Final Project Report

1. Introduction
    1. Project overviews
    2. Objectives
2. Project Initialization and Planning Phase
    1. Define Problem Statement
    2. Project Proposal (Proposed Solution)
    3. Initial Project Planning
3. Data Collection and Preprocessing Phase
    1. Data Collection Plan and Raw Data Sources Identified
    2. Data Quality Report
    3. Data Exploration and Preprocessing
4. Model Development Phase
    1. Feature Selection Report
    2. Model Selection Report
    3. Initial Model Training Code, Model Validation and Evaluation Report
5. Model Optimization and Tuning Phase
    1. Hyperparameter Tuning Documentation
    2. Performance Metrics Comparison Report
    3. Final Model Selection Justification
6. Results
    1. Output Screenshots
7. Advantages & Disadvantages
8. Conclusion
9. Future Scope
10. Appendix
    1. Source Code
    2. GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

The Liver Patient Identification – prediction of Liver diseases represents a significant public health challenge, impacting millions of individuals worldwide. Conditions such as hepatitis, cirrhosis, and liver cancer can lead to severe complications and mortality if not identified and managed early. The liver plays a crucial role in various bodily functions, including metabolism, detoxification, and production of essential proteins. As such, the timely identification of liver diseases is vital for effective treatment and improved patient outcomes.

This project aims to develop a comprehensive Liver Patient Identification System that leverages clinical data, laboratory results, and imaging studies to accurately identify individuals at risk for liver diseases. By utilizing advanced data analysis techniques and machine learning algorithms, the project seeks to enhance the early detection of liver conditions, ultimately leading to better management strategies and improved health outcomes for patients.

1.2 Objectives

The main objectives of this project are:

- **Early Detection**: To develop a system that enables the early identification of patients at risk for liver diseases, thereby facilitating timely intervention and treatment.
- **Risk Assessment**: To implement algorithms that assess individual risk factors for liver diseases, helping healthcare providers identify high-risk patients more effectively.
- **Improved Patient Outcomes**: To enhance patient management strategies through accurate identification, leading to better health outcomes and reduced morbidity associated with liver diseases.
- **Collaboration with Healthcare Providers**: To work closely with healthcare professionals to ensure the system meets clinical needs and can be seamlessly integrated into existing workflows.
- **Education and Awareness:** To raise awareness about liver diseases and the importance of early identification among both healthcare providers and the general public.

- **Research Contribution**: To contribute to the body of knowledge regarding liver diseases by analysing patterns and trends in the identified patient population.

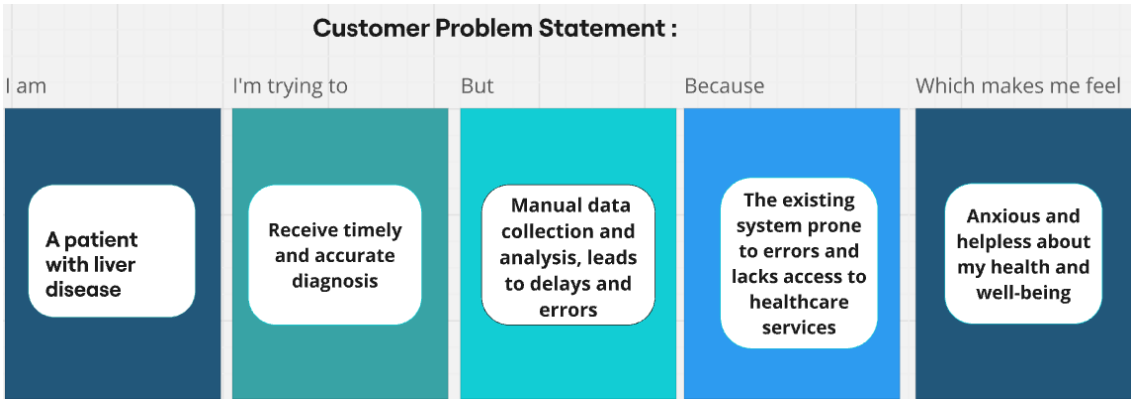
2. Project Initialization and Planning Phase

2.1 Define Problem Statement

Date	03-10-2024
Team ID	LTVIP2024TMID24892
Project Name	Liver Patient Identification – prediction of liver disease
Maximum Marks	3 Marks

Define Problem Statements (Customer Problem Statement Template):

The current system for liver patient identification is often inadequate, leading to delayed diagnosis and poor health outcomes. Many patients with liver disease are not diagnosed until they exhibit symptoms of end-stage de compensation, which can be fatal. The current system relies heavily on manual data collection and analysis, which can be time-consuming and prone to errors. Many patients with liver disease do not have access to healthcare services, making it difficult for them to receive timely diagnosis and treatment.



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A healthcare provider	Identify liver patients in a timely manner	The current system relies on manual data collection and analysis, leading to delays and errors	The existing system is inadequate and inefficient	Concerned about the well-being of my patients
PS-2	A patient with liver disease	Receive timely and accurate diagnosis and treatment	Manual data collection and analysis, leading to delays and errors	The existing system is prone to errors and lacks access to healthcare services	Anxious and helpless about my health and well-being

2.2 Project Proposal (Proposed Solution):

Date	03-10-2024
Team ID	LTVIP2024TMID24892
Project Title	Liver Patient Identification – prediction of liver disease
Maximum Marks	3 Marks

Project Proposal (Proposed Solution) Report:

The system we are proposing utilize various machine learning algorithms and techniques to identify liver patients. This system includes a machine learning-based approach for liver disease diagnosis using clinical and laboratory feature. A machine learning-based approach for liver disease diagnosis using clinical and laboratory features involves training machine learning algorithms on large datasets of clinical and laboratory features to identify patterns and relationships that can predict liver. By using this model we can predict which patient has liver disease accurately.

Project Overview	
Objective	Identifying wheather a person has a liver disease or not.
Scope	This project can identify the liver patient based on the clinical reports.
Problem Statement	
Description	Many patients with liver disease are not diagnosed until they exhibit symptoms of end-stage de compensation, which can be fatal. The current system relies heavily on manual data collection and analysis, which can be time-consuming and prone to errors.

Impact	It can be time-consuming and prone to errors. Many patients with liver disease do not have access to healthcare services, making it difficult for them to receive timely diagnosis and treatment.
Proposed Solution	
Approach	The system we are proposing utilize various machine learning algorithms and techniques to identify liver patients. This system includes a machine learning-based approach for liver disease diagnosis using clinical and laboratory feature.
Key Features	By using this model we can predict which patient has liver disease accurately. 11. Low cost 12. Less time consuming 13. Timely diagnosis

Resource Requirements:

Resource Type	**Description**	**Specification/Allocation**
Hardware		
Computing Resources	CPU/GPU specifications, number of cores	e.g., 2 x NVIDIA V100 GPUs
Memory	RAM specifications	e.g., 8 GB
Storage	Disk space for data, models, and logs	e.g., 1 TB SSD
Software		
Frameworks	Python frameworks	e.g., Flask

Libraries	Additional libraries	e.g., scikit-learn, pandas, numpy
Development Environment	IDE, version control	e.g., Jupyter Notebook, Git
Data		
Data	Source, size, format	e.g., Kaggle dataset , 584 reports, CSV

2.3 Initial Project Planning

Date	28-09-2024
Team ID	LTVIP2024TMID24892
Project Name	Liver Patient Identification – prediction of liver disease
Maximum Marks	4 Marks

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sp Sta
Sprint-1	Data Collection and Preprocessing	USN-1	Understanding & loading data	1	Low	Suchendra	2024
Sprint-1	Data Collection and Preprocessing	USN-2	Data cleaning	1	Medium	Subash	2024
Sprint-2	Data Collection and Preprocessing	USN-3	EDA	2	High	Mohan	2024

Sprint-3	Model Development	USN-4	Training the model	2	High	Venkatesh	2024
Sprint-3	Model tuning and testing	USN-5	Model tuning	1	High	Subash, Sandeep	2024
Sprint-4	Web integration and Deployment	USN-6	Building HTML templates	2	Medium	Sandeep	2024
Sprint-5	Web integration and Deployment	USN-7	Deployment	2	High	Venkatesh	2024

3.Data Collection and Preprocessing Phase

3.1. Data Collection Plan and Raw Data Sources Identified

Date	03-10-2024
Team ID	LTVIP2024TMID24892
Project Title	Liver Patient Identification – prediction of liver patient
Maximum Marks	2 Marks

Data Collection Plan & Raw Data Sources Identification:

A data collection plan for liver patient identification involves gathering relevant data from various sources to identify potential liver patients. Raw data sources can include medical claims data, laboratory results. Additionally, data from wearable devices, mobile apps, and patient-reported outcomes can also be used. The data collection plan should ensure that the data is accurate, complete, and relevant to the research question.

