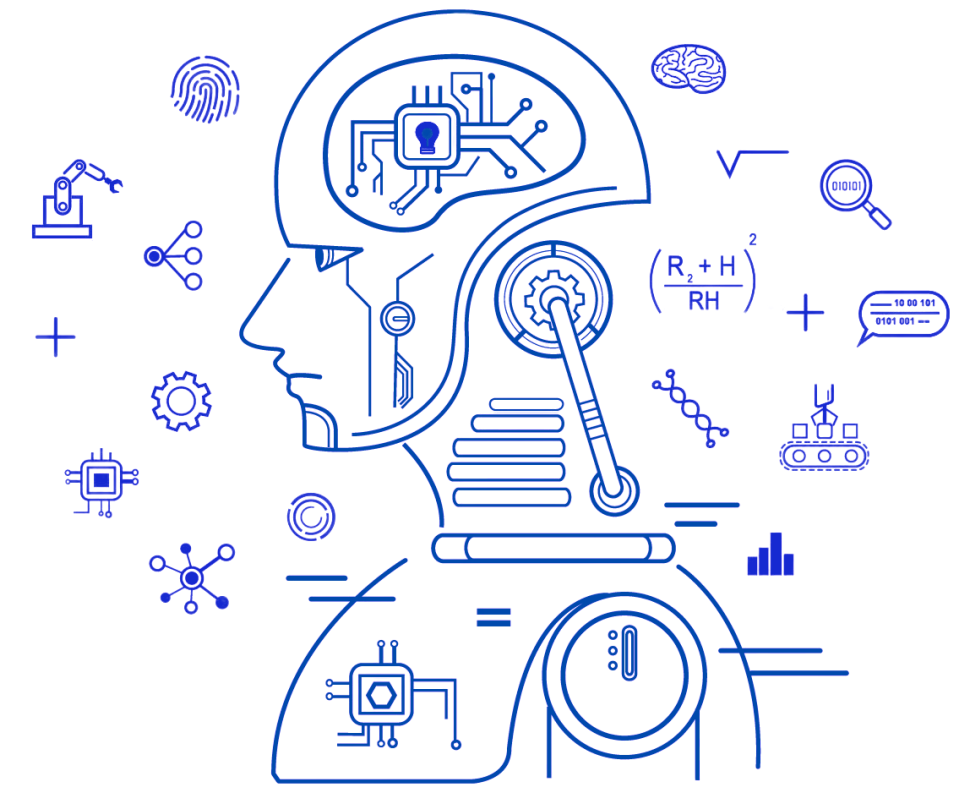


## TEAM PROJECT (E7)

Unveiling architectural design insights from digital layouts using ML



### Team-2

Harsh Anand	(22B1249)
Sravan K Suresh	(22B3936)
Chinmay Tripurwar	(22B3902)
Shashant Jindal	(22B2137)

# Contents

1

## **Problem Description**

Digitalization initiative for building layout designs and the objectives explained point-wise

2

## **Solution Summary**

Overview of all the methods used like image processing techniques, feature extraction, PCA analysis, and predictive modeling for layout classification

3

## **Detailed Results**

Comprehensive analysis and insights derived from applying data science methods to the layout dataset, including visualization of principal components, classification results, and evaluation metrics

4

## **Why Trust Our Model?**

Explanation of the robustness and effectiveness of our data science and machine learning approach in analyzing building layout designs

5

## **Challenges Faced**

Listing the problems and difficulties we had to go through while analysing and classifying the image data

6

## **Learnings and Takeaways**

Major lessons that we learnt while working on image data and the different methods and skills that we acquired in the process



