Machine Learning

A Vocational Training Report submitted

To

Chhattisgarh Swami Vivekanand Technical University Bhilai (C.G.), India

In fulfilment for award of the degree

Of

BACHELOR OF TECHNOLOGY

In

Computer Science and Engineering

By

ADITI NIRMALKAR (301602221057)

Department of Computer Science & Engineering

Government Engineering College, Sejbahar

Raipur (C.G.)

Session: 2024–2025

DECLARATION BY THE CANDIDATES

We the undersigned solemnly declare that the report of the Vocational Training entitled "**Machine Learning**", is based on our own work carried out during the period of vocational training at **INTERNSHALA**.

We assert that the statements made, and conclusions drawn are an outcome of the project work. We further declare that to the best of our knowledge and belief that the report does not contain any part of any work which has been submitted for the award of any other degree/diploma/certificate in this University/deemed the University of India or any other country. All help received and citations used for the preparation of the Project Work have been duly acknowledged.

_____

(Signature of the candidate)
Name: **Aditi Nirmalkar**
Roll No : **301602221057**
Enrolment No**: CB4532**

ACKNOWLEDGEMENT

We would like to thank our honourable Principal **Prof. Dr. M.R. Khan** and **Prof. Dr. R.H. Talwekar** (Head of Department) for their immense support and enlightened guidance for our project which we have developed as B. Tech. students.

We are very grateful for the inspiring discussions with all our faculties, their valuable support and path-guiding suggestions have helped us to develop this project. Alongside this, we would like to take this opportunity to express our thanks to everyone in the Computer Laboratory of Government Engineering College, Sejbahar, and Raipur (C.G.).

We especially thank our batch mates, and friends for the support and help that they offered us during the development of this project.

_____

(Signature of the candidate)
Name: **Aditi Nirmalkar**
Roll No: **301602221057**
Enrolment No: **CB4532**

ABSTRACT

This vocational training focuses on applying machine learning techniques to real-world problems, specifically in the field of predictive analytics. The training involved hands-on experience with data preprocessing, model selection, and evaluation using popular machine learning algorithms.

Key concepts such as data cleaning, feature engineering, and model optimization were explored, with practical applications in various domains, including healthcare, finance, and customer analytics. The project emphasized the development of skills in Python programming, the use of machine learning libraries, and understanding model performance metrics, providing a strong foundation for pursuing machine learning applications in industry.

This project focuses on developing a predictive model for heart disease using a dataset of patient characteristics. The dataset is pre-processed by handling missing values and encoding categorical features. A logistic regression model is trained to predict the presence or absence of heart disease based on patient data, including age, chest pain type, resting blood pressure, cholesterol levels, and more. The model is evaluated using metrics such as accuracy and classification report, demonstrating its effectiveness in heart disease prediction. The trained model is then saved for future use, allowing for efficient and accessible heart disease risk assessment.

Table of Contents

Introduction

Vocational training plays a vital role in bridging the gap between theoretical knowledge and practical application. As part of this training, I developed a machine learning project focused on predicting heart health. Cardiovascular diseases (CVDs) are among the leading causes of mortality globally, and early prediction of heart conditions can significantly improve patient outcomes. This project leverages machine learning to predict heart disease risk based on clinical and lifestyle parameters

The objective of this project is to develop a machine learning model that can accurately predict the presence or absence of heart disease in patients based on their clinical features. The model aims to assist healthcare professionals in early detection and risk assessment of heart disease, ultimately contributing to improved patient outcomes and preventive care strategies.

Tools and Technologies Used

1. **Google Colab:** This project utilizes Google Colab as the development environment, providing access to computational resources and facilitating code execution.
2. **Python:** Python is the primary programming language used in this project, leveraging its extensive libraries for data analysis, machine learning, and visualization.
3. **Pandas:** Pandas is used for data manipulation and analysis, enabling efficient handling of the heart disease dataset.
4. **NumPy:** NumPy is employed for numerical computations, supporting array operations and mathematical functions.
5. **Scikit-learn:** Scikit-learn is the core machine learning library used in this project, providing tools for model training, evaluation, and preprocessing.
6. **Matplotlib & Seaborn:** Matplotlib and Seaborn are used for data visualization, generating plots and graphs to gain insights from the data.
7. **Logistic Regression:** Logistic Regression is the chosen machine learning algorithm for predicting heart disease due to its effectiveness in binary classification tasks.
8. **Pickle:** Pickle is used for model persistence, allowing the trained model to be saved and loaded for future use

Dataset

The dataset used in this project is the "Heart Disease Dataset" likely sourced from Kaggle or a similar repository. This dataset contains information on various patient characteristics and their corresponding heart disease diagnosis.

