PROBLEM
Develop a machine learning model to classify product images into five categories. While minimizing manual effort in e-commerce product

GOALS
Dataset analysis
Preprocessing data
Feature extraction
Hypothesis testing
Dimensionality Reduction
ML Modelling

TEAM
Saksham Singh (2022434)
Swarnima Prasad (2022525)
Ritika Thakur (2022468)
Sidhartha Garg (2022499)

Data Science

Final Project

Dataset Description

The e-commerce product dataset will contain product information and high-quality images. Each product entry will include the following attributes

Product ID

Unique identifier for each product.

Attribute Keys

The attribute name specified for the product.

Product Image

URLs of images that are publicly available online or can be accessed using productid.jpg.

Attribute Values

The attribute values specified for the product corresponding to the attribute key.

Category

Type of garment like Tshirt, kurti etc

The dataset comprised two folders of images (train and test), totalling to 100k files, of 2 csv files (train and test) - having values for different attribute for an image, and a parquet file giving information of what information does an attribute in csv tells about a particular category.

The ML Problem Statement

Helping Local Ecommerce Businesses

Have you ever encountered a product listing on an e-commerce platform where the image showed a short - sleeve shirt, but the description claimed it was long-sleeved?

Develop a machine learning model to classify product images into five categories: Kurti, Men's Shirt, Saree, Women's Top & Tunic, and Women's T-Shirt. This solution aims to automate the categorization process, enhancing accuracy and efficiency while minimizing manual effort in e-commerce product cataloging

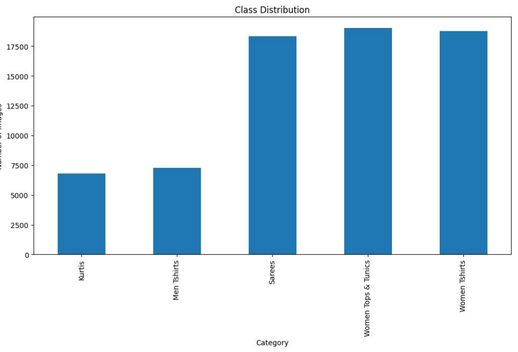
Exploratory Data Analysis

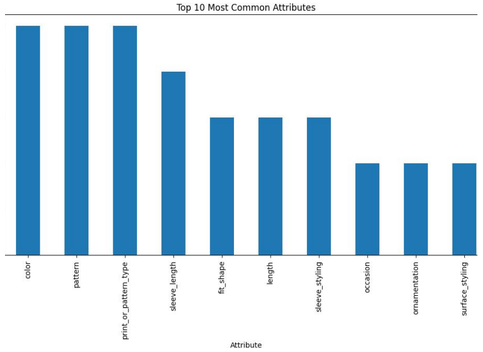
We performed EDA to understand data patterns, relationships, and anomalies, guiding data preparation and model selection.

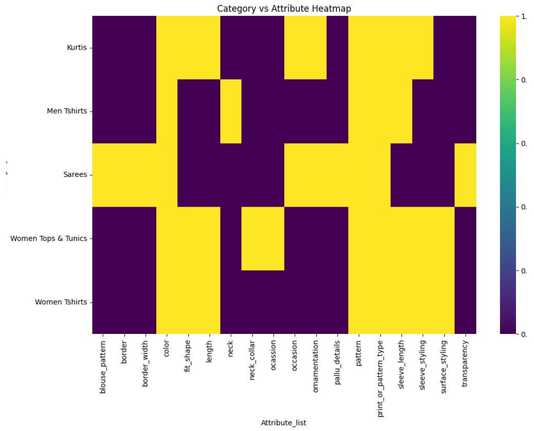
Data Attributes Analysis

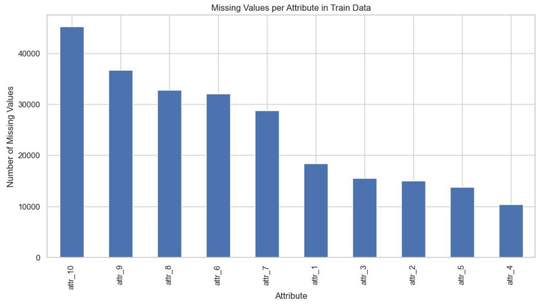
Using Image, matplotlib, seaborn libraries

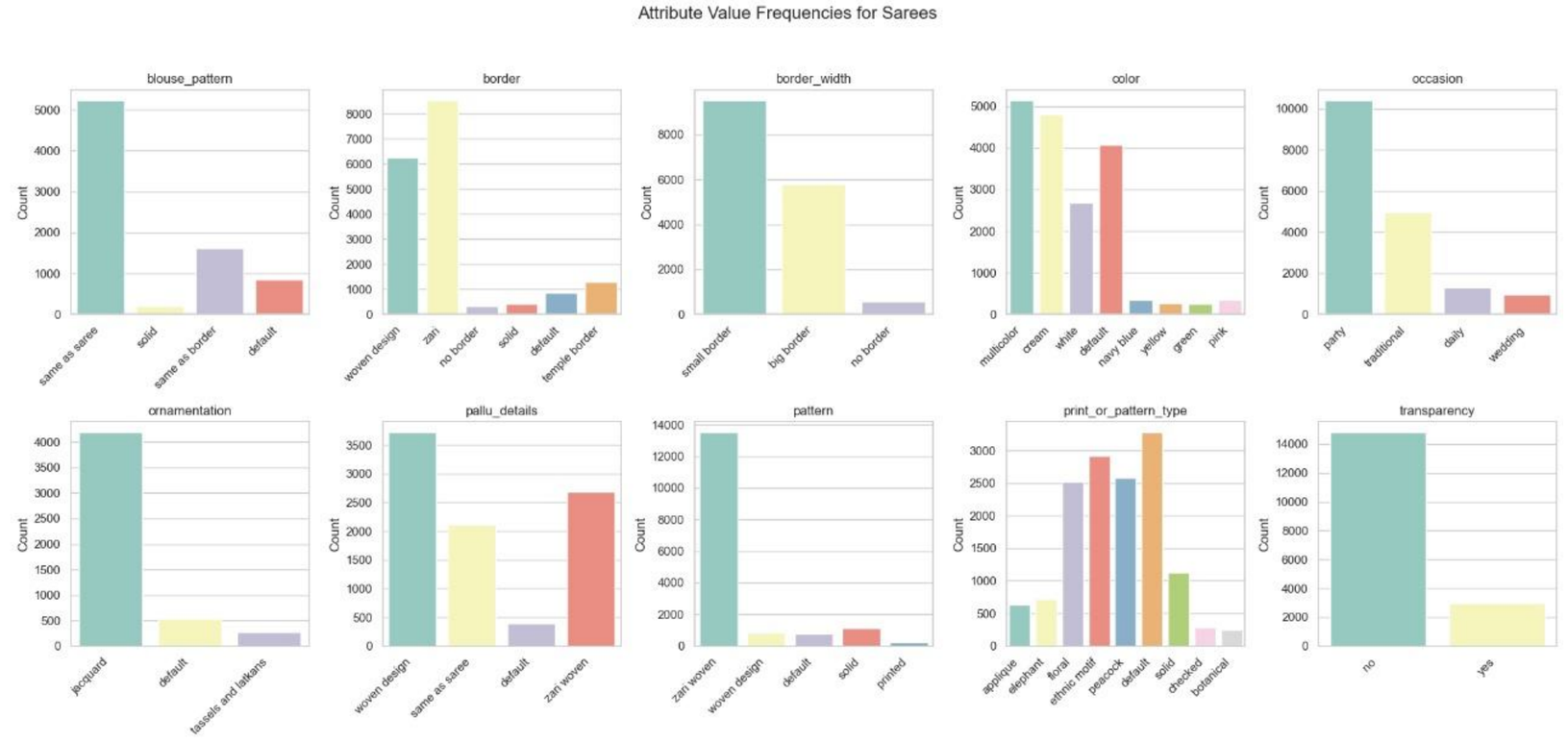
	Category	No_of_attribute	Attribute_list
0	Men Tshirts	5	[color, neck, pattern, print_or_pattern_type, ...
1	Sarees	10	[blouse_pattern, border, border_width, color, ...
2	Kurtis	9	[color, fit_shape, length, occasion, ornamenta...
3	Women Tshirts	8	[color, fit_shape, length, pattern, print_or_p...
4	Women Tops & Tunics	10	[color, fit_shape, length, neck_collar, ocassi...

