF: 79.25
Removed 'p18' with VIF: 57.87
Removed 'p21' with VIF: 44.11
Removed 'p17' with VIF: 36.90
Removed 'p01' with VIF: 25.81
Removed 'p22' with VIF: 25.55
Removed 'p05' with VIF: 14.53
Removed 'p20' with VIF: 10.49
All VIF values below threshold.

Remaining VIF values:
  Feature       VIF
0     p04  7.380432
1     p11  8.535019
2     p15  8.473799
3     p19  7.827103
4     p23  4.057173
5     p24  3.822225
```

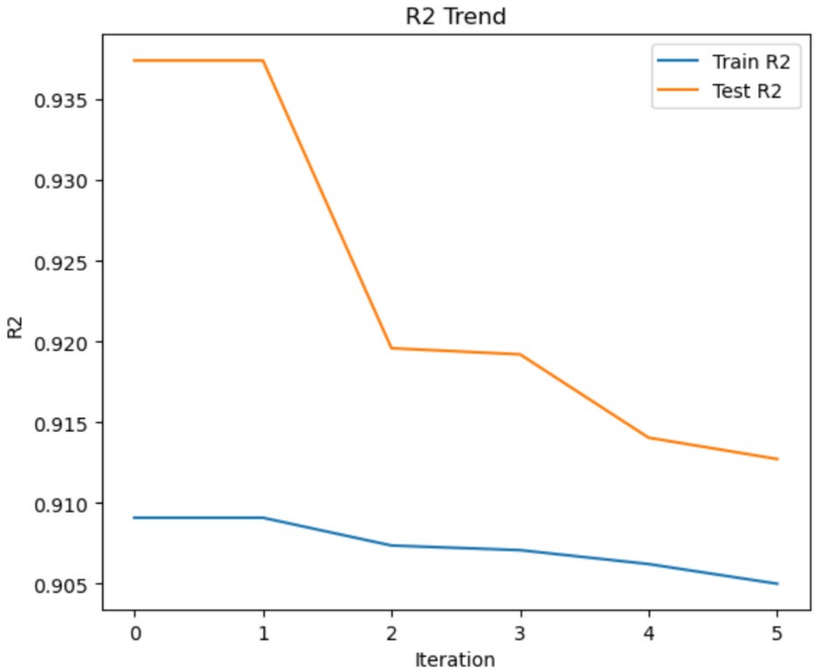
With VIF threshold = 10

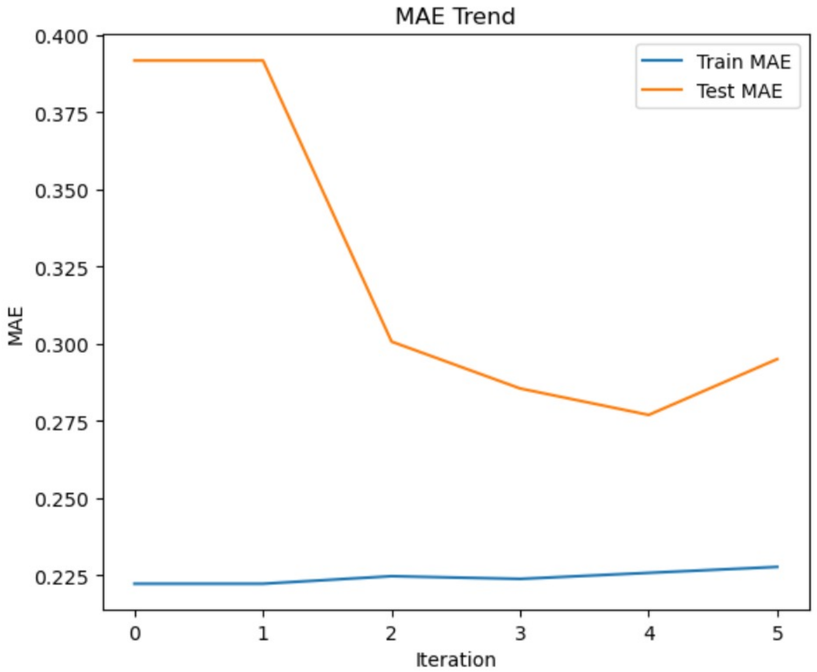
```
Removed 'p10' with VIF: 867.64
Removed 'p03' with VIF: 323.12
Removed 'p06' with VIF: 269.48
Removed 'p12' with VIF: 264.66
Removed 'p09' with VIF: 248.58
Removed 'p08' with VIF: 198.00
Removed 'p14' with VIF: 186.65
Removed 'p07' with VIF: 132.80
Removed 'p13' with VIF: 109.28
Removed 'p16' with VIF: 95.60
Removed 'p02' with VIF: 79.25
Removed 'p18' with VIF: 57.87
Removed 'p21' with VIF: 44.11
Removed 'p17' with VIF: 36.90
Removed 'p01' with VIF: 25.81
Removed 'p22' with VIF: 25.55
Removed 'p05' with VIF: 14.53
Removed 'p20' with VIF: 10.49
Removed 'p11' with VIF: 8.54
Removed 'p15' with VIF: 7.79
Removed 'p04' with VIF: 6.58
All VIF values below threshold.

