Project Initialization and Planning Phase

Date	11th jun 2025
Team ID	LTVIP2025TMID38009
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques
Maximum Marks	3 Marks

Problem Statement Definition :

The project is focused on creating an advanced machine learning-based predictive model to identify the onset or progression of liver cirrhosis in patients. Liver cirrhosis, a severe condition marked by liver tissue scarring due to prolonged damage, requires early detection and intervention to improve patient outcomes and avoid complications. By examining diverse patient data, including medical history, lab results, imaging scans, and lifestyle factors, the model aims to predict the likelihood of liver cirrhosis. This will assist healthcare professionals in making well-informed decisions regarding patient care.

Initial Project Planning Template

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	Project Initialization And Planning	RLCPC-2, RLCPC-3	• Project Planning and Proposal • Identifying and Defining the Problem Statement.	10	High	1)Muppalla sri anjaneyulu	10.06.25	16.06.25
Sprint-2	Data Collection and Data Preprocessing	RLCPC-5 RLCPC-6 RLCPC-8 RLCPC-9 RLCPC-10 RLCPC-11	• Collection of Data • Loading and Understanding of Data • Handling Null Values • Handling Categorical Data Handling Outliers Handling Duplicate Values.	9	High	2)nallamothu manohar , 3) Paladugu delhi poleswarao,	16.06.25	19.06.25

Date	28th June 2025
Team ID	LTVIP2025TMID38009
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques
Maximum Marks	4 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-3	Exploratory Data Analysis and Model Building	RLCPC-13 RLCPC-14 RLCPC-15 RLCPC-16 RLCPC-18	• Univariate Analysis. • Bivariate Analysis • Multivariate Analysis Descriptive Statistics. Model Training using Various Algorithms.	9	High	4)Papasani Saikiran	20.06.25	24.07.25
Sprint-4	Performance Testing and Model Deployment	RLCPC-20 RLCPC-21 RLCPC-23	• Testing Model with Evaluation Metrics • Hyperparameter Tuning • Integrating with Web Framework	10	High	5)Muthineni Naga Raju	24.07.25	28.07.25

Project Initialization and Planning Phase

Date	12th June 2024
Team ID	LTVIP2025TMID38009
Project Title	**Revolutionizing Liver Care : Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques**
Maximum Marks	3 Marks

Project Proposal (Proposed Solution)

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel.

Project Overview	
Objective	To develop an advanced machine learning model that predicts the onset or progression of liver cirrhosis, facilitating early detection and intervention, and improving patient outcomes.
Scope	• **Data Sources**: Integrate patient data such as medical history, lab results, and lifestyle factors. • **Model Development**: Utilize state-of-the-art machine learning techniques to create a predictive model. • **Deployment**: Implement the model in healthcare settings to support patient screening, treatment planning, and resource allocation.
Problem Statement	
Description	This project aims to revolutionize liver care by creating a machine learning model to predict liver cirrhosis. Liver cirrhosis, characterized by the scarring of liver tissue, results from long-term liver damage. The model will analyze comprehensive patient data to predict the likelihood of cirrhosis, assisting healthcare professionals in making informed decisions about patient care.

Impact	• **Early Detection**: Enables early intervention, potentially improving patient outcomes and preventing complications.

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

Software

	• **Improved Treatment**: Assists in creating personalized treatment plans for patients at risk of or already suffering from liver diseases. • **Optimized Resource Allocation**: Helps healthcare facilities prioritize high-risk patients, ensuring efficient use of resources and timely care.
Proposed Solution	
Approach	• **Data Collection**: Gather and preprocess patient data, including medical history, lab results and lifestyle factors. • **Model Training**: Develop and train machine learning models using advanced techniques. • **Validation and Testing**: Validate the model using existing patient data and test its predictive accuracy. • **Deployment**: Integrate the model into healthcare systems such as EHR for real-time use. • **Monitoring and Iteration**: Continuously monitor model performance and update as needed based on new data and outcomes.

Key Features	• **Predictive Analytics**: Provides early warning signals for liver cirrhosis onset and progression. • **Resource Optimization**: Enhances the allocation of healthcare resources by identifying high-risk patients who need immediate attention. • **Continuous Learning**: Adapts and improves over time with new data inputs and outcomes. • **User Interface**: Develop a user-friendly interface for healthcare providers.

Resource Requirements

Frameworks	Python frameworks	Flask
Libraries	Additional libraries	scikit-learn, pandas, numpy, matplotlib, seaborn.
Development Environment	IDE, version control	Jupyter Notebook, Git
Data		
Data	Source, size, format	Kaggle dataset, 950 rows X 42 columns, EXCEL

Data Collection and Preprocessing Phase

Date	16th June 2025
Team ID	LTVIP2025TMID38009
Project Title	Revolutionizing Liver Care : Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description

Data Overview

Dimensions : 950 x 42

Shape : (950,42)

```
df.shape
```

```
(950, 42)
```

Head:

S.NO	Age	Gender	Place(location where the patient lives)	Duration of alcohol consumption(years)	Quantity of alcohol consumption (quarters/day)	Type of alcohol consumed	Hepatitis B infection	Hepatitis C infection	Diabetes Result	...	Indirect (mg/dl)	Total Protein (g/dl)	Albumin (g/dl)	Globulin (g/dl)	A/G Ratio	AL.Phosphatase (U/L)	SGOT/AST (U/L)	SGPT/ALT (U/L)
0	1	50	male	rural	11	1	[illegible]	negative	negative	YES	...	1.1	8.0	3.8	4.3	0.75	192.4	96	34
1	1	50	male	rural	11	1	[illegible]	negative	negative	YES	...	1.1	8.0	3.8	4.3	0.75	192.4	96	34
2	1	50	male	rural	11	1	[illegible]	negative	negative	YES	...	1.1	8.0	3.8	4.3	0.75	192.4	96	34
3	1	50	male	rural	11	1	[illegible]	negative	negative	NO	...	1.1	8.0	3.8	4.3	0.75	192.4	96	34
4	1	50	female	rural	11	1	[illegible]	negative	negative	YES	...	1.1	8.0	3.8	4.3	0.75	192.4	96	34

5 rows × 42 columns

Overview of columns:

```
Data columns (total 42 columns):
 #   Column                                                  Non-Null Count  Dtype
---  ------                                                  --------------  -----
 0   S.NO                                                    1250 non-null   float64
 1   Age                                                     1250 non-null   float64
 2   Gender                                                  1250 non-null   object
 3   Place(location where the patient lives)                 1116 non-null   object
 4   Duration of alcohol consumption(years)                  1250 non-null   float64
 5   Quantity of alcohol consumption (quarters/day)          1250 non-null   float64
 6   Type of alcohol consumed                                1250 non-null   object
 7   Hepatitis B infection                                   1250 non-null   object
 8   Hepatitis C infection                                   1250 non-null   object
 9   Diabetes Result                                         1250 non-null   object
 10  Blood pressure (mmhg)                                   1250 non-null   object
 11  Obesity                                                 1250 non-null   object
 12  Family history of cirrhosis/ hereditary                 1250 non-null   object
 13  TCH                                                     591 non-null    float64
 14  TG                                                      591 non-null    object
 15  LDL                                                     591 non-null    object
 16  HDL                                                     582 non-null    float64
 17  Hemoglobin  (g/dl)                                      1250 non-null   float64
 18  PCV  (%)                                                1220 non-null   float64
 19  RBC  (million cells/microliter)                         698 non-null    float64
 20  MCV   (femtoliters/cell)                                1241 non-null   float64
 21  MCH  (picograms/cell)                                   592 non-null    float64
 22  MCHC  (grams/deciliter)                                 578 non-null    float64
 23  Total Count                                             1240 non-null   float64
 24  Polymorphs  (%)                                         1250 non-null   float64
 25  Lymphocytes  (%)                                        1250 non-null   float64
 26  Monocytes   (%)                                         1241 non-null   float64
 27  Eosinophils   (%)                                       1242 non-null   float64
 28  Basophils  (%)                                          1201 non-null   float64
 29  Platelet Count  (lakhs/mm)                              1250 non-null   float64
 30  Total Bilirubin    (mg/dl)                              1250 non-null   object
 31  Direct    (mg/dl)                                       1250 non-null   float64
 32  Indirect     (mg/dl)                                    1195 non-null   float64
 33  Total Protein     (g/dl)                                1189 non-null   float64
 34  Albumin   (g/dl)                                        1241 non-null   float64
 35  Globulin  (g/dl)                                        1221 non-null   float64
 36  A/G Ratio                                               785 non-null    object
 37  AL.Phosphatase      (U/L)                               1240 non-null   float64
 38  SGOT/AST      (U/L)                                     1250 non-null   float64
 39  SGPT/ALT (U/L)                                          1250 non-null   float64
 40  USG Abdomen (diffuse liver or  not)                     1250 non-null   object
 41  Predicted Value(Out Come-Patient suffering from liver  cirrosis or not) 1196 non-null   object
dtypes: float64(27), object(15)
```

Duplicate rows:

```
[732] df.duplicated().sum()
```

```
0
```

Target value to predict:

```
Predicted Value(Out Come-Patient suffering from liver  cirrosis or not)
YES    876
no      20
```

