Applied AI & ML Industry Projects Lab (AAM-IPL)
Transforming Learning into Industry Solutions

Week-2 – Principal Component Analysis

PCA and Ames Housing Project Implementation Overview

V Semester - ML and AIUP, Aug-Oct 2024

Session Date and Time: 14th Sept 2024, 10:30 AM IST – 12:00 Noon IST

Venkateswar Reddy Melachervu
Visiting Faculty and CTO, Brillium Technologies
Department of Computer Science Engineering – AI and ML (CSM)
Email: venkat.reddy.gf@gprec.ac.in

G.Pulla Reddy Engineering College (Autonomous)
G.Pulla Reddy Nagar, Nandyal Road, Kurnool, AP 518007, India
Website: https://www.gprec.ac.in

Disclaimer and Confidentiality Notice

The content of this guest lecture, including all discussions, materials, and demonstrations, code is intended for educational purposes only and reflects the views and opinions of the speaker. While every effort has been made to ensure the accuracy and relevance of the information presented, it should not be considered as legal, financial, or professional advice.

Brillium Technologies retains unrestricted ownership of all information shared during this session. Participants must not record, reproduce, distribute, or disclose any part of the lecture or materials without prior written permission from Brillium Technologies. Unauthorized use or distribution of the content may result in legal action.

Additionally, all trademarks, service marks, and intellectual properties referenced or used in this presentation are the property of their respective owners. No ownership or rights over such third-party content are claimed or implied by the author or Brillium Technologies or by GPREC.

By attending this lecture, you agree to respect the confidentiality of the information shared and refrain from using it in any unauthorized manner. Failure to comply with these terms may result in legal action.

Thank you for your understanding and cooperation.

Lecture Outline

- Machine Learning Phases
- ML Pipeline
- ML Math Model
- Higher Dimensionality
- Dimensionality Reduction
- Common Dimensionality Reduction Techniques
- PCA Overview
- PCA Math Analysis
- Ame Housing Project Overview
- Ames Housing Project Implementation Steps
- Ames Housing Project Timeline
- Ames Housing Project Output Demo

Machine Learning Phases

- What Are you trying to ***Predict***?
- ***Specific and Well Defined***

- Collect ***Best and lot of historical*** Data

- ***Identify Features*** useful for predicting
- Use Measured Characteristics or Build Computations to Build Features

- Select Machine Learning ***Algorithm***
- Train the algorithm for Estimation of ***Parameters***

- ***Predict or inference*** outcome for any unseen data sample

- ***Evaluation*** of Algorithms/Performance by Applying the Algorithm with the Estimated Parameters to a New Data Set

ML Pipeline

Data Collection
- Gathering training data

Data Preprocessing
- Wrangling – handling missing values
- Encoding categorical variables
- Data scaling – normalization / standardization.

Dimensionality Reduction
- Reduce dimensions by transforming data into lower dimensional space consisting of new features which are combinations of the original ones

Feature Selection
- Choose sub-set of existing features/dimensions without transformation of data

Feature Engineering
- Create new features based on existing ones

A

A

Model Training
- LR, SVM, DTC, etc.

Model Evaluation
- Accuracy, Precision, Recall, F1-Score, Confusion Matrix etc.

Prediction
- Use trained model to make prediction

Post-processing
- Interpret results – convert probabilities to labels
- Adjust thresholds
- Deploy model

ML Math Model

Model Hypothesis

$y_{pred} = \overline{a}^T \overline{x} + \overline{b}$

Training Data Set

$\overline{x}_1, \overline{x}_2, \overline{x}_3 \ldots \overline{x}_N, \overline{x}_i \in \mathbb{R}^D$