Data Collection Plan:

Section	Description
Project Overview	The objective of this machine learning project is to identify the persons who have liver disease based on the laboratories records.
Data Collection Plan	The data will be searched from the kaggle and other websites and indian liver patient websites.

Raw Data Sources Identified	The dataset has been collected from kaggle and from indian liver patient website.

Raw Data Sources Report:

Source Name	Description	Location/URL	Format	Size	Access Permissions
ILPD (Indian Liver Patient Dataset)	The Indian Liver Patient Dataset (ILPD) is a valuable collection of medical records that includes 583 entries. It features ten key attributes age and gender, as well as crucial laboratory measurements such as total bilirubin, alkaline phosphatase.	https://archive.ics.uci.edu/dataset/225/ilpd+indian+liver+patient+dataset	CSV	23.4 KB	Public
Kaggle dataset	This dataset also contains the same common values as	https://www.kaggle.com/datasets/abhi8923shriv/liver-disease-patient-dataset	CSV	1.2 MB	Public

3.2. Data Quality Report

Date	03-10-2024
Team ID	LTVIP2024TMID24892
Project Title	Liver Patient Identification – prediction of liver patient
Maximum Marks	2 Marks

Data Quality Report:

The Data Quality Report Template will summarize data quality issues from the selected source, including severity levels and resolution plans. It will aid in systematically identifying and rectifying data discrepancies.

Data Source	Data Quality Issue	Severity	Resolution Plan
Kaggle Dataset	Missing values are in the ['Albumin_and_Globulin_Ratio']	Low	Used mean/median imputation.
Kaggle Dataset	Categorical data in the dataset	Moderate	There is no need of Standardization and Normalization for the dataset, as we have used Ensemble Technique.

3.3. Data Collection and Preprocessing Phase

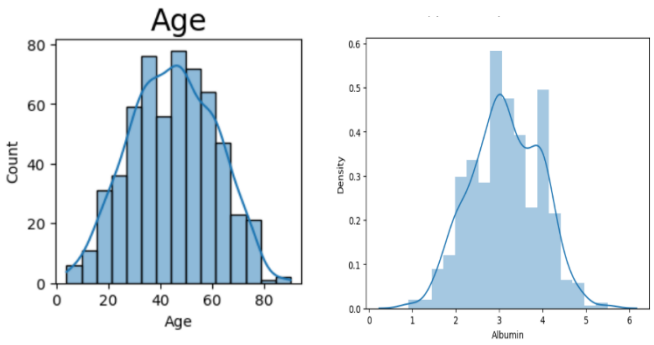
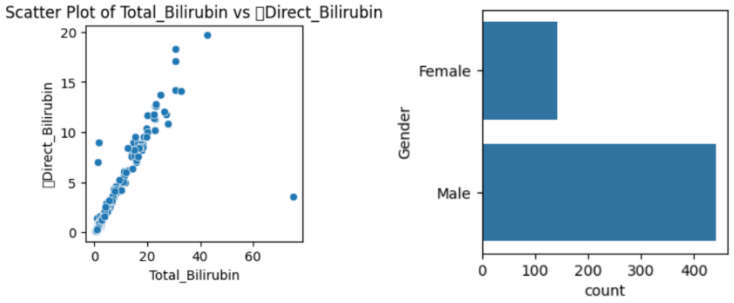
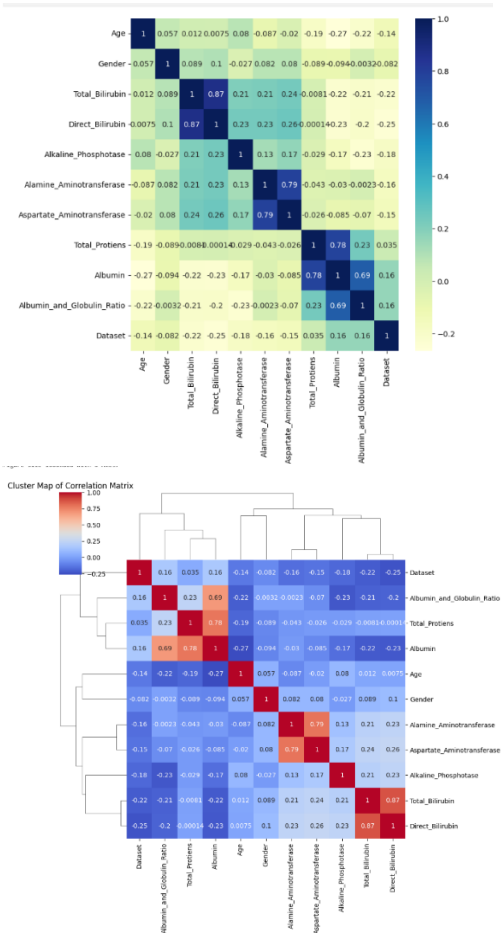
Date	03-10-2024
Team ID	LTVIP2024TMID24892
Project Title	Liver Patient Identification – prediction of liver patient
Maximum Marks	6 Marks

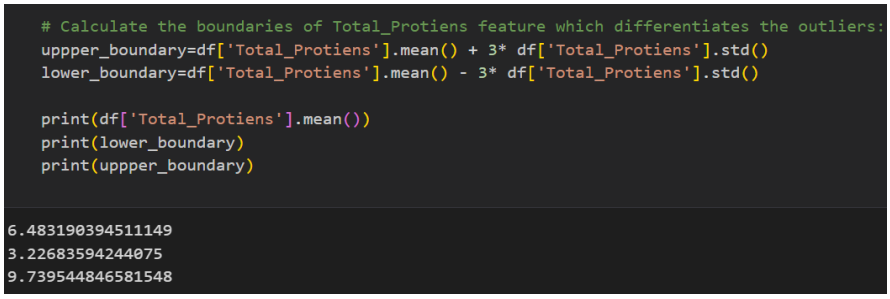
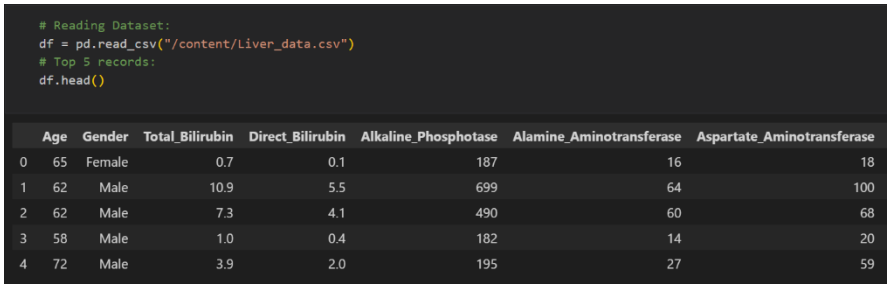
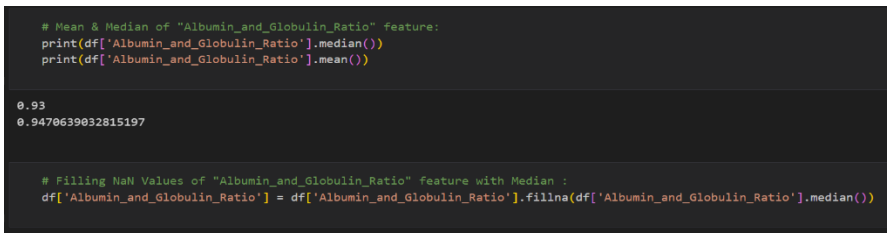
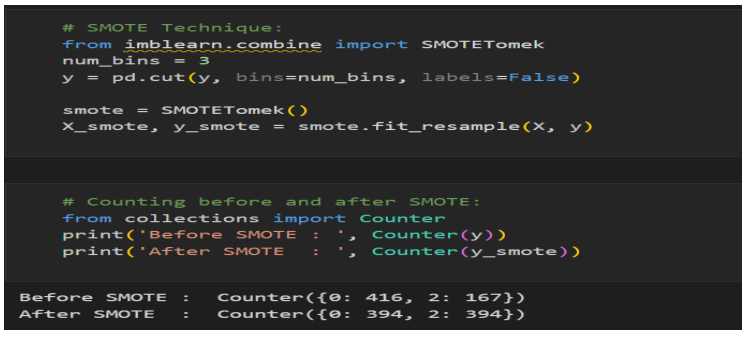
Data Exploration and Preprocessing Report

