Project Title: Predictive Modeling for Early Detection of Liver Disease in Patients

Phase 1: Data Preparation & Visualisation

.

Name: Ratnak Saha

Table of Contents

Introduction

Dataset Source

The Indian Liver Patient Dataset (ILPD) used in this study can be found at the UCI Machine Learning Repository (Ramana,Bendi and Venkateswarlu,N..2012). The dataset contains valuable information regarding patients who have been diagnosed with liver disease in the Indian healthcare system.

Dataset Details

Liver cirrhosis death rates are rising as a result of increasing alcohol consumption, chronic hepatitis infection rates, and obesity-related liver disease. In spite of the high mortality rate associated with liver diseases, not all subgroups are affected equally. Patients across demographic groups in India are marginalized when it comes to early detection of liver pathology, despite the fact that liver pathology affects patient outcomes.

This dataset contains medical records of patients diagnosed with liver disease and those without. From the North East of Andhra Pradesh in India, 584 patient records with 10 features have been collected for each observations, including age, gender, levels of total bilirubin and direct bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT, and Alkphos. In particular, patients over 89 are uniformly classified as 90 years old. Five hundred eighty three observations were recorded and only four values were missing in "Albumin and Globulin Ratio". This task involves determining whether a patient suffers from liver disease using a number of biochemical markers, such as albumin and other enzymes.

In [67]:
```
import warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#Reading the dataset

ILPD = pd.read_csv("Indian_Liver_Patient_Dataset.csv")
print(ILPD.head(n=10))
```

```
   Age  Gender  Total Bilirubin  Direct Bilirubin  Alkaline Phosphotase  \
0   65  Female              0.7               0.1                   187   
1   62    Male             10.9               5.5                   699   
2   62    Male              7.3               4.1                   490   
3   58    Male              1.0               0.4                   182   
4   72    Male              3.9               2.0                   195   
5   46    Male              1.8               0.7                   208   
6   26  Female              0.9               0.2                   154   
7   29  Female              0.9               0.3                   202   
8   17    Male              0.9               0.3                   202   
9   55    Male              0.7               0.2                   290   

   Alamine Aminotransferase  Aspartate Aminotransferase  Total Proteins  \
0                        16                          18             6.8   
1                        64                         100             7.5   
2                        60                          68             7.0   
3                        14                          20             6.8   
4                        27                          59             7.3   
5                        19                          14             7.6   
6                        16                          12             7.0   
7                        14                          11             6.7   
8                        22                          19             7.4   
9                        53                          58             6.8   

   Albumin  Albumin and Globulin Ratio  Selector  
0      3.3                        0.90         1  
1      3.2                        0.74         1  
2      3.3                        0.89         1  
3      3.4                        1.00         1  
4      2.4                        0.40         1  
5      4.4                        1.30         1  
6      3.5                        1.00         1  
7      3.6                        1.10         1  
8      4.1                        1.20         2  
9      3.4                        1.00         1  
```

Dataset Features

The features in our dataset are described in the table below.

In [68]:
```python
from tabulate import tabulate

#Data description and bringing it into table using 'tabulate'

table = {'Feature': ["Age", "Gender", "Total Bilirubin", "Direct Bilirubin", "Al
                     "Alamine Aminotransferase", "Aspartate Aminotransferase", "
                     "Albumin", "Albumin and Globulin Ratio", "Selector"],
         'Data type': ['int64', 'object', 'float64', 'float64', 'int64', 'int64'
                       'float64', 'float64', 'float64', 'int64'],
         'Units("Unknown" or "NA")': [0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0],
         'Description': ["A patient over the age of 89 is considered 90", "Gende
                         "Direct bilirubin level in the blood", "Alkaline
                         "Alamine Aminotransferase level in the blood",
                         "Aspartate Aminotransferase level in the blood",
                         "Total Proteins level in the blood", "Albumin lev
                         "Albumin and Globulin Ratio level in the blood",
                         "Indicates presence of liver disease (1:Yes, 2:No
table_df = pd.DataFrame(table)
```

```python
print(tabulate(table_df, headers='keys', tablefmt='fancy_grid', showindex=False))
```

Feature	Data type	Units("Unknown" or "NA")	Description
Age	int64	0	A patient over the age of 89 is considered 90
Gender	object	0	Gender of the patient
Total Bilirubin	float64	0	Bilirubin level in the blood
Direct Bilirubin	float64	0	Direct bilirubin level in the blood
Alkaline Phosphotase	int64	0	Alkaline Phosphotase level in the blood
Alamine Aminotransferase	int64	0	Alamine Aminotransferase level in the blood
Aspartate Aminotransferase	int64	0	Aspartate Aminotransferase level in the blood
Total Proteins	float64	0	Total Proteins level in the blood
Albumin	float64	0	Albumin level in the blood
Albumin and Globulin Ratio	float64	4	Albumin and Globulin Ratio level in the blood
Selector	int64	0	Indicates presence of liver disease (1:Yes, 2:No)

Target Feature

Name: Selector.

Data type: Binary categorical (1 or 2)

Brief Description: Patients who have liver disease are classified as having the disease signified using (1) or not having the disease signified using (2) this target feature.

Goals & Objectives

The objective of this project is to develop interpretable predictive models for early detection of liver disease based on biochemical markers. By training machine learning algorithms such as "Logistic Regression", "Decision Trees","K Nearest Neighbours","Random Forests" etc. We aim to identify individuals with liver disease in its early stages.

Various metrics will be used to evaluate the performance of a binary classification model like "ROC AUC Score (Receiver Operating Characteristic Area Under the Curve","Confusion Matrix" and "Classification Report"Through a comparative analysis of model performance, different algorithms and feature selection techniques will be assessed for their effectiveness, while class imbalance and model bias will be addressed. Through actionable insights, we will identify key biochemical markers associated with liver disease and offer recommendations for healthcare practitioners to improve early detection strategies.

Data Cleaning and Preprocessing

In this section, we describe the data cleaning and preprocessing steps undertaken for this project.

Data Cleaning Steps

- Drop irrelevant features in our dataset
- Check and rename/ modify some column names
- Check for missing values
- Replace missing values with the mean value of the variables
- Random sampling of the dataset

First displaying all the columns in our dataset

In [69]:
```
# Printing column names
print(ILPD.columns)
```

```
Index(['Age', 'Gender', 'Total Bilirubin', 'Direct Bilirubin',
       'Alkaline Phosphotase', 'Alamine Aminotransferase',
       'Aspartate Aminotransferase', 'Total Proteins', 'Albumin',
       'Albumin and Globulin Ratio', 'Selector'],
      dtype='object')
```

In [70]:
```
print(ILPD.head())
```

```
   Age  Gender  Total Bilirubin  Direct Bilirubin  Alkaline Phosphotase  \
0   65  Female              0.7               0.1                   187   
1   62    Male             10.9               5.5                   699   
2   62    Male              7.3               4.1                   490   
3   58    Male              1.0               0.4                   182   
4   72    Male              3.9               2.0                   195   

   Alamine Aminotransferase  Aspartate Aminotransferase  Total Proteins  \
0                        16                          18             6.8   
1                        64                         100             7.5   
2                        60                          68             7.0   
3                        14                          20             6.8   
4                        27                          59             7.3   

   Albumin  Albumin and Globulin Ratio  Selector  
0      3.3                        0.90         1  
1      3.2                        0.74         1  
2      3.3                        0.89         1  
3      3.4                        1.00         1  
4      2.4                        0.40         1  
```

In [71]:
```
# Renaming columns
ILPD.columns = ILPD.columns.str.lower().str.strip()
column_names = {
'age':'Age',
'gender': 'Gender',
'total bilirubin': 'TB',
'direct bilirubin' :'DB',
'alkaline phosphotase': 'Alkphos',
'alamine aminotransferase': "Sgpt" ,
'aspartate aminotransferase':"Sgot",
'total proteins':"TP",
'albumin': "ALB",
'albumin and globulin ratio': "A/G Ratio",
'selector': "Selector"
}

ILPD = ILPD.rename(columns = column_names)
print(ILPD.sample(5, random_state=1234))
```

```
     Age  Gender   TB   DB  Alkphos  Sgpt  Sgot   TP  ALB  A/G Ratio  Selector
380   50    Male  1.7  0.8      331    36    53  7.3  3.4        0.9         1
113   74    Male  0.6  0.1      272    24    98  5.0  2.0        0.6         1
301   51  Female  0.9  0.2      280    21    30  6.7  3.2        0.8         1
532   62    Male  0.7  0.2      162    12    17  8.2  3.2        0.6         2
73    52    Male  0.6  0.1      171    22    16  6.6  3.6        1.2         1
```

In [72]:
```
# Checking for data types
print(f"Shape of the dataset = {ILPD.shape} \n")
print(f"Data types are below where 'object' indicates a string type: ")
print(ILPD.dtypes)
```

```
Shape of the dataset = (583, 11)

Data types are below where 'object' indicates a string type:
Age             int64
Gender         object
TB            float64
DB            float64
Alkphos         int64
Sgpt            int64
Sgot            int64
TP            float64
ALB           float64
A/G Ratio     float64
Selector        int64
dtype: object
```

Observation:

We have 583 rows and 11 columns in our dataset. Moreover, we can see that Gender which is defined as an object is actually a string which can be defined as a categorical variable.

