



King Saud University
College of Computer and Information Science
Department of Information Technology
IT 326 – Data Mining
Semester 1444 – 1445 H

Hair Loss

Section	NAME	ID
74557	Hissah Alhano	443200617
	Ghadah suod	443200501
	Basmah Alrashid	442202996

1 Problem :

On a broad scale, there is significant interest in hair care worldwide, with an increasing awareness of the importance of hair health and the use of suitable products and treatments. Analyzing hair health provides scientific and practical insights into hair care and how to maintain its health and prevent and control many diseases and conditions resulting from poor hair care habits. That's why in our project we chose this dataset to assist in addressing hair health problems by analyzing people's data that will help us in identifying factors that lead to hair loss and disease in order to help many people to take proper hair care habits.

2 Data Mining Task :

In our project, we'll employ two data mining techniques to aid in forecasting the likelihood of genetic hair loss: classification and clustering. In classification, the model will be trained to determine whether an individual is predisposed to genetic hair loss by assigning a genetic factor as a class label, considering various attributes like hormonal changes, age, nutritional deficiencies, stress levels, etc. In clustering, the model will group individuals with similar hair characteristics into clusters, which will then be utilized to predict the outcomes for new individuals.

3 Data :

The source: <https://www.kaggle.com/datasets/amitvkulkarni/hair-health>

Number of objects: 1000 objects.

Number of attributes: 11.

- Attribute characteristics:

Attribute	Data type
id	Nominal
Genetics	Binary
Age	Numeric (integer)
Hormonal Changes	Binary
Medical Conditions	Categorical
Medications & Treatments	Categorical
Nutritional Deficiencies	Categorical
Stress	Categorical
Poor Hair Care Habit	Binary
Environmental Factors	Binary
Hair Loss	Binary

- Missing values:

```
##show & check missing values

df.replace("No Data", np.nan, inplace=True)

missing_values=df.isna().sum()

print("Missing Values in each column:")
print(missing_values)
```

```
Missing Values in each column:
Id                                0
Genetics                         0
Hormonal Changes                 0
Medical Conditions              110
Medications & Treatments         2
NutritionalDeficien             80
Stress                           0
Age                              0
Poor Hair Care                  0
Environmental Factors            0
Smoking                         0
WeightLoss                      0
Hair Loss                       0
dtype: int64
```

The missing data in our dataset labeled with "No Data", we replace it with nan to detectate the missing value.

We figured out that we have 192 missing value. 110 in "Medical Conditions", 2 "Medications & Treatments", 80 in "NutritionalDeficien"

- Statistical Measures:

```
: #Show the Min., 1st Qu., Median, Mean ,3rd Qu.,Max. for each numeric column:  
df.describe()
```

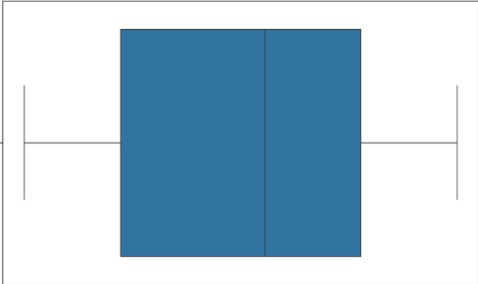
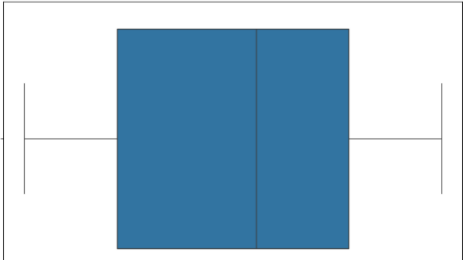
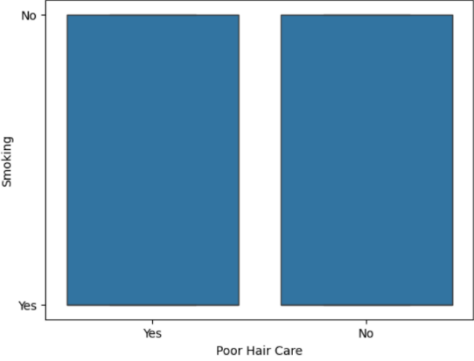
:

	Id	Age	Hair Loss
count	999.000000	999.000000	999.000000
mean	153354.673674	34.188188	0.497497
std	25516.041985	9.377980	0.500244
min	110003.000000	18.000000	0.000000
25%	131867.500000	26.000000	0.000000
50%	152951.000000	34.000000	0.000000
75%	174969.000000	42.000000	1.000000
max	199949.000000	50.000000	1.000000

- Outliers:

```
Outlier Counts:  
Age: 0 rows with outliers  
Hair Loss: 0 rows with outliers  
Total Rows with Outliers: 0
```

• Boxplot:

Graph		Description
<div><pre>axesSubplot(xlabel="Medical Conditions")</pre><p>A boxplot showing the distribution of various medical conditions. The x-axis is labeled "Medical Conditions" and lists: Eczema, Dermatitis, Ringworm, Psoriasis, Alopecia, Androp, Infest, Rheic, Derm, Allerg, Thyroid, Pro, Genetic, Alopecia. The y-axis represents the frequency of each condition. The plot shows a single box with a median line, indicating the central tendency and spread of the data.</p></div>		Boxplots provide a concise summary of the distribution, central tendency and checking data outliers
<div><pre>plt.figure(figsize=(10, 6)) sns.boxplot(x="Medications & Treatments", data=df) plt.show()</pre><p>A boxplot showing the distribution of various medications and treatments. The x-axis is labeled "Medications & Treatments" and lists: Antibiot, Antifungal, Cream, Cutane, Chemotherapy, Steroids, P, Pain, Pressure, Medication, Corticosteroids, Prescribed, Medication. The y-axis represents the frequency of each medication or treatment. The plot shows a single box with a median line, indicating the central tendency and spread of the data.</p></div>		Boxplots provide a concise summary of the distribution, central tendency and checking data outliers
<div><p>A histogram showing the distribution of poor hair care observed within the smoking. The x-axis is labeled "Poor Hair Care" and has two categories: "Yes" and "No". The y-axis is labeled "Smoking" and has two categories: "Yes" and "No". The plot shows two bars, one for "Yes" and one for "No", indicating the frequency of poor hair care for each smoking status.</p></div>		The histogram represents the poor hair care observed within the smoking. -Through the histogram, it was found that most of the people smoking they have poor hair care

• Plotting Methods:

Graph	Description																														
<table><caption>Bar Chart (Medical Conditions)</caption><thead><tr><th>Medical Conditions</th><th>Yes (Count)</th><th>No (Count)</th></tr></thead><tbody><tr><td>Eczema</td><td>35</td><td>33</td></tr><tr><td>Dermatitis</td><td>45</td><td>42</td></tr><tr><td>Psoriasis</td><td>35</td><td>32</td></tr><tr><td>Alopecia Areata</td><td>58</td><td>42</td></tr><tr><td>Scaly Infection</td><td>53</td><td>54</td></tr><tr><td>Ichthyosis</td><td>42</td><td>38</td></tr><tr><td>Dermatitis</td><td>48</td><td>45</td></tr><tr><td>Thyroid Problems</td><td>48</td><td>51</td></tr><tr><td>Autoimmune Disease</td><td>48</td><td>50</td></tr></tbody></table>	Medical Conditions	Yes (Count)	No (Count)	Eczema	35	33	Dermatitis	45	42	Psoriasis	35	32	Alopecia Areata	58	42	Scaly Infection	53	54	Ichthyosis	42	38	Dermatitis	48	45	Thyroid Problems	48	51	Autoimmune Disease	48	50	The bar chart shows the distribution of genes across different medical conditions in the dataset and allows for a quick comparison of the spread of baldness genes among individuals with various medical conditions.
Medical Conditions	Yes (Count)	No (Count)																													
Eczema	35	33																													
Dermatitis	45	42																													
Psoriasis	35	32																													
Alopecia Areata	58	42																													
Scaly Infection	53	54																													
Ichthyosis	42	38																													
Dermatitis	48	45																													
Thyroid Problems	48	51																													
Autoimmune Disease	48	50																													
<table><caption>Histogram</caption><thead><tr><th>Age</th><th>Count</th></tr></thead><tbody><tr><td>20</td><td>45</td></tr><tr><td>25</td><td>42</td></tr><tr><td>30</td><td>48</td></tr><tr><td>35</td><td>45</td></tr><tr><td>40</td><td>48</td></tr><tr><td>45</td><td>45</td></tr><tr><td>50</td><td>42</td></tr></tbody></table>	Age	Count	20	45	25	42	30	48	35	45	40	48	45	45	50	42	The histogram represents the frequency of age in the dataset and the hair loss influence. After observation, we noticed that the Age influence on hair loss.														
Age	Count																														
20	45																														
25	42																														
30	48																														
35	45																														
40	48																														
45	45																														
50	42																														
<table><caption>Pie Chart</caption><thead><tr><th>Label</th><th>Percentage</th></tr></thead><tbody><tr><td>0</td><td>50.3%</td></tr><tr><td>1</td><td>49.7%</td></tr></tbody></table>	Label	Percentage	0	50.3%	1	49.7%	The pie chart shows that the data are approximately equally distributed between (49.7% they have, 50.3 don't have) based on class label (Hair loss). No cleaning needed since there are no missing values.																								
Label	Percentage																														
0	50.3%																														
1	49.7%																														

4 Data preprocessing :

Data preprocessing is crucial in data science and machine learning, refining raw data for accurate analysis and modeling. It improves data quality, enables feature engineering, optimizes algorithm performance through normalization and standardization, and refines datasets by handling missing data and reducing dimensionality. Ultimately, it ensures accurate, robust models capable of generalizing well to new data.

- Row dataset

		Id	Genetics	Hormonal Changes	Medical Conditions	Medications & Treatments	Nutritional Deficiencies	Stress	Age	Poor Hair Care	Environmental
1		133992	Yes	No	No Data	No Data	Magnesium deficiency	Moderate	19	Yes	
2		148393	No	No	Eczema	Antibiotics	Magnesium deficiency	High	43	Yes	
3		155074	No	No	Dermatosis	Antifungal Cream	Protein deficiency	Moderate	26	Yes	
4		118261	Yes	Yes	Ringworm	Antibiotics	Biotin Deficiency	Moderate	46	Yes	
5		111915	No	No	Psoriasis	Accutane	Iron deficiency	Moderate	30	No	
6		139061	Yes	No	Psoriasis	Antibiotics	Magnesium deficiency	Low	37	No	
7		189255	Yes	Yes	No Data	No Data	Selenium deficiency	High	40	Yes	
8		112532	Yes	No	Dermatosis	Chemotherapy	Omega-3 fatty acids	High	35	Yes	
9		140785	Yes	No	Eczema	Stilacids	Selenium deficiency	Moderate	19	No	
10		187569	No	Yes	Ringworm	Rogaine	Magnesium deficiency	Moderate	49	Yes	
11		119858	Yes	Yes	Eczema	Good Pressure Medication	Biotin Deficiency	High	26	Yes	
12		159198	No	Yes	Alopecia Areata	Accutane	Zinc Deficiency	High	48	No	
13		150086	Yes	Yes	Scalp Infection	Immunomodulators	Biotin Deficiency	Moderate	20	No	
14		178256	No	No	Psoriasis	Antibiotics	Vitamin A Deficiency	High	30	Yes	
15		150104	Yes	No	Eczema	Antibiotics	Biotin Deficiency	High	34	Yes	
16		130552	Yes	Yes	Scalp Infection	Rogaine	Vitamin D Deficiency	Moderate	29	Yes	
17		116190	Yes	No	Seborrheic Dermatitis	Antidepressants	Vitamin D Deficiency	High	46	Yes	
18		181441	No	Yes	Dermatosis	Antibiotics	Zinc Deficiency	Low	19	Yes	
19		147404	Yes	Yes	Dermatosis	Accutane	Biotin Deficiency	Low	26	No	
20		136709	Yes	Yes	Seborrheic Dermatitis	Chemotherapy	Vitamin A Deficiency	High	46	Yes	
21		187362	Yes	Yes	Seborrheic Dermatitis	Accutane	Protein deficiency	High	46	No	
22		135894	No	No	No Data	Chemotherapy	Zinc Deficiency	High	20	No	
23		148874	Yes	Yes	Psoriasis	Antibiotics	Vitamin D Deficiency	Low	29	No	
24		116818	No	Yes	Scalp Infection	Antidepressants	Vitamin D Deficiency	High	37	No	
25		112062	No	Yes	Dermatitis	Antibiotics	Protein deficiency	High	33	Yes	
26		147833	Yes	Yes	Dermatosis	Heart Medication	Vitamin A Deficiency	Moderate	34	No	
27		190067	No	Yes	Dermatosis	Immunomodulators	Selenium deficiency	High	28	No	
28		114579	No	Yes	Seborrheic Dermatitis	Antifungal Cream	Zinc Deficiency	Low	41	No	

