Principal Component Analysis

Pulsar Stars

Marina Tihova
mtt862@student.bham.ac.uk

School of Computer Science
Computer Science BSc
University of Birmingham
March, 2020

Contents

1 What is PCA? 2

2 Dataset 3

2.1 Introduction 3

2.2 What are pulsar stars? 3

2.3 Properties 4

2.4 Applications 4

2.5 Attributes 5

2.5.1 Integrated pulse profile (folded profile) 5

2.5.2 DM-SNR curve 5

2.5.3 Target class 5

3 Expectations 6

4 Software 6

5 Pre-processing of the data 7

6 Insights 10

7 Calculations 11

7.1 Covariance Matrix 11

7.2 Eigenvalues and Eigenvectors 11

7.3 Final Projections 13

7.4 Explaining the projections 14

8 Conclusion 14

1 What is PCA?

In the real world, datasets may contain hundreds of attributes and naturally, some attributes will be more important than others. PCA aims to help with simplifying high-dimensional data by reducing the number of dimensions while preserving the essence of the data.

In order to reduce the dimensionality of the feature space PCA introduces new variables (principal components) that are orthogonal to each other. A principal component is a linear combination of the original predictor variables which captures the maximum variance in the data set. A downside of the resulting principal components, being a complex mixture of the original features, is that interpreting them can prove difficult.

Usually PCA is used before applying machine learning algorithms. When the input dimension is too high, PCA can speed up the process. It can also combat overfitting by filtering the irrelevant attributes which decrease the performance of some algorithms. Another common use for PCA is data visualisation since understanding two- or three-dimensional data is not as challenging as higher dimensional data.

2 Dataset

2.1 Introduction

I decided to use the HTRU2 dataset with the goal of predicting a pulsar star. HTRU2 is a data set which describes a sample of pulsar candidates collected during the High Time Resolution Universe Survey.[1]

The dataset contains a “target class” attribute explaining which samples are pulsar stars and which are not. In practice, almost all detections are caused by radio frequency interference (RFI) and noise, making legitimate signals hard to find. HTRU2 contains 16,259 spurious examples caused by RFI/noise, and 1,639 real pulsar examples. I’ll be plotting the “target” attribute against the others in order to find pulsar stars.[2]

2.2 What are pulsar stars?

Pulsars are a rare type of neutron star meaning they are exceptionally small and dense. Their curious properties make them a significant point of interest for a plethora of scientists.

