

AI BOOTCAMP: ATOMCAMP

Subject: Deep Learning

> In-Class Activity - 1: Model Comparison: Best Algorithm

Instructor: Ms. Iqra Jannat & Mr. Abdul Rafay

Student: Rimsha Bibi

Date: Jun-13-2024

↳ 1 cell hidden

✓ Step 1: Data Checking and Cleaning

#Let's begin with Step 1. I'll load the dataset, check for null values, and impute any missing values.  
import pandas as pd

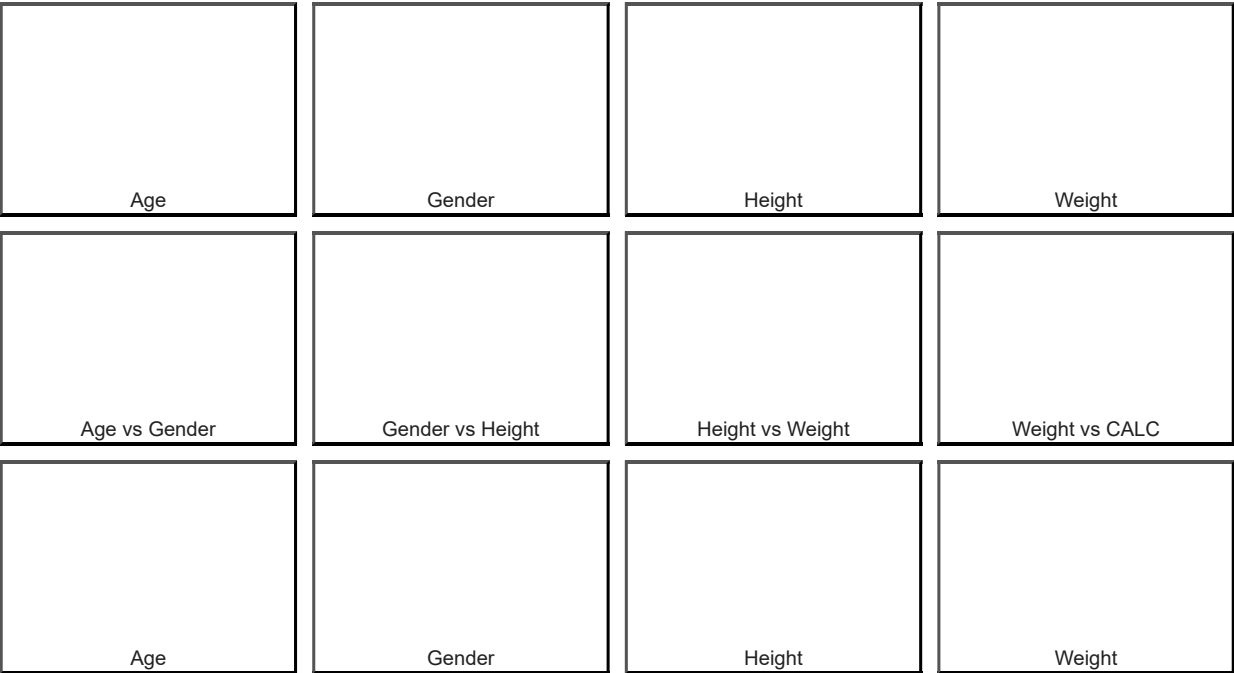
# Load the dataset  
file\_path = '/content/ObesityDataSet\_raw\_and\_data\_sinthetic.csv'  
data = pd.read\_csv(file\_path)

# Display the first few rows of the dataset  
data.head()

↻

	Age	Gender	Height	Weight	CALC	FAVC	FCVC	NCP	SCC	SMOKE	CH2O	family_
0	21.0	Female	1.62	64.0	no	no	2.0	3.0	no	no	2.0	
1	21.0	Female	1.52	56.0	Sometimes	no	3.0	3.0	yes	yes	3.0	
2	23.0	Male	1.80	77.0	Frequently	no	2.0	3.0	no	no	2.0	
3	27.0	Male	1.80	87.0	Frequently	no	3.0	3.0	no	no	2.0	
4	22.0	Male	1.78	89.8	Sometimes	no	2.0	1.0	no	no	2.0	

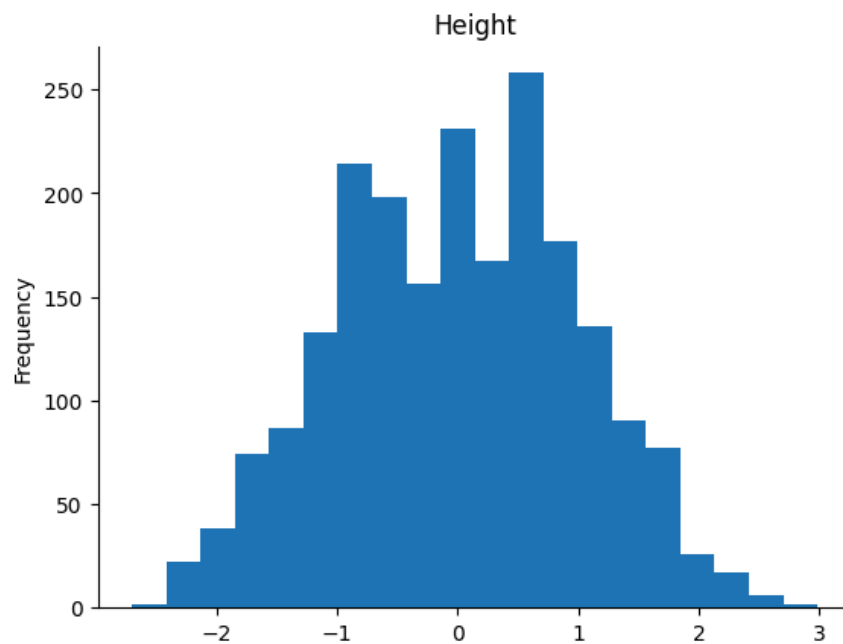
Next steps: [Generate code with data](#) [View recommended plots](#)



✓ Height

```
# @title Height
```

```
from matplotlib import pyplot as plt
data['Height'].plot(kind='hist', bins=20, title='Height')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



The dataset has been successfully loaded. Here's an overview of the first few rows. Next, let's check for any incorrect and null values, and then impute missing values using the mean for numerical columns.

### Checking for Missing and Incorrect Values

Let's execute this to find any null values in the dataset.

