Hello and Welcome to this video demonstration, where I will be showcasing my IAPT project! My task was to create a Jupyter Notebook that conveys the PCA Algorithm as a simple convenient story, targeted towards students who have just completed a **Linear Algebra** or **AI Numerical Methods** course.

Let us proceed to explore the constructed notebook.

For this demonstration, we will load the IRIS dataset by selecting the "IRIS.csv" option from the dataset selection menu.

(Choose IRIS.csv from menu)

Once loaded, the contents of the IRIS dataset are displayed in a Pandas Data frame.

(Show Data frame)

Moving on to the next section, we can now proceed to Dataset Feature Selection. Here, we can choose to remove certain features from the loaded dataset.

Let us opt to remove the final column from such dataset.

(Remove last column)

After the column is removed, we can see the updated contents of the IRIS dataset.

(Show Data frame)

We will proceed on to the next section which deals with the problem of discrete data. Note that PCA works with continuous data, thus placing the need to transform discrete data into continuous data before applying PCA on such dataset.

In this notebook, you'll find five different methods for converting discrete data into continuous data. The methods are:

1. **One-Hot Encoding**
2. **Label Encoding**
3. **Ordinal Encoding**
4. **Count Encoding**
5. **And a Word Embeddings Model**

(For each option showcase the transformed data whilst naming options)

For this demonstration we will be choosing the Ordinal Encoding method

(Choose Ordinal Encoding from the Menu)

After encoding the data, we can move on to visualizing it. We can select any three features to plot, and for this demo, we will proceed to plot the first three.

(Select first three features from Menu)

The resulting interactive plots are displayed below, where the colours are utilized to match the different axis.

(Show Interactive 2D and 3D plots)

Moving on, we need to ensure that the data we feed into the PCA has a uniform distribution. To achieve this, we must normalize the data using the Z-Score Normalization formula. Here is the formula.

(Show Z-Score Normalization Formula)

And here is the resulting Normalized Data.

(Show Normalized Data frame)

In the previous step, we normalized the data using the Z-Score Normalization formula. Now, let's take a look at the resulting normalized data. You may notice that the colour of the points has changed, as the data is now centred around zero.

(Show Interactive 2D and 3D plots)

Additionally, to calculate PCA, there are two approaches that can be used:

1. the SVD approach and
2. the covariance matrix approach.

The **SVD approach** utilises the following formula to perform Singular Value Decomposition on the normalized dataset. The resultant U and Sigma matrices are then multiplied together in order to produce the Principal Components.

(Show SVD Formula)

Here are the Principal Components obtained.

(Show Principal Components)

Next, we can calculate the Variance Ratio of the Principal Components in order to determine the importance of each component and proceed to plot a Scree Plot to illustrate such information.

Here is the Scree Plot

(Show Scree Plot)

And here are the Principal Components plotted in the respective Dimensional Graphs

(Show Interactive 2D and 3D plots)

On the other hand, the **Covariance Matrix** approach utilises the following formula to calculate the Covariance Matrix of the normalized dataset. The eigenvectors of such matrix are sorted by descending order and multiplied by the normalized dataset in order to obtain the principal components.

(Show Covariance Matrix Formula)

Here are the Principal Components obtained.

(Show Principal Components)

Here is the Scree Plot showing the importance of each principal component.

(Show Scree Plot)

And here are the Principal Components plotted in the respective Dimensional Graphs

(Show Interactive 2D and 3D plots)

Comparing both approaches we may note that although the Scree Plots are the same, the visualised data is inverted as the direction of the eigenvectors is different.

(Show Plots)

In addition to the manual method explained earlier, the PCA algorithm can also be easily implemented using the Sklearn library.

Here are the resulting visualizations obtained using this library.

(Show Interactive 2D and 3D plots)

The notebook ends with a brief summary of its contents, highlighting the strengths and weaknesses of the PCA algorithm. The summary also provides an overview of the different types of PCA, including Incremental PCA, Kernel PCA, and Sparse PCA.

Overall, the notebook has presented a comprehensive guide to the PCA algorithm, starting from loading, and selecting the dataset features, to addressing the issue of discrete data whilst also visualizing the filtered and normalized datasets. The notebook has also covered two approaches of calculating PCA from first principles, the SVD approach and the Covariance Matrix approach, as well as a more efficient implementation using the Sklearn library. With the presented interactive plots and scree plot, students will be able to better understand the importance of each principal component and the variance ratio.

(Show Conclusion paragraph)

**Thank you for your time and attention! Have a Good one!**