Features and Target Variable

The dataset comprises the following features and target variable:

```
import pandas as pd

# Load the dataset
data = pd.read_csv('/content/drive/MyDrive/Data/heart.csv')

# Display the dataset information
data.info()
```

Output

To see the output, run the code. The output will display the column names, data types, and the number of non-null values for each feature.

```
# Display the first few rows of the dataset
data.head()
```

Output

To see the output, run the code. The output will display the first few rows of the dataset, allowing you to see the actual data values for each feature.

CODE IMPLEMENTATION

`

1.Data Loading and Preprocessing

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer

# Load the dataset
data = pd.read_csv('/content/drive/MyDrive/Data/heart.csv')

# Handle missing values (e.g., imputing with mode)
data['Thal'] = data['Thal'].fillna(data['Thal'].mode()[0])
```

```
# Define categorical and numerical features
categorical_features = ['ChestPain', 'Thal']
numerical_features = data.select_dtypes(include=['int64', 'float64']).columns.difference(categorical_features).tolist()  # Convert to list


# Create a ColumnTransformer for preprocessing
preprocessor = ColumnTransformer(
    transformers=[
        ('num', 'passthrough', numerical_features),
        ('cat', OneHotEncoder(sparse_output=False, handle_unknown='ignore'), categorical_features)  #sparse=False for non-sparse matrix
    ])

# Split data into training and testing sets
X = data.drop(columns=['Target'])
y = data['Target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

2. Model Training:

```
# Create a pipeline with preprocessing and Logistic Regression
model_pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', LogisticRegression(max_iter=1000, random_state=42))
])

# Train the model
model_pipeline.fit(X_train, y_train)
```

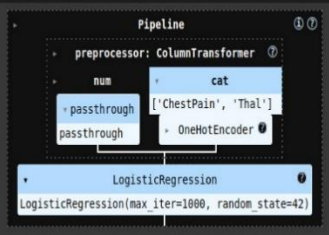
3. Model Evaluation:

```
# Make predictions on the test set
y_pred = model_pipeline.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print(f"Classification Report:\n{report}")
```