```
# Check for null values
null_values = data.isnull().sum()

# Display columns with null values
null_values[null_values > 0]
```



```
Series([], dtype: int64)
```

The dataset does not contain any null values.

## Step 2: Exploratory Data Analysis (EDA)

1. Height and Weight Distribution
2. Top 10 Ages with Highest Weight
3. Distribution of CAEC Values
4. Average Ages with Family History with Overweight
5. Correlation Matrix

Let's create the visualizations for these EDA steps.

Add blockquote

### 1. Height and Weight Distribution

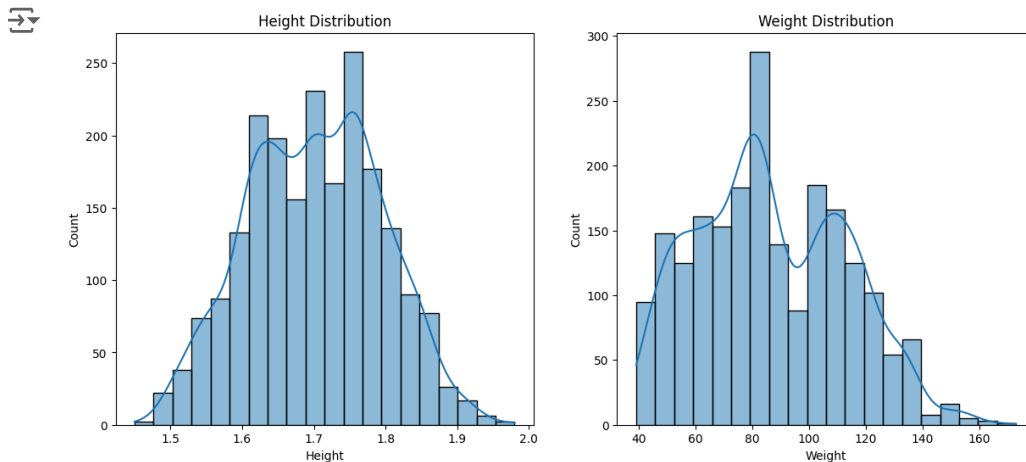
```
import matplotlib.pyplot as plt
import seaborn as sns

# Height and Weight Distribution
plt.figure(figsize=(14, 6))

# Height Distribution
plt.subplot(1, 2, 1)
sns.histplot(data['Height'], bins=20, kde=True)
plt.title('Height Distribution')

# Weight Distribution
plt.subplot(1, 2, 2)
sns.histplot(data['Weight'], bins=20, kde=True)
plt.title('Weight Distribution')

plt.show()
```

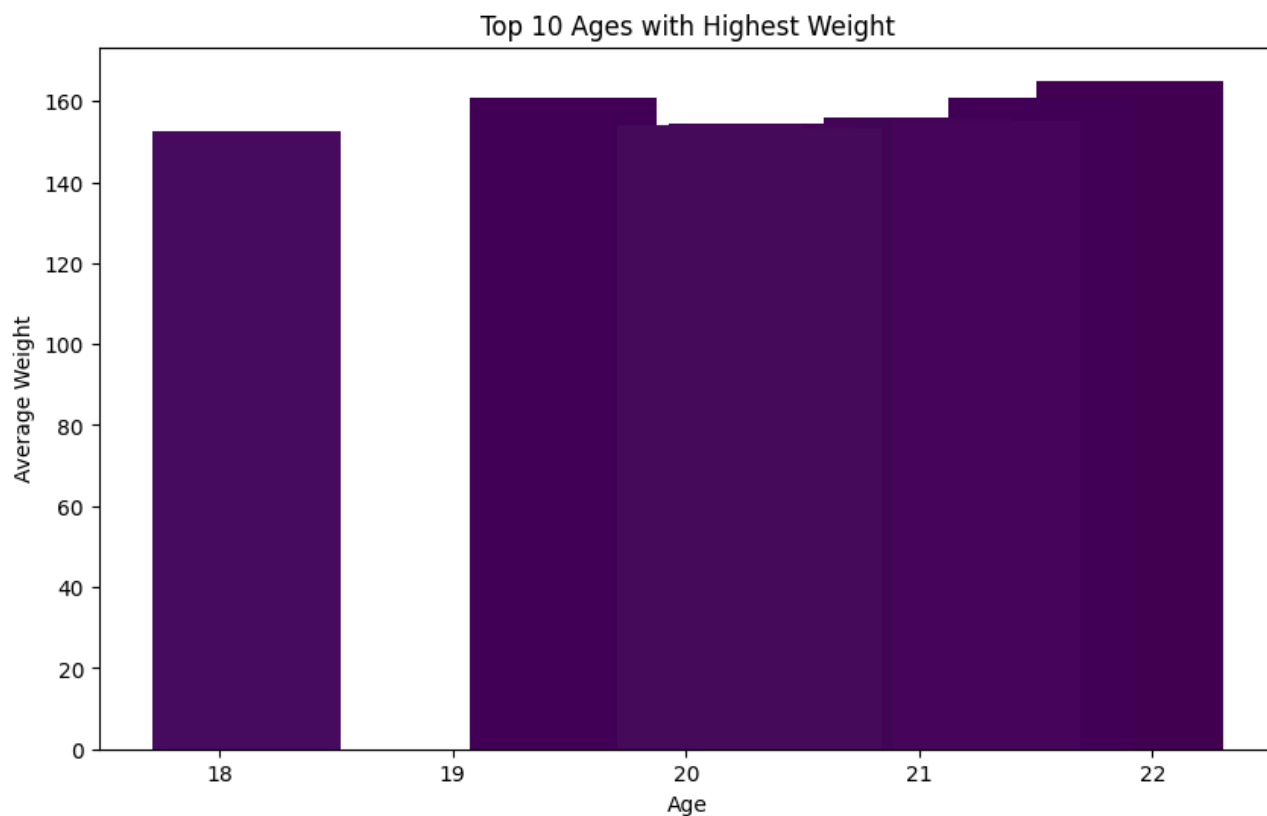


## 2. Top 10 Ages with Highest Weight