Remaining VIF values:
  Feature       VIF
0     p19  3.524640
1     p23  3.179654
2     p24  3.305996
```

With VIF threshold = 5

	R2 Train	R2 Test	F-stat	MAE Train	MAE Test	Dropped Column	VIF
0	0.909080	0.937386	101.415631	0.222215	0.391760	x1	inf
1	0.909080	0.937386	101.415631	0.222215	0.391760	x8	6.034365e+06
2	0.907361	0.919574	117.534635	0.224660	0.300672	x5	2.650951e+04
3	0.907075	0.919195	142.515493	0.223785	0.285486	x3	1.851679e+03
4	0.906214	0.914036	178.757177	0.225746	0.276917	x7	2.902148e+02
5	0.904995	0.912717	238.143808	0.227670	0.295009	None	NaN

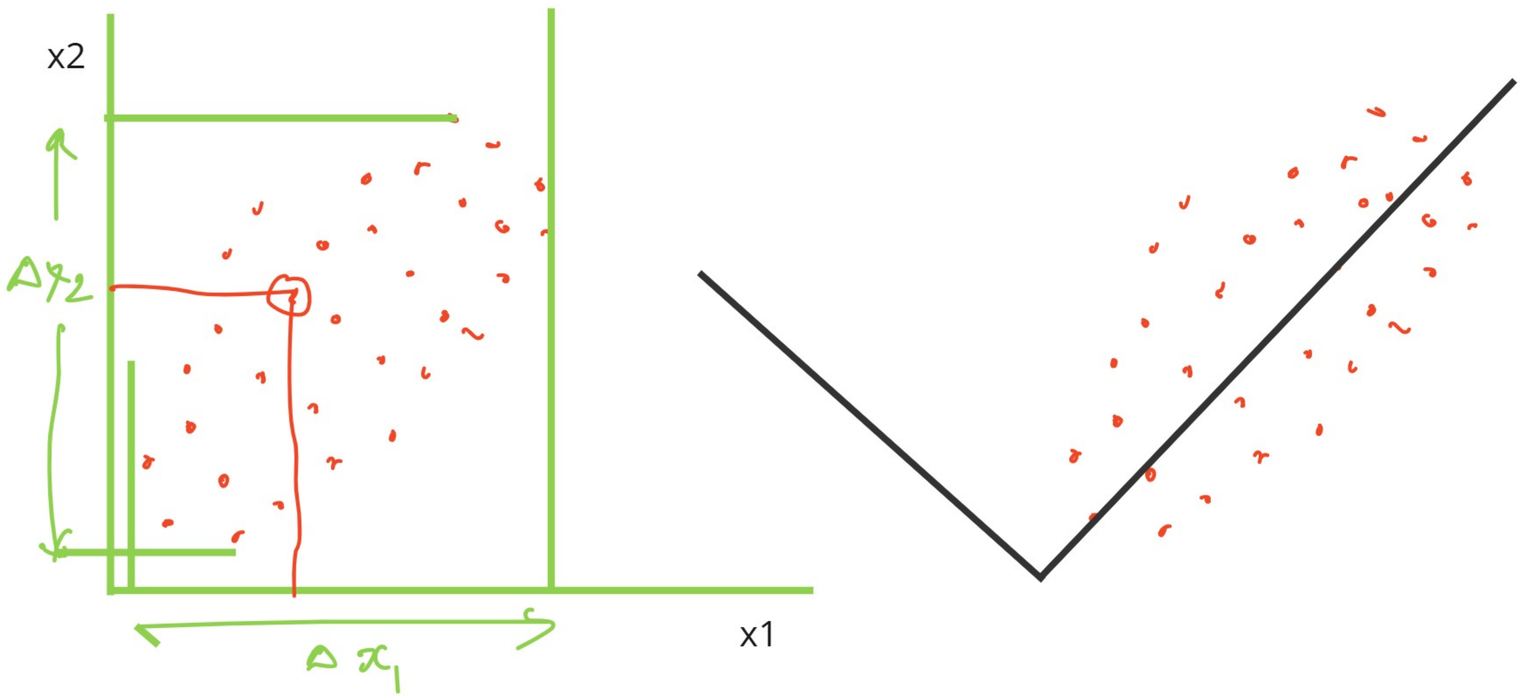




- It is important to take domain and overall model based decisions while deciding upon dropping features after steps like VIF analysis.