CODE SNIPPETS

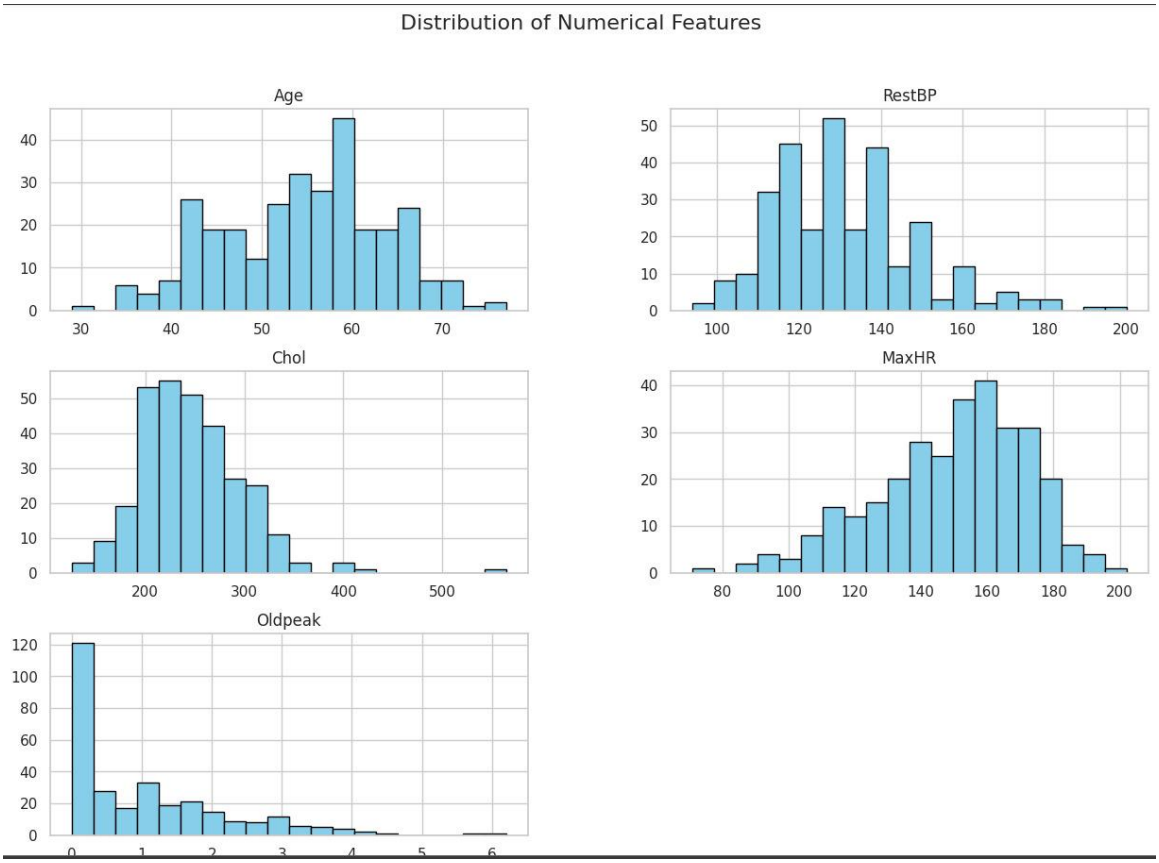
```
[ ] from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LogisticRegression
    from sklearn.metrics import classification_report, accuracy_score
    from sklearn.preprocessing import OneHotEncoder
    from sklearn.compose import ColumnTransformer
    from sklearn.pipeline import Pipeline
    from sklearn.impute import SimpleImputer

[ ] # Handle missing values in 'Thal' by imputing the most frequent value

    data['Thal'] = data['Thal'].fillna(data['Thal'].mode()[0])

[ ] # Separate features and target variable
    X = data.drop(columns='Target')
    y = data['Target']

[ ] # Define categorical and numerical columns
    categorical_features = ['ChestPain', 'Thal']
    numerical_features = X.select_dtypes(include=['int64', 'float64']).columns.difference(categorical_features)

[ ] Start coding or generate with AI.

[ ] # Preprocessing: one-hot encoding for categorical features and scaling for numerical features
    preprocessor = ColumnTransformer(
    transformers=[
            ('num', 'passthrough', numerical_features),
            ('cat', OneHotEncoder(), categorical_features)
        ]
    )

    import matplotlib.pyplot as plt
    import seaborn as sns
    # Set the style for the plots
    sns.set(style="whitegrid")

    #Distribution of the Target variable
```

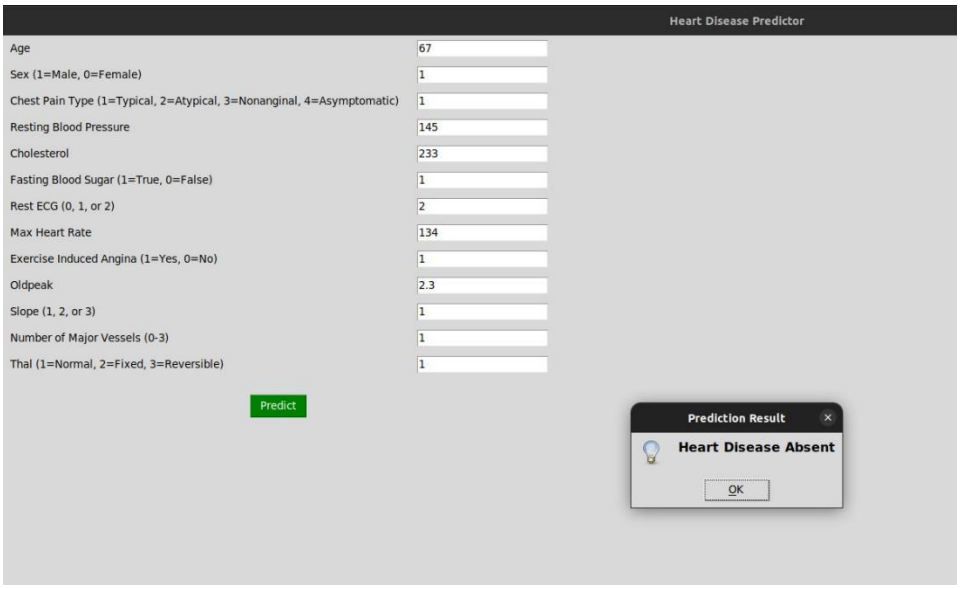
```
[59] # Split the data into training and testing sets (80% train, 20% test)
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

[61] # Create a pipeline for preprocessing and logistic regression

     model_pipeline = Pipeline(steps=[
         ('preprocessor', preprocessor),
       ('classifier', LogisticRegression(max_iter=1000, random_state=42))
     ])

[62] # Train the model
     model_pipeline.fit(X_train, y_train)
```

