ECON 441B : Intro Machine Learning Lab

Week 10, Lecture 10 | Principle Component Analysis

Sam Borghese

Friday, March 17th, 2023

1. PCA Intuition
2. PCA Algorithm
3. Plots and Interpretation
4. Coding
5. In-Class Assignment

PCA intuition

Unsupervised Learning

Unsupervised Learning is where the target variable is unknown

Is the target variable known?

Is the outcome variable “”?

	Yes (Supervised)	No (Unsupervised)
Discrete	**Classification**	**Clustering**
Continuous	**Regression**	**Dimension Reduction**

History

Principal component analysis (PCA), first proposed by Karl Pearson in 1901, is a classic method for extracting and distilling the inter-relationships among a large number of correlated variables. PCA is often regarded as the first true 'multivariate' statistical method.

$$(x_1, x_2, \ldots, x_n) \longrightarrow (PC_1, PC_2, \ldots, PC_m)$$

$$m \leq n$$

Definition

PCA is an unsupervised dimensionality reduction technique used to transform the original features of a dataset into a new set of uncorrelated features, called principal components. The primary goal of PCA is to reduce the dimensionality of the data while retaining as much information (variance) as possible. PCA does not consider the class labels or categories of the data points.

Reduce Dimensions

Linear Combinations

This dimensionality reduction is accomplished by creating new variables, Principle Components, that are linear combinations of the old variables

$$PC_1 = a_{1,1} \cdot x_1 + a_{2,1} \cdot x_2 + \ldots + a_{n.1} \cdot x_n$$

$$PC_2 = a_{1,2} \cdot x_1 + a_{2,2} \cdot x_2 + \ldots + a_{n,2} \cdot x_n$$

$$[PC] = A \cdot X$$

Why reduce dimensions?

1.) Purely as an unsupervised task

A.) Allow you to visualize your data

B.) Removal of Multicollinearity

C.) Feature importance in explaining variability

2.) Done to your independent variable before supervised learning

A.) Reduce overfitting of final model

B.) Create a simpler model (Occam's Razor, Law of Parsimony)

C.) Removes Multicollinearity

D.) Reduces training time

E.) Removes noise and excess variables (feature selection)

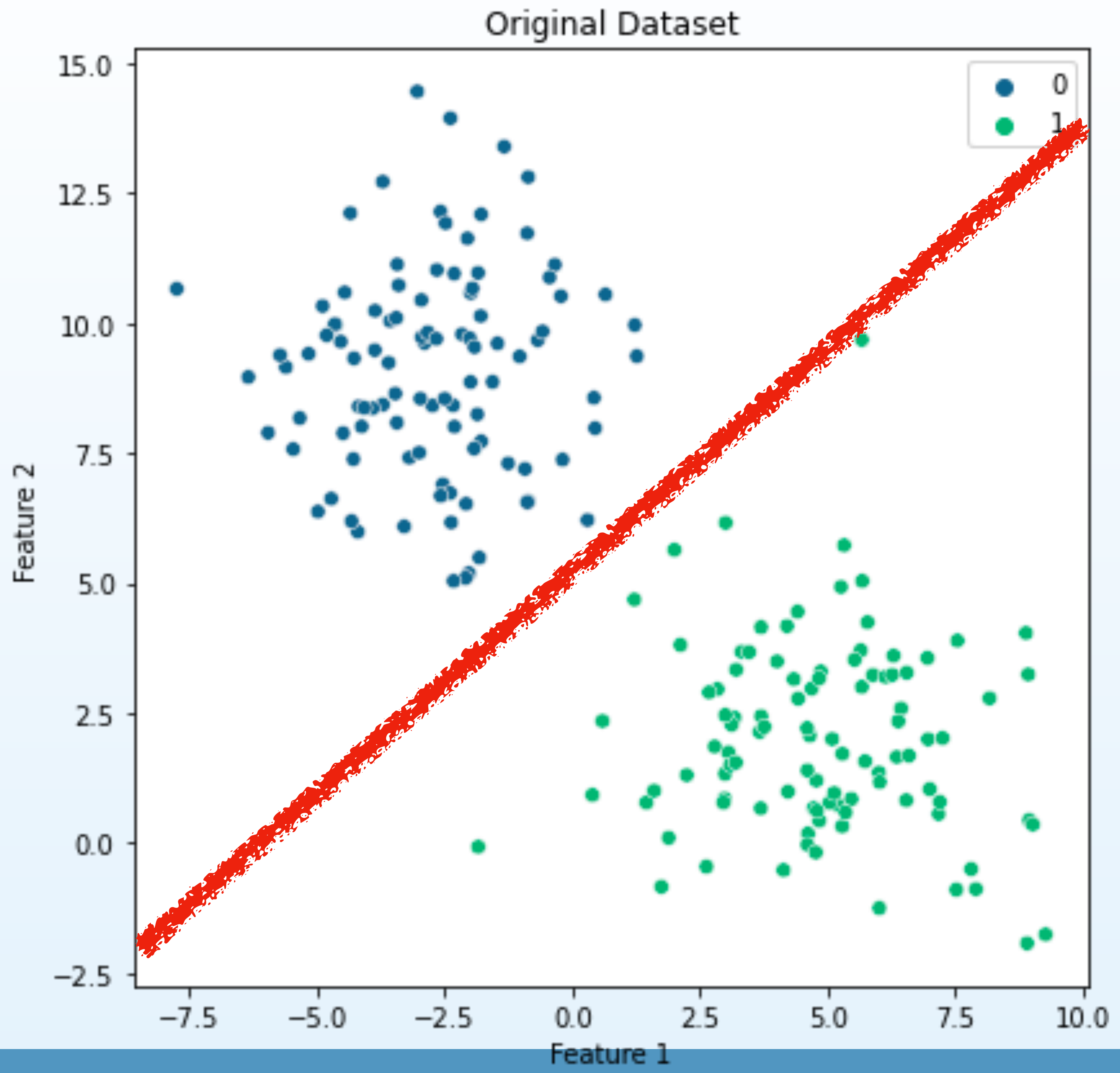
PCA Algorithm

Separable Clusters

Let us start with a case where we know the underlying clusters

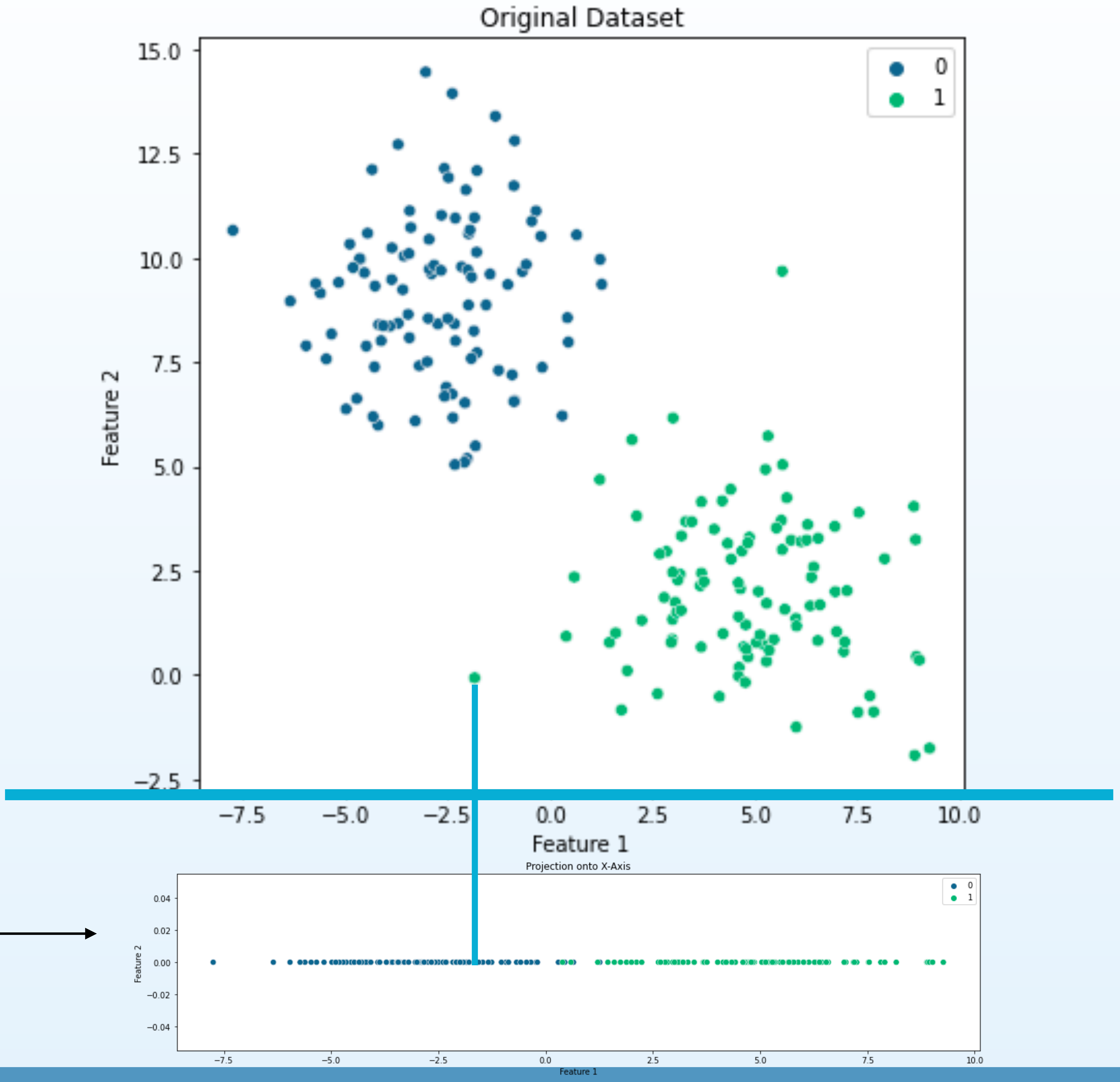
This Blue and Green gaussian distributions can be linearly separable using two dimensions

Is there a way to make this linearly separable using one dimension?



Projection onto a Line

Every point can be projected onto a line by the perpendicular intersection of the line and the point

