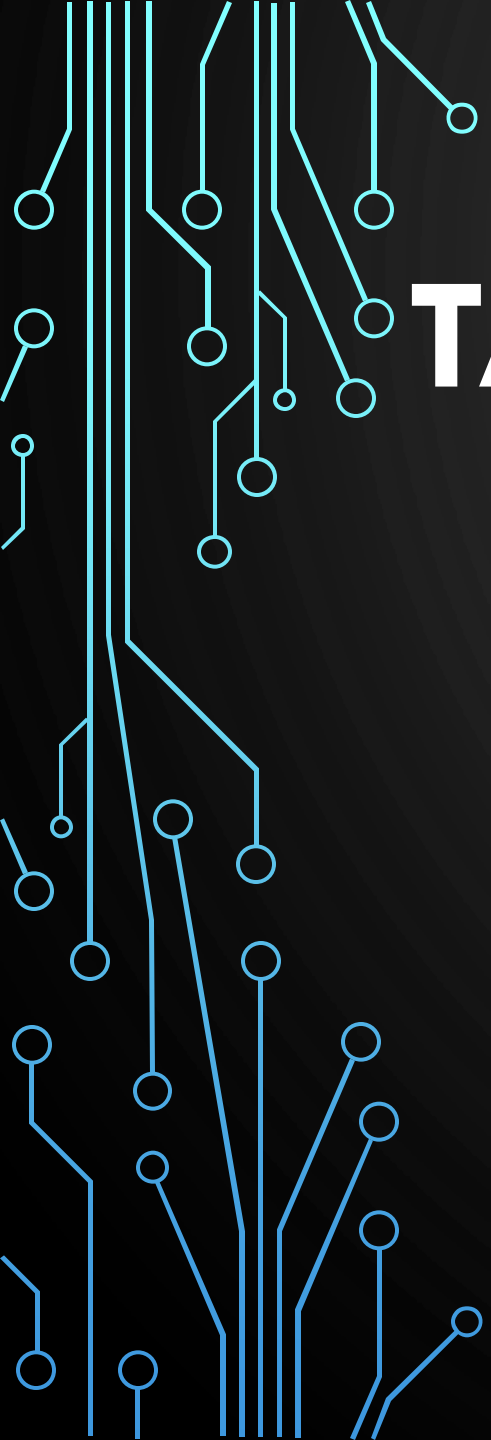
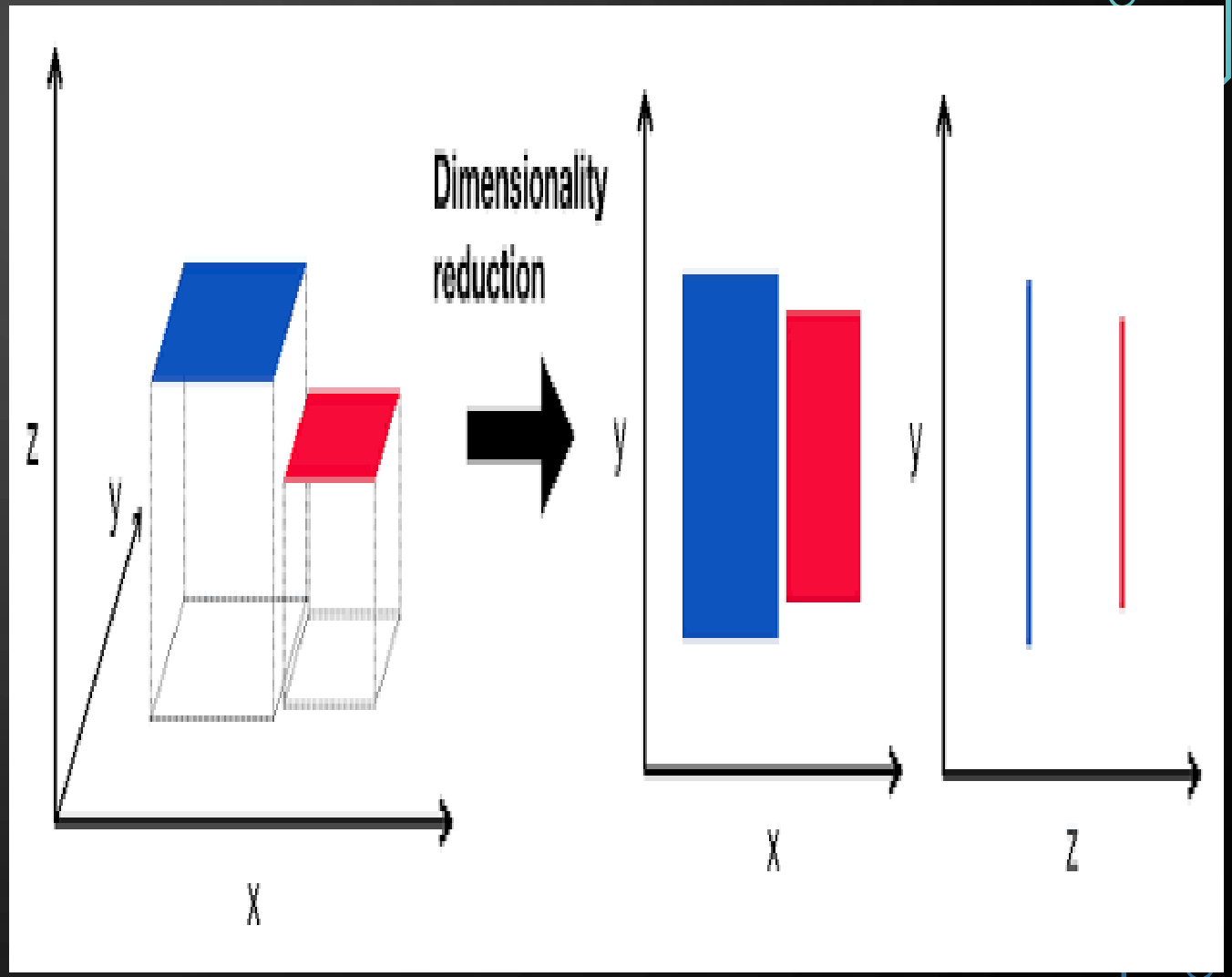


