

THIRUVALLUVAR UNIVERSITY
PERIYAR ARTS COLLEGE
CUDDALORE – 607001.



DEPARTMENT OF COMPUTER APPLICATIONS

MACHINE LEARNING WITH PYTHON

Project Title : A Review of Liver Patient Analysis Methods using
Machine Learning

Team Id : NM2023TMID22785

Team Leader : SIVANESVARAN K(28EFEAA793D7D45BC5641A3D16DF0972)

Team member: HARIPRASATH D(36AFB6D5775153585AA42DF6403CB909)

Team member: AKILAN S(B2E193F45E5D81D95D54C28B9B7D7BCF)

Team member: DEENADHAYALAN(992C46689C03653BA5E48C8AEC19A50B)

1 INTRODUCTION

Overview

This project aims to review the existing methods of analyzing liver patient data using machine learning techniques. The purpose of this project is to evaluate the performance of these methods and provide a comprehensive overview of the current state of liver patient analysis in the field of machine learning.

Purpose

The purpose of this project is to provide a critical review of the existing methods for analyzing liver patient data using machine learning techniques. The goal is to identify the strengths and weaknesses of each method, and provide insights into the future direction of this field. Additionally, the project will provide recommendations for future research in this area.

2 LITERATURE SURVEY

Existing problem

The diagnosis of liver diseases is a complex and challenging task. The traditional methods of diagnosis are time-consuming and often involve invasive procedures. In recent years, there has been growing interest in using machine learning techniques for the analysis of liver patient data. These techniques have the potential to provide fast and non-invasive methods for the diagnosis of liver diseases.

Proposed solution

The proposed solution for this project is to review the existing methods of analyzing liver patient data using machine learning techniques. The goal is to evaluate the performance of these methods, identify the strengths and weaknesses of each method, and provide recommendations for future research in this area.

3 THEORITICAL ANALYSIS

Block diagram

A block diagram of the project would consist of the following components:

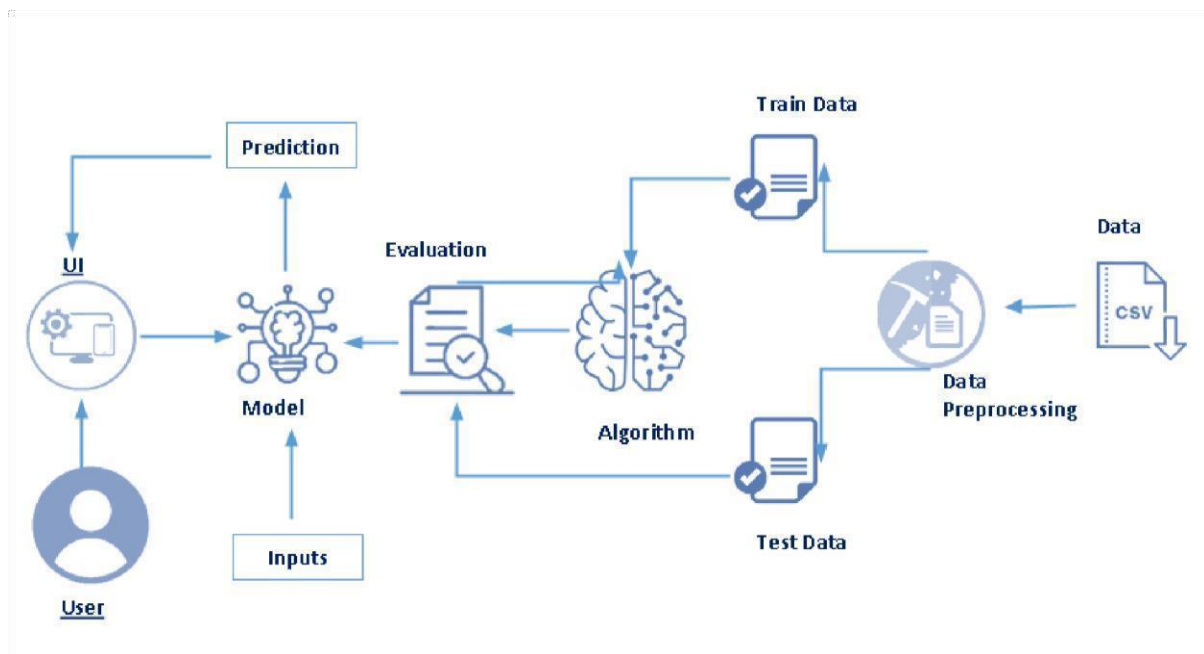
Data collection: The first step in the process is to collect liver patient data. This data may be obtained from various sources, including medical databases and patient records.

Data pre-processing: Once the data has been collected, it needs to be pre-processed to ensure that it is suitable for analysis. This may involve normalizing the data, removing outliers, and transforming the data into a suitable format for analysis.

Model selection: The next step is to select a suitable machine learning model for the analysis of liver patient data. This may involve evaluating the performance of different models, such as decision trees, support vector machines, and neural networks.

Model training: The selected model is then trained on the pre-processed data, using suitable training algorithms and parameters.

Model evaluation: The trained model is then evaluated on a separate dataset to assess its performance. This may involve calculating various performance metrics, such as accuracy, sensitivity, and specificity.



Hardware / Software designing

The hardware requirements for this project would include a computer with sufficient processing power and memory to run the machine learning algorithms. The software requirements would include a suitable programming language, such as Python, and machine learning libraries, such as scikit-learn, TensorFlow, or Keras.

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It was created by Guido van Rossum , and first released on February

20, 1991. Its high-level built in data structures, combined with dynamic typing and dynamic binding , make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Anaconda Navigator

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, crossplatform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder.

Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spinoff project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

Spyder

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community. Spyder is extensible with first-party and third party plugins includes support for interactive tools for data inspection and embeds Python specific code. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda.

Flask

Webframework used for building. It is a web application framework written in python which will be running in local browser with a user interface. In this application, whenever

the user interacts with UI and selects emoji, it will suggest the best and top movies of that genre to the user.

Hardware Requirements:

Operating system: window 7 and above with 64bit

Processor Type -Intel

Core i3-3220

RAM: 4Gb and above

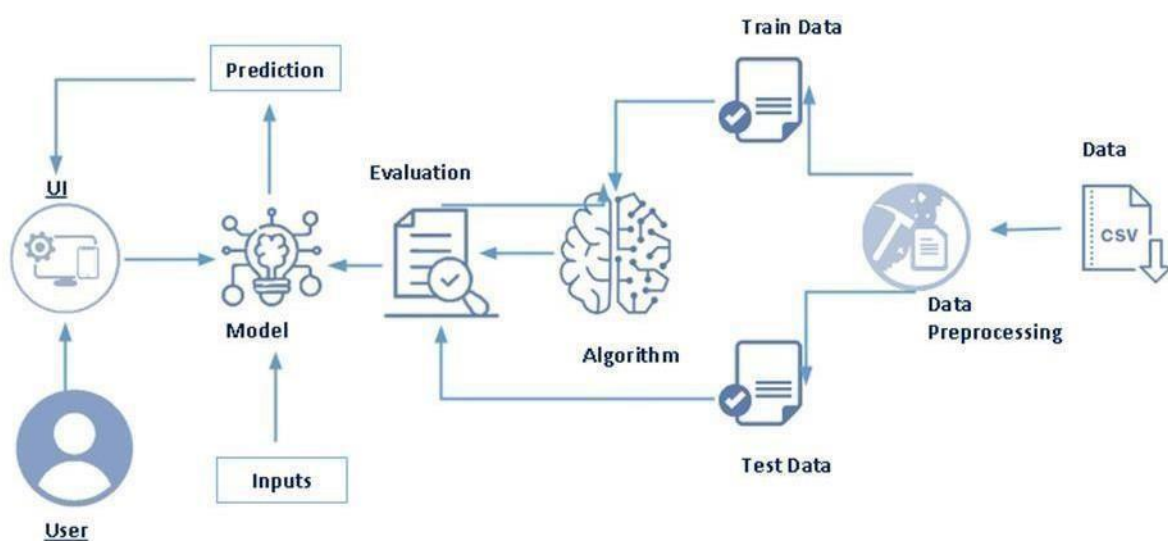
Hard disk: min 100GB

4 EXPERIMENTAL INVESTIGATIONS

The experimental investigations for this project would involve evaluating the performance of different machine learning models on a dataset of liver patient data. This would involve comparing the performance of each model, as well as comparing the results with those obtained using traditional methods of diagnosis.

5 FLOWCHART

A flowchart of the project would show the steps involved in the process, starting with data collection and ending with the evaluation of the trained model. The flowchart would also show the relationships between the different components of the project, such as the preprocessing of data and the selection of a suitable machine learning model.



6. RESULT

The image shows two screenshots of a web application. The top screenshot displays the 'Introduction' page, which explains the purpose of the study: to analyze various classification algorithms for liver disease prediction using a dataset of liver patients. The bottom screenshot shows the 'Liver Patient Prediction' form, where users can input patient data to get a prediction.