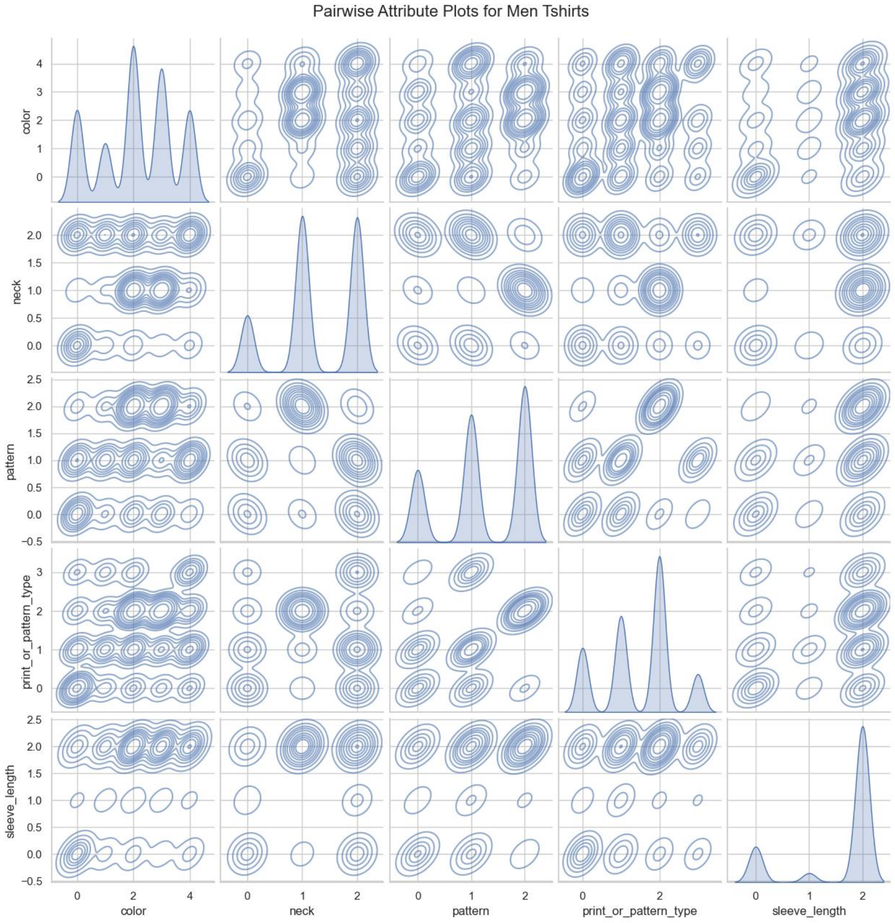






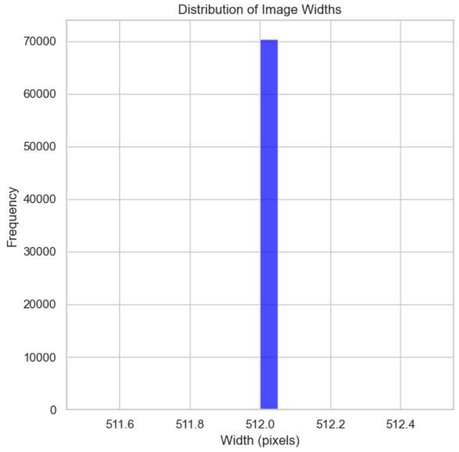


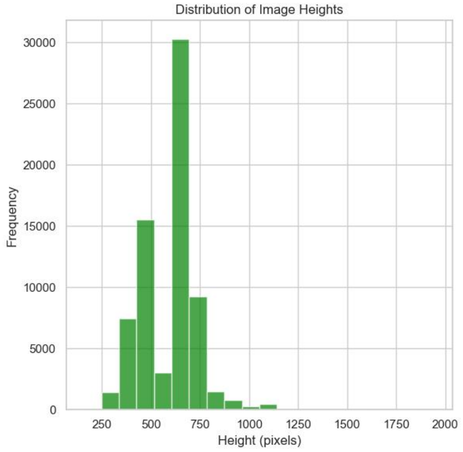


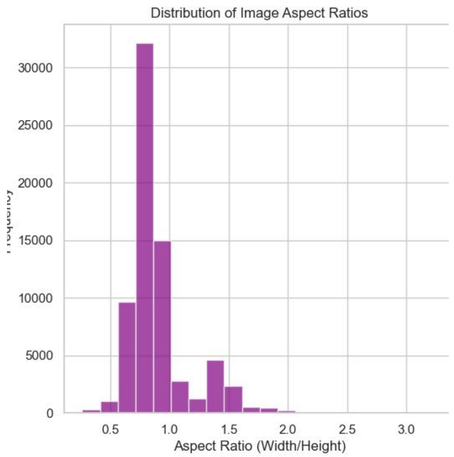


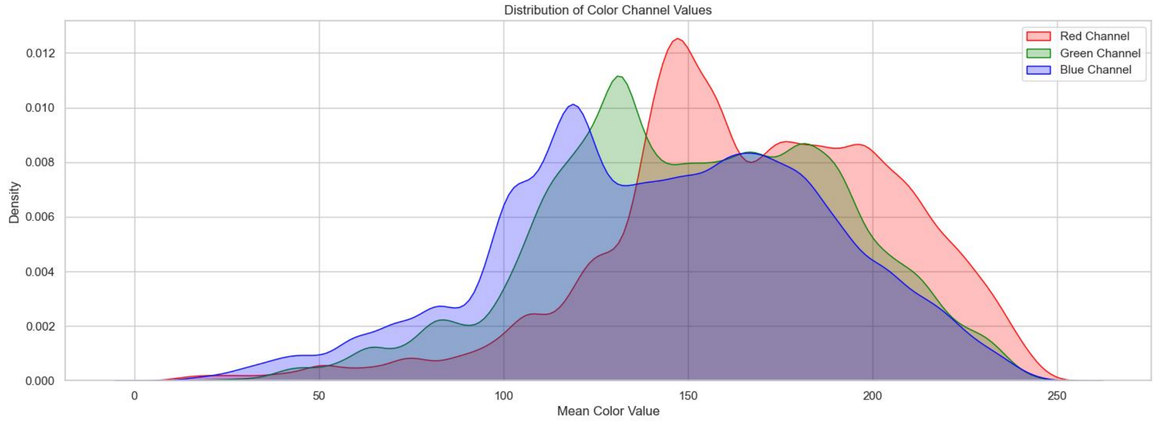
Product Image Analysis

Using Image, matplotlib, seaborn libraries



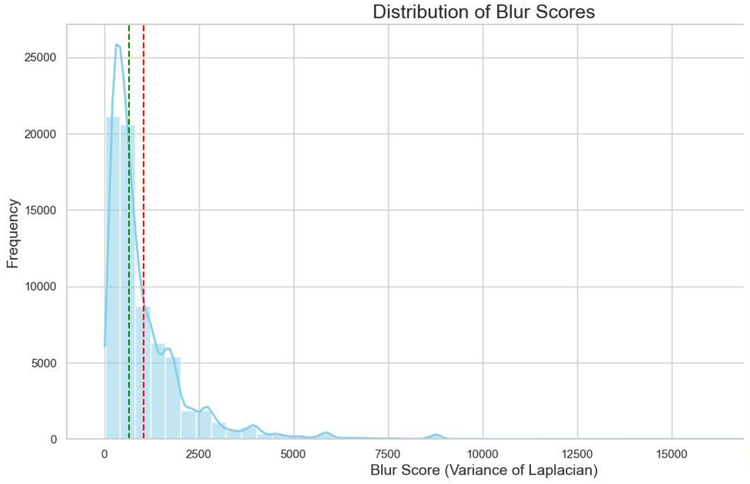






Same width, different heights across images

RGB value distribution for the dataset



Sample of Very Blurry Images

Preprocessing Steps

1. **Reducing dimensionality:**
   - Applied Principal Component Analysis (PCA) to reduce the dimensionality of our dataset while retaining most of the information.
   - Retained the most important information, reduced computational costs, and minimized noise for better model performance.

2. **Image Resizing:**
   - Each image is resized to a standard 360x360 pixels by scaling and padding with zeros as necessary, allowing uniform input size for model compatibility and reducing computational cost.

3. **Feature Extraction:**
   - **Bounding Box Contour**: Extracts the contour of the object to generate bounding boxes and silhouettes. This involves converting the image to grayscale, thresholding, and finding contours.
   - **Color Histogram**: Computes color distribution in red, green, and blue channels across the image, normalized into 20 bins per channel for compact representation.

Resizing Images

Why - Resizing Images Standardizes image size for consistent input into the model

Effect - Improves computational efficiency and supports batch processing

Reducing Dimensionality

Why - Reduces computational costs, simplifies data, eliminates noise, and improves model generalization.

Effect - Minimizes information loss, speeds up processing, enhances robustness, and aids in visualization.

Bounding Box & Silhouette Extraction

Why - Isolates main object, reducing background noise

Effect - Increases model focus on relevant features, reducing background impact

Extracting Color Histograms