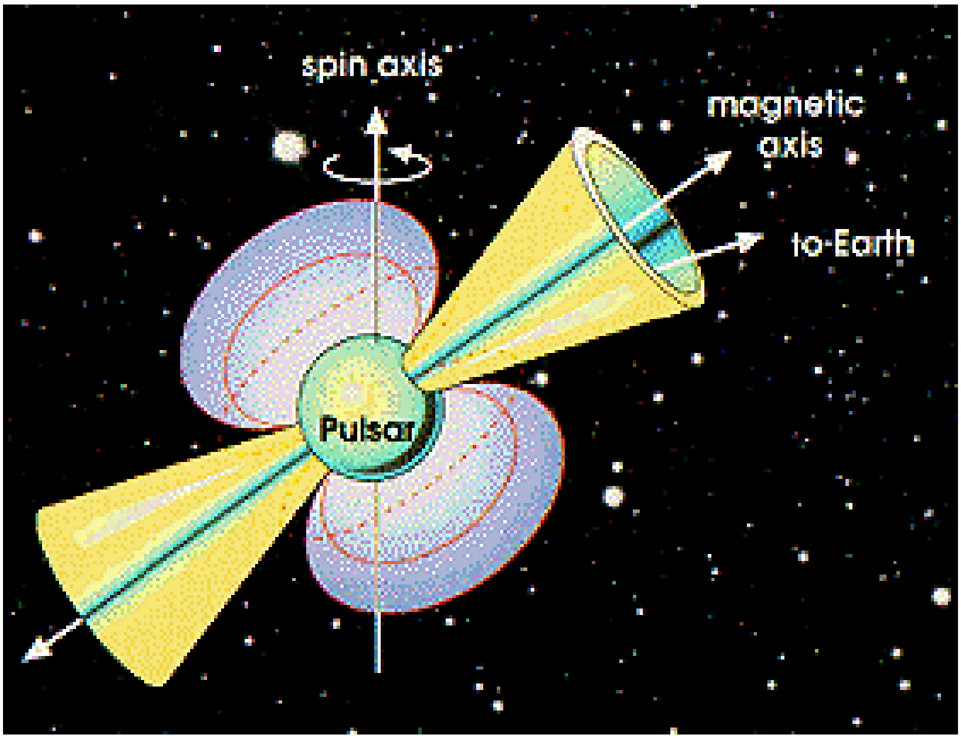


Figure 1: Pulsar star *(Image credit: NASA)*

2.3 Properties

Pulsars radiate two steady, narrow beams of light in opposite directions (see Figure 1). Although the light from the beam is steady, pulsars appear to flicker because they also spin. It's often compared to when a lighthouse appears to blink when seen by a sailor on the ocean: as the pulsar rotates, the beam of light may sweep across the Earth, then swing out of view, then swing back around again.[3]

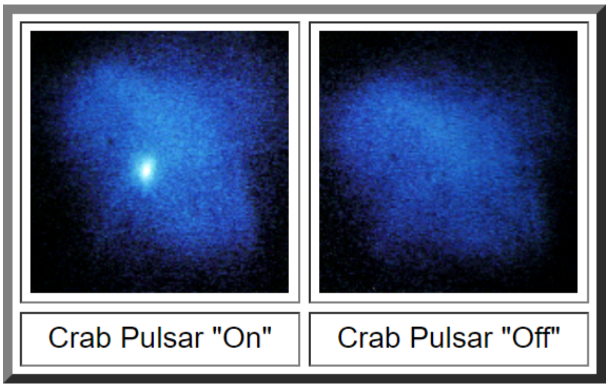


Figure 2: *(Image credit: Einstein X-ray observatory)*

Pulsars are highly magnetic. Their magnetic fields range from 100 million times to 1 quadrillion times stronger than Earth's.[4]

2.4 Applications

Pulsars are fantastic cosmic tools for scientists to study a wide range of phenomena.

Many pulsars blink with extreme precision. Pulsars can be used as precise clocks since they are considered the most accurate natural clocks in the universe.

Because pulsars are moving through space while also blinking a regular number of times per second, scientists can use many pulsars to calculate cosmic distances. Thanks to the exquisite timing of the pulses, scientists have made some of the most accurate distance measurements of cosmic objects.

The physics of neutron stars can be researched by analysing the light emitted by a pulsar. Because of their density, matter around them behaves in new unexpected ways.

Pulsars have also been used to test aspects of Albert Einstein's theory of general relativity, such as the universal force of gravity.[3]

2.5 Attributes

2.5.1 Integrated pulse profile (folded profile)

Pulsars produce periodic pulsation signals which are often too weak to detect. Integrated pulsar signals increase the signal quality and form its integrated pulse profile. Pulsars are distinguished by their *integrated pulse profiles* – all pulsars have their unique profile shape, similar to human's fingerprints. [5]
The HTRU2 dataset includes:

1. Mean of the integrated profile
2. Standard deviation of the integrated profile
3. Excess kurtosis of the integrated profile
4. Skewness of the integrated profile

2.5.2 DM-SNR curve

When observing a pulsar over a finite bandwidth, a broadening of an otherwise sharp pulse can be seen. This is the *dispersion measure (DM)*. DM is a constant defined by the column density of free electrons along the line of sight. The DM-SNR curve describes the dispersion measure given the signal to noise ratio of the integrated profile.[6]
The HTRU2 dataset includes:

5. Mean of the DM-SNR curve
6. Standard deviation of the DM-SNR curve
7. Excess kurtosis of the DM-SNR curve
8. Skewness of the DM-SNR curve

2.5.3 Target class

The HTRU2 dataset includes:

9. Target class
    - Label 1: indicating the sample is a real pulsar
    - Label 0: indicating the sample is spurious

3 Expectations

I expect to discover that not all dataset features are as important to predicting a pulsar star. By applying PCA to this dataset, hopefully, the results will show that in practice the star data is less dimensional and more reliant on certain features than others.