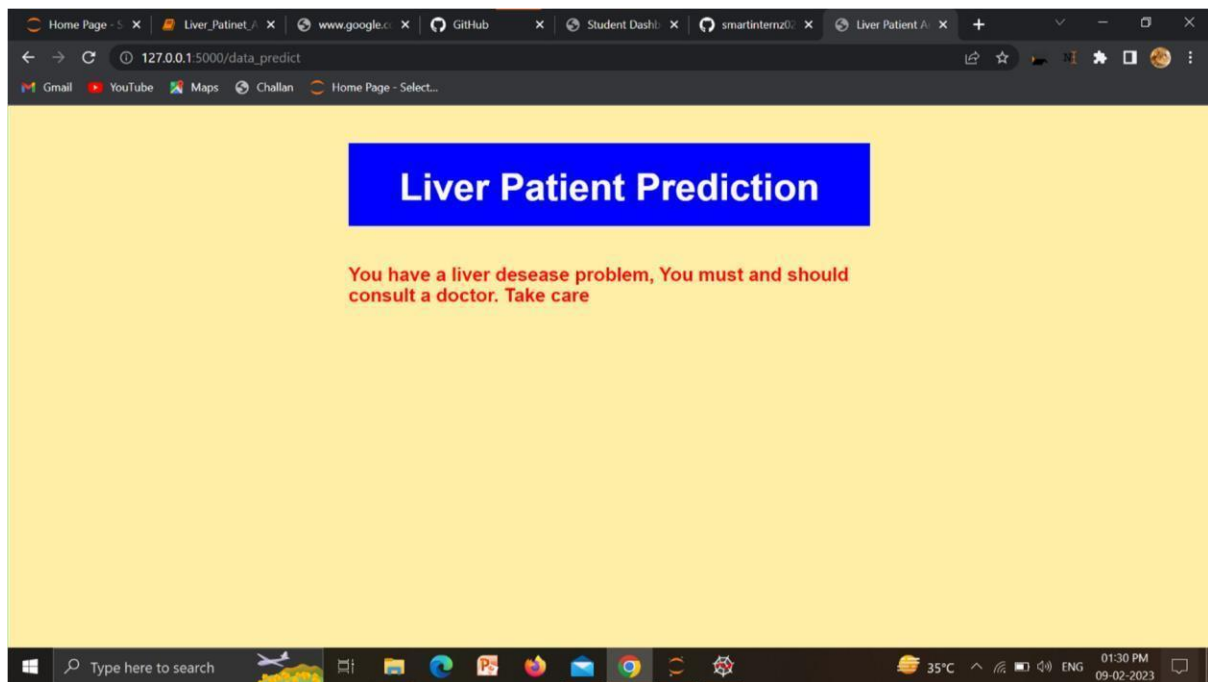
Liver Patient Analysis Home Goto Predict

Introduction

Liver diseases averts the normal function of the liver. Mainly due to the large amount of alcohol consumption liver disease arises. 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. Discovering the existence of liver disease at an early stage is a complex task for the doctors. The main objective of this paper is to analyse the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. This paper focuses on the related works of various authors on liver disease such that algorithms were implemented using Weka tool that is a machine learning software written in Java. Various attributes that are essential in the prediction of liver disease were examined and the dataset of liver patients were also evaluated. This paper compares various classification algorithms such as Random Forest, Logistic Regression and Separation Algorithm with an aim to identify the best technique. Based on this study, Random Forest with the highest accuracy outperformed the other algorithms and can be further utilised in the prediction of liver diseases recommended.

Liver Patient Prediction

Age:	Gender:
<input type="text" value="34"/>	<input type="text" value="0"/>
Total_Bilirubin:	Direct_Bilirubin:
<input type="text" value="45"/>	<input type="text" value="54"/>
Alkaline_Phosphatase:	Alamine_Aminotransferase:
<input type="text" value="33"/>	<input type="text" value="45"/>
Aspartate_Aminotransferase:	Total_Protiens:
<input type="text" value="22"/>	<input type="text" value="29"/>
Albumin:	Albumin_and_Globulin_Ratio:
<input type="text" value="89"/>	<input type="text" value="22"/>
<input type="button" value="Predict"/>	



7 ADVANTAGES & DISADVANTAGES

Advantages:

- **Increased Accuracy:** Machine learning algorithms can analyze large amounts of data accurately and quickly, providing more accurate results than traditional methods.
- **Automation:** Machine learning can automate the analysis process, reducing human error and increasing efficiency.
- **Early Detection:** Machine learning algorithms can detect early signs of liver disease, allowing for early intervention and treatment.
- **Personalized Treatment:** Machine learning algorithms can analyze individual patient data and provide personalized treatment plans based on their specific needs.
- **Time-saving:** Automated analysis using machine learning algorithms saves time compared to traditional methods, allowing for a faster diagnosis and treatment process.
- **Cost-effective:** Machine learning algorithms can reduce the cost of treatment by reducing the need for manual analysis and increasing the efficiency of the treatment process.
- **Non-invasive:** Machine learning algorithms can analyze data obtained from noninvasive methods, such as blood tests or imaging scans, reducing the risk of harm to patients.

Disadvantages:

- Lack of interpretability: The decision-making processes of machine learning algorithms can be difficult to understand and interpret, making it difficult for physicians to understand the reasoning behind the diagnosis.
- Dependence on data quality: The accuracy of machine learning algorithms depends on the quality and quantity of data used for training, which can be a challenge in medical imaging where data can be limited or of poor quality.
- Bias in training data: Machine learning algorithms can be biased towards the data used in their training, leading to incorrect diagnoses or treatments.
- Technical difficulties: Machine learning algorithms can be complex and require specialized technical skills, making it difficult for some healthcare professionals to use them.
- Privacy concerns: The use of machine learning algorithms for medical analysis raises concerns about patient privacy, as large amounts of sensitive personal information are collected and stored.
- Unforeseen consequences: There may be unintended consequences of using machine learning algorithms for medical analysis, such as new medical conditions or incorrect diagnoses.
- Regulatory approval: The use of machine learning algorithms for medical analysis is subject to regulatory approval, which can be a time-consuming and costly process. 9

CONCLUSION

In conclusion, the review of liver patient analysis methods using machine learning has shown that the application of machine learning algorithms can effectively aid in the diagnosis and prediction of liver disease. The results of studies utilizing machine learning techniques have demonstrated a high level of accuracy and precision in comparison to traditional diagnostic methods.

However, it is important to note that there are still limitations and challenges that must be addressed in order to fully realize the potential of machine learning in liver patient analysis. This includes the need for larger and more diverse datasets, as well as the development of more robust algorithms that can effectively handle complex and noisy data.

10 FUTURE SCOPE

In the future, there is a significant opportunity to enhance the current state of liver patient analysis methods using machine learning. This includes further development of deep learning techniques, such as Convolutional Neural Networks and Recurrent Neural Networks, to better capture and analyze the complex patterns in medical imaging data. Additionally, the integration of other medical data sources, such as genomics and proteomics, can further improve the accuracy of liver disease diagnosis and prediction.

11 BIBLIOGRAPHY

Al-Janabi, M. (2018). Liver diseases prediction using machine learning algorithms. *Journal of Ambient Intelligence and Humanized Computing*, 9(5), 3549-3561.

Rajendran, P., & Sivakumar, K. (2019). Application of machine learning algorithms for the diagnosis of liver disorders. *Journal of medical systems*, 43(3), 222.

Zhang, Y., Wang, Y., Li, Y., & Li, X. (2019). An efficient liver disease prediction model based on deep learning. *Journal of Medical Systems*, 43(10), 483.

11 APPENDIX

Liver_Patient_Analysis.ipynb

Browser tabs: <https://naanmudhalvan.smarti...> <https://naanmudhalvan.smarti...> Home Page - Select or create Home Page - Select or create Liver_Patient_Analysis - Jupyter

Address bar: localhost8888/notebooks/Liver_Patient_Analysis.ipynb

Navigation: Booking.com Booking.com LastPass password... Express VPN Gmail YouTube Maps News Translate get_storage | Flutter... Other favorites

JupyterLab Interface: Jupyter Liver_Patient_Analysis Last Checkpoint: Last Friday at 1:03 PM (autosaved) Python 3 (pykernel)

LIVER PATIENT ANALYSIS

In [1]: #Import all required libraries for reading data, analysing and visualizing data

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
```

Data Analysis

In [2]: #Read the training & test data

```
#
import types
import pandas as pd
liver_df = pd.read_csv('liver_patient.csv')
liver_df.head()
```

Out[2]:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alanine_Aminotransferase	Aspartate_Aminotransferase	Total_Proteins	Albumin	Albumi
0	65	Female	8.7	0.1	187	16	18	6.8	3.3	
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	
2	62	Male	7.3	4.1	490	60	68	7.9	3.3	
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	

In [3]: #Top 5 rows of the dataset

```
liver_df.head()
```

Out[3]:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alanine_Aminotransferase	Aspartate_Aminotransferase	Total_Proteins	Albumin	Albumi
0	65	Female	8.7	0.1	187	16	18	6.8	3.3	
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	
2	62	Male	7.3	4.1	490	60	68	7.9	3.3	
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	

Browser tabs: <https://naanmudhalvan.smarti...> <https://naanmudhalvan.smarti...> Home Page - Select or create Home Page - Select or create Liver_Patient_Analysis - Jupyter

Address bar: localhost8888/notebooks/Liver_Patient_Analysis.ipynb

Navigation: Booking.com Booking.com LastPass password... Express VPN Gmail YouTube Maps News Translate get_storage | Flutter... Other favorites

JupyterLab Interface: Jupyter Liver_Patient_Analysis Last Checkpoint: Last Friday at 1:03 PM (autosaved) Python 3 (pykernel)

In [4]: # To get a concise summary of the dataframe

