DSC 680 Applied Data Science

Chitramoy Mukherjee, T301 T302 2253 Winter 2025

Project-1 Final White paper

HeartGuard: Early Detection of Heart Disease

Description:

This project aims to develop a predictive model leveraging patient data to identify individuals at high risk of heart disease, enabling early intervention and improving health outcomes.

Problem Statement:

Heart disease remains one of the leading causes of mortality worldwide. Early detection and intervention are critical to reduce the burden of heart disease, improve patient outcomes, and lower healthcare costs. Despite advancements in medical technology and awareness programs, many cases of heart disease go undetected until they reach a critical stage. Leveraging data science and machine learning can significantly enhance the ability to identify individuals at risk of developing heart disease at an early stage.

Nowadays, due to asymptomatic behavior of illnesses, detecting the early symptoms of many life-threatening diseases has become a major concern. One of the most common life-threatening conditions is “silent heart attack” i.e., an attack with few or less symptoms. One of the main reasons for silent attack is the flow of blood clots to the heart and additionally sometimes patients are not informed about the purpose of medication that is included in his or her treatment. So, timely detection of heart abnormalities is so crucial for the timely treatment, which can prevent further complications like heart attack and also helps in reducing the disease progression.

Accessing the comprehensive health data on CVD currently available within hospital databases holds significant potential for the early detection and diagnosis of CVD, thereby positively impacting disease outcomes. Machine learning algorithms on patients’ dataset can help in early detection of disease which further helps in developing preventive measures and also in enhancing patient health care considerations. Clinicians can use machine learning models to predict the heart diseases early and identify various common risk factors associated with heart like gender, age, family history, diabetes, increased cholesterol levels, hypertension, smoking etc. This model can also help in predicting the population or topology where risk factor is more and that can help in building necessary medical infrastructure to provide broader preventive measures.

By providing a means to develop evidence-based clinical guidelines and management algorithms, these techniques can eliminate the need for costly and extensive clinical and laboratory investigations, reducing the associated financial burden on patients and the

healthcare system. There are some other factors which can enhance the risk of getting heart disease.

Background:

Cardiovascular diseases (CVDs) remain a significant global health challenge and a leading cause of mortality worldwide. Heart disease remains a significant cause of mortality, responsible for approximately one in every four deaths globally. This emphasizes the critical need for accurate predictive models to address early detection and intervention. Early detection of heart disease is crucial for effective treatment and prevention. Machine learning has emerged as a vital tool in healthcare due to its potential to enhance prediction accuracy. This project presents a comprehensive framework for heart disease prediction using advanced machine learning techniques in R. Several factors like High Blood pressure, High cholesterol, Smoking, Diabetes are major reasons for increased heart diseases.There are some other factors which can enhance the risk of getting heart disease.

• Family History: Early cardiovascular disease in close relatives (men < 55 years, women < 65 years).

• Metabolic Syndrome: A cluster of conditions (high blood pressure, high blood sugar, excess abdominal fat, abnormal cholesterol levels).

• Chronic Kidney Disease: Impaired kidney function affects overall health.

• Chronic Inflammatory Conditions: Conditions like rheumatoid arthritis or psoriasis.

• History of Preeclampsia or Early Menopause: These may indicate increased risk.

• High-Risk Ethnicity: Certain ethnic backgrounds may have higher susceptibility.

• Higher Triglycerides: Elevated levels contribute to heart disease risk.

Data Explanation and Data Dictionary:

Here I am using 3 datasets as below:

Heart_Disease_Classification_Dataset.csv

(n.d.), Heart Disease Classification Dataset.
https://www.gigasheet.com/sampledata/heart-disease-classification-dataset

The table, named Heart Disease Classification Dataset, has 1319 rows and 9 columns representing variables such as age, gender, blood pressure, glucose levels, and heart related metrics like troponin.

Heart_Disease_Classification_Dataset.csv

Column Name	Description
age	Age of the patient (in years)
gender	Gender of the patient (Male/Female)
impluse	Heart rate (beats per minute)
pressurehight	Systolic blood pressure (mmHg)
pressurelow	Diastolic blood pressure (mmHg)
glucose	Blood glucose level (mg/dL)
kcm	Ketone concentration in blood (mmol/L)
troponin	Troponin level, a marker of heart muscle damage (ng/mL)
class	Classification of heart disease (Normal, Mild, Moderate, Severe)

The table, named Heart Disease Classification Dataset, has 1319 rows and 9 columns representing variables such as age, gender, blood pressure, glucose levels, and heart-related metrics like troponin.

```
# Load the Heart Disease Classification Dataset csv file
heart_disease_classification_data = pd.read_csv('C:\\Users\\Chitramoy\\Desktop\\MS-DSC\\DSC-680\\Project-1\\Heart_Disease_Classification_Dataset.csv')
```

Cardiovascular_Disease_Dataset.csv

Bhanu P.D, Debnath B (2021, April 16). Cardiovascular_Disease_Dataset. https://data.mendeley.com/datasets/dzz48mvjht/1

This heart disease dataset is acquired from one of the multispecialty hospitals in India. This dataset consists of 1000 rows with 14 features/columns.

