TOPIC : HEART DISEASE PREDICTION USING LOGISTIC REGRESSION

Name: Tanwistha Guha

ID: 2310010001437

Team 1– PBEL-AI & ML (IBM)

Problem Statement:

I. Heart disease remains one of the leading causes of death worldwide, and early detection is difficult without proper analysis of patient health data.
II. Raw healthcare datasets often contain missing, inconsistent, or noisy data, which reduces the accuracy of predictive models.
III. Understanding the relationship between multiple health factors such as age, cholesterol, and blood pressure is complex and difficult to interpret manually.

Solution Proposed:

I. Develop a backend system that preprocesses and analyzes patient medical data to identify early risk patterns using data analytics techniques.
II. Implement data cleaning, normalization, and exploratory data analysis (EDA) using **NumPy** and **Pandas** to prepare high-quality input data for heart disease prediction.
III. Use **Matplotlib** and **Seaborn** to visualize correlations and trends among key attributes, enabling better insights and feature selection for the predictive model.

WHY LOGISTIC REGRESSION?

- Logistic Regression was chosen for this project because it is one of the most effective and interpretable algorithms for **binary classification problems**, such as predicting whether a patient has heart disease (**Yes/No**). It models the probability of an event occurring based on input variables like age, cholesterol, and blood pressure.

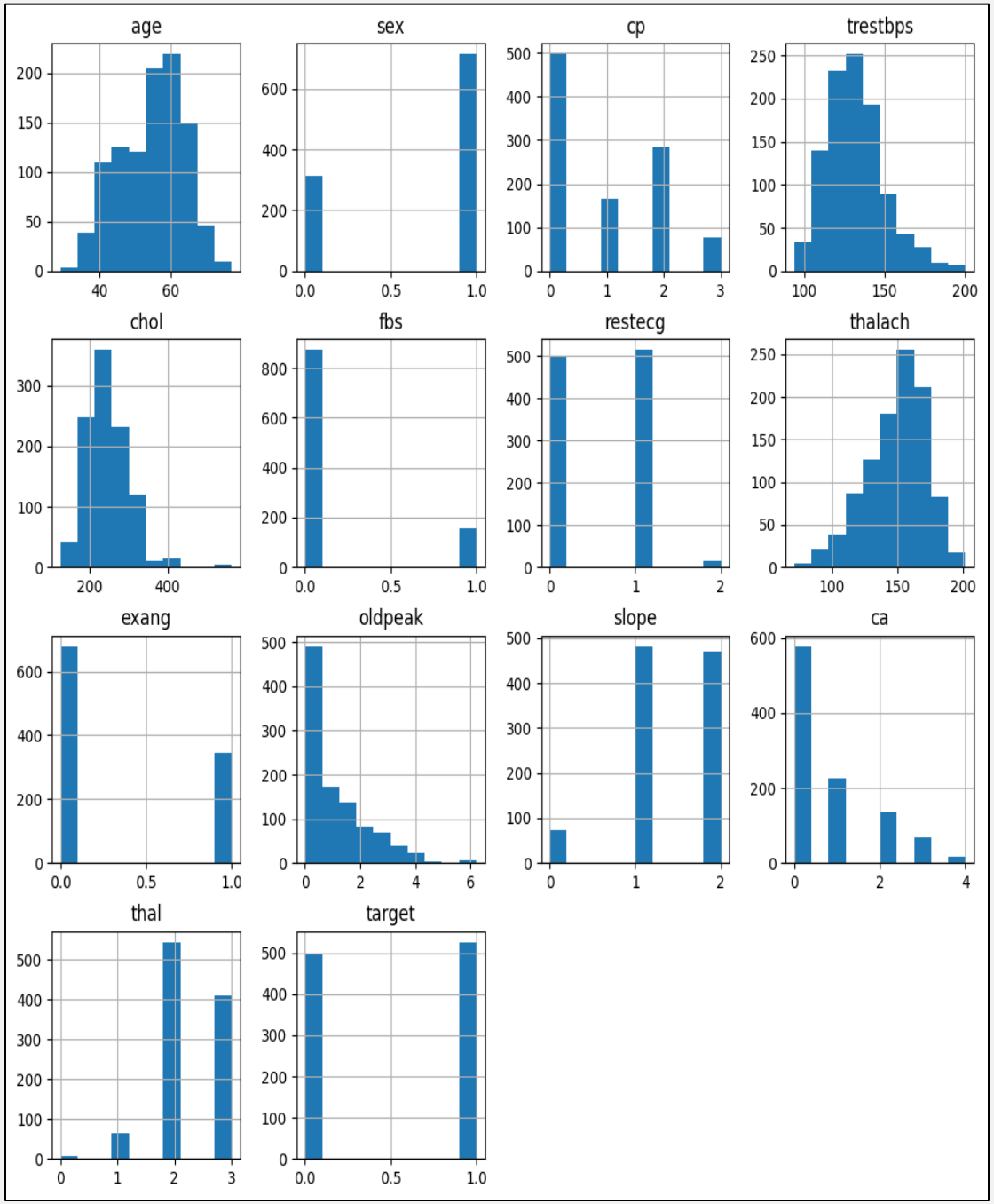
Key Reasons:

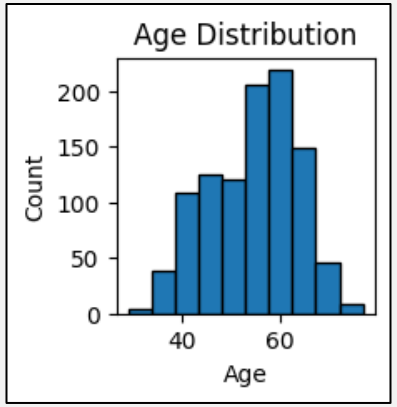
- **Binary Outcome:** Heart disease prediction is a two-class problem (disease present or not), which fits perfectly with Logistic Regression's design.
- **Interpretability:** The model provides clear insights into how each health factor contributes to the risk, making it ideal for healthcare applications where explainability is crucial.
- **Efficiency:** Logistic Regression is computationally lightweight and performs well even with moderate datasets, which is ideal for quick backend implementation.
- **Baseline Model:** It serves as a strong baseline before moving to more complex models like Random Forest or Neural Networks.

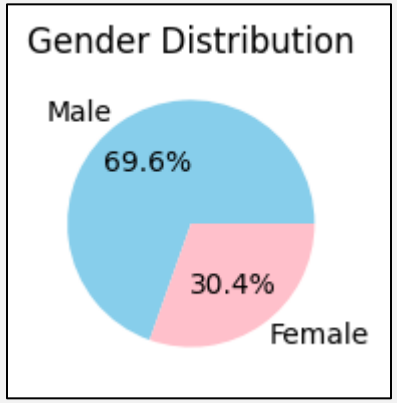
TECH STACK USED

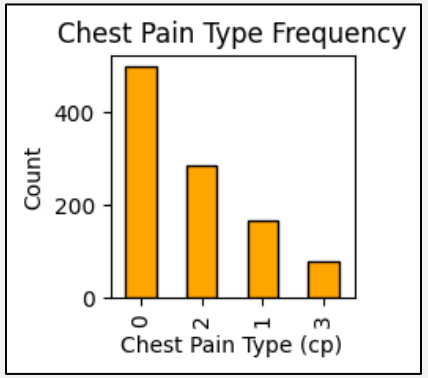
Layer	Technology	Purpose
Programming Language	**Python**	Core language for ML model and backend
Machine Learning Library	**Scikit-learn (sklearn)**	Model training, prediction, scaling, and tuning
Data Handling	**Pandas, NumPy**	Loading and manipulating data (`heart.csv`)
Data Visualization	**Matplotlib, Seaborn**	EDA, correlation plots, and histograms
Version Control (Optional)	**Git / GitHub**	Managing and sharing project code