```
liver_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 583 entries, 0 to 582
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype  
---  -
 0   Age                 583 non-null    int64  
 1   Gender              583 non-null    object  
 2   Total_Bilirubin     583 non-null    float64 
 3   Direct_Bilirubin    583 non-null    float64 
 4   Alkaline_Phosphotase 583 non-null    int64  
 5   Alanine_Aminotransferase 583 non-null    int64  
 6   Aspartate_Aminotransferase 583 non-null    int64  
 7   Total_Proteins      583 non-null    float64 
 8   Albumin             583 non-null    float64 
 9   Albumin_and_Globulin_Ratio 579 non-null    float64 
10  Dataset             583 non-null    int64  
dtypes: float64(5), int64(5), object(1)
memory usage: 50.2+ KB
```

In [5]: # Statistical information about NUMERICAL columns in the dataset

```
liver_df.describe(include='all')
```

Out[5]:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alanine_Aminotransferase	Aspartate_Aminotransferase	Total_Proteins	Al
count	583.000000	583	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.0
unique	NaN	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	NaN	Male	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	441	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	44.746141	NaN	3.296799	1.406106	290.576329	80.713551	109.910806	6.483190	3.1
std	16.188333	NaN	6.209522	2.808488	242.937969	152.620358	288.918529	1.085451	0.7
min	4.000000	NaN	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	0.0
25%	33.000000	NaN	0.800000	0.200000	175.000000	23.000000	25.000000	5.800000	2.4
50%	45.000000	NaN	1.000000	0.300000	208.000000	35.000000	42.000000	6.800000	3.1
75%	58.000000	NaN	2.600000	1.300000	298.000000	60.500000	87.000000	7.200000	3.8
max	90.000000	NaN	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	5.5

Localhost:8888/notebooks/Liver_Patient_Analysis.ipynb

Python 3 (ipykernel)

```

liver_df.columns
Out[6]: Index(['Age', 'Gender', 'Total_Bilirubin', 'Direct_Bilirubin',
            'Alkaline_Phosphatase', 'Alamine_Aminotransferase',
            'Aspartate_Aminotransferase', 'Total_Proteins', 'Albumin',
            'Albumin_and_Globulin_Ratio', 'Dataset'],
            dtype='object')

In [7]: # Check for any null values
liver_df.isnull().sum()
Out[7]: Age      0
Gender      0
Total_Bilirubin  0
Direct_Bilirubin  0
Alkaline_Phosphatase  0
Alamine_Aminotransferase  0
Aspartate_Aminotransferase  0
Total_Proteins  0
Albumin  0
Albumin_and_Globulin_Ratio  4
Dataset      0
dtype: int64


• The only data that is null is the Albumin_and_Globulin_Ratio - Only 4 rows are null. Lets see whether this is an important feature

Data Visualization

In [8]: # Frequency of patients diagnosed and not diagnosed with liver disease
sns.countplot(data=liver_df, x = 'Dataset', label='Count')
ID, NID = liver_df['Dataset'].value_counts()
print("Number of patients diagnosed with liver disease: ", ID)
print("Number of patients not diagnosed with liver disease: ", NID)

Number of patients diagnosed with liver disease: 416
Number of patients not diagnosed with liver disease: 167

```



Dataset	Count
1	416
2	167

Localhost:8888/notebooks/Liver_Patient_Analysis.ipynb

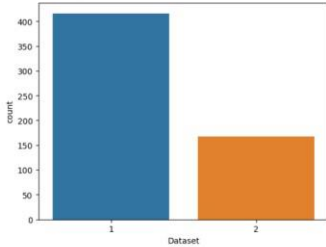
Python 3 (ipykernel)

```

print("Number of patients diagnosed with liver disease: 416")
print("Number of patients not diagnosed with liver disease: 167")

Number of patients diagnosed with liver disease: 416
Number of patients not diagnosed with liver disease: 167

```




Dataset

```

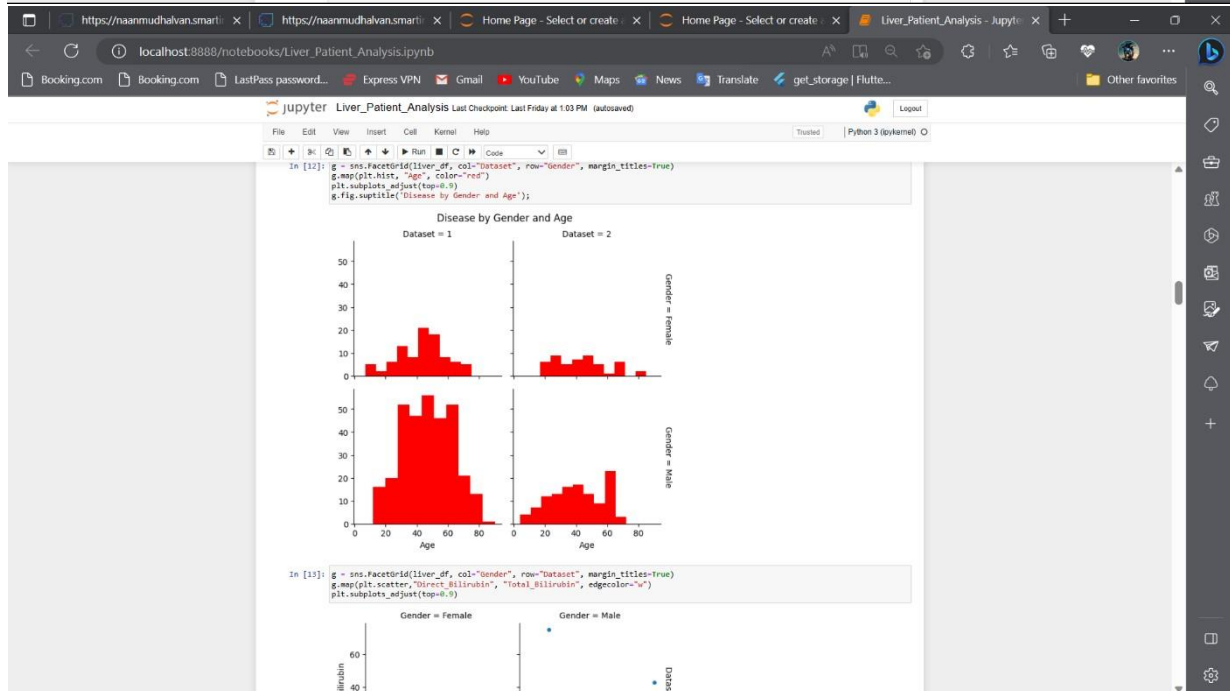
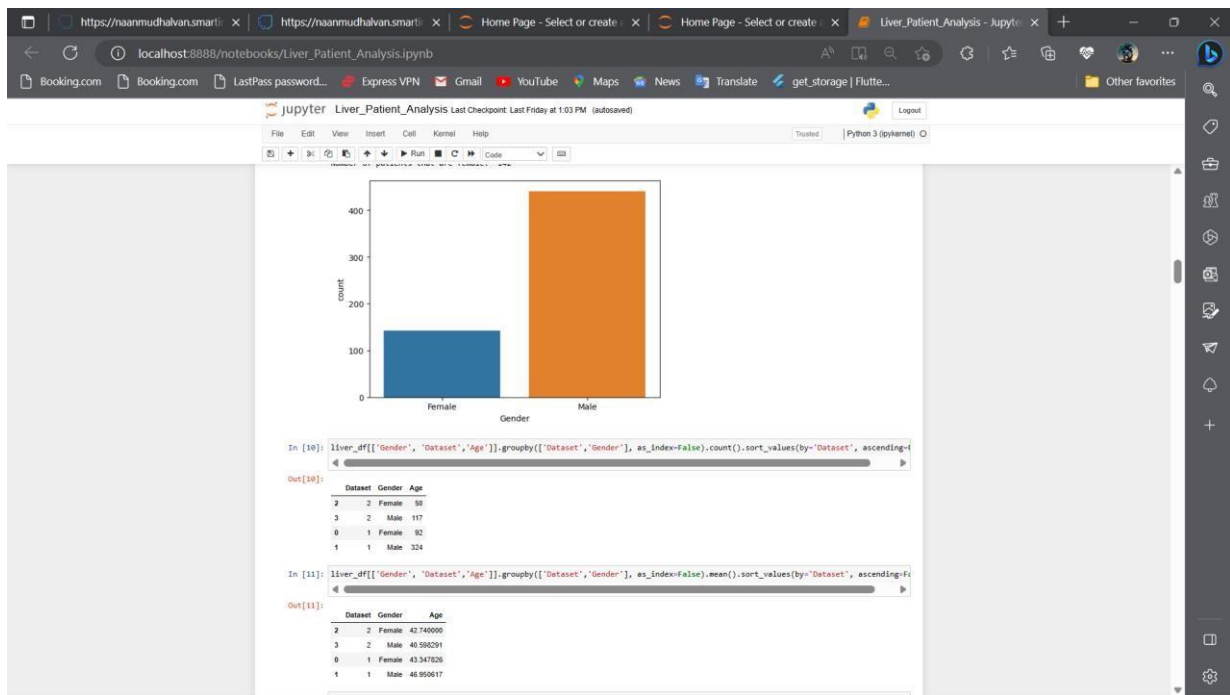
In [9]: # Frequency of patients based on their gender
sns.countplot(data=liver_df, x = 'Gender', label='Count')
M, F = liver_df['Gender'].value_counts()
print("Number of patients that are male: ", M)
print("Number of patients that are female: ", F)