Why - Captures color information that helps in identifying color-based attributes

Effect - Boosts feature representation for color classification

Hypothesis Testing

We performed a variety of tests which included Levene’s test, Chi-square test, Kruskal Wallis H-test to understand data patterns, relationships, and anomalies.

Hypotheses 1 - Bounding Box Variance Analysis

Null Hypothesis:

- Statement: Variances within each category are equal to the overall variance across categories.
- Implication: No significant difference in consistency within and across categories.

Alternate Hypothesis:

- Statement: Variances within categories differ significantly from the overall variance across categories.
- Implication: Consistency varies between within-category and across-category variances.

Test Approach

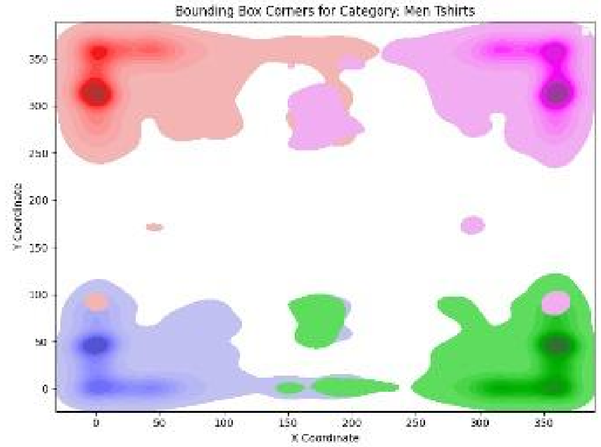
1. Variance Calculation
   - Compute variances for each bounding box attribute (x, y, w, h) within and across categories.
2. Levene's Test
   - Apply Levene's Test to assess statistical differences between variances within and across categories.
   - Threshold: Reject null hypothesis if p-value < 0.05.
3. Interpretation
   - Significant Result ($p < 0.05$): Indicates varying consistency of bounding box attributes across categories.
   - Non-significant Result: Suggests similar variance patterns within and across categories.

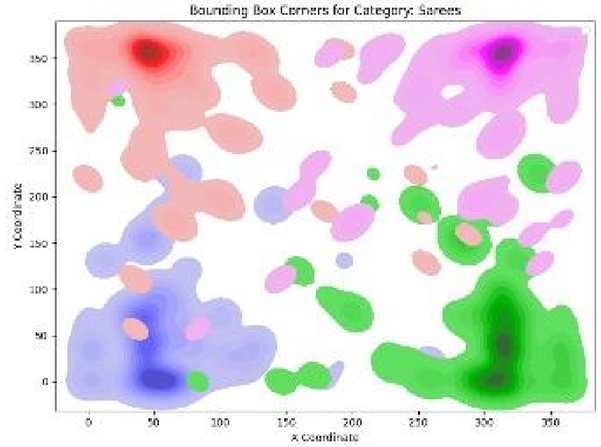
Consistent Variances:

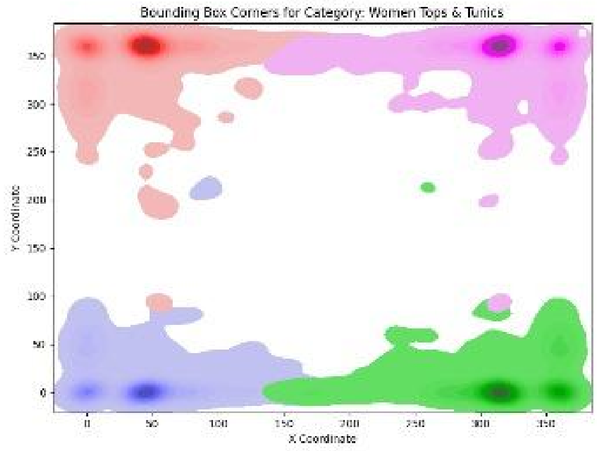
- After calculating the within-category variances and performing Levene's test, if the mean within-category variance for a bounding box attribute (like bounding_box_x, bounding_box_y, etc.) is lower than the overall variance and the p-value is significant, it suggests that bounding boxes are more consistent within each category. This would imply that the variance is smaller within categories compared to across the dataset, which supports the idea of similar framing within categories.