$y_1, y_2, y_3 \ldots y_N, y_i \in \mathbb{R}^1$

Feature Selection

$\overline{x}_i \in \mathbb{R}^d$

Features or Dimensions Selected are : 1,2,3,...D

Model Training

$\widehat{y} = X_{N \times D+1} \overline{a}$

$Loss = \mathcal{J}(\widehat{y}, \overline{y})$

$\overline{a} = \left(X^T X\right)^{-1} X^T \overline{y}$

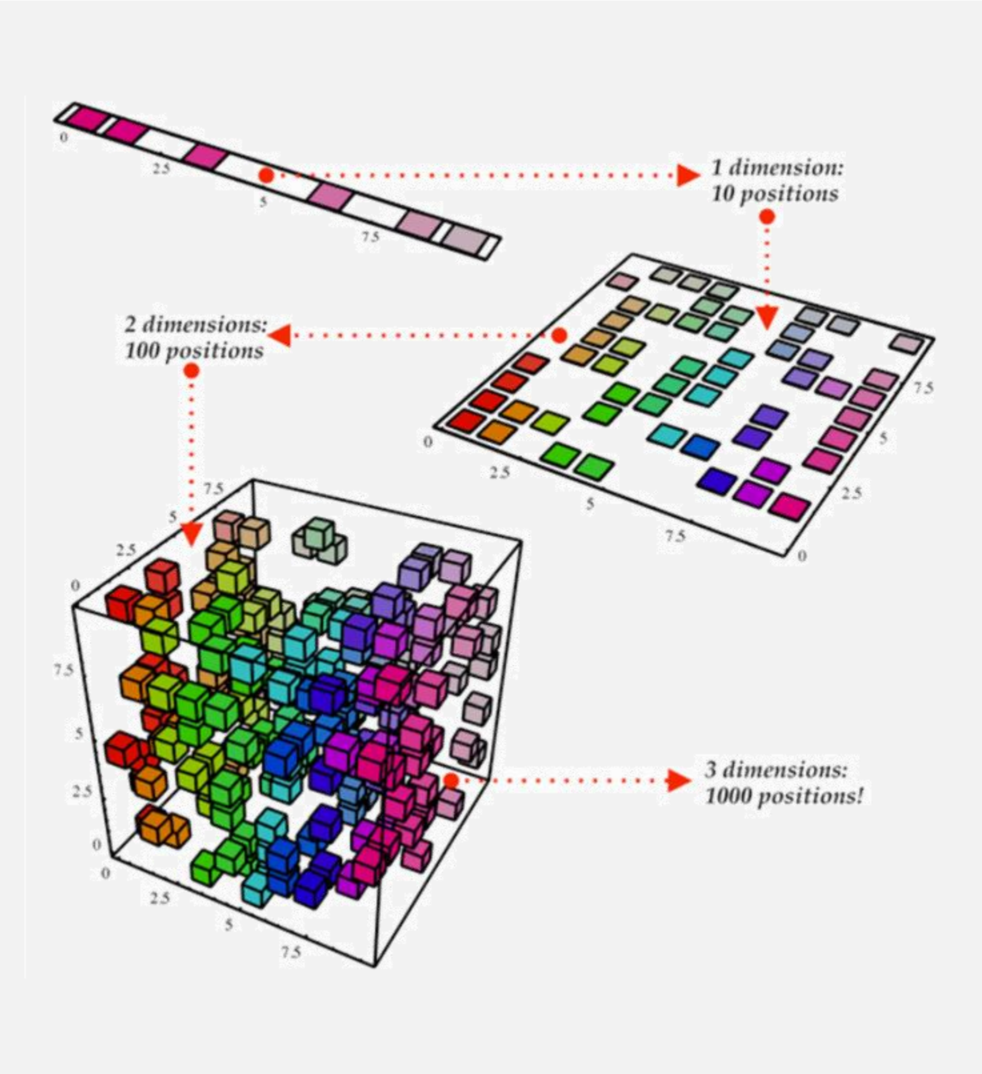
Prediction

$y_{pred} = \overline{a}^T \overline{x}_{input} + \overline{b}$

$\overline{x}_{input}$

- $\widehat{y}$ - Predicted output vector – Dx1
- $\overline{y}$ - Original/input output values/labels vector – Nx1
- y_{pred} - Predicted output - scalar
- $\overline{a}$ - Feature weights vector – Dx1
- $\overline{b}$ - Bias vector – Dx1
- X – Nx(d+1) input matrix
- $\overline{x}_{input}$ - Random input to predict

Higher Dimensionality – Impact on Data Volume



1 Dimension – 10 Data Points (Sample Size of 10)

2 Dimensions – 100 Data Points

3 Dimensions – 1000 Data Points

⋮

N Dimensions – 10^N (Sample Size of 10)