- In the adjoining case (using the curse of dimensionality data set), features have been dropped based on successive VIF analysis.
- It can be seen that model metrics like MAE initially improve (reduce) and then deterirate (increase) again.
- It can also be seen that with features progressively getting eliminated, R2 values keep dropping.
- Judicious decisions have to be made, therefore, about the which and how many features to drop - and this has to be done based on domain requirements, guided by model metrics

Principal Component Analysis

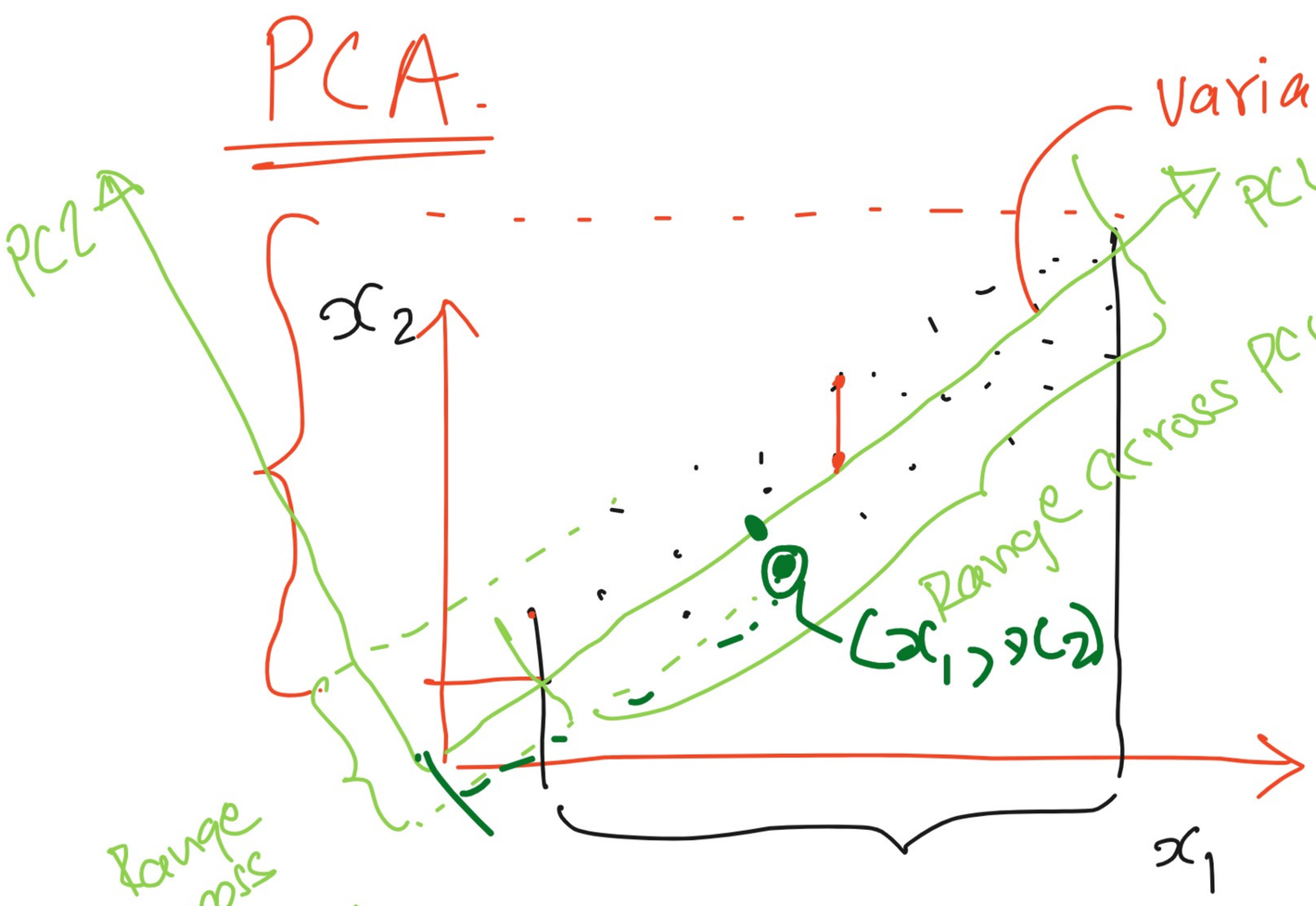
PCA - Principal Component Analysis

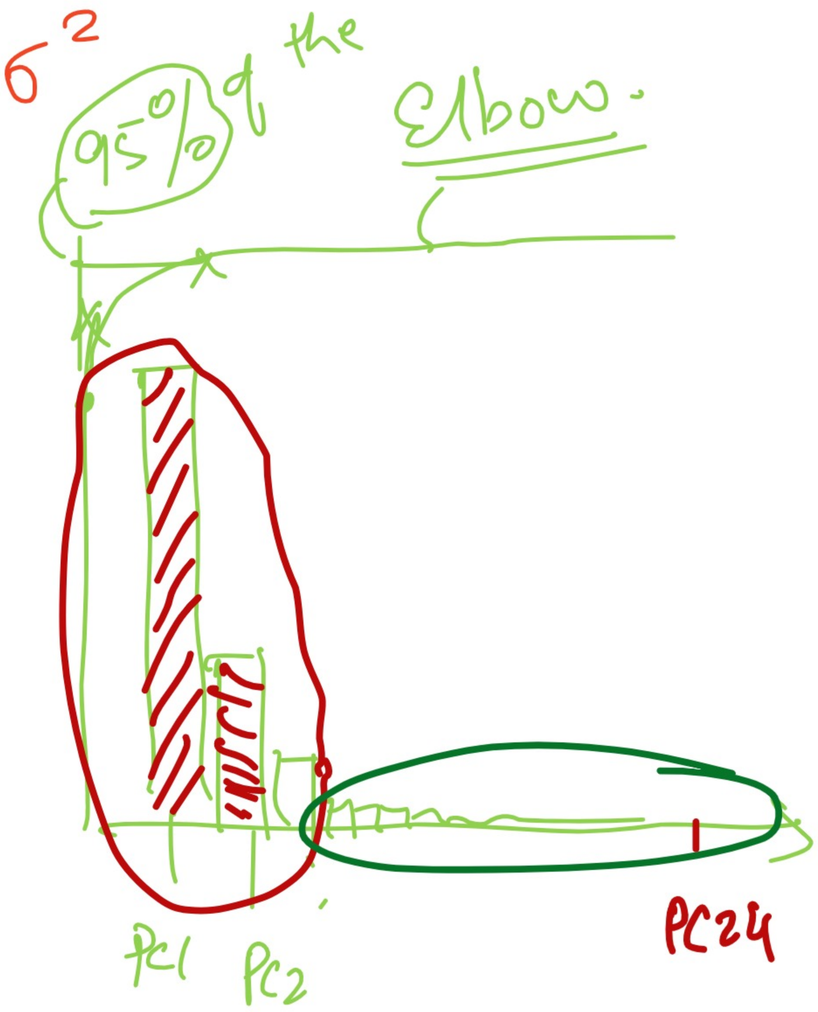


Principal Components are components along re-aligned axes such that PC1 'captures' or 'explains' or 'accounts for' the maximum variance, followed by PC2, PC3, and so on.