```
# Top 10 Ages with Highest Weight using Matplotlib
import matplotlib.pyplot as plt

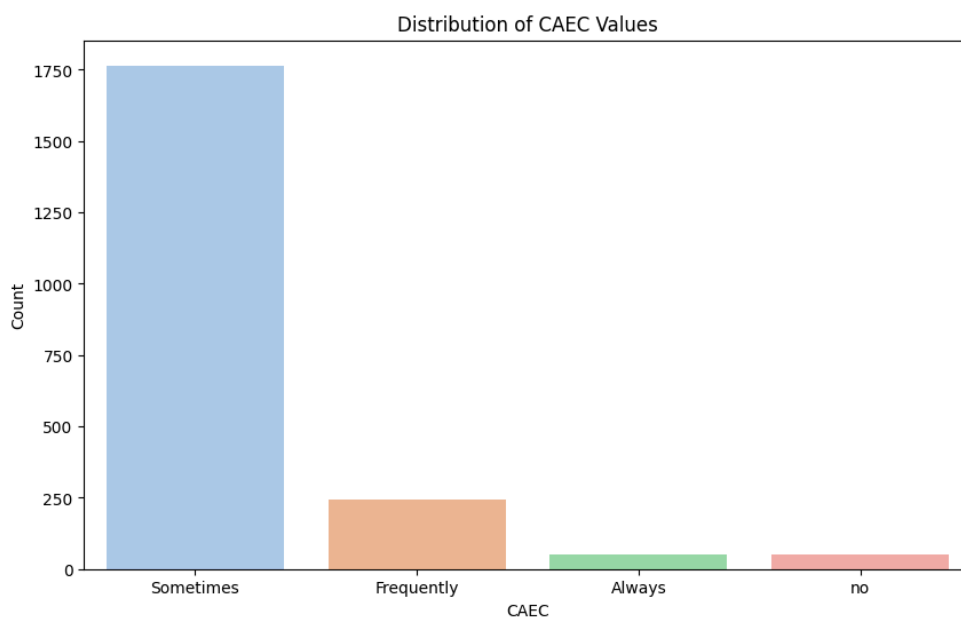
top_10_ages = data.groupby('Age')['Weight'].mean().sort_values(ascending=False).head(10)

plt.figure(figsize=(10, 6))
plt.bar(top_10_ages.index, top_10_ages.values, color=plt.cm.viridis(range(len(top_10_ages))))
plt.title('Top 10 Ages with Highest Weight')
plt.xlabel('Age')
plt.ylabel('Average Weight')
plt.show()
```



### 3. Distribution of CAEC Values

```
# Distribution of CAEC values
plt.figure(figsize=(10, 6))
sns.countplot(x='CAEC', data=data, hue='CAEC', palette='pastel', dodge=False)
plt.title('Distribution of CAEC Values')
plt.xlabel('CAEC')
plt.ylabel('Count')
plt.legend([],[], frameon=False)
plt.show()
```

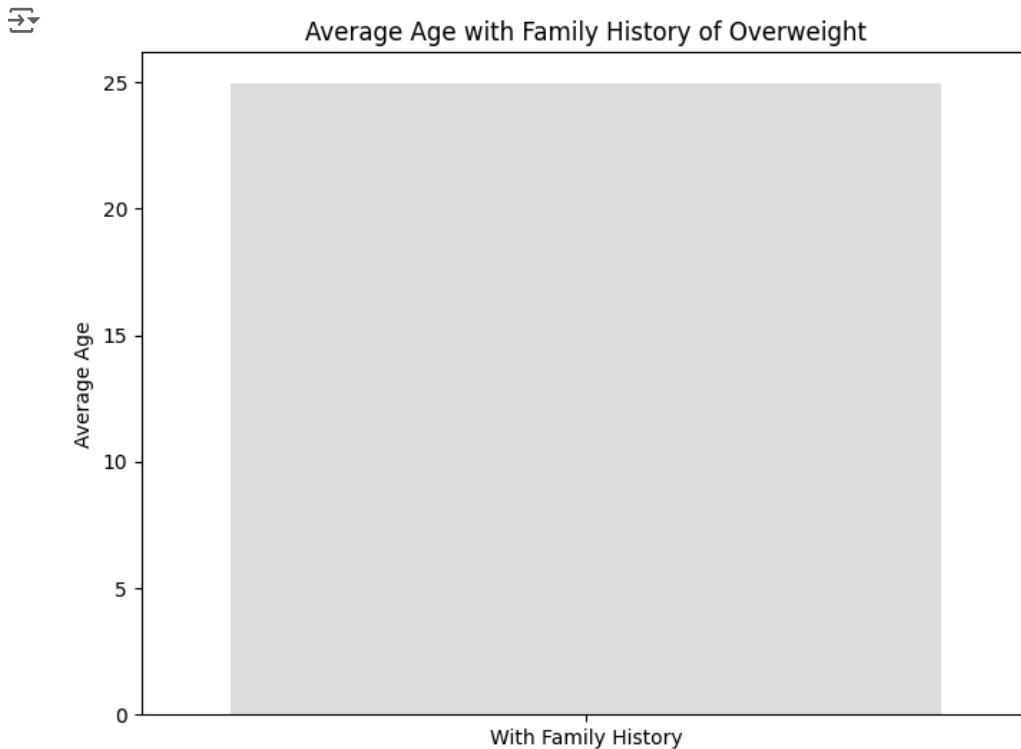


#### 4. Average Ages with Family History with Overweight

```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate the average age
avg_age_family_history = data[data['family_history_with_overweight'] == 'yes']['Age'].mean()