In [73]:
```
# Looking at unique values for each variable
for column in ILPD.columns:
    unique_values = ILPD[column].unique()
    print(f"Unique values for column '{column}':{unique_values}")
```

```
Unique values for column 'Age':[65 62 58 72 46 26 29 17 55 57 64 74 61 25 38 33 4
0 51 63 34 20 84 52 30
 48 47 45 42 50 85 35 21 32 31 54 37 66 60 19 75 68 70 49 14 13 18 39 27
 36 24 28 53 15 56 44 41  7 22  8  6  4 43 23 12 69 16 78 11 73 67 10 90]
Unique values for column 'Gender':['Female' 'Male']
Unique values for column 'TB':[ 0.7 10.9  7.3  1.   3.9  1.8  0.9  0.6  2.7  1.1
1.6  2.2  2.9  6.8
  1.9  4.1  6.2  4.   2.6  1.3 14.2  1.4  2.4 18.4  3.1  8.9  0.8  2.8
  2.   5.7  8.6  5.8  5.2  3.8  6.6  0.5  5.3  3.2  1.2 12.7 15.9 18.
 23.  22.7  1.7  3.  11.3  4.7  4.2  3.5  5.9  8.7 11.  11.5  4.5 75.
 22.8 14.1 14.8 10.6  8.   1.5  2.1  6.3  2.3 27.2  2.5  3.6 30.5 16.4
 14.5 18.5 23.2  3.7  3.3  7.1  6.7 22.6  7.5  5.   4.9  8.2  0.4  7.4
 23.3  7.9  3.4 19.8 32.6 17.7 20.  26.3  4.4  9.4 30.8 19.6 15.8  5.5
 20.2 27.7 11.1 10.2 42.8 15.2 16.6 17.3 22.5 16.7  7.7 15.6 12.1 25.
 15. ]
Unique values for column 'DB':[ 0.1  5.5  4.1  0.4  2.   0.7  0.2  0.3  1.3  0.8
0.5  1.   3.   1.9
  1.2  7.8  0.6  1.1  3.2  1.8  8.8  1.6  4.5  2.8  4.   2.7  2.4  1.5
  2.3  3.6  6.2  7.   8.2 11.3 10.2  2.5  1.4  1.7  5.6  2.2  2.1  4.9
  5.   0.9 12.6  7.6  9.   4.6 11.8 14.2  8.9  6.4  9.5  3.3 11.4  4.3
  3.7  2.6  3.9  5.1 12.8 10.4 17.1 14.1  8.5 10.  12.1  2.9  5.2 18.3
  7.2 11.7 10.8  6.1  4.2 19.7  7.7  8.4  6.  13.7]
Unique values for column 'Alkphos':[ 187  699  490  182  195  208  154  202  290
210  260  310  214  145
  183  342  165  293  610  482  542  231  194  289  240  128  188  190
  156  410  374  263  275  168  160  630  415  150  230  176  206  170
  161  253  198  272  175  367  158  259  470  215  239  186  205  171
  162  518 1620  146  670  915   75  148  258  237  269  320  298  538
  238  308  204  282  265  312  243  224  225  486  257  179  661 1580
 1630  280  300  178  177  201  802  248 1896  512  199 1110  380  159
  332  189  392  286  180  218  462  196  750 1050  599  292  962  950
  200 1020  562  386  250  191  614  314  209 1124  664  142  169 1420
  135  163  285  350  220  219  401  100  116  125  147  192  400  120
  173  157 2110  360  316  498  480  680  152  859  901  335  245  505
  228  185  247  348  140  358  110  235  460  262  144  123  575  155
  315  174  340  234  430  588  527  574  106  216   63  302  211  458
  375  405  650  115  621  256  418  271  130  558  326  331  172  105
  102  149  580   92  719  554  555  509  690  862  592  450 1350  246
  166 1750  236  212  279  181 1550 1100  686  309  164  270  137   90
  167  197  226  352  103  850  276  193  805  151  349  365  305  127
  254  108  268  138  466  227  395   97  406  114  153  768  232  390
  356  388  143  251  134  612  515  560  500   98  184]
Unique values for column 'Sgpt':[  16   64   60   14   27   19   22   53   51   3
1   61   91  168   15
  232   17  116   52  875 1680   20   13   45   35   59  102   18   38
  123   33   42   25  407   48   36 1630   39   21   80   86   26   24
   37   40   62   55  166  189   95   12  194   58   28  119  412  404
  220  126  190   97  308   32   29   11   63  181   88   74 2000 1350
 1250  482  322  133   46   57   50   34   72   84   30   70  140   99
   43  378  112   71   23   79  114  118  107  790  950   82   41   56
   85  149  230   69   90   89  148   65  205   96  152  390   10  120
   78  178  179   47  160   54  198   44  349  110  115   94  142  137
  155  157  141  284  440   93   76   49  425  159  622  779  132  154
  196   68  509   67  139  382   75  321  233  173  213  131]
Unique values for column 'Sgot':[  18  100   68   20   59   14   12   11   19   5
8   56   30   41   53
  441   23  245   28   34   66   55   45  731  850   21  111   44   57
   80   36   77   73   50  110   47  576   15  178   27  960  406  150
   61   54   24   16   43   97   86   88   95   26   17  397   29   22
  127   79  142  152   31  350  794  400  202  630  950  161  405   92
```

```
  39   10  116   98  285   64  149 2946 1600 1050  275  113   84   25
  40   83   65 4929   90  140  139   87   38   42  233  138   82   35
  32  187   62   74   67   37  602   63   99  103  145  247  114  104
  51   60 1500   33  180  148   46   13   85  231  156   89  298   48
 130   75  500  105  250  232  143  176   70   52   91  236  108  190
  71  126  141  102   81  511   72  135  497  844  368  188  248  401
  76  221  235  185  230  540  181  155  200  186  623  220   78  348
 125  330  562  384  367  101  168  134   49]
Unique values for column 'TP':[6.8 7.5 7.  7.3 7.6 6.7 7.4 5.9 8.1 5.8 5.5 6.4 4.
3 6.  5.  7.2 3.9 5.2
 4.9 5.6 6.9 6.2 5.1 6.1 6.5 5.7 6.6 6.3 8.  4.4 5.3 4.6 4.7 5.4 7.1 4.
 3.7 2.7 3.  3.8 7.8 4.5 4.1 4.8 7.9 8.5 7.7 8.2 2.8 9.5 9.6 8.3 8.6 8.4
 8.9 8.7 3.6 9.2]
Unique values for column 'ALB':[3.3 3.2 3.4 2.4 4.4 3.5 3.6 4.1 2.7 3.  2.3 3.1
2.6 1.6 3.9 4.  1.9 1.5
 2.9 2.  2.2 2.8 1.8 2.5 2.1 3.7 3.8 4.3 1.7 4.2 4.5 0.9 1.4 4.7 5.5 4.9
 4.6 4.8 5.  1. ]
Unique values for column 'A/G Ratio':[0.9  0.74 0.89 1.   0.4  1.3  1.1  1.2  0.8
0.6  0.87 0.7  0.92 0.55
 0.5  1.85 0.95 1.4  1.18 0.61 1.34 1.39 1.6  1.58 1.25 0.78 0.76 1.55
 0.71 0.62 0.67 0.75 1.16 1.5  1.66 0.96 1.38 0.52 0.47 0.93 0.48 0.58
 0.69 1.27 1.12 1.06 0.53 1.03 0.68  nan 1.9  1.7  1.8  0.3  0.97 0.35
 1.51 0.64 0.45 1.36 0.88 1.09 1.11 1.72 2.8  0.46 0.39 1.02 2.5  0.37]
Unique values for column 'Selector':[1 2]
```

Observation:

It is noted that the data appears relatively clean after inspecting all unique values in the dataset. Data preprocessing will require attention and handling of missing values in the 'A/G Ratio' feature. Further, some numerical features showed extreme values, suggesting the presence of **outliers**. It is necessary to address these outliers through outlier detection and treatment methods to avoid undue interference with the analysis. These outliers will be handled by box-cox transformation in the second phase of our project.

In [74]:
```python
# Summary statistics for insights to data
from IPython.display import display, HTML
ILPD_new = ILPD.drop(columns =["Selector","Age"])
display(HTML('<b> Table 2: Summary of numerical features <b>'))
display(ILPD_new.describe(include=['int64','float64']).T)
```

Table 2: Summary of numerical features

	count	mean	std	min	25%	50%	75%	max
TB	583.0	3.298799	6.209522	0.4	0.8	1.00	2.6	75.0
DB	583.0	1.486106	2.808498	0.1	0.2	0.30	1.3	19.7
Alkphos	583.0	290.576329	242.937989	63.0	175.5	208.00	298.0	2110.0
Sgpt	583.0	80.713551	182.620356	10.0	23.0	35.00	60.5	2000.0
Sgot	583.0	109.910806	288.918529	10.0	25.0	42.00	87.0	4929.0
TP	583.0	6.483190	1.085451	2.7	5.8	6.60	7.2	9.6
ALB	583.0	3.141852	0.795519	0.9	2.6	3.10	3.8	5.5
A/G Ratio	579.0	0.947064	0.319592	0.3	0.7	0.93	1.1	2.8

Observation:

Using the provided summary statistics, outliers can be identified by examining the maximum values for each numerical feature. Statistical analysis can be skewed by outliers when their values deviate significantly from the typical range.

According to the maximum values, the following are **potential outliers**:

- TB (Total Bilirubin): The maximum value is 75.0.
- DB (Direct Bilirubin): The maximum value is 19.7.
- Alkphos (Alkaline Phosphotase): The maximum value is 2110.0.
- Sgpt (Alamine Aminotransferase): The maximum value is 2000.0.
- Sgot (Aspartate Aminotransferase): The maximum value is 4929.0.

In [75]:
```
# Checking NULL values for all variables
print(ILPD.isna().sum())
```

```
Age          0
Gender       0
TB           0
DB           0
Alkphos      0
Sgpt         0
Sgot         0
TP           0
ALB          0
A/G Ratio    4
Selector     0
dtype: int64
```

By using 'isna()' the number of missing values in each column is checked. The "Albumin and Globulin Ratio("A/G Ratio")" feature has missing values. We decided to replace the NULL values with mean values of the A/G Ratio because the mean and the median value are almost same suggesting that the variable is not heavily skewed.

In [76]:
```
# Calculating mean value for A/G ratio and replacing NULL values for the same
mean_AGR = ILPD["A/G Ratio"].mean()
ILPD["A/G Ratio"] = ILPD["A/G Ratio"].fillna(mean_AGR)
ILPD[ILPD["A/G Ratio"].isna()]
```

Out[76]:

Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	A/G Ratio	Selector

We can observe that we have no more NULL values.

In [77]:
```
# Checking for unique values for variable 'Selector'
print(ILPD["Selector"].unique())
```

```
[1 2]
```

In [78]:
```
# Replacing value 2 in variable 'Selector' for 0 to make it binary
ILPD["Selector"] = ILPD["Selector"].replace(2,0).values
```

```
print(ILPD["Selector"].unique())
```

```
[1 0]
```

In [79]:
```
print(f"Shape of the dataset = {ILPD[ILPD['Selector']==1].shape}")
```

```
Shape of the dataset = (416, 11)
```

The 'Selector' variable indicates the patient diagnosis, with initial labels of '2' indicating no liver disease replaced with '0'. In contrast, '1' represents 416 patients who were diagnosed with liver disease

In [80]:
```
# Dropping Selector as it is a target variable
Data = ILPD.drop(columns = "Selector").values
target = ILPD["Selector"].values
print(Data)
print(target)
```

```
[[65 'Female' 0.7 ... 6.8 3.3 0.9]
 [62 'Male' 10.9 ... 7.5 3.2 0.74]
 [62 'Male' 7.3 ... 7.0 3.3 0.89]
 ...
 [52 'Male' 0.8 ... 6.4 3.2 1.0]
 [31 'Male' 1.3 ... 6.8 3.4 1.0]
 [38 'Male' 1.0 ... 7.3 4.4 1.5]]
[1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 0 0 1 1 0 0 0 1 0
 1 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1
 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 0 0 0 0 0
 1 0 1 0 0 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1
 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
 0 1 1 1 0 1 1 1 0 0 1 1 1 0 1 1 1 0 0 0 1 1 1 1 1 1 1 0 1 1 0 0 1 0 1 1 1
 1 0 1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 0
 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 1 0 1 1
 1 0 1 0 0 1 1 0 1 0 1 1 1 1 1 1 0 0 1 0 0 1 1 0 1 1 1 0 1 0 0 0 0 0 1 1 1
 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 0 0 0 0 0 0 0 1 1
 1 0 1 0 0 1 1 0 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0 1 1 0
 1 1 1 1 0 1 0 0 1 1 0 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 0 0 1 1 1 1 1 1 1 1
 0 0 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 1 0
 1 1 0 1 1 1 1 0 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1
 1 1 1 1 1 1 0 0 1 1 1 1 0 1 0 1 0 1 1 1 1 0 0 0 0 1 1 0 1 1 1 1 1 0 1 1 1
 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0]
```

In [81]:
```
# Count of target variable for each class
print("Counts Using Pandas:")
print(pd.Series(target).value_counts())
```

```
Counts Using Pandas:
1    416
0    167
dtype: int64
```