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modelling, and forming a strong foundation for insights and predictions.

Section	**Description**
Data Overview	Dimension: 584rows X 12columns Descriptive statistics:

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase
count	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000
mean	44.746141	3.298799	1.486106	290.576329	80.713551	109.910806
std	16.189833	6.209522	2.808498	242.937989	182.620356	288.918529
min	4.000000	0.400000	0.100000	63.000000	10.000000	10.000000
25%	33.000000	0.800000	0.200000	175.500000	23.000000	25.000000
50%	45.000000	1.000000	0.300000	208.000000	35.000000	42.000000
75%	58.000000	2.600000	1.300000	298.000000	60.500000	87.000000
max	90.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000

Univariate Analysis	
Bivariate Analysis	
Multivariate Analysis	

Outliers and Anomalies	
Data Preprocessing Code Screenshots	
Loading Data	 Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase 0 65 Female 0.7 0.1 187 16 18 1 62 Male 10.9 5.5 699 64 100 2 62 Male 7.3 4.1 490 60 68 3 58 Male 1.0 0.4 182 14 20 4 72 Male 3.9 2.0 195 27 59
Handling Missing Data	
Data Transformation	There is no need of Standardization and Normalization of our dataset, as we using Ensemble Technique.
Feature Engineering	

Save Processed Data	-

4.Model Development Phase

4.1. Feature Selection Report

Date	03-10-2024
Team ID	LTVIP2024TMID24892
Project Title	Liver Patient Identification – prediction of liver patient
Maximum Marks	5 Marks

Feature Selection Report:

Feature selection plays a important role in Improves Model Performance, Reduces Training Time, Increases Model Interpretability, Facilitates Better Data Understanding.

Feature	Description	Selected (Yes/No)	Reasoning
Age	Age is an important factor in liver disease diagnosis.	Yes	Liver disease can affect people of different ages.
Total Bilirubin	Total bilirubin is a key indicator of liver function.	Yes	Elevated levels can indicate liver disease.

Direct Bilirubin	Direct bilirubin is a specific type of bilirubin that is directly conjugated with glucuronic acid.	No	Elevated levels can indicate liver disease but not accurately.
Alkaline Phosphatase	Alkaline phosphatase is an enzyme that is elevated in liver disease	Yes	This feature can be used as a diagnostic marker.
Gender	Male or Female	Yes	-
Total Protein	Can be used to indicate liver disease	Yes	Elevated levels can indicate liver disease.
Alanine Aminotransferase	Alanine aminotransferase is an enzyme that is elevated in liver disease	Yes	This feature can be used as a diagnostic marker.

4.2 Model Selection Report

Date	03-10-2024
Team ID	LTVIP2024TMID24892
Project Title	Liver Patient Identification – prediction of liver patient
Maximum Marks	6 Marks

Model Selection Report

In the forthcoming Model Selection Report, various models will be outlined, detailing their descriptions, hyperparameters, and performance metrics, including Accuracy or F1 Score. This comprehensive report will provide insights into the chosen models and their effectiveness.

Model Selection Report:

Model	Description	Hyperparameters	Performance Metric (e.g., Accuracy, F1 Score)
Random forest	Random Forest can be effectively used in a liver patient identification model to improve diagnosis accuracy, reduce costs, and enhance patient	14. n_estimators 15. criterion 16. max_depth 17. min_samples_split 18. min_samples_leaf	Accuracy = 84%

	outcomes.		
SVM	SVMs can handle high-dimensional data with ease, improving model performance. This is particularly useful in liver patient data, which often involves a large number of features	19. Kernel 20. C 21. Gama	Accuracy = 73%
KNN	KNN can handle high-dimensional data with ease, reducing the curse of dimensionality and improving model performance.	1. n_neighbous 2. weights **3.** P	Accuracy = 76%

4.3 Initial Model Training Code, Model Validation and Evaluation Report

Date	03-10-2024
Team ID	LTVIP2024TMID24892
Project Title	Liver Patient Identification – prediction of liver patient
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in this. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

Train Test split for all models:

```
# Train Test Split:
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X_smote,y_smote, test_size=0.2, random_state=42)
```

Random forest model

```
# RandomForestClassifier:
from sklearn.ensemble import RandomForestClassifier
RandomForest = RandomForestClassifier()
RandomForest = RandomForest.fit(X_train,y_train)

# Predictions:
y_pred = RandomForest.predict(X_test)

# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

KNN Model:

```
#kNearest Neighbours
from sklearn.neighbors import KNeighborsClassifier
kneighbours = KNeighborsClassifier()
kneighbours = kneighbours.fit(X_train,y_train)