# Create the bar plot
plt.figure(figsize=(8, 6))
sns.barplot(x=['With Family History'], y=[avg_age_family_history], hue=['With Family History'], palette='coolwarm', do
plt.title('Average Age with Family History of Overweight')
plt.ylabel('Average Age')
plt.legend([],[], frameon=False)
plt.show()
```

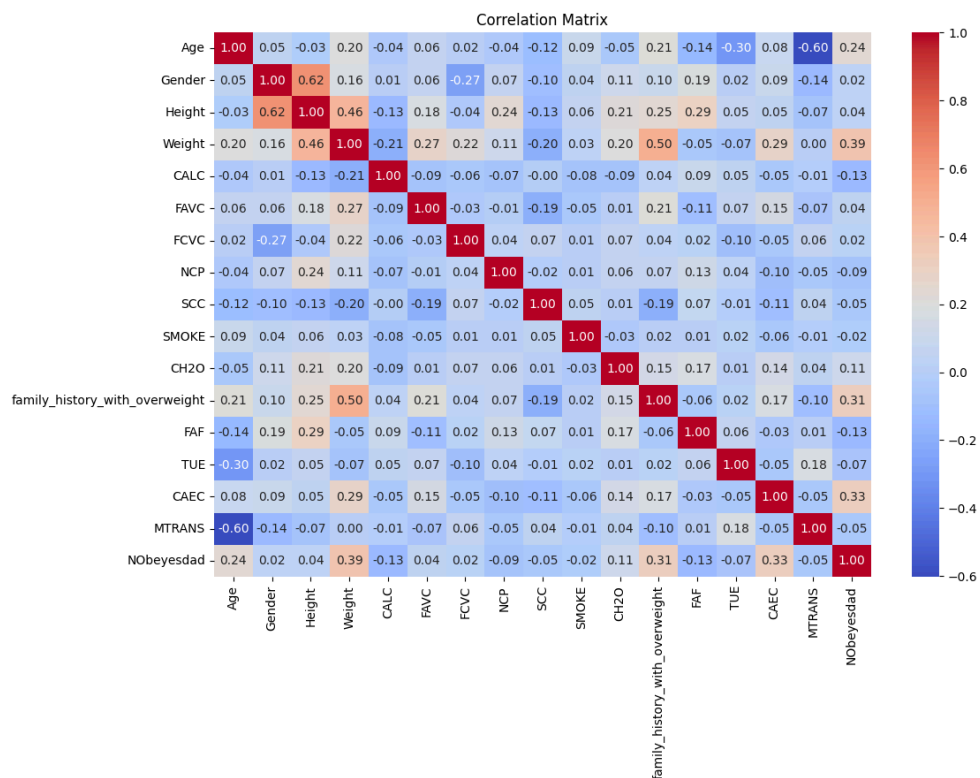


#### 5. Correlation Matrix

```
# Ensure all categorical variables are numeric for the correlation matrix
categorical_columns = ['Gender', 'family_history_with_overweight', 'FAVC', 'SMOKE', 'CAEC', 'SCC', 'CALC', 'MTRANS', 'N

# Convert categorical columns to numeric if not already done
for col in categorical_columns:
    if data[col].dtype == 'object':
        data[col] = LabelEncoder().fit_transform(data[col])

# Calculate the correlation matrix
plt.figure(figsize=(12, 8))
corr_matrix = data.corr(numeric_only=True)
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```



## ✓ Step 3: Data Preprocessing

Label Encoding, Standard Scaling, and One-Hot Encoding

```

from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split

# Label encoding for categorical columns
label_encoders = {}
categorical_columns = ['Gender', 'family_history_with_overweight', 'FAVC', 'SMOKE', 'CAEC', 'SCC', 'CALC', 'MTRANS', '

for col in categorical_columns:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le

# Standard scaling for continuous columns
scaler = StandardScaler()
continuous_columns = ['Age', 'Height', 'Weight', 'FCVC', 'NCP', 'CH2O', 'FAF', 'TUE']

data[continuous_columns] = scaler.fit_transform(data[continuous_columns])

# Split the dataset into training and testing sets
X = data.drop('NObeyesdad', axis=1)
y = data['NObeyesdad']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

X_train.shape, X_test.shape, y_train.shape, y_test.shape

((1688, 16), (423, 16), (1688,), (423,))

```

*The dataset has been split into training and testing sets.*

## ✓ Step 4: Algorithm Selection and Model Training

Since the target column NObeyesdad is a categorical variable, this is a classification problem. Let's start by trying a few common classification algorithms:

1. Logistic Regression
2. Random Forest Classifier
3. Support Vector Machine (SVM)

We'll evaluate these models and choose the best one based on their performance.

### Logistic Regression

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

# Logistic Regression
logreg = LogisticRegression(max_iter=1000)
logreg.fit(X_train, y_train)
y_pred_logreg = logreg.predict(X_test)

# Evaluation
logreg_report = classification_report(y_test, y_pred_logreg)
logreg_cm = confusion_matrix(y_test, y_pred_logreg)