Cardiovascular_Disease_Dataset.csv

Column Name	Description
patientid	Unique identifier for each patient
age	Age of the patient (in years)
gender	Gender of the patient (Male/Female)
chestpain	Type of chest pain experienced (e.g., typical angina, atypical angina, non-anginal)
restingBP	Resting blood pressure (in mmHg)
serumcholestrol	Serum cholesterol level (in mg/dL)
fastingbloodsugar	Fasting blood sugar level (1 = true; 0 = false)
restingrelectro	Resting electrocardiographic results (e.g., normal, ST-T wave abnormality)
maxheartrate	Maximum heart rate achieved (in beats per minute)
exerciseangia	Exercise-induced angina (1 = yes; 0 = no)
oldpeak	ST depression induced by exercise relative to rest
slope	The slope of the peak exercise ST segment (e.g., upsloping, flat, downsloping)
noofmajorvessels	Number of major vessels colored by fluoroscopy
target	Diagnosis of heart disease (1 = heart disease; 0 = no heart disease)

This heart disease dataset is acquired from one of the multispecialty hospitals in India.This dataset consists of 1000 rows with 14 features/columns.

```
# Load the Cardiovascular Disease Dataset csv file
cardiovascular_disease_data =pd.read_csv('C:\\Users\\Chitramoy\\Desktop\\MS-DSC\\DSC-680\\Project-1\\Cardiovascular_Disease_Dataset.csv')
```

Heart_disease.csv

Mirza Hasnine (2023). Heart Disease Dataset. https://www.kaggle.com/datasets/mirzahasnine/heart-disease-dataset/data

This heart disease dataset is a combination of multiple datasets and is a big subset including 4238 rows of individual data with 16 features/columns.

heart_disease.csv

Column Name	Description
Gender	Gender of the patient (Male/Female)
age	Age of the patient (in years)
education	Education level of the patient (e.g., primary, secondary, tertiary)
currentSmoker	Current smoking status (1 = smoker; 0 = non-smoker)
cigsPerDay	Number of cigarettes smoked per day
BPMeds	Blood pressure medication usage (1 = yes; 0 = no)
prevalentStroke	History of stroke (1 = yes; 0 = no)
prevalentHyp	History of hypertension (1 = yes; 0 = no)
diabetes	Diabetes status (1 = yes; 0 = no)
totChol	Total cholesterol level (mg/dL)
sysBP	Systolic blood pressure (mmHg)
diaBP	Diastolic blood pressure (mmHg)
BMI	Body Mass Index (kg/m^2)
heartRate	Heart rate (beats per minute)
glucose	Blood glucose level (mg/dL)
Heart_ stroke	Presence of heart disease or stroke (1 = yes; 0 = no)

This heart disease dataset is a combination of multiple datasets and is a big subset including 4238 rows of individual data with 16 features/columns.

```
# Load the Heart Disease Dataset csv file
heart_disease_data = pd.read_csv('C:\\Users\\Chitramoy\\Desktop\\MS-DSC\\DSC-680\\Project-1\\heart_disease.csv')
```

Load the Final Heart diseases dataset.

```
# Load the Final Heart Disease Dataset csv file
heart_disease_final_data = pd.read_csv('C:\\Users\\Chitramoy\\Desktop\\MS-DSC\\DSC-680\\Project-1\\heart_disease_final_data.csv')
```

An exploratory data analysis was done on these datasets along with data preparation. Below steps were taken to remove outliers and prepare the final set.

Methods/Steps:

For the **Early Detection of Heart Disease** project, we will employ a variety of analysis methods to ensure a comprehensive and accurate predictive model. Here are the key methods:

1. Data Preparation:

- Obtain datasets with relevant features (such as patient demographics, medical history, and diagnostic test results).
- Clean the data by handling missing values, outliers, and ensuring consistent formatting.
- Transform columns and merge datasets to create a final subset of clean data.

2. Feature Selection and Engineering:

- Identify relevant features that may contribute to heart failure prediction.
- Based on that, identify predictors which can later be used in generating models. Example: Gender, High blood pressure count etc.
- Create new features if needed (e.g., age groups, risk scores, or gender categorization).

3. Exploratory Data Analysis (EDA):

- Explore relationships between features/predictors and the target variable (presence/absence of heart disease).
- Visualize data distributions, correlations, and patterns.

4. Machine Learning Algorithms:

- Logistic Regression: For binary classification of heart disease risk.
- Random Forest: To handle nonlinear relationships and provide feature importance.
- Gradient Boosting Machines (GBM): For improving accuracy with ensemble methods.
- Support Vector Machines (SVM): For classification with high-dimensional data.
- Neural Networks: For capturing complex patterns in large datasets.

5. Train-Test Split:

- Divide the dataset into training and testing subsets (e.g., 70% training, 30% testing).

6. Model Training and Evaluation:

- Train the selected models on the training data.
- Evaluate model performance using metrics like accuracy, precision, recall, F1-score etc.
- Use cross-validation to prevent overfitting.

7. Model Comparison:

- Compare different models based on evaluation metrics.
- Select the best-performing model.

8. Deployment and Monitoring:

- Deploying the final model into a production environment.
- Continuously monitoring model performance and updating it as needed.

Analysis:

Inspect the features, remove outliers and summarize the data

Examine the features (columns) in each dataset to identify common features that can be used for merging.Ensure that the target variable (whether a patient has heart disease) is consistent across all 3 datasets and is binary.Ensure for common features, column names are same in all 3 datasets.Ensure common feature data types are same. A comprehensive analysis is undertaken on both the target and the features, and category variables are converted to numeric values (Eg. heart stroke 'Happened' or 'Not Happened').

After removing outliers and merging all 3 datasets, final dataset is generated which contains below common features:

Age: This column contains the ages of individuals in your dataset.

Gender: The gender of each individual (either "Male" or "Female").

Blood Pressure: The blood pressure measurements (in mmHg) for each person.

Heart Stroke: A binary variable indicating whether an individual had a heart stroke (1 for yes, 0 for no).