Object columns:

```
object_cols = df.select_dtypes(include='object').columns.tolist()
for col in object_cols:
  print(col)
```

```
Gender
Place(location where the patient lives)
Type of alcohol consumed
Hepatitis B infection
Hepatitis C infection
Diabetes Result
Blood pressure (mmhg)
Obesity
Family history of cirrhosis/ hereditary
TG
LDL
Total Bilirubin    (mg/dl)
A/G Ratio
USG Abdomen (diffuse liver or  not)
Predicted Value(Out Come-Patient suffering from liver  cirrosis or not)
```

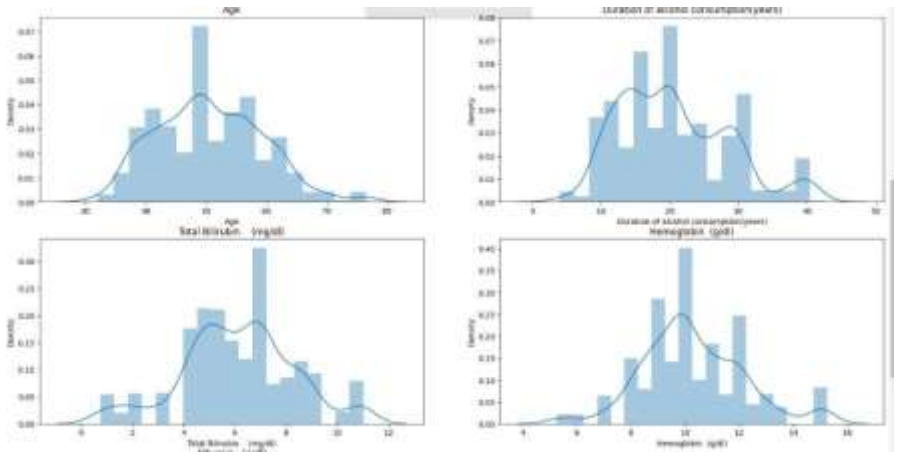
Univariate Analysis

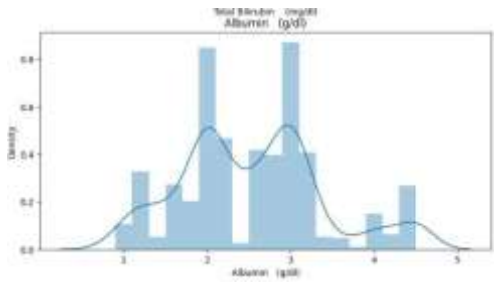
Exploration using Distplots:

Code:

```
l=['Age','Duration of alcohol consumption(years)','Total Bilirubin    (mg/dl)','Hemoglobin  (g/dl)','Albumin   (g/dl)']

plt.figure(figsize=(20, 15))

for i, col in enumerate(l):
    plt.subplot(3, 2, i + 1)
    sns.distplot(df[col])
    plt.title(col)

plt.show()
```

Plots:





Inference:

Inferences from Density Plots

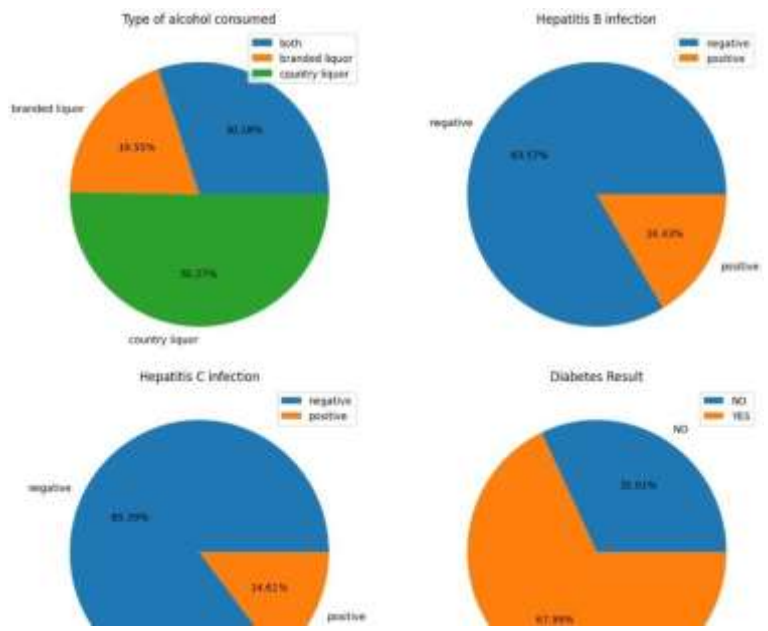
Age Distribution:

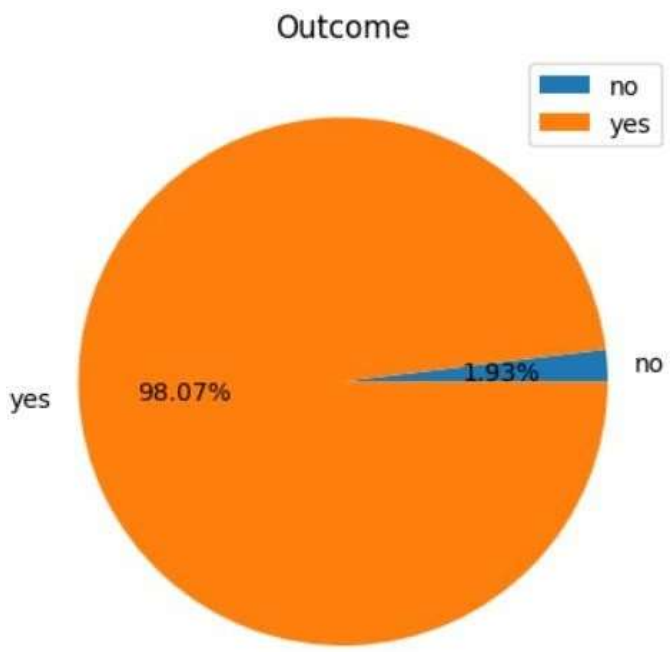
- The majority of patients fall within the 40-60 age range.
- There is a noticeable peak around the age of 50, indicating a higher frequency of patients in their early 50s.

2. **Duration of Alcohol Consumption**:
   - The duration of alcohol consumption varies widely among patients.
   - A significant proportion of patients have been consuming alcohol for around 15-25 years, with a peak at approximately 20 years.
3. **Total Bilirubin**:
   - The total bilirubin levels show a wide distribution, with a peak around 6 mg/dl.
   - There are some patients with very high bilirubin levels, indicating possible liver dysfunction.
4. **Hemoglobin Levels**:
   - Hemoglobin levels are generally distributed around a mean of approximately 10 g/dl.
   - The distribution shows a peak around 10-12 g/dl, suggesting that most patients have moderate to normal hemoglobin levels.

Representing all the important catogorical columns in pie chart

Pie charts:


Type of alcohol consumed
both
branded liquor
country liquor
branded liquor
country liquor
Hepatitis B infection
negative
positive
negative
positive
Hepatitis C infection
negative
positive
negative
positive
Diabetes Result
NO
YES
NO


Outcome
no
yes
yes
98.07%
1.93%
no

Code:

```
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

# Type of alcohol consumed
df.groupby("Type of alcohol consumed").size().plot(kind="pie", autopct="%.2f%%", ax=axes[0, 0], legend=True)
axes[0, 0].set_title("Type of alcohol consumed")

# Hepatitis B infection
df.groupby("Hepatitis B infection").size().plot(kind="pie", autopct="%.2f%%", ax=axes[0, 1], legend=True)
axes[0, 1].set_title("Hepatitis B infection")

# Hepatitis C infection
df.groupby("Hepatitis C infection").size().plot(kind="pie", autopct="%.2f%%", ax=axes[1, 0], legend=True)
axes[1, 0].set_title("Hepatitis C infection")

# Diabetes Result
df.groupby("Diabetes Result").size().plot(kind="pie", autopct="%.2f%%", ax=axes[1, 1], legend=True)
axes[1, 1].set_title("Diabetes Result")



plt.tight_layout()
plt.show()
```

Statistical analysis for individual variables:

	Age	Duration of alcohol consumption(years)	Quantity of alcohol consumption (quarters/day)	TCH	TG	LDL	HDL	Hemoglobin (g/dl)	PCV (%)	RBC (million cells/microliter)	...	Total Bilirubin (mg/dl)	Direct (mg/dl)	Indirect (mg/dl)	Total Protein (g/dl)
count	1182.000000	1182.000000	1182.000000	1182.000000	1182.000000	1182.0	1182.000000	1182.000000	1182.000000	1182.000000	...	1182.000000	1182.000000	1182.000000	1182.000000
mean	50.588323	17.857784	2.344667	196.460840	162.715736	100.3	55.221981	10.589428	35.170343	3.973412	...	4.366445	3.215440	2.330238	5.732175
std	8.622177	9.589442	0.936708	4.689123	6.594894	0.0	3.653208	1.886677	3.468803	0.177845	...	2.979027	2.064808	0.963896	1.217197
min	31.738079	3.030435	1.000000	188.883808	153.500000	100.3	34.273819	5.206258	21.000000	3.730220	...	0.300000	0.723412	0.381330	2.750000
25%	45.000000	10.000000	2.000000	194.300000	161.000000	100.3	38.000000	9.222969	32.000000	3.831766	...	2.000000	1.100000	1.000000	5.000000
50%	50.000000	17.000000	2.000000	197.344088	161.000000	100.3	55.406224	10.000000	36.000000	3.931766	...	3.200000	3.200000	2.124024	6.000000
75%	56.238032	25.000000	3.000000	197.344088	166.000000	100.3	55.406224	11.000000	38.000000	4.098197	...	7.000000	4.200000	3.000000	6.000000
max	73.548037	45.000000	4.000000	202.867498	173.000000	100.3	58.215838	15.000000	48.500000	4.267735	...	11.000000	8.600000	4.580158	8.300000