TASK 2_2

PRINCIPAL COMPONENT ANALYSIS

PCA is a method used to reduce the number of dimensions in your dataset, which means fewer parameters before feeding the data to a model.

- This leads to simpler models and removes redundant information.
- It helps reduce features while keeping the most important combined information.
- It also makes data easier to visualize because 2D data is much simpler to plot and understand than, for example, 7D data.



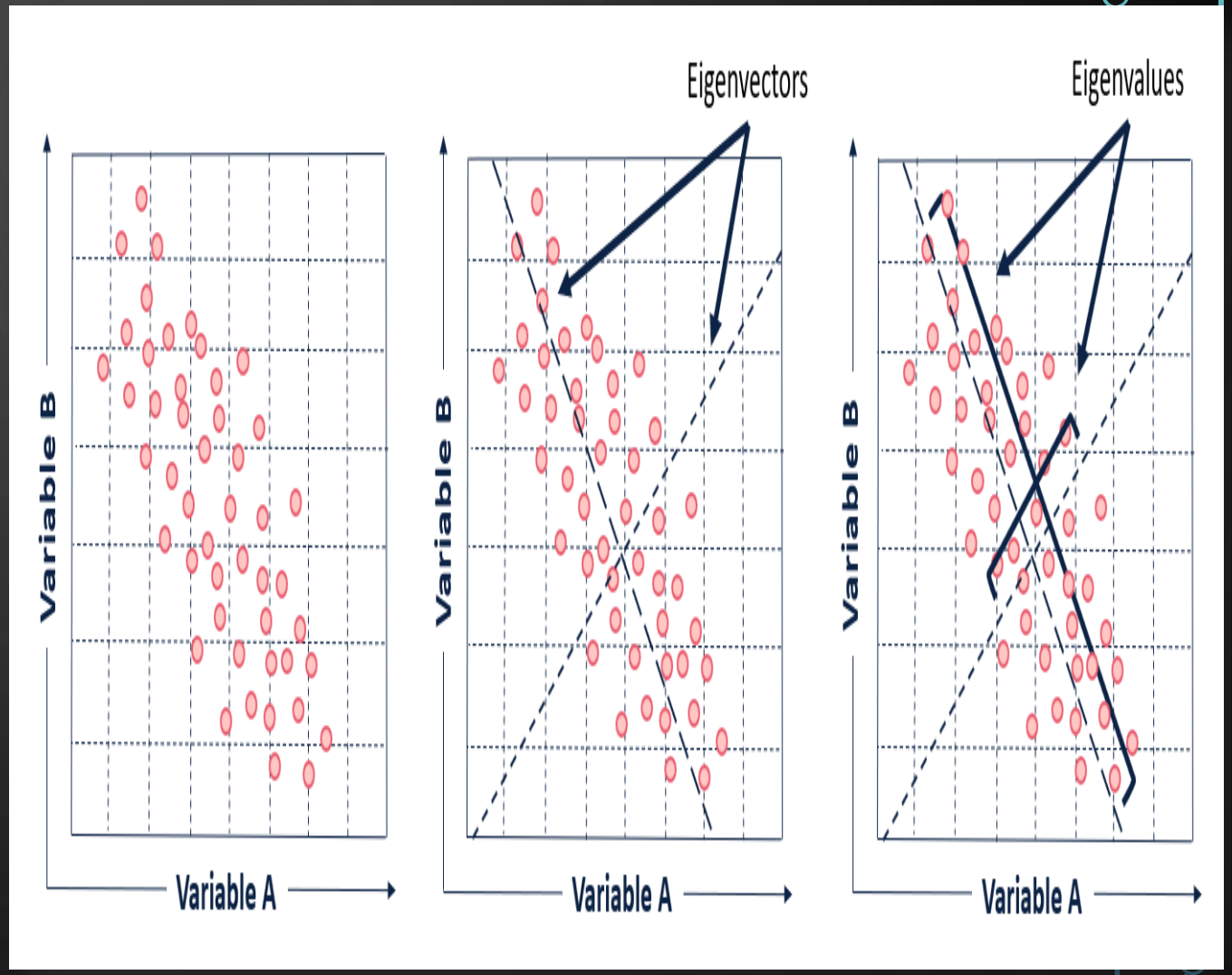
PRINCIPAL COMPONENT ANALYSIS - THEORY

