

Data Collection and Preprocessing Phase

Date	28 June 2024
Team ID	739857
Project Title	Predictive Pulse: Harnessing Machine Learning For Blood Pressure Analysis.

Maximum Marks	6 Marks

Data Exploration and Preprocessing Report

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	Description

Descriptive Analysis:-

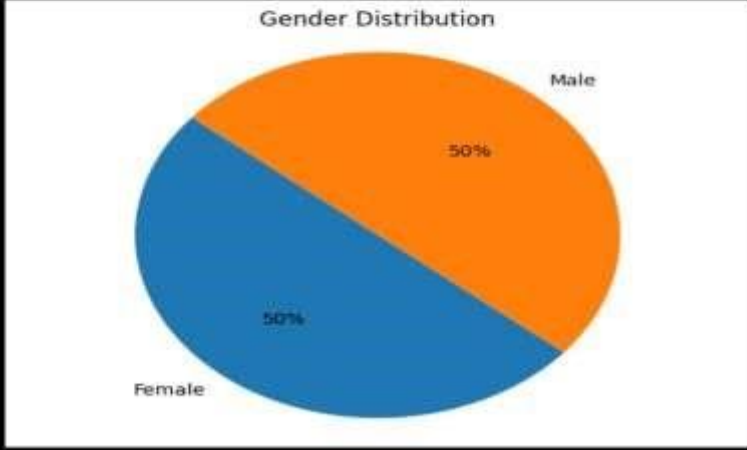
```
df.describe()
```

Python

	Gender	Age	History	Patient	TakeMedication	Severity	BreathShortness	VisualChanges	Nosebleeding	WhenDiagnosed	Systolic	Diastolic	ControlledDiet	Stages
count	1825	1825	1825	1825	1825	1825	1825	1825	1825	1825	1825	1825	1825	1825
unique	2	4	2	2	3	3	2	2	3	3	5	5	2	6
top	Female	51-64	Yes	No	No	Moderate	No	No	No	<1 Year	111 - 120	81 - 90	No	HYPERTENSION (Stage-1)
freq	913	475	1657	984	744	697	976	940	984	625	1008	708	984	648