8 rows × 30 columns

Albumin (g/dl)	Globulin (g/dl)	A/G Ratio	AL.Phosphatase (U/L)	SGOT/AST (U/L)	SGPT/ALT (U/L)
182.000000	1182.000000	1182.000000	1182.000000	1182.000000	1182.000000
2.965578	3.130965	1.056125	124.464881	87.083213	61.483339
1.207149	0.910346	0.575430	30.762279	29.061998	22.207486
0.900000	1.000000	0.090000	50.771505	32.000000	23.000000
2.000000	2.500000	0.640000	104.730578	61.000000	43.000000
2.900000	3.000000	0.900000	119.656197	84.000000	60.000000
3.875198	3.800000	1.490000	146.000000	109.565245	74.212239
6.687995	5.750000	2.765000	206.000000	182.413113	121.030597

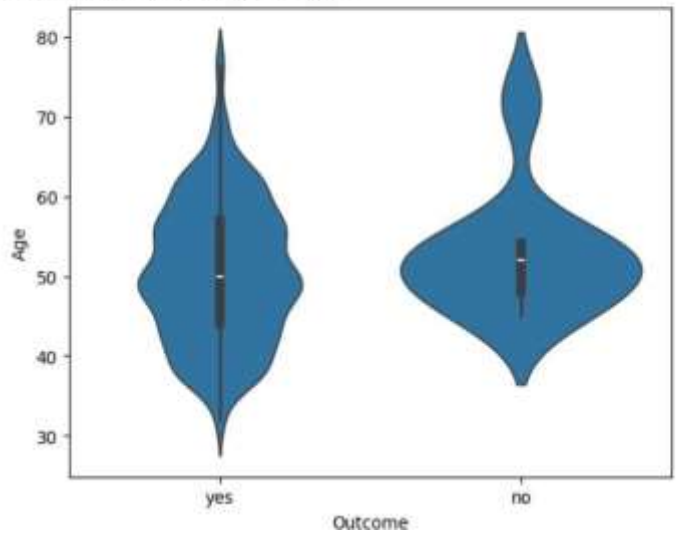
Mean of all numerical columns:

Age	50.588614
Duration of alcohol consumption(years)	20.552632
Quantity of alcohol consumption (quarters/day)	2.195489
TCH	195.816696
TG	163.541353
LDL	106.040279
HDL	34.914618
Hemoglobin (g/dl)	10.266305
PCV (%)	33.900873
RBC (million cells/microliter)	3.386582
MCV (femtoliters/cell)	87.434408
MCH (picograms/cell)	30.512111
MCHC (grams/deciliter)	31.907273
Total Count	8149.711704
Polymorphs (%)	66.932331
Lymphocytes (%)	26.006445
Monocytes (%)	3.633432
Eosinophils (%)	2.269037
Basophils (%)	0.469048
Platelet Count (lakhs/mm)	1.441933
Total Bilirubin (mg/dl)	6.118582
Direct (mg/dl)	3.704834
Indirect (mg/dl)	2.423035
Total Protein (g/dl)	5.595907
Albumin (g/dl)	2.529510
Globulin (g/dl)	3.225369
A/G Ratio	0.855725
AL.Phosphatase (U/L)	132.292207
SGOT/AST (U/L)	80.383459

Median:

Age	50.000000
Duration of alcohol consumption(years)	20.000000
Quantity of alcohol consumption (quarters/day)	2.000000
TCH	197.423932
TG	161.000000
LDL	106.000000
HDL	35.516464
Hemoglobin (g/dl)	10.000000
PCV (%)	35.000000
RBC (million cells/microliter)	3.386582
MCV (femtoliters/cell)	87.000000
MCH (picograms/cell)	30.512111
MCHC (grams/deciliter)	31.907273
Total Count	7500.000000
Polymorphs (%)	65.000000
Lymphocytes (%)	27.000000
Monocytes (%)	3.000000
Eosinophils (%)	2.000000
Basophils (%)	0.000000
Platelet Count (lakhs/mm)	1.400000
Total Bilirubin (mg/dl)	6.000000
Direct (mg/dl)	3.600000
Indirect (mg/dl)	2.400000
Total Protein (g/dl)	6.000000
Albumin (g/dl)	2.500000
Globulin (g/dl)	3.100000
A/G Ratio	0.780000
AL.Phosphatase (U/L)	130.000000
SGOT/AST (U/L)	74.000000
SGPT/ALT (U/L)	49.000000

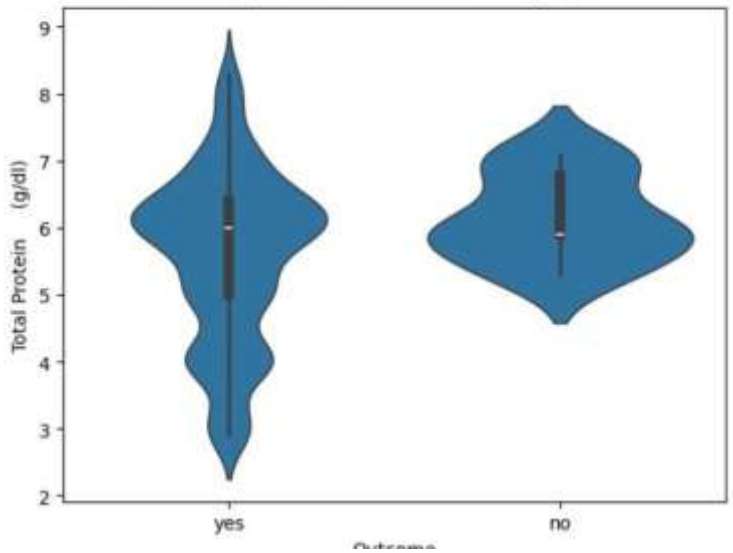
dtype: float64

Bivariate Analysis	**Violin Plots Between Two Variables:**
	How does age vary with outcome
	`sns.violinplot(y=df["Age"],x=df["Outcome"])`
	`<Axes: xlabel='Outcome', ylabel='Age'>`
	 
	
	Inference:
	Inferences from Violin Plot
	The violin plot shows the age distribution for patients with and without liver cirrhosis.
	• **Patients with Liver Cirrhosis (Yes):**
	▫ Broader age distribution with multiple peaks.
	▫ Concentration around 50-60 years.
	• **Patients without Liver Cirrhosis (No):**
	▫ More uniform age distribution.
	▫ Noticeable peak around 50 years.
	Conclusion
	Liver cirrhosis affects a wider range of ages, especially 50-60 years, while the age distribution for patients without cirrhosis is more consistent.

How does protein influence outcome

```
sns.violinplot(y=df["Total Protein     (g/dl)"],x=df["Outcome"])
```

```
<Axes: xlabel='Outcome', ylabel='Total Protein     (g/dl)'>
```



Inference:

Total Protein Distribution:

Patients with liver cirrhosis ("yes") have a wider distribution of total protein levels ranging from approximately 3 g/dl to 9 g/dl.

Patients without liver cirrhosis ("no") have a slightly narrower distribution, with total protein levels ranging from approximately 4.5 g/dl to 8 g/dl.

Median Total Protein Levels:

The median total protein level in patients with liver cirrhosis is slightly higher than in those without liver cirrhosis, as indicated by the white dot in the center of each violin.els.

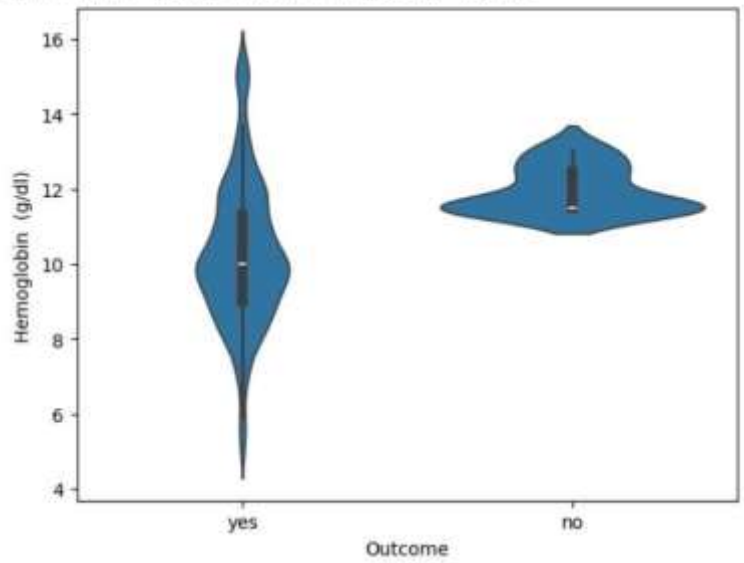
Double-click (or enter) to edit

PROTEIN LEVEL HAS CONSIDERABLE EFFECT ON OUTCOME

How does haemoglobin affect the outcome

```
sns.violinplot(y=df["Hemoglobin  (g/dl)"],x=df["Outcome"])
```

```
<Axes: xlabel='Outcome', ylabel='Hemoglobin  (g/dl)'>
```



Inference:

Distribution:

Cirrhosis ("yes"): Hemoglobin levels range broadly from approximately 4 g/dl to 16 g/dl.