the main idea of PCA is to use **eigenvectors** as a new basis for the data system. Since every vector in the system can be made by combining the basis vectors (called the span), these eigenvectors can describe all the data points after transformation.

Eigenvectors don't change direction like other vectors—they act like the "rotation points" of the system, making them very important.

They can be seen as the constants of the vector that the rest of variables rotate about

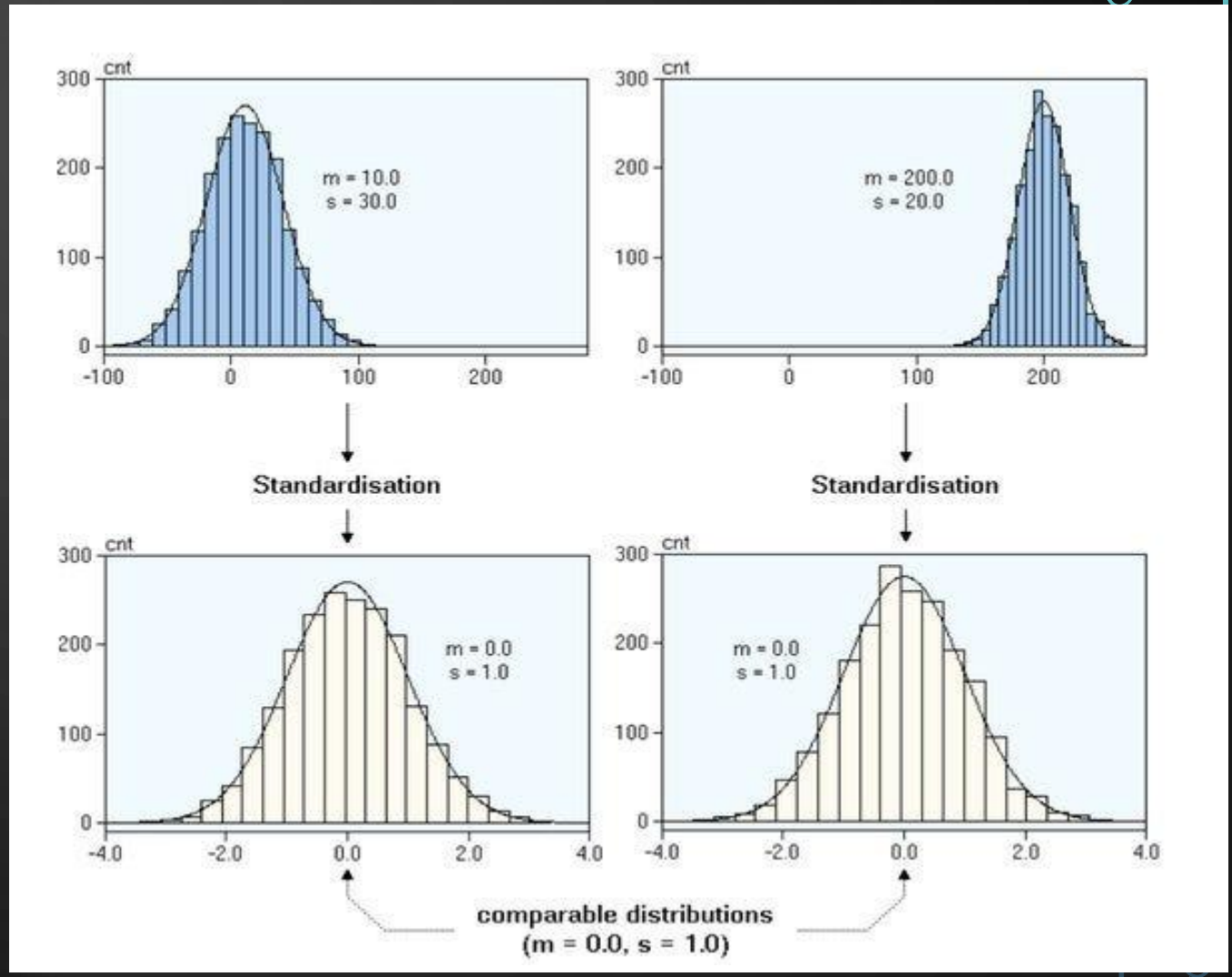
There can be multiple eigenvectors, and how important each one is depends on its **eigenvalue**.