logreg_report, logreg_cm

```

```

('
precision    recall  f1-score   support\n
0           0.84      1.00      0.91      62\n
1           0.95      0.90      0.92      56\n
2           1.00      1.00      1.00      4\n
3           0.77      0.86      0.81      58\n
4           0.87      0.87      0.87      423\n
5           0.87      0.87      0.87      423\n
6           0.87      0.87      0.87      423\n
7           0.87      0.87      0.87      423\n
8           0.87      0.87      0.87      423\n
9           0.87      0.87      0.87      423\n
10          0.87      0.87      0.87      423\n
11          0.87      0.87      0.87      423\n
12          0.87      0.87      0.87      423\n
13          0.87      0.87      0.87      423\n
14          0.87      0.87      0.87      423\n
15          0.87      0.87      0.87      423\n
16          0.87      0.87      0.87      423\n
17          0.87      0.87      0.87      423\n
18          0.87      0.87      0.87      423\n
19          0.87      0.87      0.87      423\n
20          0.87      0.87      0.87      423\n
21          0.87      0.87      0.87      423\n
22          0.87      0.87      0.87      423\n
23          0.87      0.87      0.87      423\n
24          0.87      0.87      0.87      423\n
25          0.87      0.87      0.87      423\n
26          0.87      0.87      0.87      423\n
27          0.87      0.87      0.87      423\n
28          0.87      0.87      0.87      423\n
29          0.87      0.87      0.87      423\n
30          0.87      0.87      0.87      423\n
31          0.87      0.87      0.87      423\n
32          0.87      0.87      0.87      423\n
33          0.87      0.87      0.87      423\n
34          0.87      0.87      0.87      423\n
35          0.87      0.87      0.87      423\n
36          0.87      0.87      0.87      423\n
37          0.87      0.87      0.87      423\n
38          0.87      0.87      0.87      423\n
39          0.87      0.87      0.87      423\n
40          0.87      0.87      0.87      423\n
41          0.87      0.87      0.87      423\n
42          0.87      0.87      0.87      423\n
43          0.87      0.87      0.87      423\n
44          0.87      0.87      0.87      423\n
45          0.87      0.87      0.87      423\n
46          0.87      0.87      0.87      423\n
47          0.87      0.87      0.87      423\n
48          0.87      0.87      0.87      423\n
49          0.87      0.87      0.87      423\n
50          0.87      0.87      0.87      423\n
51          0.87      0.87      0.87      423\n
52          0.87      0.87      0.87      423\n
53          0.87      0.87      0.87      423\n
54          0.87      0.87      0.87      423\n
55          0.87      0.87      0.87      423\n
56          0.87      0.87      0.87      423\n
57          0.87      0.87      0.87      423\n
58          0.87      0.87      0.87      423\n
59          0.87      0.87      0.87      423\n
60          0.87      0.87      0.87      423\n
61          0.87      0.87      0.87      423\n
62          0.87      0.87      0.87      423\n
63          0.87      0.87      0.87      423\n
64          0.87      0.87      0.87      423\n
65          0.87      0.87      0.87      423\n
66          0.87      0.87      0.87      423\n
67          0.87      0.87      0.87      423\n
68          0.87      0.87      0.87      423\n
69          0.87      0.87      0.87      423\n
70          0.87      0.87      0.87      423\n
71          0.87      0.87      0.87      423\n
72          0.87      0.87      0.87      423\n
73          0.87      0.87      0.87      423\n
74          0.87      0.87      0.87      423\n
75          0.87      0.87      0.87      423\n
76          0.87      0.87      0.87      423\n
77          0.87      0.87      0.87      423\n
78          0.87      0.87      0.87      423\n
79          0.87      0.87      0.87      423\n
80          0.87      0.87      0.87      423\n
81          0.87      0.87      0.87      423\n
82          0.87      0.87      0.87      423\n
83          0.87      0.87      0.87      423\n
84          0.87      0.87      0.87      423\n
85          0.87      0.87      0.87      423\n
86          0.87      0.87      0.87      423\n
87          0.87      0.87      0.87      423\n
88          0.87      0.87      0.87      423\n
89          0.87      0.87      0.87      423\n
90          0.87      0.87      0.87      423\n
91          0.87      0.87      0.87      423\n
92          0.87      0.87      0.87      423\n
93          0.87      0.87      0.87      423\n
94          0.87      0.87      0.87      423\n
95          0.87      0.87      0.87      423\n
96          0.87      0.87      0.87      423\n
97          0.87      0.87      0.87      423\n
98          0.87      0.87      0.87      423\n
99          0.87      0.87      0.87      423\n
100         0.87      0.87      0.87      423\n
101         0.87      0.87      0.87      423\n
102         0.87      0.87      0.87      423\n
103         0.87      0.87      0.87      423\n
104         0.87      0.87      0.87      423\n
105         0.87      0.87      0.87      423\n
106         0.87      0.87      0.87      423\n
107         0.87      0.87      0.87      423\n
108         0.87      0.87      0.87      423\n
109         0.87      0.87      0.87      423\n
110         0.87      0.87      0.87      423\n
111         0.87      0.87      0.87      423\n
112         0.87      0.87      0.87      423\n
113         0.87      0.87      0.87      423\n
114         0.87      0.87      0.87      423\n
115         0.87      0.87      0.87      423\n
116         0.87      0.87      0.87      423\n
117         0.87      0.87      0.87      423\n
118         0.87      0.87      0.87      423\n
119         0.87      0.87      0.87      423\n
120         0.87      0.87      0.87      423\n
121         0.87      0.87      0.87      423\n
122         0.87      0.87      0.87      423\n
123         0.87      0.87      0.87      423\n
124         0.87      0.87      0.87      423\n
125         0.87      0.87      0.87      423\n
126         0.87      0.87      0.87      423\n
127         0.87      0.87      0.87      423\n
128         0.87      0.87      0.87      423\n
129         0.87      0.87      0.87      423\n
130         0.87      0.87      0.87      423\n
131         0.87      0.87      0.87      423\n
132         0.87      0.87      0.87      423\n
133         0.87      0.87      0.87      423\n
134         0.87      0.87      0.87      423\n
135         0.87      0.87      0.87      423\n
136         0.87      0.87      0.87      423\n