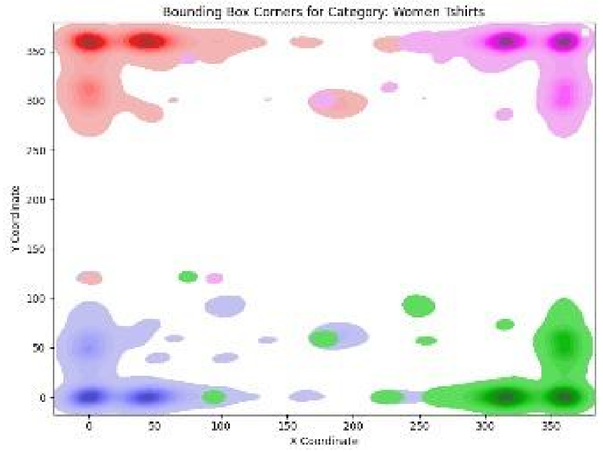
Inconsistent Variances:

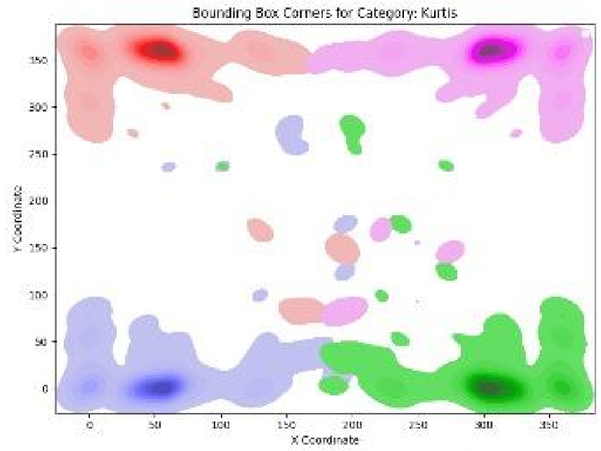
- If the mean within-category variance is higher than the overall variance and the Levene's test is significant, it indicates that bounding boxes are less consistent within each category. This would suggest greater variability in framing even within categories, possibly due to variations in how products are presented in images.











	Attribute	Overall Variance	Mean Within-Category Variance	\
0	bounding_box_x	2048.753085	1855.830536	
1	bounding_box_y	1851.489795	1565.508997	
2	bounding_box_w	5023.666952	4478.164980	
3	bounding_box_h	4481.576046	4079.009192	

	Levene Statistic	p-value
0	147.368229	9.982502e-126
1	1838.211225	0.000000e+00
2	335.500224	1.437120e-286
3	1073.124111	0.000000e+00

```
Significant difference in variances for bounding_box_x (p-value = 9.98250209498638e-126). T
Significant difference in variances for bounding_box_y (p-value = 0.0). This suggests varian
Significant difference in variances for bounding_box_w (p-value = 1.4371198513554306e-286).
Significant difference in variances for bounding_box_h (p-value = 0.0). This suggests varian
```

This suggests variance within categories is significantly different from variance across categories. Null hypothesis rejected.

	Attribute	Overall Mean	Mean Within-Category Mean	Levene Statistic	p-value
0	bounding_box_x	43.711934	42.417888	147.368229	9.982502e-126
1	bounding_box_y	24.433367	23.718586	1838.211225	0.000000e+00
2	bounding_box_w	275.400638	277.669252	335.500224	1.437120e-286
3	bounding_box_h	316.006181	315.857662	1073.124111	0.000000e+00

```
Significant difference in means for bounding_box_x (p-value = 9.9825020949863 81e-126). T
Significant difference in means for bounding_box_y (p-value = 0.0). This suggests means
Significant difference in means for bounding_box_w (p-value = 1.4371198513554306e-286).
Significant difference in means for bounding_box_h (p-value = 0.0). This suggests means
```

Conducting the same test for means suggests that means within categories are significantly different from the overall mean. Null hypothesis rejected.

Reasons to use Levene’s test:

- **Assumption of Homogeneity of Variances**: Levene's test verifies the assumption of equal variances (homoscedasticity) essential for accurate results in further tests like ANOVA.

- **Robustness to Non-Normality**: Unlike Bartlett's test, Levene's test is less affected by non-normality, making it suitable for real-world data.

- **Understanding Variability**: It reveals whether variances within categories differ from the overall variance, indicating consistency in bounding box attributes.

Hypotheses 2 - Color histogram across different color attributes

Null Hypothesis (H0):
There is no significant difference in color consistency (as measured by histograms of RGB values) across different color attributes (e.g., red, blue, green).

Alternative Hypothesis (H1):
There is a significant difference in color consistency across different color attributes.

Test Method

- Statistical Test: Kruskal-Wallis H-test
- This non-parametric test is used to compare three or more independent samples to determine if they come from the same distribution. It is appropriate when the assumptions for ANOVA (such as normality and homogeneity of variances) are violated.
- It tests whether the distributions of the histograms for different colors are the same or differ significantly.

If the p-value from the Kruskal-Wallis H-test is less than 0.05:
Conclusion:

- Reject the null hypothesis (H0). This indicates that there is a statistically significant difference in color consistency across the color attributes you are comparing. In practical terms, this means that at least one of the color attributes has a different distribution of color consistency than the others.

If the p-value is greater than 0.05:
Conclusion:

- Fail to reject the null hypothesis (H0). This suggests that there is no significant difference in color consistency across the color attributes, implying that the color distributions are similar for the attributes considered.

```
Kruskal-Wallis H-test Result: KruskalResult(statistic=586.3381854513796, pvalue=2.832170179004449e-115)
Reject H0: There is a significant difference in color consistency across color attributes.
```

Reasons to use Kruskal Wallis H-test:

- **Non-parametric Nature:** The Kruskal-Wallis test does not assume normality in the data, making it suitable for datasets that may not follow a Gaussian distribution.

- **Comparing Multiple Groups:** It allows for the comparison of three or more independent groups simultaneously, which is efficient when analyzing multiple color attributes.

- **Robustness to Variance Differences:** Unlike ANOVA, the Kruskal-Wallis test is less affected by violations of the homogeneity of variances assumption, making it reliable even when group variances are unequal.