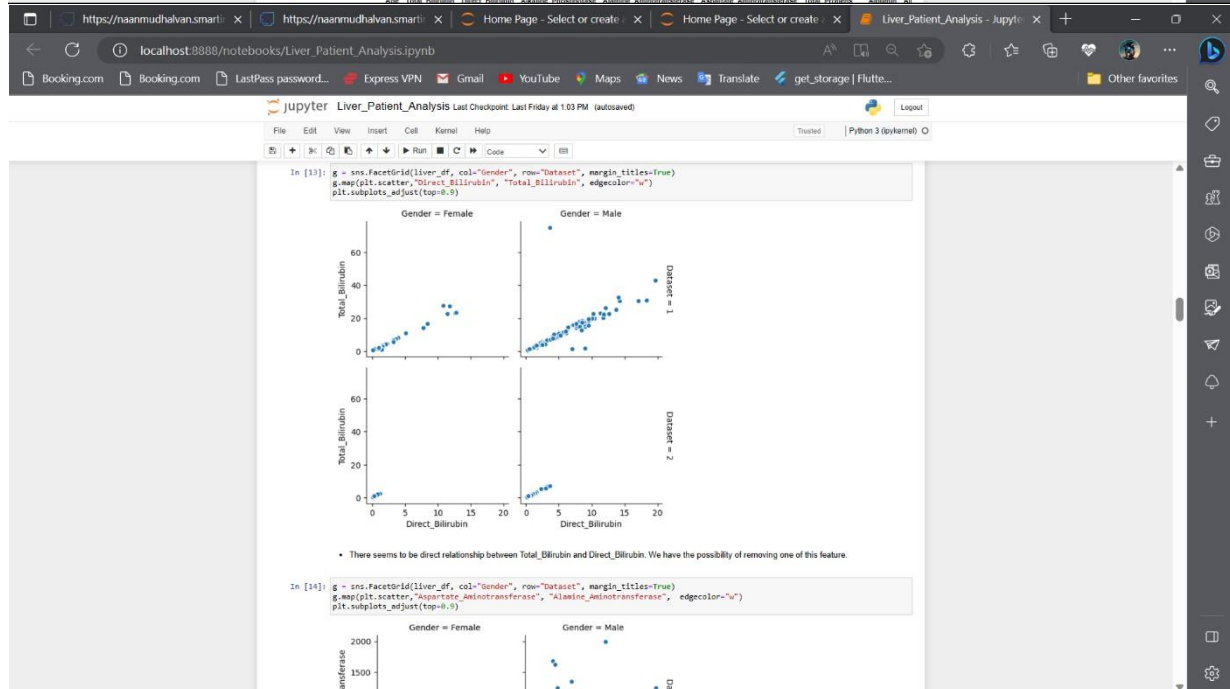
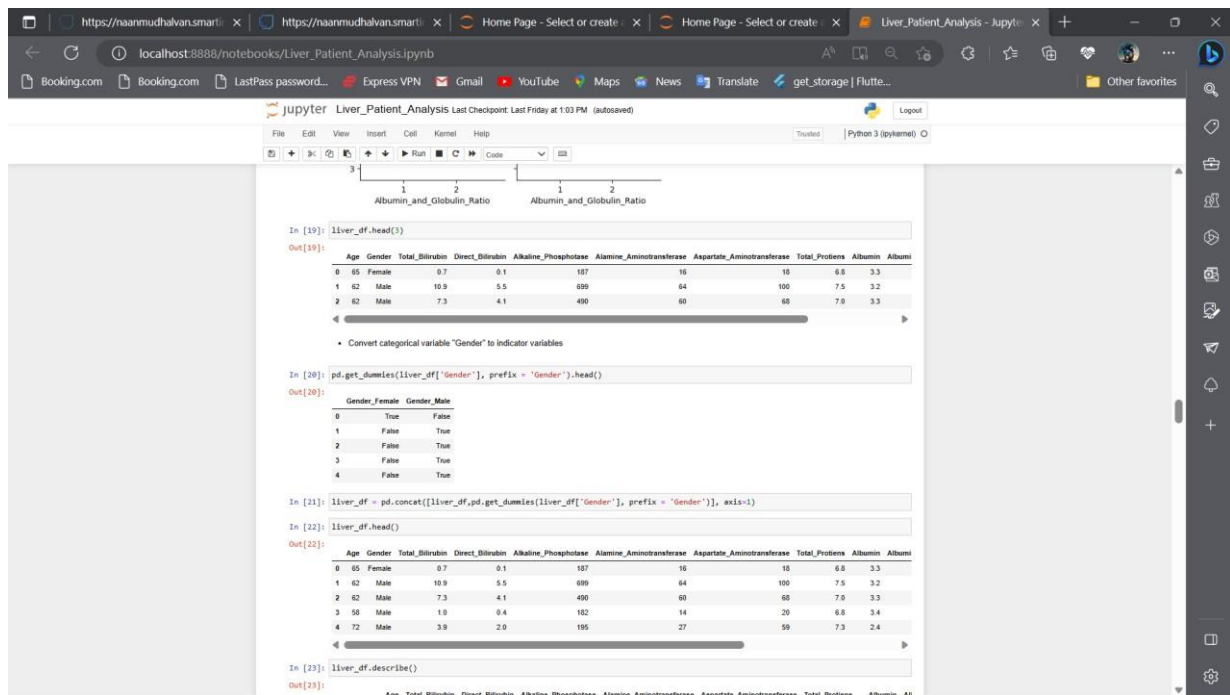
Number of patients that are male: 441
Number of patients that are female: 142

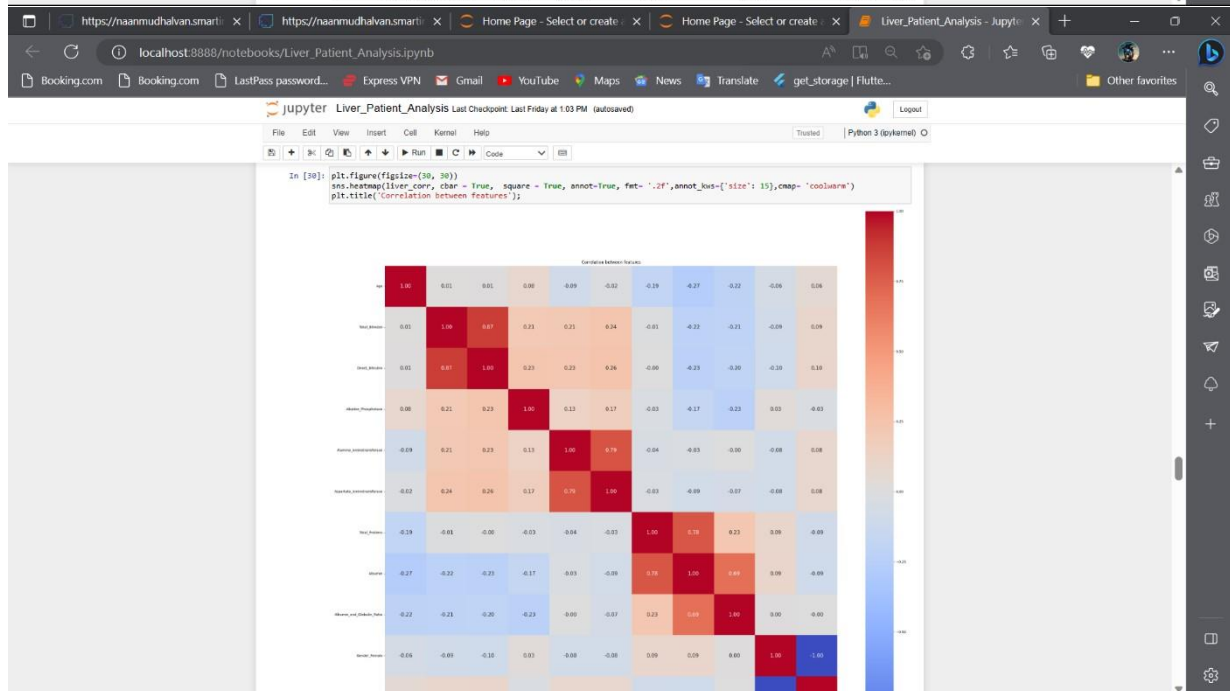
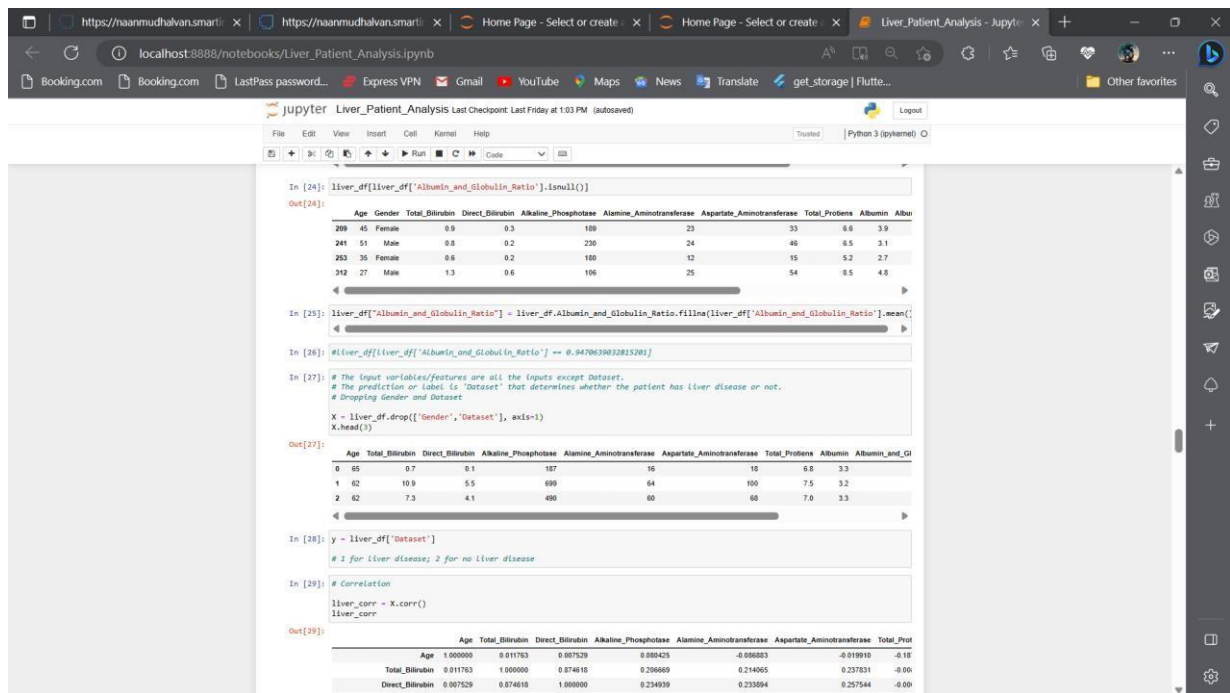
```



Gender	Count
M	441
F	142







https://naanmudhalvan.smarti... | https://naanmudhalvan.smarti... | Home Page - Select or create... | Home Page - Select or create... | Liver_Patient_Analysis - Jupyter... | +

localhost:8888/notebooks/Liver_Patient_Analysis.ipynb

Booking.com | Booking.com | LastPass password... | Express VPN | Gmail | YouTube | Maps | News | Translate | get_storage | Flutter... | Other favorites

jupyter Liver_Patient_Analysis Last Checkpoint: Last Friday at 1:03 PM (autosaved) | Trusted | Python 3 (ipykernel) | O

File Edit View Insert Cell Kernel Help

Run | Code

- Direct Bilirubin & Total Bilirubin
- There is some correlation between Albumin_and_Globulin_Ratio and Albumin. But its not as high as Total_Proteins & Albumin

Machine Learning