# Predictions:
y_pred = kneighbours.predict(X_test)

# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

Support vector Machine Model:

```
#kNearest Neighbours
from sklearn.neighbors import KNeighborsClassifier
kneighbours = KNeighborsClassifier()
kneighbours = kneighbours.fit(X_train,y_train)

# Predictions:
y_pred = kneighbours.predict(X_test)

# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Random Forest	`print(classification_report(y_test,y_pred))` Accuracy: 0.8291139240506329 precision recall f1-score support 0 0.91 0.74 0.82 82 2 0.77 0.92 0.84 76 accuracy 0.83 158 macro avg 0.84 0.83 0.83 158 weighted avg 0.84 0.83 0.83 158	82%	`confusion_matrix(y_test,y_pred)` array([[61, 21], [6, 70]])
Support Vector Machine	`print(classification_report(y_test,y_pred))` Accuracy: 0.7531645569620253 precision recall f1-score support 0 0.86 0.62 0.72 82 2 0.69 0.89 0.78 76 accuracy 0.75 158 macro avg 0.78 0.76 0.75 158 weighted avg 0.78 0.75 0.75 158	75%	`confusion_matrix(y_test,y_pred)` array([[40, 42], [9, 67]])
KNN	`print(classification_report(y_test,y_pred))` Accuracy: 0.6772151898734177 precision recall f1-score support 0 0.82 0.49 0.61 82 2 0.61 0.88 0.72 76 accuracy 0.68 158 macro avg 0.72 0.68 0.67 158 weighted avg 0.72 0.68 0.67 158	67%	`print(confusion_matrix(y_test,y_pred))` [[51 31] [8 68]]

5.Model Optimization and Tuning Phase

5.1. Hyperparameter Tuning Documentation

Date	03-10-2024
Team ID	LTVIP2024TMID24892
Project Title	Liver Patient Identification – prediction of liver patient
Maximum Marks	10 Marks

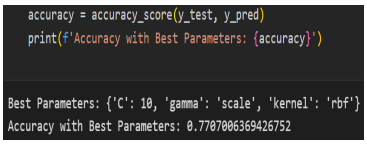
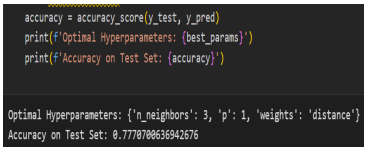
Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

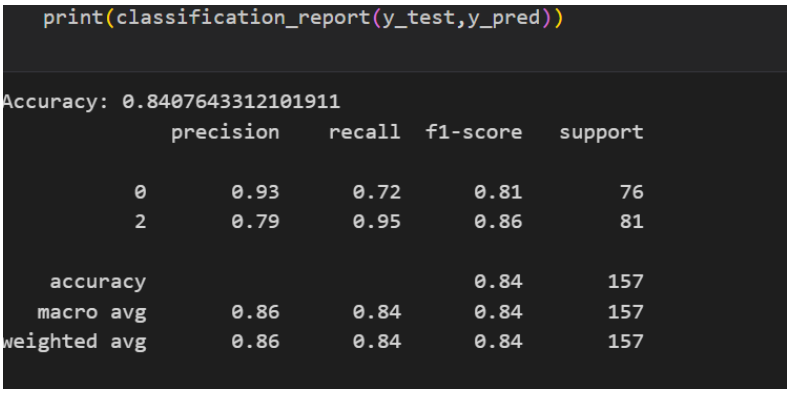
Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values

Random Forest	`rf_classifier = RandomForestClassifier()`
	
	`param_grid = {`
	`'n_estimators': [50, 100, 200],`
	`'criterion': ['gini', 'entropy'],`
	`'max_depth': [None, 10, 20, 30],`
	`'min_samples_split': [2, 5, 10],`
	`'min_samples_leaf': [1, 2, 4],`
	`}`