No cirrhosis ("no"): Hemoglobin levels are more concentrated, ranging from about 11 g/dl to 14 g/dl. Median Levels:

Cirrhosis: The median hemoglobin level is around 10 g/dl.

No Cirrhosis: The median hemoglobin level is also around 11.5 g/dl.

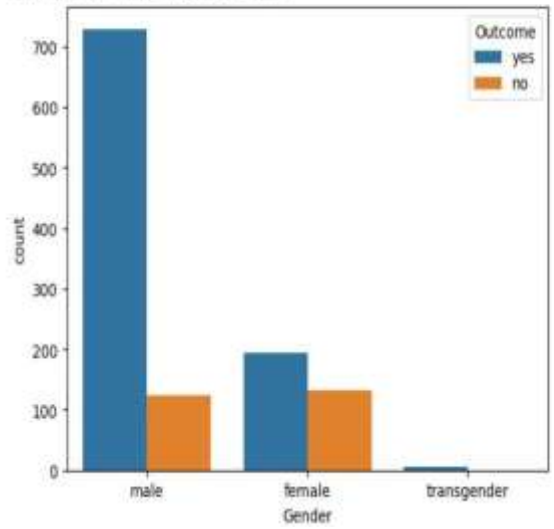
Comparison:

- Liver disease is associated with a wider range of hemoglobin levels.
- No liver disease shows more consistent hemoglobin levels centered around 11.5 g/dl.

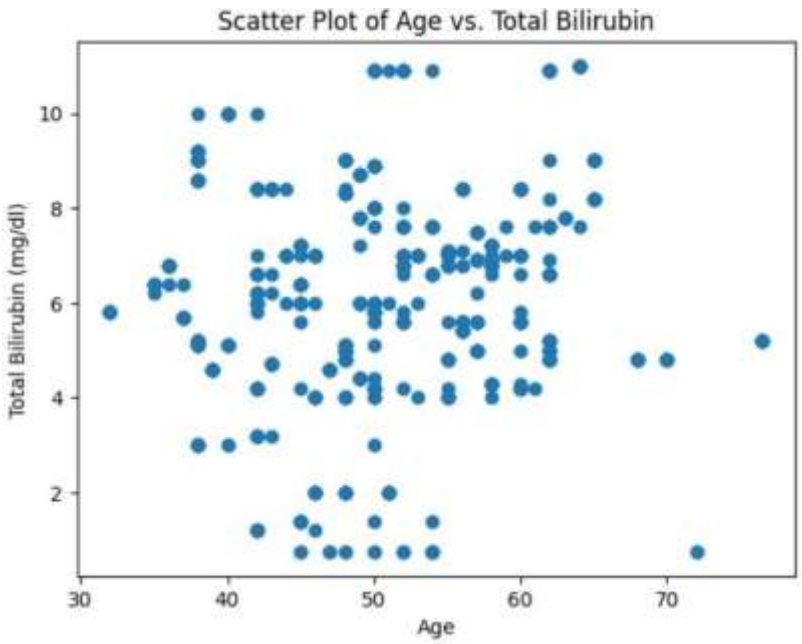
How does the distribution of gender affect the outcome

```
sns.countplot(data=df,x="Gender",hue="Outcome")
```

```
<Axes: xlabel='Gender', ylabel='count'>
```

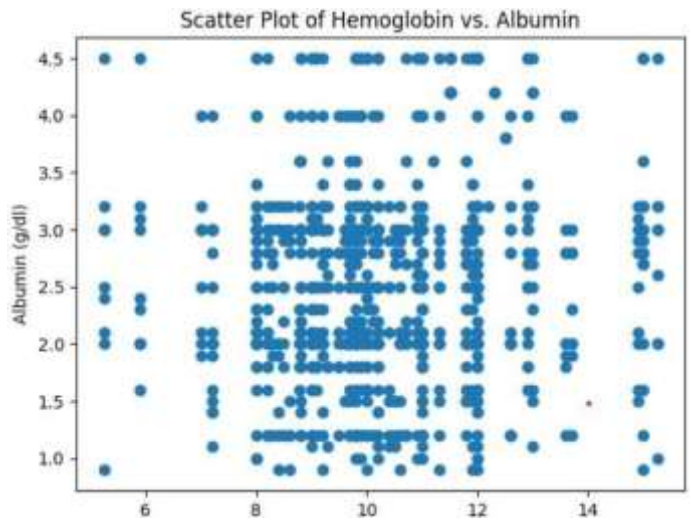


Scatter Plots:



Inference:
No Clear Trend:

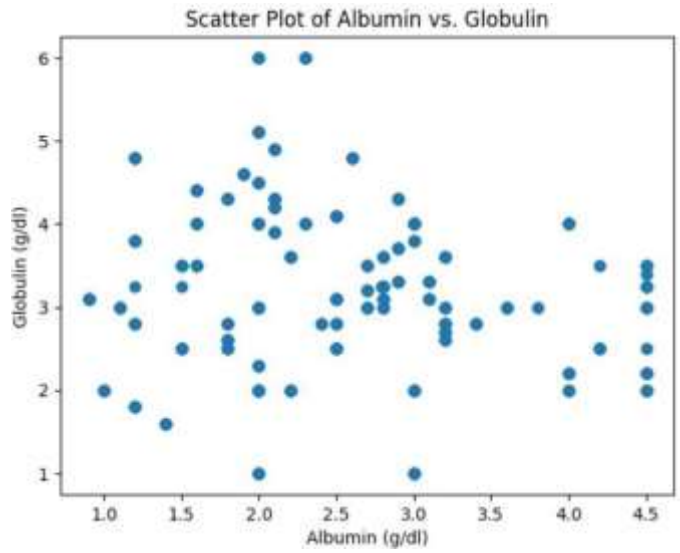
- There doesn't appear to be a clear linear relationship between age and Total Bilirubin levels.
- Total Bilirubin levels are spread across the age range without a consistent pattern.



Inference:

A large cluster of data points is concentrated around Haemoglobin levels of 8 to 12 g/dl and Albumin levels of 1.3 to 3 g/dl.

This suggests that most individuals in the dataset have Haemoglobin levels within this range.



Inference:

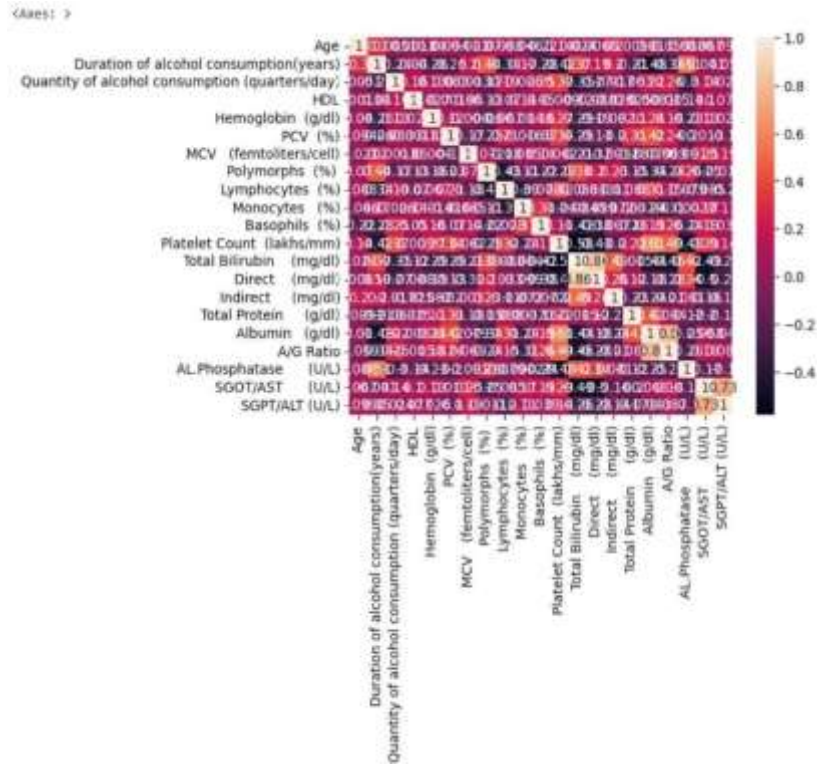
Inferences from the Scatter Plot of Albumin vs. Globulin:

1. **No Strong Correlation**:
   - The scatter plot indicates no strong linear relationship between albumin and globulin levels. The data points are widely scattered, suggesting that variations in albumin levels do not directly predict changes in globulin levels.
2. **Range of Values**:
   - Most albumin levels fall within the range of 2.0 to 4.0 g/dl, while globulin levels range from 2.0 to 5.0 g/dl. There are some outliers with higher globulin levels up to 6.0 g/dl and albumin levels up to 4.5 g/dl, indicating diverse liver function profiles among the patients.