When plotting the data over the first few principal components I expect to see clear separation between the spurious and pulsar star samples. This implies that if I plugged a new sample in and plotted it on the same projection, I'd be able to see whether it has features making it more prone to be either a real pulsar or just a frequency noise.

4 Software

I decided to use Python for conducting PCA. After researching various approaches for PCA, such as *Python*, *MATLAB*, or the language *R*, and writing sample code in them, I found Python to be the most convenient and versatile approach for me. Python contains tools for handling huge matrices, performing large computations and is often used for machine learning purposes.

For this project I used *IPython* alongside *Jupyter Notebook* for applying PCA, plotting projections, and documenting the process using markdown language. The libraries I relied on were:

- **pandas** for data manipulation. Pandas supplied me with an appealing way to visualise two dimensional matrices and to modify a dataset in terms of a table with an index and columns.
- **NumPy** for the large collection of high-level mathematical functions they provide. Pandas works closely with NumPy for handling multi-dimensional arrays.
- **scikit-learn** because it features various classification, regression and clustering algorithms including PCA.
- **Matplotlib** to plot the results.

Furthermore, Python was the best choice for me since I've extensively used pandas and NumPy in my experience prior university. All the code I wrote for this project can be found in my GitHub repository.

5 Pre-processing of the data

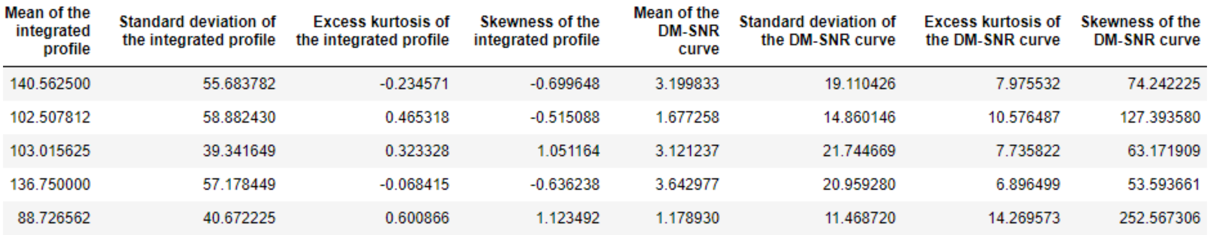
Before my final dataset choice, I tried a few more datasets which had a lot of categorical data columns. After converting to numerical values, I discovered that said datasets were complex and too dependent of all features thus making them not suitable for finding interesting results via PCA. Excluding the target class, the pulsar dataset has only numerical values and conversion was not needed in this case.

Before starting PCA, I removed the target class column from the dataset table so it wouldn't throw off the data.

After removing the class, I normalised the data.

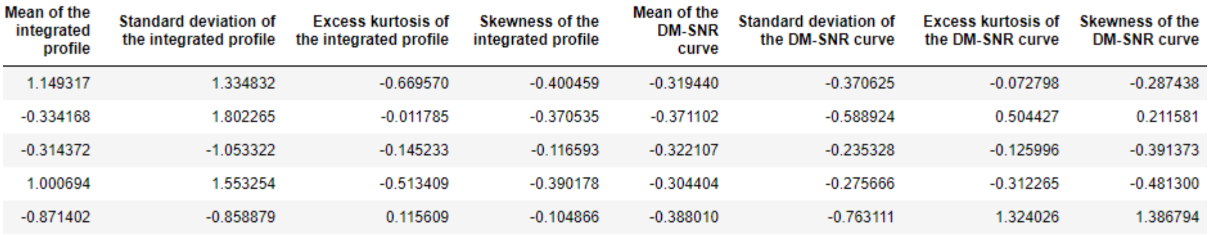
$$X_{scaled} = \frac{X - m}{\sqrt{Variance}}$$

The values before and after scaling for the first 5 samples are as follows:



Mean of the integrated profile	Standard deviation of the integrated profile	Excess kurtosis of the integrated profile	Skewness of the integrated profile	Mean of the DM-SNR curve	Standard deviation of the DM-SNR curve	Excess kurtosis of the DM-SNR curve	Skewness of the DM-SNR curve
140.562500	55.683782	-0.234571	-0.699648	3.199833	19.110426	7.975532	74.242225
102.507812	58.882430	0.465318	-0.515088	1.677258	14.860146	10.576487	127.393580
103.015625	39.341649	0.323328	1.051164	3.121237	21.744669	7.735822	63.171909
136.750000	57.178449	-0.068415	-0.636238	3.642977	20.959280	6.896499	53.593661
88.726562	40.672225	0.600866	1.123492	1.178930	11.468720	14.269573	252.567306

Figure 3: Data before scaling



Mean of the integrated profile	Standard deviation of the integrated profile	Excess kurtosis of the integrated profile	Skewness of the integrated profile	Mean of the DM-SNR curve	Standard deviation of the DM-SNR curve	Excess kurtosis of the DM-SNR curve	Skewness of the DM-SNR curve
1.149317	1.334832	-0.669570	-0.400459	-0.319440	-0.370625	-0.072798	-0.287438
-0.334168	1.802265	-0.011785	-0.370535	-0.371102	-0.588924	0.504427	0.211581
-0.314372	-1.053322	-0.145233	-0.116593	-0.322107	-0.235328	-0.125996	-0.391373
1.000694	1.553254	-0.513409	-0.390178	-0.304404	-0.275666	-0.312265	-0.481300
-0.871402	-0.858879	0.115609	-0.104866	-0.388010	-0.763111	1.324026	1.386794

Figure 4: Data after scaling

The next 6 figures display information for the mean, variance and covariance before and after normalisation respectively.

Mean of the integrated profile	111.07996834492681
Standard deviation of the integrated profile	46.549531561534295
Excess kurtosis of the integrated profile	0.4778572581019128
Skewness of the integrated profile	1.7702789980713511
Mean of the DM-SNR curve	12.614399658311525
Standard deviation of the DM-SNR curve	26.326514703918694
Excess kurtosis of the DM-SNR curve	8.303556116638275
Skewness of the DM-SNR curve	104.85770870366196

Figure 5: Mean before normalisation

Mean of the integrated profile	−0
Standard deviation of the integrated profile	−0
Excess kurtosis of the integrated profile	0
Skewness of the integrated profile	0
Mean of the DM-SNR curve	−0
Standard deviation of the DM-SNR curve	0
Excess kurtosis of the DM-SNR curve	0
Skewness of the DM-SNR curve	0

Figure 6: Mean after normalisation (to 10 decimal places)

Mean of the integrated profile	658.0363246090876
Standard deviation of the integrated profile	46.82662485126457
Excess kurtosis of the integrated profile	1.1321172606433363
Skewness of the integrated profile	38.04102827819613
Mean of the DM-SNR curve	868.603132969526
Standard deviation of the DM-SNR curve	379.08200556314085
Excess kurtosis of the DM-SNR curve	20.30372936498847
Skewness of the DM-SNR curve	11344.713243182858

Figure 7: Variance before normalisation

Mean of the integrated profile	1
Standard deviation of the integrated profile	1
Excess kurtosis of the integrated profile	1
Skewness of the integrated profile	1
Mean of the DM-SNR curve	1
Standard deviation of the DM-SNR curve	1
Excess kurtosis of the DM-SNR curve	1
Skewness of the DM-SNR curve	1

Figure 8: Variance after normalisation

$$\begin{bmatrix} 658.07 & 96.05 & -23.85 & -116.89 & -225.94 & -153.35 & 27.09 & 393.56 \\ 96.05 & 46.83 & -3.8 & -22.78 & 1.39 & -6.35 & 0.91 & 20.18 \\ -23.85 & -3.8 & 1.13 & 6.21 & 12.99 & 8.97 & -1.64 & -24.31 \\ -116.89 & -22.78 & 6.21 & 38.04 & 74.91 & 49.86 & -9.14 & -134.54 \\ -225.94 & 1.39 & 12.99 & 74.91 & 868.65 & 457.11 & -81.81 & -1112.15 \\ -153.35 & -6.35 & 8.97 & 49.86 & 457.11 & 379.1 & -71.05 & -1194.15 \\ 27.09 & 0.91 & -1.64 & -9.14 & -81.81 & -71.05 & 20.3 & 443.36 \\ 393.56 & 20.18 & -24.31 & -134.54 & -1112.15 & -1194.15 & 443.36 & 11345.35 \end{bmatrix}$$

