Data Collection and Preprocessing Phase

Date	30/06/2025
Team ID	LTVIP2025TMID36055
Project Title	Revolutionizing Liver care : Predicting Liver cirrhosis using Advanced machine learning Techniques
Maximum Marks	6 Marks

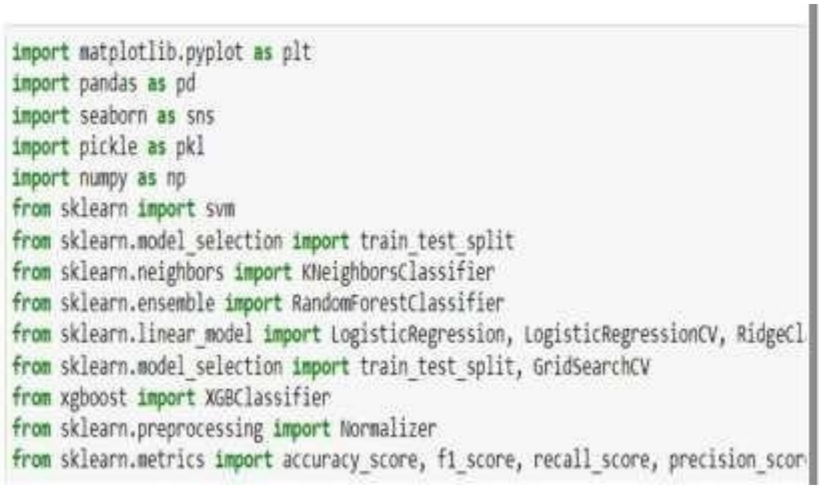
Data Exploration and Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

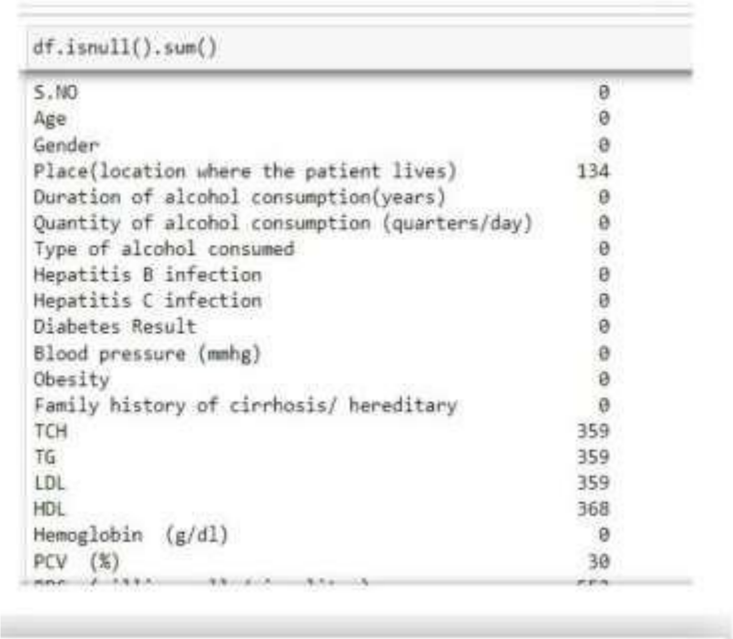
Section	Description
Data Overview	There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc. In this project we have used .csv data
Data Preparation	These are the general steps of pre-processing the data before using it for machine learning
Handling missing values	We use Handling missing values For checking the null values
Handling categorical data	As we can see our dataset has categorical data we must convert the categorical data to integer encoding or binary encoding

Handling Outliers in Data	With the help of boxplot, outliers are visualized. And here we are going to find upper bound and lower bound of numerical features with some mathematical formula

Data Preprocessing Code Screenshots	
Collect the dataset	Please refer to the link given below to download the dataset. link: [liver cirrhosis prediction (kaggle.com)](kaggle.com)