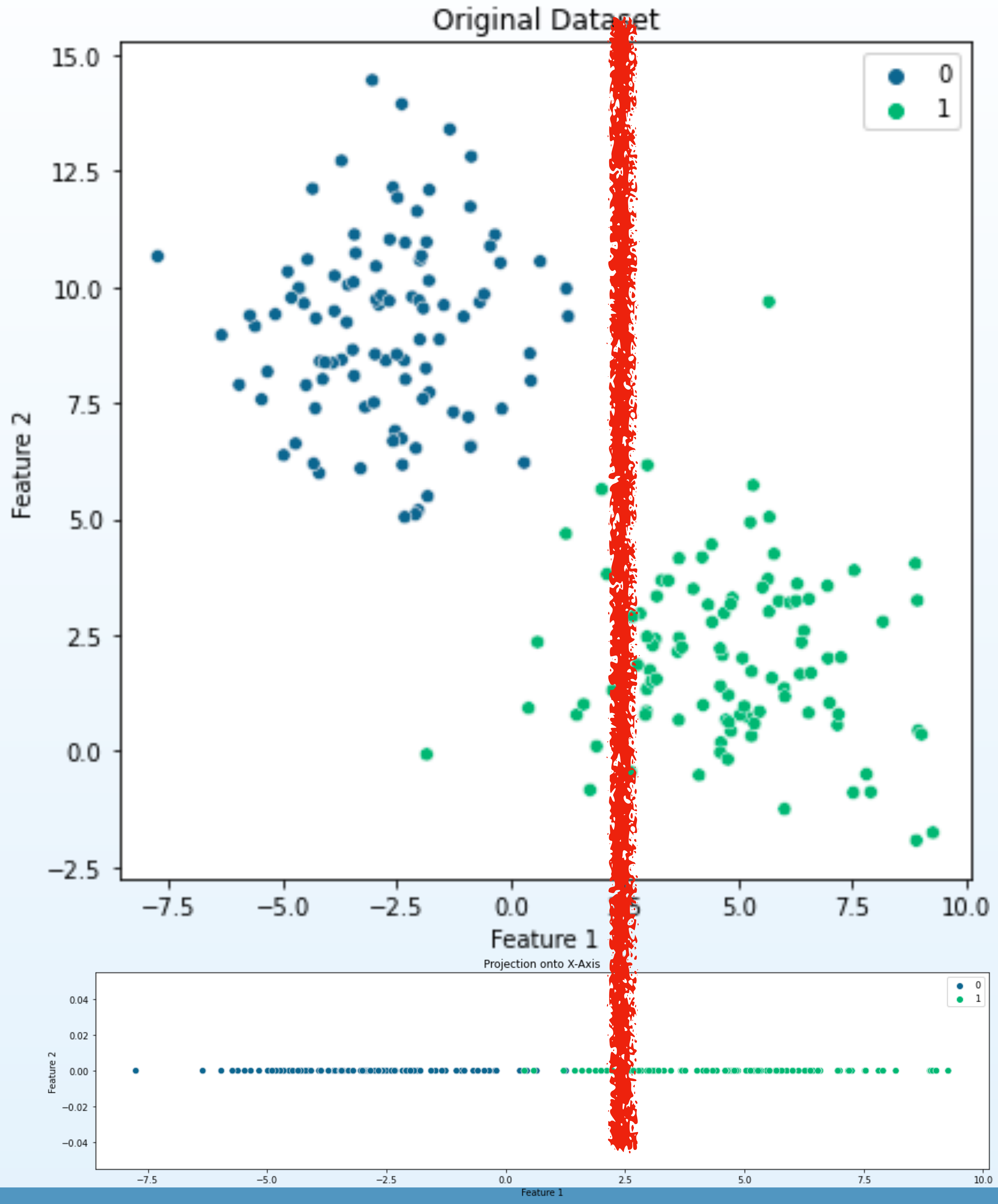


Projection onto the X-axis

Projection onto a Line

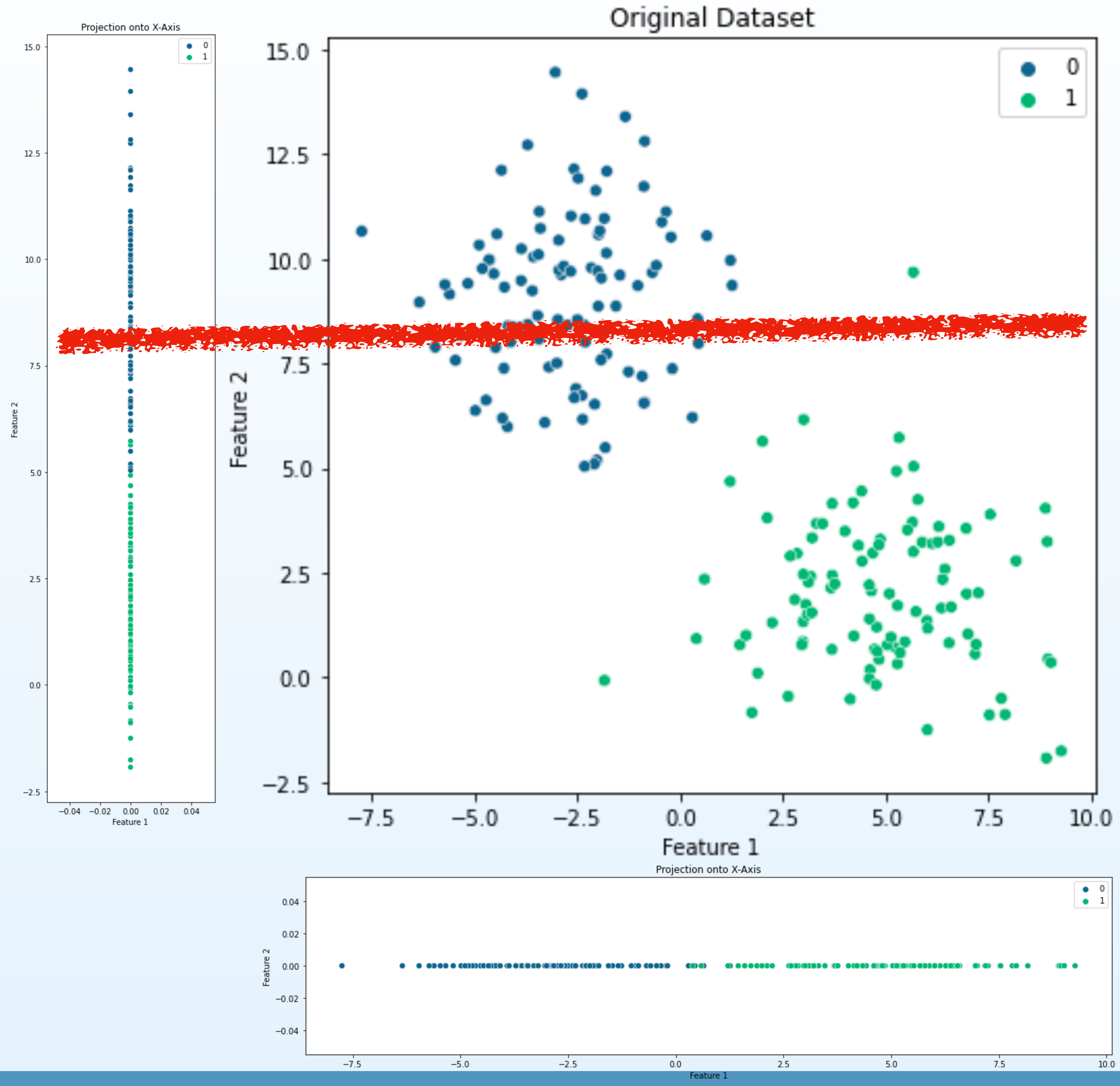
This data is not linearly separable on the x-axis projection