1. Encoding

Code:

```
## Encoding catogrical data
from sklearn.preprocessing import LabelEncoder
import pandas as pd
from scipy import stats

data1 = pd.read_csv("Data_cleand.csv")

le = LabelEncoder()
data1['Genetics'] = le.fit_transform(data1['Genetics'])

data1['Hormonal Changes'] = le.fit_transform(data1['Hormonal Changes'])

data1['Poor Hair Care'] = le.fit_transform(data1['Poor Hair Care'])
data1['WeightLoss'] = le.fit_transform(data1['WeightLoss'])

data1['Environmental Factors'] = le.fit_transform(data1['Environmental Factors'])

data1['Smoking'] = le.fit_transform(data1['Smoking'])

data1['Medical Conditions'] = le.fit_transform(data1['Medical Conditions'])

data1['Medications & Treatments'] = le.fit_transform(data1['Medications & Treatments'])

data1['NutritionalDeficien'] = le.fit_transform(data1['NutritionalDeficien'])

data1['Stress'] = le.fit_transform(data1['Stress'])

print(data1)

data1.to_csv('Data_cleand.csv', index=False) #to save file after encoding .
```

For all (nominal) attribute.

		id	Genetics	Hormonal Changes	Medical Conditions	Medications & Treatments	NutritionalDeficien	Stress	Age	Poor Hair Care	Environment
1		148393	0	0	15	1	12	0	43.0	1	
2		155074	0	0	14	14	15	2	28.0	1	
3		118261	1	1	17	1	0	2	46.0	1	
4		111915	0	0	16	0	1	2	30.0	0	
5		139661	1	0	16	1	12	1	37.0	0	
6		112032	1	0	14	16	14	0	35.0	1	
7		140785	1	0	15	20	16	2	19.0	0	
8		167999	0	1	17	19	12	2	49.0	1	
9		118858	1	1	15	15	0	0	26.0	1	
10		159158	0	1	0	0	20	0	48.0	0	
11		156086	1	1	18	18	0	2	20.0	0	
12		178256	0	0	16	1	17	0	30.0	1	
13		150154	1	0	15	1	0	0	34.0	1	
14		130552	1	1	18	19	18	2	29.0	1	
15		116190	1	0	19	12	18	0	46.0	1	
16		194441	0	1	14	1	20	1	19.0	1	
17		147404	1	1	14	0	0	1	26.0	0	
18		136709	1	1	19	16	17	0	46.0	1	
19		167362	1	1	19	0	15	0	46.0	0	
20		148974	1	1	16	1	18	1	29.0	0	
21		116818	0	1	18	12	18	0	37.0	0	
22		142062	0	1	12	1	15	0	33.0	1	
23		147833	1	1	14	17	17	2	34.0	0	
24		190967	0	1	14	16	16	0	28.0	0	
25		114579	0	1	19	14	20	1	41.0	0	
26		159949	0	1	20	16	14	1	35.0	1	
27		133091	0	1	12	1	1	2	45.0	0	
28		157912	1	0	12	20	15	2	30.0	1	

Description: Encoding is essential in data analysis for two main reasons. Firstly, it enables the handling of categorical data by converting it into numerical format, allowing algorithms to process it effectively. Secondly, encoded data facilitates comparison and statistical computation across different categories, streamlining data analysis. Overall, encoding ensures compatibility with analysis techniques, improves model performance, and enhances data integrity.

Data transformation:

2. Discretization:

Code:

```
[11]: # Discretization
column_to_discretize= 'Age'
num_bins=3
data1['discretized_' + column_to_discretize] = pd.cut(data1[column_to_discretize], bins=num_bins, labels=False)
print("Original DataFrame:")
print(data1[['Age', 'discretized_Age']])

data1.to_csv('data_discretize.csv', index=False) #to save file after discretize .
```

Here we discretize (Age) attribute.

Data before the discretization:

	Id	Genetics	Hormonal Changes	Medical Conditions	Medications & Treatments	Nutritional Deficien	Stress	Age	Poor Hair Care	Environment
1	168393	0	0	15	1	12	0	43.0	1	1
2	155074	0	0	14	14	15	2	26.0	1	1
3	118261	1	1	17	1	0	2	46.0	1	1
4	111915	0	0	16	0	1	2	30.0	0	0
5	139661	1	0	16	1	12	1	37.0	0	0
6	112032	1	0	14	16	14	0	35.0	1	1
7	140785	1	0	15	20	16	2	19.0	0	0
8	167999	0	1	17	19	12	2	49.0	1	1
9	118856	1	1	15	15	0	0	26.0	1	1
10	159158	0	1	0	0	20	0	48.0	0	0
11	155086	1	1	16	18	0	2	23.0	0	0
12	179256	0	0	16	1	17	0	30.0	1	1
13	150194	1	0	15	1	0	0	34.0	1	1
14	130552	1	1	18	19	18	2	29.0	1	1
15	116190	1	0	19	12	18	0	46.0	1	1
16	194441	0	1	14	1	20	1	19.0	1	1
17	147404	1	1	14	0	0	1	26.0	0	0
18	136709	1	1	19	16	17	0	46.0	1	1
19	187362	1	1	19	0	15	0	46.0	0	0
20	148974	1	1	16	1	18	1	29.0	0	0
21	118818	0	1	18	12	18	0	37.0	0	0
22	142062	0	1	12	1	15	0	33.0	1	1
23	147833	1	1	14	17	17	2	34.0	0	0
24	190967	0	1	14	18	16	0	28.0	0	0
25	114579	0	1	19	14	20	1	41.0	0	0
26	159949	0	1	20	16	14	1	35.0	1	1
27	133091	0	1	12	1	1	2	45.0	0	0
28	157912	1	0	12	20	15	2	30.0	1	1

Data after the discretization:

	Stress	Age	Poor Hair Care	Environmental Factors	Smoking	Weight Loss	Hair Loss	discretized_Age
1	0	43	1	1	0	0	0	2
2	2	26	1	1	0	1	0	0
3	2	46	1	1	0	0	0	2
4	2	30	0	1	1	0	1	1
5	1	37	0	1	0	1	1	1
6	0	35	1	0	1	0	0	1
7	2	19	0	0	1	1	1	0
8	2	49	1	1	1	0	0	2
9	0	26	1	1	1	0	0	0
10	0	48	0	0	0	0	1	2
11	2	20	0	1	1	0	1	0
12	0	30	1	1	1	1	0	1
13	0	34	1	1	0	1	0	1
14	2	29	1	0	0	1	0	1
15	0	48	1	1	0	1	0	2
16	1	19	1	0	0	1	1	0
17	1	26	0	0	1	0	0	0
18	0	46	1	1	0	1	1	2
19	0	46	0	1	0	1	1	2
20	1	29	0	0	1	1	0	1
21	0	37	0	0	1	0	1	1
22	0	33	1	1	0	0	0	1
23	2	34	0	0	0	0	0	1
24	0	28	0	0	0	0	0	0
25	1	41	0	0	0	0	0	2
26	1	35	1	1	1	0	0	1
27	2	45	0	0	0	1	1	2
28	2	30	1	1	0	0	0	1

Description: Discretization is essential in dataset analysis for two main reasons. Firstly, it simplifies continuous data by dividing it into distinct intervals or bins, enhancing interpretability. Secondly, it reduces sensitivity to outliers by converting continuous variables into discrete categories, making the data more robust. Overall, discretization improves data analysis by simplifying complexity and enhancing data robustness.