Figure 9: Covariance Matrix before normalisation (to 2 decimal places)

$$\begin{bmatrix} 1.0 & 0.55 & -0.87 & -0.74 & -0.3 & -0.31 & 0.23 & 0.14 \\ 0.55 & 1.0 & -0.52 & -0.54 & 0.01 & -0.05 & 0.03 & 0.03 \\ -0.87 & -0.52 & 1.0 & 0.95 & 0.41 & 0.43 & -0.34 & -0.21 \\ -0.74 & -0.54 & 0.95 & 1.0 & 0.41 & 0.42 & -0.33 & -0.2 \\ -0.3 & 0.01 & 0.41 & 0.41 & 1.0 & 0.8 & -0.62 & -0.35 \\ -0.31 & -0.05 & 0.43 & 0.42 & 0.8 & 1.0 & -0.81 & -0.58 \\ 0.23 & 0.03 & -0.34 & -0.33 & -0.62 & -0.81 & 1.0 & 0.92 \\ 0.14 & 0.03 & -0.21 & -0.2 & -0.35 & -0.58 & 0.92 & 1.0 \end{bmatrix}$$

Figure 10: Covariance Matrix after normalisation (to 2 decimal places)

6 Insights

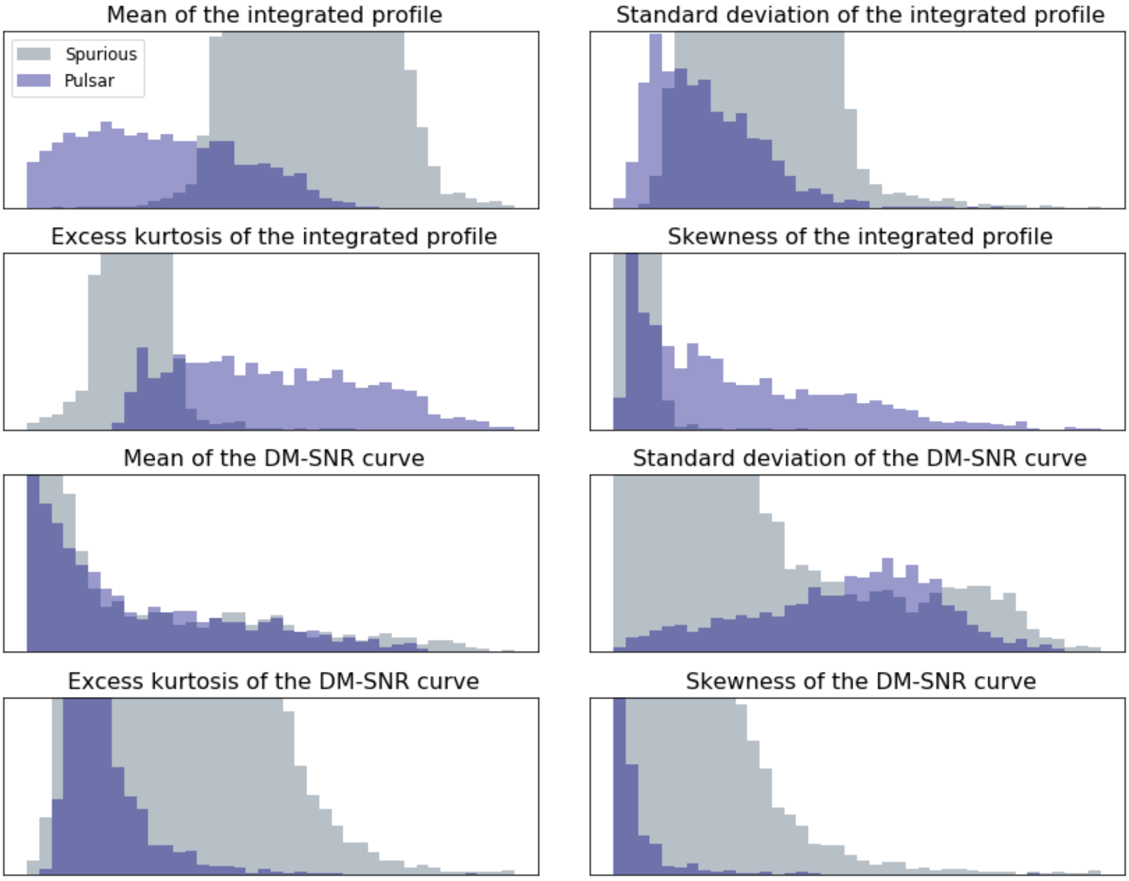


Figure 11: Target histograms

Figure 11 shows the results of plotting projections of the data onto pairs of its existing dimensions. Since the number of spurious examples in the dataset is significantly larger than the true stars, I had to limit the y axis so the overlap can be shown more clearly in the plots.

The projections reveal that the **DM-SNR curve** values don't hold much essential information due to the targets mostly overlapping in said plots. Both targets describe similar values for their DM-SNR curves therefore using only the DM-SNR curve data won't be enough to distinguish noise from a real pulsar star using PCA.

Observing the **integrated profile** plots shows us that there's a difference in the two targets. The *skewness* and *excess kurtosis* of the integrated profile for the pulsar stars seem to be generally higher than the spurious stars. The *mean* and the *standard deviation* on the other hand usually take on lower values for the pulsars. The integrated profile features will prove more crucial and will contribute more when applying PCA. This shows us that the data is potentially less dimensional than it appears to be.