Multivariate Analysis	**Correlation Matrix:**

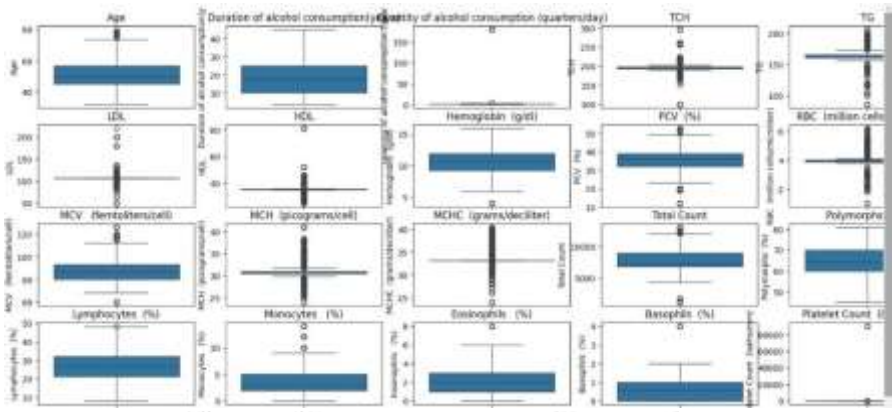
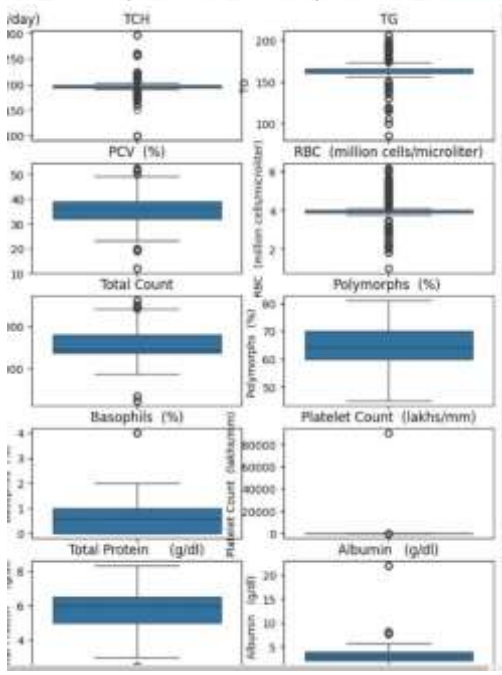
	Age	Duration of alcohol consumption(years)	Quantity of alcohol consumption (quarters/day)	HDL	Hemoglobin (g/dl)	PCV (%)	MCV (femtoliters/cell)	Polymorphs (%)	Lymphocytes (%)	Monocytes (%)	...	Platelet Count (lakhs/cmm)	Total Bilirubin (mg/dl)	Direct (mg/dl)	Indirect (mg/dl)
Age	1.000000	0.004468	-0.014839	0.007521	0.005843	0.007591	0.044936	0.192051	-0.201666	-0.011715	...	-0.011571	0.11111	0.088815	0.241744
Duration of alcohol consumption(years)	0.004468	1.000000	0.01311	0.112448	-0.000020	-0.002314	0.275167	0.263651	-0.363058	0.195233	...	-0.000403	-0.190130	-0.258064	0.596273
Quantity of alcohol consumption (quarters/day)	-0.014839	0.01311	1.000000	0.087620	-0.020515	-0.195242	-0.003968	-0.001462	0.152467	-0.043946	...	0.111797	-0.058669	-0.070314	0.006486
HDL	0.007521	0.112448	-0.087620	1.000000	-0.031186	-0.042623	-0.003328	-0.170694	-0.014407	0.114646	...	-0.145020	0.006465	-0.001302	0.038587
Hemoglobin (g/dl)	0.005843	-0.000020	-0.020515	-0.031186	1.000000	-0.006740	-0.008803	-0.023486	0.010444	-0.000877	...	0.027363	-0.046225	0.015173	-0.015270
PCV (%)	0.007591	-0.002314	-0.195242	-0.042623	-0.006740	1.000000	-0.215278	-0.081485	0.060000	-0.014451	...	0.200573	0.008345	0.124941	-0.147214
MCV (femtoliters/cell)	0.044936	0.275167	-0.003968	-0.003328	-0.008803	-0.215278	1.000000	0.099023	-0.223936	0.114024	...	0.020725	-0.001073	-0.008698	-0.007002
Polymorphs (%)	0.192051	0.263651	-0.001462	-0.170694	-0.023486	-0.081485	0.099023	1.000000	-0.475791	-0.129826	...	-0.000582	0.006242	-0.006245	0.160371
Lymphocytes (%)	-0.201666	-0.363058	0.152467	-0.014407	0.010444	0.060000	-0.223936	-0.475791	1.000000	-0.320436	...	0.052883	0.087115	0.223006	-0.302456
Monocytes (%)	-0.011715	0.195233	-0.043946	0.114646	-0.000877	-0.014451	0.114024	-0.129826	-0.320436	1.000000	...	-0.003730	-0.150023	-0.192125	-0.072397
Basophils (%)	-0.200273	-0.107298	0.130676	-0.072190	0.052183	-0.296858	0.170891	-0.084883	-0.180879	0.039419	...	-0.027985	-0.234260	-0.108647	0.084270
Platelet Count (lakhs/cmm)	0.011571	-0.003403	0.111797	-0.145020	0.027363	0.200573	0.020725	-0.000582	0.052883	-0.003730	...	1.000000	-0.213864	-0.130197	-0.164126
Total Bilirubin (mg/dl)	0.11111	-0.190130	-0.058669	0.006465	-0.046225	0.008345	-0.001073	0.006242	0.087115	-0.150023	...	-0.213864	1.000000	0.963861	0.525914
Direct (mg/dl)	0.088815	-0.258064	-0.070314	-0.001302	0.015173	0.124941	-0.008698	-0.006245	0.223006	-0.192125	...	-0.130197	0.963861	1.000000	0.229858
Indirect (mg/dl)	0.241744	0.596273	0.006486	0.038587	-0.015270	-0.147214	-0.007002	0.160371	-0.302456	-0.072397	...	-0.164126	0.525914	0.229858	1.000000
Total Protein (g/dl)	0.001947	-0.127566	-0.170583	0.028104	0.006243	0.210468	0.110044	-0.070410	0.118241	0.000974	...	-0.165074	0.217345	0.207692	-0.214080
Albumin (g/dl)	-0.025472	-0.280224	-0.084484	-0.191825	0.008944	0.346276	-0.128855	-0.150786	0.387131	-0.244547	...	0.030975	-0.380115	0.085884	-0.202710
A/G Ratio	-0.072870	-0.150170	0.12869	-0.170873	0.040881	0.120890	0.021159	-0.044486	0.181401	-0.081881	...	0.049302	-0.247992	-0.171119	-0.104798
AL.Phosphatase (U/L)	0.117445	0.012634	-0.084731	-0.140857	-0.008015	0.003818	-0.043214	0.047375	0.088048	0.008877	...	-0.119040	0.000090	0.084656	-0.036868	
SGOT/AST (u/L)	0.001609	0.000867	-0.062026	0.071819	-0.000007	-0.152200	0.275113	-0.150152	-0.215484	0.210727	...	0.037232	-0.245566	-0.072328	-0.115588

Heatmap:



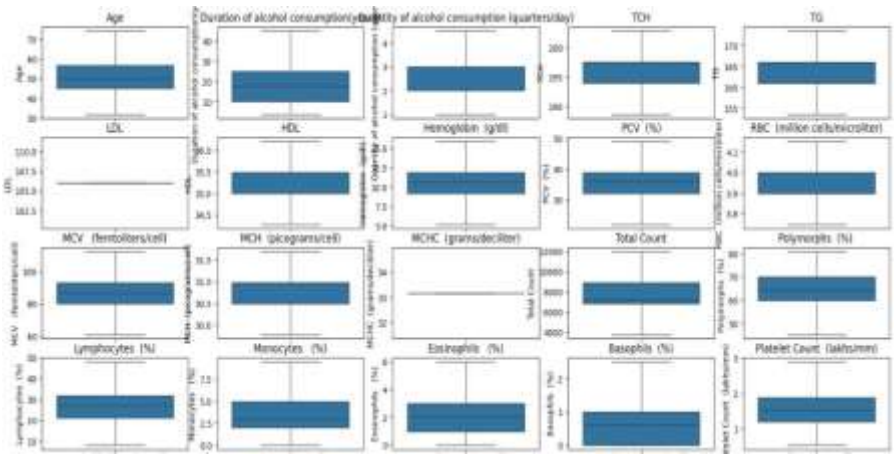
Columns having high correlation:

```
[456] correlation_matrix = df1.corr(numeric_only=True)
      high_correlation_pairs = []
      for i in range(len(correlation_matrix.columns)):
          for j in range(i + 1, len(correlation_matrix.columns)):
              if abs(correlation_matrix.iloc[i, j]) > 0.8:
                  high_correlation_pairs.append((correlation_matrix.columns[i], correlation_matrix.columns[j], correlation_matrix.iloc[i, j]))
      for pair in high_correlation_pairs:
          print(f"{pair[0]} and {pair[1]}: {pair[2] * 100:.2f}%")

Total Bilirubin (mg/dl) and Direct (mg/dl): 86.07%
```