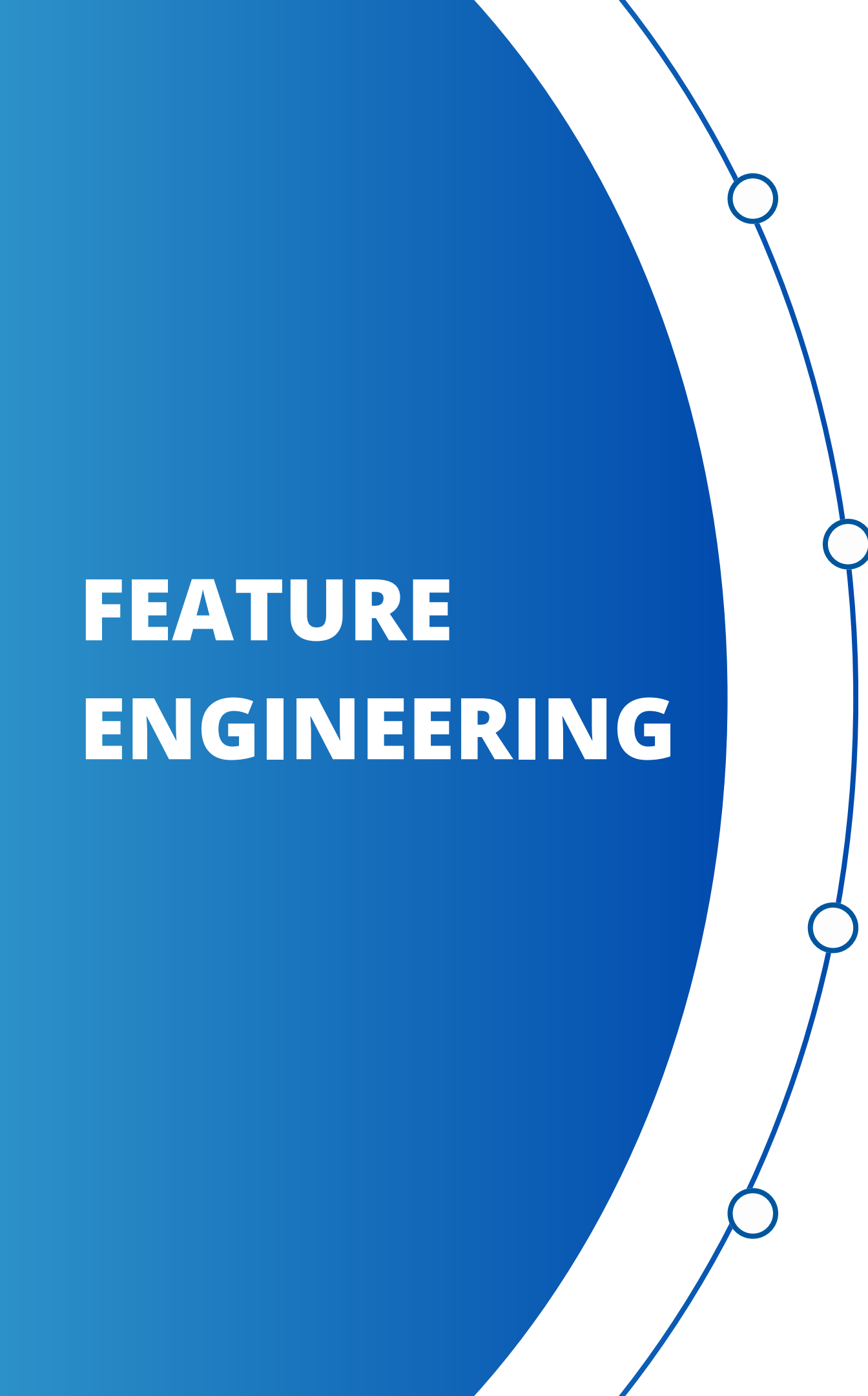
# PROBLEM STATEMENT

- The company has turned their building layouts into digital images, each being 640x480 pixels. They have 1183 such images
- Our goal is to group similar designs together using data science methods to group similar designs into families
- Designs will also be sorted into Low, Medium, and High Complexity categories, making it easier to manage and compare them
- The company plans to predict building layout that is similar to specified dimensions and complexity preferences, speeding up the design process



## **SOLUTION SUMMARY**

- Identify key features using OpenCV for image processing
- Performed PCA to determine the most relevant features
- Used these features to categorize the layouts into six families based on similarities using Gaussian Mixture Algorithm
- The second approach for clustering was DBSCAN clustering for grouping layout into families
- Applied K-means clustering to analyze complexity by grouping the layouts into clusters of High, Medium and Low complexity
- Observing the families, we identified the need for symmetry and curvature analysis to enhance classification
- Employed a custom similarity search algorithm to suggest designs based on input features