=> We lose variance in the data projecting it onto one dimension



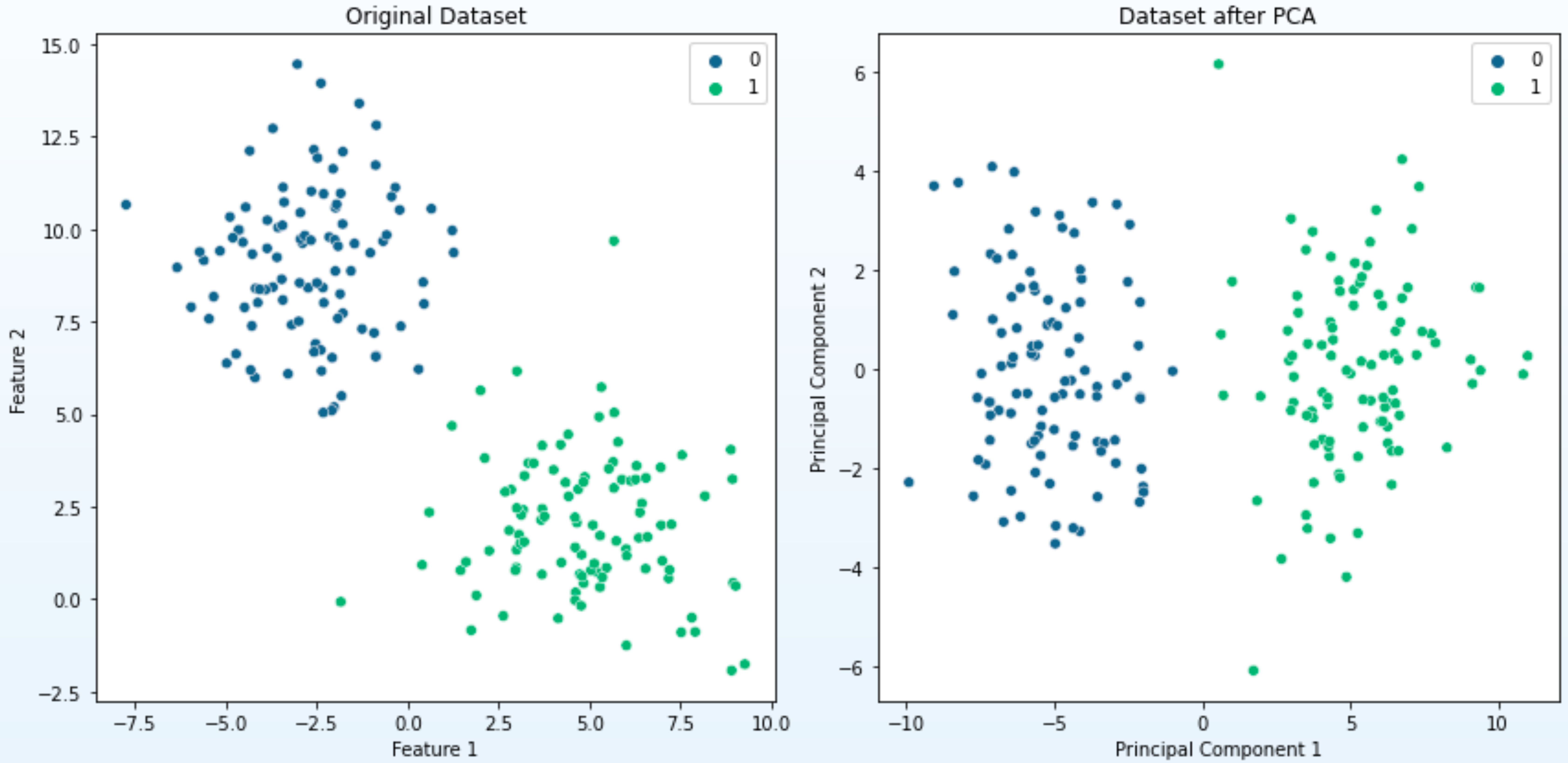
Projection onto a Line

The same is true for the y-axis



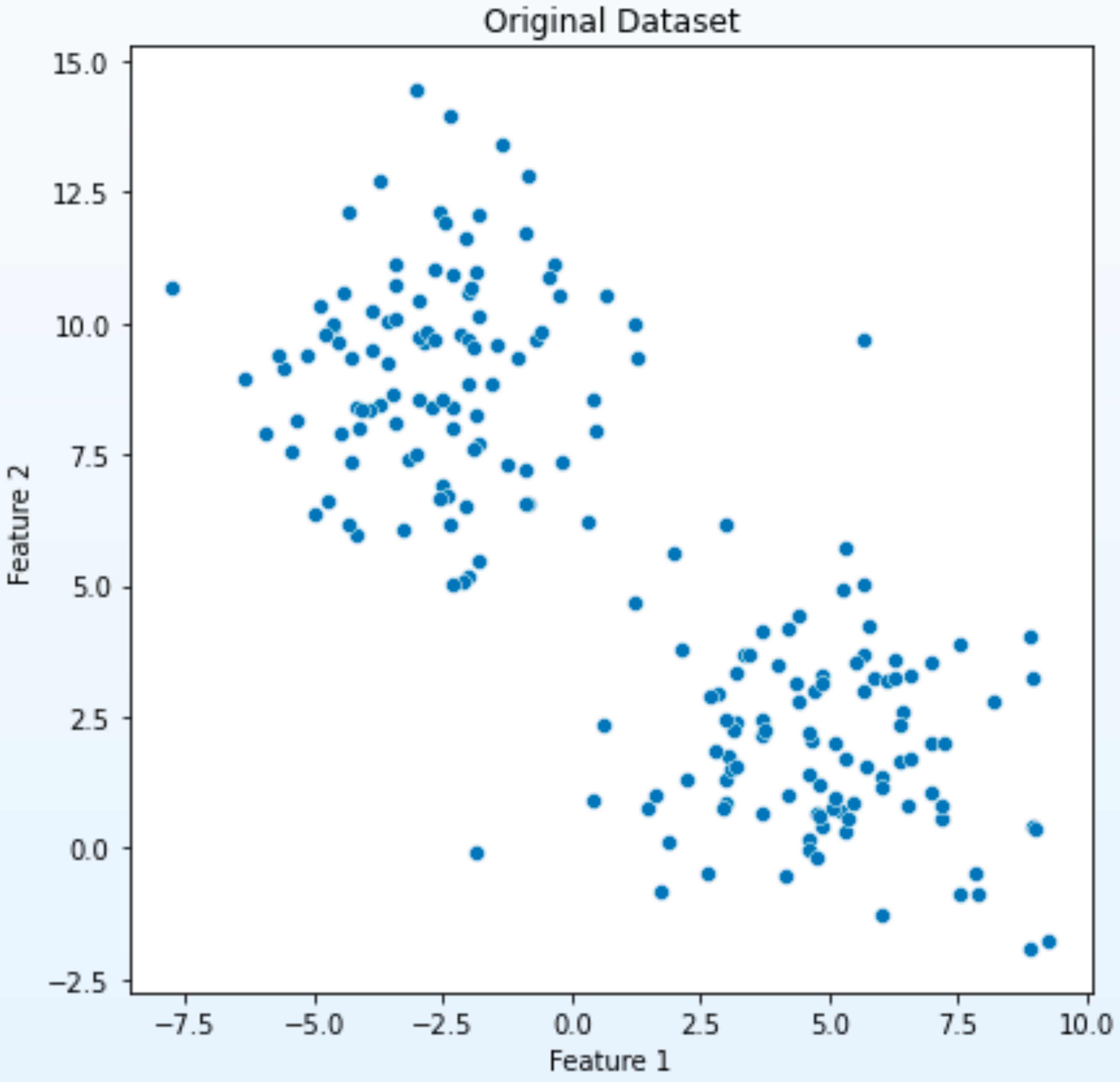
PCA Outcome

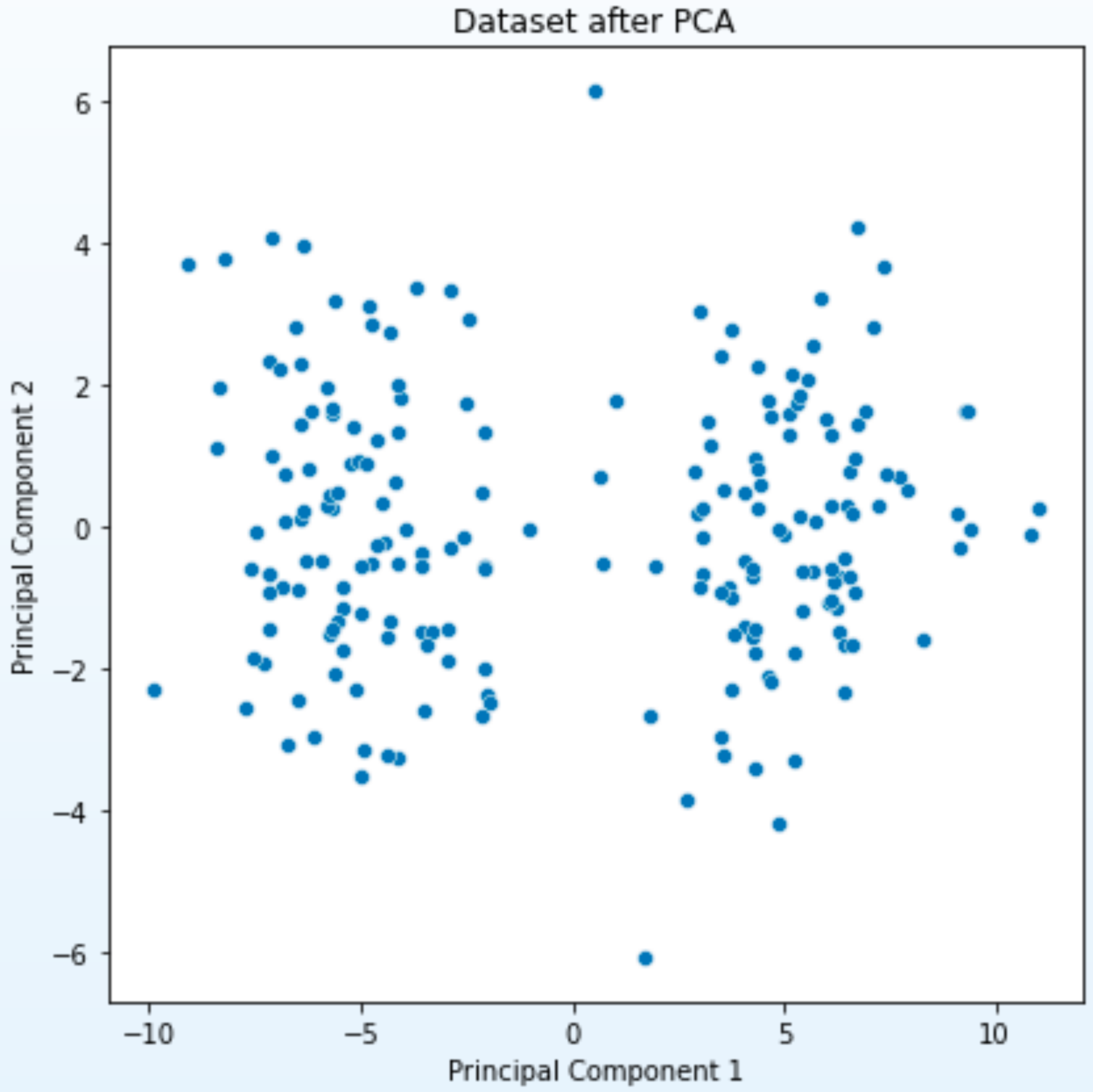
The data is linearly separable



PCA Outcome

Key to Note :
The outcome does not change whether or not we know the underlying.

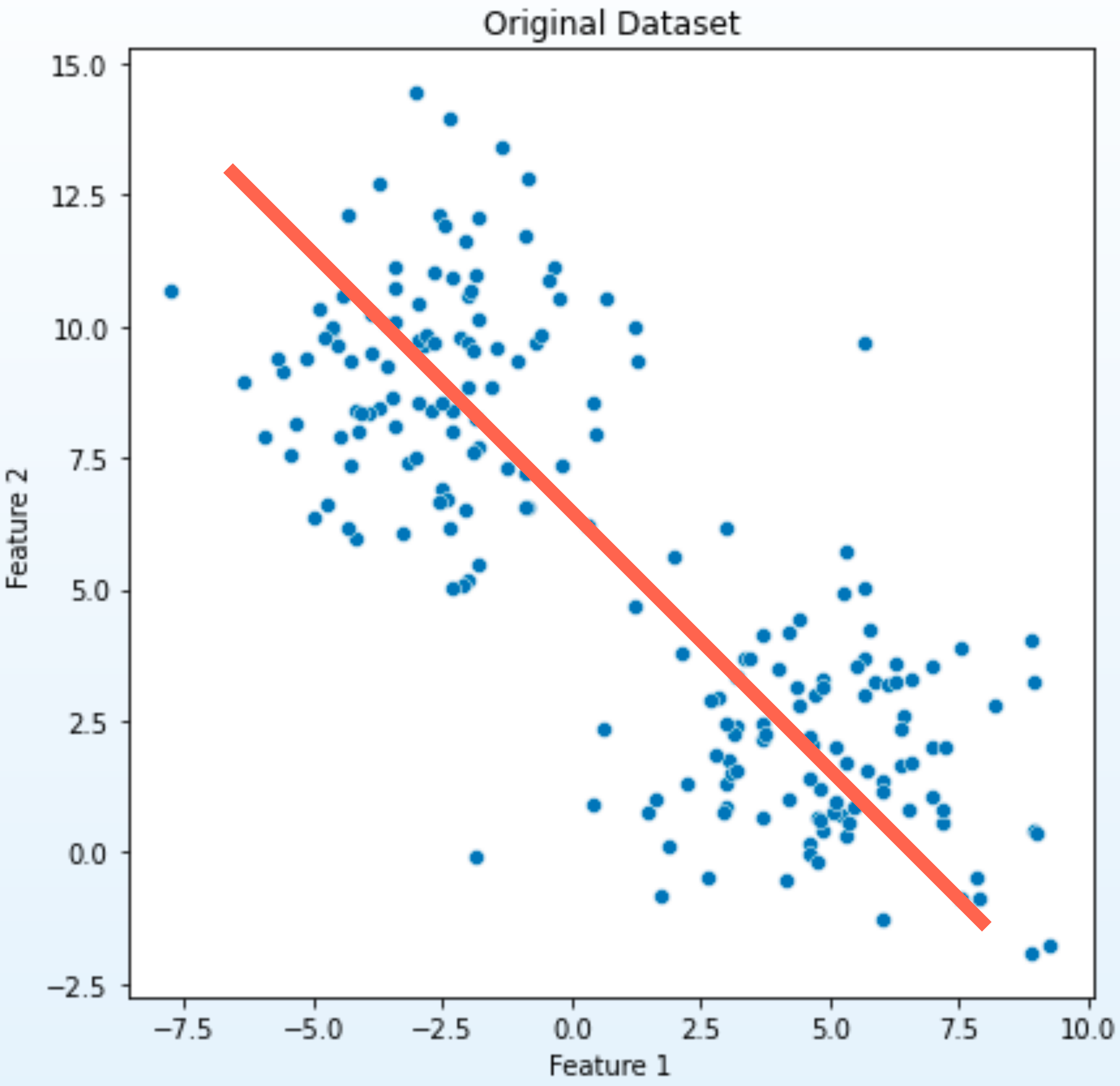




Alternate Interpretation

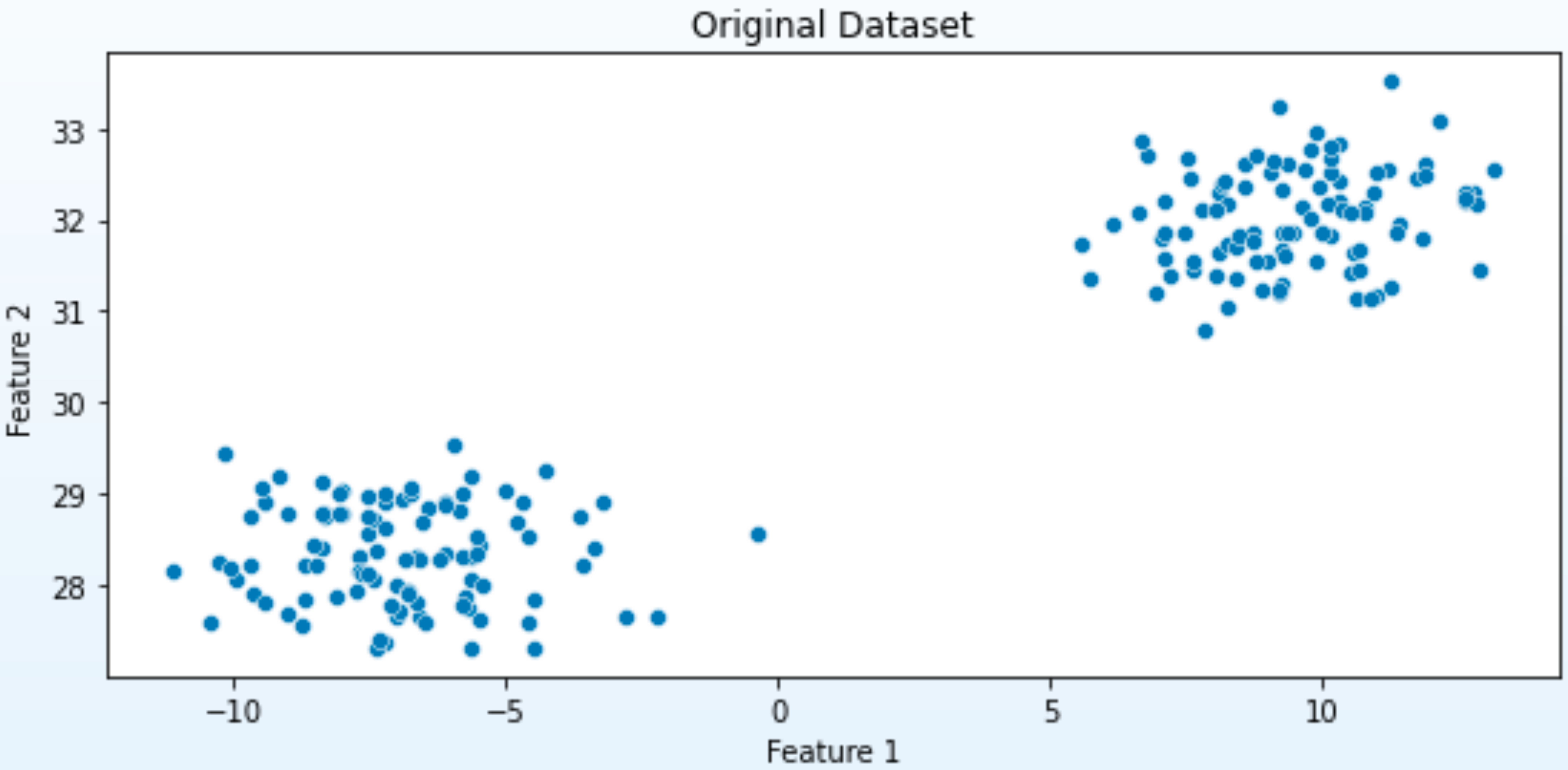
We see Feature 1 and Feature 2 have a high correlation