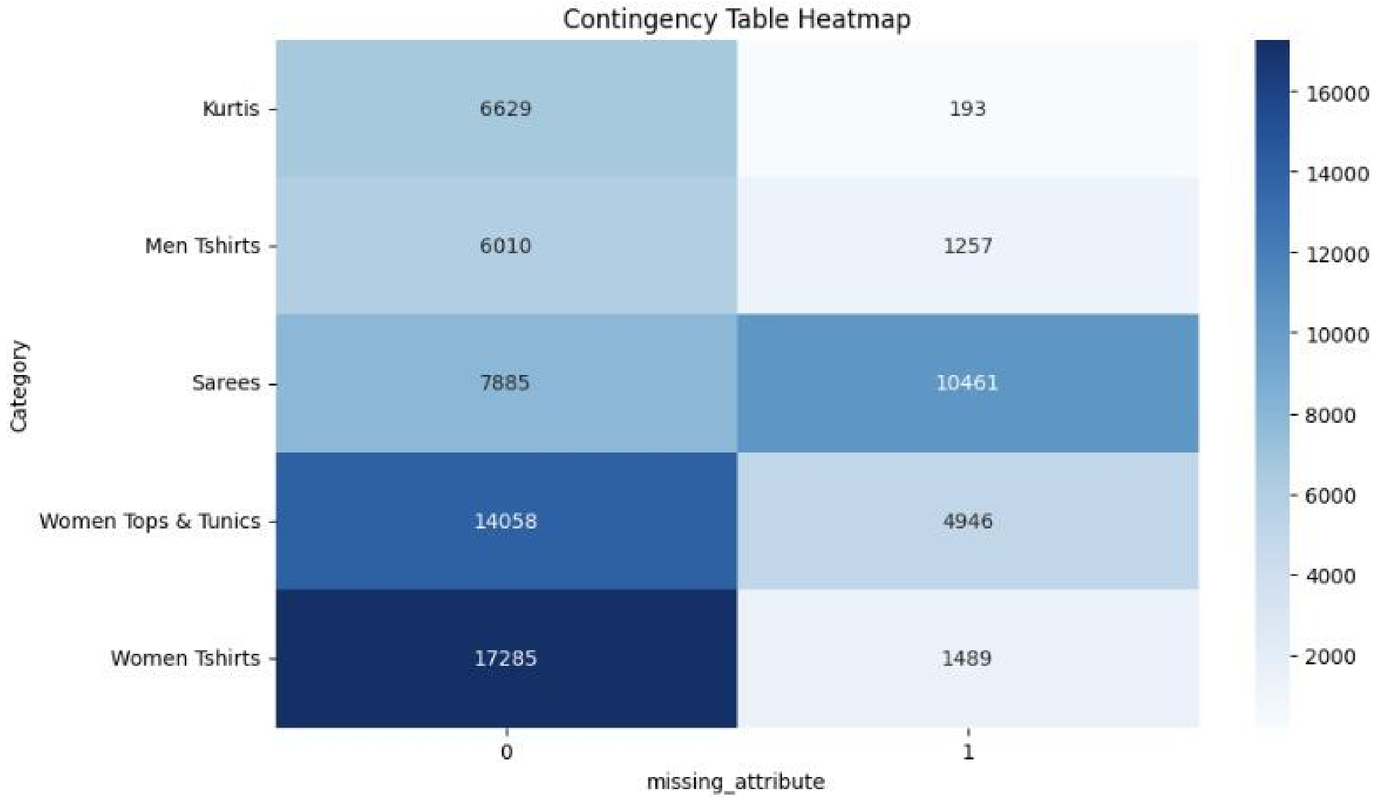
Hypotheses 3 - Hypothesis on Completeness of Attribute Data by Category

Null Hypothesis (H0):
The distribution of missing attribute values is the same across all product categories.

Alternative Hypothesis (H1):
The distribution of missing attribute values differs among product categories.

Test Method

- Chi-square statistic: 14503.9518
- P-value: 0.0000
- Reject the null hypothesis: Missing attribute values are more likely in some categories than others.



Reasons to use Chi square test:

Using a chi-square test to analyze missing attribute values is effective because it assesses the association between missingness and categorical variables, helping identify patterns in data incompleteness. This non-parametric test provides statistical significance, indicating whether observed differences in missing values across categories are likely due to chance or reflect a systematic issue. Additionally, it enables researchers to take targeted actions to improve data quality based on the identified categories with higher missingness.

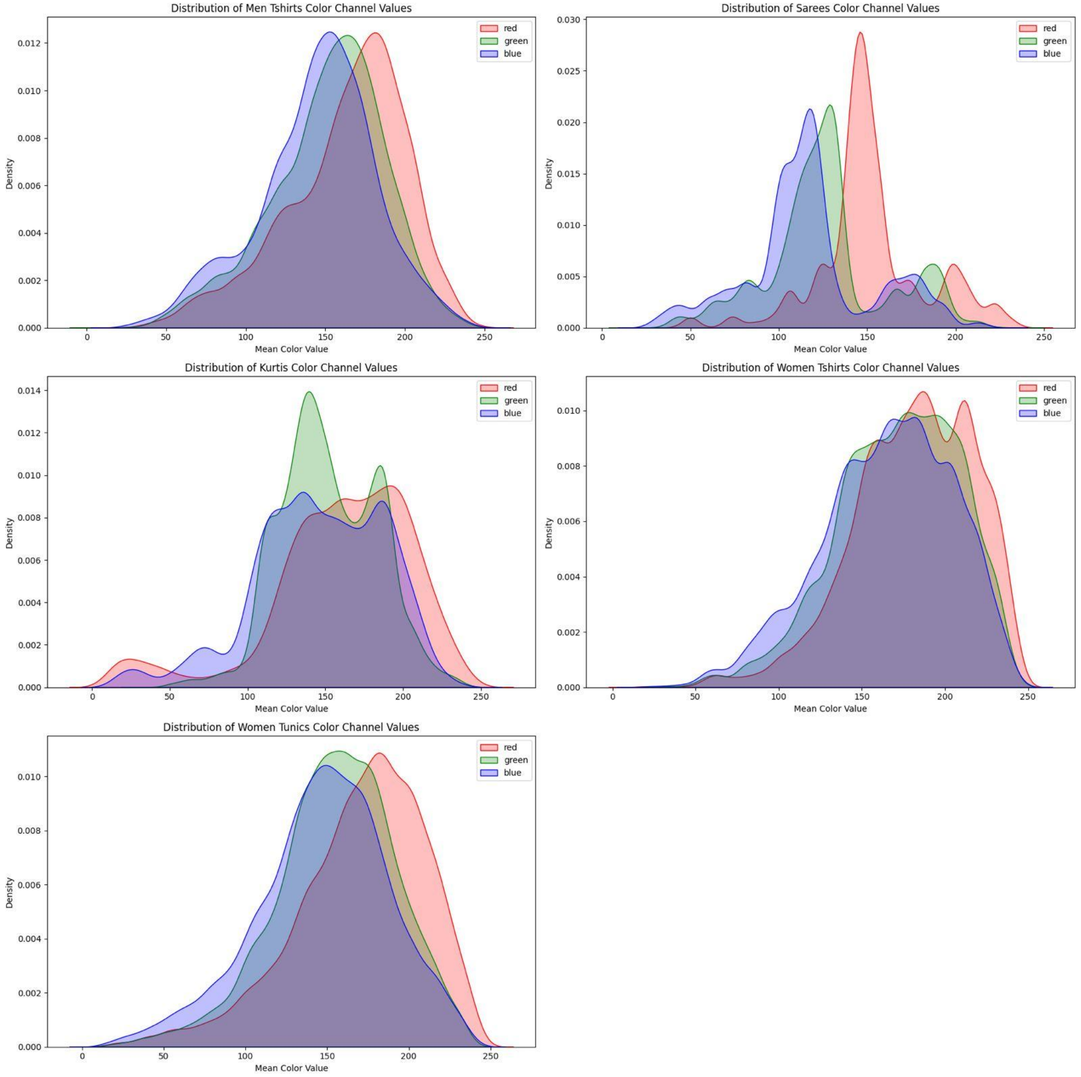
Hypotheses 4 -Color histogram across different categories

Null Hypothesis (H0):
There is no significant difference in color consistency (as measured by histograms of RGB values) across different categories.

Alternative Hypothesis (H1):
There is a significant difference in color consistency across different color categories.

Test Method

- Statistical Test: Kruskal-Wallis H-test
- Same as before for different color attribute, now we do for different categories



If the p-value from the Kruskal-Wallis H-test is less than 0.05:
Conclusion:

- Reject the null hypothesis (H0). This indicates that there is a statistically significant difference in color consistency across the categories you are comparing. In practical terms, this means that at least one of the color attributes has a different distribution of color consistency than the others.