# FEATURE ENGINEERING

**We used OpenCV library and image processing methods to extract following features :**

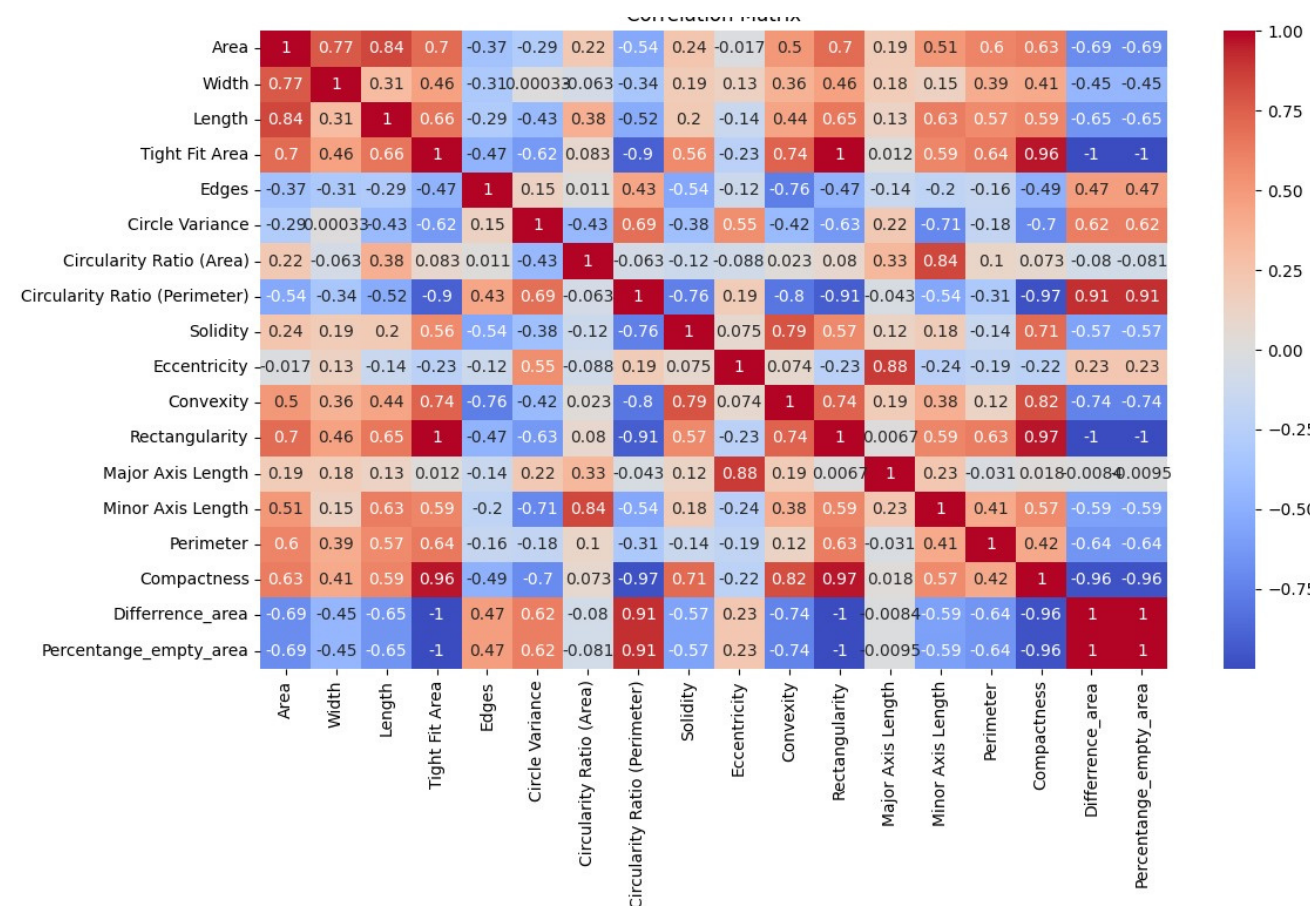
- Area of layout
- Area of border box
- No. of edges
- Circle Variance
- Circularity Ratio (Area)
- Circularity Ratio (Perimeter)
- Solidity
- Eccentricity
- Convexity
- Rectangularity
- Major Axis Length
- Minor Axis Length
- Perimeter
- Compactness factor
- Difference in area
- Percentange of empty area