Data volume increases exponentially with increase in data dimension

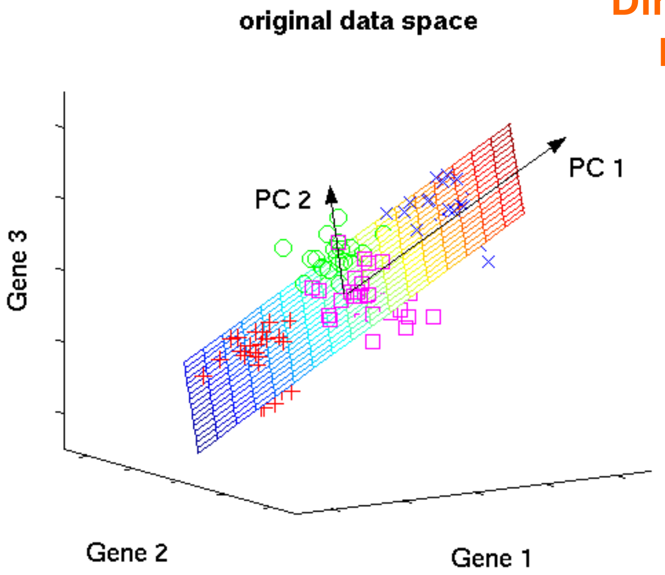
Higher Dimensionality of Data is a Boon and Bane!

Generated by ChatGPT-4o for this Presentation

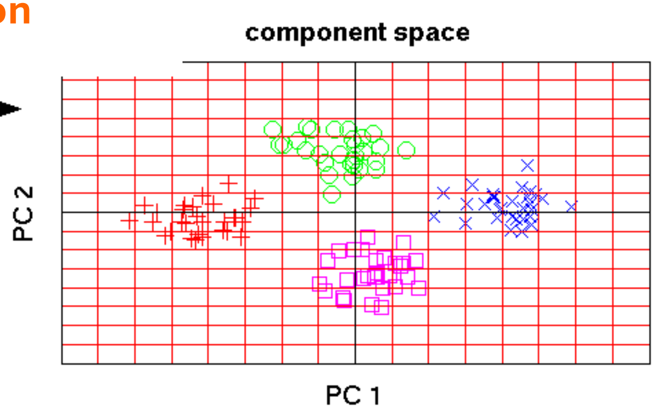
- **Golden Rule of ML : The more data, the better**
- Example of Vast Dimensional Data
  - Life sciences - A single data sample can have millions of features/points – genome, metabolites etc.
- More Data Points
  - Double-edged sword
  - Indiscriminate data points presence results in
    - Introduces noise
    - Slows down training process
    - Reduces model performance
- **So more data can HURT!**
  - Curse of dimensionality
    - Increased computational complexity ➔ higher costs of training and inference
    - Data points become sparse ➔ hard to find meaningful patterns
  - Overfitting ➔ Complex model with poor generalization to unseen data
  - Feature redundancy
  - Data storage and processing challenges

Need a better way to deal with high-dimensionality for Model Efficiency

Introducing Dimensionality Reduction...



Dimensionality Reduction



Three Variables - “Genes”

Two Reduced Variables – Principal Components

3D ➔ 2D

Reducing the number of variables without losing much information to retain or improve model performance

Image Courtesy: [Non-linear PCA by Matthias Scholz]

How Dimensionality Reduction Helps?

Simplifies Models

- Fewer Features ➔ Models become Simpler ➔ Faster Computation & Reduced Storage Requirements

Reduces Overfitting

- Elimination of redundant and/or irrelevant features reduces the chances of overfitting

Improves Visualization

- Reducing higher dimensions to 2D or 3D makes it easy to visualize the patterns

Decreases Noise

- Helps filter out noise by focusing on the relevant features that carry the most variance, improving the model's overall accuracy

Common Dimensionality Reduction Techniques

Feature Selection

Retaining the most relevant variables

Example: ***Time spent on treadmill*** and ***Calories burnt*** **➔** ***Calories Burnt***