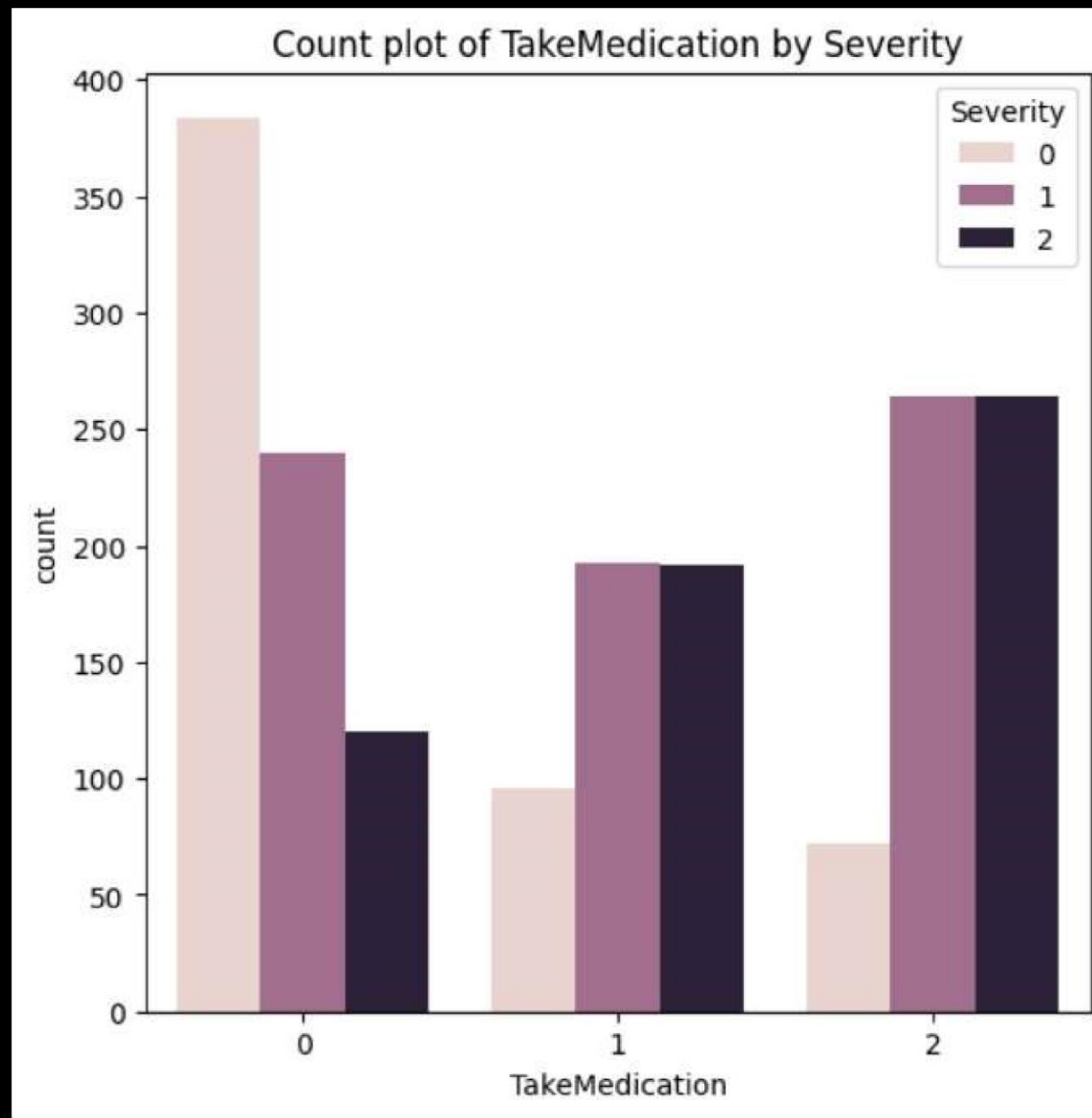
Data

Overview

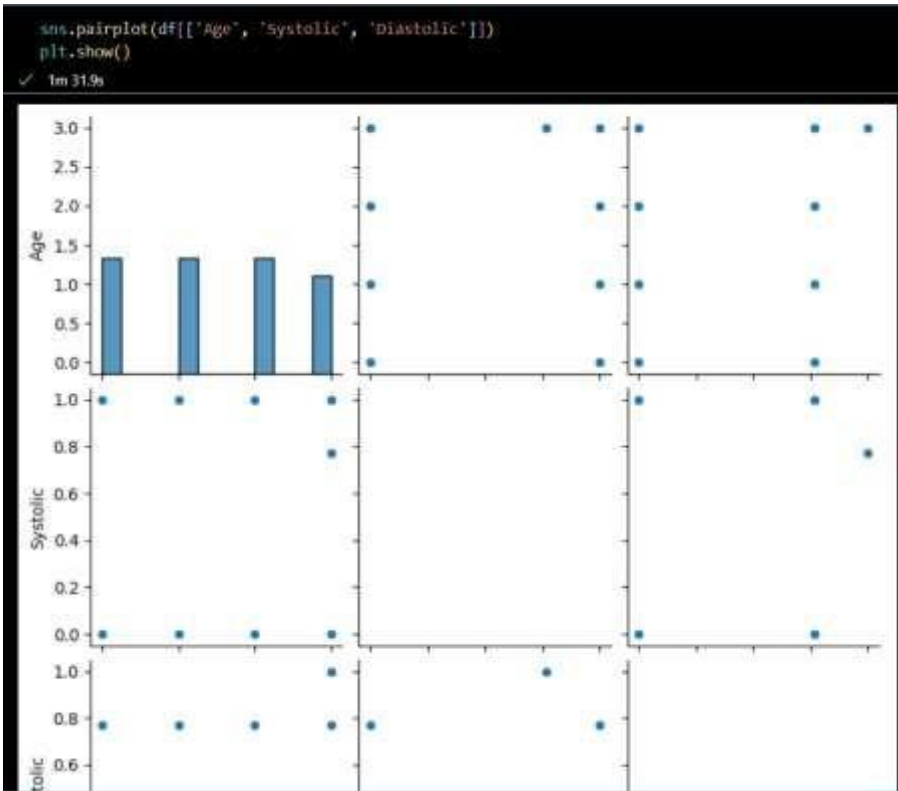
Univariate Analysis	<pre>gender_counts = df['Gender'].value_counts() # Plotting the pie chart plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.0f%%', startangle=140) plt.title('Gender Distribution') plt.axis('equal') plt.show()</pre>  <p>The pie chart, titled "Gender Distribution", illustrates an equal 50/50 split between Male and Female. The Male segment is colored orange and the Female segment is colored blue. Both segments are labeled with "50%".</p> <table border="1"><thead><tr><th>Gender</th><th>Percentage</th></tr></thead><tbody><tr><td>Male</td><td>50%</td></tr><tr><td>Female</td><td>50%</td></tr></tbody></table>	Gender	Percentage	Male	50%	Female	50%
Gender	Percentage						
Male	50%						
Female	50%						

```
sns.countplot(x='TakeMedication', hue='Severity', data=df)  
plt.title('Count plot of TakeMedication by Severity')  
plt.show()
```

✓ 0.1s



Bivariate
Analysis

Multivariate Analysis	
Outliers and Anomalies	-
Data Preprocessing Code Screenshots	

Loading Data

```
#importing data
df = pd.read_csv('patient_data.csv')
```

Python

```
df.head()
```

Python

	C	Age	History	Patient	TakeMedication	Severity	BreathShortness	VisualChanges	NoseBleeding	Whendiagnosed	Systolic	Diastolic	ControlledDiet	Stages
0	Male	18-34	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No	HYPERTENSION (Stage-1)
1	Female	18-34	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No	HYPERTENSION (Stage-1)
2	Male	35-50	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No	HYPERTENSION (Stage-1)
3	Female	35-50	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No	HYPERTENSION (Stage-1)
4	Male	51-64	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No	HYPERTENSION (Stage-1)

Handling

Missing Data

```
#checking for null values
df.isnull().sum()
```

```
Gender          0
Age             0
History         0
Patient         0
TakeMedication  0
Severity        0
BreathShortness 0
VisualChanges   0
NoseBleeding    0
Whendiagnosed   0
Systolic        0
Diastolic       0
ControlledDiet  0
Stages          0
dtype: int64
```

Data Transformation	<pre>#converting categorical into numerical value from sklearn.preprocessing import LabelEncoder columns = ['Gender' , 'Severity' , 'History' , 'Patient', 'TakeMedication', 'BreathShortness', 'VisualChanges', 'NoseBleeding', 'ControlledDiet', 'Stages'] label_encoder = LabelEncoder() for col in columns: df[col] = label_encoder.fit_transform(df[col])</pre>
------------------------	--

Feature Engineering	Attached the codes in final submission.
Save Processed Data	-