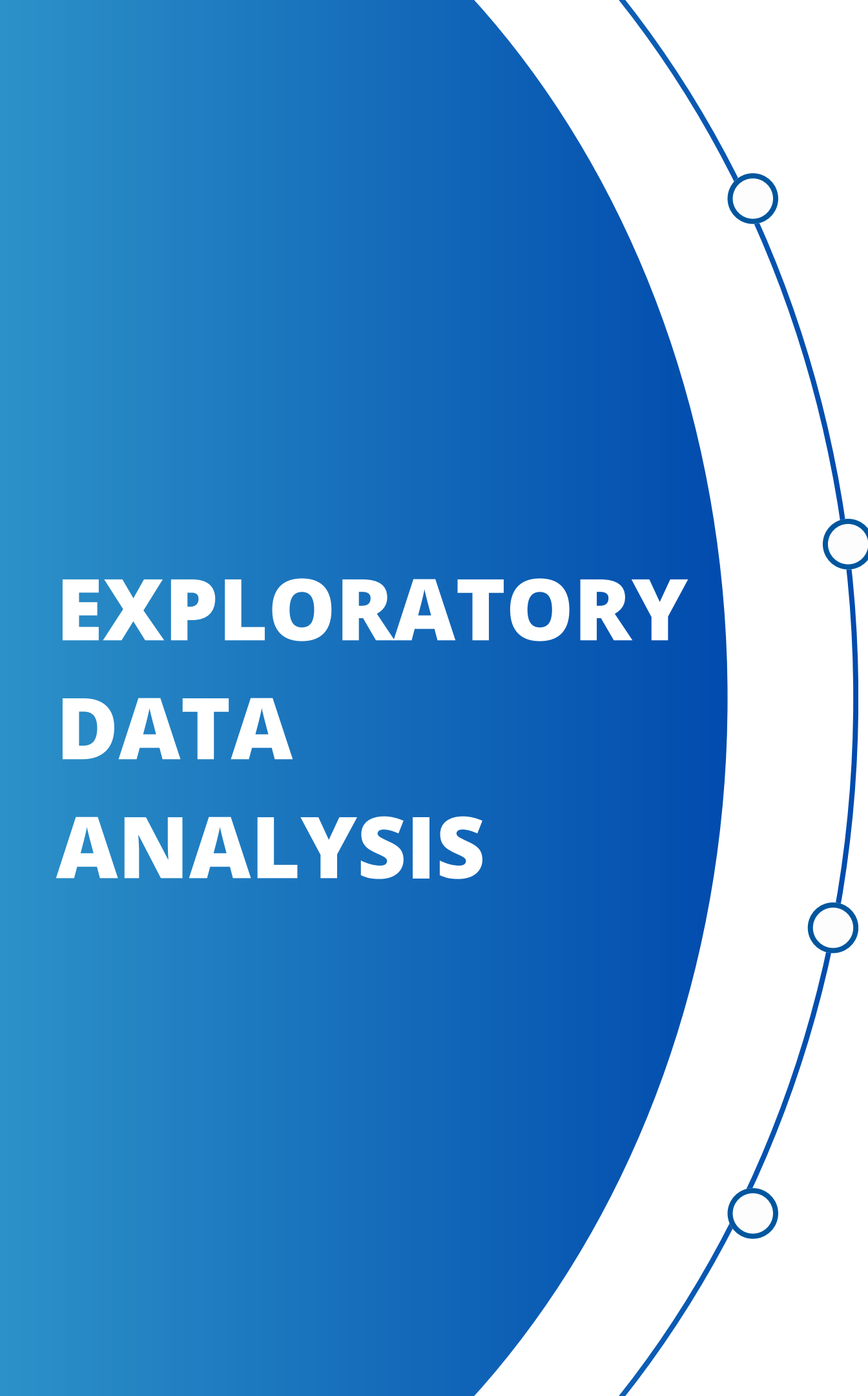


# EXPLORATORY DATA ANALYSIS

## The Exploratory Data analysis of features gave following results:

- There were no missing values found, ie: No blank images
- Many images were detected to be duplicate, hence its copies were dropped from our data. A total of 173 images were removed
- High correlation between box fit area and compactness and between area difference and compactness
- There were very less outliers in the extracted features
- High positive correlation of box area with percentage filled area and area difference

