# Data Collection and Preprocessing Phase

| | |
|---|---|
| Date | 08 August 2025 |
| Skill Wallet ID | **SWUID20250188325** |
| Project Title | Predictive Pulse: Harnessing Machine Learning for Blood Pressure Analysis |
| Maximum Marks | 6 Marks |

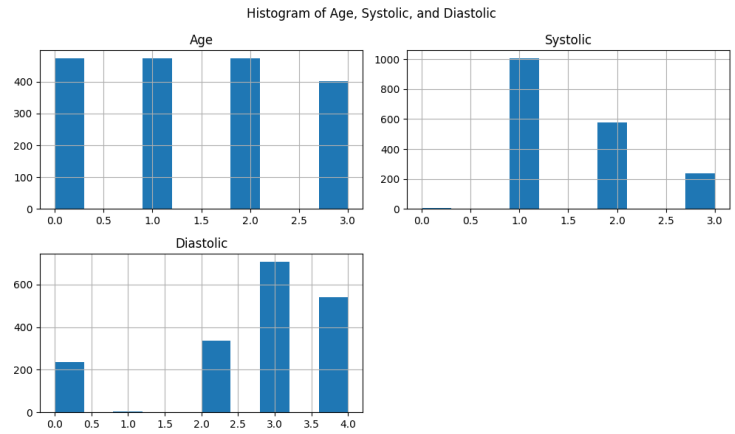
## Data Exploration and Preprocessing Report

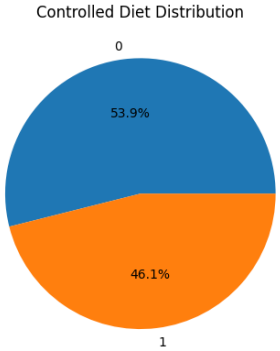
The dataset variables will be statistically analyzed to identify patterns and correlations in blood pressure stages, patient demographics, and medical history. Python will be employed for preprocessing tasks such as label encoding of categorical variables, normalization of numeric features, and feature engineering for model readiness.
Data cleaning will handle missing values, inconsistent data entries, and outliers to ensure high-quality inputs for subsequent analysis and model training — forming a strong foundation for accurate hypertension stage predictions.

| Section | Description |
|---|---|
| Data Overview | Dimension:<br>1825 rows × 14 columns<br>Descriptive statistics: |
| Univariate Analysis | |

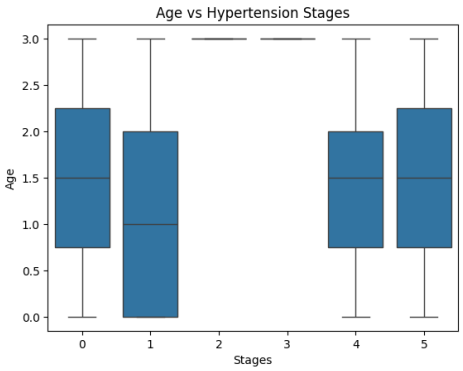
| | C | Age | History | Patient | TakeMedication | Severity | BreathShortness | VisualChanges | NoseBleeding | Whendiagnoused | Systolic | Diastolic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Male | 18-34 | Yes | No | No | Mild | No | No | No | <1 Year | 111 - 120 | 81 - 90 |
| 1 | Female | 18-34 | Yes | No | No | Mild | No | No | No | <1 Year | 111 - 120 | 81 - 90 |
| 2 | Male | 35-50 | Yes | No | No | Mild | No | No | No | <1 Year | 111 - 120 | 81 - 90 |
| 3 | Female | 35-50 | Yes | No | No | Mild | No | No | No | <1 Year | 111 - 120 | 81 - 90 |
| 4 | Male | 51-64 | Yes | No | No | Mild | No | No | No | <1 Year | 111 - 120 | 81 - 90 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1820 | Female | 35-50 | Yes | No | No | Sever | No | No | No | >5 Years | 111 - 120 | 70 - 80 |
| 1821 | Male | 51-64 | Yes | No | No | Sever | No | No | No | >5 Years | 111 - 120 | 70 - 80 |
| 1822 | Female | 51-64 | Yes | No | No | Sever | No | No | No | >5 Years | 111 - 120 | 70 - 80 |
| 1823 | Male | 65+ | Yes | No | No | Sever | No | No | No | >5 Years | 111 - 120 | 70 - 80 |
| 1824 | Female | 65+ | Yes | No | No | Sever | No | No | No | >5 Years | 111 - 120 | 70 - 80 |