PCA makes 1 variable that accounts for this correlation



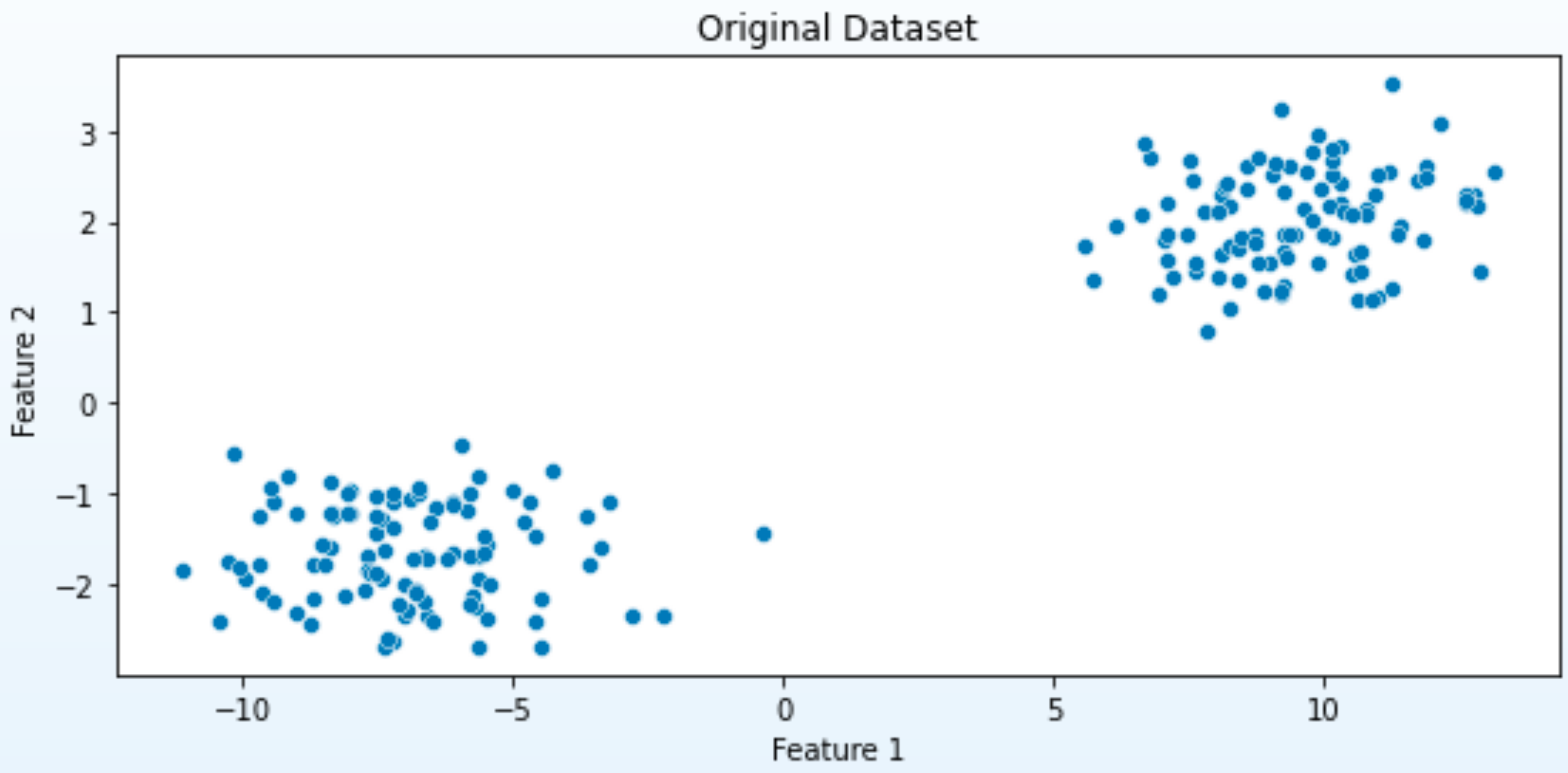
PCA ALGORITHM

Say we have some data and we are looking for "a linear combination of features to best describe the variance of this data



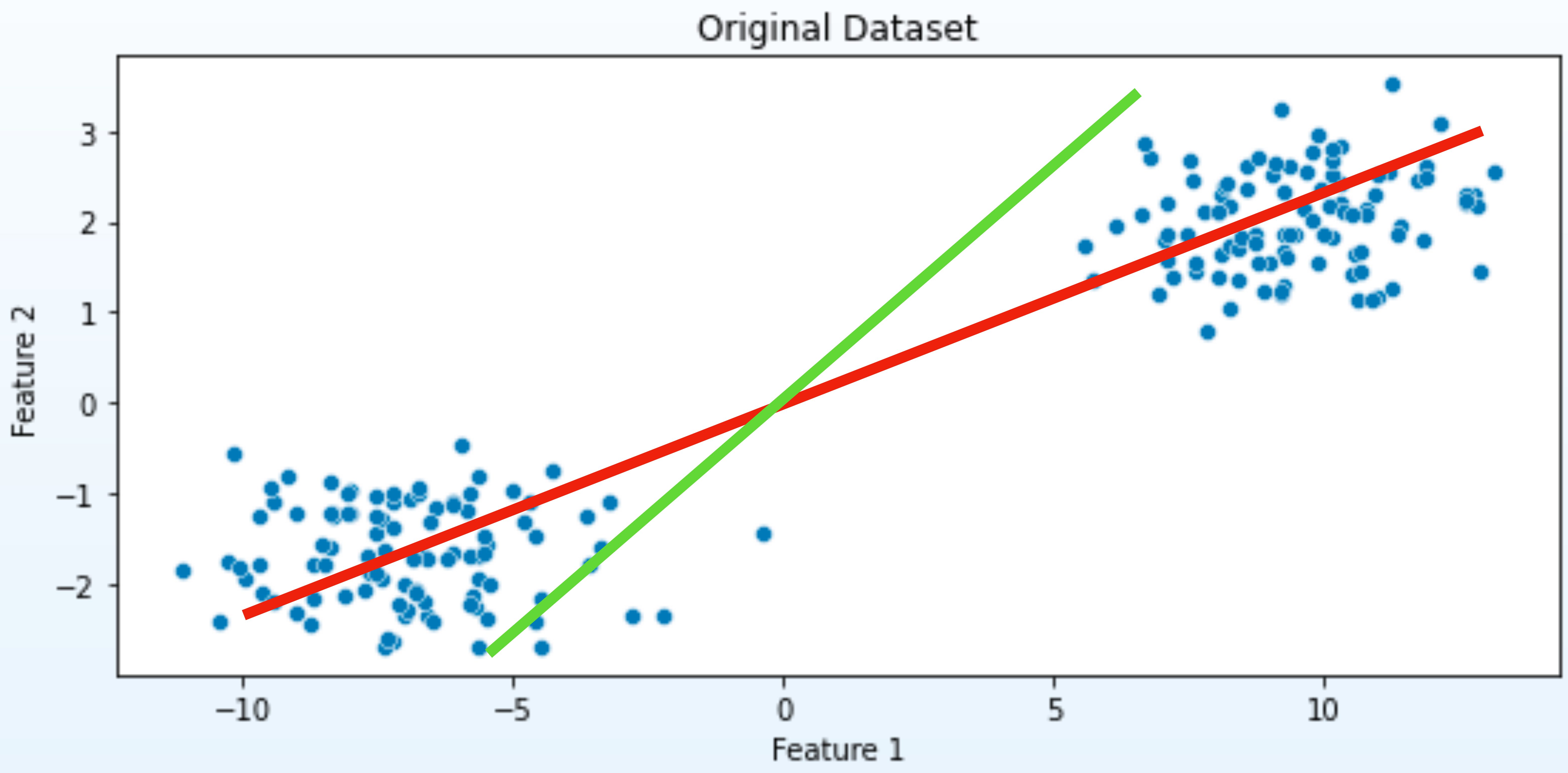
PCA ALGORITHM

Step 1 : Move the center of the data to the origin



PCA ALGORITHM

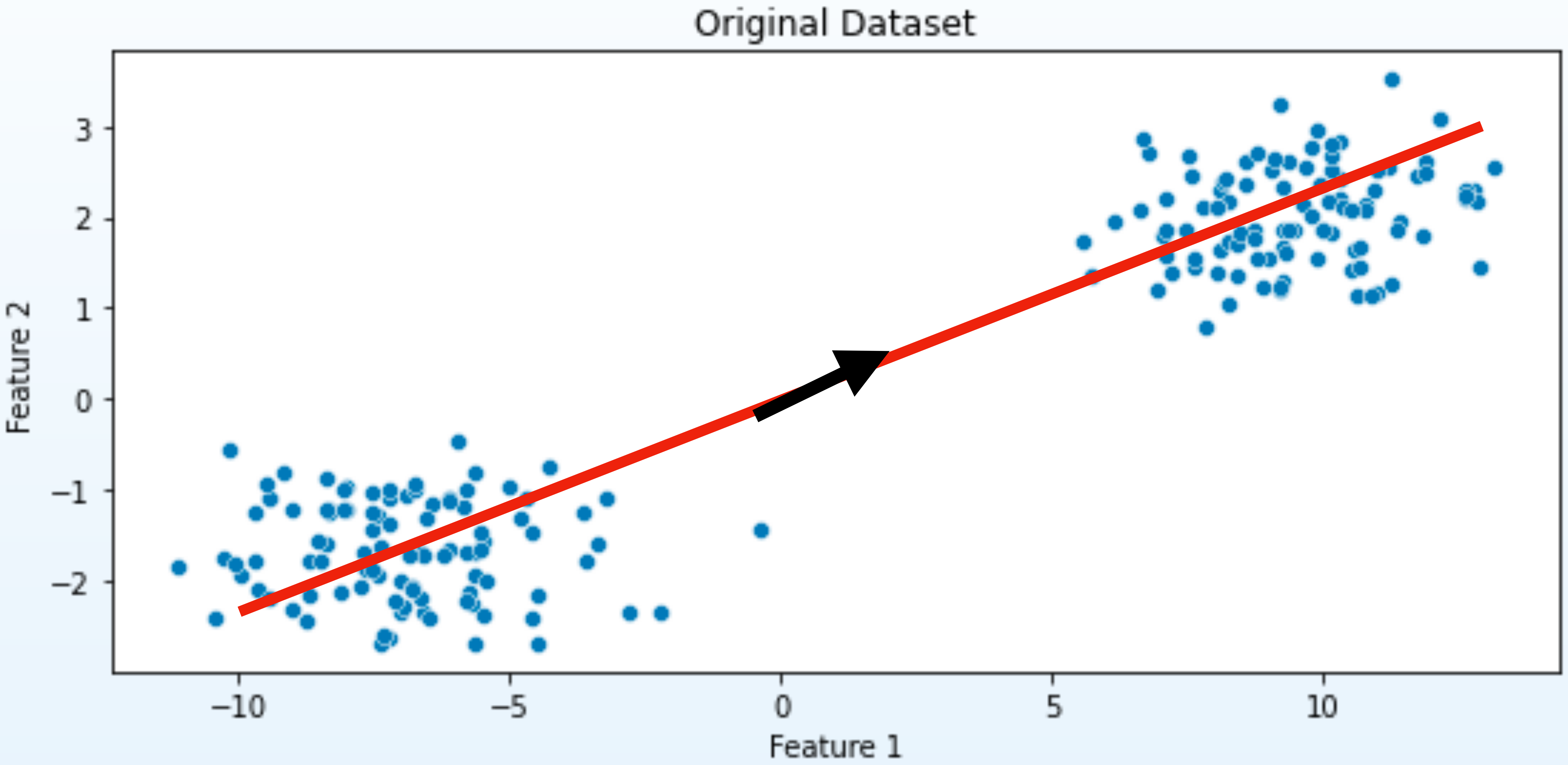
Step 2 : Find a line that has the closest distance to each point based on the perpendicular distance