PRINCIPAL COMPONENT ANALYSIS - METHOD

1] **Standardization**: making all the variables centered around the same point, so their values become comparable.

- For example, if you have **age** (values around tens) and **salary** (values around thousands), plotting them together without standardization makes one look like an outlier to the other_no real relation. By standardizing, you make both variables centered around 0 with the same scale (standard deviation).
- You can also set weights to make some variables affect the model more.
- IDs aren't standardized because they're not real parameters; they just identify data (like car names) and don't represent a value to analyze.



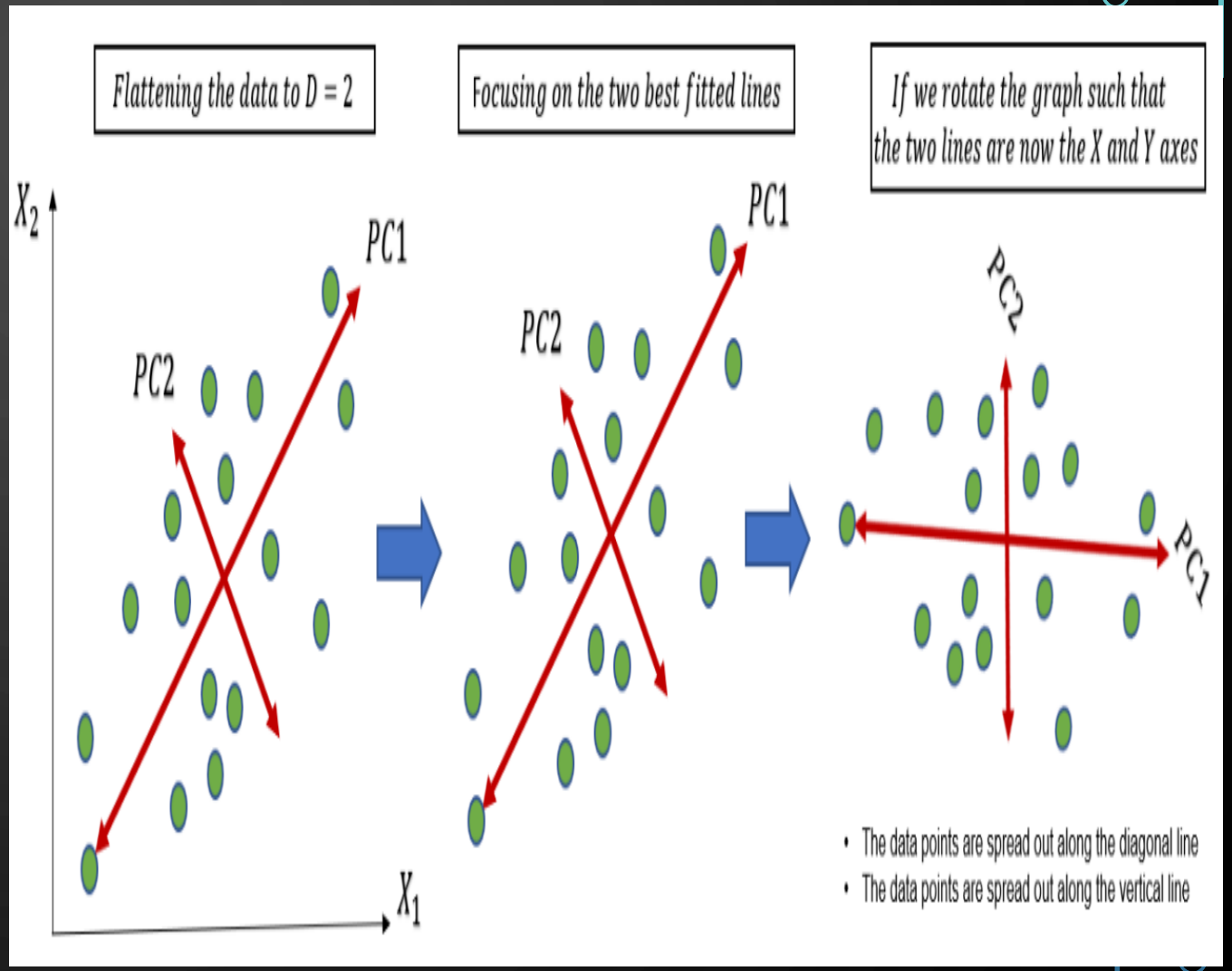
PRINCIPAL COMPONENT ANALYSIS - METHOD

2] **Calculate Covariance Matrix :** This shows how the parameters relate to each other.

- If two variables are highly related (high covariance), you might be able to keep one and drop the other without hurting the accuracy.

3] finding principal components : PCA identifies new basis where the data spread the most

- PC1 : direction of maximum variance
- PC2 : next best direction , perpendicular to PC1
- Then you multiply your vectors by this new matrix to transform it into a 2 dimensions grid with the "most effective" parameters as the axis



Principal Component Analysis (PCA)

💪 Advantages

- Visualize data by reducing dimensions
- Removes multicollinearity
- Removes noise from data
- Reduces model training time
- Reduces model parameters

👎 Disadvantages

- High run-time
- No feature interpretability
- Loss of information
- Only offers linear dimensionality reduction
- Affected by outliers