[37]:
```
# Load the Final Heart Disease Dataset csv file
heart_disease_final_data = pd.read_csv('C:\\Users\\Chitramoy\\Desktop\\MS-DSC\\DSC-680\\Project-1\\heart_disease_final_data.csv')
```

[45]:
```
# Preview the dataframe heart_disease_final_data
heart_disease_final_data.head(5)
```

[45]:

	age	gender	blood_pressure	heart_stroke
0	64	Male	160	0
1	55	Male	160	0
2	64	Male	120	1
3	55	Male	112	0
4	58	Female	112	0

[49]:
```
# Total record count
len(heart_disease_final_data)
```

[49]:
```
6323
```

[51]:
```
#Show top 30 rows
heart_disease_final_data.head(30)
```

[51]:

	age	gender	blood_pressure	heart_stroke
0	64	Male	160	0
1	55	Male	160	0
2	64	Male	120	1
3	55	Male	112	0
4	58	Female	112	0
5	32	Female	179	0
6	63	Male	214	1
7	44	Female	154	0
8	67	Male	160	0
9	44	Female	166	0
10	63	Female	150	0
11	64	Male	199	1
12	54	Female	122	0
13	47	Male	120	0
14	61	Male	118	1
15	86	Female	114	1
16	45	Female	100	0
17	37	Female	107	0
18	45	Male	109	1
19	60	Male	151	0
20	48	Male	98	1
21	52	Male	109	1
22	30	Male	110	1
23	50	Male	104	1
24	72	Male	106	1
25	42	Male	150	0
26	72	Female	152	0
27	47	Female	134	1
28	63	Male	135	0
29	54	Male	131	1

```
[53]: ## display summary of the dataset using describe function
      heart_disease_final_data.describe(include='all')
```

[53]:

	age	gender	blood_pressure	heart_stroke
count	6323.000000	6323	6323.000000	6323.000000
unique	NaN	2	NaN	NaN
top	NaN	Male	NaN	NaN
freq	NaN	3278	NaN	NaN
mean	51.840266	NaN	133.672624	0.303021
std	10.897241	NaN	25.029424	0.459600
min	30.000000	NaN	42.000000	0.000000
25%	43.000000	NaN	116.000000	0.000000
50%	51.000000	NaN	129.000000	0.000000
75%	60.000000	NaN	147.000000	1.000000
max	103.000000	NaN	295.000000	1.000000

Split the data into Training and Testing data

In this project, 70% of data has been considered as Training data and 30% Testing data. Analysis of data will be performed on Training data. Testing data will be used later for performance evaluation of model.

```
# Generating training indices
train_indices = np.random.choice(
    heart_disease_final_data.index,
    size=int(0.7 * len(heart_disease_final_data)),
    replace=False
)

# Splitting the dataset
heart_disease_final_data_train = heart_disease_final_data.loc[train_indices]
heart_disease_final_data_test = heart_disease_final_data.drop(train_indices)
```

Below analysis has been conducted in this project:

1. Summarized data is showing total number of heart strokes for Male and Female. 0 means 'No' heart_stroke and 1 means 'Yes' heart_stroke. Male count of heart strokes is more than Female count as per the data.

```
# Further Summarizing the data by creating a contingency table
# Display the contingency table
```

```
# Creating a contingency table
contingency_table = pd.crosstab(
    heart_disease_final_data_train['gender'],
    heart_disease_final_data_train['heart_stroke']
)

print(contingency_table)
```

```
heart_stroke     0    1
gender
Female        1656  466
Male          1429  875
```

Calculated Mean/Average age of Male and Female population getting heart strokes.

a) Mean age of Male individuals with heart strokes: 56 years

b) Mean age of Female individuals with heart strokes: 58 years

```
print(f"Mean age of Male individuals with heart strokes: {mean_age_heart_stroke:.2f} years")
```

```
Mean age of Male individuals with heart strokes: 56.16 years
```

```
mean_age_heart_stroke = heart_disease_final_data_train.loc[
    (heart_disease_final_data_train['heart_stroke'] == 1) &
    (heart_disease_final_data_train['gender'] == 'Female'),
    'age'
].mean(skipna=True)
```

```
print(f"Mean age of Female individuals with heart strokes: {mean_age_heart_stroke:.2f} years")
```

```
Mean age of Female individuals with heart strokes: 58.01 years
```

3. The histogram plot is used to visualize the heart stroke count based on age and gender. Overall Male count of getting heart stroke is higher than Female count. Male with age group 50 - 70 years old have encountered maximum heart strokes. Male plotting is a bimodal distribution. Female plotting is a normal distribution and Female with age group 60 years old have encountered maximum heart strokes as compared to other female population.

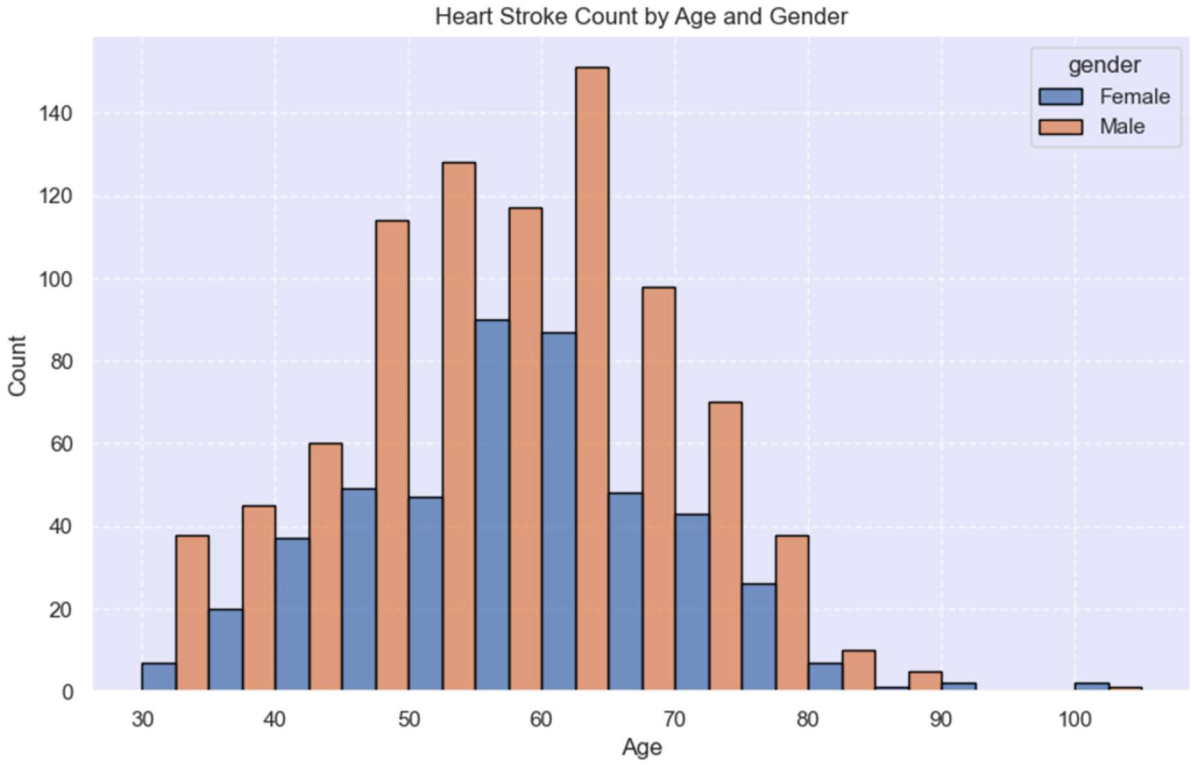
   The second plot is showing that in 'Male' gender, heart stroke is more prevalent with blood pressure > 150 (in mm HG) where 'Female' gender is not showing significant correlation with increased blood pressure. This visualization indicates that chances of having heart stroke is high in 'Male' gender with increased blood pressure.