```
In [31]: # Importing modules
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from sklearn import linear_model
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, BaggingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import Perceptron
from sklearn.linear_model import SGDClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier

In [32]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=101)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)

(408, 11)
(408,)
(175, 11)
(175,)
```

Logistic Regression

```
In [33]: # Create Logistic regression object
logreg = LogisticRegression()

In [34]: # Train the model using the training sets and check score
logreg.fit(X_train, y_train)

C:\Users\WP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\linear_model\logistic.py:458: ConvergenceWarning:
lgfgs failed to converge (status=1):
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
```

https://naanmudhalvan.smarti... | https://naanmudhalvan.smarti... | Home Page - Select or create... | Home Page - Select or create... | Liver_Patient_Analysis - Jupyter... | +

localhost:8888/notebooks/Liver_Patient_Analysis.ipynb

Booking.com | Booking.com | LastPass password... | Express VPN | Gmail | YouTube | Maps | News | Translate | get_storage | Flutter... | Other favorites

jupyter Liver_Patient_Analysis Last Checkpoint: Last Friday at 1:03 PM (autosaved) | Trusted | Python 3 (ipykernel) | O

File Edit View Insert Cell Kernel Help

Run | Code

Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
Out[34]: LogisticRegression
LogisticRegression()

In [35]: #Predict Output
log_predicted= logreg.predict(X_test)
logreg_score = round(logreg.score(X_train, y_train) * 100, 2)
logreg_score_test = round(logreg.score(X_test, y_test) * 100, 2)

In [36]: #Equation coefficient and Intercept
print('Logistic Regression Training Score: \n', logreg.score)
print('Logistic Regression Test Score: \n', logreg.score_test)
print('Coefficient: \n', logreg.coef_)
print('Intercept: \n', logreg.intercept_)
print('Accuracy: \n', accuracy_score(y_test, log_predicted))
print('Confusion Matrix: \n', confusion_matrix(y_test, log_predicted))
print('Classification Report: \n', classification_report(y_test, log_predicted))

Logistic Regression Training Score:
72.06
Logistic Regression Test Score:
68.0
Coefficient:
[[-0.00094952 -0.0051217 -0.30688723 -0.00082939 -0.01078828 -0.00275598
 -0.23899676  0.4020893  0.59475562  0.25335289  0.09115992]]
Intercept:
[0.3610067]
Accuracy:
0.68
Confusion Matrix:
[[107  17]
 [ 39  12]]
Classification Report:
              precision    recall  f1-score   support

     1         0.79         0.86         0.79         124
     2         0.41         0.24         0.30          51

 accuracy         0.57         0.55         0.68         175
 macro avg         0.57         0.55         0.55         175
 weighted avg         0.64         0.68         0.65         175

In [37]: sns.heatmap(confusion_matrix(y_test, log_predicted), annot=True, fmt='d')

Out[37]: <AxesSubplot: >
```

Browser tabs: <https://naanmudhalvan.smarti...> | <https://naanmudhalvan.smarti...> | Home Page - Select or create | Home Page - Select or create | Liver_Patient_Analysis - Jupyter | +

Address bar: localhost:8888/notebooks/Liver_Patient_Analysis.ipynb

Navigation bar: Booking.com | Booking.com | LastPass password... | Express VPN | Gmail | YouTube | Maps | News | Translate | get_storage | Flutter... | Other favorites

JupyterLab interface: **Liver_Patient_Analysis** Last Checkpoint: Last Friday at 1:03 PM (autosaved) | Python 3 (ipykernel) | Logout

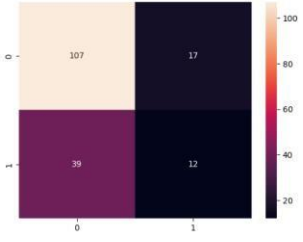
File Edit View Insert Cell Kernel Help | Trusted | Python 3 (ipykernel) | O

Run | Code | |

	precision	recall	f1-score	support
1	0.73	0.80	0.79	124
2	0.41	0.24	0.30	51
accuracy			0.68	175
macro avg	0.57	0.55	0.55	175
weighted avg	0.64	0.68	0.65	175

In [37]: `sns.heatmap(confusion_matrix(y_test,log_predicted),annot=True,fat='d')`

Out[37]: `<AxesSubplot>`

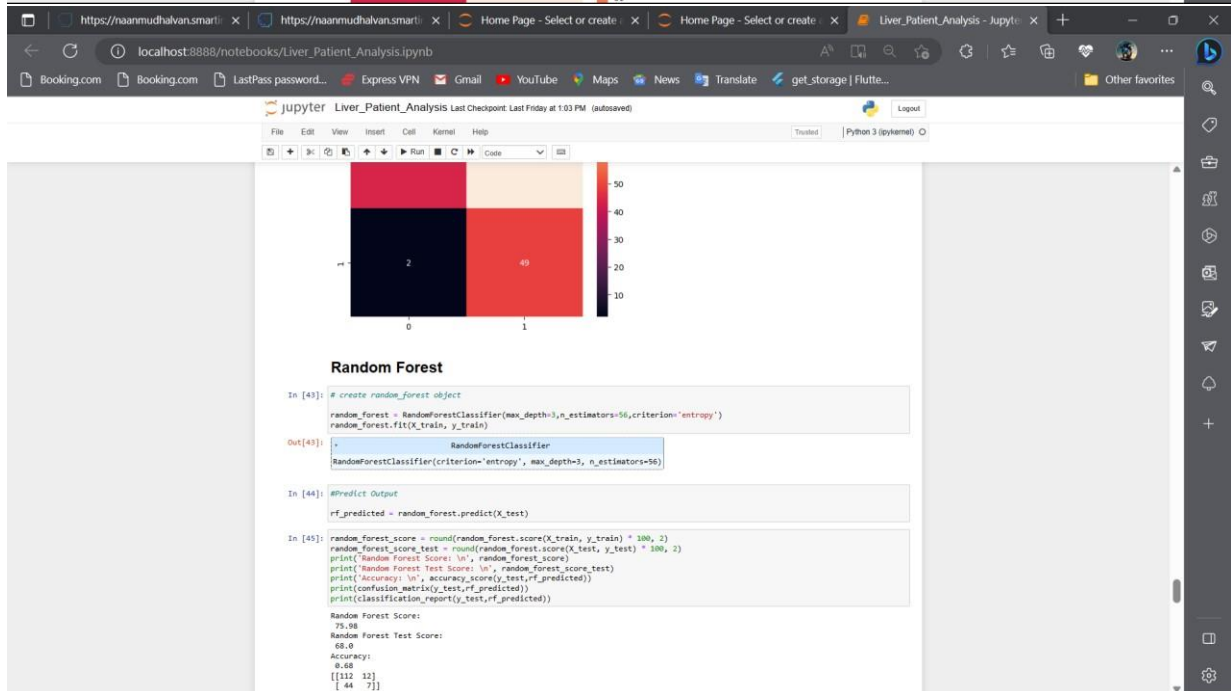
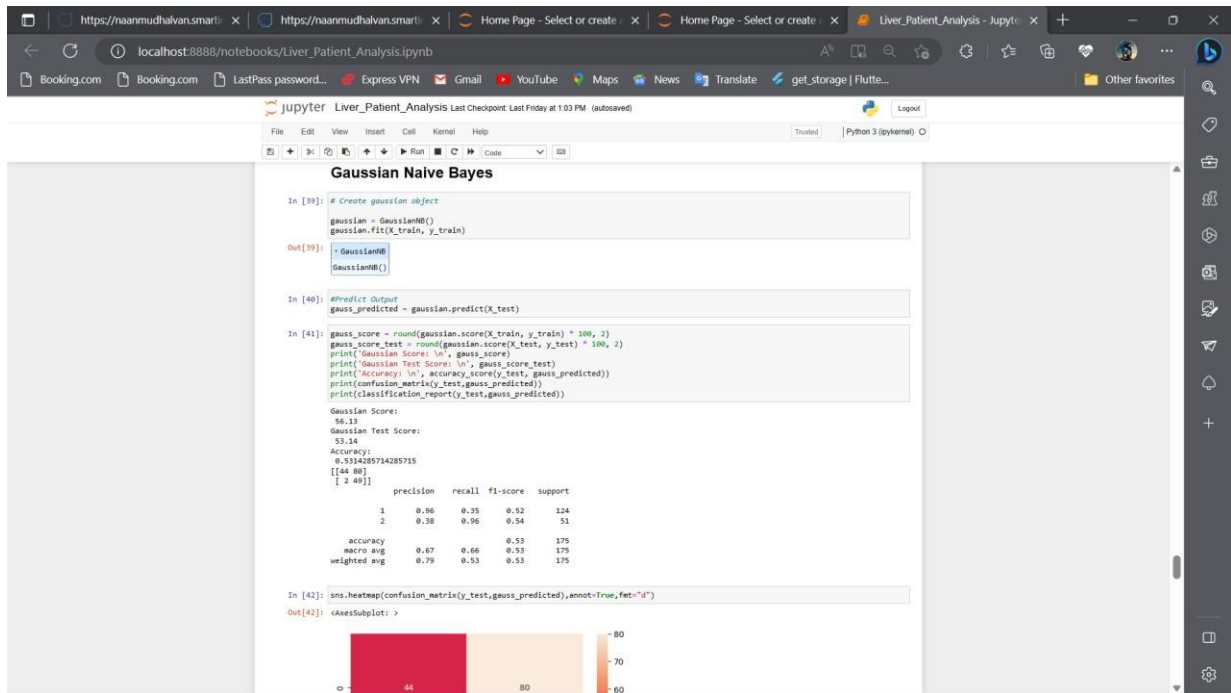