Outliers and Anomalies	**Identification using boxplot:**
	
	``
	c=0
	plt.figure(figsize=(20,15))
	for i in df.columns:
	  if(type(df[i][0])!=str):
	  plt.subplot(7,5,c+1)
	  # Attempt to convert the column to numeric, handling errors by coercing them to NaN
	  sns.boxplot(df[i].apply(pd.to_numeric, errors='coerce'))
	  plt.title(i)
	  c=c+1
	plt.show()
	``
	
	 
	
	 
	
	Removal using IQR:

```
def remove_outliers(df, columns):
  for col in columns:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df[col] = np.where(df[col] < lower_bound, lower_bound,np.where(df[col] > upper_bound, upper_bound, df[col]))

numerical_columns = df.select_dtypes(include=['int64', 'float64']).columns
remove_outliers(df, numerical_columns)
```

After removing:



Data Preprocessing Code Screenshots

Loading Data	`[890] df=pd.read_excel("HealthCareData.xlsx")`

Handling Missing Data	```df.isnull().sum()``` S.NO

Missing values in Data:

S.NO	0
Age	0
Gender	0
Place(location where the patient lives)	134
Duration of alcohol consumption(years)	0
Quantity of alcohol consumption (quarters/day)	0
Type of alcohol consumed	0
Hepatitis B infection	0
Hepatitis C infection	0
Diabetes Result	0
Blood pressure (mmhg)	0
Obesity	0
Family history of cirrhosis/ hereditary	0
TCH	359
TG	359
LDL	359
HDL	368
Hemoglobin (g/dl)	0
PCV (%)	30
RBC (million cells/microliter)	552
MCV (femtoliters/cell)	9
MCH (picograms/cell)	658
MCHC (grams/deciliter)	672
Total Count	10
Polymorphs (%)	0
Lymphocytes (%)	0
Monocytes (%)	9
Eosinophils (%)	8
Basophils (%)	49
Platelet Count (lakhs/mm)	0
Total Bilirubin (mg/dl)	0
Direct (mg/dl)	0
Indirect (mg/dl)	55
Total Protein (g/dl)	61
Albumin (g/dl)	9
Globulin (g/dl)	29
A/G Ratio	359
AL.Phosphatase (U/L)	10
SGOT/AST (U/L)	0
SGPT/ALT (U/L)	0
USG Abdomen (diffuse liver or not)	0
Predicted Value(Out Come-Patient suffering from liver cirrosis or not)	54

dtype: int64

Cleaning Numerical columns:

We can see TG LDL and Bilirubin are object type but they have numeric values

```
print(df["TG"].head(3))
print(df["LDL"].head(3))
print(df["Total Bilirubin    (mg/dl)"].head(3))
```

```
0    115
1    115
2    115
Name: TG, dtype: object
0    120
1    120
2    120
Name: LDL, dtype: object
0    7
1    7
2    7
Name: Total Bilirubin    (mg/dl), dtype: object
```

By using value_counts() we can notice that:

- TG contains a row - 130LDL
- LD contains a row - HDL
- Bilirubin contains a row - o.4

```
[901] print(df["TG"].value_counts())
      print(df["LDL"].value_counts())
      print(df["Total Bilirubin    (mg/dl)"].value_counts())
```

Dropping those rows

```
[902] df = df[df['TG'] != '130LDL']
      df = df[df['LDL'] != 'HDL']
      df = df[df['Total Bilirubin    (mg/dl)'] != 'o.4']
```

Converting into float

```
[903] df["TG"] = df["TG"].astype(float)
      df["LDL"] = df["LDL"].astype(float)
      df["Total Bilirubin    (mg/dl)"] = df["Total Bilirubin    (mg/dl)"].astype(float)
```

Filling numeric columns with mean:

Filling all numerical columns with their mean

```
[904] numerical_columns = df.select_dtypes(include=['int64', 'float64']).columns
      for col in numerical_columns:
        df[col].fillna(df[col].mean(), inplace=True)

      df.isnull().sum()
```

```
S.NO                                                                  0
Age                                                                   0
Gender                                                                0
Place(location where the patient lives)                             133
Duration of alcohol consumption(years)                                0
Quantity of alcohol consumption (quarters/day)                        0
Type of alcohol consumed                                              0
Hepatitis B infection                                                 0
Hepatitis C infection                                                 0
Diabetes Result                                                       0
Blood pressure (mmhg)                                                 0
Obesity                                                               0
Family history of cirrhosis/ hereditary                               0
TCH                                                                   0
TG                                                                    0
LDL                                                                   0
HDL                                                                   0
Hemoglobin  (g/dl)                                                    0
PCV  (%)                                                              0
RBC  (million cells/microliter)                                       0
MCV   (femtoliters/cell)                                              0
MCH  (picograms/cell)                                                 0
MCHC  (grams/deciliter)                                               0
Total Count                                                           0
Polymorphs  (%)                                                       0
Lymphocytes  (%)                                                      0
Monocytes   (%)                                                       0
Eosinophils   (%)                                                     0
Basophils  (%)                                                        0
Platelet Count  (lakhs/mm)                                            0
Total Bilirubin    (mg/dl)                                            0
Direct    (mg/dl)                                                     0
Indirect     (mg/dl)                                                  0
Total Protein     (g/dl)                                              0
Albumin   (g/dl)                                                      0
Globulin  (g/dl)                                                      0
A/G Ratio                                                           437
AL.Phosphatase      (U/L)                                             0
SGOT/AST      (U/L)                                                   0
SGPT/ALT (U/L)                                                        0
USG Abdomen (diffuse liver or  not)                                   0
Predicted Value(Out Come-Patient suffering from liver  cirrosis or not)   54
dtype: int64
```

Cleaning Abnormalities found in data:

Removing the abnormalities

```
[403] df = df[df["Platelet Count  (lakhs/mm)"] != 90000.000]
```

```
[300] df["Quantity of alcohol consumption (quarters/day)"].value_counts()

Quantity of alcohol consumption (quarters/day)
2      528
3      198
1      158
4       54
100     16
5        1
Name: count, dtype: int64
```

Removing the abnormalities

```
[400] df["Quantity of alcohol consumption (quarters/day)"] = df["Quantity of alcohol consumption (quarters/day)"].replace(100, 5)

df["Quantity of alcohol consumption (quarters/day)"].value_counts()

Quantity of alcohol consumption (quarters/day)
2    528
3    198
1    158
4     54
5     17
Name: count, dtype: int64
```

```
df=df[df["Albumin   (g/dl)"]!=22.0]
```

```
df=df[df["Globulin  (g/dl)"]!=30.0]
```

Cleaning A/G Ratio:

Making it in the correct format

```
[907] df["A/G Ratio"] = round(df["Albumin   (g/dl)"]/df["Globulin  (g/dl)"],2)

df["A/G Ratio"].value_counts()

A/G Ratio
1.00    99
0.75    87
0.67    49
0.43    30
0.50    30
        ..
1.46     1
1.11     1
1.84     1
1.29     1
2.08     1
Name: count, Length: 137, dtype: int64

[909] df["A/G Ratio"]=df["A/G Ratio"].astype(float)

[910] df["A/G Ratio"].fillna(df["A/G Ratio"].mean(), inplace=True)
```

Cleaning And Transforming Blood Pressure:

```
df["Blood pressure (mmhg)"] = df["Blood pressure (mmhg)"].str.replace('/', '/').str.split('/').apply(lambda x: float(x[0]) / float(x[1]))
```

+ Code + Text

Cleansing Categorical Columns:

Viewing the spread of data in Categorical columns

```
for i in df.columns:
  if df[i].dtype == 'object' and i!="Blood pressure (mmhg)":
    print(df[i].value_counts())
    print("-"*50)
```

```
Gender
male           841
female         194
female         133
transgender      5
Name: count, dtype: int64
--------------------------------------------------
Place(location where the patient lives)
rural    566
urban    473
 ocun      1
Name: count, dtype: int64
--------------------------------------------------
Type of alcohol consumed
country liquor     586
branded liquor     299
both               287
 branded liquor      1
Name: count, dtype: int64
Hepatitis B infection
negative    909
Positive    263
positive      1
Name: count, dtype: int64
--------------------------------------------------
Hepatitis C infection
negative    920
Positive    251
positive      2
Name: count, dtype: int64
--------------------------------------------------
Diabetes Result
YES    647
NO     526
Name: count, dtype: int64
--------------------------------------------------
Obesity
no     624
yes    549
Name: count, dtype: int64
--------------------------------------------------
Family history of cirrhosis/ hereditary
no         984
yes        177
husband     12
Name: count, dtype: int64
--------------------------------------------------
USG Abdomen (diffuse liver or  not)
YES    910
no     263
```

Removing all the abnormalities:

Cleaning the Place column

```
[913] df = df[df['Place(location where the patient lives)'] != ' ocun']
```

Cleaning the Gender column

```
[914] df["Gender"].replace("female ","female",inplace=True)
```

```
<ipython-input-914-fc8ed781fdc6>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  df["Gender"].replace("female ","female",inplace=True)
```

```
[915] df["Gender"].value_counts()
```

```
Gender
male           840
female         327
transgender      5
Name: count, dtype: int64
```