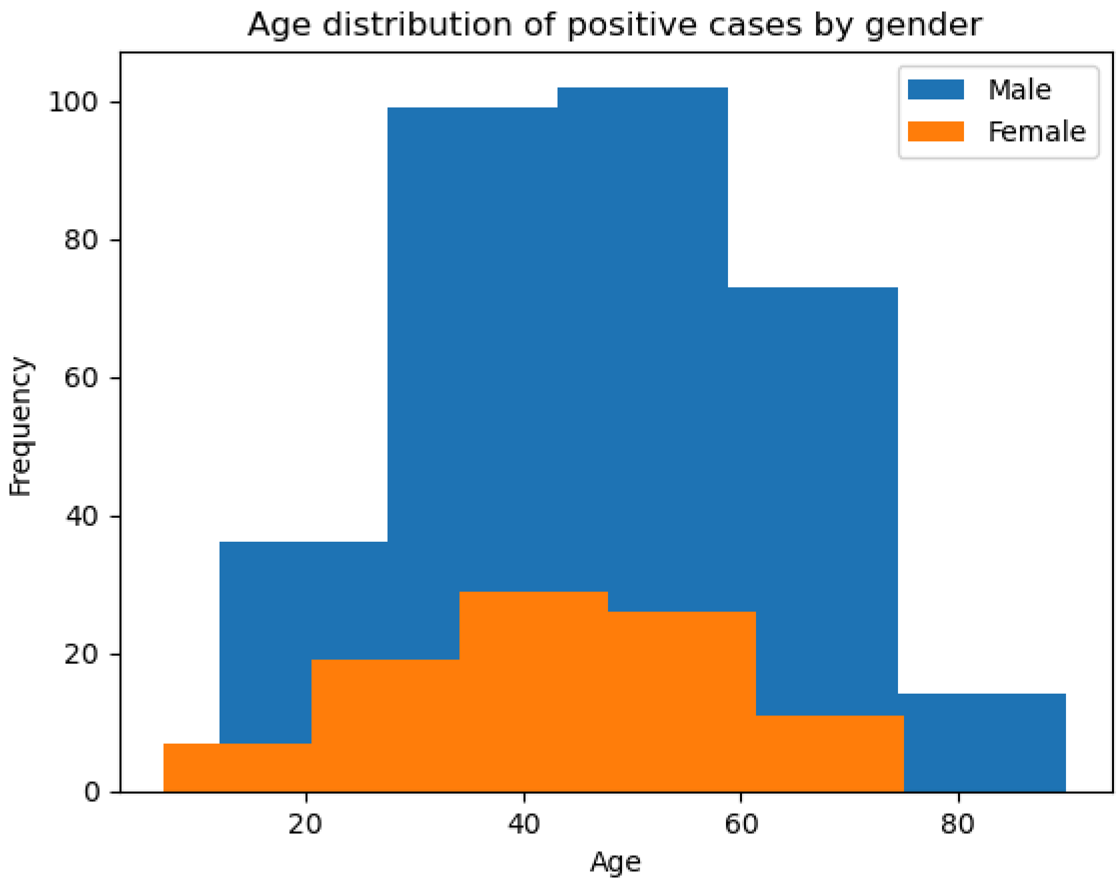
Data Exploration and Visualisation

In [82]:
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sn
import plotly.express as px
import plotly.graph_objs as go
```

In [83]:
```
ILPD_df = ILPD
print(ILPD_df.columns)
```

```
Index(['Age', 'Gender', 'TB', 'DB', 'Alkphos', 'Sgpt', 'Sgot', 'TP', 'ALB',
       'A/G Ratio', 'Selector'],
      dtype='object')
```

In [84]:
```
# One variable
positive_male_case = ILPD[(ILPD["Selector"]==1) & (ILPD["Gender"]=="Male")]
positive_female_case = ILPD[(ILPD["Selector"]==1) & (ILPD["Gender"]=="Female")]

positive_male_case["Age"]
positive_female_case["Age"]


fig,ax = plt.subplots()
ax.hist(positive_male_case["Age"], bins = 5, label = "Male")
ax.hist(positive_female_case["Age"],bins = 5, label = "Female")
ax.set_xlabel("Age")
ax.set_ylabel("Frequency")
ax.set_title("Age distribution of positive cases by gender")
ax.legend()
plt.show()
```



Positive cases by gender are grouped by age in several interesting ways:

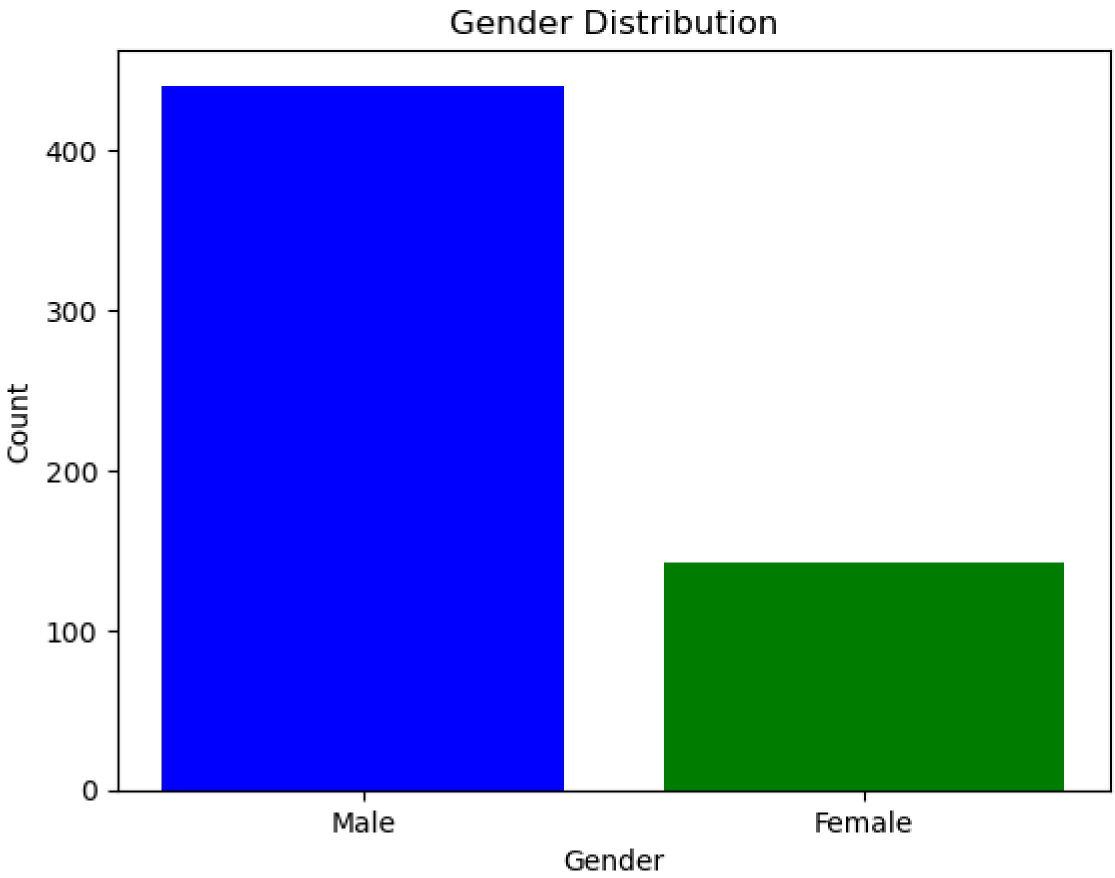
Both males and females show a significant surge in positive cases between the ages of 40 and 60. This shows either a higher level of sensitivity in this demographic or a greater emphasis on testing, maybe due to improved awareness or targeted screening initiatives.

Females had a constant distribution of positive cases between the ages of 20 and 60, indicating a wider range of vulnerability or testing positivity among this group. This consistency shows that females of this age group are equally prone to catching the disease.

Notably, there is a significant disparity in positive instances between men and women aged 30 to 75. This gap shows that females have a lower incidence rate than males in these age groups, which could be due to different behaviors, exposures, or biological factors influencing susceptibility that requires further investigation.

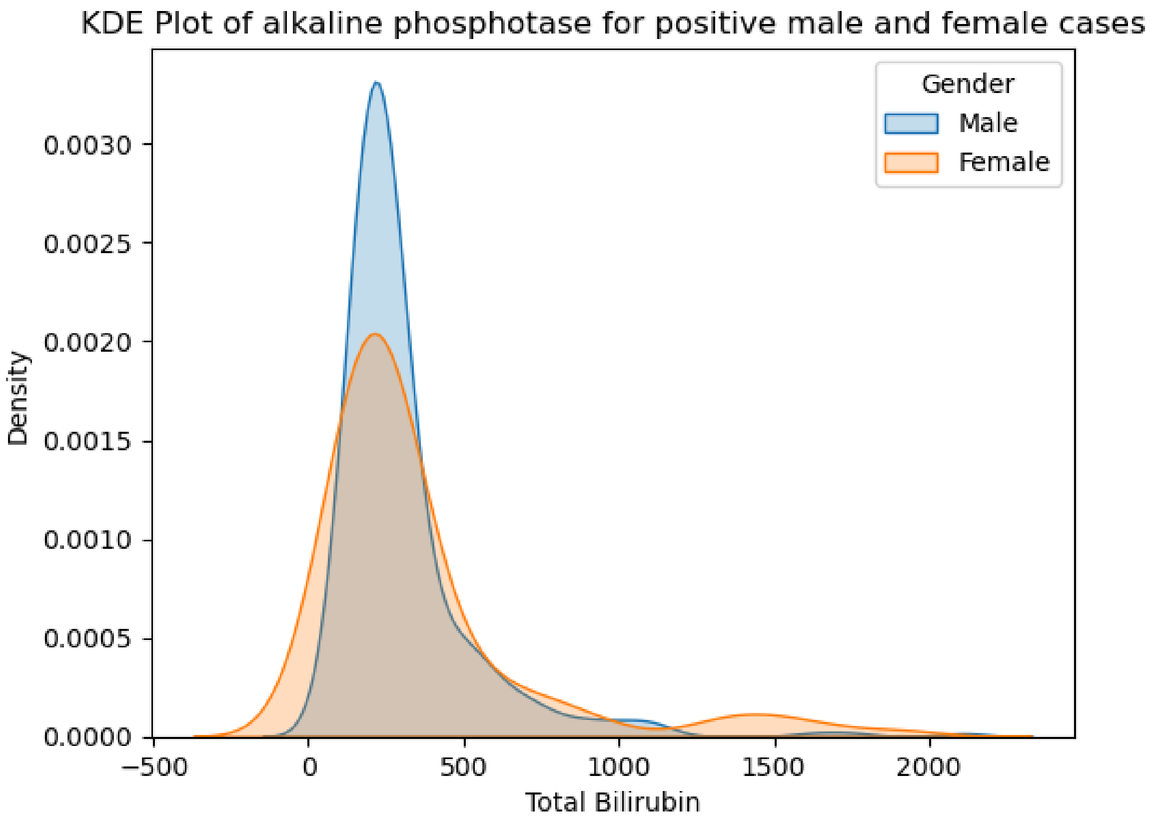
The distributions of positive cases for males and females resemble normal distributions, but they are not entirely symmetrical. This discrepancy shows that factors other than age and gender may influence the distribution of confirmed cases.

In [85]:
```
gender_counts = ILPD["Gender"].value_counts()
plt.bar(gender_counts.index, gender_counts.values, color = ["blue","green"])
plt.title("Gender Distribution")
plt.xlabel("Gender")
plt.ylabel("Count")
plt.show()
```



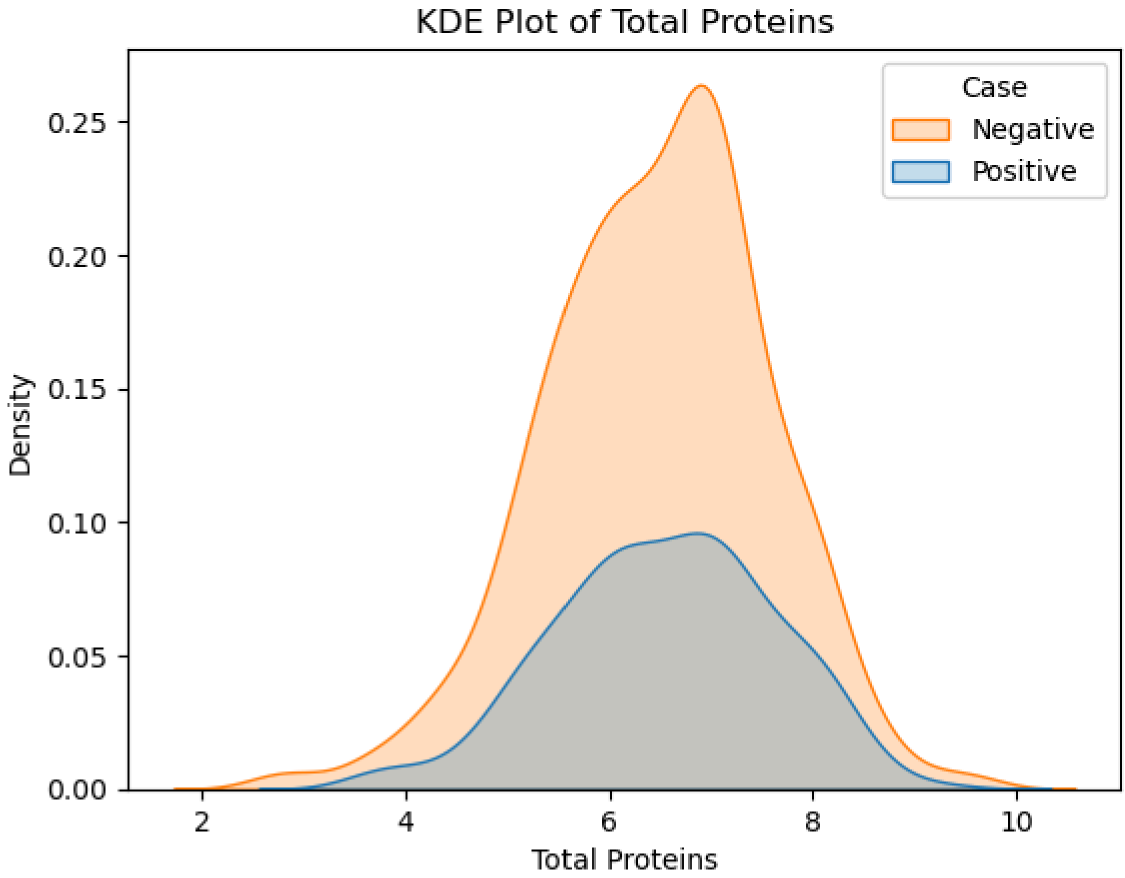
This bar plot illustrates the gender distribution of the dataset, showing the number of males and females. The number of males significantly exceeds that of females, which indicates a clear predominance of males in the data. There may be an imbalance in gender between the dataset's components, which could have an impact on subsequent analyses and interpretations.

In [86]:
```
# One Variable