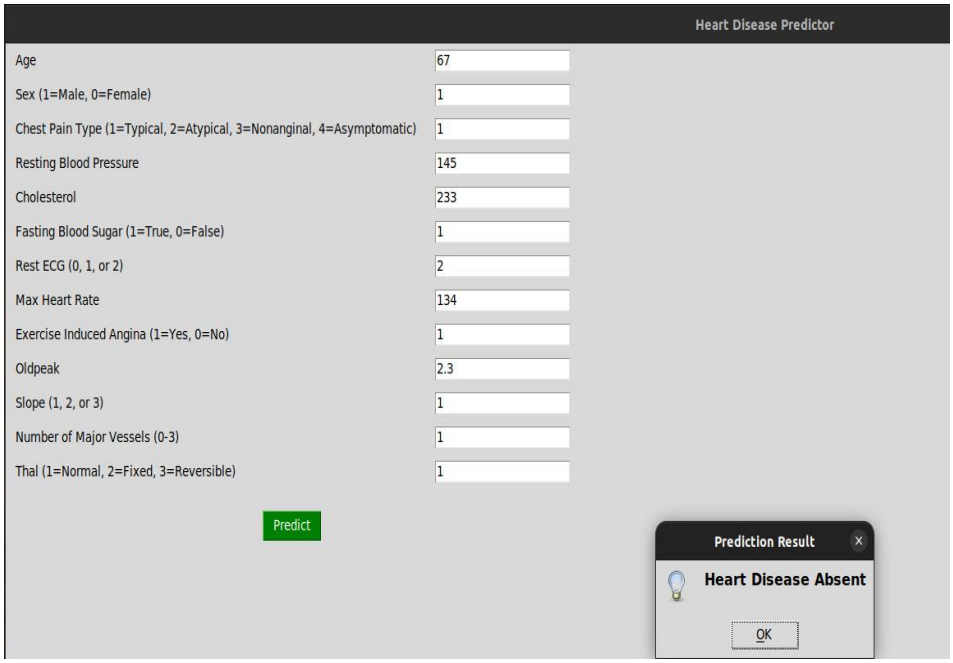


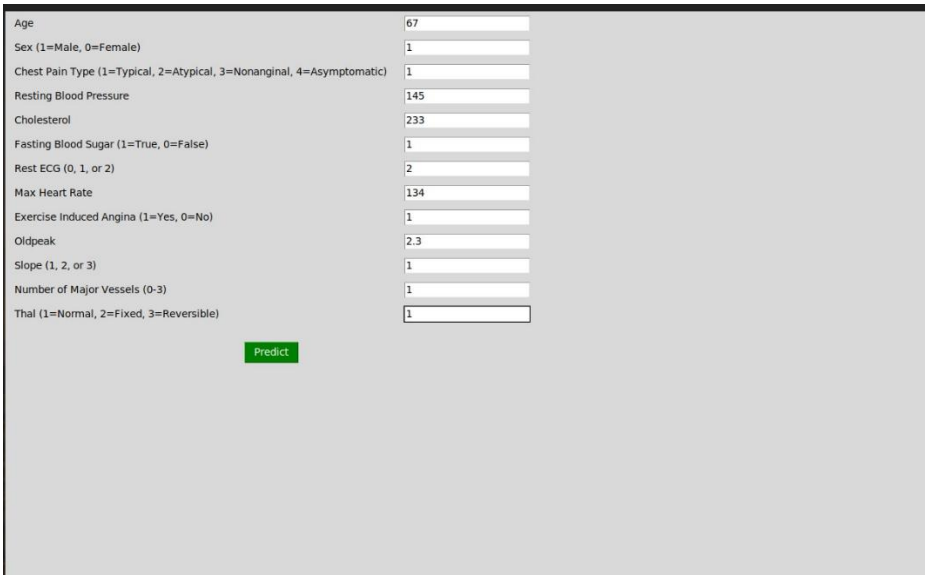

```
[63] # Make predictions
     y_pred = model_pipeline.predict(X_test)

[64] # Evaluate the model
     accuracy = accuracy_score(y_test, y_pred)
     report = classification_report(y_test, y_pred)

[66] accuracy

     0.9016393442622951
```