If the first few PCs cumulatively explain more than, say, 90% of the data variance, they may be sufficient for creating effective ML models. PCA thus serves the purpose of 'feature reduction'.

PCA.





$$y = f'(PC1, PC2, PC3, \cdots PC_k)$$

Downside → "NO EXPLAINABILITY"

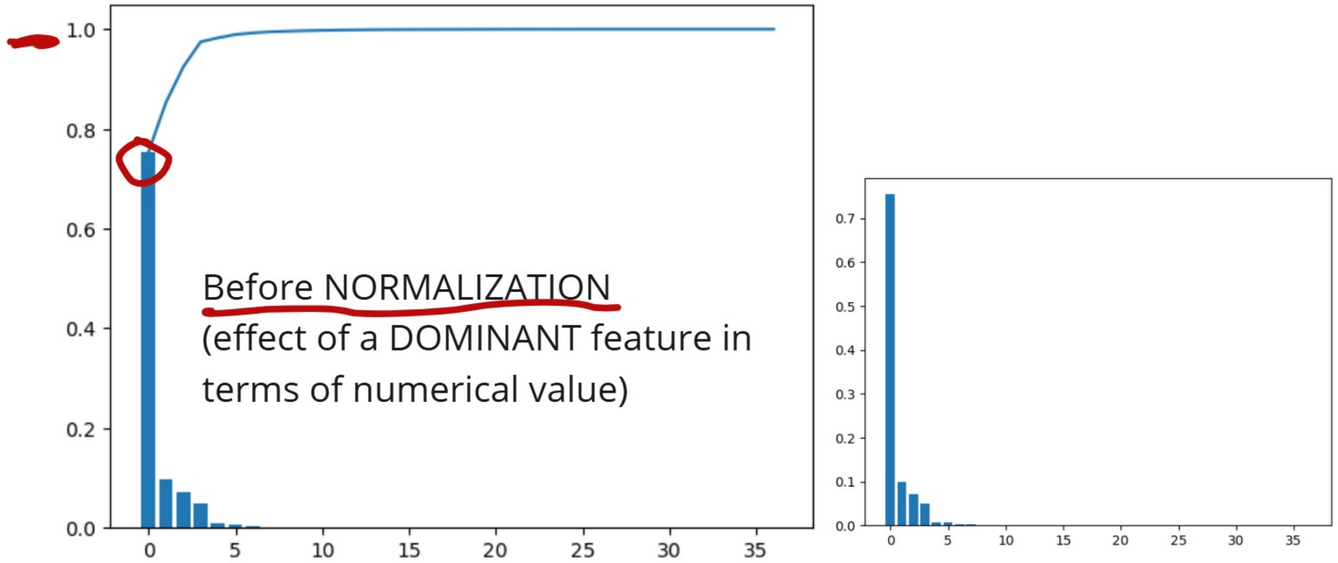
Each PC is a combination of all of the original axes.