   The third scatterplot visualizes the relationship between age and blood pressure, with points color-coded by heart stroke status and styled by gender. This allows for identifying patterns or trends, such as how age and blood pressure vary with stroke risk, while distinguishing between genders. The "coolwarm" palette highlights variations in stroke presence.

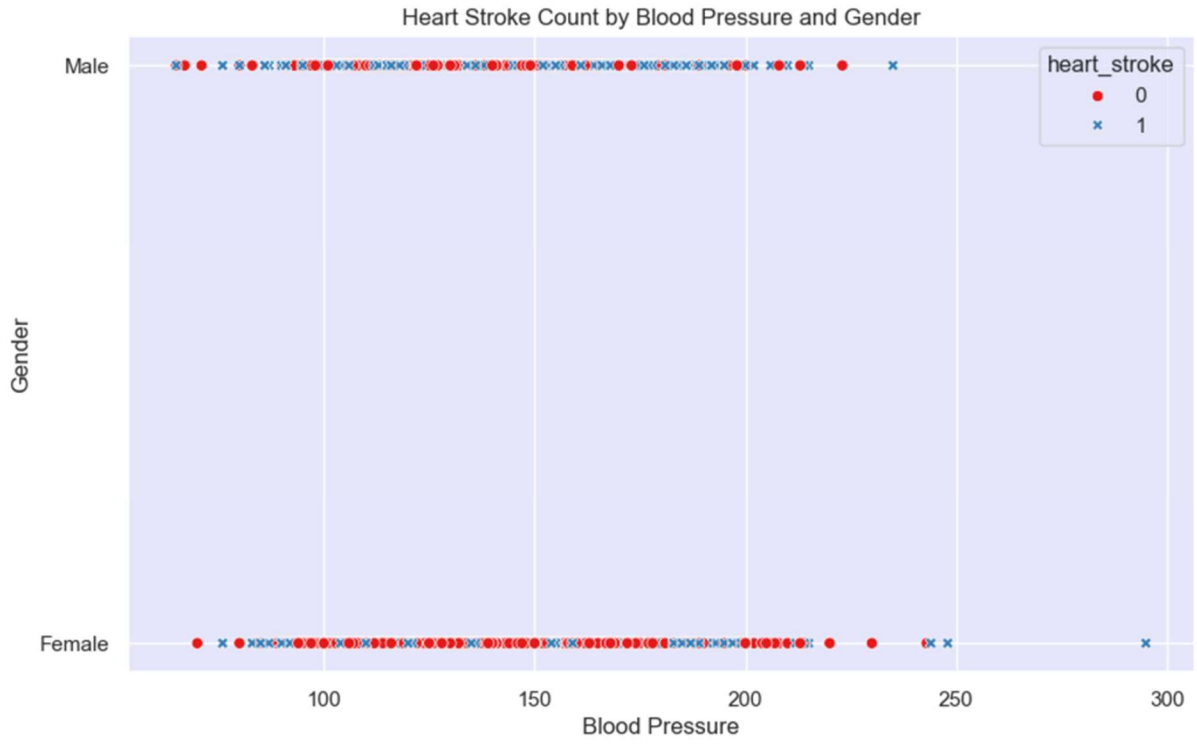
The counterplot displays the distribution of heart stroke cases by gender, with bars segmented by stroke status (e.g., presence or absence of a stroke) using the "Set2" color palette. It provides a clear comparison of stroke occurrence across genders. This visualization helps identify any gender-based differences in heart stroke prevalence.

The boxplot illustrates the distribution of blood pressure values for each gender, highlighting key statistics like median, interquartile range, and potential outliers. Using the "pastel" palette, it visually compares blood pressure trends between genders. This helps in identifying gender-based differences in blood pressure levels.