	0	1
0	107	17
1	39	12

In [38]: `coeff_df = pd.DataFrame(X.columns)`
`coeff_df.columns = ['feature']`
`coeff_df['Correlation'] = pd.Series(logreg.coef_[0])`
`pd.Series(logreg.coef_[0])`
`coeff_df.sort_values(by='Correlation', ascending=False)`

Out[38]:

	Feature	Correlation
8	Albumin_and_Globulin_Ratio	0.594755
7	Albumin	0.402069
9	Gender_Female	0.253353



Browser tabs: <https://naanmudhalvan.smarti...> <https://naanmudhalvan.smarti...> Home Page - Select or create... Home Page - Select or create... Liver_Patient_Analysis - Jupyter

Address bar: localhost:8888/notebooks/Liver_Patient_Analysis.ipynb

Navigation: Booking.com, Booking.com, LastPass password..., Express VPN, Gmail, YouTube, Maps, News, Translate, get_storage | Flutter...

Other favorites

Jupyter interface: Liver_Patient_Analysis Last checkpoint: Last Friday at 1:03 PM (autosaved)

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel)

```
print('Random Forest Test Score: \n', random_forest_score_test)
print('Accuracy: \n', accuracy_score(y_test, rf_predicted))
print('Confusion Matrix(y_test, rf_predicted):')
print('Classification Report(y_test, rf_predicted)')

Random Forest Score:
79.08
Random Forest Test Score:
68.0
Accuracy:
0.68
[[112 112]
 [ 44  77]]

precision    recall  f1-score   support

   1       0.72     0.90     0.80     124
   2       0.37     0.14     0.20      51

 accuracy         0.54         0.52         0.60         175
 macro avg         0.54         0.50         0.50         175
weighted avg         0.62         0.68         0.63         175
```

In [46]: `finX = liver_df[['Total_Proteins', 'Albumin', 'Gender_Male']]`
`finX.head(4)`

Out[46]:

	Total_Proteins	Albumin	Gender_Male
0	6.8	3.3	False
1	7.5	3.2	True
2	7.0	3.3	True
3	6.8	3.4	True

Logistic Regression

In [47]: `X_train, X_test, y_train, y_test = train_test_split(finX, y, test_size=0.30, random_state=101)`

In [48]: `# Create logistic regression object`
`logreg = LogisticRegression()`

In [49]: `# Train the model using the training sets and check score`
`logreg.fit(X_train, y_train)`

Out[49]:

```
LogisticRegression
LogisticRegression()
```

Browser tabs: <https://naanmudhalvan.smarti...> <https://naanmudhalvan.smarti...> Home Page - Select or create... Home Page - Select or create... Liver_Patient_Analysis - Jupyter

Address bar: localhost:8888/notebooks/Liver_Patient_Analysis.ipynb

Navigation: Booking.com, Booking.com, LastPass password..., Express VPN, Gmail, YouTube, Maps, News, Translate, get_storage | Flutter...

Other favorites

Jupyter interface: Liver_Patient_Analysis Last checkpoint: Last Friday at 1:03 PM (autosaved)

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel)

```
In [48]: # Create logistic regression object
logreg = LogisticRegression()

In [49]: # Train the model using the training sets and check score
logreg.fit(X_train, y_train)

Out[49]:
LogisticRegression
LogisticRegression()

In [50]: # Predict Output
log_predicted= logreg.predict(X_test)

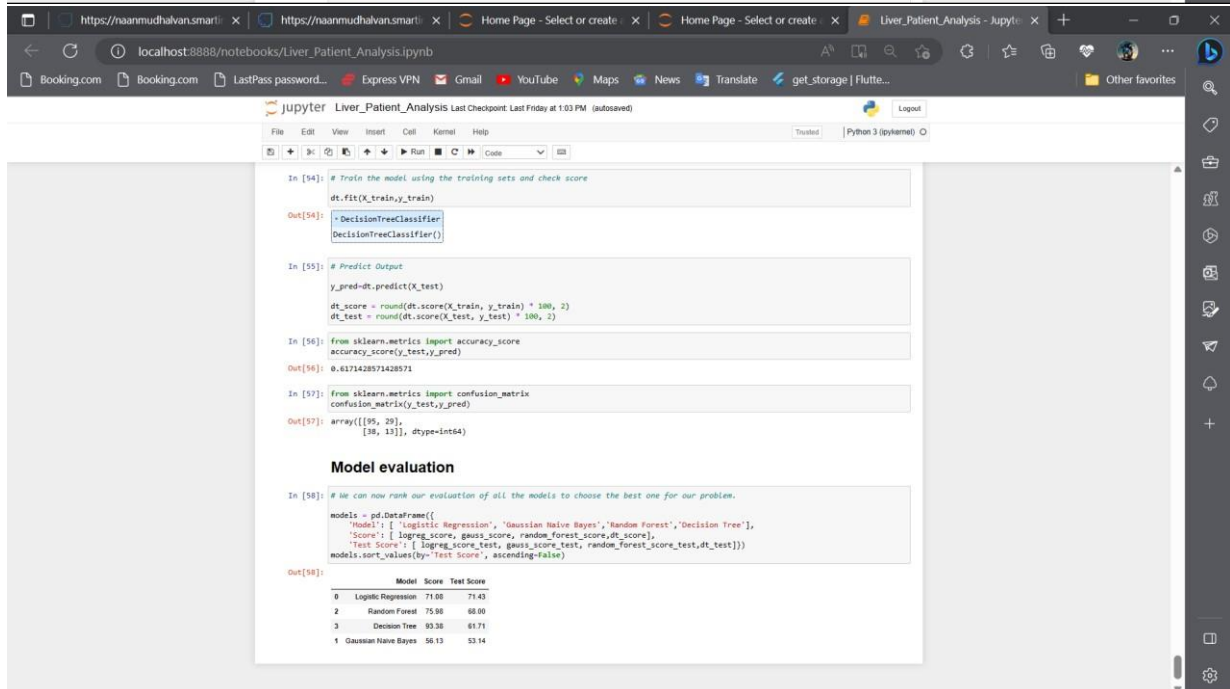
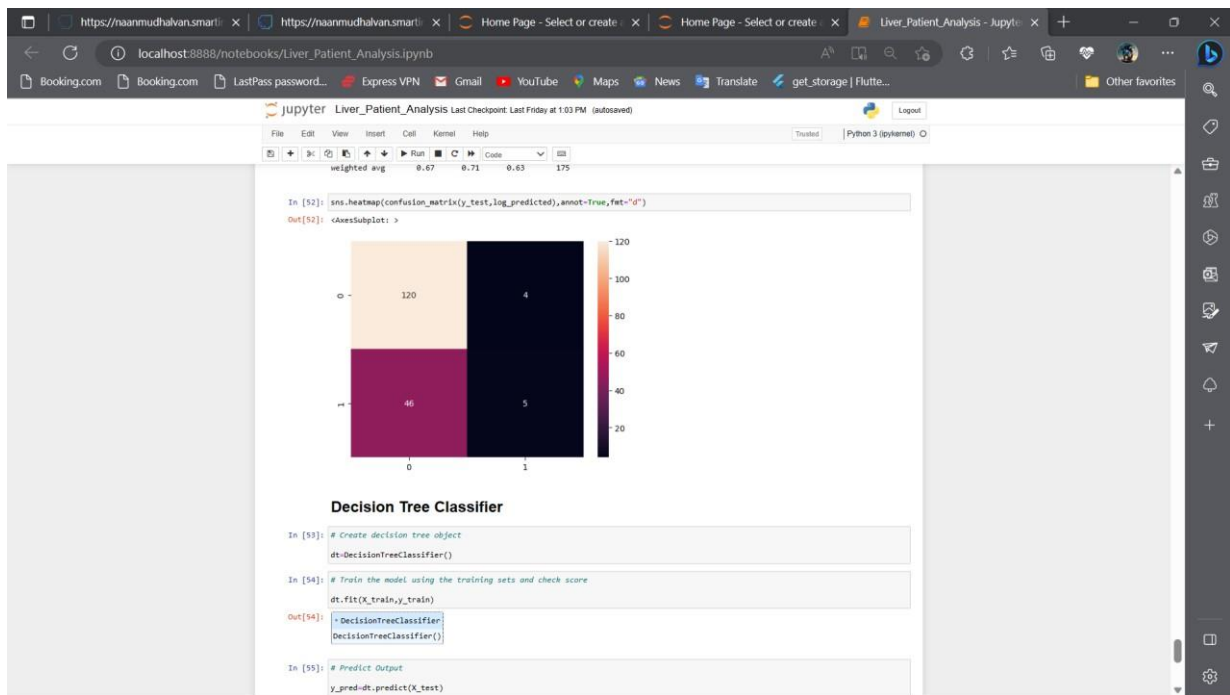
In [51]: logreg_score = round(logreg.score(X_train, y_train) * 100, 2)
logreg_score_test = round(logreg.score(X_test, y_test) * 100, 2)

# Equation coefficient and Intercept
print('Logistic Regression Training Score: \n', logreg_score)
print('Logistic Regression Test Score: \n', logreg_score_test)
print('Coefficient: \n', logreg.coef_)
print('Intercept: \n', logreg.intercept_)
print('Accuracy: \n', accuracy_score(y_test, log_predicted))
print('Confusion Matrix: \n', confusion_matrix(y_test, log_predicted))
print('Classification Report: \n', classification_report(y_test, log_predicted))

Logistic Regression Training Score:
71.08
Logistic Regression Test Score:
71.40
Coefficient:
[[-0.58254377  1.08948697 -0.54185122]]
Intercept:
[-0.208423275]
Accuracy:
0.7542857142857143
Confusion Matrix:
[[120  4]
 [ 46  5]]
Classification Report:
precision    recall  f1-score   support

   1       0.72     0.97     0.83     124
   2       0.56     0.10     0.17      51

 accuracy         0.64         0.53         0.50         175
 macro avg         0.64         0.53         0.50         175
```