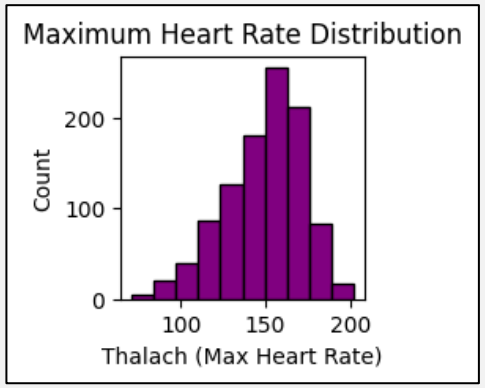
VARIOUS CHARTS

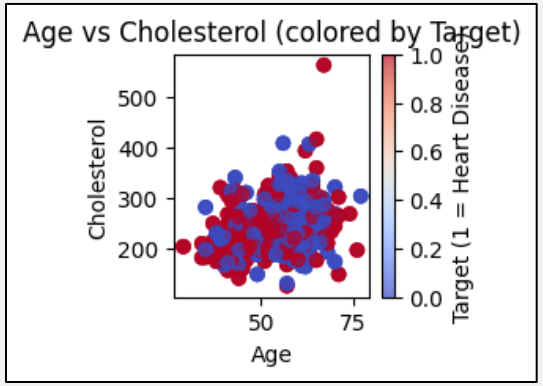


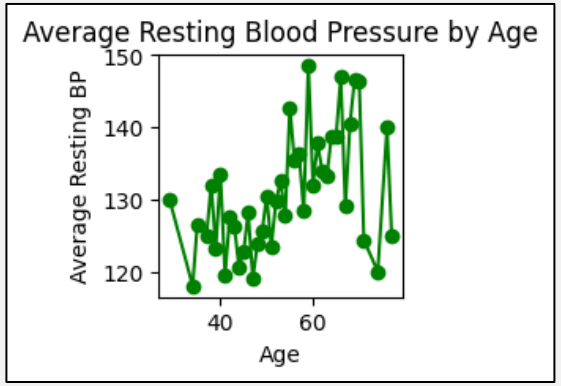












```
[7]: #to get statistical info
df.describe()
```

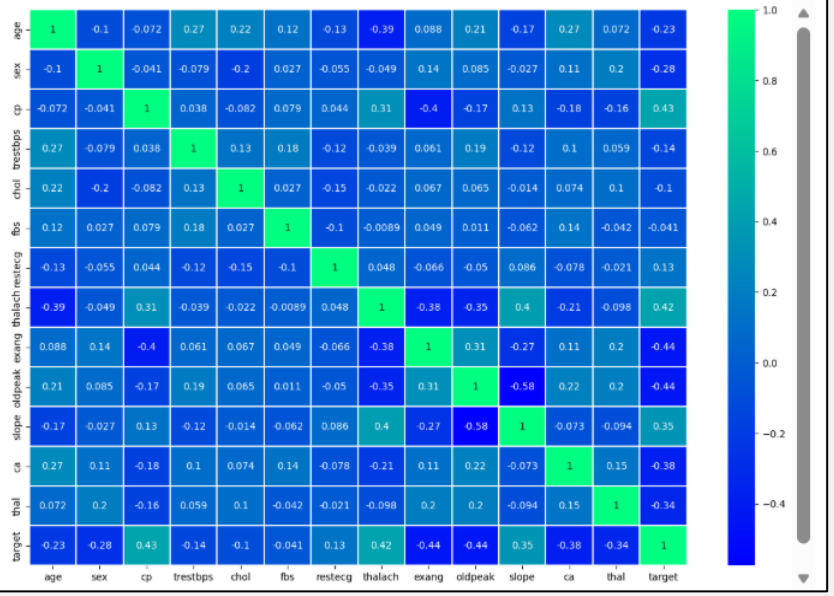
[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.754146
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.030798
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.000000
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.000000
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000	2.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000

```
[32]: #final model implementation
input = (62,0,0,124,209,0,1,163,0,0.0,2,0,2)
arr1 = np.asarray(input)
print(arr1)

arr2 = arr1.reshape(1,-1)

y_predict = model.predict(arr2)

if(y_predict == 1) :
    print ("The patient is having heart disease")
else:
    print("The patient is not having heart disease")
```

```
[ 62.   0.   0. 124. 209.   0.   1. 163.   0.   0.   2.   0.   2.]
The patient is having heart disease
```

```
[11]: #correlation matrix
plt.figure(figsize = (15,10))
sns.heatmap(df.corr() , linewidth = .01, annot=True , cmap="winter")
plt.show()
```




```
[13]: #choosing X (independent , input variables) and y (dependent , output variables)
#in this case my output is the 'target' variable so considering it as y.
#X -> all the rows except target.
X = df.iloc[: , :-1]
y = df.iloc[:,-1]
```

```
[14]: #spliting the data into training and testing data
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split (X,y,test_size=0.25,random_state=42)
```

```
[15]: X_test
```