Feature Creation

Creating smaller set of new variables that combinations of the input variables

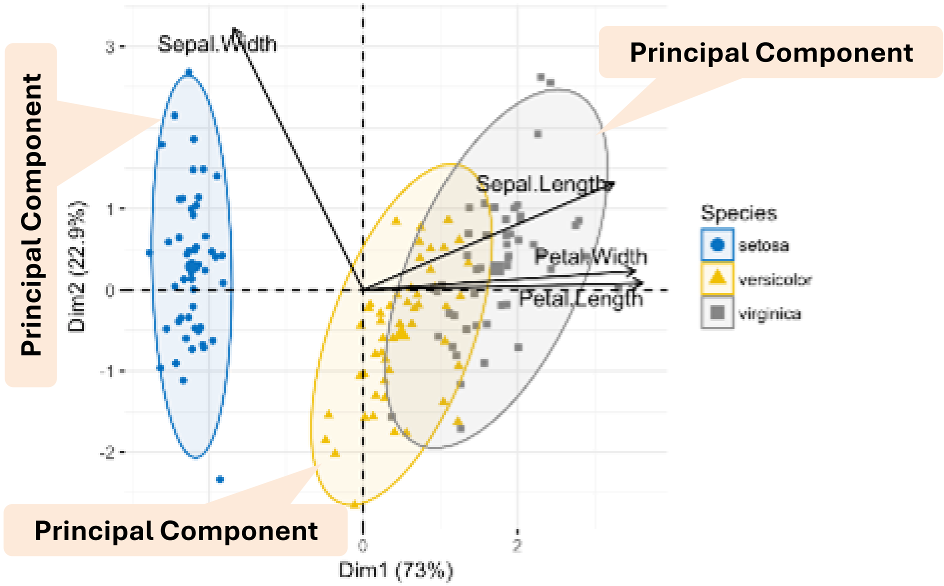
Example: ***Body Weight*** and ***Height*** **➔ BMI** $\left[\frac{\boldsymbol{Weight\ (kg)}}{\boldsymbol{height\ (m)^2}}\right]$

Common Dimensionality Reduction Techniques

Missing Value Ratio	Low Variance Filter	High Correlation Filter	Random Forest
Backward Feature Elimination	Forward Feature Selection	Factor Analysis	**PCA**
Independent Component Analysis	Linear Discriminant Analysis	T-Distributed Stochastic Neighbour Embedding	Uniform Manyfold Approximation and Projection

PCA Overview

- Technique to extract new set of variables from an existing large set of variables aka dimensions
- The newly extracted variables are called **Principal Components**
- The process is **Principal Components Analysis**
- A **principal component** is a linear combination of a set of original variables
- **First principal component** explains the maximum variance in dataset
- **Second principal component** explains the second maximum variance and so on
- Principal components are un-correlated



Images Courtesy: [Statistical Tools for High-throughput Data Analysis]

PCA Analysis – EVD Approach

- Consider a data set of N data points (vectors) of each size Dx1
  - $\bar{x}_1, \bar{x}_2, \bar{x}_3 \dots \bar{x}_N \quad \bar{x}_i \in \mathbb{R}^{D\times 1}$
- Estimate the mean of the data set
  - $\bar{\mu} = \frac{1}{N}\Sigma_{i=1}^{N}\bar{x}_i$
- Subtract the mean from each data point/vector
  - $\tilde{x}_i = \bar{x}_i - \bar{\mu}$
- Find the covariance estimate matrix of the data set
  - $R = \frac{1}{N}\Sigma_{i=1}^{N}\tilde{x}_i\tilde{x}_i^T$
- We need a vector $\bar{\boldsymbol{p}}$ that gives maximum spread for first principal component
  - The eigenvector $\bar{\boldsymbol{e}}_1$ corresponding to **maximum eigenvalue of $\boldsymbol{R}$** $\Rightarrow R\bar{e}_1 = \lambda_{max}\bar{e}_1$ provides the direction of maximum spread
  - And the eigenvector $\bar{\boldsymbol{e}}_2$ corresponding to **second maximum eigenvalue of $\boldsymbol{R}$** $\Rightarrow R\bar{e}_2 = \lambda_{max}\bar{e}_2$ provides the direction of maximum spread
  - And…so on
- The **principal directions** are $\bar{e}_1, \bar{e}_2, \bar{e}_3, \dots \bar{e}_p$ that correspond to p largest eigen vectors $\tilde{P} = [\bar{e}_1\ \bar{e}_2\ \bar{e}_3\ \dots \bar{e}_p]$
- The principal components are projections of $\tilde{x}_i$ along **the principal directions**
  - $$\check{x}_i = \begin{bmatrix} \bar{e}_1^T \\ \bar{e}_2^T \\ \bar{e}_3^T \\ \vdots \\ \bar{e}_p^T \end{bmatrix} \tilde{x}_i = \tilde{P}^T\tilde{x}_i = \begin{bmatrix} \bar{e}_1^T\tilde{x}_i \\ \bar{e}_2^T\tilde{x}_i \\ \bar{e}_3^T\tilde{x}_i \\ \vdots \\ \bar{e}_p^T\tilde{x}_i \end{bmatrix}$$