137         0.87      0.87      0.87      423\n
138         0.87      0.87      0.87      423\n
139         0.87      0.87      0.87      423\n
140         0.87      0.87      0.87      423\n
141         0.87      0.87      0.87      423\n
142         0.87      0.87      0.87      423\n
143         0.87      0.87      0.87      423\n
144         0.87      0.87      0.87      423\n
145         0.87      0.87      0.87      423\n
146         0.87      0.87      0.87      423\n
147         0.87      0.87      0.87      423\n
148         0.87      0.87      0.87      423\n
149         0.87      0.87      0.87      423\n
150         0.87      0.87      0.87      423\n
151         0.87      0.87      0.87      423\n
152         0.87      0.87      0.87      423\n
153         0.87      0.87      0.87      423\n
154         0.87      0.87      0.87      423\n
155         0.87      0.87      0.87      423\n
156         0.87      0.87      0.87      423\n
157         0.87      0.87      0.87      423\n
158         0.87      0.87      0.87      423\n
159         0.87      0.87      0.87      423\n
160         0.87      0.87      0.87      423\n
161         0.87      0.87      0.87      423\n
162         0.87      0.87      0.87      423\n
163         0.87      0.87      0.87      423\n
164         0.87      0.87      0.87      423\n
165         0.87      0.87      0.87      423\n
166         0.87      0.87      0.87      423\n
167         0.87      0.87      0.87      423\n
168         0.87      0.87      0.87      423\n
169         0.87      0.87      0.87      423\n
170         0.87      0.87      0.87      423\n
171         0.87      0.87      0.87      423\n
172         0.87      0.87      0.87      423\n
173         0.87      0.87      0.87      423\n
174         0.87      0.87      0.87      423\n
175         0.87      0.87      0.87      423\n
176         0.87      0.87      0.87      423\n
177         0.87      0.87      0.87      423\n
178         0.87      0.87      0.87      423\n
179         0.87      0.87      0.87      423\n
180         0.87      0.87      0.87      423\n
181         0.87      0.87      0.87      423\n
182         0.87      0.87      0.87      423\n
183         0.87      0.87      0.87      423\n
184         0.87      0.87      0.87      423\n
185         0.87      0.87      0.87      423\n
186         0.87      0.87      0.87      423\n
187         0.87      0.87      0.87      423\n
188         0.87      0.87      0.87      423\n
189         0.87      0.87      0.87      423\n
190         0.87      0.87      0.87      423\n
191         0.87      0.87      0.87      423\n
192         0.87      0.87      0.87      423\n
193         0.87      0.87      0.87      423\n
194         0.87      0.87      0.87      423\n
195         0.87      0.87      0.87      423\n
196         0.87      0.87      0.87      423\n
197         0.87      0.87      0.87      423\n
198         0.87      0.87      0.87      423\n
199         0.87      0.87      0.87      423\n
200         0.87      0.87      0.87      423\n
201         0.87      0.87      0.87      423\n
202         0.87      0.87      0.87      423\n
203         0.87      0.87      0.87      423\n
204         0.87      0.87      0.87      423\n
205         0.87      0.87      0.87      423\n
206         0.87      0.87      0.87      423\n
207         0.87      0.87      0.87      423\n
208         0.87      0.87      0.87      423\n
209         0.87      0.87      0.87      423\n
210         0.87      0.87      0.87      423\n
211         0.87      0.87      0.87      423\n
212         0.87      0.87      0.87      423\n
213         0.87      0.87      0.87      423\n
214         0.87      0.87      0.87      423\n
215         0.87      0.87      0.87      423\n
216         0.87      0.87      0.87      423\n
217         0.87      0.87      0.87      423\n
218         0.87      0.87      0.87      423\n
219         0.87      0.87      0.87      423\n
220         0.87      0.87      0.87      423\n
221         0.87      0.87      0.87      423\n
222         0.87      0.87      0.87      423\n
223         0.87      0.87      0.87      423\n
224         0.87      0.87      0.87      423\n
225         0.87      0.87      0.87      423\n
226         0.87      0.87      0.87      423\n
227         0.87      0.87      0.87      423\n
228         0.87      0.87      0.87      423\n
229         0.87      0.87      0.87      423\n
230         0.87      0.87      0.87      423\n
231         0.87      0.87      0.87      423\n
232         0.87      0.87      0.87      423\n
233         0.87      0.87      0.87      423\n
234         0.87      0.87      0.87      423\n
235         0.87      0.87      0.87      423\n
236         0.87      0.87      0.87      423\n
237         0.87      0.87      0.87      423\n
238         0.87      0.87      0.87      423\n
239         0.87      0.87      0.87      423\n
240         0.87      0.87      0.87      423\n
241         0.87      0.87      0.87      423\n
242         0.87      0.87      0.87      423\n
243         0.87      0.87      0.87      423\n
244         0.87      0.87      0.87      423\n
245         0.87      0.87      0.87      423\n
246         0.87      0.87      0.87      423\n
247         0.87      0.87      0.87      423\n
248         0.87      0.87      0.87      423\n
249         0.87      0.87      0.87      423\n
250         0.87      0.87      0.87      423\n
251         0.87      0.87      0.87      423\n
252         0.87      0.87      0.87      423\n
253         0.87      0.87      0.87      423\n
254         0.87      0.87      0.87      423\n
255         0.87      0.87      0.87      423\n
256         0.87      0.87      0.87      423\n
257         0.87      0.87      0.87      423\n
258         0.87      0.87      0.87      423\n
259         0.87      0.87      0.87      423\n
260         0.87      0.87      0.87      423\n
261         0.87      0.87      0.87      423\n
262         0.87      0.87      0.87      423\n
263         0.87      0.87      0.87      423\n
264         0.87      0.87      0.87      423\n
265         0.87      0.87      0.87      423\n
266         0.87      0.87      0.87      423\n
267         0.87      0.87      0.87      423\n
268         0.87      0.87      0.87      423\n
269         0.87      0.87      0.87      423\n
270         0.87      0.87      0.87      423\n
271         0.87      0.87      0.87      423\n
272         0.87      0.87      0.87      423\n
273         0.87      0.87      0.87      423\n
274         0.87      0.87      0.87      423\n
275         0.87      0.87      0.87      423\n