Cleaning alcohol comumption

```
[916] df["Type of alcohol consumed"].replace(" branded liquor","branded liquor",inplace=True)
```

```
<ipython-input-916-54b9cbf74f46>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
  df["Type of alcohol consumed"].replace(" branded liquor","branded liquor",inplace=True)
```

```
df["Type of alcohol consumed"].value_counts()
```

```
Type of alcohol consumed
country liquor    586
branded liquor    300
both              206
Name: count, dtype: int64
```

Cleaning hepatitis column

```
[918] df["Hepatitis B infection"].replace("Positive","positive",inplace=True)
      df["Hepatitis C infection"].replace("Positive","positive",inplace=True)
```

```
[919] df["Hepatitis B infection"].value_counts()
```

```
Hepatitis B infection
negative    988
positive    264
```

```
[920] df["Hepatitis C Infection"].value_counts()
```

```
Hepatitis C Infection
negative    919
positive    253
Name: count, dtype: int64
```

Cleaning family history column

```
df["Family history of cirrhosis/ hereditary"].replace("husband","yes",inplace=True)
df["Family history of cirrhosis/ hereditary"].value_counts()
```

```
Family history of cirrhosis/ hereditary
no     983
yes    189
Name: count, dtype: int64
```

Converting rest of columns to proper format

```
[922] df["Predicted Value(Out Come-Patient suffering from liver  cirrosis or not)"].replace("YES","yes",inplace=True)
df["Predicted Value(Out Come-Patient suffering from liver  cirrosis or not)"].value_counts()
```

```
Predicted Value(Out Come-Patient suffering from liver  cirrosis or not)
yes    874
no     244
Name: count, dtype: int64
```

After cleaning:

```
Gender
male          840
female        327
transgender     5
Name: count, dtype: int64
--------------------------------------------------
Place(location where the patient lives)
rural    566
urban    473
Name: count, dtype: int64
--------------------------------------------------
Type of alcohol consumed
country liquor    586
branded liquor    300
both              286
Name: count, dtype: int64
--------------------------------------------------
Hepatitis B infection
negative    908
positive    264
Name: count, dtype: int64
--------------------------------------------------
Hepatitis C infection
negative    919
positive    253
Name: count, dtype: int64
--------------------------------------------------
Diabetes Result
YES    647
NO     525
Name: count, dtype: int64
--------------------------------------------------
Obesity
no     623
yes    549
Name: count, dtype: int64
--------------------------------------------------
Family history of cirrhosis/ hereditary
no     983
yes    189
Name: count, dtype: int64
--------------------------------------------------
USG Abdomen (diffuse liver or  not)
YES    910
no     262
Name: count, dtype: int64
--------------------------------------------------
```

Cleaning the outcome:

```
[50] df["Outcome"].value_counts()
```

```
Outcome
yes    859
no      18
Name: count, dtype: int64
```

```
df["Outcome"].isnull().sum()
```

```
54
```

Filling all null values of the column with yes

```
[52] df["Outcome"].fillna("yes", inplace=True)
```

Data Transformation

Encoding all the categorical columns:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for i in X.columns:
  if X[i].dtype == 'object':
    X[i] = le.fit_transform(X[i])
```

```
y_encoded =(le.fit_transform(y))
```

Encoded Data:

	Age	Quantity of alcohol consumption (quarters/day)	Diabetes Result	Blood pressure (mmhg)	Hemoglobin (g/dl)	PCV (%)	Polymorphs (%)	Lymphocytes (%)	Platelet Count (lakhs/mm)	Total Bilirubin (mg/dl)	Indirect (mg/dl)	Total Protein (g/dl)	Albumin (g/dl)	Globulin (g/dl)
0	55.0	2.0	1	32	12.0	40.0	60.0	35.0	1.5	7.0	3.0	6.0	3.0	4.0
1	55.0	2.0	1	32	9.2	40.0	60.0	35.0	1.5	7.0	3.0	6.0	3.0	4.0
2	55.0	2.0	1	32	10.2	40.0	60.0	35.0	1.5	7.0	3.0	6.0	3.0	4.0
3	55.0	2.0	0	32	7.2	40.0	60.0	35.0	1.5	7.0	3.0	6.0	3.0	4.0
4	55.0	2.0	1	32	10.2	40.0	60.0	35.0	1.5	7.0	3.0	6.0	3.0	4.0

Feature Engineering	**Feature Importance:**
	
	``python
	from sklearn.ensemble import RandomForestClassifier
	
	model = RandomForestClassifier(n_estimators=100)
	model.fit(X, y)
	importances = model.feature_importances_
	
	# Print feature importances
	for feature, importance in zip(X.columns, importances):
	    print(f"{feature}: {importance:.4f}")
	``
	
	``
	Age: 0.0006
	Gender: 0.0000
	Duration of alcohol consumption(years): 0.1940
	Quantity of alcohol consumption (quarters/day): 0.0206
	Type of alcohol consumed: 0.0000
	Hepatitis B infection: 0.0000
	Hepatitis C infection: 0.0000
	Diabetes Result: 0.0044
	Blood pressure (mmhg): 0.0001
	Obesity: 0.0000
	Family history of cirrhosis/ hereditary: 0.0001
	TCH: 0.0001
	TG: 0.0001
	LDL: 0.0002
	HDL: 0.0003
	Hemoglobin (g/dl): 0.0011
	PCV (%): 0.0007
	RBC (million cells/microliter): 0.0282
	MCV (femtoliters/cell): 0.0007
	MCH (picograms/cell): 0.0194
	MCHC (grams/deciliter): 0.0534
	Total Count: 0.0010
	Polymorphs (%) : 0.0104
	Lymphocytes (%): 0.0058
	Monocytes (%): 0.0025
	Eosinophils (%): 0.0000
	Basophils (%): 0.0074
	Platelet Count (lakhs/mm): 0.0203
	Total Bilirubin (mg/dl): 0.1604
	Direct (mg/dl): 0.1125
	Indirect (mg/dl): 0.0092
	Total Protein (g/dl): 0.0024
	Albumin (g/dl): 0.0800
	Globulin (g/dl): 0.0003
	A/G Ratio: 0.0518
	AL.Phosphatase (U/L): 0.0204
	SGOT/AST (U/L): 0.0199
	SGPT/ALT (U/L): 0.0114
	USG Abdomen (diffuse liver or not): 0.1605
	``

	Removing Unecessary Features: **INFERENCE** In the given output of feature importances from the RandomForestClassifier model, features have an importance score of 0 or very less features are: Gender Hepatitis B infection Hepatitis C infection Family history of cirrhosis/ hereditary TCH TG LDL HDL MCV (femtoliters/cell) DROPPING ALL UNECESSARY COLUMNS [953] drop_col=["Type of alcohol consumed","Gender","Direct (mg/dl)","MCH (picograms/cell)","MCHC (grams/deciliter)", [954] for col in drop_col: if col in X.columns: X.drop(columns=[col],inplace=True)
Save Processed Data	`X.to_csv('new_data1.csv', index=False)`

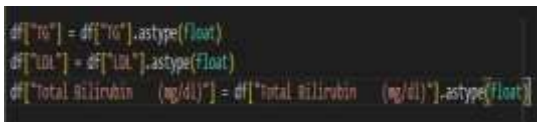
Data Collection and Preprocessing Phase

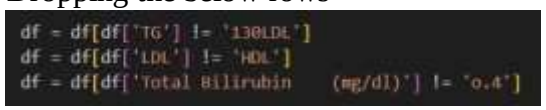
Date	17th June 2025
Team ID	LTVIP2025TMID38009
Project Title	Revolutionizing Liver Care : Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques
Maximum Marks	2 Marks

Data Quality Report:

The Data Quality Report 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	• Too many NULL values	Moderate	Filling the Numerical Columns with Mean and removing abnormalities from categorical column.
			Changing the data type by type casting. Ex-

Kaggle	• Wrong Data Type	Moderate	

Kaggle	• Ambiguous string entries in multiple column	Low	Dropping the below rows 

Kaggle	• Highly imbalanced outcome	Moderate	• Synthetically generating 300 rows with minority class
			 
			• Checking the value counts.
			`df["Outcome"].value_counts()`
			`Outcome`
			`yes  859`
			`no  18`
			`Name: count, dtype: int64`
			• Filling the null values with
			`df["Outcome"].fillna("yes", inplace=True)`

Kaggle	• Many unimportant Features	Moderate	• Removing the features by analyzing the importance scores.