Variance

- For a scalar set $X = \{x_1, x_2, x_3, \dots, x_n\}$
  - $Variance - \sigma_X^2 = \frac{1}{n}\Sigma_{i=1}^{n}x_i$
- For a random vector $\bar{x} = [x_1, x_2, x_3, \dots, x_n]^T$
  - $Var(\bar{x}) = \mathbb{E}[(\bar{x} - \mathbb{E}[\bar{x}])(\bar{x} - \mathbb{E}[\bar{x}])^T]$
  - $\mathbb{E}[\bar{x}]$ - Expectation/Mean vector of $\bar{x}$

Covariance

- For two scalar variables $X = \{x_1, x_2, x_3, \dots, x_n\}$ and $Y = \{y_1, y_2, y_3, \dots, y_n\}$
  - $Cov(X, Y) = \frac{1}{n}\Sigma_{i=1}^{n}(x_i - \mu_X)(y_i - \mu_Y)$
  - μ_X, μ_Y are the means of X and Y
- For two random vectors $\bar{x}$,$\bar{y}$
  - $Cov(\bar{x}, \bar{y}) = \mathbb{E}[(\bar{x} - \mathbb{E}[\bar{x}])(\bar{y} - \mathbb{E}[\bar{y}])^T]$

Correlation

- For two scalar variables X and Y
  - $R(X, Y) = \frac{Cov(X,Y)}{\sigma_X\sigma_Y}$
  - σ_X, σ_Y are the SD of X and Y
- For two random vectors $\bar{x}$,$\bar{y}$
  - $Corr(\bar{x}, \bar{y}) = D_{\bar{x}}^{-1}Cov(\bar{x}, \bar{y})D_{\bar{y}}^{-1}$
  - $D_{\bar{x}}, D_{\bar{y}}$ are diagonal matrices of SDs of $\bar{x}$ and $\bar{y}$

EVD - Eigenvalue Decomposition

- EVD is a matrix factorization technique that decomposes a square matrix into a set of eigenvectors and eigenvalues
- Given a square matrix $A_{n\times n}$, the eigenvalue decomposition of A is:
  - $\boldsymbol{A = U\Lambda U^{-1}}$
- U is matrix of eigenvectors of A – each column is an eigenvector
- Λ is a diagonal matrix of eigenvalues of A where each diagonal element λ_i is an eigenvalue corresponding to the i^{th} eigenvector in U
- U^{-1} is the inverse of eigenvector matrix U
- $UU^T = U^TU = I$

PCA Analysis – SVD Approach

- Consider a data set of N data points (vectors) of each size Dx1
  - $\bar{x}_1, \bar{x}_2, \bar{x}_3 \ldots \bar{x}_N \quad \bar{x}_i \in \mathbb{R}^{D\times 1}$
- Estimate the mean of the data set
  - $\bar{\mu} = \frac{1}{N}\Sigma_{i=1}^{N}\bar{x}_i$
- Subtract the mean from each data point/vector
  - $\tilde{x}_i = \bar{x}_i - \bar{\mu}$
- Derive data matrix X as - $X = \frac{1}{\sqrt{N-1}}\begin{bmatrix}\tilde{x}_1^T \\ \tilde{x}_2^T \\ \tilde{x}_3^T \\ \vdots \\ \tilde{x}_N^T\end{bmatrix}$
- Find the covariance estimate matrix of the data set - $R = \frac{1}{N}\Sigma_{i=1}^{N}\tilde{x}_i\tilde{x}_i^T$
- The SVD of X is $U\Sigma V^T$
- The eigenvectors of R are **the right singular vectors of $\boldsymbol{X}$, $\boldsymbol{V^T}$** $= \begin{bmatrix}\bar{\boldsymbol{v}}_1 \\ \bar{\boldsymbol{v}}_2 \\ \vdots \\ \bar{\boldsymbol{v}}_p \\ \vdots \\ \bar{\boldsymbol{v}}_N\end{bmatrix}$
- Hence, the **principal directions** are $\bar{v}_1, \bar{v}_2, \bar{v}_3, \ldots \bar{v}_p$ right singular vectors that correspond to p largest **singular values of $\boldsymbol{V^T}$** i.e. $\tilde{P} = \left[\bar{v}_1\bar{v}_2\ \bar{v}_3\ \ \ldots \bar{v}_p\right]^{\mathrm{T}}$
- The principal components are projections of $\tilde{x}_i$ along **the principal directions**
  - $\tilde{x}_i = \left[\bar{v}_1\bar{v}_2\ \bar{v}_3\ \ \ldots \bar{v}_p\right]^{\mathrm{T}}\tilde{x}_i = \tilde{P}\ \tilde{x}_i = \begin{bmatrix}\bar{v}_1\tilde{x}_i \\ \bar{v}_2\tilde{x}_i \\ \bar{v}_3\tilde{x}_i \\ \vdots \\ \bar{v}_p\tilde{x}_i\end{bmatrix}$