If the p-value is greater than 0.05:
Conclusion:

- Fail to reject the null hypothesis (H0). This suggests that there is no significant difference in color consistency across the categories, implying that the color distributions are similar for the attributes considered.

```
Kruskal-Wallis H-test results for red values: H-statistic = 7241.353359852603, p-value = 0.0
Kruskal-Wallis H-test results for green values: H-statistic = 13353.422911708638, p-value = 0.0
Kruskal-Wallis H-test results for blue values: H-statistic = 13112.62089601969, p-value = 0.0
The red color distributions are significantly different across categories.
The green color distributions are significantly different across categories.
The blue color distributions are significantly different across categories.

 ____________________________________________________


At least one of the color channels has significantly different distributions across categories.
Rejection of the null hypothesis: The color distributions are not the same across categories.
We can use the color distributions as features for classification.
```

ML Modellling

We performed a variety of machine learning models which included **Random Forest, Multi-Layer Perceptron, Convolutional Neural Network** for classification and further analysis.

ML Models Used for Classification

Random Forest (RF):

Why Is It Used: Good for baseline performance on structured data. Handles overfitting using ensemble learning.

Running Time: Fast on structured datasets due to parallel tree computations. However taking into account the time taken for feature extraction makes it relatively slower than CNN.

Multi-Layer Perceptron (MLP):

Why Is It Used: Capable of learning non-linear relationships between image features.

Running Time: Moderate, as it processes flattened pixel values. However feature extractions slow down the process making this slower than CNN as well.

Convolutional Neural Network (CNN):

Why Is It Used: Specialized for image data, capturing spatial and hierarchical features.

Running Time: Slower due to higher computational complexity and parameter tuning.

Features and Scaling

Hog , Gabor, Edge:

- HOG Computes gradient orientations in 8 bins with L2-Hys normalization to robustly detect shapes and resist lighting variations.
- Gabor Filter Analysis: Captures fine and broad texture patterns using multi-scale filters with four frequencies and six orientations.
- Edge Feature Detection: Extracts structural elements with Canny edge detection and summarizes them in a normalized 16-bin histogram.
- All these features totaled to **14,176** features for a single **128x128** image with 3 color channels

Principal Component Analysis:

- PCA or Principal Component Analysis was used to reduce the dimensionality of data while preserving important features.
- Used Incremental PCA by scikit-learn, reducing the number of final features to **100** principal components to capture roughly 72% of the data variance (we were restricted by the computational cost).

*this feature set was used for MLP and RF models

ML Models Used for Classification

Random Forest (RF):
Undersampled Data (5k train, 2.5k test)

Model Performance:

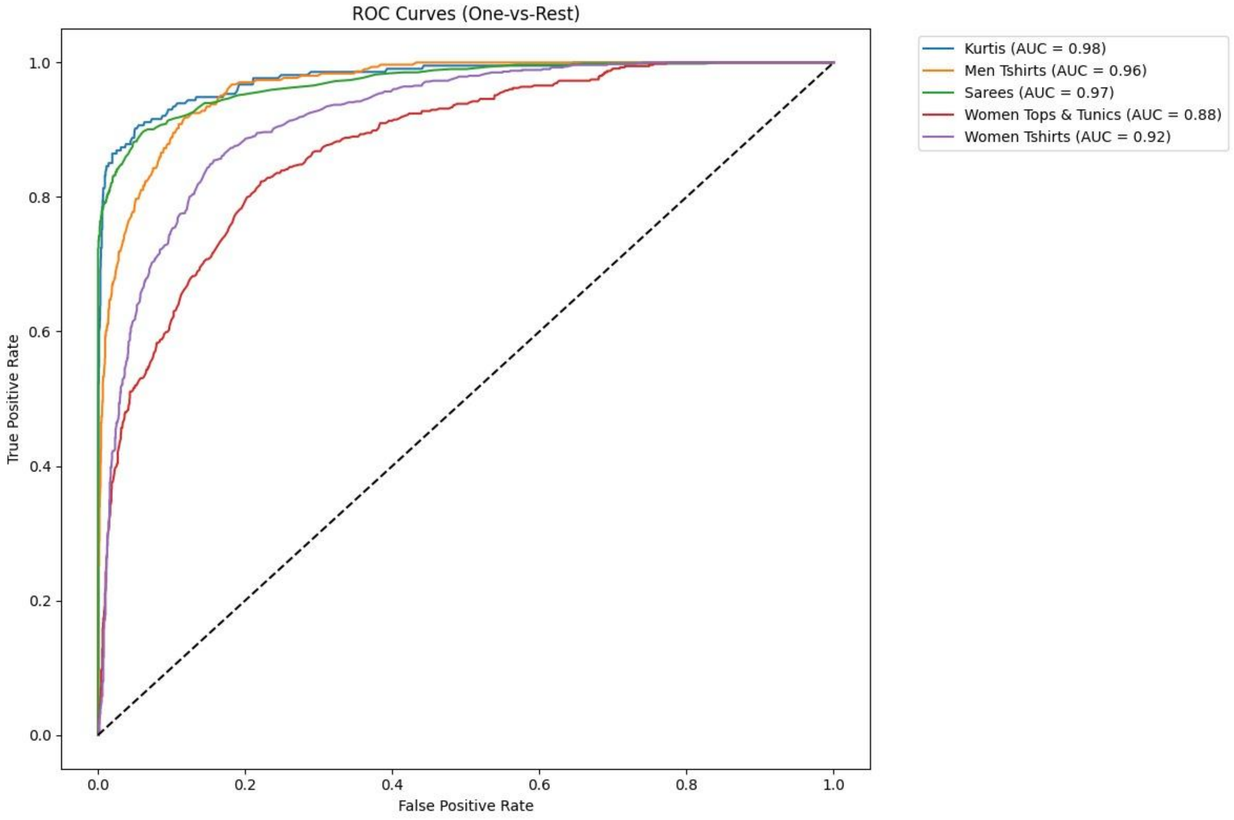
	precision	recall	f1-score	support
Kurtis	0.80	0.86	0.83	214
Men Tshirts	0.71	0.73	0.72	306
Sarees	1.00	0.70	0.82	614
Women Tops & Tunics	0.53	0.75	0.62	554
Women Tshirts	0.77	0.71	0.74	812
accuracy			0.73	2500
macro avg	0.76	0.75	0.75	2500
weighted avg	0.77	0.73	0.74	2500

Classification report for under sampled data

ACCURACY = 73%

Parameters for Random Forest Classifier

```
rf = RandomForestClassifier(
    n_estimators=100,
    max_depth=None,
    n_jobs=-1,
    random_state=42,
    class_weight='balanced'
)
```