REFERENCES

HTTPS://WWW.GEEKSFORGEEKS.ORG/DATA-ANALYSIS/PRINCIPAL-COMPONENT-ANALYSIS-PCA/

HTTPS://BLOG.DAILYDOSEOFDS.COM/P/THE-ADVANTAGES-AND-DISADVANTAGES

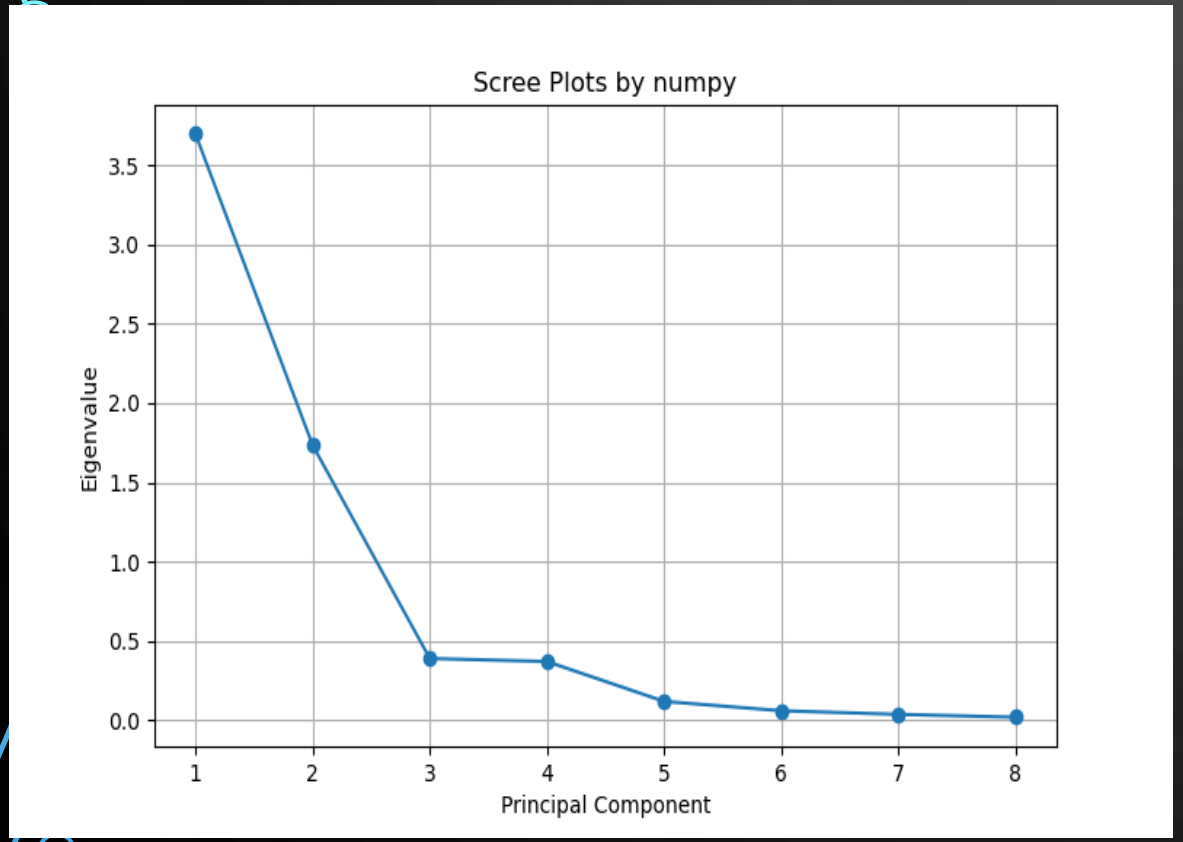