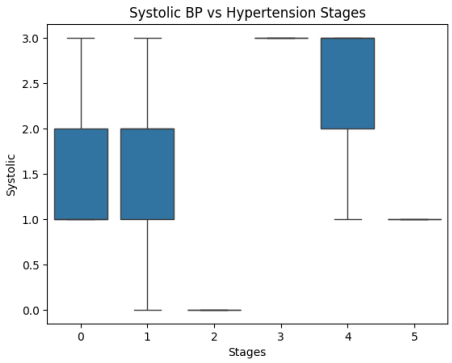
1825 rows × 14 columns

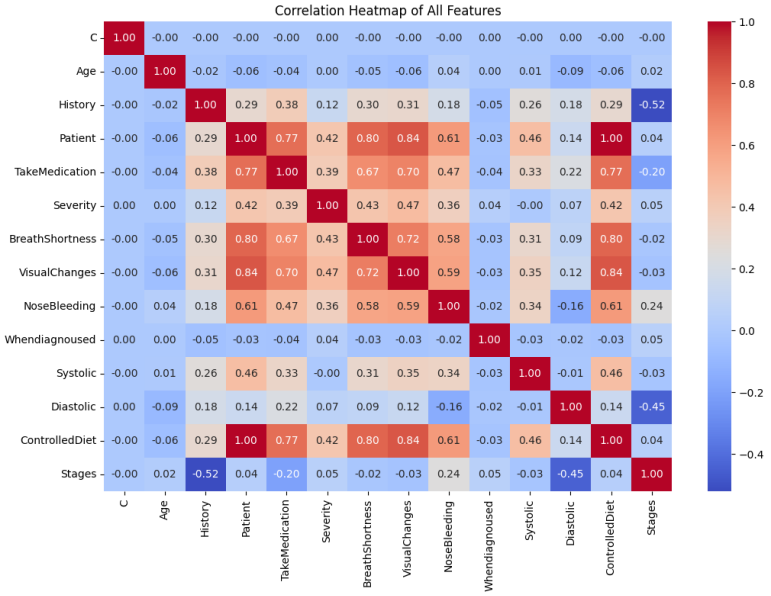



Bivariate Analysis



| | |
|---|---|
| Multivariate Analysis |  |
| Outliers and Anomalies | - |

## Data Preprocessing Code Screenshots

### Loading Data

```python
# Read the Dataset
df = pd.read_csv("patient_data.csv")
df.head()
```

Python

| | C | Age | History | Patient | TakeMedication | Severity | BreathShortness | VisualChanges | NoseB |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Male | 18-34 | Yes | No | No | Mild | No | No | |
| 1 | Female | 18-34 | Yes | No | No | Mild | No | No | |
| 2 | Male | 35-50 | Yes | No | No | Mild | No | No | |
| 3 | Female | 35-50 | Yes | No | No | Mild | No | No | |
| 4 | Male | 51-64 | Yes | No | No | Mild | No | No | |