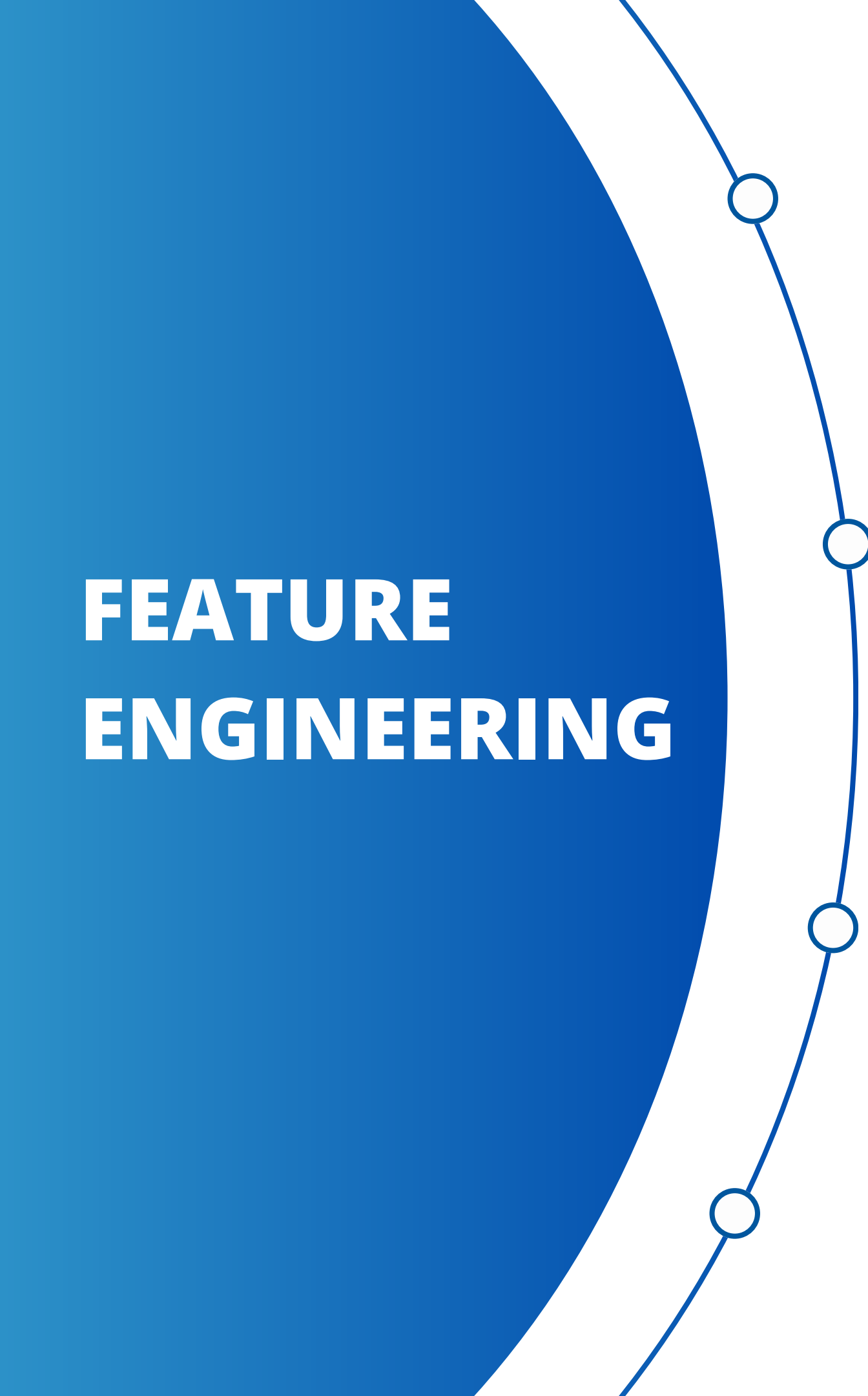




# EXPLORATORY DATA ANALYSIS

**The Exploratory Data analysis of features gave following results:**

- The no. of edges are between 4 to 12, 75% of layouts have less than 7 edges
- The box area of all layouts is almost similar
- The percentage empty area is less than 50% for all the layers
- High positive correlation of box area with percentage filled area and area difference