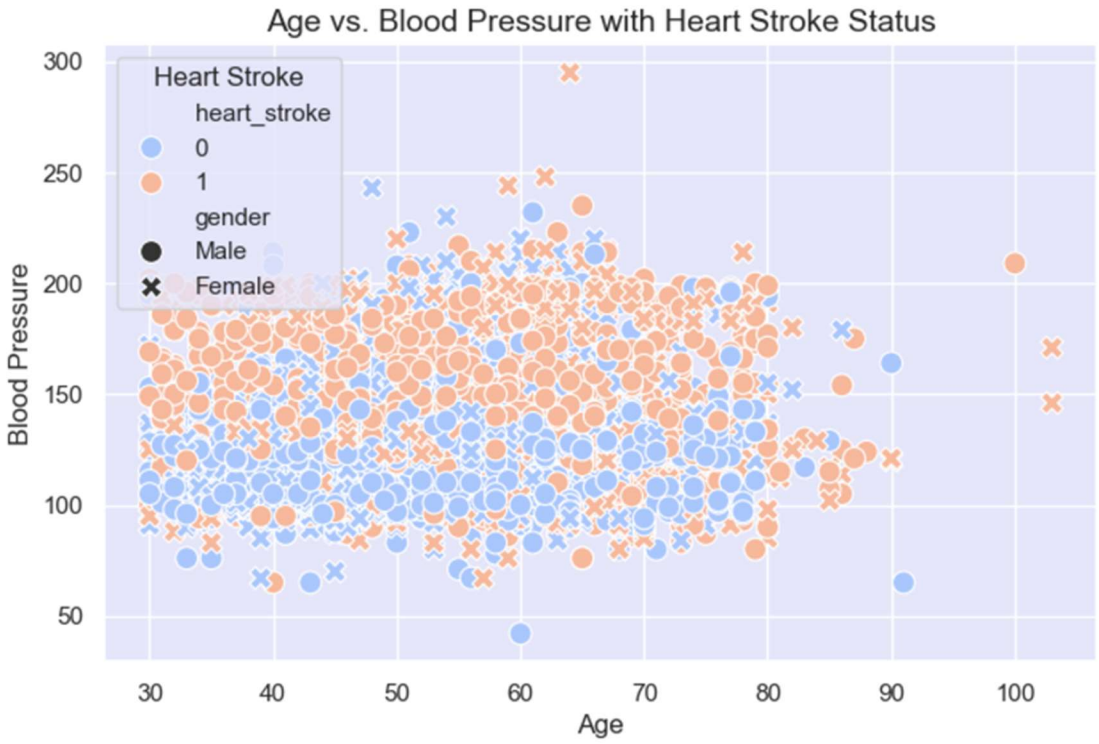
```
# Create a histogram to show heart stroke count by age and gender
plt.figure(figsize=(10, 6))
sns.histplot(
    data=with_heart_stroke_data,
    x='age',
    hue='gender',
    bins=range(int(with_heart_stroke_data['age'].min()), int(with_heart_stroke_data['age'].max()) + 5, 5),
    multiple='dodge',
    edgecolor='black'
)

# Add labels and title
plt.title("Heart Stroke Count by Age and Gender")
plt.xlabel("Age")
plt.ylabel("Count")
plt.grid(visible=True, linestyle='--', alpha=0.7)
plt.show()
```



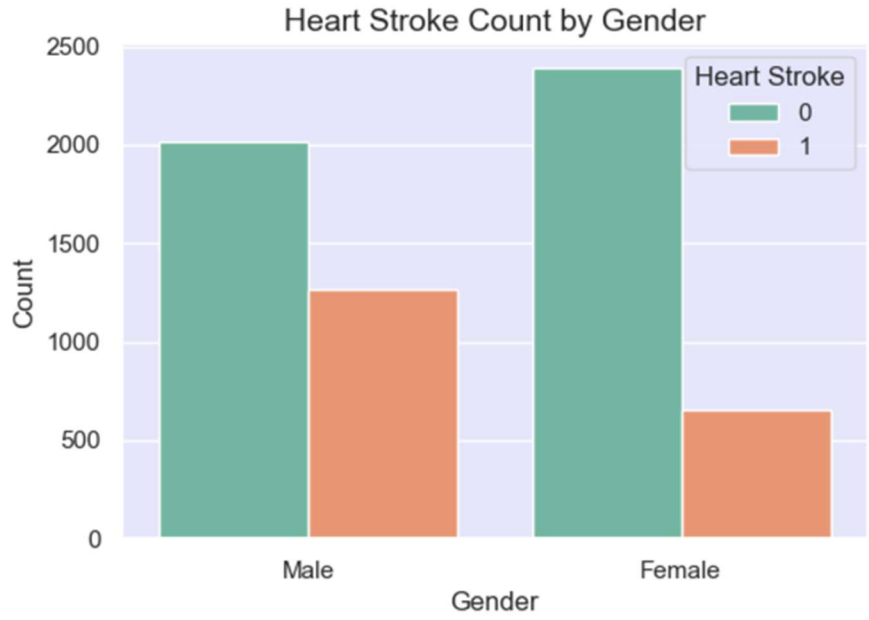
```
# Create a plot to show heart stroke count by blood pressure and gender
plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=heart_disease_final_data_train,
    x='blood_pressure',
    y='gender',
    hue='heart_stroke',
    palette='Set1',
    style='heart_stroke',
    legend='full'
)

# Add labels and title
plt.title("Heart Stroke Count by Blood Pressure and Gender")
plt.xlabel("Blood Pressure")
plt.ylabel("Gender")
plt.show()
```