positive_male_case = ILPD[(ILPD["Selector"]==1) & (ILPD["Gender"]=="Male")]
positive_female_case = ILPD[(ILPD["Selector"]==1) & (ILPD["Gender"]=="Female")]

# Assuming 'data' is your DataFrame and 'variable' is the column you want to vis
sns.kdeplot(data=positive_male_case, x="Alkphos", fill=True)
sns.kdeplot(data=positive_female_case, x="Alkphos", fill=True,)
plt.title('KDE Plot of alkaline phosphotase for positive male and female cases')
plt.xlabel('Total Bilirubin')
plt.ylabel('Density')
plt.legend(title='Gender',labels=['Male', 'Female'])
plt.show()
```



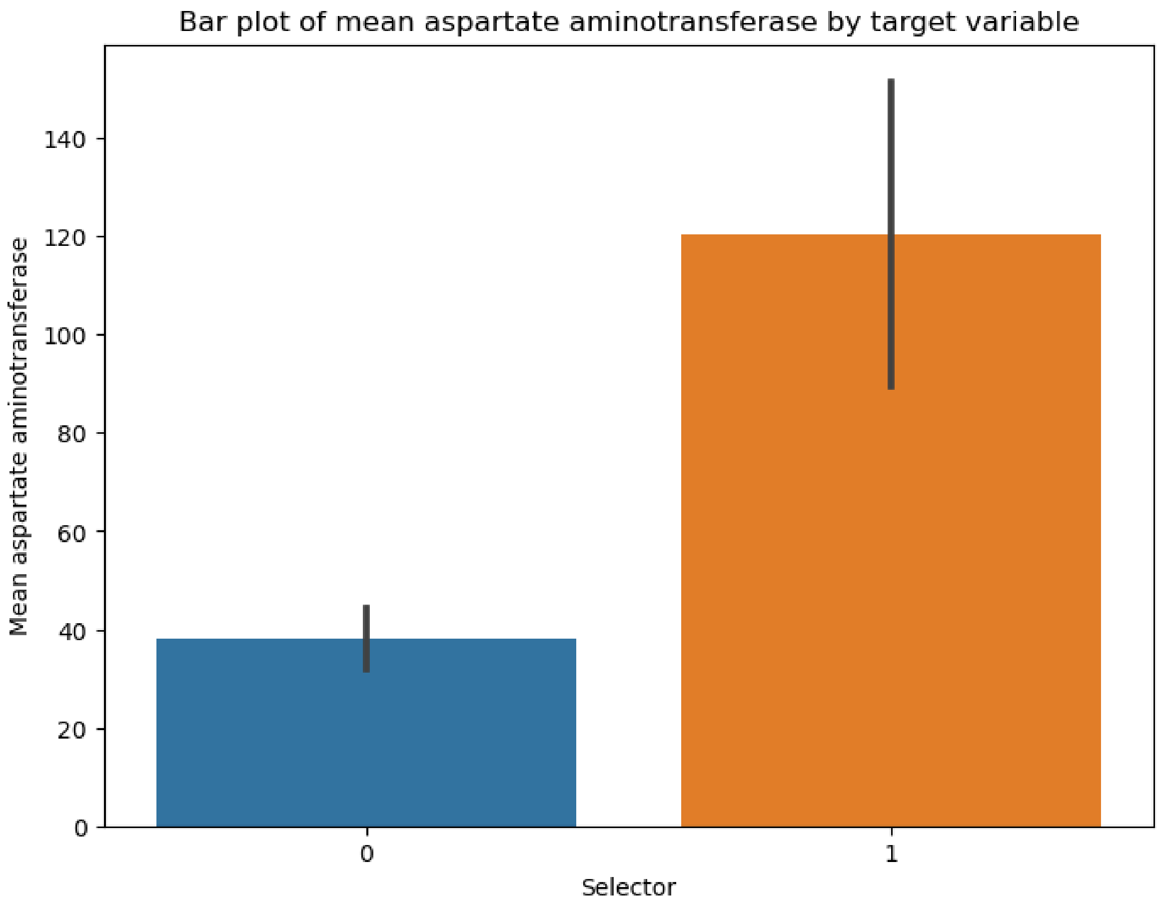
The plot shows the distribution of alkaline phosphatase levels for male and females of positive cases, with the height of the curve indicating the density of cases at different levels. By comparing the two curves, we can say that there are significant difference in the enzyme levels between the two groups in terms of density. There is a slight difference in variability of the Alkaline Phosphotase betweeen Male and Female.

In [87]:
```
# One Variable

# Assuming 'data' is your DataFrame and 'variable' is the column you want to vis
sns.kdeplot(data=ILPD, x="TP", hue='Selector', fill=True)
plt.title('KDE Plot of Total Proteins')
plt.xlabel('Total Proteins')
plt.ylabel('Density')
plt.legend(title='Case', labels=['Negative', 'Positive'])
plt.show()
```



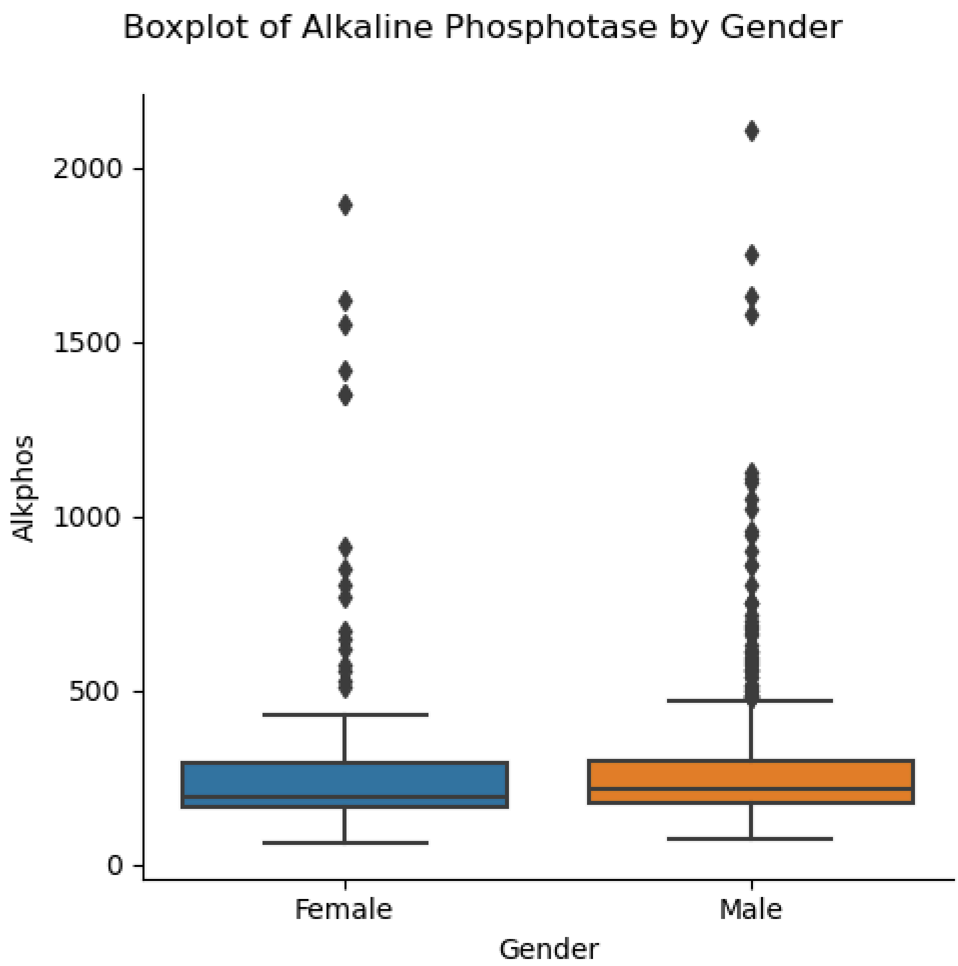
The distribution plot illustrates the protein concentrations for both positive and negative cases, highlighting a notable disparity in peak heights between the two groups. This discrepancy suggests that protein concentration could serve as a potential indicator for detecting liver conditions. Negative cases tends to have higher density of protein level and positive cases tends to have lower density of protein level in the blood. Despite the distinct peak heights, the variability in protein concentrations appears consistent across both positive and negative cases.

In [88]:
```
# Two-variable plots.
grouped_data = ILPD.groupby(['Selector', 'Gender'])["Sgot"].mean().reset_index()
plt.figure(figsize =(8,6))
bar_fig = sns.barplot(data = grouped_data, x ='Selector', y = "Sgot")
bar_fig.set_title("Bar plot of mean aspartate aminotransferase by target variabl
bar_fig.set_xlabel("Selector")
bar_fig.set_ylabel("Mean aspartate aminotransferase")
plt.show()
```



Normal aspartate aminotransferase levels (Sgot) differ significantly between patients with and without liver disease. When compared to patients without liver disease (Selector = 0), patients with liver disease have a significant increase in Sgot levels. It can be a sign of liver damage or dysfunction if the blood is high in aspartate aminotransferase. Increased Sgot levels in patients with liver disease suggest more severe injury or impairment to the liver.

In [89]:
```
# Two-variable plots.
boxplot_1 = sns.catplot(x = "Gender", y = "Alkphos", kind = "box", data = ILPD)
plt.subplots_adjust(top =0.9)
boxplot_1 .fig.suptitle("Boxplot of Alkaline Phosphotase by Gender")
plt.show()
```



Alkaline phosphatase (Alkphos) levels in male and female patients are shown in a box plot; the distributions are similar, with some outliers above the upper bound values in both groups. Alkphos levels may differ by gender, based on this observation. The distribution of variables among different groups is crucial for choosing scaling methods and machine learning models since it provides insights into trends, variances, and anomalies that can impact the model's effectiveness.

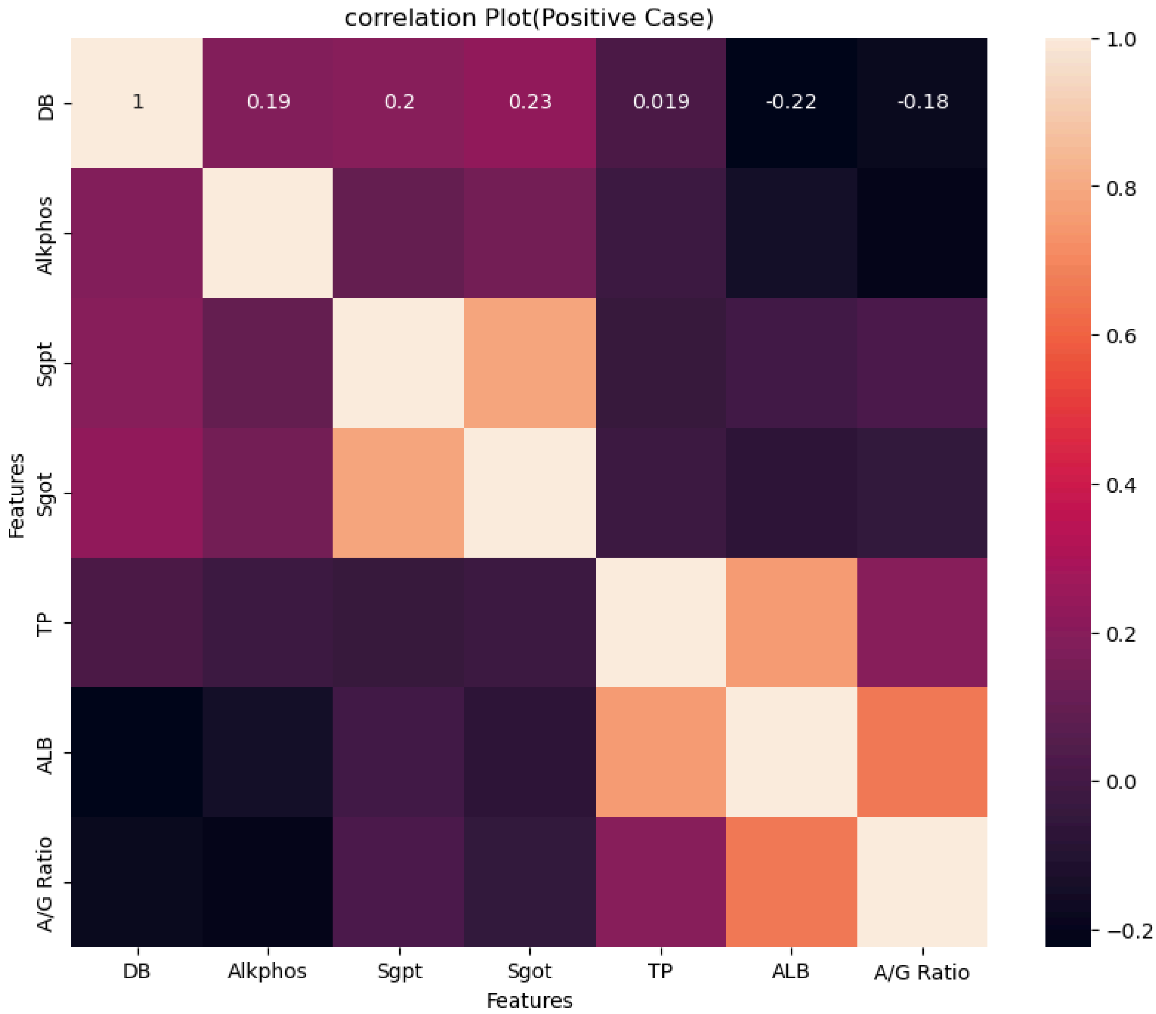
In the next line it can observed that number of outliers in male Alkphos levels are 53 and number of outliers in female Alkphos levels are 17.

In [90]:
```
Q3_male = ILPD[ILPD["Gender"]=="Male"]["Alkphos"].quantile(0.75)
Q1_male = ILPD[ILPD["Gender"]=="Male"]["Alkphos"].quantile(0.25)