3. Normalization:

Code:

```
[15]: #Normalization
columns_to_normalize = ['Age']
data_to_normalize = data1[columns_to_normalize]
minmax_scaler = MinMaxScaler()
normalized_data_minmax = minmax_scaler.fit_transform(data_to_normalize)
data1[columns_to_normalize] = normalized_data_minmax
print("Min-Max scaled data (only age column):")
print(data1)

data1.to_csv('data_normalize.csv', index=False) #to save file after normalize .
Min-Max scaled data (only age column):
```

Here we discretize (Age) attribute.

Data before the normalization:

		Id	Genetics	Hormonal Changes	Medical Conditions	Medications & Treatments	Nutritional Deficien	Stress	Age	Poor Hair Care	Environmental
1		146393	0	0	15	1	12	0	43.0	1	
2		150674	0	0	14	14	15	2	26.0	1	
3		118281	1	1	17	1	0	2	46.0	1	
4		111815	0	0	16	0	1	2	30.0	0	
5		139681	1	0	16	1	12	1	37.0	0	
6		112032	1	0	14	16	14	0	35.0	1	
7		140785	1	0	15	20	16	2	19.0	0	
8		187699	0	1	17	19	12	2	49.0	1	
9		118958	1	1	15	15	0	0	26.0	1	
10		109106	0	1	0	0	20	0	48.0	0	
11		156086	1	1	18	18	0	2	20.0	0	
12		178256	0	0	16	1	17	0	30.0	1	
13		150154	1	0	15	1	0	0	34.0	1	
14		130552	1	1	18	19	18	2	29.0	1	
15		116190	1	0	19	12	18	0	46.0	1	
16		194441	0	1	14	1	20	1	19.0	1	
17		147404	1	1	14	0	0	1	26.0	0	
18		136709	1	1	19	16	17	0	46.0	1	
19		187362	1	1	19	0	15	0	46.0	0	
20		148974	1	1	16	1	18	1	29.0	0	
21		116818	0	1	18	12	18	0	37.0	0	
22		142052	0	1	12	1	15	0	33.0	1	
23		147623	1	1	14	17	17	2	34.0	0	
24		150967	0	1	14	18	10	0	28.0	0	
25		114579	0	1	19	14	20	1	41.0	0	
26		159940	0	1	20	16	14	1	35.0	1	
27		133091	0	1	12	1	1	2	45.0	0	
28		157912	1	0	12	20	15	2	30.0	1	

Data after the normalization:

	Medical Conditions	Medications & Treatments	Nutritional Deficien	Stress	Age	Poor Hair Care	Environmental Factors	Smoking	Weight Loss	Hair Loss
1	4	1	2	0	0.78125	1	1	0	0	
2	3	3	4	2	0.25	1	1	0	1	
3	6	1	0	2	0.875	1	1	0	0	
4	0	0	1	2	0.375	0	1	1	0	
5	0	1	2	1	0.59375	0	1	0	1	
6	3	0	3	0	0.53125	1	0	1	0	
7	4	9	5	2	0.03125	0	0	1	1	
8	6	8	2	2	0.96875	1	1	1	0	
9	4	4	0	0	0.25	1	1	1	0	
10	0	0	9	0	0.9375	0	0	0	0	
11	7	7	0	2	0.0625	0	1	1	0	
12	5	1	6	0	0.375	1	1	1	1	
13	4	1	0	0	0.5	1	1	0	1	
14	7	8	7	2	0.34375	1	0	0	1	
15	8	2	7	0	0.875	1	1	0	1	
16	3	1	9	1	0.03125	1	0	0	1	
17	3	0	0	1	0.25	0	0	1	0	
18	8	5	6	0	0.875	1	1	0	1	
19	8	0	4	0	0.875	0	1	0	1	
20	5	1	7	1	0.34375	0	0	1	1	
21	7	2	7	0	0.59375	0	0	1	0	
22	2	1	4	0	0.46875	1	1	0	0	
23	3	8	8	2	0.9	0	0	0	0	
24	3	7	5	0	0.3125	0	0	0	0	
25	8	3	9	1	0.71875	0	0	0	0	
26	9	5	3	1	0.53125	1	1	1	0	
27	2	1	1	2	0.84375	0	0	0	1	
28	2	9	4	2	0.375	1	1	0	0	

Description: Normalization in dataset analysis is essential for two main reasons. Firstly, it ensures features are on a similar scale, preventing dominance by larger magnitudes, crucial for algorithms like K-nearest neighbors or support vector machines. Secondly, normalization enhances model performance, especially in algorithms sensitive to feature scales such as neural networks and SVMs. Overall, normalization ensures consistent, comparable data for accurate and reliable analysis and modeling.

• Data after preprocessing:

riter:

	Id	Genetics	Hormonal Changes	Medical Conditions	Medications & Treatments	NutritionalDeficien	Stress	Age	Poor Hair Care	Environmental Factors
1	148393	0	0	4	1	2	0	0.78125	1	1
2	155074	0	0	3	3	4	2	0.25	1	1
3	118261	1	1	6	1	0	2	0.875	1	1
4	111915	0	0	5	0	1	2	0.375	0	1
5	139661	1	0	5	1	2	1	0.99375	0	1
6	112032	1	0	3	5	3	0	0.53125	1	0
7	140785	1	0	4	9	5	2	0.03125	0	0
8	187999	0	1	6	8	2	2	0.96875	1	1
9	118858	1	1	4	4	0	0	0.25	1	1
10	159158	0	1	0	0	9	0	0.9375	0	0
11	156086	1	1	7	7	0	2	0.0625	0	1
12	178256	0	0	5	1	6	0	0.375	1	1
13	150154	1	0	4	1	0	0	0.5	1	1
14	130552	1	1	7	8	7	2	0.34375	1	0
15	116190	1	0	8	2	7	0	0.875	1	1
16	194441	0	1	3	1	9	1	0.03125	1	0
17	147404	1	1	3	0	0	1	0.25	0	0
18	136709	1	1	8	5	6	0	0.875	1	1
19	187362	1	1	8	0	4	0	0.875	0	1
20	148974	1	1	5	1	7	1	0.34375	0	0
21	116818	0	1	7	2	7	0	0.59375	0	0
22	142062	0	1	2	1	4	0	0.46875	1	1
23	147833	1	1	3	6	6	2	0.5	0	0
24	190967	0	1	3	7	5	0	0.3125	0	0
25	114579	0	1	8	3	9	1	0.71875	0	0
26	159949	0	1	9	5	3	1	0.53125	1	1
27	133091	0	1	2	1	1	2	0.84375	0	0
28	157912	1	0	2	9	4	2	0.375	1	1

5 Data Mining Technique:

First, we applied both supervised and unsupervised learning to our data using classification and clustering techniques.