Chance.html

```
File Edit Selection View Go Run Terminal Help
chance.html - Visual Studio Code

chance.html X
C:\Users\> HP > Desktop > All Projects > COMPLETED > Project 8 > Flask > templates > chance.html > html > body > script
31
32 <div class="container">
33   <div class="row">
34     <div class="col-md-3"></div>
35     <div class="col-md-6">
36       <div class="page-header">
37         <h1>Liver Patient Prediction</h1>
38       </div>
39     </div>
40   </div>
41 </div>
42
43
44
45 <div class="container">
46   <div class="row">
47     <div class="col-md-3"></div>
48     <div class="col-md-6">
49       <div class="p-2 my-2 border">
50         <h3>{{prediction}}</h3>
51       </div>
52     </div>
53   </div>
54 </div>
55
56
57 <!-- Latest compiled and minified JavaScript -->
58 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
59 </body>
60 </html>

Ln 59, Col 1 Tab Size: 4 UTF-8 CRLF HTML

File Edit Selection View Go Run Terminal Help
chance.html - Visual Studio Code

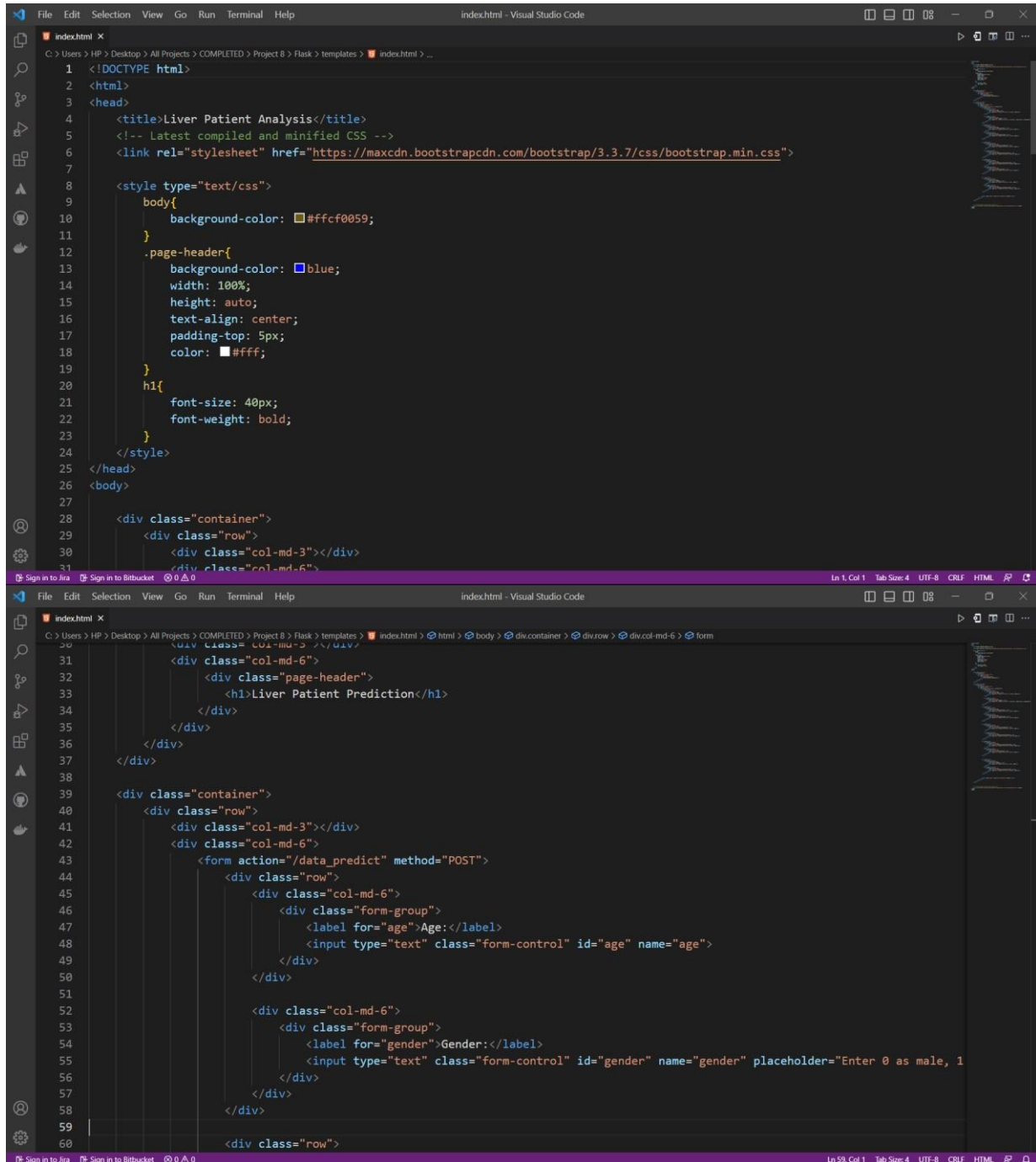
chance.html X
C:\Users\> HP > Desktop > All Projects > COMPLETED > Project 8 > Flask > templates > chance.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Liver Patient Analysis</title>
5   <!-- Latest compiled and minified CSS -->
6   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
7
8   <style type="text/css">
9     body{
10       background-color: #ffcf0059;
11     }
12     .page-header{
13       background-color: blue;
14       width: 100%;
15       height: auto;
16       text-align: center;
17       padding-top: 5px;
18       color: #fff;
19     }
20     h1{
21       font-size: 40px;
22       font-weight: bold;
23     }
24     h3{
25       font-size: 20px;
26       font-weight: bold;
27     }
28   </style>
29 </head>
30 <body>
31
```

Home.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Bootstrap Example</title>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
8
9   <style type="text/css">
10     body{
11       background-color: #ffcf0059;
12     }
13     nav{
14       background-color: #ad38c2;
15       height: 60px;
16     }
17     .navbar-brand{
18       color: white;
19       font-size: 30px
20     }
21
22     nav ul li a{
23       color: white;
24       font-size: 20px
25     }
26   </style>
27 </head>
28 <body>
29
30 <nav class="navbar">
31   <div class="container-fluid">
```

