STEM ROBOTICS Internship

Project Review

Name: Abhinav S

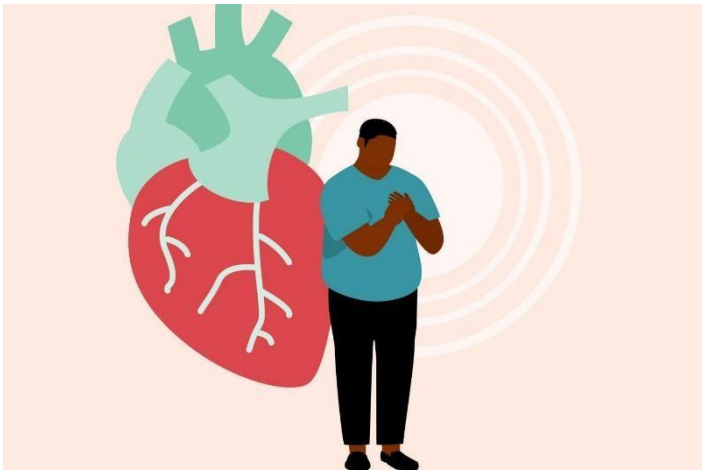
Project: Heart Disease Prediction using Logistic Regression

This project explores the use of machine learning to predict the presence of heart disease based on patient data. Using Logistic Regression, a basic yet powerful classification technique, the model was trained on real-world healthcare data to identify patterns and make accurate predictions.

This work was completed as part of my internship and reflects my learning in data analysis, model training, and evaluation using Python and scikit-learn.

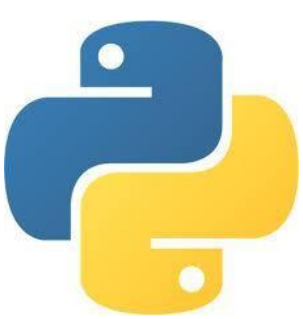
1. Introduction

This project aims to predict the likelihood of heart disease in patients using logistic regression. It was carried out as part of an internship to gain practical exposure to machine learning techniques, particularly in the healthcare domain. The objective was to apply theoretical knowledge to real-world data using Python and popular ML libraries.



2. Tools and Technologies

- Python (programming language)
- Pandas and NumPy (data manipulation)
- Scikit-learn (machine learning framework)
- Google Colab (cloud-based notebook environment)




3.Methodology

The project followed a structured machine learning pipeline:

1. Data Acquisition: The dataset (`heart.csv`) was loaded using pandas.read_csv() in Google Colab.

2. Data Exploration & Cleaning: Basic checks for null values and data types were performed using isnull().sum() and describe().

3. Visualization: No visualizations were implemented; the dataset was explored using summary statistics and column structure.

4. Preprocessing: The dataset was split into training and testing sets using train_test_split, and the features were standardized using StandardScaler.

5. Model Training: A logistic regression model was trained using scikitlearn’s LogisticRegression() on the processed dataset.

6. Evaluation: The model's performance was evaluated using the accuracy_score metric.

We upload the `heart.csv` file using Colab's file upload utility.

```
from google.colab import files
files.upload()
```

```
Choose Files heart.csv
• heart.csv(text/csv) - 11328 bytes, last modified: 6/28/2025 - 100% done
Saving heart.csv to heart (1).csv
{'heart (1).csv':
b'\xef\xbb\xbfage,sex,cp,trestbps,chol,fbs,restecg,thalach,exar
```

```
[40] heart_data.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

```
heart_data.isnull().sum()
```

	0
age	0
sex	0
cp	0
trestbps	0
chol	0
fbs	0
restecg	0
thalach	0

Train-Test Split

bold text We divide the dataset into training and test sets.

```
[46] X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.1,stratify=Y,random_state=1)
```

```
[47] print(X.shape,X_train.shape,X_test.shape)
(303, 13) (272, 13) (31, 13)
```

```
[48] print(X_train)
```

```
[50] model=LogisticRegression()

#training the logisticregression model by using the training data
model.fit(X_train,Y_train)
LogisticRegression()

/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: Conv
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
 LogisticRegression
LogisticRegression()
```

```
accuracy on training data: 0.8455882352941176

[55] #accuracy on test data
     X_test_prediction=model.predict(X_test)
     test_data_accuracy=accuracy_score(X_test_prediction,Y_test)

[56] print('accuracy on the test data:',test_data_accuracy)

     accuracy on the test data: 0.9354838709677419
```

4.Results & Insights

The logistic regression model achieved satisfactory classification results on the test data. Metrics such as accuracy and F1-score demonstrated that the model can serve as a baseline approach for early-stage heart disease detection. The use of visual evaluation also improved interpretability.

5.Conclusion

Through this internship project, valuable experience was gained in practical ML development and evaluation. From data preparation to model interpretation, the workflow adhered to real-world expectations. This project reinforces the importance of data understanding and the potential of logistic regression in medical predictive modeling.