For Classification: Classification is a versatile data analysis technique used for predictive modeling, decision-making, and gaining insights into data patterns. It predicts new data points' categories, aids in tasks like sentiment analysis and fraud detection, and evaluates model performance with metrics like accuracy and precision. This technique includes dividing the dataset into two sets:

- Training dataset: Used for building the decision tree.
- Testing dataset: Used to evaluate the constructed model.

Lastly, to evaluate our model we measure the accuracy of the dataset using a confusion matrix.

For Clustering: Clustering uncovers hidden structures in data by identifying natural groupings based on similarity. It's used for pattern discovery, exploration, anomaly detection, and data compression. In marketing, clustering enables market segmentation for targeted marketing strategies based on customer behavior and preferences.

For clustering, since it's unsupervised learning, it doesn't use a class label for implementing the cluster thus we deleted the class label attribute "Hair Loss" and used all other attributes in cluster all of them, we need to convert nominal attribute, so we used encoding method. To implement the clusters we used the K-mean algorithm, which is an algorithm that produces K clusters, which each cluster is represented by the center point of the cluster and assigns each object to the nearest cluster, then iteratively recalculates the center, and reassigns the object until the center point of each cluster does not change that means the object in the right cluster.

For the packages we used NumPy and Pandas handle data manipulation. Scikit-learn includes KMeans for clustering and DecisionTreeClassifier for classification. StandardScaler scales features, train_test_split splits data, and sklearn.metrics evaluates models. LabelEncoder encodes variables, plot_tree visualizes trees, and

matplotlib.pyplot creates plots. Yellowbrick.cluster provides SilhouetteVisualizer for visualizing cluster quality. These tools are vital for Python's data tasks.

6 Evaluation and Comparison

- Classification [90% training, 10% test]:

Figure (1) Gini

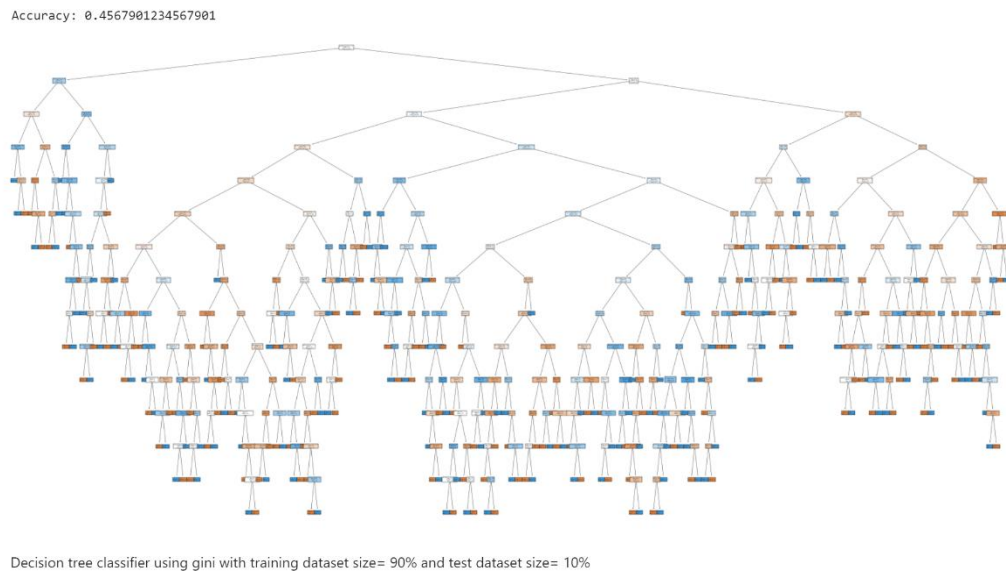


Figure (2) Entropy

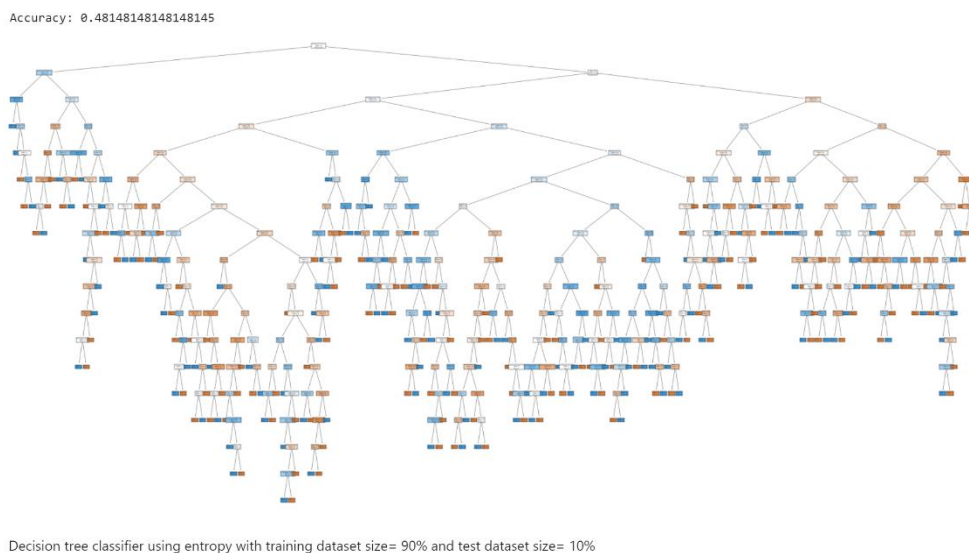
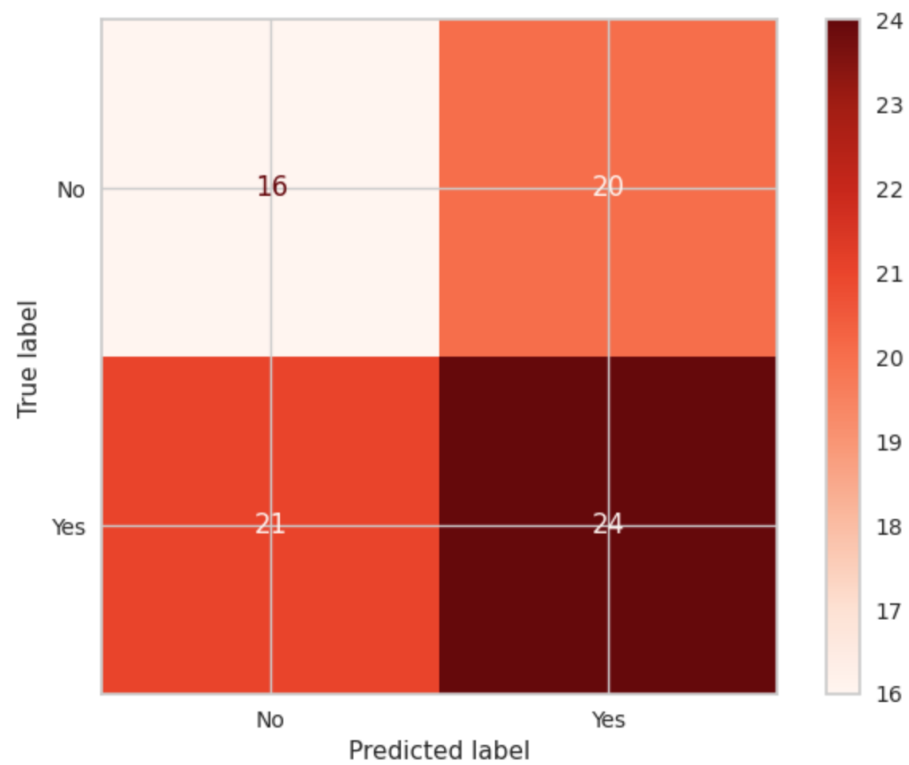