7 Calculations

7.1 Covariance Matrix

$$\begin{bmatrix}
4.13428 & 0.0 & -0.0 & 0.0 & -0.0 & -0.0 & 0.0 & -0.0 \\
0.0 & 2.14472 & 0.0 & 0.0 & -0.0 & 0.0 & -0.0 & -0.0 \\
-0.0 & 0.0 & 0.80939 & -0.0 & 0.0 & 0.0 & -0.0 & -0.0 \\
0.0 & 0.0 & -0.0 & 0.45745 & 0.0 & 0.0 & 0.0 & 0.0 \\
-0.0 & -0.0 & 0.0 & 0.0 & 0.25824 & 0.0 & 0.0 & 0.0 \\
-0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.15989 & -0.0 & 0.0 \\
0.0 & -0.0 & -0.0 & 0.0 & 0.0 & -0.0 & 0.02044 & 0.0 \\
-0.0 & -0.0 & -0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.01603
\end{bmatrix}$$

Figure 12: Covariance Matrix after PCA (to 5 decimal places)

7.2 Eigenvalues and Eigenvectors

$$\begin{array}{c}
4.134277694661789 \\
2.144724928035271 \\
0.8093941863429204 \\
0.4574503833736197 \\
0.25824134528449544 \\
0.1598882366882819 \\
0.020443080572961073 \\
0.016027147331548922
\end{array}$$

Figure 13: Eigenvalues

$$\begin{bmatrix}
0.36 & 0.36 & 0.01 & 0.29 & 0.74 & 0.02 & 0.31 & -0.03 \\
0.21 & 0.43 & -0.44 & -0.76 & 0.02 & 0.05 & -0.06 & -0.0 \\
-0.42 & -0.32 & -0.09 & -0.3 & 0.15 & 0.05 & 0.77 & -0.06 \\
-0.4 & -0.31 & -0.08 & -0.18 & 0.64 & -0.02 & -0.54 & 0.03 \\
-0.35 & 0.25 & -0.57 & 0.33 & -0.07 & -0.61 & 0.03 & -0.09 \\
-0.39 & 0.32 & -0.23 & 0.26 & -0.07 & 0.75 & -0.06 & -0.24 \\
0.37 & -0.41 & -0.27 & 0.04 & 0.0 & 0.02 & -0.05 & -0.78 \\
0.29 & -0.4 & -0.58 & 0.2 & -0.02 & 0.25 & 0.03 & 0.56
\end{bmatrix}$$

Figure 14: Eigenvectors (to 2 decimal places)