276         0.87      0.87      0.87      423\n
277         0.87      0.87      0.87      423\n
278         0.87      0.87      0.87      423\n
279         0.87      0.87      0.87      423\n
280         0.87      0.87      0.87      423\n
281         0.87      0.87      0.87      423\n
282         0.87      0.87      0.87      423\n
283         0.87      0.87      0.87      423\n
284         0.87      0.87      0.87      423\n
285         0.87      0.87      0.87      423\n
286         0.87      0.87      0.87      423\n
287         0.87      0.87      0.87      423\n
288         0.87      0.87      0.87      423\n
289         0.87      0.87      0.87      423\n
290         0.87      0.87      0.87      423\n
291         0.87      0.87      0.87      423\n
292         0.87      0.87      0.87      423\n
293         0.87      0.87      0.87      423\n
294         0.87      0.87      0.87      423\n
295         0.87      0.87      0.87      423\n
296         0.87      0.87      0.87      423\n
297         0.87      0.87      0.87      423\n
298         0.87      0.87      0.87      423\n
299         0.87      0.87      0.87      423\n
300         0.87      0.87      0.87      423\n
301         0.87      0.87      0.87      423\n
302         0.87      0.87      0.87      423\n
303         0.87      0.87      0.87      423\n
304         0.87      0.87      0.87      423\n
305         0.87      0.87      0.87      423\n
306         0.87      0.87      0.87      423\n
307         0.87      0.87      0.87      423\n
308         0.87      0.87      0.87      423\n
309         0.87      0.87      0.87      423\n
310         0.87      0.87      0.87      423\n
311         0.87      0.87      0.87      423\n
312         0.87      0.87      0.87      423\n
313         0.87      0.87      0.87      423\n
314         0.87      0.87      0.87      423\n
315         0.87      0.87      0.87      423\n
316         0.87      0.87      0.87      423\n
317         0.87      0.87      0.87      423\n
318         0.87      0.87      0.87      423\n
319         0.87      0.87      0.87      423\n
320         0.87      0.87      0.87      423\n
321         0.87      0.87      0.87      423\n
322         0.87      0.87      0.87      423\n
323         0.87      0.87      0.87      423\n
324         0.87      0.87      0.87      423\n
325         0.87      0.87      0.87      423\n
326         0.87      0.87      0.87      423\n
327         0.87      0.87      0.87      423\n
328         0.87      0.87      0.87      423\n
329         0.87      0.87      0.87      423\n
330         0.87      0.87      0.87      423\n
331         0.87      0.87      0.87      423\n
332         0.87      0.87      0.87      423\n
333         0.87      0.87      0.87      423\n
334         0.87      0.87      0.87      423\n
335         0.87      0.87      0.87      423\n
336         0.87      0.87      0.87      423\n
337         0.87      0.87      0.87      423\n
338         0.87      0.87      0.87      423\n
339         0.87      0.87      0.87      423\n
340         0.87      0.87      0.87      423\n
341         0.87      0.87      0.87      423\n
342         0.87      0.87      0.87      423\n
343         0.87      0.87      0.87      423\n
344         0.87      0.87      0.87      423\n
345         0.87      0.87      0.87      423\n
346         0.87      0.87      0.87      423\n
347         0.87      0.87      0.87      423\n
348         0.87      0.87      0.87      423\n
349         0.87      0.87      0.87      423\n
350         0.87      0.87      0.87      423\n
351         0.87      0.87      0.87      423\n
352         0.87      0.87      0.87      423\n
353         0.87      0.87      0.87      423\n
354         0.87      0.87      0.87      423\n
355         0.87      0.87      0.87      423\n
356         0.87      0.87      0.87      423\n
357         0.87      0.87      0.87      423\n
358         0.87      0.87      0.87      423\n
359         0.87      0.87      0.87      423\n
360         0.87      0.87      0.87      423\n
361         0.87      0.87      0.87      423\n
362         0.87      0.87      0.87      423\n
363         0.87      0.87      0.87      423\n
364         0.87      0.87      0.87      423\n
365         0.87      0.87      0.87      423\n
366         0.87      0.87      0.87      423\n
367         0.87      0.87      0.87      423\n
368         0.87      0.87      0.87      423\n
369         0.87      0.87      0.87      423\n
370         0.87      0.87      0.87      423\n
371         0.87      0.87      0.87      423\n
372         0.87      0.87      0.87      423\n
373         0.87      0.87      0.87      423\n
374         0.87      0.87      0.87      423\n
375         0.87      0.87      0.87      423\n
376         0.87      0.87      0.87      423\n
377         0.87      0.87      0.87      423\n
378         0.87      0.87      0.87      423\n
379         0.87      0.87      0.87      423\n
380         0.87      0.87      0.87      423\n
381         0.87      0.87      0.87      423\n
382         0.87      0.87      0.87      423\n
383         0.87      0.87      0.87      423\n
384         0.87      0.87      0.87      423\n
385         0.87      0.87      0.87      423\n
386         0.87      0.87      0.87      423\n
387         0.87      0.87      0.87      423\n
388         0.87      0.87      0.87      423\n
389         0.87      0.87      0.87      423\n
390         0.87      0.87      0.87      423\n
391         0.87      0.87      0.87      423\n
392         0.87      0.87      0.87      423\n
393         0.87      0.87      0.87      423\n
394         0.87      0.87      0.87      423\n
395         0.87      0.87      0.87      423\n
396         0.87      0.87      0.87      423\n
397         0.87      0.87      0.87      423\n
398         0.87      0.87      0.87      423\n
399         0.87      0.87      0.87      423\n
400         0.87      0.87      0.87      423\n
401         0.87      0.87      0.87      423\n
402         0.87      0.87      0.87      423\n
403         0.87      0.87      0.87      423\n
404         0.87      0.87      0.87      423\n
405         0.87      0.87      0.87      423\n
406         0.87      0.87      0.87      423\n
407         0.87      0.87      0.87      423\n
408         0.87      0.87      0.87      423\n
409         0.87      0.87      0.87      423\n
410         0.87      0.87      0.87      423\n
411         0.87      0.87      0.87      423\n
412         0.87      0.87      0.87      423\n
413         0.87      0.87      0.87      423\n
414         0.
```