Figure (3) Confusion Matrix

Accuracy: 0.49382716049382713
[[16 20]
[21 24]]



Accuracy of prediction with test size 0.1 lies between 45%-49%

Rules extracted from the decision tree:

Classification [80% training, 20% test]:

Figure (1) Gini

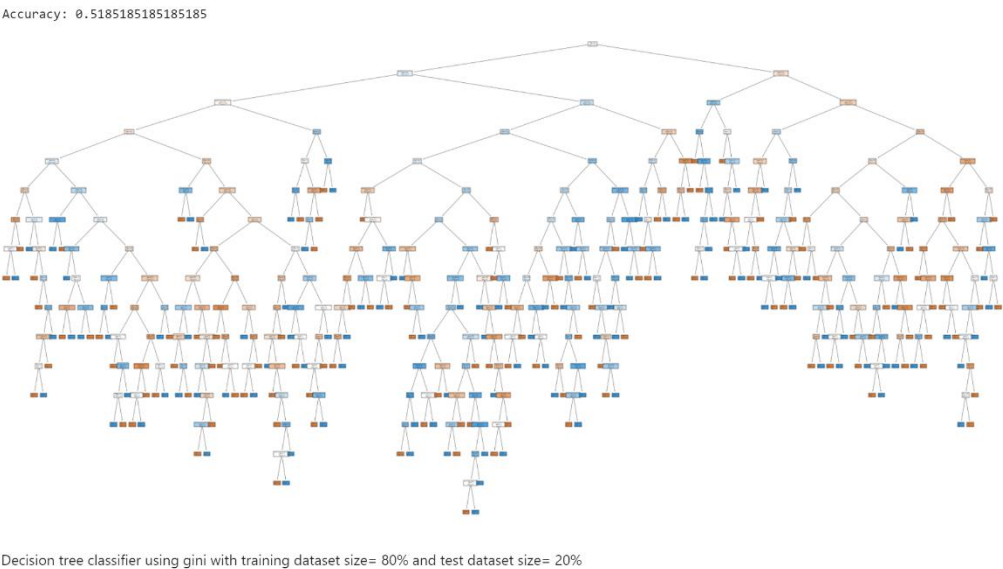


Figure (2) Entropy

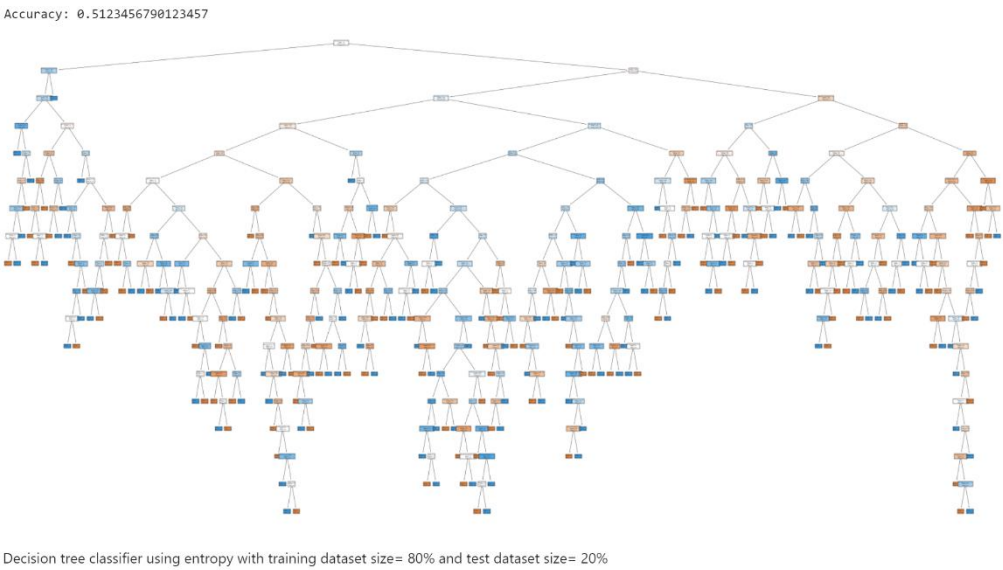
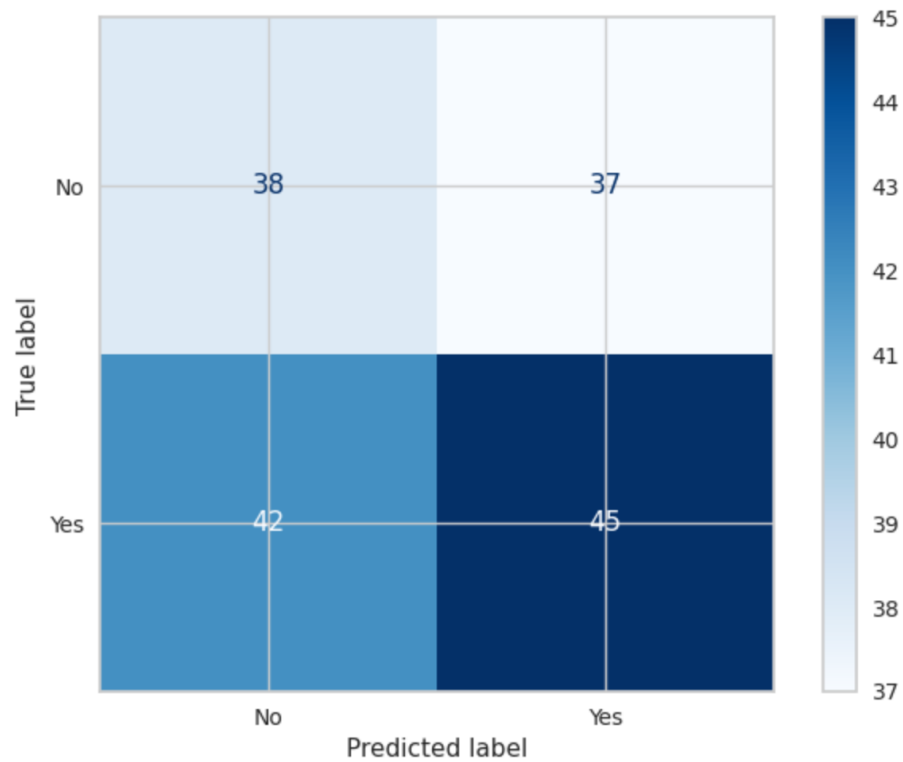