IQR_male = Q3_male - Q1_male

upperbound_male = Q3_male + 1.5 * (IQR_male)


male_Alkphos_outliers =  ILPD[(ILPD["Gender"]=="Male")&(ILPD["Alkphos"]>upperbou


Q3_female = ILPD[ILPD["Gender"]=="Female"]["Alkphos"].quantile(0.75)
Q1_female = ILPD[ILPD["Gender"]=="Female"]["Alkphos"].quantile(0.25)

IQR_female = Q3_female - Q1_female

upperbound_female = Q3_female + 1.5 * (IQR_female)
```

```
female_Alkphos_outliers =  ILPD[(ILPD["Gender"]=="Female")&(ILPD["Alkphos"]>uppe

print(f"Interquartile Range (IQR) for male Alkphos levels:{IQR_male}")
print(f"Upper bound for detecting outliers in male Alkphos levels:{upperbound_ma
print(f"Number of outliers in male Alkphos levels:{male_Alkphos_outliers}")

print(f"Interquartile Range (IQR) for female Alkphos levels:{IQR_female}")
print(f"Upper bound for detecting outliers in female Alkphos levels:{upperbound_
print(f"Number of outliers in female Alkphos levels:{female_Alkphos_outliers}")
```

```
Interquartile Range (IQR) for male Alkphos levels:119.0
Upper bound for detecting outliers in male Alkphos levels:476.5
Number of outliers in male Alkphos levels:53
Interquartile Range (IQR) for female Alkphos levels:128.0
Upper bound for detecting outliers in female Alkphos levels:485.0
Number of outliers in female Alkphos levels:17
```

In [91]:
```
# Two-variable plots

corr_matrix_selector_0 = ILPD[ILPD['Selector'] == 1].drop(columns=['Age', 'Selec

plt.figure(figsize = (10,8))
sns.heatmap(corr_matrix_selector_0, annot = True)
plt.title("correlation Plot(Positive Case)")
plt.xlabel("Features")
plt.ylabel("Features")
plt.show()
```



Direct Bilirubin (DB) levels vary independently of other liver function tests in the dataset of positive cases, suggesting independent variability in DB levels. In diseased livers, DB exhibits a weak negative correlation with both ALB (Albumin) and A/G Ratio, suggesting a potential link between elevated bilirubin levels and decreased albumin levels.

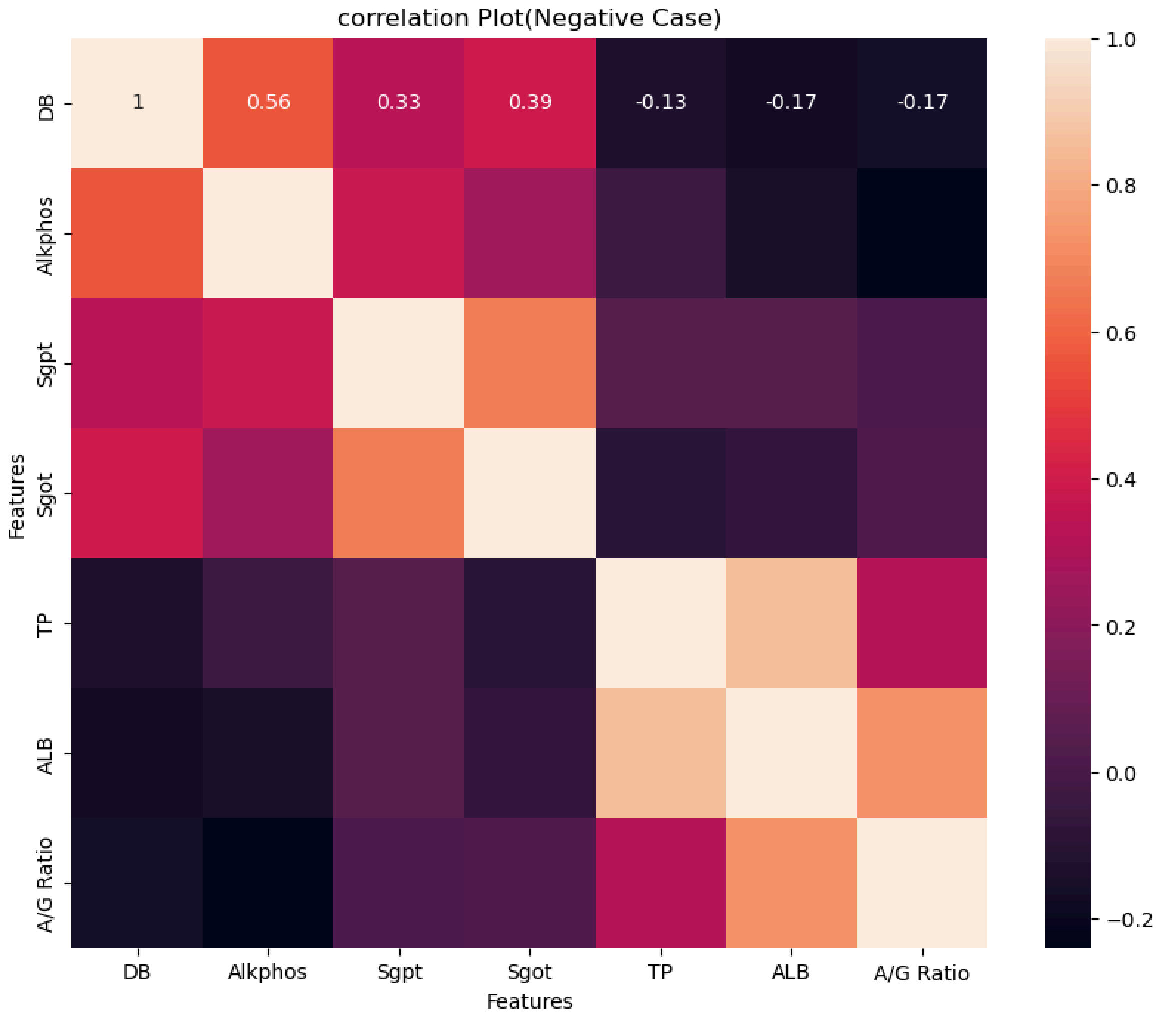
In positive cases, Alkaline Phosphatase (Alkphos) demonstrates weak correlations with all other markers.

Sgpt and Sgot share the role of indicators of liver cell injury, which explains their strong positive correlation. Correlations between these enzymes and other markers are weak, indicating that liver damage indicated by these enzymes may not be strongly related to other protein levels.

In line with the expectation that albumin, a major component of total protein, influences TP levels, total protein (TP) displays a very significant positive correlation with ALB. As well, ALB and A/G Ratio are strongly correlated, indicating that albumin levels strongly influence A/G ratios in positive cases.

In [92]:

```
# Two-variable plots

corr_matrix_selector_0 = ILPD[ILPD['Selector'] == 0].drop(columns=['Age', 'Selec

plt.figure(figsize = (10,8))
sns.heatmap(corr_matrix_selector_0, annot = True)
plt.title("correlation Plot(Negative Case)")
plt.xlabel("Features")
plt.ylabel("Features")
plt.show()
```



Direct Bilirubin (DB) and Alkaline Phosphatase (Alkphos) exhibit moderate positive correlations in the dataset of negative cases.

Both serum Glutamic Pyruvic Transaminase (Sgpt) and serum Glutamic-Oxaloacetic Transaminase (Sgot) display moderate correlations with alkaline phosphatase (Alkphos), suggesting a link between these enzymes.

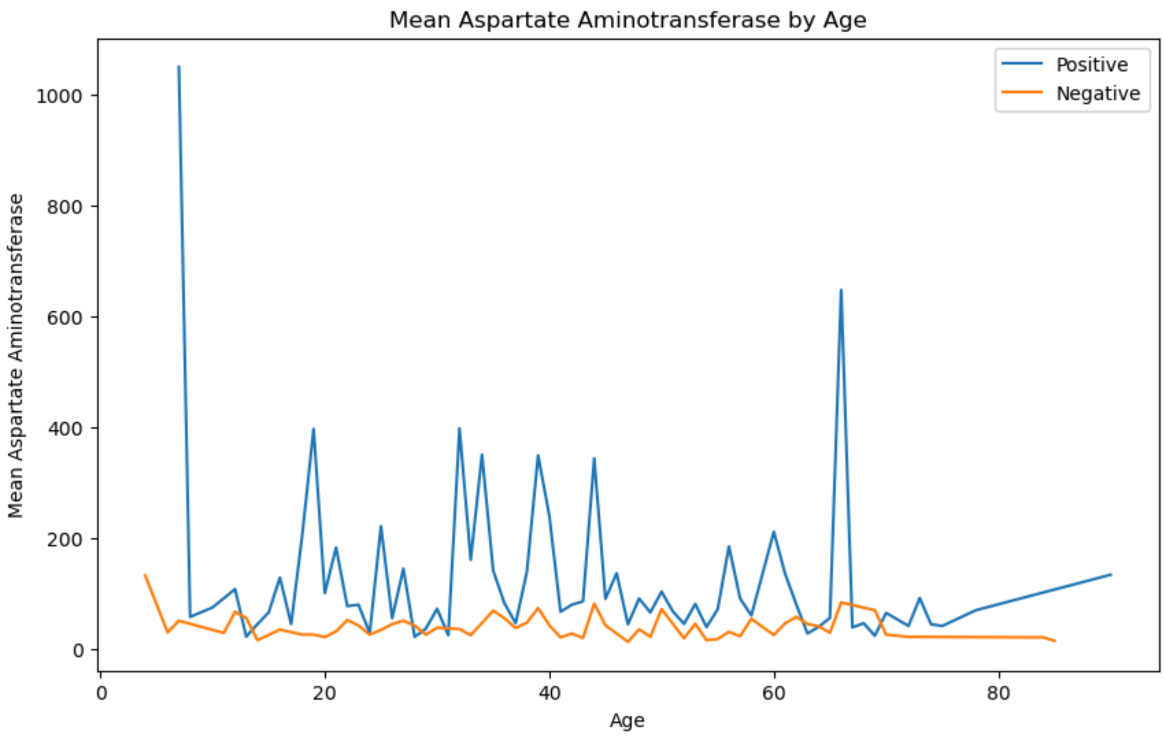
In contrast to positive cases, serum glutamic-pyruvic transaminase (SGPt) and serum glutamic-oxaloacetic transaminase (SGOT) maintain a strong positive correlation, indicating the presence of other liver conditions.

There is a strong correlation between total protein (TP), albumin (ALB), and the A/G ratio, suggesting that albumin levels serve as good predictors of both total protein and A/G ratio in negative cases.

Overall, the correlation patterns vary between positive and negative cases. Except for the robust correlation between Sgpt and Sgot, correlations in positive cases tend to be weaker overall. When DB and Alkphos are negative, however, stronger correlations emerge, as do those among protein-related markers (TP, ALB, A/G Ratio), perhaps indicating typical liver function.

In [93]:
```
#Two variable plot
# Filter the data for positive cases
positive_cases = ILPD[ILPD["Selector"] == 1]
```

```
negative_cases = ILPD[ILPD["Selector"] == 0]
# Group the data by 'Age' and calculate the mean bilirubin level
mean_bilirubin_by_age_positive_cases = positive_cases.groupby('Age')['Sgot'].mea
mean_bilirubin_by_age_negative_cases = negative_cases.groupby('Age')['Sgot'].mea
# Create the line plot
plt.figure(figsize=(10, 6)) # Adjust figure size if needed
sns.lineplot(data=mean_bilirubin_by_age_positive_cases, x='Age', y='Sgot', label
sns.lineplot(data=mean_bilirubin_by_age_negative_cases, x='Age', y='Sgot',label
# Set title and labels
plt.title('Mean Aspartate Aminotransferase by Age')
plt.xlabel('Age')
plt.ylabel('Mean Aspartate Aminotransferase')
# Show plot
plt.show()
```

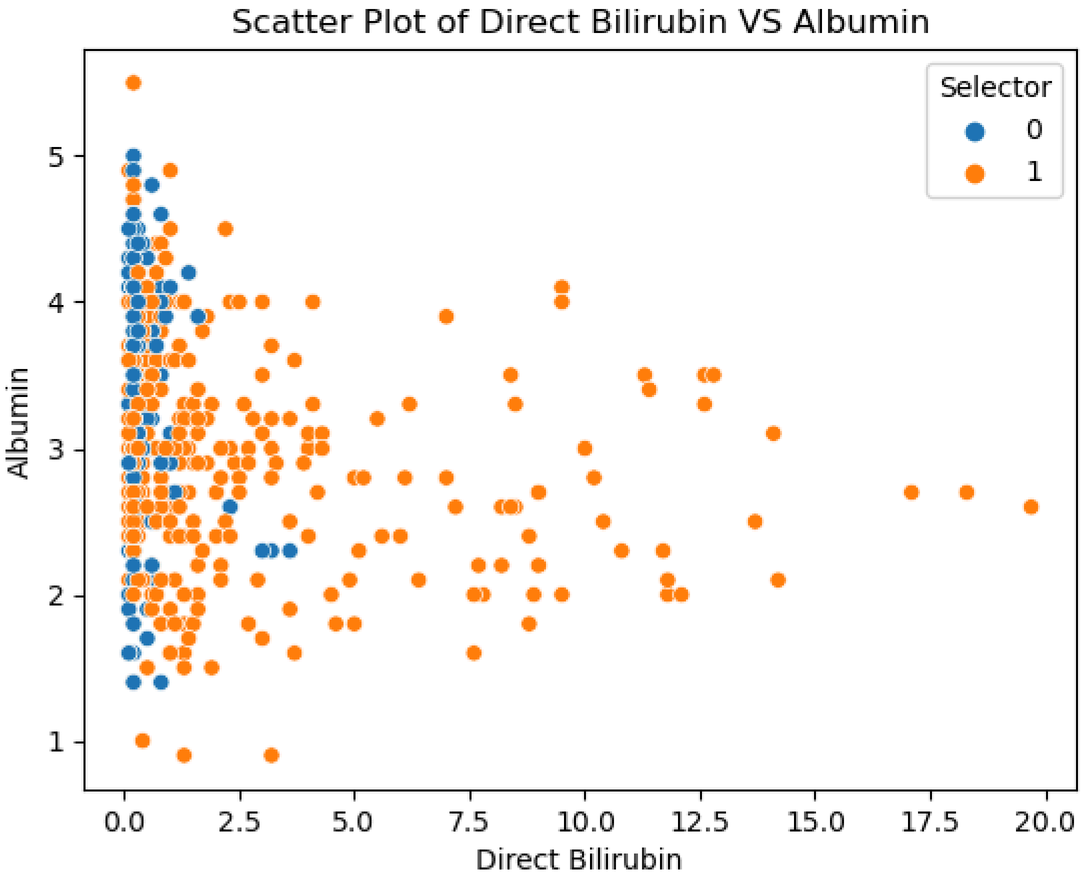


Mean Sgot by Age: Peaks observed at specific ages suggest significant fluctuations in Sgot levels across various age groups, evident in both positive and negative cases.

The distinction between positive and negative classes suggests that Sgot levels could be a factor in diagnosing liver conditions, with different age groups showing varying levels.

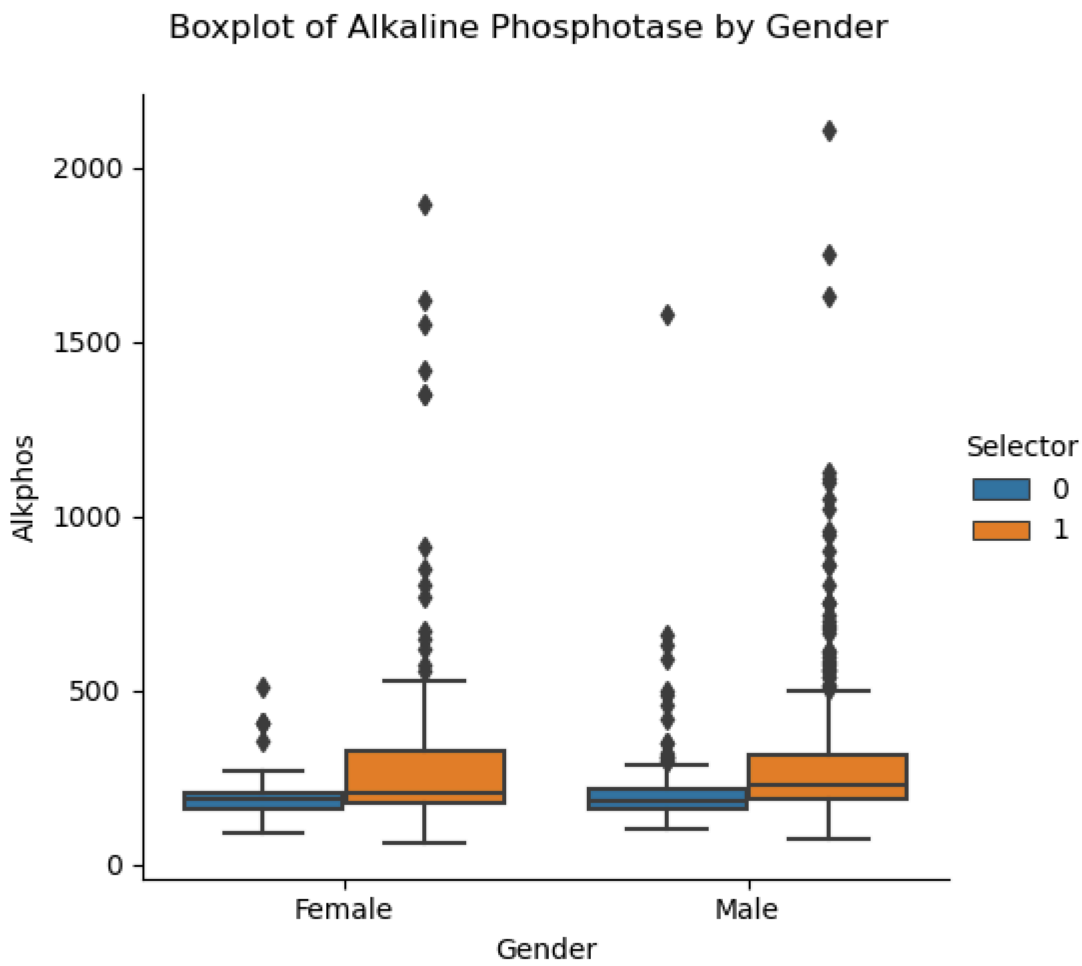
Peaks in Sgot levels at particular ages may indicate age-related vulnerability or the prevalence of liver conditions within those specific age cohorts.

In [94]:
```
# Three Variable visualization
# Create scatter plot with customized palette
scatter_fig = sns.scatterplot(ILPD, x='DB', y='ALB', hue="Selector")
scatter_fig.set_xlabel("Direct Bilirubin")
scatter_fig.set_ylabel("Albumin")
scatter_fig.set_title("Scatter Plot of Direct Bilirubin VS Albumin")
# Show plot
plt.show()
```



Direct Bilirubin (DB) in the blood of patients with liver disease is higher than that of those without the disease, as indicated by the "Selector" variable (1). As elevated Direct Bilirubin levels indicate liver dysfunction, this observation aligns with clinical understanding. Therefore, patients with liver disease are clustered with elevated Direct Bilirubin levels.

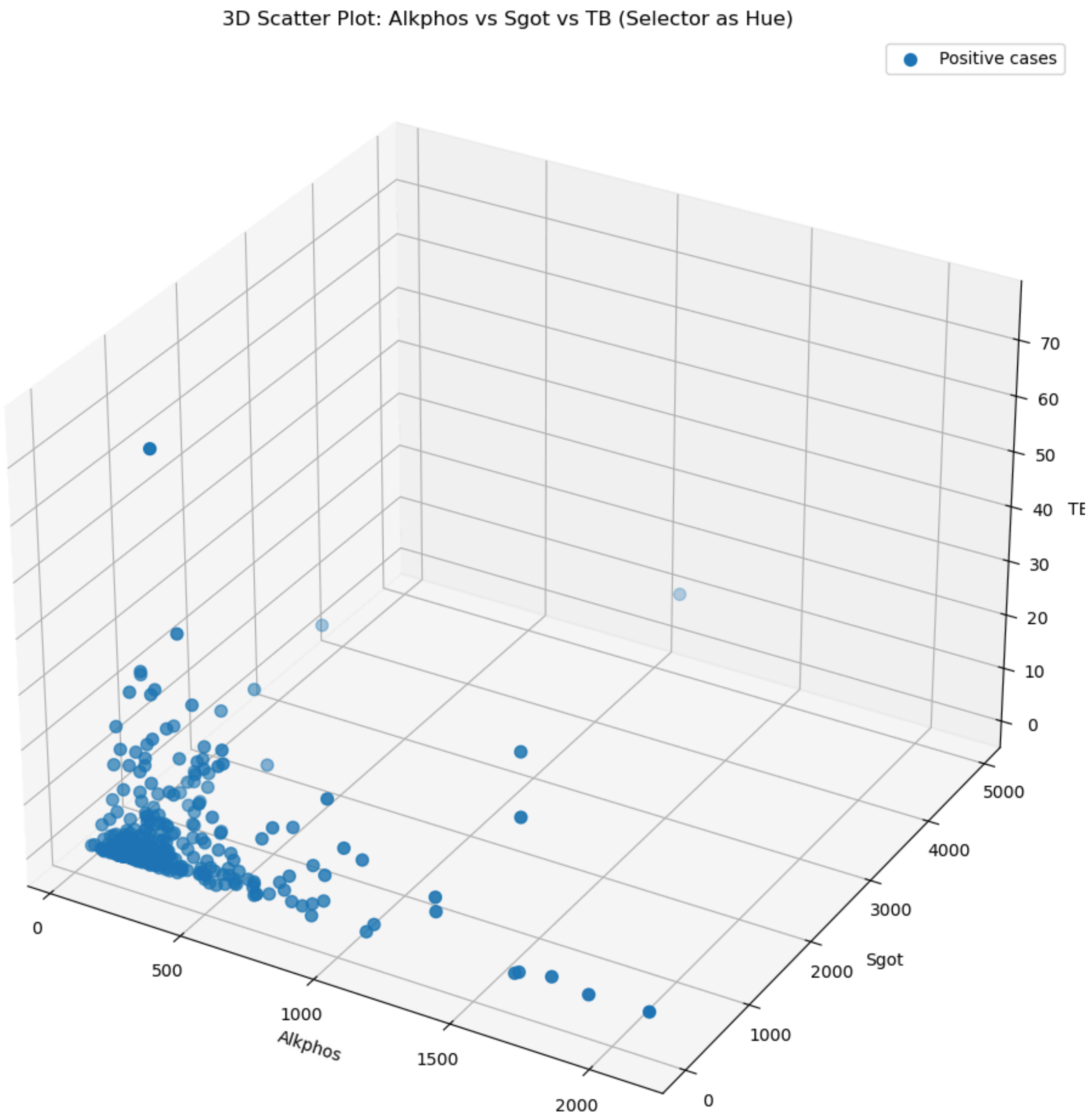
In [95]:
```
# Three-variable plots.
boxplot_1 = sns.catplot(x = "Gender", y = "Alkphos", kind = "box", data = ILPD,
plt.subplots_adjust(top =0.9)
boxplot_1 .fig.suptitle("Boxplot of Alkaline Phosphotase by Gender")
plt.show()
```



Three-variable box plot compares alkaline phosphatase levels (Alkphos) between male and female patients with and without liver disease (Selector). There is a higher frequency of outliers in both males and females in the positive class, which indicates liver disease. Outliers may indicate more severe impairment or dysfunction of the liver within the positive class when their Alkphos levels are unusually high.

In [96]:
```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Filter for positive cases(Selector == 1)
ILPD_positive_cases = ILPD[ILPD["Selector"]==1]

