|  |  |  |
| --- | --- | --- |
| **S.NO** | **PATICULARS** | **PAGE.NO.** |
|  | **ABSTRACT** | 2-4 |
| 1 | **BASIC UNDERSTANDING OF CONCEPT** | 5-8 |
| 2 | **OVERVIEW OF PROJECT** | 9-10 |
| 3 | **INSTALLATION AND SETUP** | 11-12 |
| 4 | **DATA EXPLORATION** | 13-14 |
| 5 | **DATA PREPROCESSING** | 15-16 |
| 6 | **MODEL TRAINING** | 17-18 |
| 7 | **MODEL EVALUATION** | 19-20 |
| 8 | **VISUALIZATION** | 21-22 |
| 9 | **FINAL FULL CODE** | 23-26 |
| 10 | **OUTPUT** | 27-29 |
| 11 | **CONCLUSION AND FUTURE WORK** | 30-31 |

**ABSTRACT**

In this project, we will be creating a model which will help us to predict the HEART DISEASE PAITENTS in future. We will be training the model with 6 different algorithms to see which algorithm will give us the best performance, based on that we will be using the same model for predicting the HEART DISEASE PAITENTS. This will help us to predecting in the right person. For this we would be using several classification algorithms like KNN, SVM, LDA, Logistic Regression, Decision Tree, Naive Bayes etc. model. But the best algorithm is logistic regression. The dataset will be divided into training set and test set. Using the training set the model will be built and using the test set we will determine the accuracy of the algorithm by applying the test set to the training set. We will be generating a confusion matrix which will help to determine our results accurately. Using this project several reports will be generated to determine the accuracy of the model.

**Existing System:**

The existing system focusses on the algorithms such as Decision tree, LDA, Random Forest and XGBoost to determine the accuracy. These algorithms especially Random Forest works well when we are working with small samples of data. Random Forests tend to train each tree independently, using a random sample of the data. This would help to make the model more robust than a single decision tree, and less likely to overfit on the training data. XGBoost build trees one at a time, where each new tree helps to correct errors made by previously trained tree. With each tree that is being added, the model becomes even more expressive.

**Disadvantages:**

1.No interpretability

2.Overfitting can easily occur

3.Need to choose the number of trees

**Proposed System:**

In the proposed system we will be using the same old LDA and Decision tree in addition to the we would using other algorithms like Support Vector Machine,K nearest Neighbours and Naïve Bayes to analyse the sales of an organisation.It helps us to work on the data which contains large samples,thereby helping us to determine the result accurately.These algorithms tend to be more reliable.

**Advantages:**

1.Easy to interpret

2.No overfitting

3.High Accuracy

**SYSTEM SPECIFICATION:**

Hardware Requirements:

* System: RYZEN 9 5900HS
* Hard disk: 1TB SSD
* Monitor: XRGB
* Mouse: Optical Mouse
* RAM: 16GB

**SOFTWARE REQUIREMENTS**

* Operating System: Windows 11
* Coding Language: Python
* Dataset : MS Excel
* Software IDE : visual studio code
* Documentation : Microsoft Office

\

**CHAPTER-1:**

**BASIC UNDERSTANDING OF CONCEPT**

**Machine Learning**

The term ‘machine learning’ is often, incorrectly, interchanged with Artificial Intelligence, but machine learning is actually a subfield/type of AI. Machine learning is also often referred to as predictive analytics, or predictive modelling.

Coined by American computer scientist Arthur Samuel in 1959, the term ‘machine learning’ is defined as a “computer’s ability to learn without being explicitly programmed”.

At its most basic, machine learning uses programmed algorithms that receive and analyse input data to predict output values within an acceptable range. As new data is fed to these algorithms, they learn and optimise their operations to improve performance, developing ‘intelligence’ over time.

There are four types of machine learning algorithms: supervised, semi-supervised, unsupervised and reinforcement.

**Supervised learning**

In supervised learning, the machine is taught by example. The operator provides the machine learning algorithm with a known dataset that includes desired inputs and outputs, and the algorithm must find a method to determine how to arrive at those inputs and outputs. While the operator knows the correct answers to the problem, the algorithm identifies patterns in data, learns from observations and makes predictions. The algorithm makes predictions and is corrected by the operator – and this process continues until the algorithm achieves a high level of accuracy/performance.

Under the umbrella of supervised learning fall: Classification, Regression and Forecasting.