PCA ALGORITHM

Step 3 : Calculate the Eigen vector (unit vector) in that direction

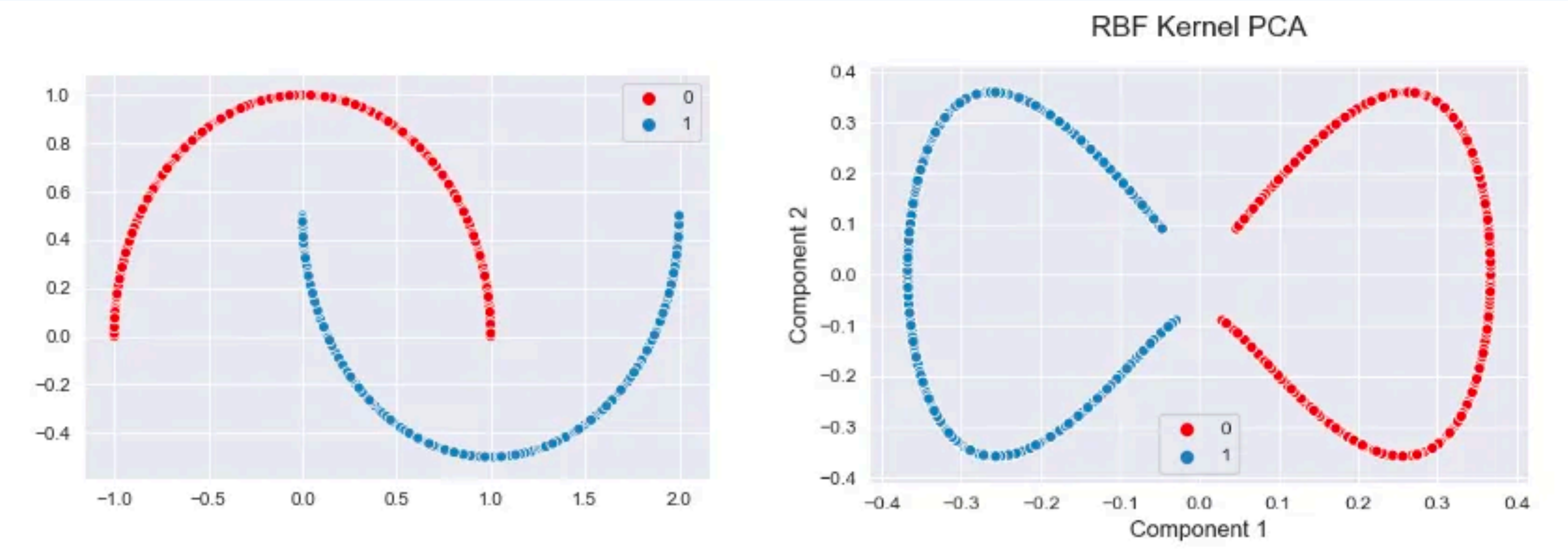
.83 Feature 1
.55 Feature 2



The variance (spread) of this data is better explained by Feature 1 and best explained by the linear combination above

Beyond PCA : There are many other models

There are many other dimensionality reduction techniques not covered here.

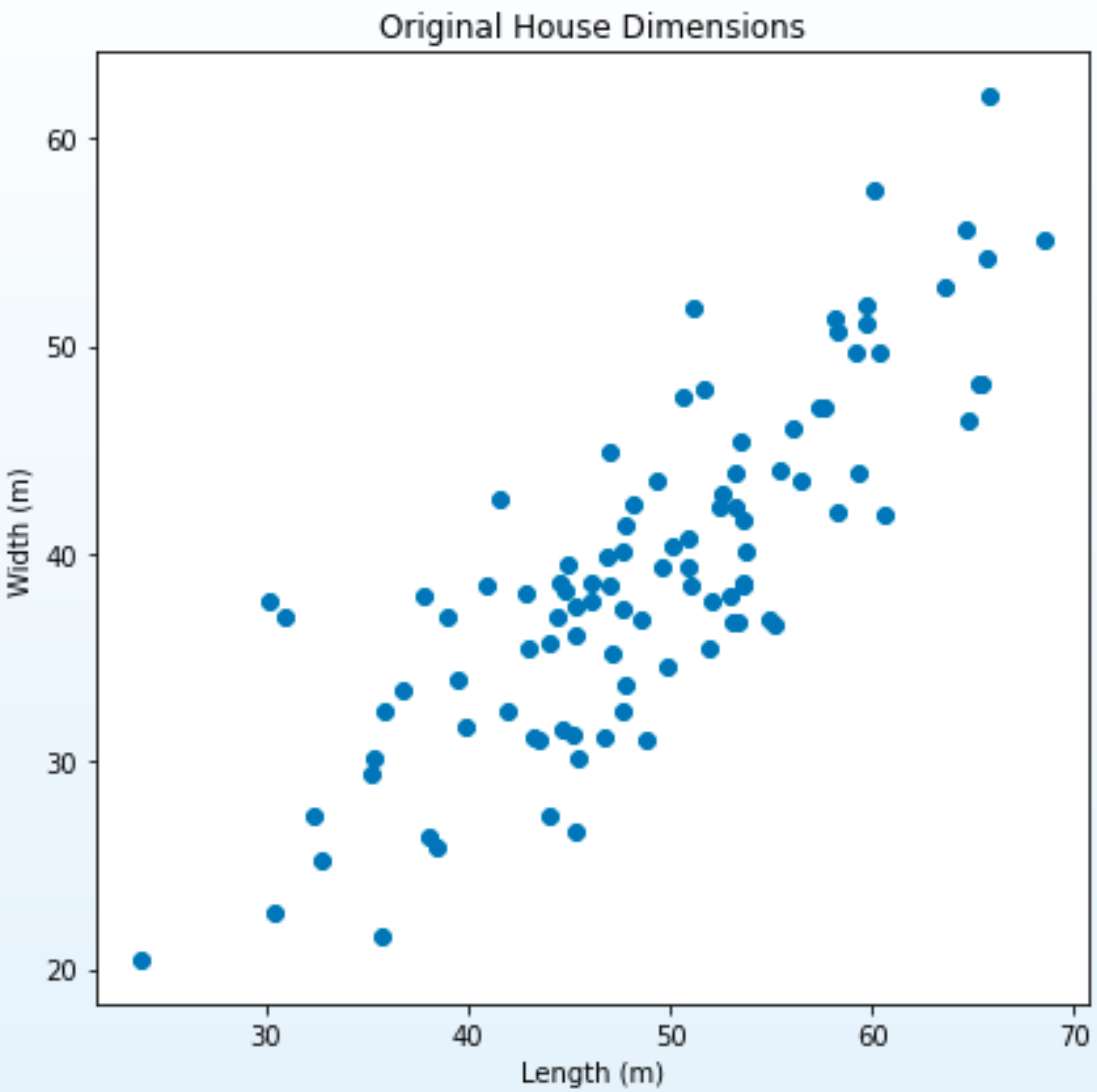


Plots and Interpretation

PCA Example 1

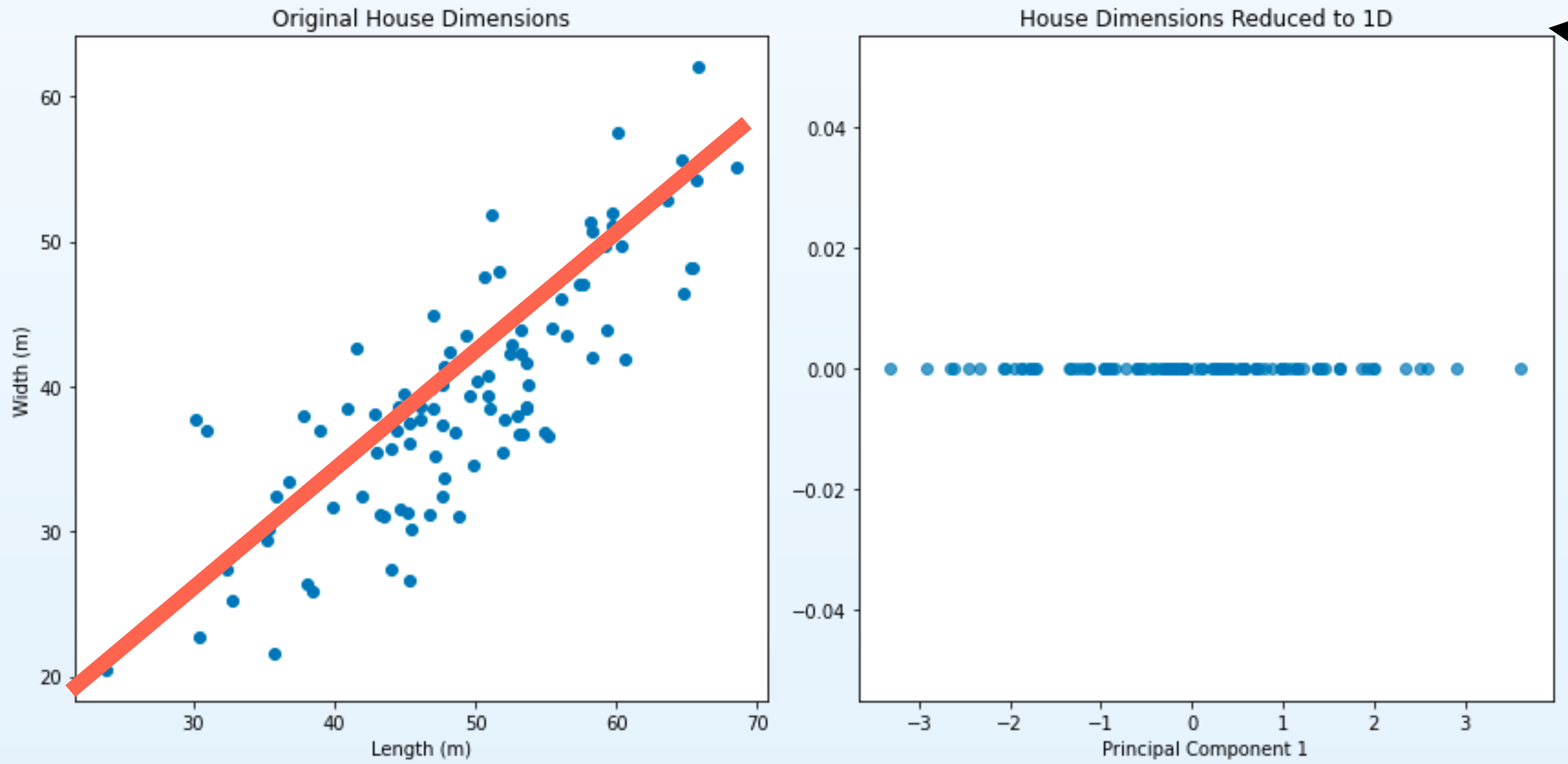
You have house dimensions
Width and Length