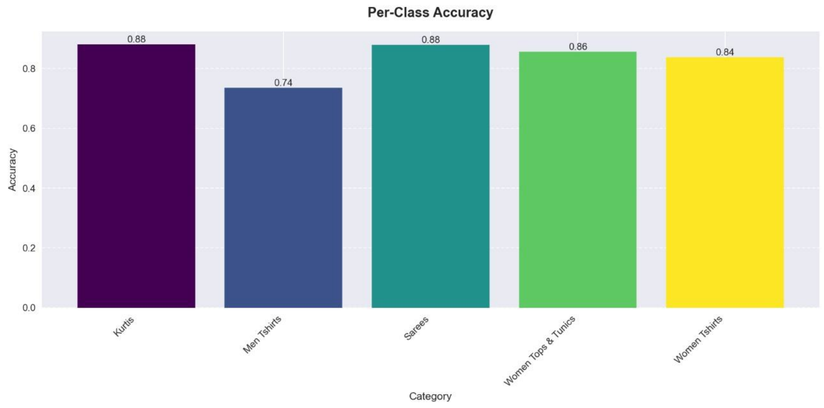
ML Models Used for Classification

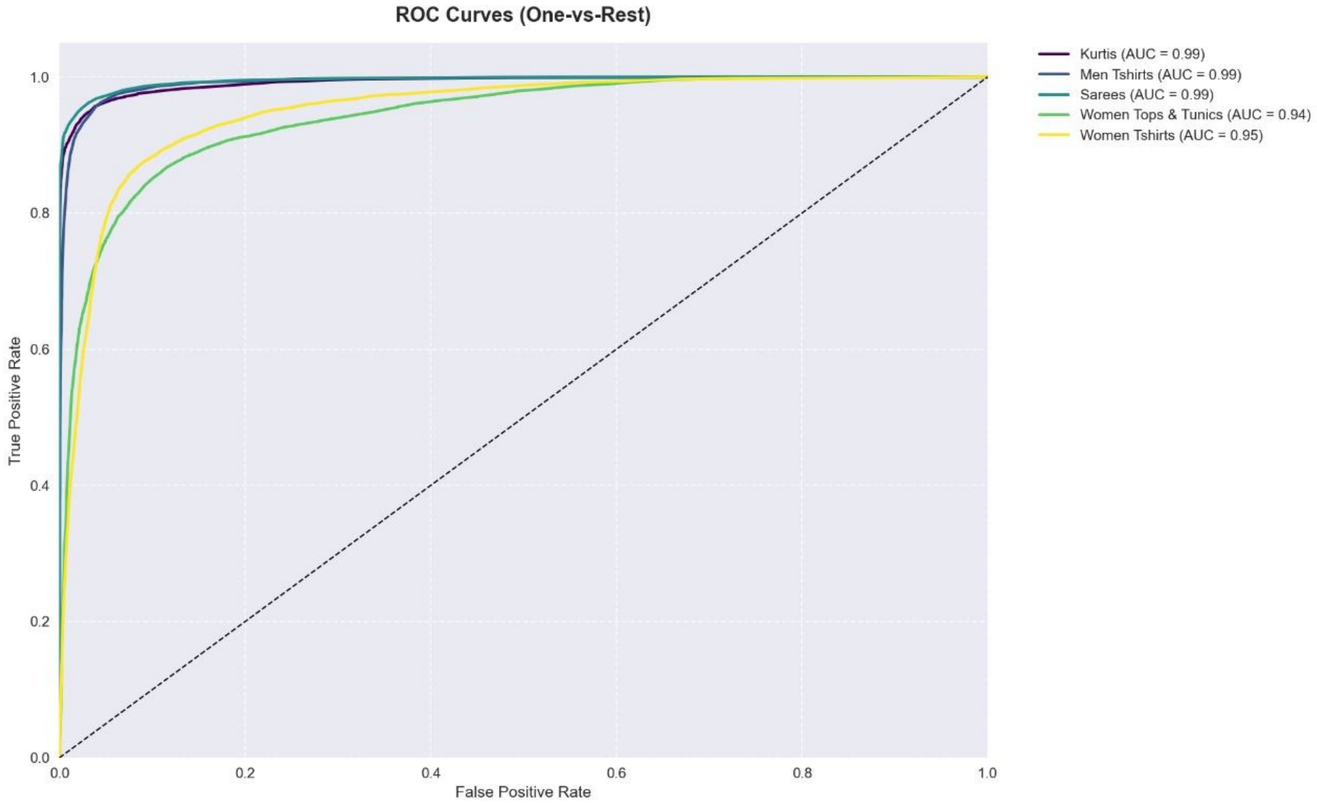
Random Forest (RF):
Full Dataset

Model Performance:

	precision	recall	f1-score	support
Kurtis	0.95	0.88	0.92	2460
Men Tshirts	0.97	0.74	0.84	3787
Sarees	0.99	0.88	0.93	7102
Women Tops & Tunics	0.68	0.86	0.76	6925
Women Tshirts	0.83	0.84	0.83	9931
accuracy			0.84	30205
macro avg	0.88	0.84	0.86	30205
weighted avg	0.86	0.84	0.85	30205

Classification report for full dataset





ACCURACY = 84%

ML Models Used for Classification

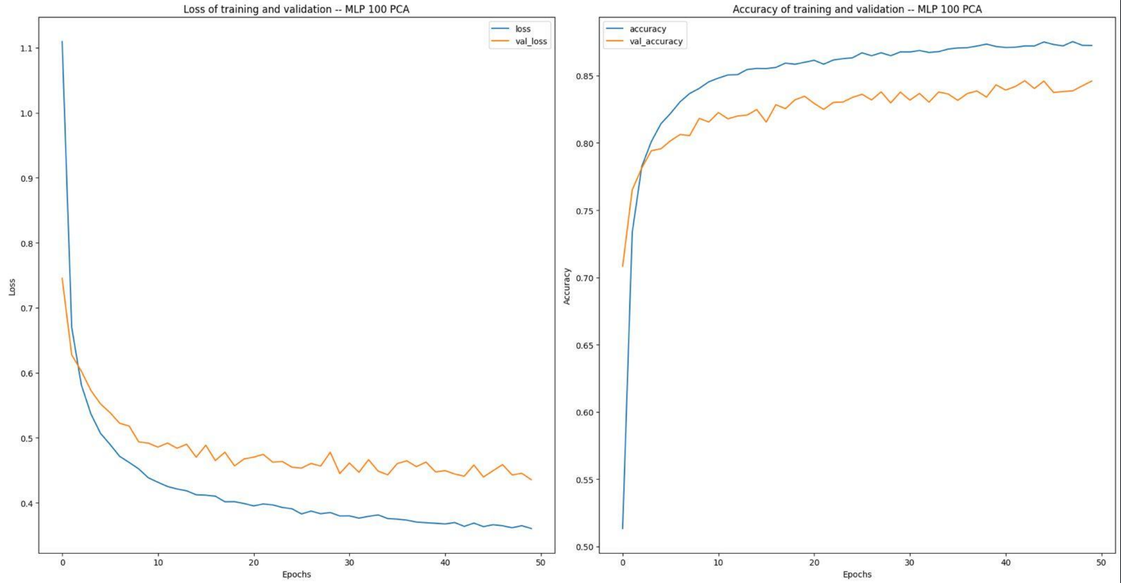
Multi-Layer Perceptron (MLP):
Full Dataset

```
Accuracy: 0.8459526568448932
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.91      0.91      2460
           1       0.94      0.75      0.84      3787
           2       0.99      0.89      0.94      7102
           3       0.74      0.78      0.76      6925
           4       0.79      0.88      0.83      9931

    accuracy                           0.85     30205
   macro avg       0.87      0.84      0.85     30205
weighted avg       0.86      0.85      0.85     30205
```