1. Classification: In classification tasks, the machine learning program must draw a conclusion from observed values and determine to  
   what category new observations belong. For example, when filtering emails as ‘spam’ or ‘not spam’, the program must look at existing observational data and filter the emails accordingly.
2. Regression: In regression tasks, the machine learning program must estimate – and understand – the relationships among variables. Regression analysis focuses on one dependent variable and a series of other changing variables – making it particularly useful for prediction and forecasting.
3. Forecasting: Forecasting is the process of making predictions about the future based on the past and present data, and is commonly used to analyse trends.

**Semi-supervised learning**

Semi-supervised learning is similar to supervised learning, but instead uses both labelled and unlabelled data. Labelled data is essentially information that has meaningful tags so that the algorithm can understand the data, whilst unlabelled data lacks that information. By using this combination, machine learning algorithms can learn to label unlabelled data.

**Unsupervised learning**

Here, the machine learning algorithm studies data to identify patterns. There is no answer key or human operator to provide instruction. Instead, the machine determines the correlations and relationships by analysing available data. In an unsupervised learning process, the machine learning algorithm is left to interpret large data sets and address that data accordingly. The algorithm tries to organise that data in some way to describe its structure. This might mean grouping the data into clusters or arranging it in a way that looks more organised.

As it assesses more data, its ability to make decisions on that data gradually improves and becomes more refined.

Under the umbrella of unsupervised learning, fall:

1. Clustering: Clustering involves grouping sets of similar data (based on defined criteria). It’s useful for segmenting data into several groups and performing analysis on each data set to find patterns.
2. Dimension reduction: Dimension reduction reduces the number of variables being considered to find the exact information required.

**Reinforcement learning**

Reinforcement learning focuses on regimented learning processes, where a machine learning algorithm is provided with a set of actions, parameters and end values. By defining the rules, the machine learning algorithm then tries to explore different options and possibilities, monitoring and evaluating each result to determine which one is optimal. Reinforcement learning teaches the machine trial and error. It learns from past experiences and begins to adapt its approach in response to the situation to achieve the best possible result.

**CHAPTER 2:**

**OVERVIEW OF PROJECT**

**Overview of the Project:**

This project aims to predict heart disease using a logistic regression model. The dataset used is heart.csv, which contains various features related to heart health.

**Objectives**:

* To build a predictive model for heart disease.
* To evaluate the model’s performance using accuracy metrics.

**Importance of Heart Disease Prediction**:

Heart disease is a leading cause of death worldwide. Early prediction can help in timely intervention and treatment, potentially saving lives.

**Background**:

Heart disease encompasses a range of conditions that affect the heart, including coronary artery disease, arrhythmias, and congenital heart defects. Predictive modeling in healthcare can significantly improve patient outcomes by enabling early diagnosis and personalized treatment plans.

**CHAPTER 3:**

**INSTALLATION AND SETUP**

**Install Visual Studio Code:**

1. **Download Visual Studio Code:**
   * Go to the official Visual Studio Code website.
   * Click on the “Download” button for your operating system.
2. **Run the Installer:**
   * Open the downloaded installer.
   * Follow the installation instructions.
   * Optionally, check the box to create a desktop icon.

**Required Software and Libraries**:

* Python
* numpy
* pandas
* matplotlib
* seaborn
* scikit-learn

**Installation Instructions**:

To install the required libraries, use the following pip commands:

“pip install numpy pandas matplotlib seaborn scikit-learn”

**Setting Up the Environment**:

Ensure that you have a suitable Python environment set up. You can use virtual environments to manage dependencies:

“python -m venv myenv

source myenv/bin/activate # On Windows use `myenv\Scripts\activate`”

**CHAPTER 4:**

**DATA EXPLORATION**

**Data set used:**

[**heart disease dataset**](https://www.kaggle.com/datasets/satish1477/heart-disease-dataset)

**Loading the Dataset**:

import pandas as pd

dataset = pd.read\_csv("heart.csv")

**Initial Data Exploration**:

print(dataset.shape)

print(dataset.head(5))

**Understanding the Dataset Structure**:

The dataset contains various features such as age, sex, chest pain type, resting blood pressure, cholesterol levels, and more. Each feature provides valuable information that can help in predicting heart disease.

**Descriptive Statistics:**

print(dataset.describe())

**Data Visualization:**

Visualizing the data helps in understanding the distribution and relationships between features.

import matplotlib.pyplot as plt

import seaborn as sns

plt.figure(figsize=(10, 6))

sns.heatmap(dataset.corr(), annot=True, fmt=".2f")

plt.title("Correlation Matrix")

plt.show()

**CHAPTER 5:**

**DATA PREPROCESSING**

**Handling Missing Values:**

dataset.isnull().sum()

If there are any missing values, they need to be handled appropriately, either by filling them with a suitable value or by removing the affected rows.