FOR MORE DETAILED INFORMATION ABOUT :

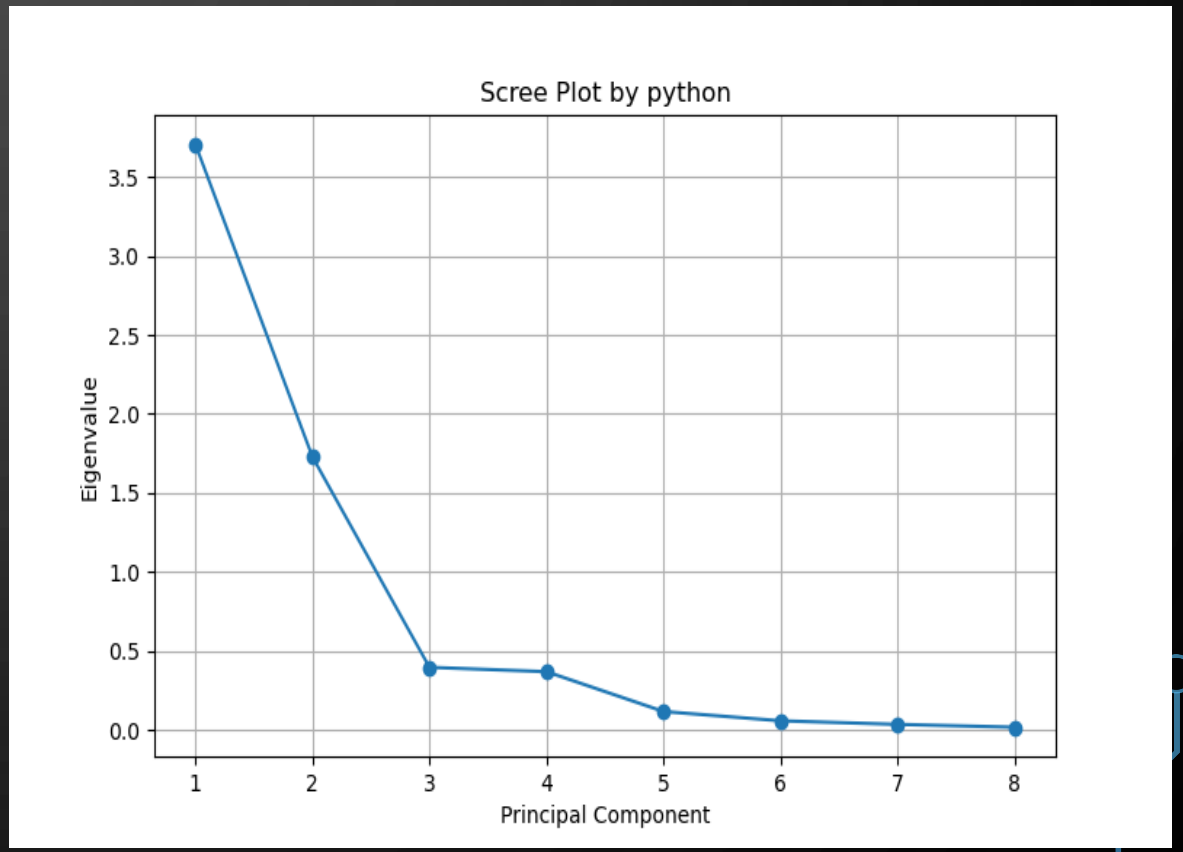
- LINEAR TRANSFORMATION
- EIGENVECTORS AND EIGENVALUES
- REDUCING DIMENSIONALITY
- SPAN AND KERNAL