Figure (3) Confusion Matrix

Accuracy: 0.5123456790123457
[[38 37]
[42 45]]



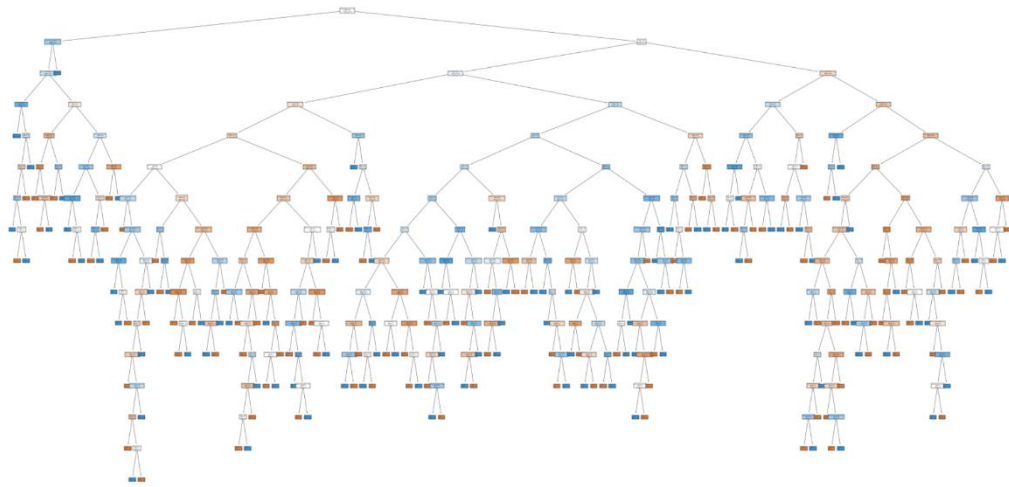
Accuracy of prediction with test size 0.2 lies between 47%-51%

Rules extracted from the decision tree:

- Classification [70% training, 30% test]:

Figure (1) Gini

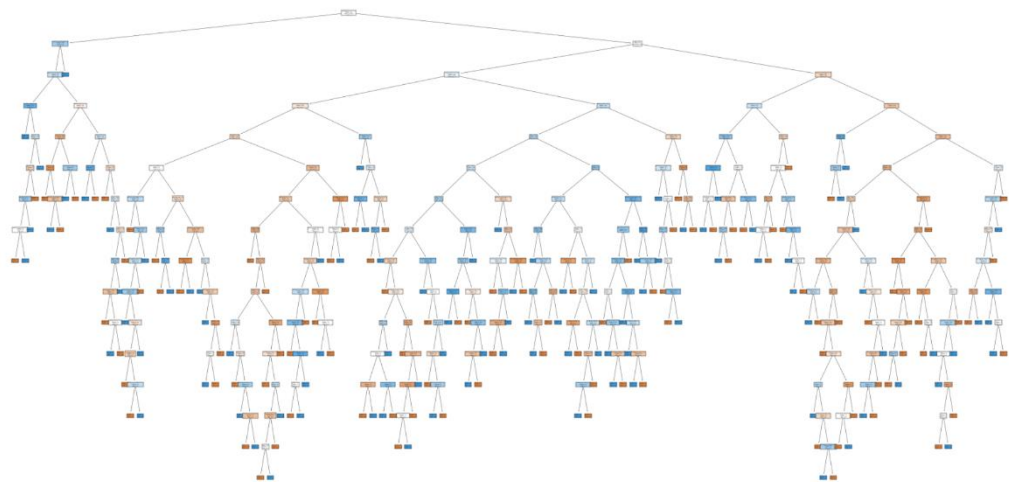
Accuracy: 0.48148148148145



Decision tree classifier using gini with training dataset size= 70% and test dataset size= 30%

Figure (2) Entropy

Accuracy: 0.51440329218107



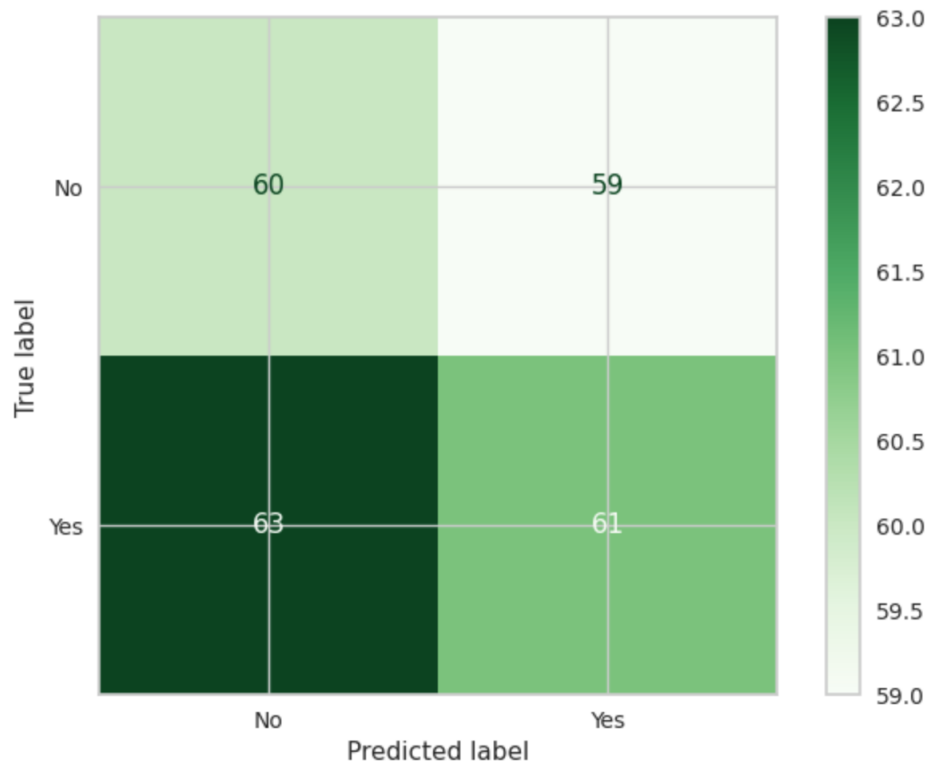
Decision tree classifier using entropy with training dataset size= 70% and test dataset size= 30%

Figure (3) Confusion Matrix

Accuracy: 0.49794238683127573

[[60 59]

[63 61]]



Accuracy of prediction with test size 0.3 lies between 46%-50%

Rules extracted from the decision tree:

Mining task	Comparison Criteria						
Classification	We tried 3 different sizes for dataset splitting to create the decision tree:						
		90% Training data, 10% Test data.		80% Training data, 20% Test data. [BEST]		70% Training data, 30% Test data.	
		IG	Gini Index	IG	Gini Index	IG	Gini Index
	Accuracy	42%	44%	53 %	50%	43%	48%
Clustering							
		K=2		K=3		K=4	
	Average Silhouette width	0.1122		0.0925		0.0777	
	total within-cluster sum of square	10214.99		9644.20		9359.58	

- Clustering [K=2]:

Figure (1)



[111]: <AxesSubplot:title={'center':'Silhouette Plot of KMeans Clustering for 809 Samples in 2 Centers'}, xlabel='silhouette coefficient values', ylabel='cluster label'>

Based on the KMeans Clustering graph for 809 samples with 2 centers, the majority of silhouette scores being positive further supports the idea that the samples are appropriately assigned to their clusters and are sufficiently distant from neighboring clusters. This suggests that the clustering solution has effectively segmented the data points into clear and well-defined clusters.

• Clustering [K=3]:

Figure (1)

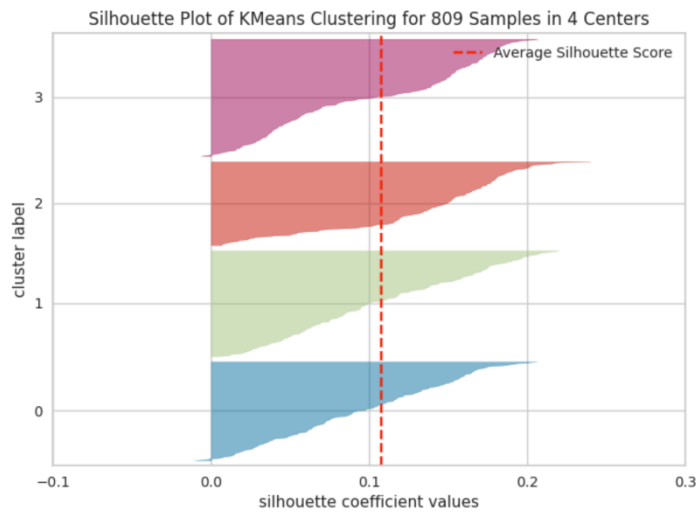


```
[114]: <AxesSubplot:title=('center': 'Silhouette Plot of KMeans Clustering for 809 Samples in 3 Centers'), xlabel='silhouette coefficient values', ylabel='cluster label'>
```

In this silhouette plot, cluster 0 harbors numerous points with high silhouette scores, some even being negative, suggesting that these points are more suitably grouped. The dashed red line denotes the average silhouette score for all samples, hovering just above 0.1 in this instance, indicating that, on average, the clusters are only moderately distinct from each other.

- Clustering [K=4]:

Figure (1)

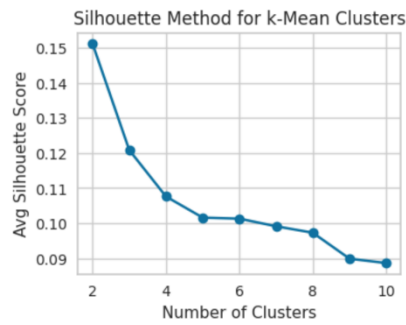


```
[117]: <AxesSubplot:title={'center':'Silhouette Plot of KMeans Clustering for 809 Samples in 4 Centers'}, xlabel='silhouette coefficient values', ylabel='cluster label'>
```

From the KMeans Clustering graph for 809 samples with 4 centers, the prevalence of positive silhouette scores is a clear positive sign. These scores indicate that the samples align well with their clusters and maintain a considerable distance from neighboring clusters. This further strengthens the assertion that the clustering approach effectively segregated the data points into cohesive, clearly defined clusters.

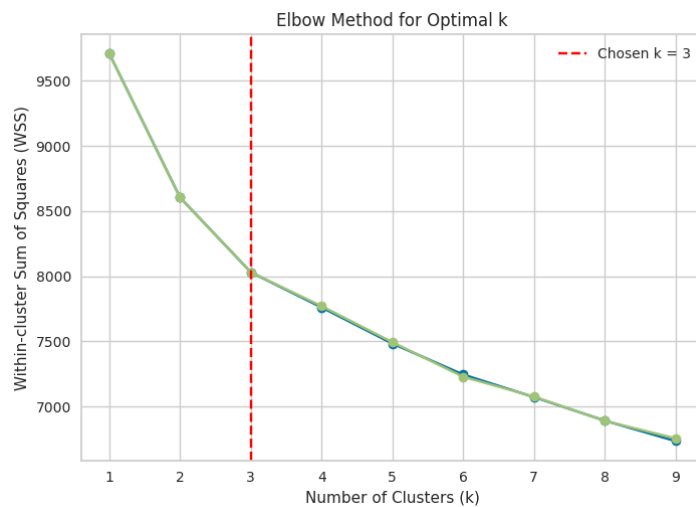
- Optimal Number of Clusters:

Figure (1) Silhouette Method



Silhouette analysis suggests that 2 clusters are optimal for the dataset as they yield the highest silhouette score among the tested range (2 to 10 clusters), indicating the best structure for segmentation.

Figure (2) Elbow Method



Based on the plot provided, the elbow seems to occur at $k=3$, where the line begins to flatten out. This suggests that three clusters might be the optimal choice.

Figure (3) Clustering Model Evaluation

Evaluate the models of Clustering:

	K=2	K=3	K=4
WSS	10214.99	9644.20	9359.58
Average Silhouette Score	0.1122	0.0925	0.0777

We've determined that K=2 is the optimal choice for our clustering model based on the metrics we've evaluated. This decision stems from K=2 yielding the highest silhouette width and having the highest WSS value compared to K=3 and K=4. Additionally, the silhouette plot of KMeans clustering for 809 samples with 2 centers played a crucial role in selecting K=2 as the best option, indicating the formation of distinct and cohesive clusters.

Click to add a cell.

7 Findings:

At first, we studied the predict hair fall dataset, what is each attribute do and how it will affect to each other. We analyzed some cases, to know how the attribute will help

us to predict the best result. After that we did some preprocessing methods to our data, for example: cleaning data and transformation. Then we started our data mining technique which is classification, and we came up with these results:

The evaluation models results of the accuracy in 3 different Comparison Criteria:

- Training set 90% and Testing set 10% accuracy = 0.4567
- Training set 80% and Testing set 20% accuracy = 0.5185
- Training set 70% and Testing set 30% accuracy = 0.4818

The best evaluation model that has best accuracy in the binary trees was the second model which is [80% , 20%]

8. References:

- <https://www.kaggle.com/>
- <https://universeofdatascience.com/how-to-remove-outliers-from-data-in-r/>