**Splitting the Data:**

from sklearn.model\_selection import train\_test\_split

predictors = dataset.drop("target", axis=1)

target = dataset["target"]

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(predictors, target, test\_size=0.20, random\_state=0)

**Feature Selection:**

Selecting relevant features is crucial for building an effective model. In this project, all features except the target variable are used as predictors.

**CHAPTER 6:**

**MODEL TRAINING**

**Logistic Regression Model**:

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome.

**Training the Model:**

from sklearn.linear\_model import LogisticRegression

lr = LogisticRegression()

lr.fit(X\_train, Y\_train)

**Hyperparameter Tuning**:

Hyperparameter tuning can improve the model’s performance. Techniques such as Grid Search or Random Search can be used to find the best parameters.

**CHAPTER 7:**

**MODEL EVALUATION**

**Making Predictions**:

Y\_pred\_lr = lr.predict(X\_test)

**Evaluating Model Performance**:

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

score\_lr = round(accuracy\_score(Y\_pred\_lr, Y\_test) \* 100, 2)

print("The accuracy score achieved using Logistic Regression is: " + str(score\_lr) + " %")

**Confusion Matrix**:

cm = confusion\_matrix(Y\_test, Y\_pred\_lr)

sns.heatmap(cm, annot=True, fmt="d")

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()

**Classification Report**:

print(classification\_report(Y\_test, Y\_pred\_lr))

**CHAPTER 8:**

**VISUALIZATION**

**Accuracy Plot**:

plt.figure(figsize=(8,6))

plt.bar(["Logistic Regression"], [score\_lr])

plt.xlabel("Model")

plt.ylabel("Accuracy (%)")

plt.title("Model Accuracy")

plt.show()

**Target Count Plot**:

plt.figure(figsize=(8,6))

sns.countplot(x="target", data=dataset)

plt.xlabel("Target")

plt.ylabel("Count")

plt.title("Target Count")

plt.show()

**Feature Importance**:

Understanding which features are most important can provide insights into the model’s decision-making process.

importance = lr.coef\_[0]

feature\_importance = pd.Series(importance, index=predictors.columns).sort\_values(ascending=False)

feature\_importance.plot(kind='bar')

plt.title("Feature Importance")

plt.show()