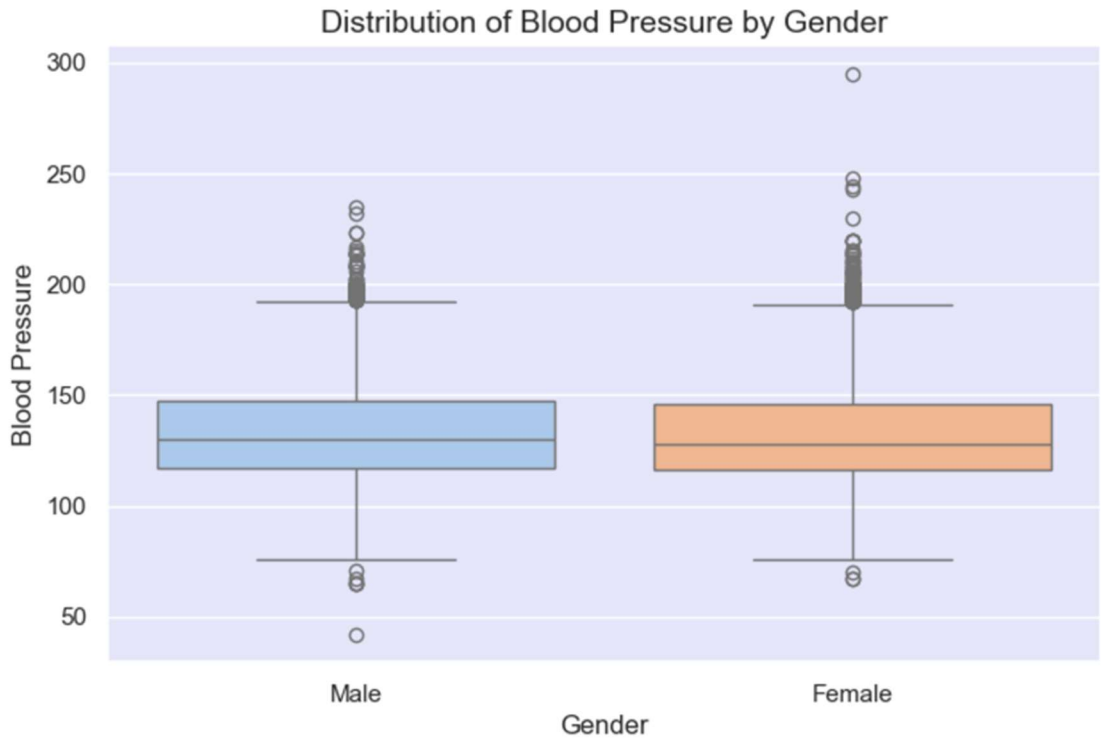

```
plt.figure(figsize=(8, 5))
sns.scatterplot(
    x='age',
    y='blood_pressure',
    hue='heart_stroke',
    style='gender',
    data=heart_disease_final_data,
    palette='coolwarm',
    s=100
)
plt.title('Age vs. Blood Pressure with Heart Stroke Status', fontsize=14)
plt.xlabel('Age', fontsize=12)
plt.ylabel('Blood Pressure', fontsize=12)
plt.legend(title='Heart Stroke', loc='upper left')
plt.show()
```




```
plt.figure(figsize=(6, 4))
sns.countplot(x='gender', hue='heart_stroke', data=heart_disease_final_data, palette='Set2')
plt.title('Heart Stroke Count by Gender', fontsize=14)
plt.xlabel('Gender', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.legend(title='Heart Stroke', loc='upper right')
plt.show()
```



```
plt.figure(figsize=(8, 5))
sns.boxplot(x='gender', y='blood_pressure', data=heart_disease_final_data, palette='pastel')
plt.title('Distribution of Blood Pressure by Gender', fontsize=14)
plt.xlabel('Gender', fontsize=12)
plt.ylabel('Blood Pressure', fontsize=12)
plt.show()
```



4. Identified correlation between predictors and target variable. Age and blood_pressure are explanatory (predictor) variables and heart_stroke is a binary target variable.

a) The correlation coefficient between age and blood_pressure is -0.01043512, which indicates a negative correlation between the two variables.

b) The Pearson correlation coefficient between blood pressure and heart strokes is 0.21 which indicates a positive linear relationship between blood pressure and the occurrence of heart strokes.

c) The Pearson correlation coefficient between age and heart strokes is 0.29 which indicates a positive linear relationship between age and the occurrence of heart strokes.

```
# Creating a correlation matrix
correlation_matrix = with_heart_stroke_data[['age', 'blood_pressure']].corr()

# Printing the correlation matrix
print(correlation_matrix)
```

```
                    age  blood_pressure
age            1.000000       -0.006312
blood_pressure -0.006312        1.000000
```

```
# Computing the correlation between blood_pressure and heart_stroke
correlation = heart_disease_final_data_train['blood_pressure'].corr(heart_disease_final_data_train['heart_stroke'])

# Printing the correlation value
print(correlation)
```

```
0.20166864936592918
```

```
# Computing the correlation between age and heart_stroke
correlation_age = heart_disease_final_data_train['age'].corr(heart_disease_final_data_train['heart_stroke'])

# Display the correlation coefficient
print(f"Pearson Correlation Coefficient: {correlation_age:.2f}")
```

```
Pearson Correlation Coefficient: 0.29
```

5. A binary logistic regression model got generated for the final dataset. The model aims to predict the occurrence of heart strokes based on several predictor variables. The response variable is heart_stroke (binary outcome: yes/no). The predictors include age, gender, and blood pressure.

```
# Encode 'gender' as numeric
le = LabelEncoder()
heart_disease_final_data['gender'] = le.fit_transform(heart_disease_final_data['gender'])

# Check for missing values and drop them if present
heart_disease_final_data = heart_disease_final_data.dropna()

# Define features (X) and target (y)
X = heart_disease_final_data[['age', 'gender', 'blood_pressure']]
y = heart_disease_final_data['heart_stroke']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
# Train Logistic Regression model
log_model = LogisticRegression()
log_model.fit(X_train, y_train)

# Predict and evaluate
y_pred_log = log_model.predict(X_test)
accuracy_log = accuracy_score(y_test, y_pred_log)
print("Logistic Regression:")
print(f"Accuracy: {accuracy_log:.2f}")
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_log))
print("Classification Report:")
print(classification_report(y_test, y_pred_log))
```

```
Logistic Regression:
Accuracy: 0.72
Confusion Matrix:
[[1219   88]
 [ 450  140]]
Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.93      0.82      1307
           1       0.61      0.24      0.34       590

    accuracy                           0.72      1897
   macro avg       0.67      0.58      0.58      1897
weighted avg       0.69      0.72      0.67      1897
```