		`accuracy = accuracy_score(y_test, y_pred)`
		`print(f'Optimal Hyperparameters: {best_params}')`
		`print(f'Accuracy on Test Set: {accuracy}')`
		
		`Optimal Hyperparameters: {'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 1,`
		`Accuracy on Test Set: 0.8343949044585988`

SVM		
KNN		

5.2 Performance Metrics Comparison Report (2 Marks):

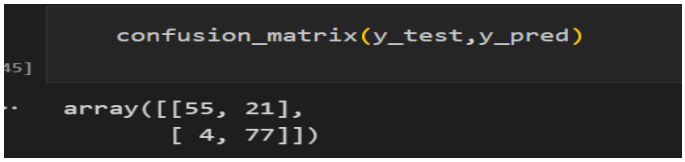
Model	Baseline Metric, optimal metrics
Random Forest	

SVM	`print(classification_report(y_test,y_pred))` Accuracy: 0.732484076433121 precision recall f1-score support 0 0.81 0.58 0.68 76 2 0.69 0.88 0.77 81 accuracy 0.73 157 macro avg 0.75 0.73 0.72 157 weighted avg 0.75 0.73 0.73 157

KNN	`print('Accuracy:', accuracy_score(y_test,y_pred))`
	`print(classification_report(y_test,y_pred))`
	
	Accuracy: 0.7643312101910829
	precision recall f1-score support
	
	0 0.87 0.61 0.71 76
	2 0.71 0.91 0.80 81
	
	accuracy 0.76 157
	macro avg 0.79 0.76 0.76 157
	weighted avg 0.79 0.76 0.76 157

5.3 Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest	`print(classification_report(y_test,y_pred))` Accuracy: 0.8407643312101911 precision recall f1-score support 0 0.93 0.72 0.81 76 2 0.79 0.95 0.86 81 accuracy 0.84 157 macro avg 0.86 0.84 0.84 157 weighted avg 0.86 0.84 0.84 157

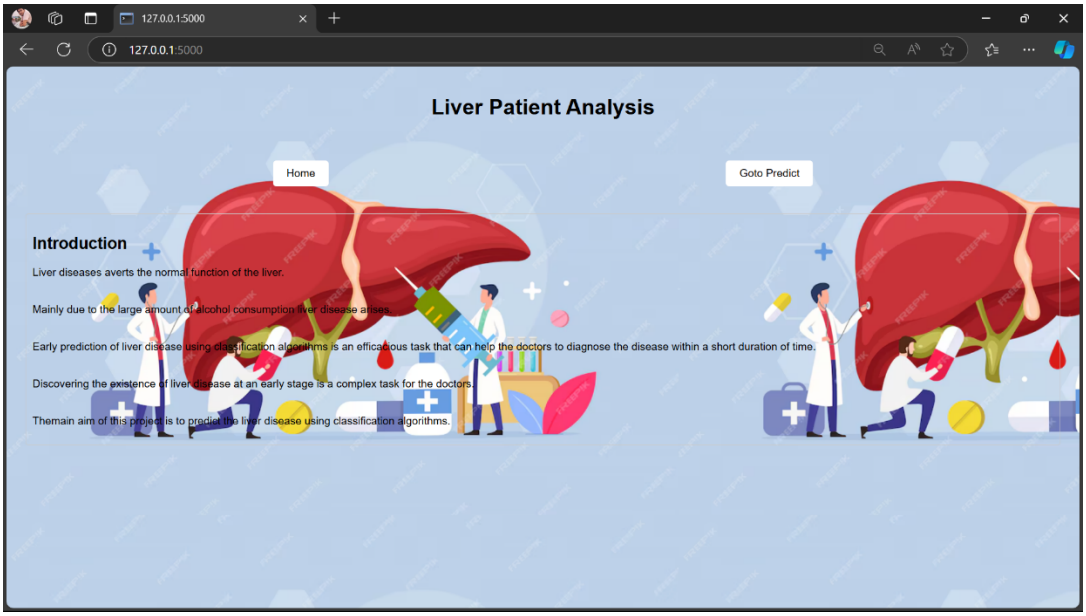
	 4. This model has been selected because it has the high accuracy and f1-score compared to the other model mentioned above.

NOTE: I have done other models like Gradient Boosting Classifier, AdaBoost Classifier these models will be available in the liver.ipynb file.

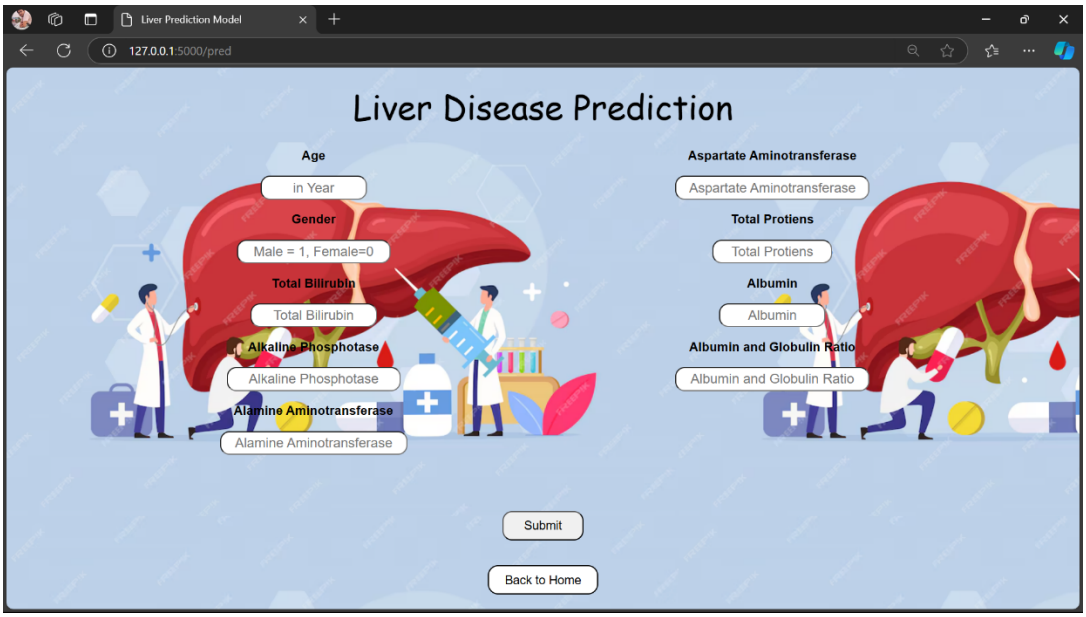
6.Results:

Output screens:

#Home Page

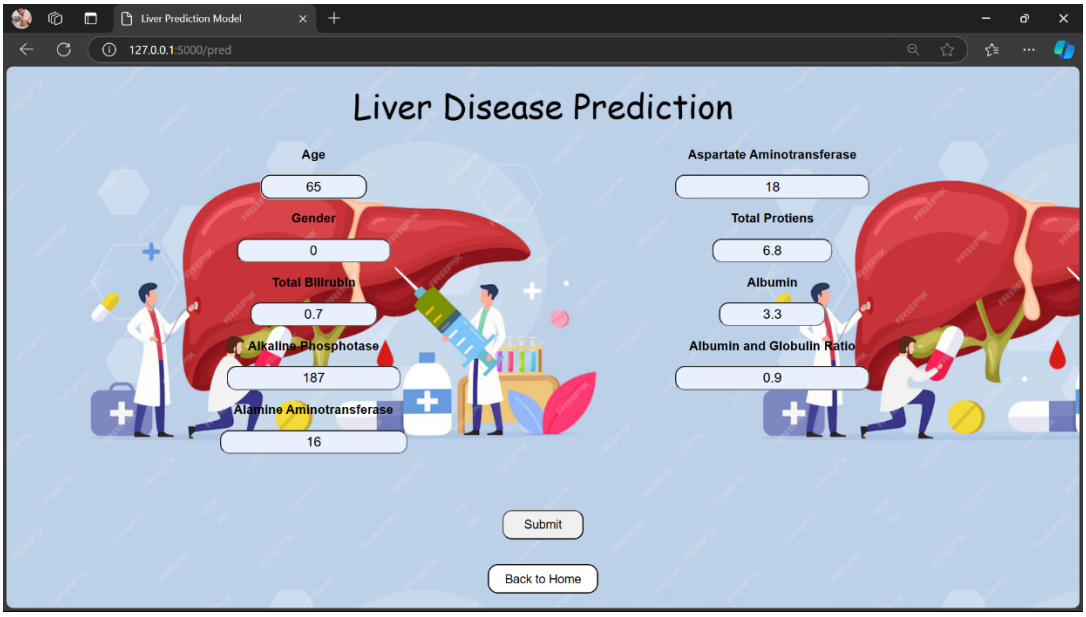


#Index Page



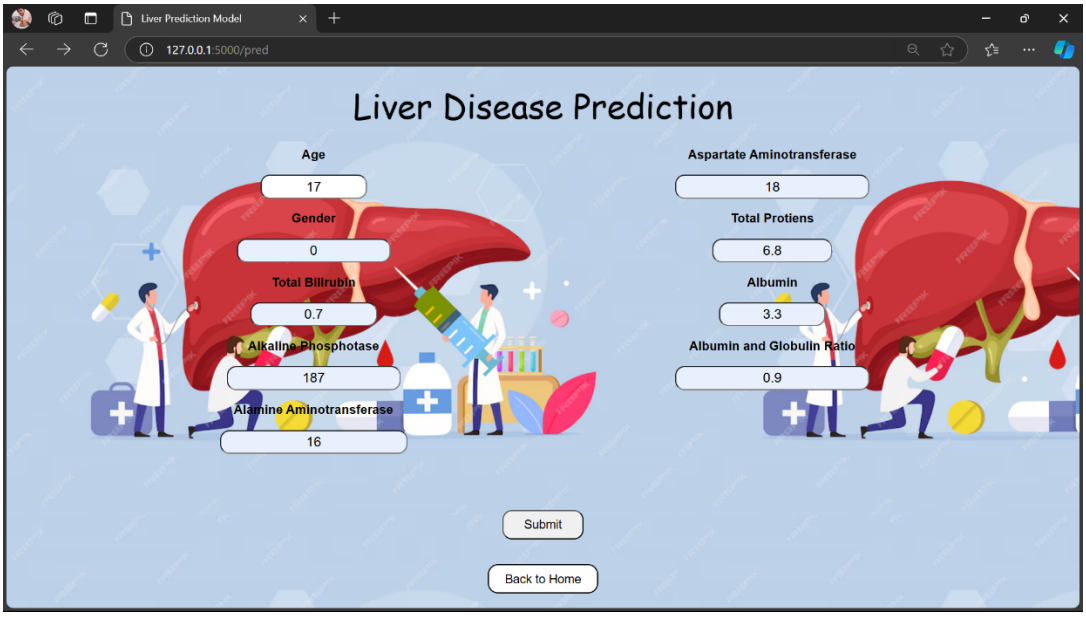
After Execution:

#Input values (If the patient doesn’t have a liver disease)

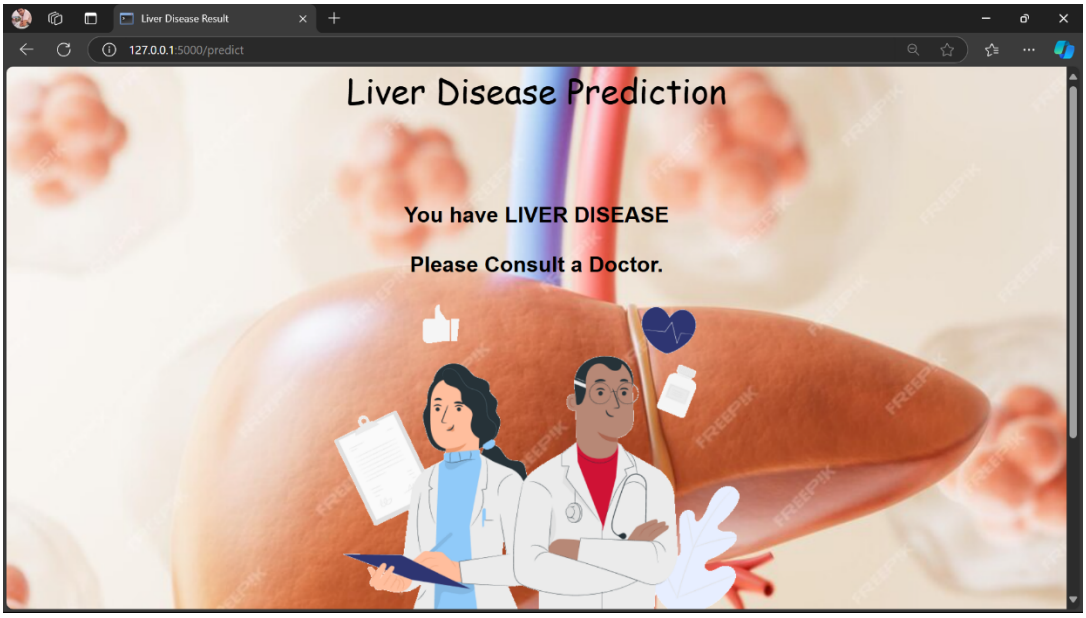


#Output (If the patient doesn’t have a liver disease)

#Input values (If the patient have liver disease)



#Output (Patient have a liver disease)



#pyNoteBook Output Screenings:

RandomForestClassifier