PLEASE CHECK THE FOLDER LABELED “RESEARCH” THAT I WROTE WHILE STUDYING SESSIONS

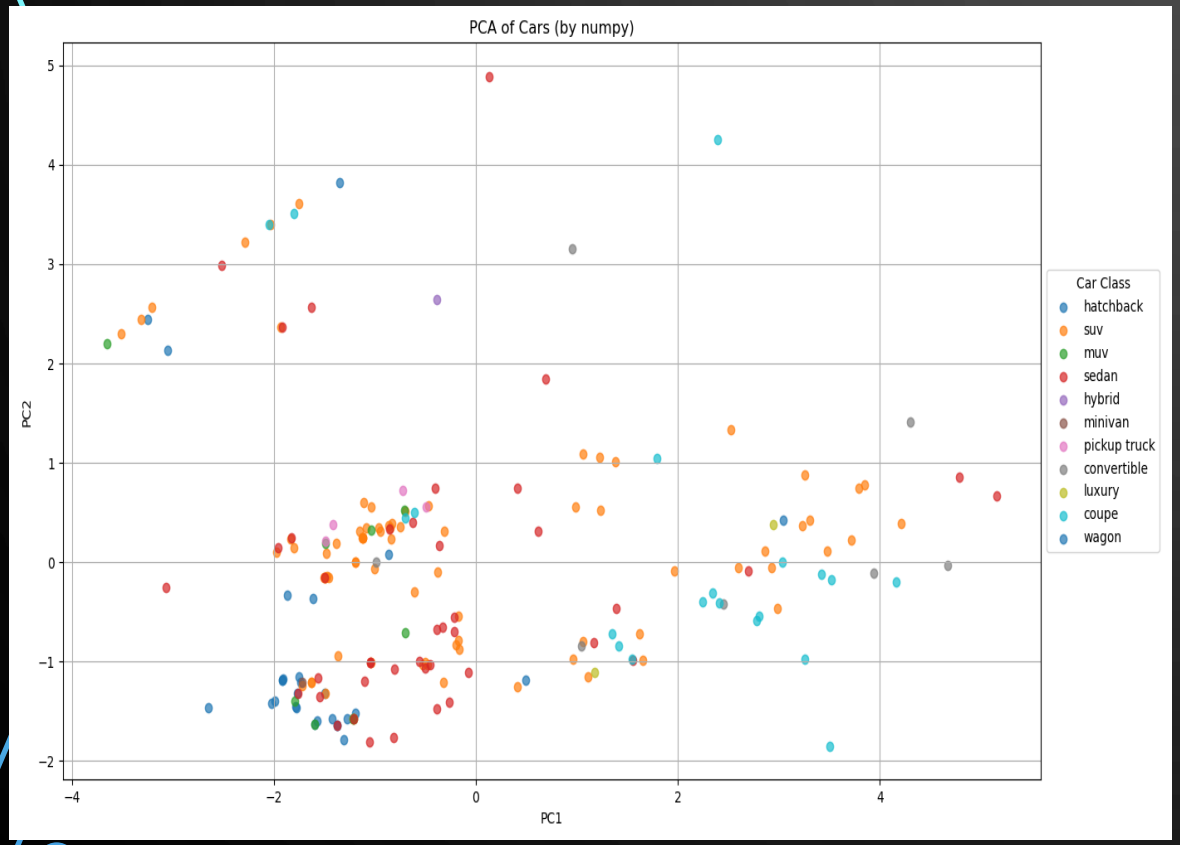
AND HTTPS://WWW.YOUTUBE.COM/PLAYLIST?LIST=PLZHQOBOWTQDPD3MIZZM2XVFITGF8HE_AB

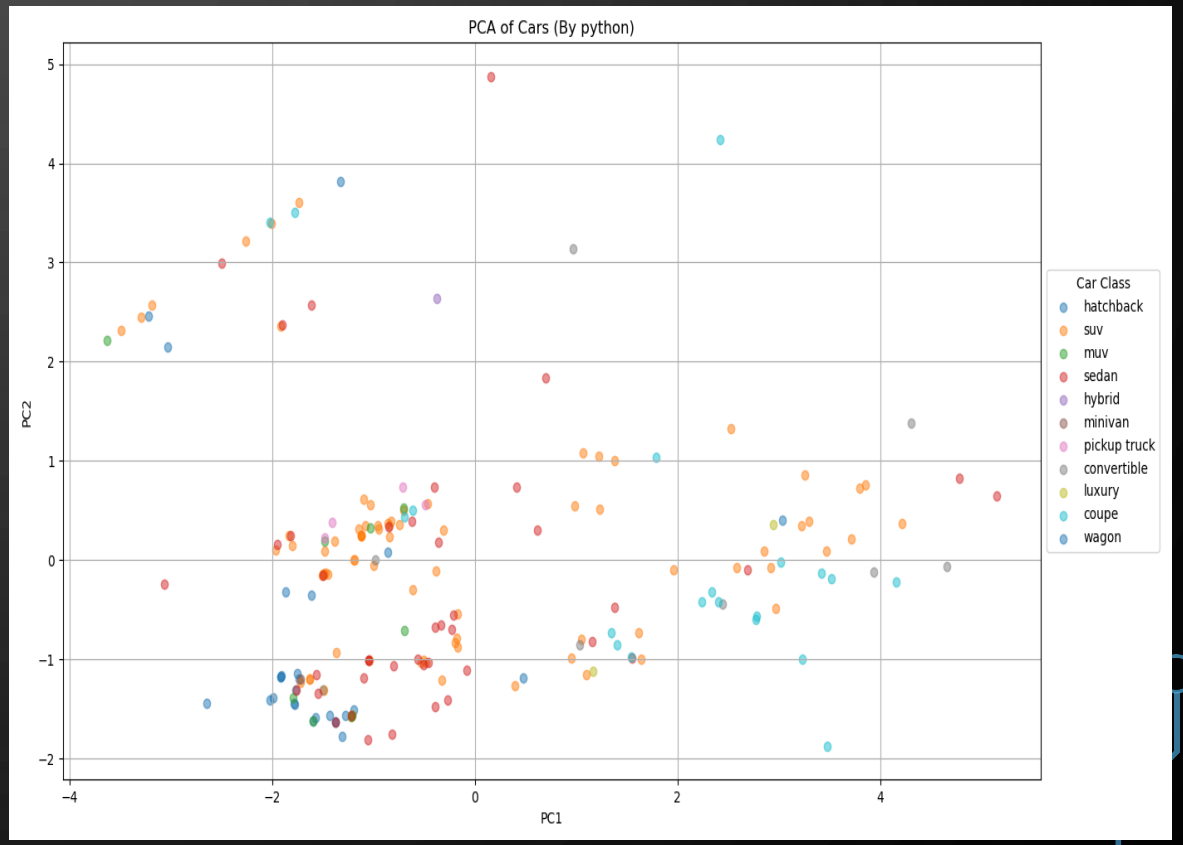
TASK RESULTS





TASK RESULTS





TIME COMPARISON

AMOUNT	PYTHON	NUMPY
1 df	**0.11283**	**0.2267**
100 df	0.19728	0.3602
1000 df	1.1023	1.02
10 k df	7.9109	7.5
100 k df	Crushes	38.922

FINDINGS

Numpy is usually faster but we didn't build the code fully in python as we used decomposition by np which would have taken a lot of cycle time by python

- Nonetheless the time difference between python and numpy becomes more obvious as the dataset gets bigger
  - Due to numpy using some features in pure C