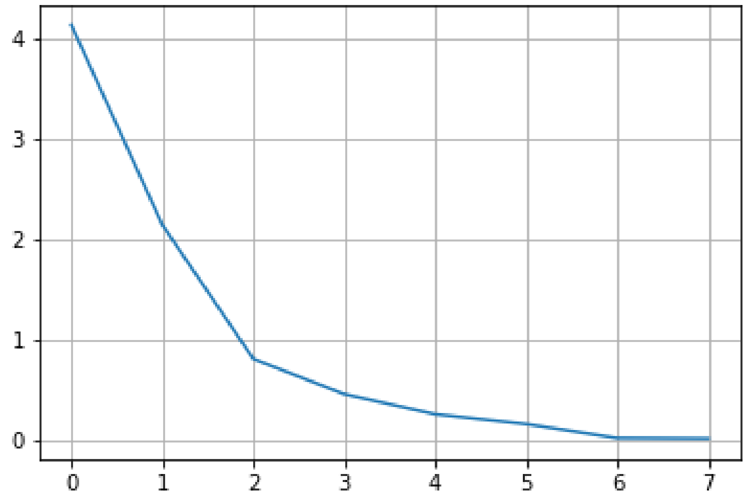


Figure 15: Eigenvalues plotted

The eigenvalues plot on Figure 15 clearly shows that the first 3 eigenvalues explain most of the data. After calculating the explained variance ratio, we can see that the first one, two, and three principal components reveal 52%, 79%, and 89% of the data respectively (see Figure 16). Thus, by using the first three components for the final projection, the plot would have access to nearly 90% of the information.