6. A Support Vector Machine model got generated for the final dataset. The model aims to predict the occurrence of heart strokes based on several predictor variables.

```
# Train SVM model
svm_model = SVC(kernel='linear', random_state=42)
svm_model.fit(X_train, y_train)

# Predict and evaluate
y_pred_svm = svm_model.predict(X_test)
accuracy_svm = accuracy_score(y_test, y_pred_svm)
print("Support Vector Machine:")
print(f"Accuracy: {accuracy_svm:.2f}")
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_svm))
print("Classification Report:")
print(classification_report(y_test, y_pred_svm))
```

```
Support Vector Machine:
Accuracy: 0.69
Confusion Matrix:
[[1307    0]
 [ 590    0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.69      1.00      0.82      1307
           1       0.00      0.00      0.00       590

    accuracy                           0.69      1897
   macro avg       0.34      0.50      0.41      1897
weighted avg       0.47      0.69      0.56      1897
```

7. A Random Forest model got generated for the final dataset. The model aims to predict the occurrence of heart strokes based on several predictor variables.

```
# Train Random Forest model
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)

# Predict and evaluate
y_pred_rf = rf_model.predict(X_test)
accuracy_rf = accuracy_score(y_test, y_pred_rf)
print("Random Forest:")
print(f"Accuracy: {accuracy_rf:.2f}")
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_rf))
print("Classification Report:")
print(classification_report(y_test, y_pred_rf))
```

```
Random Forest:
Accuracy: 0.71
Confusion Matrix:
[[1069  238]
 [ 318  272]]
Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.82      0.79      1307
           1       0.53      0.46      0.49       590

    accuracy                           0.71      1897
   macro avg       0.65      0.64      0.64      1897
weighted avg       0.70      0.71      0.70      1897
```

Comparison of the outcome of 3 different model.

```
# Display the accuracy of all models
print("Model Comparison:")
print(f"Logistic Regression Accuracy: {accuracy_log:.2f}")
print(f"Random Forest Accuracy: {accuracy_rf:.2f}")
print(f"Support Vector Machine Accuracy: {accuracy_svm:.2f}")
```

```
Model Comparison:
Logistic Regression Accuracy: 0.72
Random Forest Accuracy: 0.71
Support Vector Machine Accuracy: 0.69
```

Explanation of the Model outcome:

1. Accuracy

- The **accuracy** of the logistic regression model is **72%**, meaning the model correctly predicts whether a person had a heart stroke or not 72% of the time.
- The **accuracy** of the SVM is **69%**, meaning the model correctly predicts whether a person had a heart stroke or not 69% of the time.
- The **accuracy** of the Random Forest Model is **71%**, meaning the model correctly predicts whether a person had a heart stroke or not 71% of the time.

2. Confusion Matrix for Logistic Regression Model

- **True Negatives (1219)**: The model correctly identified 1219 people without a heart stroke.
- **False Positives (88)**: The model incorrectly predicted a heart stroke for 88 people who did not have one.
- **False Negatives (450)**: The model missed 450 actual heart stroke cases, predicting them as no stroke.
- **True Positives (140)**: The model correctly identified 140 people with a heart stroke.

3. Classification Report for Logistic Regression Model

- **Class 0 (No Stroke)**:
    - **Precision (73%)**: Out of all predictions of "No Stroke," 73% were correct.
    - **Recall (93%)**: The model identified 93% of the actual "No Stroke" cases.
    - **F1-Score (82%)**: A balanced measure of precision and recall for the "No Stroke" class.
- **Class 1 (Stroke)**:
    - **Precision (61%)**: Out of all predictions of "Stroke," 61% were correct.
    - **Recall (24%)**: The model only identified 24% of the actual "Stroke" cases.

- **F1-Score (34%)**: The model struggles to balance precision and recall for the "Stroke" class.

4. Macro and Weighted Averages for Logistic Regression Model

- **Macro Avg**: The unweighted average of precision, recall, and F1-score across both classes:
  - Precision: 67%
  - Recall: 58%
  - F1-Score: 58%
- **Weighted Avg**: Takes class imbalance into account:
  - Precision: 69%
  - Recall: 72%
  - F1-Score: 67%

Conclusion:

The model performs well for "No Stroke" predictions but struggles significantly with "Stroke" predictions due to **class imbalance** and **low recall for Class 1 (Stroke)**.

Actionable Steps:

1. Address **class imbalance** using techniques like oversampling the minority class (e.g., SMOTE) or undersampling the majority class.
2. Tune the decision threshold to improve recall for stroke cases.
3. Consider alternative models like **Random Forest** or **Gradient Boosting** that can better handle imbalanced datasets.

This output highlights the importance of evaluating models beyond accuracy, focusing on precision, recall, and their implications for different classes.

Limitations:

Based on the summary statistics of this model and correlation data, this might not be an ideal model to identify key factors of heart disease. There can be other factors/predictors which can be potential indicators of heart disease. Challenges in generating better model are as follows:

1. Merging multiple different datasets is challenging when there are less common features. In this case, the final dataset is containing very less predictors [age, gender and blood_pressure] which might not provide clear insight on early detection of heart disease.

2. To generate better model,this will require collecting more datasets so that the final dataset can contain multiple features with relevant data.

Ethics/Challenges:

Although AI has the potential to transform cardiovascular disease analysis by leveraging large amounts of data from clinical trial data, medical records, or data generated from sensors such as ECG rhythm monitors, there are huge concerns related to the likelihood for biased predictions among women and underserved minorities. I think it is a slippery slope as these groups are underrepresented in cardiology, and most of the clinical evidence and guideline recommendations in the field are largely based on clinical trials that frequently exclude these patient populations.

1. **Data Quality:**

   Incompleteness: Missing values in crucial variables, which can bias the analysis.
   Inconsistency: Variability in data formats and measurement units across sources.

2. **Data Accessibility:**

   Privacy Regulations: Restrictions due to laws like HIPAA can limit access to detailed patient data.
   Data Silos: Health data may be fragmented across different institutions and systems, making it hard to consolidate.