SVD – Singular Value Decomposition

- SVD is a matrix factorization technique that decomposes any matrix into 3 other matrices
- Given a matrix $A_{m\times n}$, the SVD of A is:
  - $\boldsymbol{A = U\Sigma V^T}$
- U is an $m \times m$ matrix whose columns are the **left singular vectors of A**
- Σ is an $m \times n$ diagonal matrix of singular values of A
- V is an $n \times n$ matrix whose columns are the **right singular vectors of A** and V^T is the transpose of V
- $UU^T = U^TU = I = VV^T = V^TV$

$$\begin{bmatrix} & & \\ & \boldsymbol{X} & \\ & & \end{bmatrix} = \begin{bmatrix} & & \\ & \boldsymbol{U} & \\ & & \end{bmatrix}\begin{bmatrix}\sigma_1 & 0 & 0 & \ldots & 0 \\ 0 & \sigma_2 & 0 & \ldots & 0 \\ 0 & 0 & \sigma_3 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \sigma_N\end{bmatrix}\begin{bmatrix} & & \\ & \boldsymbol{V^T} & \\ & & \end{bmatrix}$$

AAM-IPL Project - Ames Housing

Project Context

- You are given the data set of Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, Iowa, USA from 2006 to 2010. The source of this data is Ames, Iowa Assessor's Office.
- This data set comprises of 2930 data samples/rows of each containing 80 columns aka fields aka features aka dimensions. Additionally, some of the column values might be missing in the rows.
- The data set is a classic mix of nominal/categorical values, discrete values, continuous values, and ordinal values for many input features/dimensions.

Provided Files

- Ames Housing Data – ***ames.csv***
- Project shell code file - ***AAM-IPL-Wk-1-PCA-Ames-Housing-Shell-Code.ipynb***

Development Environment

- Computing Language – Python
- IDE – Visual Studio Code with Jupyter Notebook

AAM-IPL Project - Ames Housing

Project Implementation

- Print the number of columns and number of rows/data samples/records in the provided data set
- Identify numerical and categorical columns and print them
- Fill the missing numeric values with mean of the respective column and categorical/ordinal columns with "Missing" category value
- Scale numerical feature and encode categorical features and apply this transformation on the data set
- Calculate cumulative explained variance up to the number of original features (79)
- Plot explained variance ratio and cumulative explained variance ratio against principal components
- Set the threshold of 0.9 (90%) for the cumulative explained variance to find optimal principal components (hyper parameter) and print the count of principal components
- Plot heatmap of principal components correlation
- Pair plot the first 5 principal components
- Generate the PDF from the Jupyter file (code and output) and upload in the respective AAM-IPL assignment of Google Classroom

Scree Plot

- Line plot showing the proportion of the total variance explained by each principal component in a dataset after performing **Principal Component Analysis (PCA)**.
- The x-axis represents the principal components in order of importance.
- The y-axis shows the explained variance ratio for each principal component, which indicates how much of the dataset's variance is captured by that component.
- The idea is to look for the "elbow" or point of inflection in the plot, where the explained variance stops increasing significantly.

Ames Housing Project Timeline

Sr. No.	Date	Project Topic	Comments
10-09-2024 – Tuesday 8:00 PM – PCA Project Details Announcement – Topic, Data Set, Shell Code etc. Announcement Channels – Google Class, Industry Projects WhatsApp Group.			
2	14-09-2024 - Saturday Duration: 1.5 Hrs	Principal Component Analysis (PCA) – Overview/Recap, Interactive Q&A	Online – Google Class
3	15-09-2024 - Sunday Duration: 1.5 Hrs	Principal Component Analysis (PCA) – Implementation, Output Demonstration, Interactive Q&A	Online – Google Class
15-09-2024 – Sunday 11:59 PM - Deadline to upload the project code submission by all students in Google Class.			

Ames Housing Project Output Demonstration

Interested in building a Gen AI application?

Reach out to venkat@brillium.in


Q & A

THANK YOU!

AAML-IPL Brought You in Partnership with:

Brillium Technologies

Sector 7, HSR Layout, Bengaluru 560102, Karnataka, India

Website: www.brillium.in | Email: connect@brillium.in