```python
# RandomForestClassifier:
from sklearn.ensemble import RandomForestClassifier
RandomForest = RandomForestClassifier()
RandomForest = RandomForest.fit(X_train,y_train)

# Predictions:
y_pred = RandomForest.predict(X_test)

# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
Accuracy: 0.8407643312101911
              precision    recall  f1-score   support

           0       0.90      0.75      0.82        76
           2       0.80      0.93      0.86        81

    accuracy                           0.84       157
   macro avg       0.85      0.84      0.84       157
weighted avg       0.85      0.84      0.84       157
```

Support vector Machine

```
[ ]  from sklearn import svm
     supportvector = svm.SVC()
     supportvector = supportvector.fit(X_train,y_train)

     # Predictions:
     y_pred = supportvector.predict(X_test)

     # Performance:
     print('Accuracy:', accuracy_score(y_test,y_pred))
     print(classification_report(y_test,y_pred))
```

```
Accuracy: 0.6942675159235668
              precision    recall  f1-score   support

           0       0.75      0.55      0.64        76
           2       0.66      0.83      0.74        81

    accuracy                           0.69       157
   macro avg       0.71      0.69      0.69       157
weighted avg       0.71      0.69      0.69       157
```

SMOTE Technique

```
[ ]  # SMOTE Technique:
     from imblearn.combine import SMOTETomek
     num_bins = 3
     y = pd.cut(y, bins=num_bins, labels=False)

     smote = SMOTETomek()
     X_smote, y_smote = smote.fit_resample(X, y)
```

```
     # Counting before and after SMOTE:
     from collections import Counter
     print('Before SMOTE : ', Counter(y))
     print('After SMOTE  : ', Counter(y_smote))
```

```
Before SMOTE :  Counter({0: 416, 2: 167})
After SMOTE  :  Counter({0: 391, 2: 391})
```

```
[ ]  # Train Test Split:
     from sklearn.model_selection import train_test_split
     X_train,X_test,y_train,y_test = train_test_split(X_smote,y_smote, test_size=0.2, random_state=42)
```

KNN

```
[ ]  from sklearn.neighbors import KNeighborsClassifier
     kneighbours = KNeighborsClassifier()
     kneighbours = kneighbours.fit(X_train,y_train)

     # Predictions:
     y_pred = kneighbours.predict(X_test)

     # Performance:
     print('Accuracy:', accuracy_score(y_test,y_pred))
     print(classification_report(y_test,y_pred))
```

```
Accuracy: 0.7452229299363057
              precision    recall  f1-score   support

           0       0.80      0.63      0.71        76
           2       0.71      0.85      0.78        81

    accuracy                           0.75       157
   macro avg       0.76      0.74      0.74       157
weighted avg       0.75      0.75      0.74       157
```

7. Advantages & Disadvantages

- **Advantages:**

  - **Early Intervention:** By identifying patients at risk for liver diseases early, healthcare providers can initiate treatments sooner, potentially preventing disease progression and complications.

  - **Personalized Care:** The system can analyse individual patient data to provide tailored management plans based on specific risk factors, leading to more effective treatment strategies.

  - **Data-Driven Decisions:** Access to comprehensive data allows healthcare professionals to make informed decisions, improving diagnostic accuracy and treatment outcomes.

- **Resource Optimization:** By streamlining the identification process, healthcare facilities can optimize their resources, ensuring that patients who need immediate care receive it promptly.

- **Enhanced Monitoring:** The system can facilitate ongoing monitoring of patients with liver conditions, allowing for timely adjustments to treatment plans based on their changing health status.

- **Increased Awareness:** The project can help raise awareness about liver health among both patients and healthcare providers, promoting preventive measures and lifestyle changes that can reduce the risk of liver diseases.

- **Research Opportunities:** The data collected through the system can contribute to research on liver diseases, helping to identify trends, risk factors, and treatment efficacy.

➢ Disadvantages:

- **High Implementation Costs**: Setting up a comprehensive identification system can involve significant financial investment. This includes costs for technology, software, training staff, and maintaining the system. Smaller healthcare facilities may struggle to afford these expenses.

- **Privacy Concerns:** Handling sensitive patient data raises privacy issues. There is a risk of data breaches or unauthorized access, which could compromise patient confidentiality and trust in the healthcare system.

- **Over-Reliance on the System:** If healthcare providers depend too heavily on the identification system, they might overlook clinical signs or symptoms that are not captured by the system. This could lead to missed diagnoses or inadequate patient care.

- **Technical Issues:** Like any technology, the system may experience technical problems, such as software bugs or system outages. These issues can disrupt patient identification processes and delay care.

- **Limited Access:** Patients in rural or underserved areas may not have access to the necessary technology or infrastructure to utilize the system effectively. This can create disparities in care and limit the system's overall effectiveness.

- **Resistance to Change:** Some healthcare providers may be resistant to adopting a new system, especially if they are accustomed to traditional methods. This resistance can hinder the successful implementation and utilization of the identification system.

8. Conclusion:

- In conclusion, while a liver patient identification system offers several advantages like early detection and personalized treatment, it also comes with notable disadvantages such as high implementation costs and privacy concerns.
- Balancing these pros and cons is essential for healthcare providers to ensure that the system enhances patient care without compromising safety or accessibility.

9. Future Scope:

The future scope of a liver patient identification system is quite promising, with several key areas for potential growth and development:

- **Integration with Artificial Intelligence:** Future systems could leverage AI and machine learning to enhance diagnostic accuracy. By analysing large datasets, AI can identify patterns and predict liver disease progression, leading to more effective early interventions.

- **Telemedicine and Remote Monitoring:** With the rise of telehealth, integrating the identification system with remote monitoring tools can allow healthcare providers to track patient health in real-time. This can lead to timely adjustments in treatment plans and improved patient engagement.

- **Personalized Medicine:** As research in genomics and personalized medicine advances, the identification system could incorporate genetic data to provide tailored treatment options. This would enable healthcare providers to understand how individual patients may respond to specific therapies.

- **Improved Data Analytics:** Future systems may utilize advanced data analytics to provide insights into patient populations, helping healthcare providers identify trends in liver disease and optimize resource allocation. This can inform public health strategies and preventive measures.

- **Enhanced Patient Education:** The system can evolve to include educational components that inform patients about liver health, risk factors, and lifestyle changes. Empowering patients with knowledge can lead to better health outcomes and proactive management of their conditions.

- **Interoperability with Other Health Systems**: Ensuring that the liver patient identification system can communicate seamlessly with other healthcare systems will be crucial. This interoperability can facilitate comprehensive patient care by allowing providers to access a patient's full medical history.

- **Focus on Preventive Care:** Future developments may emphasize preventive care by identifying at-risk populations and implementing screening programs. This proactive approach can help reduce the incidence of liver diseases through early detection and lifestyle modifications.

- **Regulatory and Ethical Frameworks:** As technology advances, establishing robust regulatory and ethical frameworks will be essential to address privacy concerns and ensure patient safety. This will help build trust in the system and encourage wider adoption.

10. APPENDIX

10.1 Source code:

App.py

```
from flask import Flask, render_template, request
import numpy as np
import pickle

app = Flask(__name__)
model = pickle.load(open('Liver2.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('home.html')

@app.route('/pred',methods=['GET'])
def index():
    return render_template('index.html')

@app.route("/predict", methods=['POST'])
def predict():

        Age = int(request.form['Age'])
        Gender = int(request.form['Gender'])
        Total_Bilirubin = float(request.form['Total_Bilirubin'])
        Alkaline_Phosphotase = int(request.form['Alkaline_Phosphotase'])
        Alamine_Aminotransferase = int(request.form['Alamine_Aminotransferase'])
```