$$PC1 \rightarrow f(x_1 \cdots \cdots x_k)$$
$$PC2 \rightarrow f(x_1 \cdots \cdots x_k)$$
$$\vdots$$
$$PC_k \rightarrow f(x_1 \cdots \cdots x_k)$$

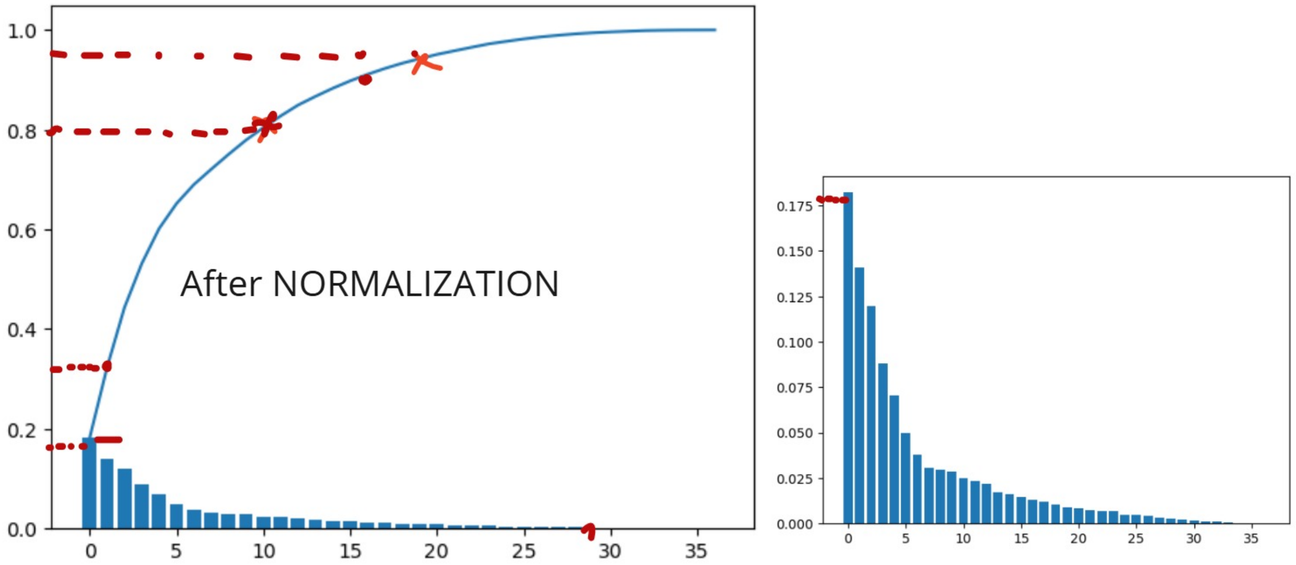
Importance of NORMALIZATION

Note : Data needs to be normalized before performing PCA

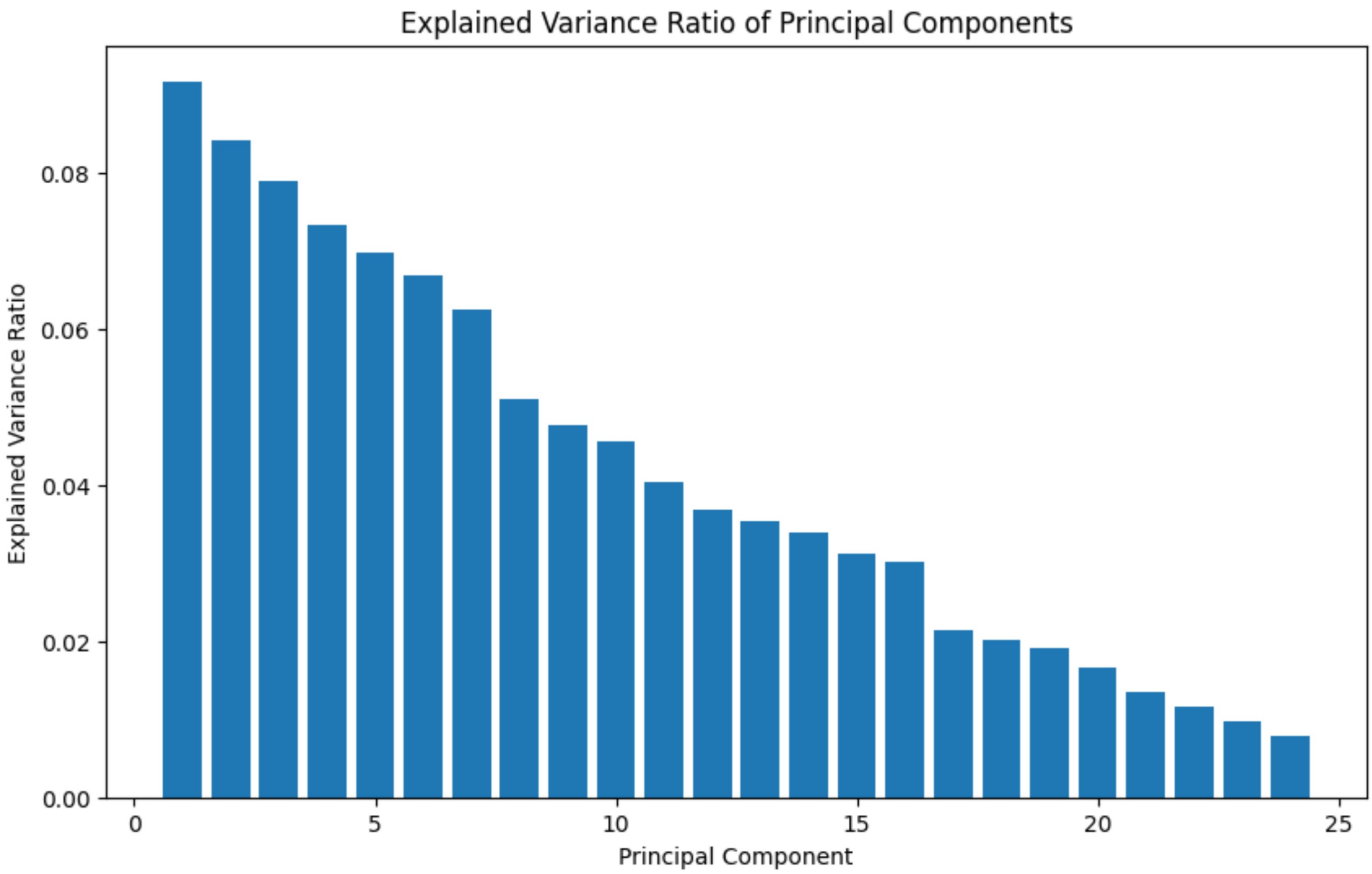
```
[7.54261365e-01 9.87869904e-02 7.12964455e-02 5.03510215e-02
 7.98749378e-03 6.55441064e-03 3.36504308e-03 2.09332441e-03
 1.18752903e-03 1.01662107e-03 7.74969179e-04 5.43124838e-04
 4.13982528e-04 3.22268906e-04 1.99904846e-04 1.82358764e-04
 1.22357384e-04 9.84701179e-05 8.87915581e-05 8.02456161e-05
 7.55522105e-05 6.24074825e-05 5.04485632e-05 3.08481492e-05
 1.86827712e-05 1.49416419e-05 1.33783774e-05 2.66135642e-06
 2.02403308e-06 9.51393151e-07 7.49623526e-07 2.39654410e-07
 1.99355029e-07 8.26718788e-08 7.06173620e-08 4.41103905e-08
 2.33592435e-11]
```

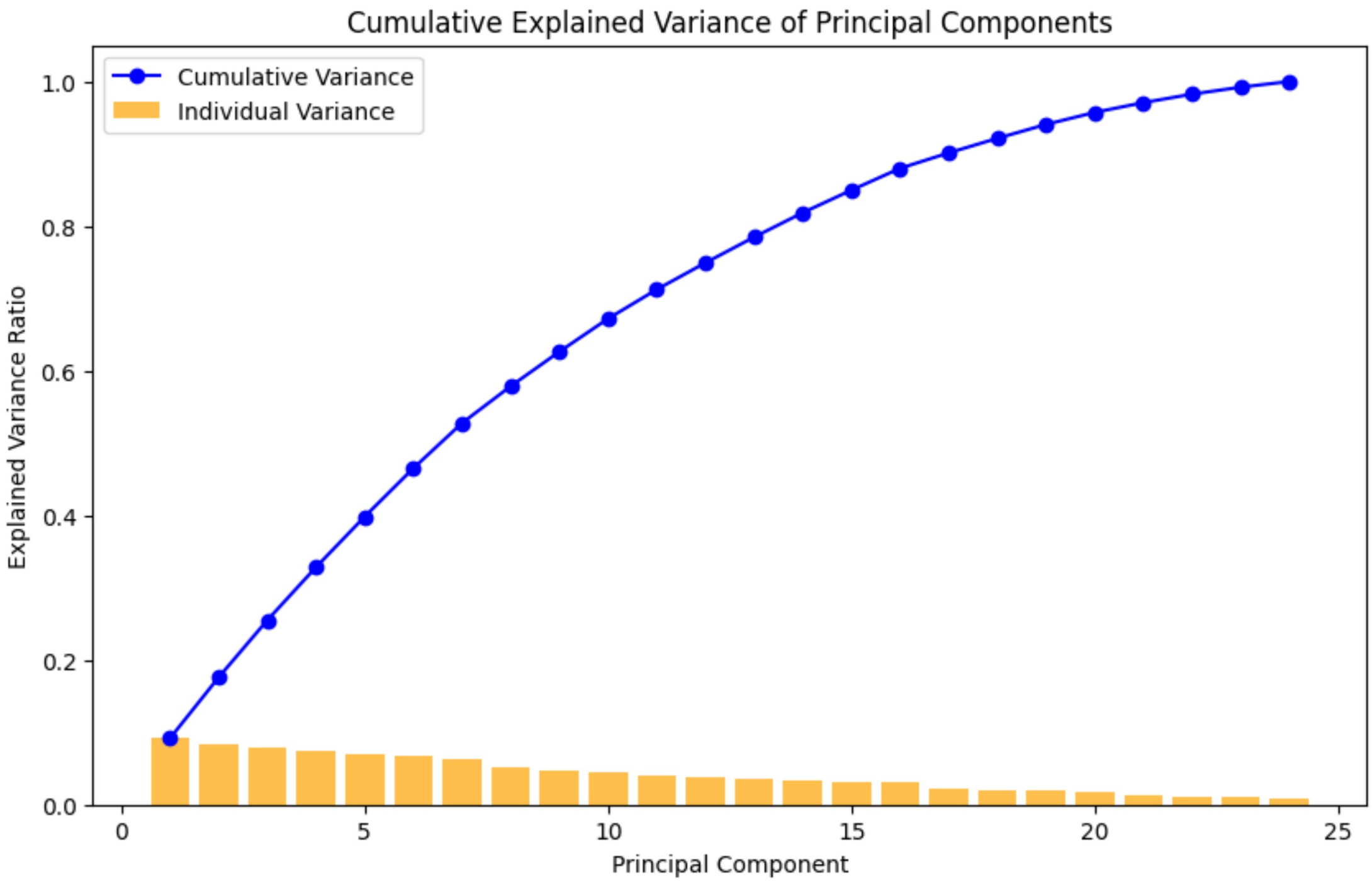



```
[1.82187813e-01 1.41144034e-01 1.19882002e-01 8.83869333e-02
 7.02808938e-02 4.99639610e-02 3.80999668e-02 3.08267651e-02