These are both highly correlated
with Area



PCA Example 1

1 single Principle component is created can contain most of the information of two dimensions



AKA Score Plot

PCA Example 1

Loadings : How much of the original features ate in the Principle Components?

```
loadings = pca.components_
```

```
Loadings: [[-0.70710678 -0.70710678]]
Contribution of Length to Principal Component 1: -0.71
Contribution of Width to Principal Component 1: -0.71
```

$$loadings = [PC_1, PC_2, \ldots, PC_n]$$

$$PC_i = [feature_1, feature_2, \ldots, feature_m]$$

PCA Example 1

Loadings : How much of the original features ate in the Principle Components?

```
loadings = pca.components_
```

```
Loadings: [[-0.70710678 -0.70710678]]
Contribution of Length to Principal Component 1: -0.71
Contribution of Width to Principal Component 1: -0.71
```

```
np.sqrt(loadings[0, 0]**2 + loadings[0, 1]**2)
```

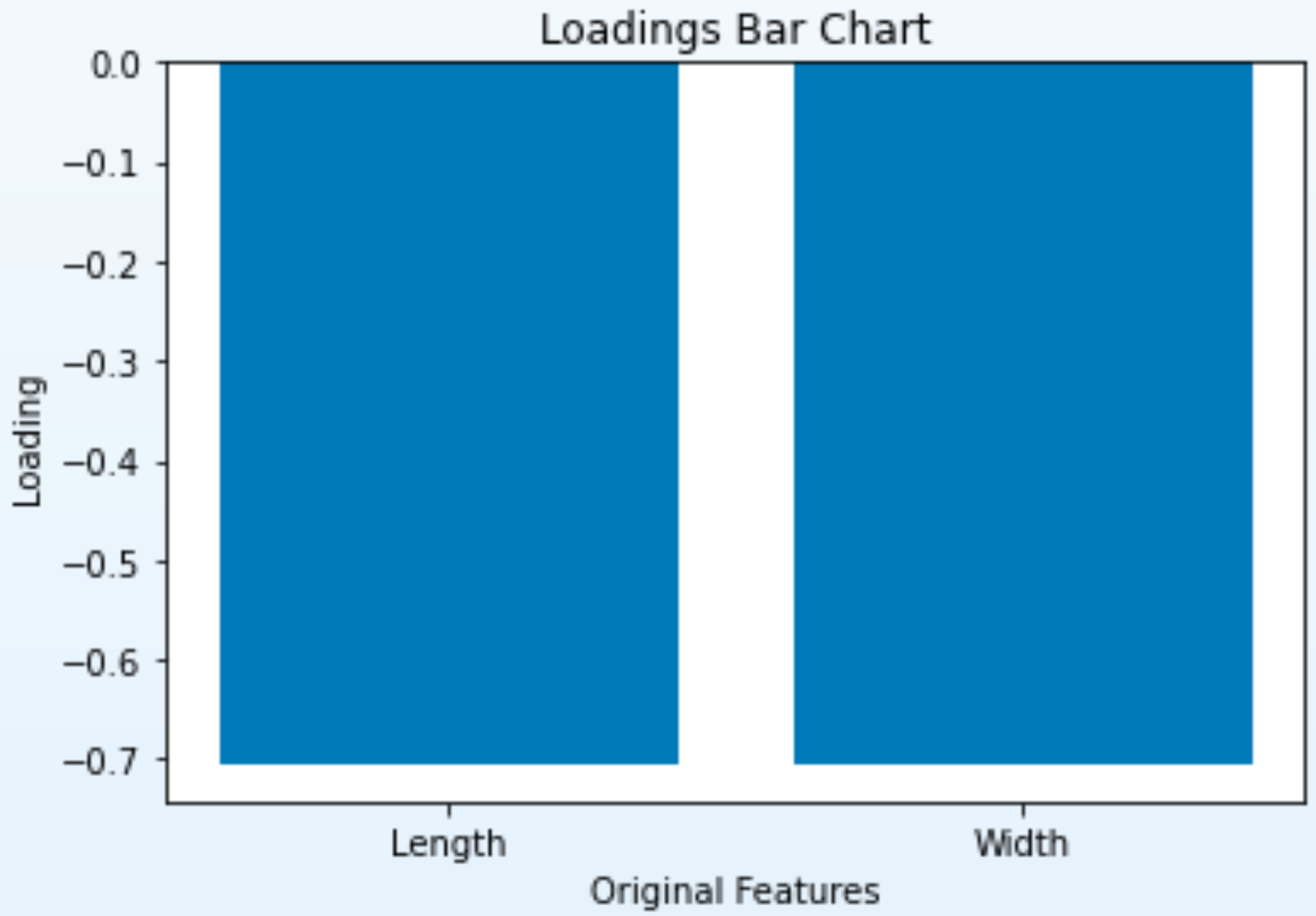
```
0.9999999999999999
```

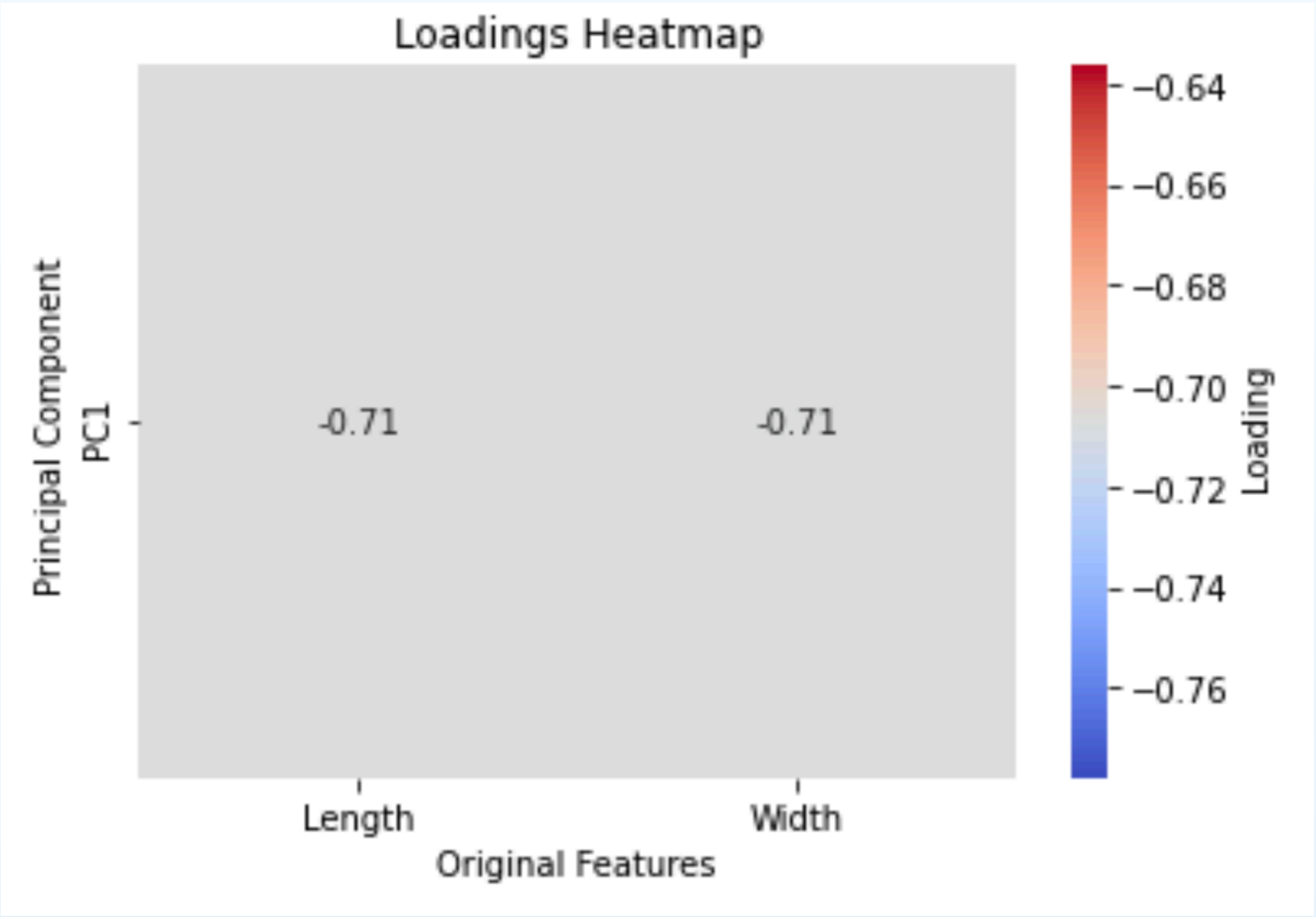
Loadings are always
A unit vectors

PCA Example 1

Visualize the Loadings

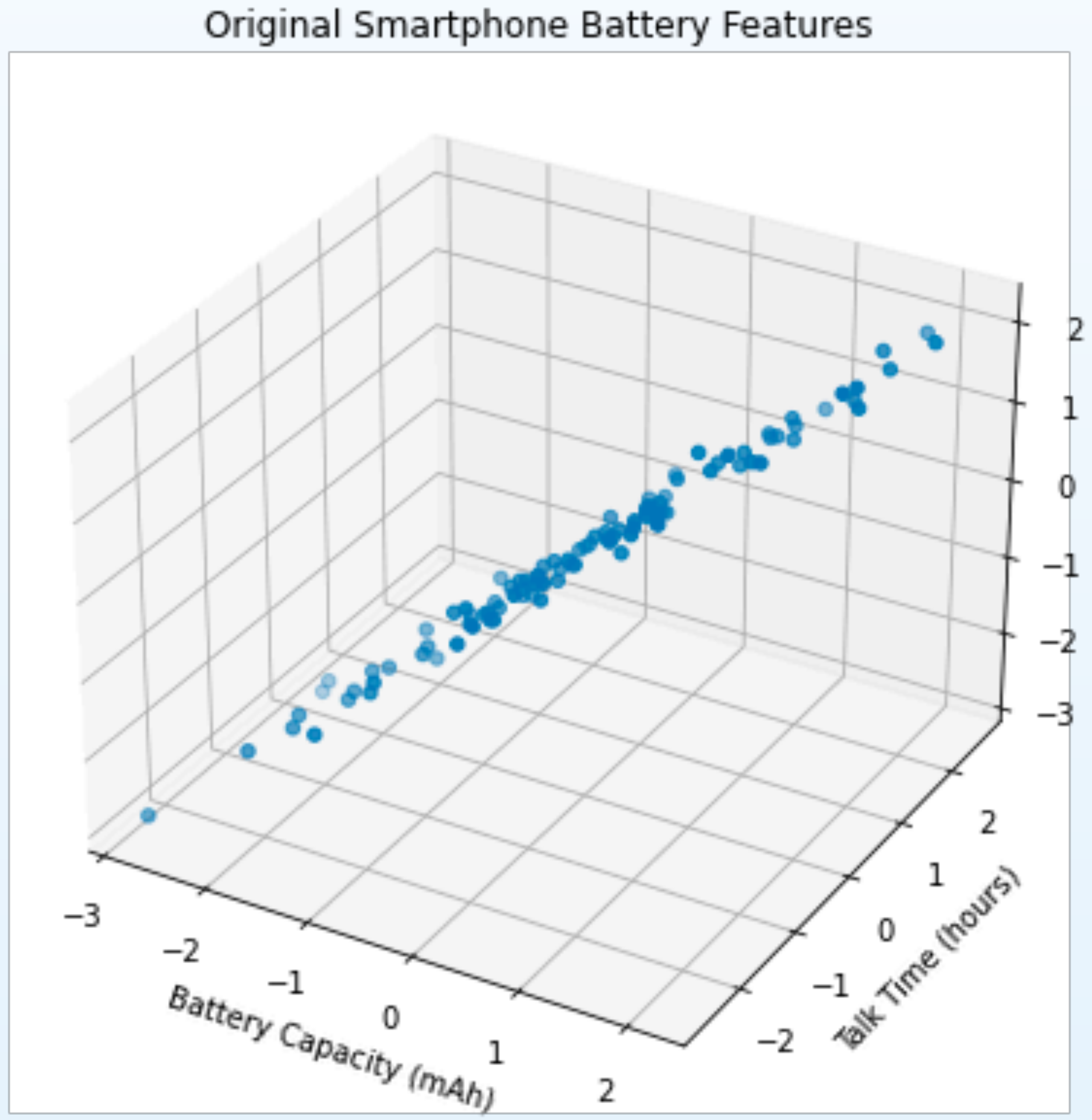
1.) Bar Plot
2.) Heat Map

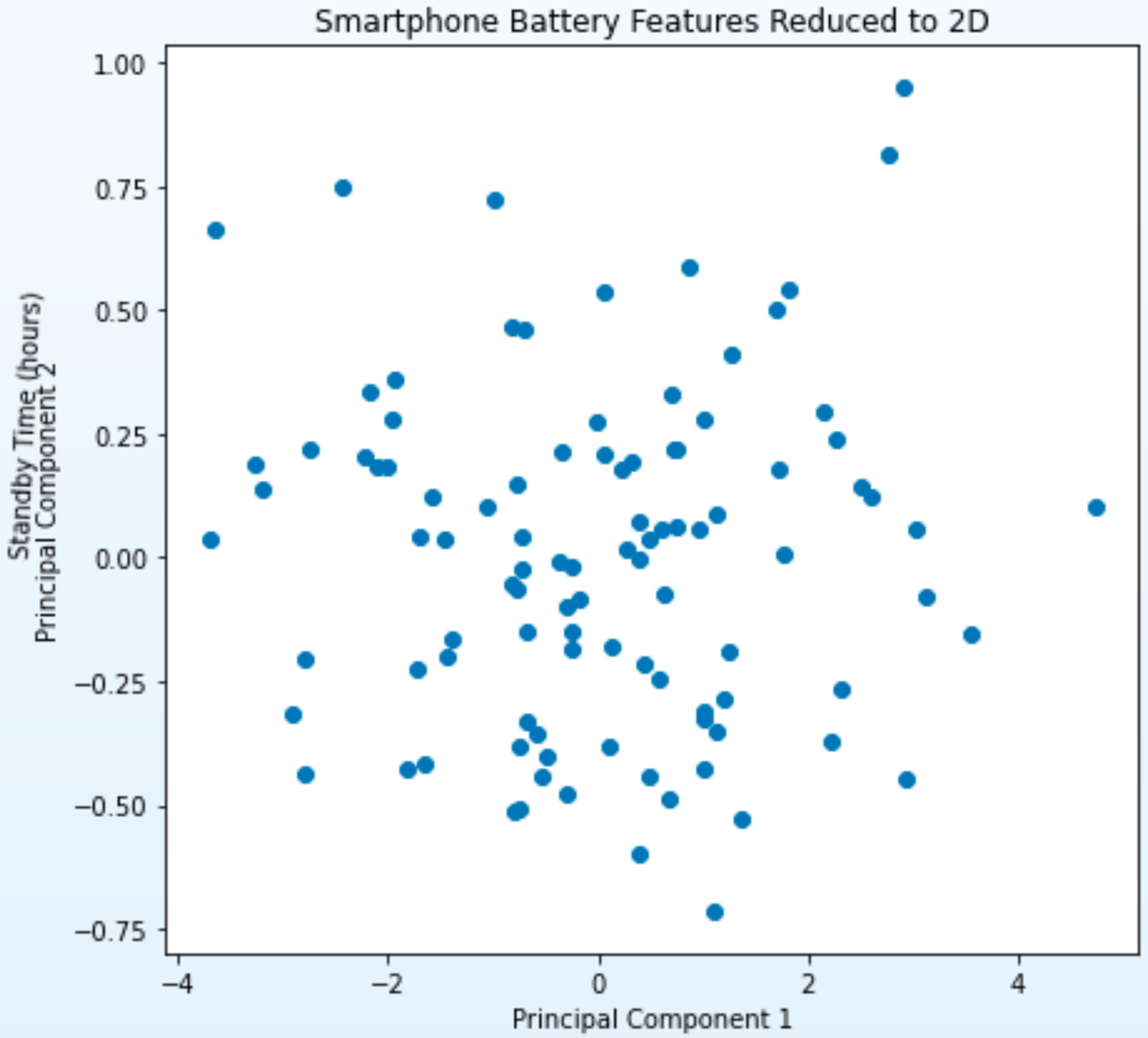




PCA Example 2

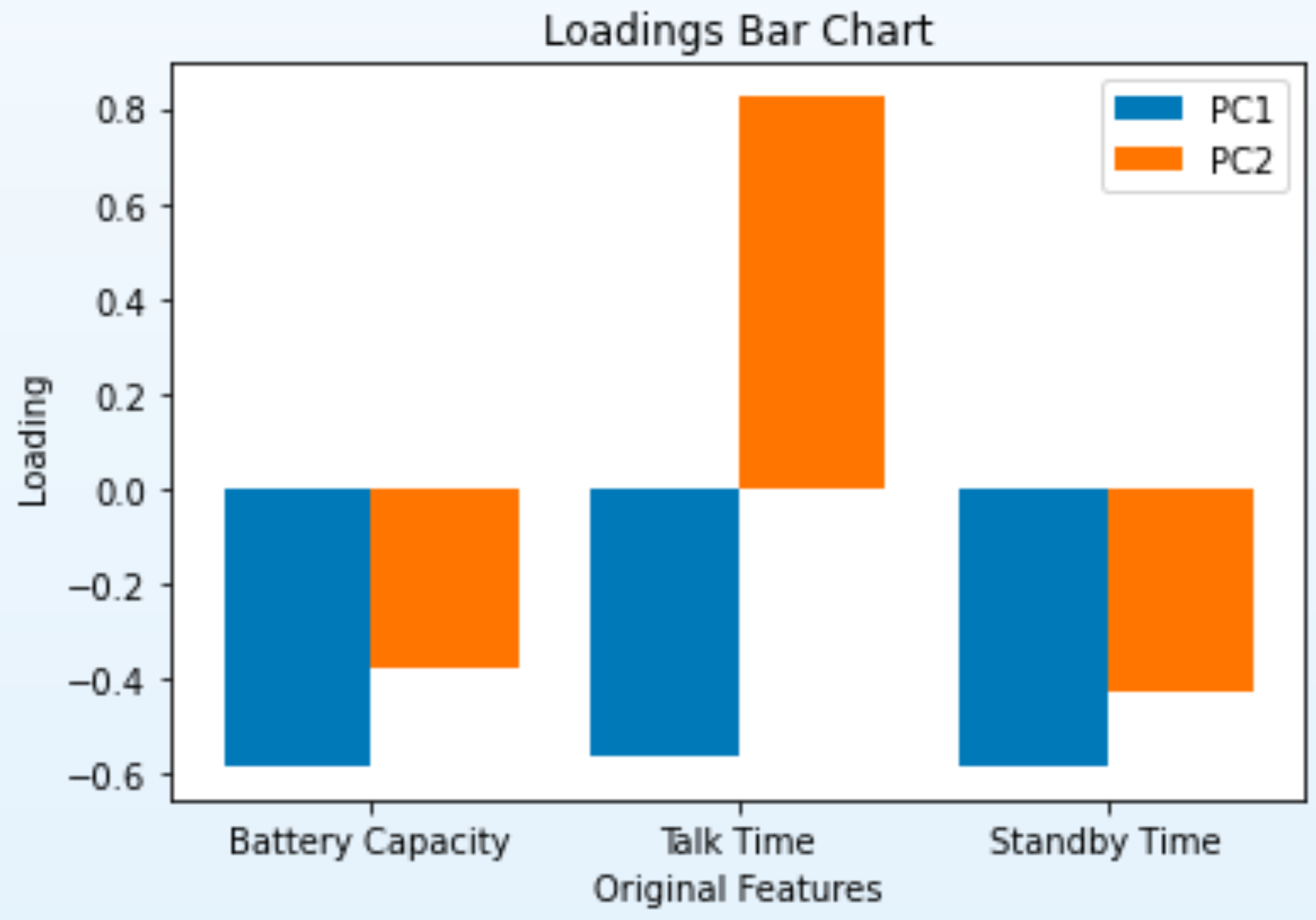
1 single Principle component is created can contain most of the information of two dimensions

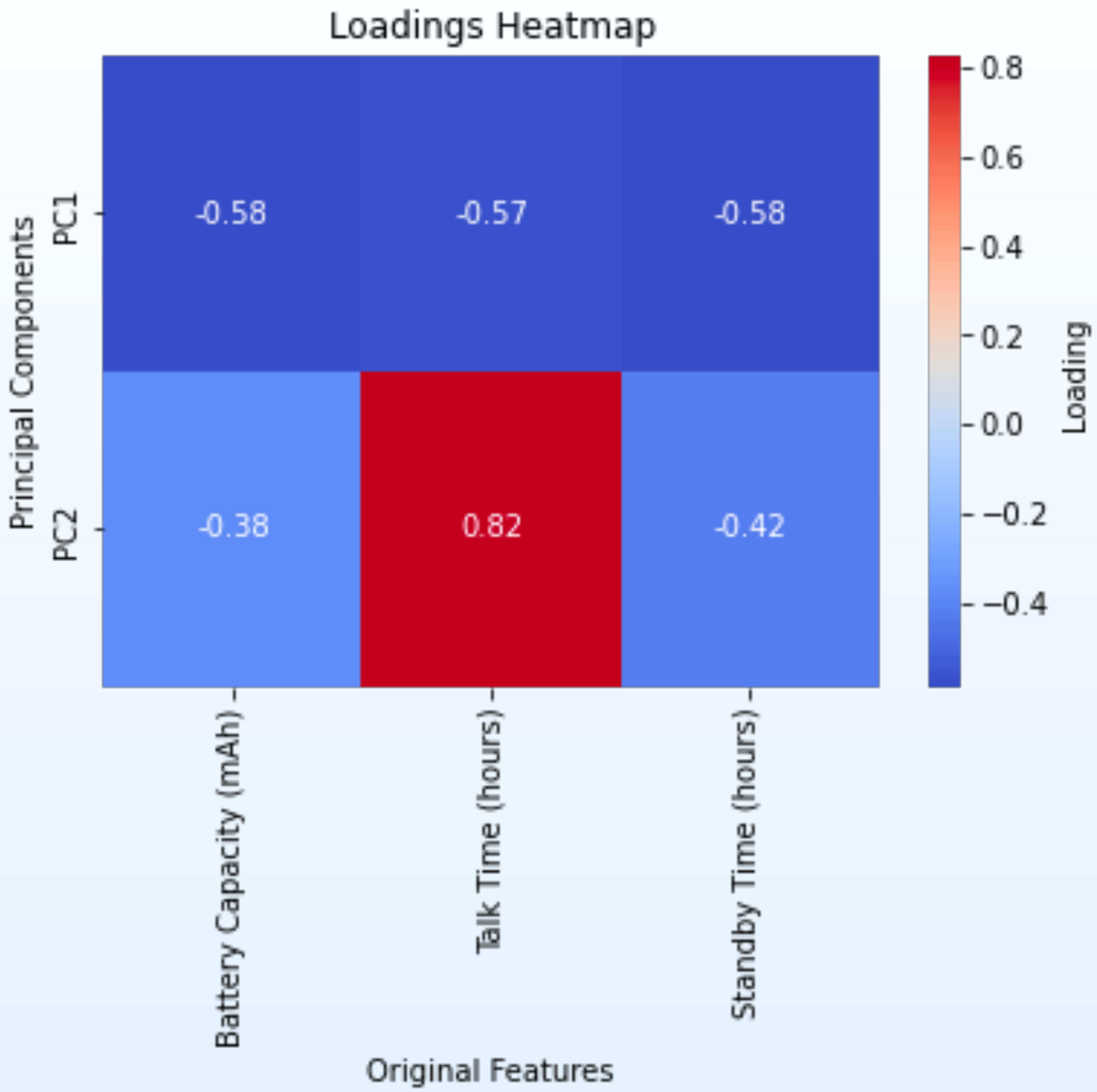




PCA Example 2

1 single Principle component is created can contain most of the information of two dimensions