### Handling Missing Data

```python
" Data Preprocessing - handling missing values "

# Replace blank strings and 'Unknown' with NaN
df.replace(r'^\s*$', np.nan, regex=True, inplace=True)
df.replace('Unknown', np.nan, inplace=True)

# View missing values in each column
print("Missing values before filling:\n", df.isnull().sum())

# Fill missing values
for column in df.columns:
    if df[column].isnull().sum() > 0:
        if df[column].dtype == 'object':
            df[column].fillna(df[column].mode()[0], inplace=True)  # Mode for categorical
        else:
            df[column].fillna(df[column].median(), inplace=True)  # Median for numeric

# Confirm missing values are handled
print("Missing values after filling:\n", df.isnull().sum())
```

```
Missing values before filling:
 C                   0
Age                 0
History             0
Patient             0
TakeMedication      0
Severity            0
BreathShortness     0
VisualChanges       0
NoseBleeding        0
Whendiagnoused      0
Systolic            0
Diastolic           0
ControlledDiet      0
Stages              0
dtype: int64
Missing values after filling:
 C                   0
Age                 0
History             0
Patient             0
TakeMedication      0
Severity            0
BreathShortness     0
VisualChanges       0
...
Diastolic           0
ControlledDiet      0
Stages              0
dtype: int64
```

| | |
|---|---|
| Data Transformation | ```from sklearn.preprocessing import LabelEncoder<br><br>categorical_columns = []<br>encoders = {}  # To store encoders for each column<br><br>for col in df.columns:<br>    if df[col].dtype == 'object':<br>        df[col] = df[col].str.strip()<br>        df[col] = df[col].replace("121- 130", "121 - 130")<br>        le = LabelEncoder()  # Create a new encoder for each column<br>        df[col] = le.fit_transform(df[col])<br>        encoders[col] = le  # Store the encoder<br>        print(f"Encoded {col} with classes: {le.classes_}")<br>        print(f"Unique values in {col}: {df[col].unique()}")<br>        categorical_columns.append(col)```<br><br>```Encoded C with classes: ['Female' 'Male']<br>Unique values in C: [1 0]<br>Encoded Age with classes: ['18-34' '35-50' '51-64' '65+']<br>Unique values in Age: [0 1 2 3]<br>Encoded History with classes: ['No' 'Yes']<br>Unique values in History: [1 0]<br>Encoded Patient with classes: ['No' 'Yes']<br>Unique values in Patient: [0 1]<br>Encoded TakeMedication with classes: ['No' 'Yes']<br>Unique values in TakeMedication: [0 1]<br>Encoded Severity with classes: ['Mild' 'Moderate' 'Sever']<br>Unique values in Severity: [0 2 1]<br>Encoded BreathShortness with classes: ['No' 'Yes']<br>Unique values in BreathShortness: [0 1]<br>Encoded VisualChanges with classes: ['No' 'Yes']<br>Unique values in VisualChanges: [0 1]<br>Encoded NoseBleeding with classes: ['No' 'Yes']<br>Unique values in NoseBleeding: [0 1]<br>Encoded Whendiagnoused with classes: ['1 - 5 Years' '<1 Year' '>5 Years']<br>Unique values in Whendiagnoused: [1 0 2]<br>Encoded Systolic with classes: ['100+' '111 - 120' '121 - 130' '130+']<br>Unique values in Systolic: [1 2 3 0]<br>Encoded Diastolic with classes: ['100+' '130+' '70 - 80' '81 - 90' '91 - 100']<br>Unique values in Diastolic: [3 4 0 1 2]<br>Encoded ControlledDiet with classes: ['No' 'Yes']<br>...<br>Encoded Stages with classes: ['HYPERTENSION (Stage-1)' 'HYPERTENSION (Stage-2)'<br> 'HYPERTENSION (Stage-2).' 'HYPERTENSIVE CRISI' 'HYPERTENSIVE CRISIS'<br> 'NORMAL']<br>Unique values in Stages: [0 1 4 2 3 5]``` |
| Feature Engineering | Attached the codes in final submission. |
| Save Processed Data | The “ Random Forest ” is used when maximum accuracy and robustness are required, it is Best for Analysis. |