```python
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100)
model.fit(X, y)
importances = model.feature_importances_

# Print feature importances
for feature, importance in zip(X.columns, importances):
    print(f"{feature}: {importance:.4f}")
```

```
Age: 0.0006
Gender: 0.0000
Duration of alcohol consumption(years): 0.1940
Quantity of alcohol consumption (quarters/day): 0.0206
Type of alcohol consumed: 0.0000
Hepatitis B infection: 0.0000
Hepatitis C infection: 0.0000
Diabetes Result: 0.0044
Blood pressure (mmhg): 0.0001
Obesity: 0.0000
Family history of cirrhosis/ hereditary: 0.0001
TCH: 0.0001
TG: 0.0001
LDL: 0.0002
HDL: 0.0003
Hemoglobin  (g/dl): 0.0011
PCV  (%): 0.0007
RBC  (million cells/microliter): 0.0282
MCV  (femtoliters/cell): 0.0007
MCH  (picograms/cell): 0.0194
MCHC  (grams/deciliter): 0.0534
Total Count: 0.0010
Polymorphs  (%) : 0.0104
Lymphocytes  (%): 0.0058
Monocytes  (%): 0.0025
Eosinophils  (%): 0.0000
Basophils  (%): 0.0074
Platelet Count  (lakhs/mm): 0.0203
Total Bilirubin  (mg/dl): 0.1604
Direct  (mg/dl): 0.1125
Indirect  (mg/dl): 0.0092
Total Protein  (g/dl): 0.0024
Albumin  (g/dl): 0.0000
Globulin  (g/dl): 0.0003
A/G Ratio: 0.0518
AL.Phosphatase  (U/L): 0.0204
SGOT/AST  (U/L): 0.0199
SGPT/ALT (U/L): 0.0114
USG Abdomen (diffuse liver or  not): 0.1605
```

```python
for col in drop_col:
  if col in X.columns:
    X.drop(columns=[col],inplace=True)
```

Data Collection and Preprocessing Phase

Date	19th June 2025
Team ID	LTVIP2025TMID38009
Project Title	Revolutionizing Liver Care : Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques
Maximum Marks	2 Marks

Data Collection Plan and Raw Data Sources Identification:

Section	Description
Project Overview	The project aims to develop a predictive model using advanced machine learning techniques to detect the onset or progression of liver cirrhosis in patients. Liver cirrhosis is a serious condition characterized by the scarring of the liver tissue, often resulting from long-term liver damage. Early detection and intervention are crucial for better patient outcomes and to prevent complications. By analyzing various patient data such as medical history, lab results and lifestyle factors, the model will provide predictions regarding the likelihood of liver cirrhosis, helping healthcare professionals make informed decisions about patient care.
Data Collection Plan	Data will be collected from various sources, including medical records, lab results, imaging data, and patient lifestyle information. Specifically, the raw data for this project has been sourced from Kaggle, where a dataset relevant to liver cirrhosis prediction is available.
Raw Data Sources Identified	The primary raw data source identified for this project is a dataset from Kaggle, titled "Liver Cirrhosis Prediction." The dataset contains various patient records with relevant features necessary for

	building the predictive model. The dataset includes medical history, lab test results, and other related health information. The dataset is available in excel format and can be downloaded using the following link: Kaggle Liver Cirrhosis Prediction Dataset.

Raw Data Sources

Source Name	Description	Location /URL	Format	Size	Access Permissions
Kaggle	**Demographics**: Age, gender, and location (rural/urban). **Alcohol Consumption:** Duration, quantity, and type. **Medical History:** Hepatitis B/C, diabetes, blood pressure, obesity, family history of cirrhosis. **Biochemical Markers:** Various blood and liver function test results. **Diagnostic Imaging:** Abdominal ultrasound results. **Outcome:** Indicator of liver cirrhosis presence.	https://w ww.kaggle.com/d atasets/bhavanipriya222/livercirrhosisprediction	EXCEL	240KB	Public

Model Development Phase

Date	20th June 2025
Team ID	LTVIP2025TMID38009
Project Title	Revolutionizing Liver Care : Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques
Maximum Marks	5 Marks

Feature Selection Report

In the forthcoming update, each feature will be accompanied by a brief description. Users will indicate whether it's selected or not, providing reasoning for their decision. This process will streamline decision-making and enhance transparency in feature selection.

Feature	Description	Selected (Yes/No)	Reasoning
Age	It is a numeric column that represents age of an individual	Yes	This data is more widespread among both the classes and would be efficient in explaining the target variable
Quantity of alcohol consumption (quarters/day)	It isa numeric column that has values ranging from 1 to 5	Yes	Alcohol consumption has achieved a good feature importance and would be a good feature to explain the target.

Diabetes Result	It is an object column which has values YES and NO	Yes	Diabetes provides a good base to diagnose liver cirrhosis

Blood pressure (mmhg)	It is an object column that represent the BP of an individual	Yes	In the final model it was found out that it has an importance score of about 0.04.Which makes it a good feature to assess the target
PCV (%): Polymorphs Lymphocytes Platelet Count (lakhs/mm) Indirect	All these are numeric columns that indicate several lab results provided by an individual	Yes	All these features had a relatively good importance score of more than 0.07 in the final model which states that they influence the output pretty well
Haemoglobin	It is a numeric column that represents the total Haemoglobin levels	Yes	Liver disease is associated with a wider range of Haemoglobin levels. No liver disease shows more consistent Haemoglobin levels centered around 11.5 g/dl. This makes it a good feature to be taken

Total Protein	It is a numeric column that represents the total Protein levels	Yes	Patients with liver cirrhosis ("yes") have a wider distribution of total protein levels ranging from approximately 3 g/dl to 9 g/dl. Patients without liver cirrhosis ("no") have a slightly narrower distribution, with total protein levels ranging from approximately 4.5 g/dl to 8 g/dl. This make it a good feature to include

AL.Phosphatase	It is a numeric column that represents the phosphate levels.	Yes	Both of these features had the highest importance score of 0.1 and 0.2 which makes them a good feature to be taken to predict the target.
USG Abdomen	It is an object column that states whether a person has diffused liver or not		

Type of alcohol consumed Gender Direct MCH MCHC Obesity Family history of cirrhosis/ hereditary TCH LDL HDL MCV Total Count Monocytes Basophils (%) SGOT/AST	Combination of numerical and categorical columns representing lifestyle,lab results taken.	No	All of these features either had negligible importance score or were highly inefficient . Thescores would range from 0.00 – 0.003 which makes them highly inefficient to predict the target. Hence they were removed

SGPT/ALT RBC Quantity of alcohol consumption Eosinophils TG Hepatitis B infection Hepatitis C infection			
Duration of alcohol consumption Total Bilirubin	Both these are numerical which depict lab results	No	Both of them had a very high score which made the model completely biased. The model only took these two rows without giving importance to any other features. Hence these were dropped.

Model Development Phase

Date	21th June 2024
Team ID	LTVIP2025TMID38009
Project Title	Revolutionizing Liver Care : Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. 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:

Using SVM to test the model

Splitting the data into Train and Test

```
[695] from sklearn.model_selection import train_test_split, cross_val_score
```

```
[696] X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)
```

Since the outcome is highly skewed we oversample the data

```
[697] from imblearn.over_sampling import RandomOverSampler
      os=RandomOverSampler(random_state=0)
      X_resampled, y_resampled = os.fit_resample(X_train, y_train)
```

```
88] model = svm.SVC()
   model.fit(X_resampled, y_resampled)
   y_pred = model.predict(X_test)
   print("Test Accuracy:", accuracy_score(y_test, y_pred))

   from sklearn.metrics import confusion_matrix, classification_report

   confusion_matrix = confusion_matrix(y_test, y_pred)

   print("Confusion Matrix:")
   print(confusion_matrix)

   classification_report = classification_report(y_test, y_pred)

   print("Classification Report:")
   print(classification_report)
```

```
Test Accuracy: 0.902834008097166
```

Using Logistic Regression to test the model

```
from sklearn.metrics import confusion_matrix, classification_report

model = LogisticRegression(penalty="l1",C=0.01,solver="liblinear")
model.fit(X_resampled, y_resampled)


y_pred = model.predict(X_test)

print("Test Accuracy:", accuracy_score(y_test, y_pred))
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy
SUPPORT VECTOR MACHINE	Classification Report: precision recall f1-score support 0 0.72 0.97 0.82 58 1 0.99 0.88 0.93 189 accuracy 0.90 247 macro avg 0.85 0.92 0.88 247 weighted avg 0.92 0.90 0.91 247	Test Accuracy: 0.902834008097166
Model 2	Screenshot of the classification report	Accuracy Value
LOGISTIC REGRESSION	Classification Report: precision recall f1-score support 0 0.85 0.97 0.90 58 1 0.99 0.95 0.97 189 accuracy 0.95 247 macro avg 0.92 0.96 0.94 247 weighted avg 0.96 0.95 0.95 247	Test Accuracy: 0.951417004048583 Confusion Matrix:

MODEL 1 CONFUSION MATRIX	MODEL 2 CONFUSION MATRIX

Confusion Matrix: [[56 2] [18 171]]	Confusion Matrix: [[56 2] [10 179]]

Model Development Phase

Date	23th June 2025
Team ID	LTVIP2025TMID38009
Project Title	Revolutionizing Liver Care : Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques
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)

SUPPORT VECTOR MACHINE	This type of model uses decision boundaries (Hyperplanes) to classify the target variable. This is useful for binary classification.	Default Parameters	Test Accuracy: 0.902834008097166 F1-score : 0 0.82 1 0.93 Recall: 0 0.97 1 0.88

Model 2	Brief description	Hyperparameters used	Performance metric value
LOGISTIC REGRESSION	This type of model uses probability / sigmoid curve to classify binary target variables. This is done using sigmoid curves	max_iter=1000, penalty="l1", solver="liblinear", C=0.01	Test Accuracy: 0.951417004048583 F1-score : 0 0.90 1 0.97 Recall: 0 0.97 1 0.95
Model 3	Brief description	Hyperparameters used	Performance metric value

DECISION TREE CLASSIFIER	Uses entropy to make decisions and provide classifications	criterion="entropy", max_depth=3, min_samples_leaf=300	Test Accuracy: 0.9757085020242915 Recall: 0.9682539682539683 F1 Score: 0.9838709677419354