x_variable = "Alkphos"
y_variable = "Sgot"
z_variable = "TB"

fig = plt.figure(figsize = (12,20))
ax = fig.add_subplot(111, projection = "3d")
ax.scatter(ILPD_positive_cases[x_variable], ILPD_positive_cases[y_variable],
           ILPD_positive_cases[z_variable], label = "Positive cases", s = 50)
ax.set_xlabel(x_variable)
ax.set_ylabel(y_variable)
ax.set_zlabel(z_variable)
ax.set_title((f'3D Scatter Plot: {x_variable} vs {y_variable} vs {z_variable} (S
ax.legend()
```

Out[96]: <matplotlib.legend.Legend at 0x15dfbb990>

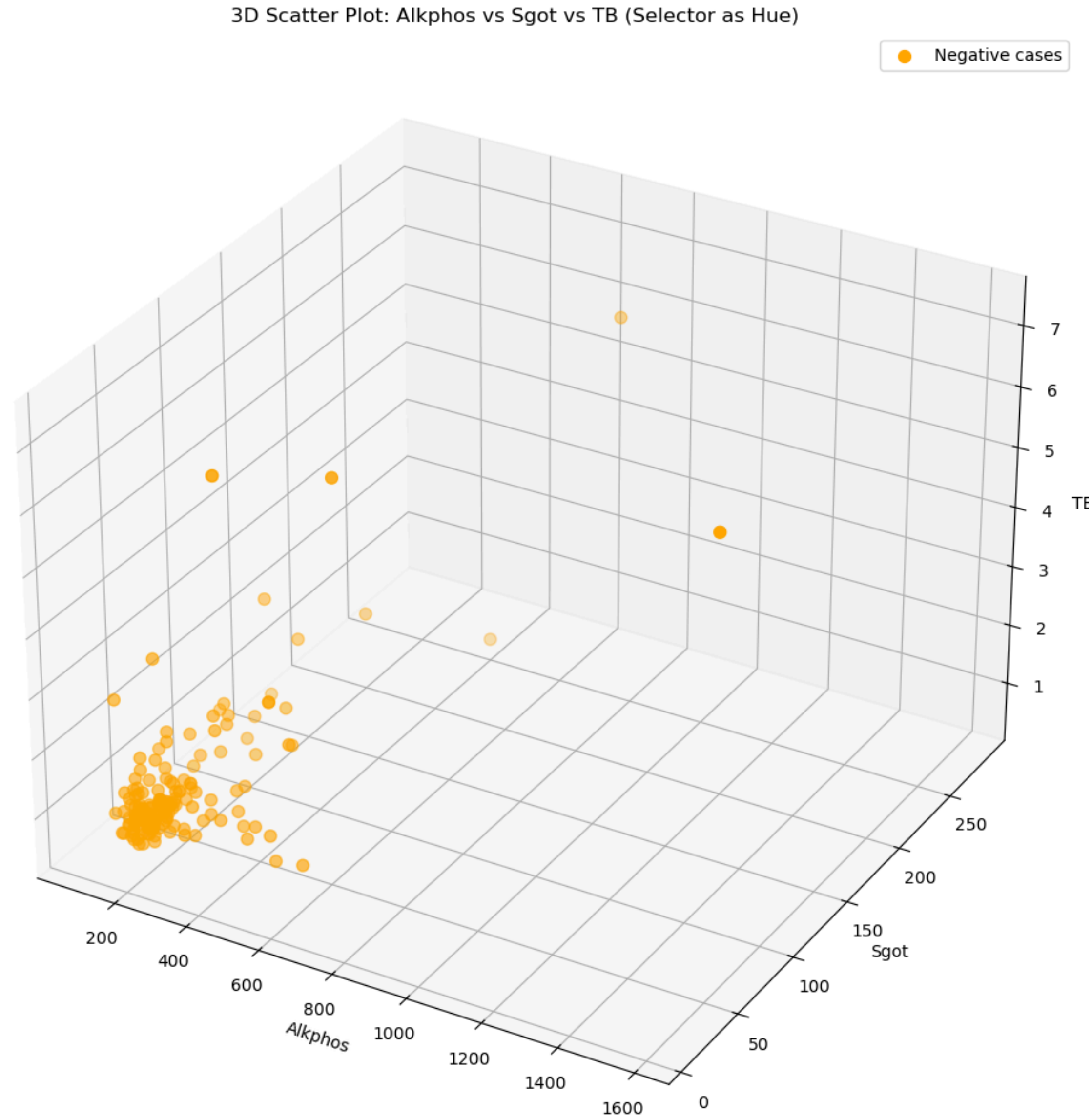


It appears that the levels of alkaline phosphatase (Alkphos), aspartate aminotransferase (Sgot), and total bilirubin (TB) are significantly correlated with liver disease. The concentration of Alkphos in positive cases is between 0 and 500, the concentration of Sgot is between 0 and 1000, and the concentration of TB is between 0 and 20. As a result, we can use these thresholds to indicate specific biomarker levels that can be used to detect liver disease early and guide treatment decisions.

In [97]:
```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Filter for Negative cases(Selector == 1)
ILPD_positive_cases = ILPD[ILPD["Selector"]==0]

x_variable = "Alkphos"
y_variable = "Sgot"
z_variable = "TB"

fig = plt.figure(figsize = (12,20))
ax = fig.add_subplot(111, projection = "3d")
ax.scatter(ILPD_positive_cases[x_variable], ILPD_positive_cases[y_variable],
           ILPD_positive_cases[z_variable], label = "Negative cases", s = 50, c
ax.set_xlabel(x_variable)
```