```
        Aspartate_Aminotransferase = int(request.form['Aspartate_Aminotransferase'])
        Total_Protiens = float(request.form['Total_Protiens'])
        Albumin = float(request.form['Albumin'])
        Albumin_and_Globulin_Ratio =
float(request.form['Albumin_and_Globulin_Ratio'])


        values =
np.array([[Age,Gender,Total_Bilirubin,Alkaline_Phosphotase,Alamine_Aminotran
sferase,Aspartate_Aminotransferase,Total_Protiens,Albumin,Albumin_and_Globuli
n_Ratio]])
        prediction = model.predict(values)


        if prediction == 2:
            return render_template('noChance.html')
        else:
            return render_template('chance.html')


if __name__ == "__main__":
    app.run(debug=True)
```

#Home.html

```
<!DOCTYPE html>
<html>


<head>
    <style>
        body {
```

```
        background-image: url("https://img.freepik.com/premium-
vector/liver-disease-treatment-design-concept-with-tiny-people_7087-
973.jpg?w=996");
        /* Light yellow background */
        font-family: Arial, Helvetica, sans-serif;
    }

    h1 {
        color: black;
        /* Purple header */
        text-align: center;
        padding: 20px;
    }

    .container {
        display: flex;
        justify-content: space-between;
        padding: 20px;
        margin: 0 auto;
        max-width: 800px;
        /* Adjust for desired width */
        animation: slide-in 1.5s;
    }

    .nav-buttons {
        background-color: white;
```

```
    /* Purple button background */
    color: black;
    padding: 10px 20px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    border-radius: 5px;
}

.nav-buttons:hover {
    background-color: lightblue;
    /* Darker purple on hover */
}

.content {
    padding: 20px;
    animation: slide-in 1.5s;
}

.introduction {
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    margin-bottom: 20px;
    animation: slide-in 1.5s;
```

```
        }

        @keyframes slide-in {
            from {
                transform: translateX(100%);
            }

            to {
                transform: translateX(0);
            }
        }

        a {
            text-decoration: none;
        }
    </style>
</head>

<body>

    <h1>Liver Patient Analysis</h1>

    <div class="container">
        <div>
            <a href="/" class="nav-buttons">Home</a>
        </div>
```

```
    <div>
        <a href="/pred" class="nav-buttons">Goto Predict</a>
    </div>
</div>

<div class="content">
    <div class="introduction">
        <h2>Introduction</h2>
        <p>Liver diseases averts the normal function of the liver.
<br><br><br> Mainly due to the large amount of
            alcohol
            consumption liver disease arises. <br><br><br> Early prediction of
liver disease using classification
            algorithms
            is
            an efficacious task that can help the doctors to diagnose the disease
within a short duration of time.
            <br><br><br>
            Discovering the existence of liver disease at an early stage is a
complex task for the doctors.
            <br><br><br>Themain
            aim of this project is to predict the liver disease using classification
algorithms.
        </p>
    </div>
</div>
```

```
</body>


</html>
```

#Index.html

```
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <title>Liver Prediction Model</title>


</head>


<body>
    <div class="container">
        <h2 class='container-heading'><span class="heading_font">Liver
Disease Prediction</span></h2>
        <form action="{{ url_for('predict') }}" method="POST">
            <div class="input-fields">
                <div class="left-column">
                    <h3>Age</h3>
                    <input id="first" name="Age" placeholder="in Year"
required="required">
                    <br>
                    <h3>Gender</h3>
                    <input id="second" name="Gender" placeholder="Male = 1,
Female=0" required="required">
```

```
            <br>
            <h3>Total Bilirubin</h3>
            <input id="third" name="Total_Bilirubin" placeholder="Total
Bilirubin" required="required">
            <br>
            <h3>Alkaline Phosphotase</h3>
            <input id="fourth" name="Alkaline_Phosphotase"
placeholder="Alkaline Phosphotase"
               required="required">
            <br>
            <h3>Alamine Aminotransferase</h3>
            <input id="fifth" name="Alamine_Aminotransferase"
placeholder="Alamine Aminotransferase"
               required="required">
         </div>
         <div class="right-column">
            <h3>Aspartate Aminotransferase</h3>
            <input id="sixth" name="Aspartate_Aminotransferase"
placeholder="Aspartate Aminotransferase"
               required="required">
            <br>
            <h3>Total Protiens</h3>
            <input id="seventh" name="Total_Protiens" placeholder="Total
Protiens" required="required">
            <br>
            <h3>Albumin</h3>
            <input id="eighth" name="Albumin" placeholder="Albumin"
required="required">
```

```
                <br>
                <h3>Albumin and Globulin Ratio</h3>
                <input id="ninth" name="Albumin_and_Globulin_Ratio"
placeholder="Albumin and Globulin Ratio"
                    required="required">
            </div>
        </div>
        <br><br><br><br>
        <div>
            <button id="sub" type="submit ">Submit</button>
        </div>
        <br>
        <br>
        <div>
            <a href="/" class="nav-buttons">Back to Home</a>
        </div>
    </form>
</div>



<style>
    /* Heading Font */
    .container-heading {
        margin: 0;
    }
```

```
.heading_font {
    color: black;
    font-family: 'Pacifico', cursive;
    font-size: 50px;
    font-weight: normal;
}

/* Box */
#first {
    border-radius: 14px;
    height: 30px;
    width: 150px;
    font-size: 20px;
    text-align: center;
}

#second {
    border-radius: 14px;
    height: 30px;
    width: 220px;
    font-size: 20px;
    text-align: center;
}

#third {
    border-radius: 14px;
```

```
    height: 30px;
    width: 180px;
    font-size: 20px;
    text-align: center;
}

#fourth {
    border-radius: 14px;
    height: 30px;
    width: 250px;
    font-size: 20px;
    text-align: center;
}

#fifth {
    border-radius: 14px;
    height: 30px;
    width: 270px;
    font-size: 20px;
    text-align: center;
}

#sixth {
    border-radius: 14px;
    height: 30px;
    width: 280px;
```

```
    font-size: 20px;
    text-align: center;
}

#seventh {
    border-radius: 14px;
    height: 30px;
    width: 170px;
    font-size: 20px;
    text-align: center;
}

#eighth {
    border-radius: 14px;
    height: 30px;
    width: 150px;
    font-size: 20px;
    text-align: center;
}

#ninth {
    border-radius: 14px;
    height: 30px;
    width: 280px;
    font-size: 20px;
    text-align: center;
```

```
}

/* Submit Button */
#sub {
    width: 120px;
    height: 43px;
    text-align: center;
    border-radius: 14px;
    font-size: 18px;
}

#hom {
    width: 120px;
    height: 43px;
    text-align: center;
    border-radius: 14px;
    font-size: 18px;
}

/* Slide Bar Animation */
.container {
    position: relative;
    animation: slide-in 1.5s;
}

@keyframes slide-in {
```

```
    from {
        transform: translateX(100%);
    }

    to {
        transform: translateX(0);
    }
}

/* Add a delay to the animation for each input field */
#first {
    animation-delay: 0.5s;
}

#second {
    animation-delay: 1s;
}

#third {
    animation-delay: 1.5s;
}

#fourth {
    animation-delay: 2s;
}
```

```
#fifth {
    animation-delay: 2.5s;
}

#sixth {
    animation-delay: 3s;
}

#seventh {
    animation-delay: 3.5s;
}

#eighth {
    animation-delay: 4s;
}

#ninth {
    animation-delay: 4.5s;
}

/* Background Image */
body {
    background-image: url("https://img.freepik.com/premium-vector/liver-disease-treatment-design-concept-with-tiny-people_7087-973.jpg?w=996");
}
```