# FEATURE ENGINEERING

Upon performing principle component analysis, the following features were found to be most influential and relevant:

- Image area
- Difference of area
- Number of edges
- Compactness factor



# TASK 1

## Approach 1: Gaussian Mixture Clustering

- Using the relevant features extracted after PCA as inputs to the Gaussian Mixture Algorithm
- To identify the optimal no. of components for the gaussian mixture model we completed the Bayesian Information Criterion (BIC) and the number of components that minimizes the BIC is selected as the optimal number of components
- Used this GMM model to predict family labels for each layout
- No. of Families : **6**
- We have used Silhouette Score, Davies–Bouldin index as the metric to evaluate our model performance

# CLASSIFICATION INTO FAMILIES

## Family 1:

Index: 0001



Index: 0006



Index: 0007



Index: 0010



Index: 0013



## Family 2:

Index: 0002



Index: 0012



Index: 0020



Index: 0035



Index: 0051



## Family 3:

Index: 0003



Index: 0008



Index: 0009



Index: 0016



Index: 0017



# CLASSIFICATION INTO FAMILIES

## Family 4:

Index: 0004



Index: 0015



Index: 0021



Index: 0024



Index: 0025



## Family 5:

Index: 0005



Index: 0011



Index: 0014



Index: 0018



Index: 0030



## Family 6:

Index: 0022



Index: 0027



Index: 0087



Index: 0091



Index: 0092



# TASK 1

## Approach 2: DBSCAN

- Using the relevant features extracted after PCA as inputs to the DBSCAN algorithm
- Initialised epsilon and minimum samples for our model
- Used this GMM model to predict family labels for each layout
- No. of Families : **46**
- We have used Silhouette Score and Davies Bouldin Index as the metric to evaluate our model performance

## TASK 2

### Complexity Analysis

- Using the features from PCA analysis and families obtained from task 1 as input parameters to KNN we did Complexity analysis
- The algorithm uses the elbow method to determine the optimal number of clusters. It computes the inertia (within-cluster sum of squares) for different values of  $k$  (number of clusters) ranging from 1 to 5
- Using KNN we predicted the high, medium and low complexity clusters
- We used Silhouette Score metric to evaluate our model performance
- We also used visual inspection to determine separation of clusters