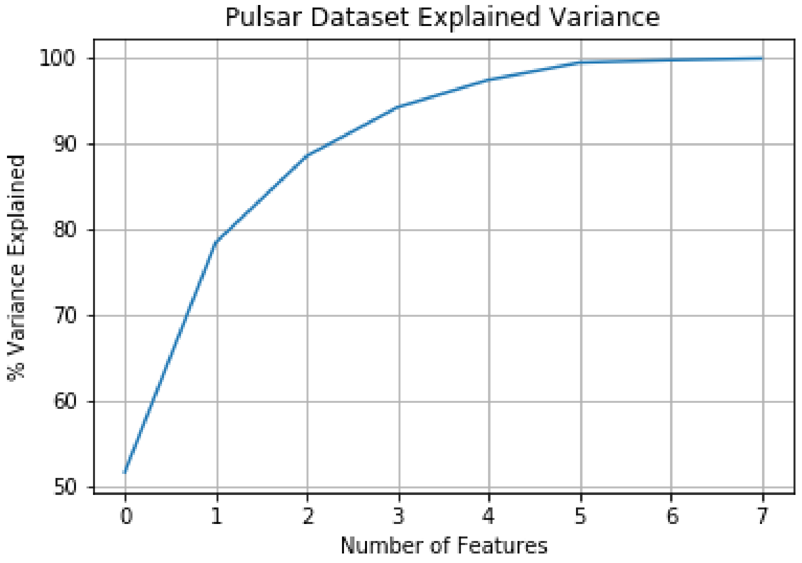


Figure 16: % explained variance

7.3 Final Projections

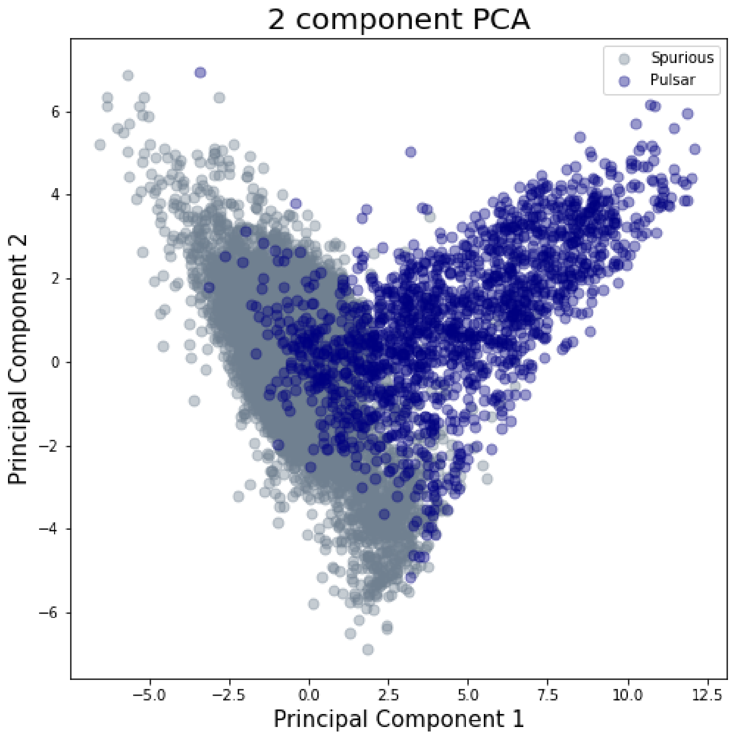


Figure 17: 2-component projection

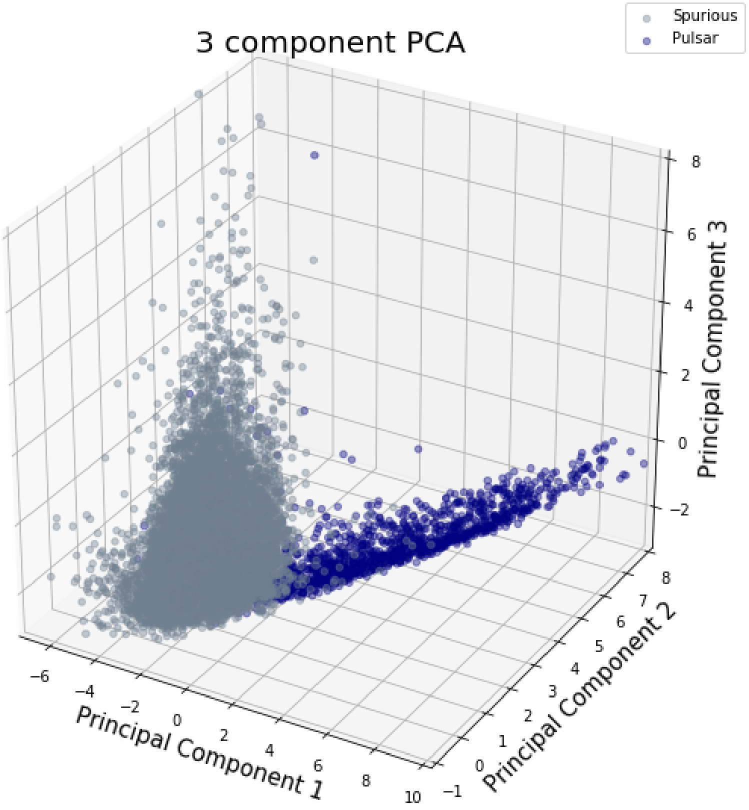


Figure 18: 3-component projection

7.4 Explaining the projections

Plotting onto the first two components shows a separation between spurious and pulsar stars while still having some ambiguous points on the overlap in between the two target classes. Adding the third component further lowers the ambiguity and gives us a clear idea of identifying a pulsar. This tells us that the initial 8-dimensional data, as a matter of fact, only needs three dimensions to accurately display a pulsar candidate.

8 Conclusion

PCA can essentially reveal the type of dataset we have – whether all dimensions are impactful to the final result or if the data is, in fact, simpler than what we're initially presented with. If a dataset is too complex, a further algorithm can be applied to the already processed data in order to reach the desired output. PCA is a powerful tool that can simplify complex data and help us visualise it effectively.

In conclusion, PCA proved successful in reducing the dimensionality of my dataset, confirming my prior predictions. The final projections show a clear separation and can help distinguish a pulsar star when a new sample is compared with the data.

References

[1] M. J. Keith et al. *The High Time Resolution Universe Pulsar Survey - I. System Configuration and Initial Discoveries.* Monthly Notices of the Royal Astronomical Society, 2010.

[2] Dr Robert Lyon. Uci machine learning repository. *HTRU2 Data Set*, 2017.

[3] Calla Cofield. What are pulsars? *Uses of pulsars*, 2016.

[4] Wikipedia. Pulsar. *History of observation*, 2020.

[5] Kuo Liu. Introduction to pulsar, pulsar timing, and measuring of pulse time-of-arrivals. *Integrated pulse profile: pulsar's fingerprint*, 2017.

[6] Swinburne University of Technology. Pulsar dispersion measure. *Origin of Dispersion Measure*, 2017.