```
/* Color */
body {
    font-family: Arial, Helvetica, sans-serif;
    text-align: center;
    margin: 0;
    padding: 0;
    width: 100%;
    height: 100vh;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    }

/* Container */
.container {
    width: 80%;
    margin: 0 auto;
    padding: 20px;
    border-radius: 10px;

}

/* Input Fields */
.input-fields {
```

```
    display: flex;
    justify-content: space-between;
    flex-wrap: wrap;
}

.left-column,
.right-column {
    width: 45%;
    margin: 10px;
}

/* Adding a delay to the animation for each input field */
.left-column input {
    animation-delay: 0.5s;
}

.right-column input {
    animation-delay: 2.5s;
}

#sub {
    animation-delay: 5s;
}
.nav-buttons {
    background-color: white;
    color: black;
```

```
            padding: 10px 20px;
            text-decoration: none;
            display: inline-block;
            width: 120px;
            height: 20px;
            text-align: center;
            border-radius: 15px;
            font-size: 18px;
            border-color: black;
            border: 2px solid black;
        }

        .nav-buttons:hover {
            background-color: lightblue;
        }
    </style>
</body>

</html>
<!DOCTYPE html>
<html>
```

#Chance.html

```
<!DOCTYPE html>
<html lang="en">

<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Liver Disease Result</title>
</head>

<body>

    <div class="container">
        <form action="{{ url_for('predict')}}" method="post">
            <h2 class='container-heading'><span class="heading_font">Liver
Disease Prediction</span></h2>

            <br><br>

            <div class="results">
                <h2>{{ prediction }}</h2>
                <h1><span class='safe'>Good to hear<br><br>You DON'T have
LIVER DISEASE.</span></h1>
            </div>
            <br><br><br>
            <div class="containers">
                <div>
                    <a href="/" class="nav-buttons">Home</a>
                </div>
                <br><br><br>
                <div>
```

```
                <a href="/pred" class="nav-buttons">Goto Predict</a>
            </div>
        </div>

    </form>

</div>
<div>
    <br><br> <br><br><br><br>
</div>

<style>
    /* Background Image */
    body {
        background-image: url("https://img.freepik.com/premium-vector/healthy-happy-smiling-cute-liver-character-vector-illustration-flat-cartoon-style_192280-535.jpg?w=740");
        background-position: center;
        background-repeat: no-repeat;
        background-size: cover;
        /* Add this property to scale the image to cover the full screen */
        height: 100%;
        /* Set the height to 100 viewport height */
        margin: 0;
    }
```

```
/* Color */
body {
    font-family: Arial, Helvetica, sans-serif;
    text-align: center;
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
}

/* Heading Font */
.container-heading {
    margin: 0;
}

.heading_font {
    color: black;
    font-family: 'Pacifico', cursive;
    font-size: 50px;
    font-weight: normal;
}

/* Slide Bar Animation */
.container {
    position: relative;
    animation: slide-in 1.5s;
```

```
}

@keyframes slide-in {
    from {
        transform: translateX(100%);
    }

    to {
        transform: translateX(0);
    }
}

.nav-buttons {
    background-color: white;
    /* Purple button background */
    color: black;
    padding: 10px 20px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    border-radius: 5px;
}

.nav-buttons:hover {
    background-color: lightblue;
```

```
            /* Darker purple on hover */
        }

        .containers {
            display: flex;
            justify-content: space-between;
            padding: 20px;
            margin: 0 auto;
            max-width: 800px;
            /* Adjust for desired width */
            animation: slide-in 1.5s;
        }
    </style>

</body>

</html>
```

#No Chance.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Liver Disease Result</title>
```

```
</head>

<body>

    <div class="container">
        <form action="{{ url_for('predict')}}" method="post">
            <h2 class='container-heading'><span class="heading_font">Liver Disease Prediction</span></h2>

            <br><br>

            <div class="results">
                <h2>{{ prediction }}</h2>
                <h1><span class='danger'><br><br>You have LIVER DISEASE <br><br>Please Consult a Doctor.</span>
                </h1>
                <img class="gif" src="{{ url_for('static', filename='dr.gif')}}" alt="LIVER Image">
            </div>
            <br><br><br>
            <div class="containers">
                <div>
                    <a href="/" class="nav-buttons">Home</a>
                </div>
                <br><br><br>
                <div>
                    <a href="/pred" class="nav-buttons">Goto Predict</a>
```

```
            </div>
        </div>

    </form>

</div>
<div>
    <br><br> <br><br><br><br><br><br><br><br>
</div>

<style>
    /* Background Image */
    body {
        background-image: url("https://img.freepik.com/premium-photo/liver-organ-with-health-care-concept-3d-rendering_772449-22029.jpg?w=1060");
        background-position: center;
        background-repeat: no-repeat;
        background-size: cover;
        /* Add this property to scale the image to cover the full screen */
        height: 100%;
        /* Set the height to 100 viewport height */
        margin: 0;

    }
```

```
/* Color */
body {
    font-family: Arial, Helvetica, sans-serif;
    text-align: center;
    margin: 0;
    padding: 0;
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
}

/* Heading Font */
.container-heading {
    margin: 0;
}

.heading_font {
    color: black;
    font-family: 'Pacifico', cursive;
    font-size: 50px;
    font-weight: normal;
}

/* Slide Bar Animation */
.container {
```

```
    position: relative;
    animation: slide-in 1.5s;
}

@keyframes slide-in {
    from {
        transform: translateX(100%);
    }

    to {
        transform: translateX(0);
    }
}

.nav-buttons {
    background-color: white;
    /* Purple button background */
    color: black;
    padding: 10px 20px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    border-radius: 5px;
}
```

```
        .nav-buttons:hover {
            background-color: lightblue;
        }

        .containers {
            display: flex;
            justify-content: space-between;
            padding: 20px;
            margin: 0 auto;
            max-width: 800px;
            /* Adjust for desired width */
            animation: slide-in 1.5s;
        }
    </style>

</body>

</html>
```

#Liver.py

```
# Importing Libraries:
import pandas as pd
import numpy as np
import pickle
```

```
# for displaying all feature from dataset:
pd.pandas.set_option('display.max_columns', None)

# Reading Dataset:
dataset = pd.read_csv("Dataset/Liver_data.csv")

# Filling NaN Values of "Albumin_and_Globulin_Ratio" feature with
Median:
dataset['Albumin_and_Globulin_Ratio'] =
dataset['Albumin_and_Globulin_Ratio'].fillna(dataset['Albumin_and_Globuli
n_Ratio'].median())

# Label Encoding:
dataset['Gender'] = np.where(dataset['Gender']=='Male', 1,0)

# Droping 'Direct_Bilirubin' feature:
dataset = dataset.drop('Direct_Bilirubin', axis=1)

# Independent and Dependent Feature:
X = dataset.iloc[:, :-1]
y = dataset.iloc[:, -1]

# SMOTE Technique:
from imblearn.combine import SMOTETomek
num_bins = 3 # Adjust this based on your data
y = pd.cut(y, bins=num_bins, labels=False)
```

```
smote = SMOTETomek()

X_smote, y_smote = smote.fit_resample(X, y)


# Train Test Split:

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test = train_test_split(X_smote,y_smote,
test_size=0.2, random_state=42)


# RandomForestClassifier:

from sklearn.ensemble import RandomForestClassifier

RandomForest = RandomForestClassifier()

RandomForest = RandomForest.fit(X_train,y_train)


# Creating a pickle file for the classifier

filename = 'Liver2.pkl'

pickle.dump(RandomForest, open(filename, 'wb'))
```

GitHub & project Demo-links:

GitHub: https://github.com/Vadigi789/Liver-patient-identification

Demo link:
https://drive.google.com/file/d/1MDC3lz__SNDQgHXNGP3OZsDC45xGUOZ1/view?usp=drive_link