## **Artificial Intelligence and Machine Learning**

Project Report

Semester-IV (Batch-2022)

**Case Study**: - **Heart Disease Prediction Using Logistic Regression\*\***

[Url:-](about:blank) <https://drive.google.com/file/d/1Zua9qIorK6iikpPSGtbbXjseqx5li_wo/view?usp=sharing>

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Rajeev Thakur Yashika Taneja

Roll Number: -2210990979

Group - 14

**Department of Computer Science and Engineering**

**Chitkara University Institute of Engineering & Technology,**

## **Chitkara University, Punjab**

**\*\*Case Study: Heart Disease Prediction Using Logistic Regression\*\***

* **\*\*Objective:\*\***

The objective of this analysis is to predict the likelihood of heart disease occurrence using logistic regression based on various risk factors.

* **\*\*Methods Used:\*\***

- Pandas library

- NumPy library

- Matplotlib library

- Seaborn library

- Scikit-learn library

* **Methods:**

- `read\_csv()`: Reads a CSV file and converts it into a data frame.

- `info()`: Provides basic information about the data frame, such as column types and missing values.

- `describe()`: Generates descriptive statistics of the data.

- `dropna()`: Removes rows with missing values from the dataset.

- `corr()`: Calculates the correlation matrix between numerical features.

- `heatmap()`: Plots a heatmap to visualize the correlation matrix.

- `pairplot()`: Generates a pairplot to visualize pairwise relationships between features.

- `catplot()`: Plots categorical plots such as countplots.

- `train\_test\_split()`: Splits the dataset into training and testing sets.

- `LogisticRegression()`: Initializes the logistic regression model.

- `fit()`: Trains the logistic regression model.

- `predict()`: Predicts the target variable using the trained model.

- `score()`: Calculates the accuracy score of the model.

- `confusion\_matrix()`: Computes the confusion matrix to evaluate the model's performance.

- `classification\_report()`: Generates a classification report including precision, recall, and F1-score.

* **\*\*Analysis Steps:\*\***

1. \*\*Data Loading and Exploration:\*\*

- Read the data from the CSV file using `read\_csv()` method.

- Analyze the structure and summary statistics of the dataset using `info()` and `describe()` methods.

2. \*\*Data Cleaning:\*\*

- Remove rows with missing values using `dropna()` method.

3. \*\*Data Visualization:\*\*

- Visualize the correlation between features using a heatmap with `corr()` and `heatmap()` methods.

- Generate a pairplot to visualize pairwise relationships between features using `pairplot()` method.

- Plot countplots to analyze the distribution of categorical variables.

4. \*\*Machine Learning Part:\*\*

- Separate the data into features (independent variables) and target (dependent variable).

- Split the dataset into training and testing sets using `train\_test\_split()` method.

- Initialize the logistic regression model using `LogisticRegression()` method.

- Train the model using the training data with `fit()` method.

- Test the model's performance using the testing data and calculate the prediction score with `score()` method.

- Evaluate the model's performance using confusion matrix and classification report with `confusion\_matrix()` and `classification\_report()` methods.

5. \*\*Visualization of Model Evaluation:\*\*

- Plot the confusion matrix to visualize the model's performance using `heatmap()` method.

* \*\*Conclusion:\*\*

The logistic regression model trained on the Framingham Heart Study dataset achieved a prediction score of [mention score here]. The model demonstrated [mention evaluation metrics here] performance in predicting the likelihood of heart disease occurrence. Further optimization and evaluation may be required to improve the model's accuracy and generalizability.

This structured approach provides a comprehensive understanding of the steps involved in predicting heart disease occurrence using logistic regression and evaluating the model's performance.