[15]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
527	62	0	0	124	209	0	1	163	0	0.0	2	0	2
359	53	0	2	128	216	0	0	115	0	0.0	2	0	0
447	55	1	0	160	289	0	0	145	1	0.8	1	1	3
31	50	0	1	120	244	0	1	162	0	1.1	2	0	2
621	48	1	0	130	256	1	0	150	1	0.0	2	2	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
940	57	0	0	140	241	0	1	123	1	0.2	1	0	3
787	51	1	0	140	298	0	1	122	1	4.2	1	3	3
926	54	1	0	110	206	0	0	108	1	0.0	1	1	2
249	42	1	2	130	180	0	1	150	0	0.0	2	0	2
552	43	0	0	132	341	1	0	136	1	3.0	1	0	3

257 rows × 13 columns

```
[16]: y_test
```

```
[16]: 527    1
359    1
447    0
31     1
621    0
      ..
940    0
787    0
926    0
249    1
552    0
Name: target, Length: 257, dtype: int64
```

```
[17]: #creating a logistic regression model
from sklearn.linear_model import LogisticRegression
model = LogisticRegression ()
```

```
[18]: #training the model on training data after creating object of LogisticRegression
model.fit(X_train , y_train)
```

```
[19]: #predicting the model on testing data
y_predict = model.predict(X_test)
print(y_predict)
```

```
[1 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 0 0 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1
 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 1 0 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 0 1 0 1 0 1 1
 1 1 0 1 1 1 0 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 1 0 1 0 0 0 0 0 0 1 1 0 1 0 1 0 1
 1 1 0 1 1 1 1 1 0 0 1 0 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1
 1 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1
 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 1 0 0 0 1 1 0 1 1 0 1 0 0 0 0 1 0 1 0 1 0
 0 1 1 1 1 1 0 0 1 0 1 1 0 0 0 0 1 1 0 1 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0]
```

```
[21]: #to get the accuracy
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test , y_predict))
```

```
0.8054474708171206
```

```
[22]: from sklearn.metrics import confusion_matrix
print (confusion_matrix(y_test,y_predict))
```

```
[[ 99  33]
 [ 17 108]]
```

```
[23]: from sklearn.metrics import classification_report
print (classification_report (y_test , y_predict) )
```

```
              precision    recall  f1-score   support

           0       0.85      0.75      0.80       132
           1       0.77      0.86      0.81       125

    accuracy                           0.81       257
   macro avg       0.81      0.81      0.81       257
weighted avg       0.81      0.81      0.81       257
```

```
: y_test
```

```
: 527    1
359    1
447    0
31     1
621    0
      ..
940    0
787    0
926    0
249    1
552    0
Name: target, Length: 257, dtype: int64
```

LEARNING OUTCOME.

- The Heart Disease Prediction project successfully demonstrates how data analysis and visualization can help in understanding key factors that influence heart health. By using Python libraries such as **NumPy**, **Pandas**, and **Matplotlib**, we were able to clean, analyze, and visualize important health parameters like age, cholesterol level, blood pressure, and heart rate.
- Through various plots—such as histograms, pie charts, bar graphs, and scatter plots—we gained valuable insights into patterns within the dataset. For example, we observed how age and cholesterol correlate with heart disease, and how factors like chest pain type and resting blood pressure vary across individuals.
- This backend project provides a strong foundation for future integration with a predictive model that can help in early detection of heart disease using machine learning techniques.

I would like to express my sincere gratitude to **Himanshu Sir** for his constant guidance, encouragement, and support throughout this project. His insightful feedback and deep understanding of Artificial Intelligence and Machine Learning helped me strengthen my analytical approach and improve the project's quality.
The knowledge and skills I gained under his mentorship during the **IBM AI-ML Workshop** have been invaluable, and I am truly thankful for the opportunity to learn from him.