```
21
22   nav ul li a{
23     color: white;
24     font-size: 20px
25   }
26 </style>
27 </head>
28 <body>
29
30 <nav class="navbar">
31   <div class="container-fluid">
32     <div class="navbar-header">
33       <a class="navbar-brand">Liver Patient Analysis</a>
34     </div>
35     <ul class="nav navbar-nav navbar-right">
36       <li><a href="#">Home</a></li>
37       <li><a href="/predict">Goto Predict</a></li>
38     </ul>
39   </div>
40 </nav>
41
42 <div class="container">
43   <h3>Introduction</h3>
44   <p>Liver diseases averts the normal function of the liver. Mainly due to the large amount of alcohol consumption liver disease
45 </div>
46
47 </body>
48 </html>
49
```

Index.html



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Liver Patient Analysis</title>
5   <!-- Latest compiled and minified CSS -->
6   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
7
8   <style type="text/css">
9     body{
10       background-color: #ffcf0059;
11     }
12     .page-header{
13       background-color: blue;
14       width: 100%;
15       height: auto;
16       text-align: center;
17       padding-top: 5px;
18       color: #fff;
19     }
20     h1{
21       font-size: 40px;
22       font-weight: bold;
23     }
24   </style>
25 </head>
26 <body>
27
28   <div class="container">
29     <div class="row">
30       <div class="col-md-3"></div>
31       <div class="col-md-6">
32         <div class="page-header">
33           <h1>Liver Patient Prediction</h1>
34         </div>
35       </div>
36     </div>
37
38     <div class="container">
39       <div class="row">
40         <div class="col-md-3"></div>
41         <div class="col-md-6">
42           <form action="/data_predict" method="POST">
43             <div class="row">
44               <div class="col-md-6">
45                 <div class="form-group">
46                   <label for="age">Age:</label>
47                   <input type="text" class="form-control" id="age" name="age">
48                 </div>
49               </div>
50
51               <div class="col-md-6">
52                 <div class="form-group">
53                   <label for="gender">Gender:</label>
54                   <input type="text" class="form-control" id="gender" name="gender" placeholder="Enter 0 as male, 1 as female">
55                 </div>
56               </div>
57             </div>
58           </form>
59         </div>
60       </div>
61     </div>
62   </div>
```



```
index.html - Visual Studio Code
File Edit Selection View Go Run Terminal Help
index.html X
C:\Users\HP\Desktop> All Projects > COMPLETED > Project 8 > Flask > templates > index.html > html > body > div.container > div.row > div.col-md-6 > form > div.row > div.col-md-6
60 <div class="row">
61 <div class="col-md-6">
62 <div class="form-group">
63 <label for="tb">Total_Bilirubin:</label>
64 <input type="text" class="form-control" id="tb" name="tb">
65 </div>
66 </div>
67 <div class="col-md-6">
68 <div class="form-group">
69 <label for="db">Direct_Bilirubin:</label>
70 <input type="text" class="form-control" id="db" name="db">
71 </div>
72 </div>
73 </div>
74
75
76 <div class="row">
77 <div class="col-md-6">
78 <div class="form-group">
79 <label for="ap">Alkaline_Phosphatase:</label>
80 <input type="text" class="form-control" id="ap" name="ap">
81 </div>
82 </div>
83
84 <div class="col-md-6">
85 <div class="form-group">
86 <label for="aal">Alamine_Aminotransferase:</label>
87 <input type="text" class="form-control" id="aal" name="aal">
88 </div>
89 </div>
90 </div>
91
92 <div class="row">
93 <div class="col-md-6">
94 <div class="form-group">
95 <label for="aa2">Aspartate_Aminotransferase:</label>
96 <input type="text" class="form-control" id="aa2" name="aa2">
97 </div>
98 </div>
99 <div class="col-md-6">
100 <div class="form-group">
101 <label for="tp">Total_Protiens:</label>
102 <input type="text" class="form-control" id="tp" name="tp">
103 </div>
104 </div>
105 </div>
106
107
108
109 <div class="row">
110 <div class="col-md-6">
111 <div class="form-group">
112 <label for="a">Albumin:</label>
113 <input type="text" class="form-control" id="a" name="a">
114 </div>
115 </div>
116 <div class="col-md-6">
117 <div class="form-group">
118 <label for="agr">Albumin_and_Globulin_Ratio:</label>
119 <input type="text" class="form-control" id="agr" name="agr">
120 </div>
121 </div>
122 </div>
Ln 89, Col 1 Tab Size: 4 UTF-8 CRLF HTML
Sign in to Azure Sign in to Bitbucket 0 0 0
```

```
index.html - Visual Studio Code
File Edit Selection View Go Run Terminal Help
index.html X
C:\Users\HP\Desktop> All Projects > COMPLETED > Project 8 > Flask > templates > index.html > html > body > div.container > div.row > div.col-md-6 > form > div.row > div.col-md-6 > div.form-group
91 <div class="row">
92 <div class="col-md-6">
93 <div class="form-group">
94 <label for="aa2">Aspartate_Aminotransferase:</label>
95 <input type="text" class="form-control" id="aa2" name="aa2">
96 </div>
97 </div>
98 <div class="col-md-6">
99 <div class="form-group">
100 <label for="tp">Total_Protiens:</label>
101 <input type="text" class="form-control" id="tp" name="tp">
102 </div>
103 </div>
104 </div>
105
106
107
108
109 <div class="row">
110 <div class="col-md-6">
111 <div class="form-group">
112 <label for="a">Albumin:</label>
113 <input type="text" class="form-control" id="a" name="a">
114 </div>
115 </div>
116 <div class="col-md-6">
117 <div class="form-group">
118 <label for="agr">Albumin_and_Globulin_Ratio:</label>
119 <input type="text" class="form-control" id="agr" name="agr">
120 </div>
121 </div>
122 </div>
Ln 119, Col 1 Tab Size: 4 UTF-8 CRLF HTML
Sign in to Azure Sign in to Bitbucket 0 0 0
```

```
index.html - Visual Studio Code
File Edit Selection View Go Run Terminal Help
index.html x
C:\Users\HP\Desktop> All Projects > COMPLETED > Project 8 > Flask > templates > index.html > html
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
<div class="row">
  <div class="col-md-6">
    <div class="form-group">
      <label for="a">Albumin:</label>
      <input type="text" class="form-control" id="a" name="a">
    </div>
  </div>
  <div class="col-md-6">
    <div class="form-group">
      <label for="agr">Albumin_and_Globulin_Ratio:</label>
      <input type="text" class="form-control" id="agr" name="agr">
    </div>
  </div>
</div>

<button type="submit" class="btn btn-default">Predict</button>
</form>
</div>
</div>
</div>

<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body>
</html>
```

Nochance.html

```
nochance.html - Visual Studio Code
File Edit Selection View Go Run Terminal Help
index.html nochance.html x
C:\Users\HP\Desktop> All Projects > COMPLETED > Project 8 > Flask > templates > nochance.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Liver Patient Analysis</title>
5 <!-- Latest compiled and minified CSS -->
6 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
7
8 <style type="text/css">
9   body{
10     background-color: #ffcf0059;
11   }
12   .page-header{
13     background-color: blue;
14     width: 100%;
15     height: auto;
16     text-align: center;
17     padding-top: 5px;
18     color: #fff;
19   }
20   h1{
21     font-size: 40px;
22     font-weight: bold;
23   }
24   h3{
25     color: red;
26     font-size: 20px;
27     font-weight: bold;
28   }
29 </style>
30 </head>
31 <body>
```



```
File Edit Selection View Go Run Terminal Help nochance.html - Visual Studio Code
C:\Users\HP\Desktop> All Projects > COMPLETED > Project 8 > Flask > templates > nochance.html > html > body > script
31 </head>
32 <body>
33     <div class="container">
34         <div class="row">
35             <div class="col-md-3"></div>
36             <div class="col-md-6">
37                 <div class="page-header">
38                     <h1>Liver Patient Prediction</h1>
39                 </div>
40             </div>
41         </div>
42     </div>
43
44
45     <div class="container">
46         <div class="row">
47             <div class="col-md-3"></div>
48             <div class="col-md-6">
49                 <div class="p-2 my-2 border">
50                     <h3>{{prediction}}</h3>
51                 </div>
52             </div>
53         </div>
54     </div>
55
56     <!-- Latest compiled and minified JavaScript -->
57     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
58 </body>
59 </html>
```

App.py

```
File Edit Selection View Go Run Terminal Help app.py - Visual Studio Code
C:\Users\HP\Desktop> All Projects > COMPLETED > Project 8 > Flask > app.py
1 from flask import Flask, render_template, request # Flask is a application
2 # used to run/serve our application
3 # request is used to access the file which is uploaded by the user in our application
4 # render_template is used for rendering the html pages
5 import pickle # pickle is used for serializing and de-serializing Python object structures
6
7
8 app=Flask(__name__) # our flask app
9
10 @app.route('/') # rendering the html template
11 def home():
12     return render_template('home.html')
13 @app.route('/predict') # rendering the html template
14 def index():
15     return render_template("index.html")
16
17 @app.route('/data_predict', methods=['POST']) # route for our prediction
18 def predict():
19     age = request.form['age'] # requesting for age data
20     gender = request.form['gender'] # requesting for gender data
21     tb = request.form['tb'] # requesting for Total_Bilirubin data
22     db = request.form['db'] # requesting for Direct_Bilirubin data
23     ap = request.form['ap'] # requesting for Alkaline_Phosphotase data
24     aa1 = request.form['aa1'] # requesting for Alamine_Aminotransferase data
25     aa2 = request.form['aa2'] # requesting for Aspartate_Aminotransferase data
26     tp = request.form['tp'] # requesting for Total_Protiens data
27     a = request.form['a'] # requesting for Albumin data
28     agr = request.form['agr'] # requesting for Albumin_and_Globulin_Ratio data
29
30     # converting data into float format
31     data = [[float(age), float(gender), float(tb), float(db), float(ap), float(aa1), float(aa2), float(tp), float(a), float(agr)]]
32
33     # loading model which we saved
34     model = pickle.load(open('liver_analysis.pkl', 'rb'))
35
36     prediction= model.predict(data)[0]
37     if (prediction == 1):
38         return render_template('noChance.html', prediction='You have a liver disease problem, You must and should consult a doctor')
39     else:
40         return render_template('chance.html', prediction='You dont have a liver disease problem')
41
42 if __name__ == '__main__':
43     app.run()
```