Classification report for full dataset



Parameters for Multi-Layer Perceptron

ACCURACY = 85%

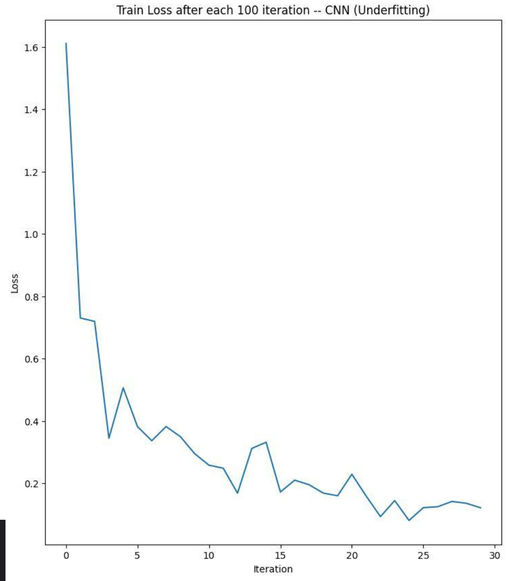
ML Models Used for Classification

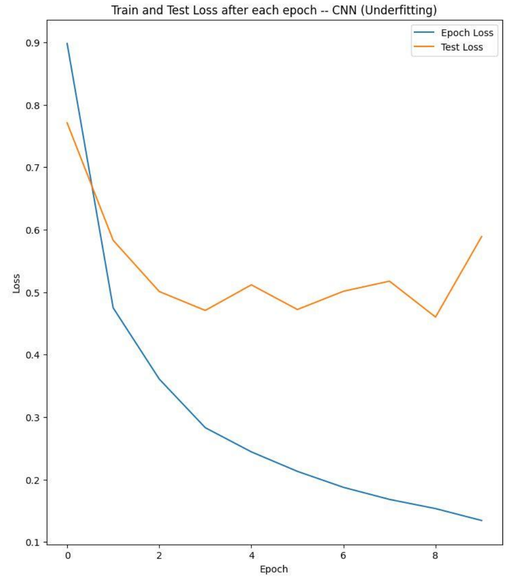
Convolutional Neural Network (CNN):
Undersampled Data (25k train, 10k test)

0.8058

	precision	recall	f1-score	support
Kurtis	0.70	0.95	0.81	813
Men Tshirts	0.76	0.91	0.83	1280
Sarees	0.99	0.88	0.93	2316
Women Tops & Tunics	0.68	0.81	0.74	2338
Women Tshirts	0.87	0.67	0.76	3253
accuracy			0.81	10000
macro avg	0.80	0.84	0.81	10000
weighted avg	0.82	0.81	0.81	10000

Classification report for under sampled data





Parameters for Convolutional Neural Network

```python
class CategoryClassifier(nn.Module):
    def __init__(self, num_classes=5):
        super(CategoryClassifier, self).__init__()

        self.model = nn.Sequential(
            nn.Conv2d(3, 32, kernel_size=3, padding=1, stride=2),     # 32x64x64
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2),                    # 32x32x32

            nn.Conv2d(32, 64, kernel_size=3, padding=1, stride=2),    # 64x16x16
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2),                    # 64x8x8

            nn.Flatten(),

            nn.Linear(64*8*8, 512),                                   # 512
            nn.ReLU(),
            nn.Dropout(0.5),

            nn.Linear(512, 64),                                       # 64
            nn.ReLU(),
            nn.Dropout(0.5),

            nn.Linear(64, num_classes)                                # 5
        )
```

ACCURACY = 81%

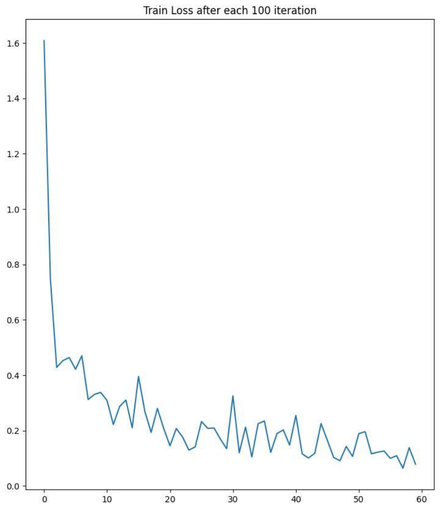
ML Models Used for Classification

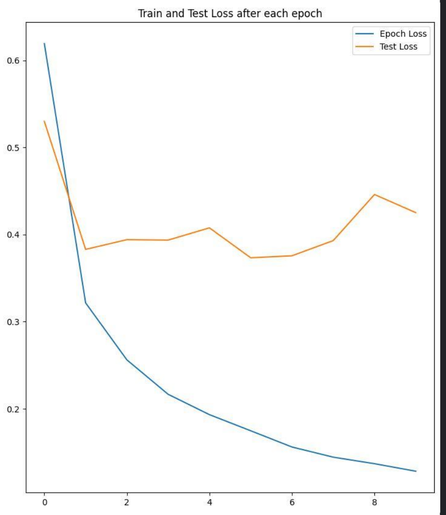
Convolutional Neural Network (CNN): Full Dataset

0.8822380400595928

	precision	recall	f1-score	support
Kurtis	0.90	0.91	0.90	2460
Men Tshirts	0.93	0.88	0.90	3787
Sarees	0.98	0.95	0.96	7102
Women Tops & Tunics	0.78	0.84	0.81	6925
Women Tshirts	0.87	0.86	0.86	9931
accuracy			0.88	30205
macro avg	0.89	0.89	0.89	30205
weighted avg	0.88	0.88	0.88	30205

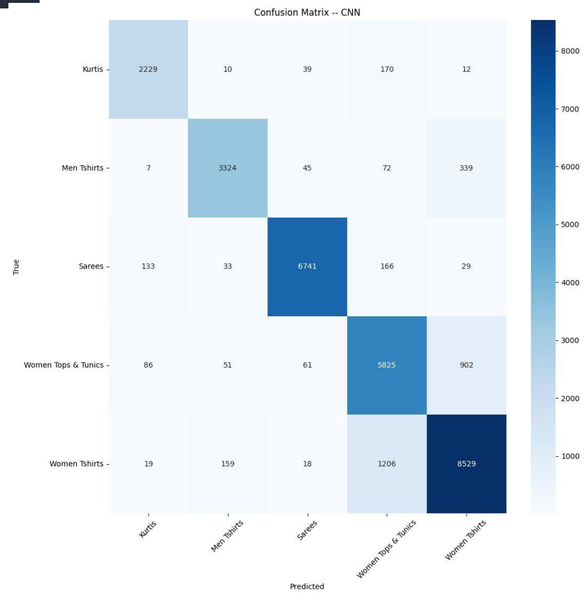
Classification report for under sampled data





ACCURACY = 88%

Confusion Matrix for Convolutional Neural Network



Conclusion and Futurework

Conclusion:

- Conducted hypotheses testing to analyse our dataset.
- Conducted preprocessing steps like scaling, resizing and feature extraction.
- Successfully implemented models (RF, MLP, CNN) to classify images into five categories: Kurti, Men's shirt, Saree, Women's top/tunic, and Women's t-shirt.

Future Work:

- Extend to multi-label classification for attributes like color, pattern, and sleeve length.
- Enable real-time product search across multiple e-commerce platforms for enhanced shopping experiences.
- Adopt distributed training techniques for faster model scaling on extensive datasets.
- Deploy the model for automated real-time product cataloging.

REFS

https://docs.scipy.org/doc/scipy/reference/stats.html
https://www.medcalc.org/manual/statistical-tables.php
https://scikit-learn.org/1.5/index.html
https://pytorch.org/docs/stable/index.html
https://www.tensorflow.org/guide/keras

WHY?

CSE558 Data Science Group Project

TEAM

Saksham Singh (2022434)
Swarnima Prasad (2022525)
Ritika Thakur (2022468)
Sidhartha Garg (2022499)

Thank You

:)