 2.98342213e-02 2.84815612e-02 2.49268628e-02 2.35483667e-02
 2.17418607e-02 1.73831972e-02 1.63501067e-02 1.47978112e-02
 1.30720181e-02 1.18307523e-02 1.05092250e-02 9.01490576e-03
 8.63737539e-03 7.21624942e-03 7.11038058e-03 6.89243449e-03
 5.03197986e-03 4.82978737e-03 4.14962998e-03 3.18816726e-03
 2.93711660e-03 2.14386597e-03 1.64215733e-03 1.35758835e-03
 1.28773381e-03 6.51744979e-04 3.97614261e-04 1.45522755e-04
 1.17392623e-04]
```



Principal Component Analysis of the 'blood test' data set
(It can be seen that there are NO dominant PCs)





Uses of PCA

Data understanding:

- PCA can be used to identify the most important variables in a dataset and to understand the relationships between them.
- PCA can be used to detect outliers and anomalies in the data by identifying data points that are far from the rest of the data.

Data visualization:

- PCA can be used to visualize high-dimensional data in a low-dimensional space, typically 2D or 3D, by reducing the dimensionality of the data while preserving the most important information.
- PCA can be used to identify clusters and patterns in the data that are not visible in the high-dimensional space.

Data simplification:

- PCA can be used to simplify the data by reducing the number of variables or features in the dataset while preserving the most important information.
- PCA can be used to remove noise and redundancy from the data by identifying the most important components and ignoring the rest.

Dimensionality reduction:

- PCA can be used to reduce the dimensionality of the data by finding a new set of variables that are smaller than the original set but still contain most of the information in the original set.
- PCA can be used to transform a large set of variables into a smaller one that still contains most of the information in the large set.

Machine learning:

- PCA can be used as a preprocessing step in machine learning to reduce the dimensionality of the data and to improve the performance of the model.
- PCA can be used to remove multicollinearity and to improve the interpretability of the model by identifying the most important variables.