# TASK 2

Results

## Low Complexity

Index: 0001



Index: 0002



Index: 0005



Index: 0006



Index: 0007



## Medium Complexity

Index: 0004



Index: 0015



Index: 0022



Index: 0027



Index: 0036



## High Complexity

Index: 0003



Index: 0008



Index: 0009



Index: 0013



Index: 0016



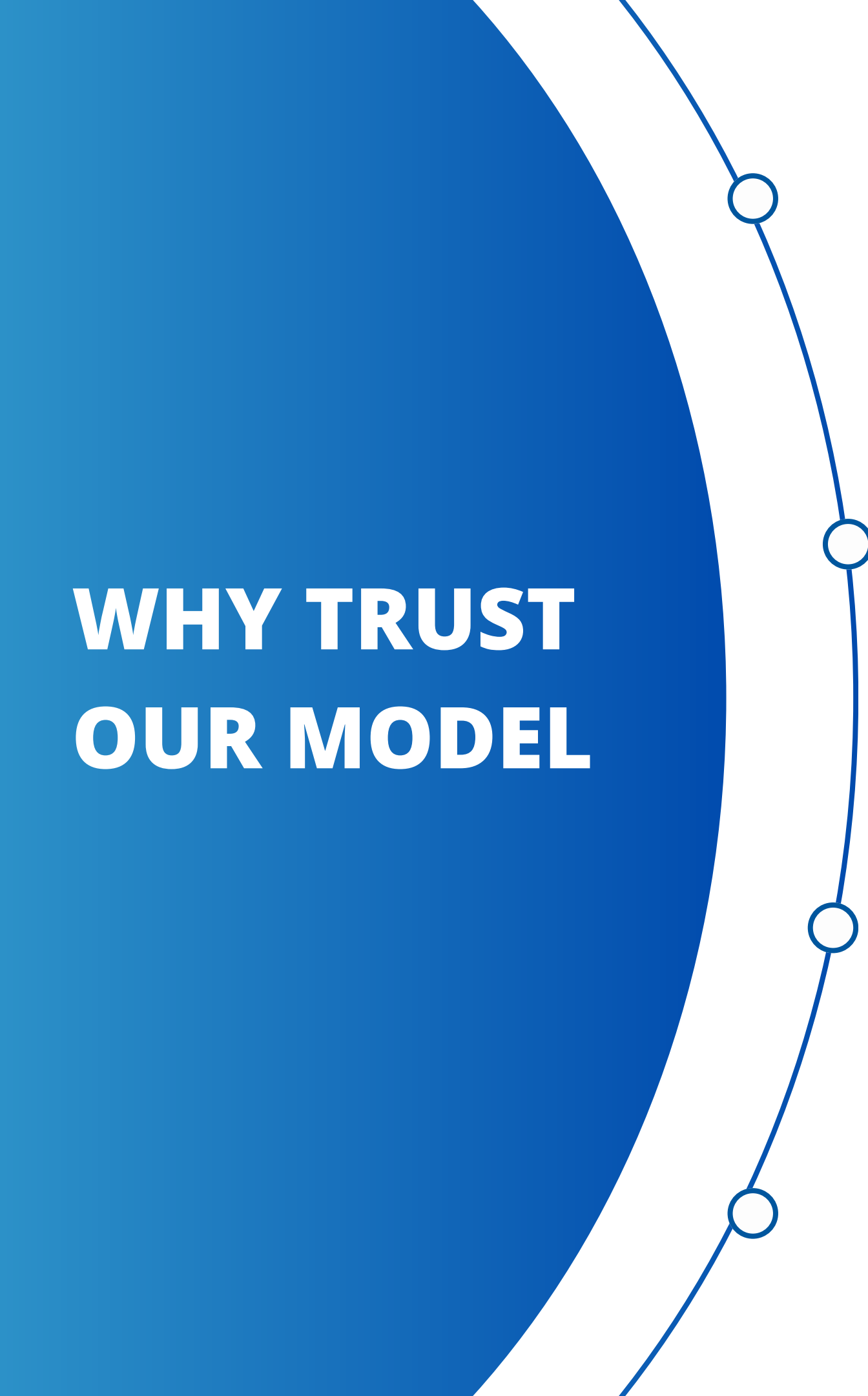




## TASK 3

### Prediction of similar layout

- Using design family, complexity, box area, filled area percentage and number of edges as input parameters
- We created a custom similarity prediction function to determine the most similar layout based on input parameters
- The function predicts the layout based on following priorities:
  - Design Family
  - Complexity
  - No. of Edges
  - Percentage filled area
  - Box area



## WHY TRUST OUR MODEL

- We identified 173 duplicate images successfully which were removed
- We used very intuitive features like no. of edges, perimeter and compactness
- The model metrics like Silhouette Score and Davies Bouldin Index were satisfactory for our models
- The cross validation results for prediction of layout using input features were satisfactory




# CHALLENGES FACED

- Identifying and removing similar images from 1183 images
- Identifying clustering models that work the best for image data
- Clustering of images were difficult as always some outlier image having one or two similar feature crept in
- What features to use for prediction
  - We did feature engineering to mine new features but found it difficult to eliminate the less relevant ones
  - PCA gave results and we struggled a bit to conclude regarding correlated features



# LEARNING AND TAKEAWAYS

- Performing EDA on bitmaps
- Working with bitmaps using various functions of OpenCV library
- Extracting various features of bitmaps like no. of edges, contour area, box fit area
- Validating clustering models using metrics like Silhouette Score, Davies–Bouldin index
- Performing PCA to remove highly correlated features



# APPROACH FOR FINAL PROBLEM

## Innovative Design Generation:

- We can explore generative design algorithms like GANs and VAEs, and research if we can train them on the given dataset
- We can then use these trained models to generate new layout designs based on a predefined set of parameters instead of just predicting the closest design
- These parameters might include the maximum usable area , the complexity of the layout , number of edges, layout area etc



*Thank  
you!*