Feature Importance Calculation

1 single Principle component is created can contain most of the information of two dimensions

$$f(x_1) = a_{1,1}^2 + \ldots + a_{1,m}^2$$

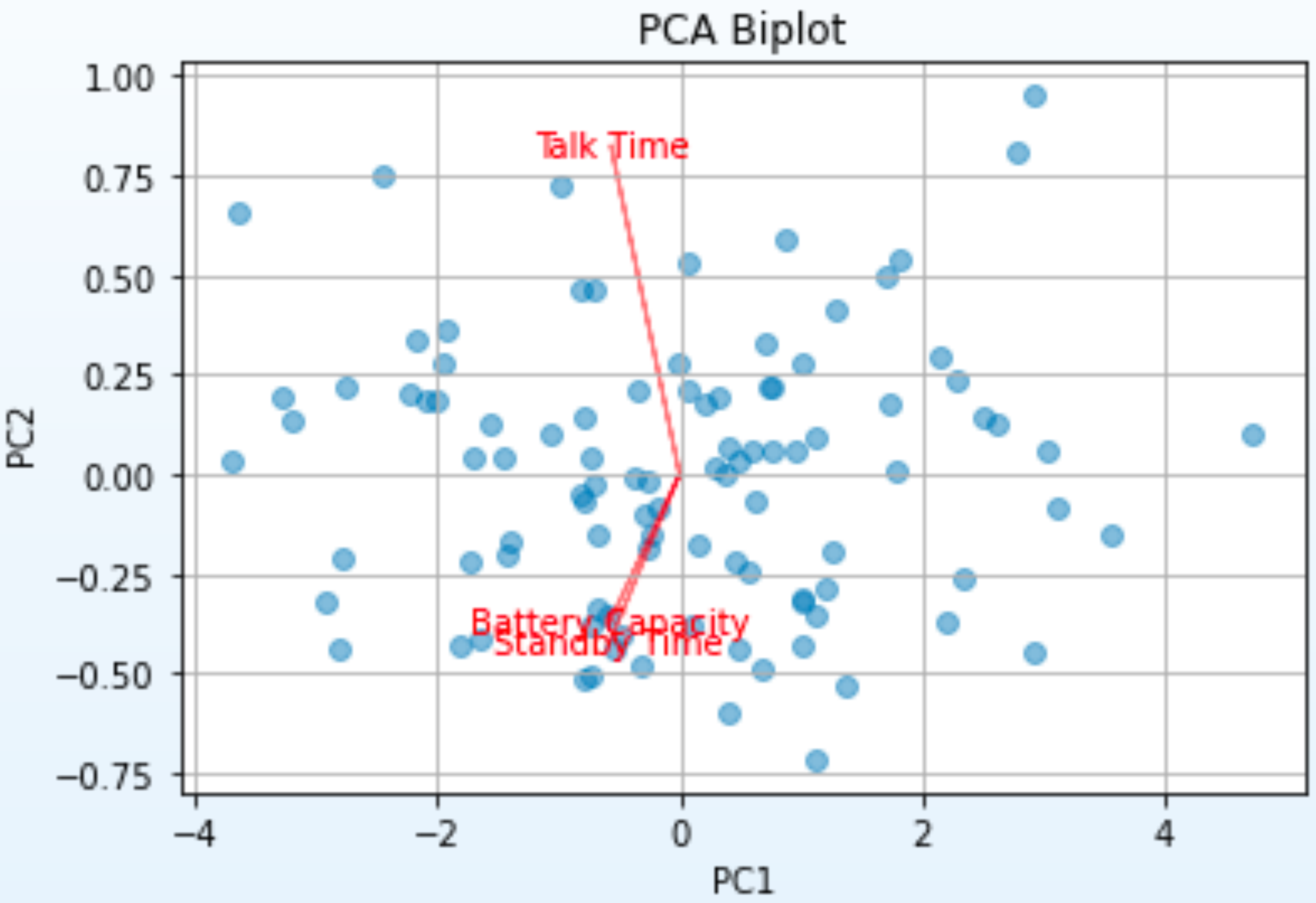
$f(\cdot)$: Feature importance of an input variable

```
Battery Capacity: 0.48
Talk Time: 1.00
Standby Time: 0.52
```

Biplot

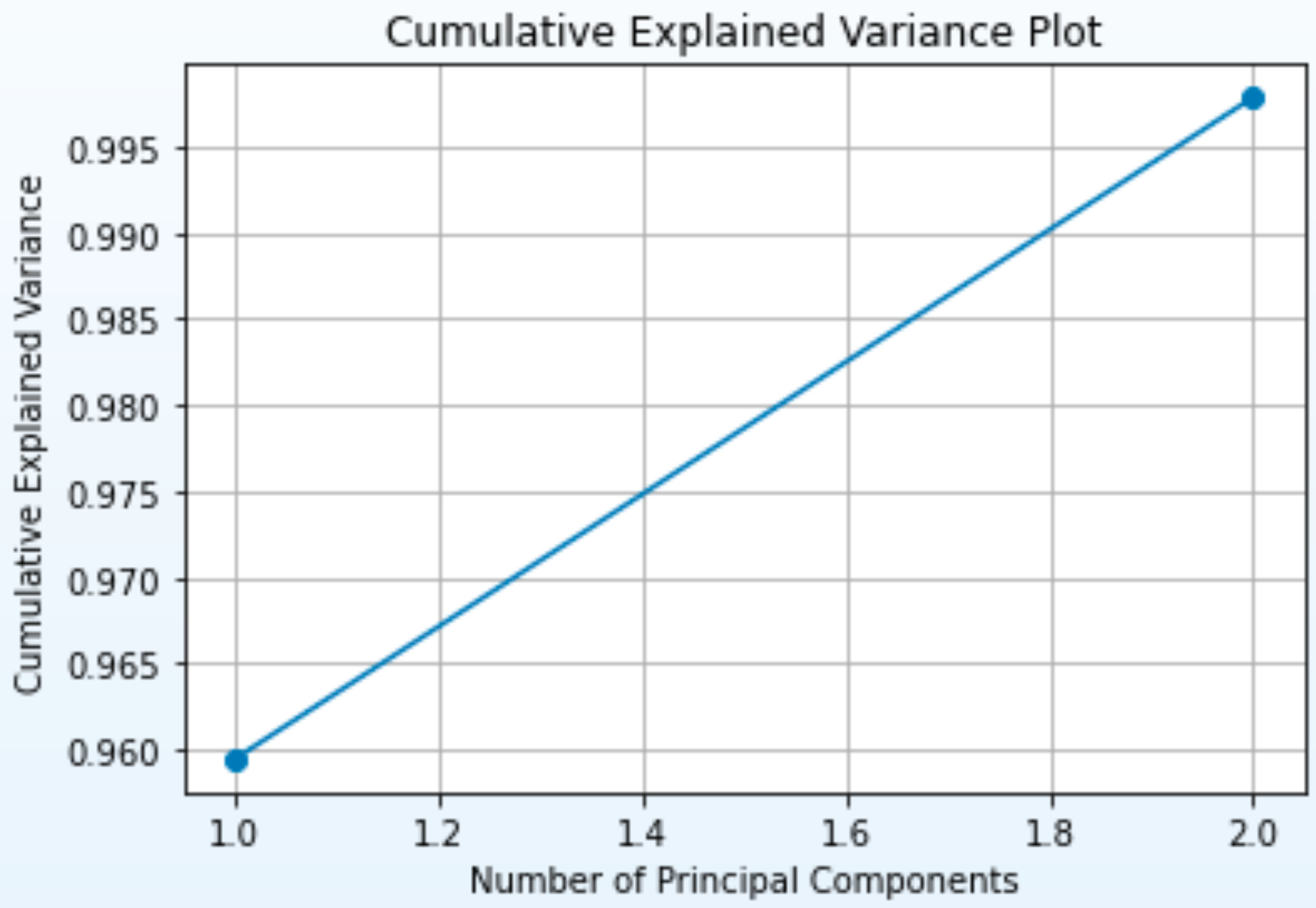
Graph of Multivariate data that has information on
1.) The transformed data in the reduced dimension space
2.) Loadings (contribution of the original variable to the new components

Does this by :
Data projected on the new PCs
And Vector representing the original features



Cumulative Explained Variance Plot

For each additional PC, we explain some more variance of the data



Coding

Coding

Import packages and scale

Step 1: Packages

```
import seaborn as sns
from sklearn.decomposition import PCA
```

Step 2: Scale your data

```
scaler = StandardScaler().fit(X)
X_scaled = scaler.transform(X)
```

Coding

To Begin, you trying a Machine Learning Model the same as we've done

Step 3: Get Loadings
And other metrics

```
loadings = pca.components_
pca.explained_variance_ratio_
```

Step 4: Visualize and interpret

```
sns.heatmap(loadings,
```

In - Class Assignment

Data : Same as Week 9

Country-data.csv

country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
Afghanistan	90.2	10	7.58	44.9	1610	9.44	56.2	5.82	553
Albania	16.6	28	6.55	48.6	9930	4.49	76.3	1.65	4090
Algeria	27.3	38.4	4.17	31.4	12900	16.1	76.5	2.89	4460
Angola	119	62.3	2.85	42.9	5900	22.4	60.1	6.16	3530
Antigua and Barbud	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200
Argentina	14.5	18.9	8.1	16	18700	20.9	75.8	2.37	10300
Armenia	18.1	20.8	4.4	45.3	6700	7.77	73.3	1.69	3220
Australia	4.8	19.8	8.73	20.9	41400	1.16	82	1.93	51900
Austria	4.3	51.3	11	47.8	43200	0.873	80.5	1.44	46900
Azerbaijan	39.2	54.3	5.88	20.7	16000	13.8	69.1	1.92	5840
Bahamas	13.8	35	7.89	43.7	22900	-0.393	73.8	1.86	28000
Bahrain	8.6	69.5	4.97	50.9	41100	7.44	76	2.16	20700
Bangladesh	49.4	16	3.52	21.8	2440	7.14	70.4	2.33	758
Barbados	14.2	39.5	7.97	48.7	15300	0.321	76.7	1.78	16000
Belarus	5.5	51.4	5.61	64.5	16200	15.1	70.4	1.49	6030

0.) Import and Clean Data
1.) Run a PCA Algorithm to get 2 Principle Components for the 9 X features
2.) Plot a Score Plot
3.) Rank the features in order of importance according to PCA
4.) Plot a heat map of the feature importance
5.) Plot a correlation plot of the original features. What do you notice between graphs of 4 & 5
6.) Run a PCA with 9 PCs. Plot the Cumulative Explained Variance Plot. How many PCs should we use if we want to retain 95% of the variance?