3. **Data Volume and Complexity:**

   High Dimensionality: Large number of features and complex relationships can complicate model training and interpretation.
   Scalability: Handling large datasets efficiently requires significant computational resources.

Future Uses/Additional Applications:

Effectively leveraging AI to improve health outcomes requires an understanding of the potential and purpose of ML algorithms. More cardiovascular screenings can be arranged to gather more data across multiple demographic areas. Like Mammogram bus, healthcare units can offer CVD screening centers or mobile bus for larger population. Data will be collected from CVD screenings, medical records (ECG reports) and smart digital tools which will feed into ML algorithms and get classified based on multiple factors [Age, Gender, Blood Pressure, Cholesterol, Diabetes, Menopause, BMI etc.]. This life course approach can facilitate identifying

and addressing early detection, with many opportunities for improvement that can be potentially aided by AI.

Advantages

1. Early diagnosis of heart failure based on pattern match and family history.
2. Optimal medical therapy and early treatment can be provided.
3. Identification of race and demographic area which can be at potential risk for heart diseases based on the model results.
4. AI can identify coronary plaques more accurately than clinicians.
5. Faster reading and interpretation of ECG records which is not possible by a clinician manually.
6. It can be used to assess structural diseases, such as valvular disease, to help determine the classification and staging of the disease.
7. Using machine learning techniques, the cost of conducting a long list of expensive clinical and laboratory investigations will be eliminated, reducing the financial burden on patients and the healthcare system.

Recommendations:

Regular Screenings:

- Encourage regular cardiovascular screenings, especially for individuals over the age of 40 or those with a family history of heart disease.
- Utilize non-invasive tests such as ECGs, stress tests, and echocardiograms to detect early signs of heart disease.

Healthy Lifestyle:

- Promote a heart-healthy diet rich in fruits, vegetables, whole grains, and lean proteins.
- Advise regular physical activity, aiming for at least 150 minutes of moderate exercise per week.
- Encourage maintaining a healthy weight, managing stress, and avoiding smoking and excessive alcohol consumption.

Medical Management:

- Ensure patients adhere to prescribed medications, including antihypertensives, statins, and antiplatelets.

- Regularly monitor and control risk factors such as high blood pressure, high cholesterol, and diabetes.

Public Awareness:

- Raise public awareness about the importance of early detection and the common signs and symptoms of heart disease.
- Organize community outreach programs and workshops to educate people on heart disease prevention and early detection methods.

Technology Integration:

- Leverage wearable technology and mobile apps to monitor vital signs and provide real-time health data to both patients and healthcare providers.
- Use telemedicine services to facilitate regular follow-ups and consultations, especially for those in remote or underserved areas.

By implementing these recommendations, HeartGuard can significantly contribute to the early detection and prevention of heart disease, ultimately saving lives and improving the quality of healthcare.

Implementation Plan:

Training and Education

- Staff Training: Train healthcare providers on how to use the HeartGuard system effectively and interpret the model's predictions.
- Patient Education: Inform patients about the benefits of early detection and how the HeartGuard system works.

Monitoring and Maintenance

- **Continuous Monitoring**: Regularly monitor the model's performance and update it with new data to maintain its accuracy.
- **Feedback Loop**: Establish a feedback loop with users to gather ongoing input and make necessary improvements.
- **Compliance and Security**: Ensure the system complies with healthcare regulations and maintains patient data privacy and security.

Launch and Scale-Up

- **Full Launch**: Roll out HeartGuard to all intended users, starting with key hospitals and expanding to wider healthcare networks.
- **Scale-Up**: Plan for scalability to handle increasing data volumes and user bases as the system gains adoption.

By following this implementation plan, HeartGuard can effectively deploy its early detection model for heart disease, improving patient outcomes through timely and accurate diagnosis.

Ethical Assessment:

Privacy and Confidentiality:

- Handling sensitive health data can pose risks to patient privacy. Implement strong data anonymization techniques and ensure compliance with regulations like HIPAA to protect patient identity and data confidentiality.

Informed Consent:

- Ensuring patients are aware of how their data will be used and obtaining their explicit consent. Clearly communicate the purpose of data collection, usage, and sharing policies, and obtain consent from patients before including their data in the study.

Bias and Fairness:

- Potential biases in the data or model can lead to unfair treatment of certain groups (e.g., gender, race, socioeconomic status). Conduct thorough bias assessments, use balanced datasets, and incorporate fairness metrics to ensure the model provides equitable outcomes for all population groups.

Data Security:
The risk of data breaches or unauthorized access to sensitive health information. Implement robust cybersecurity measures, including encryption, access controls, and regular security audits to protect data integrity and security.

Transparency and Explainability:

- Ensuring that the model's decisions can be understood and trusted by healthcare professionals and patients. Use interpretable machine learning models and provide clear explanations for predictions to enhance transparency and build trust.

Impact on Patient-Provider Relationship:

- The introduction of predictive models might affect the dynamics of the patient-provider relationship. Educate healthcare providers on the use of predictive models as supportive tools, rather than replacements for professional judgment and patient interaction.
- By addressing these ethical considerations, the Early Detection of Heart Disease project can be conducted responsibly, with a focus on maximizing benefits while minimizing potential harms.

APA References:

- Darshanshelar, D. (2024). HeartGuard: Predictive healthcare for heart disease. GitHub repository. https://github.com/Darshanshelar96k/HeartGuard-Predictive-Healthcare-for-Heart-Disease
- Deepa, R., Sadu, V. B., & C, P. G. (2024). Early prediction of cardiovascular disease using machine learning: Unveiling risk factors from health records. AIP Advances, 14(3), 035049. https://doi.org/10.1063/5.0191990
- Atqarana. (2024). Heart Disease Prediction and Monitoring System. GitHub repository. https://github.com/Atqarana/Heart-Disease-Prediction-Monitoring