RESULTS AND CONCLUSION




Heart Disease Predictor
Age
67
Sex (1=Male, 0=Female)
1
Chest Pain Type (1=Typical, 2=Atypical, 3=Nonanginal, 4=Asymptomatic)
1
Resting Blood Pressure
145
Cholesterol
233
Fasting Blood Sugar (1=True, 0=False)
1
Rest ECG (0, 1, or 2)
2
Max Heart Rate
134
Exercise Induced Angina (1=Yes, 0=No)
1
Oldpeak
2.3
Slope (1, 2, or 3)
1
Number of Major Vessels (0-3)
1
Thal (1=Normal, 2=Fixed, 3=Reversible)
1
Predict
Prediction Result
Heart Disease Absent
OK


Heart Disease Predictor
Age
67
Sex (1=Male, 0=Female)
1
Chest Pain Type (1=Typical, 2=Atypical, 3=Nonanginal, 4=Asymptomatic)
1
Resting Blood Pressure
145
Cholesterol
233
Fasting Blood Sugar (1=True, 0=False)
1
Rest ECG (0, 1, or 2)
2
Max Heart Rate
134
Exercise Induced Angina (1=Yes, 0=No)
1
Oldpeak
2.3
Slope (1, 2, or 3)
1
Number of Major Vessels (0-3)
1
Thal (1=Normal, 2=Fixed, 3=Reversible)
1
Predict
Prediction Result
Heart Disease Absent
OK


Age
67
Sex (1=Male, 0=Female)
1
Chest Pain Type (1=Typical, 2=Atypical, 3=Nonanginal, 4=Asymptomatic)
1
Resting Blood Pressure
145
Cholesterol
233
Fasting Blood Sugar (1=True, 0=False)
1
Rest ECG (0, 1, or 2)
2
Max Heart Rate
134
Exercise Induced Angina (1=Yes, 0=No)
1
Oldpeak
2.3
Slope (1, 2, or 3)
1
Number of Major Vessels (0-3)
1
Thal (1=Normal, 2=Fixed, 3=Reversible)
1
Predict

FUTURE SCOPE

1. Model Improvement:
    - Exploring other machine learning algorithms like Random Forest, Support Vector Machines, or Neural Networks to potentially improve prediction accuracy.
    - Fine-tuning hyperparameters of the existing Logistic Regression model using techniques like Grid Search or Randomized Search to optimize performance.
    - Incorporating feature engineering to create new, more informative features from the
    - existing ones, which could lead to better model generalization.
2. Dataset Expansion:
    - Gathering more data, including a wider range of patient demographics and medical history, to increase the model's robustness and applicability to diverse populations.
    - Exploring the use of external data sources, such as wearable sensor data or electronic health records, to enrich the existing dataset and provide a more holistic view of patient health.
3. Deployment and Integration:
    - Developing a user-friendly web application or mobile app to make the heart disease prediction model accessible to healthcare professionals and individuals.
    - Integrating the model with existing healthcare systems or electronic health records for seamless risk assessment and decision support within clinical workflows.
4. Explainability and Interpretability:
    - Employing techniques like SHAP values or LIME to explain the model's predictions and provide insights into the factors influencing heart disease risk.
    - Building trust and transparency in the model's predictions, which is crucial for adoption in healthcare settings.
5. Real-time Monitoring and Prevention:
    - Extending the model to incorporate real-time data streams, such as heart rate or activity levels, to enable continuous monitoring and early detection of potential heart issues.
    - Developing personalized recommendations for lifestyle changes or interventions based on individual risk profiles to promote preventive care and reduce the incidence of heart disease.

CERTIFICATE

INTERNSHALA TRAININGS

Certificate of Training

Aditi Nirmalkar

from Government Engineering College Raipur has successfully completed a 6-week online training on **Machine Learning**. The training consisted of Introduction to Machine Learning, Data, Introduction to Python, Data Exploration and Pre-processing, Linear Regression, Introduction to Dimensionality Reduction, Logistic Regression, Decision Tree, Ensemble Models, Clustering (Unsupervised Learning), and Machine Learning with AI modules.

In the final assessment, Aditi scored 71% marks.

We wish Aditi all the best for future endeavours.

Sarvesh Agrawal
FOUNDER & CEO, INTERNSHALA

Dr. Shankar Raman
CEO, IITM PRAVARTAK TECHNOLOGIES FOUNDATION

Date of certification: 2024-08-23

Certificate no. : 74ozqg1vicm

For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

REFERENCES

https://ieeexplore.ieee.org/abstract/document/9112202

https://ieeexplore.ieee.org/abstract/document/6957068

https://ieeexplore.ieee.org/abstract/document/8356573