**CHAPTER 9:**

**FINAL FULL CODE**

**#importing the requirements**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**import os**

**print(os.listdir())**

**import warnings**

**warnings.filterwarnings('ignore')**

**#opening data set**

**dataset = pd.read\_csv("heart.csv")**

**type(dataset)**

**dataset.shape**

**dataset.head(5)**

**#spliting the test and training data sets**

**from sklearn.model\_selection import train\_test\_split**

**predictors = dataset.drop("target",axis=1)**

**target = dataset["target"]**

**X\_train,X\_test,Y\_train,Y\_test = train\_test\_split(predictors,target,test\_size=0.20,random\_state=0)**

**X\_train.shape**

**X\_test.shape**

**Y\_train.shape**

**Y\_test.shape**

**#importing the model**

**from sklearn.metrics import accuracy\_score**

**from sklearn.linear\_model import LogisticRegression**

**lr = LogisticRegression()**

**lr.fit(X\_train,Y\_train)**

**Y\_pred\_lr = lr.predict(X\_test)**

**Y\_pred\_lr.shape**

**score\_lr = round(accuracy\_score(Y\_pred\_lr,Y\_test)\*100,2)**

**print("The accuracy score achieved using Logistic Regression is: "+str(score\_lr)+" %")**

**# Accuracy Plot**

**plt.figure(figsize=(8,6))**

**plt.bar(["Logistic Regression"], [score\_lr])**

**plt.xlabel("Model")**

**plt.ylabel("Accuracy (%)")**

**plt.title("Model Accuracy")**

**plt.show()**

**# Count Plot**

**plt.figure(figsize=(8,6))**

**sns.countplot(x="target", data=dataset)**

**plt.xlabel("Target")**

**plt.ylabel("Count")**

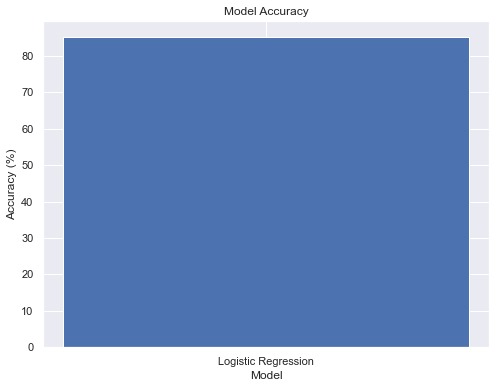
**plt.title("Target Count")**

**plt.show()**

**CHAPTER 10:**

**OUTPUT**

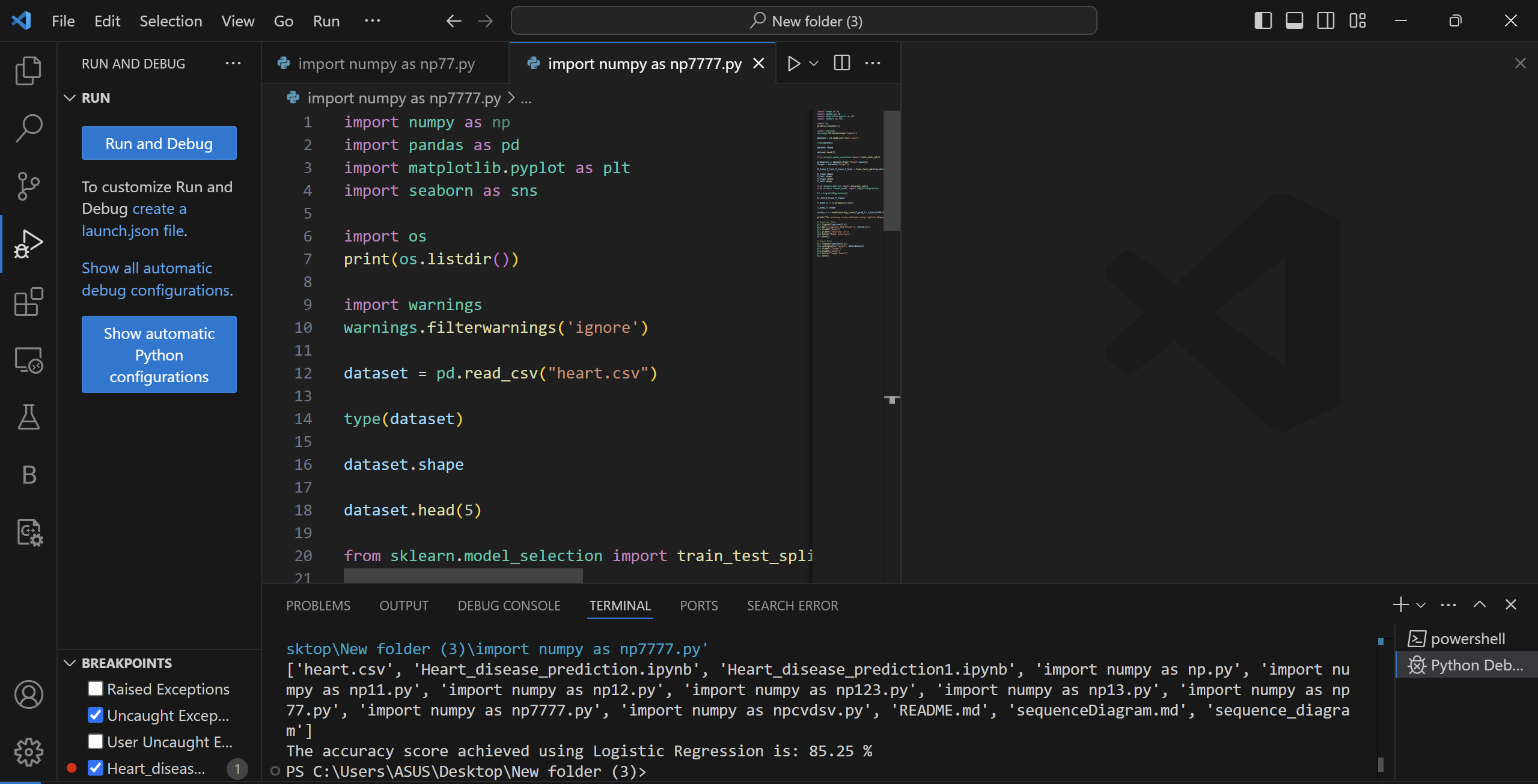
**MODEL ACCURACY:**



**TARGET COUNT:**



**ACCURACY SCREEN SHOT:85.25%**

****

**CHAPTER 11:**

**CONCLUSION AND FUTURE WORK**

**Summary of Findings:**

The logistic regression model achieved an accuracy of 85.25**%** in predicting heart disease.

**Potential Improvements**:

* Exploring different machine learning models.
* Tuning hyperparameters for better performance.
* Using more advanced techniques like cross-validation.

**Future Work**:

* Implementing additional models and comparing their performance.
* Collecting more data to improve model accuracy.
* Integrating the model into a web application for real-time predictions.