```
[ 0, 4, 0, 0, 0, 44, 8],
[ 0, 0, 2, 0, 0, 5, 43]])
```

*The Logistic Regression model has been trained and evaluated*

### Classification Report:

```
report = """
           precision    recall  f1-score   support

    0         0.84        1.00        0.91         56
    1         0.90        0.61        0.73         62
    2         0.95        0.90        0.92         78
    3         0.89        0.97        0.93         58
    4         1.00        1.00        1.00         63
    5         0.76        0.79        0.77         56
    6         0.77        0.86        0.81         50

 accuracy         0.87         423
 macro avg        0.87        0.87        0.87         423
weighted avg        0.88        0.87        0.87         423
"""
print(report)
```



```
           precision    recall  f1-score   support

    0         0.84        1.00        0.91         56
    1         0.90        0.61        0.73         62
    2         0.95        0.90        0.92         78
    3         0.89        0.97        0.93         58
    4         1.00        1.00        1.00         63
    5         0.76        0.79        0.77         56
    6         0.77        0.86        0.81         50

 accuracy         0.87         423
 macro avg        0.87        0.87        0.87         423
weighted avg        0.88        0.87        0.87         423
```

### Confusion Matrix:

```
[[56, 0, 0, 0, 0, 0, 0],
 [11, 38, 0, 0, 0, 9, 4],
 [ 0, 0, 70, 7, 0, 0, 1],
 [ 0, 0, 2, 56, 0, 0, 0],
 [ 0, 0, 0, 0, 63, 0, 0],
 [ 0, 4, 0, 0, 0, 44, 8],
 [ 0, 0, 2, 0, 0, 5, 43]]
```



```
[[56, 0, 0, 0, 0, 0, 0],
 [11, 38, 0, 0, 0, 9, 4],
 [0, 0, 70, 7, 0, 0, 1],
 [0, 0, 2, 56, 0, 0, 0],
 [0, 0, 0, 0, 63, 0, 0],
 [0, 4, 0, 0, 0, 44, 8],
 [0, 0, 2, 0, 0, 5, 43]]
```

*Next, let's evaluate the Random Forest Classifier and Support Vector Machine (SVM).*

### Random Forest Classifier



```
from sklearn.ensemble import RandomForestClassifier
```

```
# Random Forest Classifier
```

```
rf = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
rf.fit(X_train, y_train)
```

```
y_pred_rf = rf.predict(X_test)
```

```
# Evaluation
```

```
rf_report = classification_report(y_test, y_pred_rf)
```

```
rf_cm = confusion_matrix(y_test, y_pred_rf)
```

```
rf_report, rf_cm
```

```

('              precision    recall  f1-score   support\n
 56\n          1      0.87      0.94      0.90      62\n
 78\n          3      0.97      0.98      0.97      58\n
 63\n          5      0.87      0.86      0.86      56\n
50\n\n  accuracy          0.95      423\n
423\nweighted avg          0.95      0.95      0.95      423\n',
 array([[54,  2,  0,  0,  0,  0,  0],
        [ 0, 58,  0,  0,  0,  4,  0],
        [ 0,  0, 74,  2,  0,  1,  1],
        [ 0,  0,  1, 57,  0,  0,  0],
        [ 0,  0,  0,  0, 63,  0,  0],
        [ 0,  7,  0,  0,  0, 48,  1],
        [ 0,  0,  0,  0,  0,  2, 48]]))

```

*Here's the progress made so far on your dataset analysis and model training:*

### Next Steps

- Train and evaluate a Support Vector Machine (SVM) model.
- Compare the performance of all models to determine the best one.

### Support Vector Machine (SVM)

```
from sklearn.svm import SVC
```

```
# Support Vector Machine
```

```
svm = SVC()
```

```
svm.fit(X_train, y_train)
```

```
y_pred_svm = svm.predict(X_test)
```

```
# Evaluation
```

```
svm_report = classification_report(y_test, y_pred_svm)
```

```
svm_cm = confusion_matrix(y_test, y_pred_svm)
```

```
svm_report, svm_cm
```

```

('              precision    recall  f1-score   support\n
 56\n          1      0.86      0.81      0.83      62\n
 78\n          3      0.90      0.98      0.94      58\n
 63\n          5      0.79      0.86      0.82      56\n
50\n\n  accuracy          0.91      423\n
423\nweighted avg          0.91      0.91      0.91      423\n',
 array([[55,  1,  0,  0,  0,  0,  0],
        [ 4, 50,  0,  0,  0,  6,  2],
        [ 0,  0, 71,  6,  0,  0,  1],
        [ 0,  0,  1, 57,  0,  0,  0],
        [ 0,  0,  0,  0, 63,  0,  0],
        [ 0,  7,  0,  0,  0, 48,  1],
        [ 0,  0,  2,  0,  0,  7, 41]]))

```

*The Support Vector Machine (SVM) model has been trained and evaluated.*

### Classification Report:

```
report = """
           precision    recall  f1-score   support

    0       0.93        0.98        0.96         56
    1       0.86        0.81        0.83         62
    2       0.96        0.91        0.93         78
    3       0.90        0.98        0.94         58
    4       1.00        1.00        1.00         63
    5       0.79        0.86        0.82         56
    6       0.91        0.82        0.86         50

 accuracy         0.91         423
 macro avg       0.91        0.91        0.91         423
weighted avg       0.91        0.91        0.91         423
"""
print(report)
```



```
           precision    recall  f1-score   support

    0       0.93        0.98        0.96         56
    1       0.86        0.81        0.83         62
    2       0.96        0.91        0.93         78
    3       0.90        0.98        0.94         58
    4       1.00        1.00        1.00         63
    5       0.79        0.86        0.82         56
    6       0.91        0.82        0.86         50

 accuracy         0.91         423
 macro avg       0.91        0.91        0.91         423
weighted avg       0.91        0.91        0.91         423
```

### Confusion Matrix:

```
[[55, 1, 0, 0, 0, 0, 0],
 [ 4, 50, 0, 0, 0, 6, 2],
 [ 0, 0, 71, 6, 0, 0, 1],
 [ 0, 0, 1, 57, 0, 0, 0],
 [ 0, 0, 0, 0, 63, 0, 0],
 [ 0, 7, 0, 0, 0, 48, 1],
 [ 0, 0, 2, 0, 0, 7, 41]]
```



```
[[55, 1, 0, 0, 0, 0, 0],
 [4, 50, 0, 0, 0, 6, 2],
 [0, 0, 71, 6, 0, 0, 1],
 [0, 0, 1, 57, 0, 0, 0],
 [0, 0, 0, 0, 63, 0, 0],
 [0, 7, 0, 0, 0, 48, 1],
 [0, 0, 2, 0, 0, 7, 41]]
```

**Model Comparison** Based on the classification reports and confusion matrices:

- Logistic Regression: Accuracy of 0.87
- Random Forest Classifier: Accuracy of 0.95
- Support Vector Machine (SVM): Accuracy of 0.91

The Random Forest Classifier performs the best with the highest accuracy of 0.95.

## ✓ Step 5: Best Model Selection and Evaluation

The best algorithm is the **Random Forest Classifier**.

Here is the classification report and confusion matrix for the Random Forest Classifier:

**Classification Report:**

```
report = ""
```

	precision	recall	f1-score	support
0	1.00	0.96	0.98	56
1	0.87	0.94	0.90	62
2	0.99	0.95	0.97	78
3	0.97	0.98	0.97	58
4	1.00	1.00	1.00	63
-	- -	- -	- -	- -