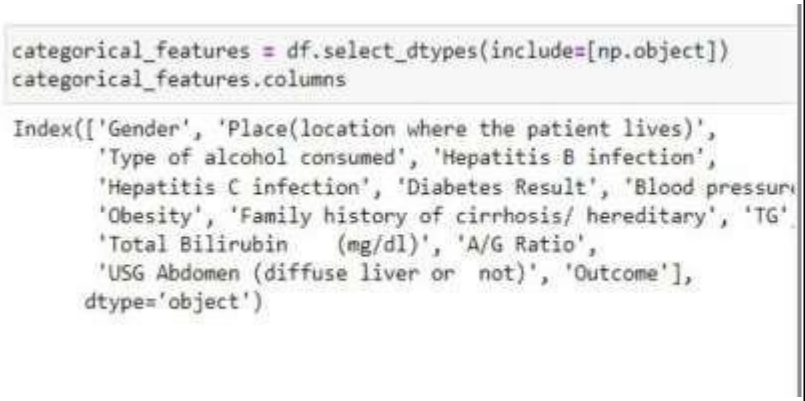
Importing the libraries	
	`import matplotlib.pyplot as plt`
	`import pandas as pd`
	`import seaborn as sns`
	`import pickle as pkl`
	`import numpy as np`
	`from sklearn import svm`
	`from sklearn.model_selection import train_test_split`
	`from sklearn.neighbors import KNeighborsClassifier`
	`from sklearn.ensemble import RandomForestClassifier`
	`from sklearn.linear_model import LogisticRegression, LogisticRegressionCV, RidgeCl`
	`from sklearn.model_selection import train_test_split, GridSearchCV`
	`from xgboost import XGBClassifier`
	`from sklearn.preprocessing import Normalizer`
	`from sklearn.metrics import accuracy_score, f1_score, recall_score, precision_scor`

Loading Data	We use the code `df=pd.read_csv("/content/HealthCare.csv")` For reading the dataset

<table>
<tr>
<td>Handling missing values</td>
<td>


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

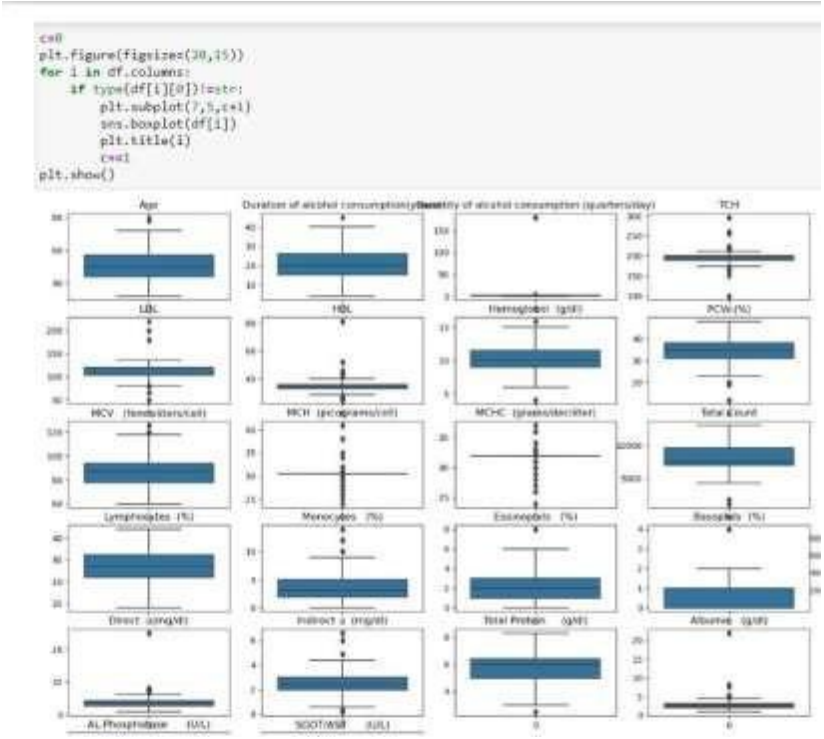
```
S.NO                                                0
Age                                                 0
Gender                                              0
Place(location where the patient lives)           134
Duration of alcohol consumption(years)              0
Quantity of alcohol consumption (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
```

</td>
</tr>
<tr>
<td>Handling Categorical values</td>
<td>


```
categorical_features = df.select_dtypes(include=[np.object])
categorical_features.columns
```

```
Index(['Gender', 'Place(location where the patient lives)',
       'Type of alcohol consumed', 'Hepatitis B infection',
       'Hepatitis C infection', 'Diabetes Result', 'Blood pressur
       'Obesity', 'Family history of cirrhosis/ hereditary', 'TG'
       'Total Bilirubin    (mg/dl)', 'A/G Ratio',
       'USG Abdomen (diffuse liver or  not)', 'Outcome'],
      dtype='object')
```

</td>
</tr>
</table>

Handling Outliers	

```
c=0
plt.figure(figsize=(30,15))
for i in df.columns:
    if type(df[i][0])!=str:
        plt.subplot(7,5,c+1)
        sns.boxplot(df[i])
        plt.title(i)
        c+=1
plt.show()
```