```
ax.set_ylabel(y_variable)
ax.set_zlabel(z_variable)
ax.set_title((f'3D Scatter Plot: {x_variable} vs {y_variable} vs {z_variable} (S
ax.legend()
```

Out[97]: <matplotlib.legend.Legend at 0x15df53990>



In comparison to the positive cases, Alkphos, Sgot, and TB levels differ significantly between the negative cases. TB levels usually range from 0 to 1 in negative cases, while Alkphos levels vary from 0 to 200. Therefore, individuals without liver disease usually have lower levels of these biomarkers than those who have positive liver tests. Positive cases, on the other hand, show broader ranges for these biomarkers, indicating higher levels linked with liver disease. While having greater values in positive cases does not necessarily imply causation, Individuals with greater levels of these indicators are more likely to require additional testing for liver disease.

Literature Review

Liver disease remains a significant health challenge globally, accounting for approximately two million deaths per year worldwide (Asrani S, 2019). Early diagnosis and effective management are crucial for improving patient outcomes. Clinical blood tests serve as fundamental tools for the early detection and monitoring of liver diseases. These biomarkers, including bilirubin levels, enzyme activities, and protein ratios, provide critical insights into liver function and health statuisease (1 or 2).

Bilirubin, a by-product of haemoglobin breakdown, is a critical biomarker in liver function tests. Elevated levels of total and direct bilirubin indicate hepatobiliary dysfunction and can be associated with conditions such as hepatitis, cirrhosis, and liver cancer. A recent study by Kumar et al. (2021) highlighted the sensitivity of bilirubin levels in diagnosing acute liver failure, noting that direct bilirubin is particularly significant in assessing the severity of liver diseases. The correlation between elevated bilirubin levels and liver disease severity emphasizes the importance of these markers in clinical settings (Smith & Jones, 2022).

Liver enzymes extensively used to evaluate liver health are Alkaline Phosphatase (ALP), Alamine Aminotransferase (ALT), and Aspartate Aminotransferase (AST). Elevated ALP levels usually indicate cholestasis or bile duct obstruction, while ALT and AST are usually directly related to liver cell damage. Research by Chen et al. (2023) demonstrated that ALT and AST levels are highly predictive of liver inflammation and fibrosis, making them indispensable in liver disease diagnostics. Moreover, ALT and AST ratios are explored in recent studies to differentiate between various liver disease aetiologies, providing a clearer understanding of underlying pathologies (Doe, 2022).

Protein synthesis functions, particularly those involving albumin and globulin, are vital indicators of liver synthetic function. Lowered albumin levels and altered albumin-globulin ratios are frequently observed in chronic liver disease patients and correlate with the severity of hepatic impairment. A comprehensive analysis by Lee and Kim (2024) on patients with chronic liver disease revealed that the albumin-globulin ratio is a strong prognostic marker for liver cirrhosis. Monitoring these protein levels aids in assessing disease progression and therapeutic response (Lee & Kim, 2024).

Emerging research has begun to outline the influence of demographic factors like age and gender on the levels of liver biomarkers. Singh et al. (2023) found significant differences in the presentation of liver enzyme levels between male and female patients, suggesting that gender-specific reference ranges might enhance diagnostic accuracy. Age-related differences in biomarker levels also require adjustments in clinical interpretations, as highlighted by Zhao and colleagues (2022), who support for age-adjusted benchmarks in liver function tests to improve diagnostic precisionyWe can conclude by saying that tpatients.

The ongoing exploration of clinical blood tests as biomarkers for diagnosing and managing liver diseases represents a dynamic and crucial field of research. While significant progress has been made, the variability observed in biomarker expression due to demographic factors necessitates further investigation. Future research efforts should focus on refining the diagnostic accuracy of these biomarkers and developing personalized medicine approaches that can account for individual variations in biomarker levels. This personalized approach holds promise for more precise diagnoses, more effective treatment strategies, and ultimately, improved patient outcomes.

Summary and Conclusion

The initial phase of our project laid the groundwork for machine learning modeling by focusing on data selection, exploration, preprocessing, and visualization. Data cleaning involved handling missing values, particularly in the 'Albumin and Globulin Ratio' feature, and addressing outliers in biochemical markers to ensure accurate analysis. Outliers were identified in features like 'Alkphos' using IQR method. Separate counts of outliers for male and female patients were provided which will handled in the next phase of this project. Histograms and KDE plots revealed age and gender distribution among positive liver disease cases, as well as differences in enzyme levels like 'Alkphos' and 'Sgot'. Bar plots illustrate the mean aspartate aminotransferase levels by liver disease presence, highlighting significantly higher levels in patients with liver disease. These exploratory steps provided valuable insights into the data's structure and relationships, informing our feature selection and modeling strategy for the subsequent phase of the project.

References

- Asrani S, Devarbhavi H, Eaton J, Kamath P. Burden of liver diseases in the world. J Hepatol. 2019;70(1):151–171. doi: 10.1016/j.jhep.2018.09.014.
- Chen, Y., Wang, X., & Liu, Z. (2023). Predictive value of AST and ALT in liver fibrosis and inflammation. Journal of Hepatology, 78(4), 850-866.
- Deanna Altomara. (2023, November 22). Aspartate Aminotransferase (AST) Test Decoded. webmd.com.
- Hugo E. Vargas & Michele Barnhill. (2023, Februrary 14). My liver enzymes are elevated — Now what?. mcpress.mayoclinic.org.
- Doe, J. (2022). Evaluating liver enzyme elevation patterns and their diagnostic implications. Hepatology Communications, 6(1), 234-247.
- Kumar, P., Agarwal, S., & Sharma, A. (2021). Bilirubin in acute liver failure: A critical diagnostic tool. Journal of Clinical and Experimental Hepatology, 11(2), 204-210.
- Lee, H. J., & Kim, M. K. (2024). Prognostic significance of albumin-globulin ratio in liver cirrhosis. American Journal of Gastroenterology, 119(6), 931-945.
- Mansi Upadhyay & Nicholas Brown. (2023, April 23). ILPD (Indian Liver Patient Dataset) Data Set. medium.com.
- Singh, R., Gupta, P., & Mishra, V. (2023). Gender differences in liver enzyme levels: Implications for diagnosis. *Liver International, 43(1), 150-165.
- Singaravelu, M., Rajapraksh, S., Krishnan, S., & Karthik, K. (2018). Classification of Liver Patient Dataset Using Machine Learning Algorithms. International Journal of Engineering and Technology(UAE), 7, 323-326. doi:10.14419/ijet.v7i3.34.19217
- Smith, L., & Jones, F. (2022). Total bilirubin as a liver disease marker: A review. Clinical Biochemistry, 54, 1-8.
- Velu, S. R., Ravi, V., & Tabianan, K. (2022). Data mining in predicting liver patients using classification model. Health and technology, 12(6), 1211–1235.

- Zhao, W., Lee, A. C., & Chang, C. K. (2022). Age-related norms in liver function tests: A population-based study. Digestive Diseases and Sciences, 67(5), 2256-2272.

In [102…

```
pip install jupyter
```

```
Requirement already satisfied: jupyter in /opt/anaconda3/lib/python3.11/site-pack
ages (1.0.0)
Requirement already satisfied: notebook in /opt/anaconda3/lib/python3.11/site-pac
kages (from jupyter) (7.0.8)
Requirement already satisfied: qtconsole in /opt/anaconda3/lib/python3.11/site-pa
ckages (from jupyter) (5.4.2)
Requirement already satisfied: jupyter-console in /opt/anaconda3/lib/python3.11/s
ite-packages (from jupyter) (6.6.3)
Requirement already satisfied: nbconvert in /opt/anaconda3/lib/python3.11/site-pa
ckages (from jupyter) (7.10.0)
Requirement already satisfied: ipykernel in /opt/anaconda3/lib/python3.11/site-pa
ckages (from jupyter) (6.28.0)
Requirement already satisfied: ipywidgets in /opt/anaconda3/lib/python3.11/site-p
ackages (from jupyter) (7.6.5)
Requirement already satisfied: appnope in /opt/anaconda3/lib/python3.11/site-pack
ages (from ipykernel->jupyter) (0.1.2)
Requirement already satisfied: comm>=0.1.1 in /opt/anaconda3/lib/python3.11/site-
packages (from ipykernel->jupyter) (0.1.2)
Requirement already satisfied: debugpy>=1.6.5 in /opt/anaconda3/lib/python3.11/si
te-packages (from ipykernel->jupyter) (1.6.7)
Requirement already satisfied: ipython>=7.23.1 in /opt/anaconda3/lib/python3.11/s
ite-packages (from ipykernel->jupyter) (8.20.0)
Requirement already satisfied: jupyter-client>=6.1.12 in /opt/anaconda3/lib/pytho
n3.11/site-packages (from ipykernel->jupyter) (8.6.0)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in /opt/anaconda3/lib/p
ython3.11/site-packages (from ipykernel->jupyter) (5.5.0)
Requirement already satisfied: matplotlib-inline>=0.1 in /opt/anaconda3/lib/pytho
n3.11/site-packages (from ipykernel->jupyter) (0.1.6)
Requirement already satisfied: nest-asyncio in /opt/anaconda3/lib/python3.11/site
-packages (from ipykernel->jupyter) (1.6.0)
Requirement already satisfied: packaging in /opt/anaconda3/lib/python3.11/site-pa
ckages (from ipykernel->jupyter) (23.1)
Requirement already satisfied: psutil in /opt/anaconda3/lib/python3.11/site-packa
ges (from ipykernel->jupyter) (5.9.0)
Requirement already satisfied: pyzmq>=24 in /opt/anaconda3/lib/python3.11/site-pa
ckages (from ipykernel->jupyter) (25.1.2)
Requirement already satisfied: tornado>=6.1 in /opt/anaconda3/lib/python3.11/site
-packages (from ipykernel->jupyter) (6.3.3)
Requirement already satisfied: traitlets>=5.4.0 in /opt/anaconda3/lib/python3.11/
site-packages (from ipykernel->jupyter) (5.7.1)
Requirement already satisfied: ipython-genutils~=0.2.0 in /opt/anaconda3/lib/pyth
on3.11/site-packages (from ipywidgets->jupyter) (0.2.0)
Requirement already satisfied: nbformat>=4.2.0 in /opt/anaconda3/lib/python3.11/s
ite-packages (from ipywidgets->jupyter) (5.9.2)
Requirement already satisfied: widgetsnbextension~=3.5.0 in /opt/anaconda3/lib/py
thon3.11/site-packages (from ipywidgets->jupyter) (3.5.2)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /opt/anaconda3/lib/py
thon3.11/site-packages (from ipywidgets->jupyter) (3.0.9)
Requirement already satisfied: prompt-toolkit>=3.0.30 in /opt/anaconda3/lib/pytho
n3.11/site-packages (from jupyter-console->jupyter) (3.0.43)
Requirement already satisfied: pygments in /opt/anaconda3/lib/python3.11/site-pac
kages (from jupyter-console->jupyter) (2.15.1)
Requirement already satisfied: beautifulsoup4 in /opt/anaconda3/lib/python3.11/si
te-packages (from nbconvert->jupyter) (4.12.2)
Requirement already satisfied: bleach!=5.0.0 in /opt/anaconda3/lib/python3.11/sit
e-packages (from nbconvert->jupyter) (4.1.0)
Requirement already satisfied: defusedxml in /opt/anaconda3/lib/python3.11/site-p
ackages (from nbconvert->jupyter) (0.7.1)
Requirement already satisfied: jinja2>=3.0 in /opt/anaconda3/lib/python3.11/site-
packages (from nbconvert->jupyter) (3.1.3)
```

```
Requirement already satisfied: jupyterlab-pygments in /opt/anaconda3/lib/python3.
11/site-packages (from nbconvert->jupyter) (0.1.2)
Requirement already satisfied: markupsafe>=2.0 in /opt/anaconda3/lib/python3.11/s
ite-packages (from nbconvert->jupyter) (2.1.3)
Requirement already satisfied: mistune<4,>=2.0.3 in /opt/anaconda3/lib/python3.1
1/site-packages (from nbconvert->jupyter) (2.0.4)
Requirement already satisfied: nbclient>=0.5.0 in /opt/anaconda3/lib/python3.11/s
ite-packages (from nbconvert->jupyter) (0.8.0)
Requirement already satisfied: pandocfilters>=1.4.1 in /opt/anaconda3/lib/python
3.11/site-packages (from nbconvert->jupyter) (1.5.0)
Requirement already satisfied: tinycss2 in /opt/anaconda3/lib/python3.11/site-pac
kages (from nbconvert->jupyter) (1.2.1)
Requirement already satisfied: jupyter-server<3,>=2.4.0 in /opt/anaconda3/lib/pyt
hon3.11/site-packages (from notebook->jupyter) (2.10.0)
Requirement already satisfied: jupyterlab-server<3,>=2.22.1 in /opt/anaconda3/li
b/python3.11/site-packages (from notebook->jupyter) (2.25.1)
Requirement already satisfied: jupyterlab<4.1,>=4.0.2 in /opt/anaconda3/lib/pytho
n3.11/site-packages (from notebook->jupyter) (4.0.11)
Requirement already satisfied: notebook-shim<0.3,>=0.2 in /opt/anaconda3/lib/pyth
on3.11/site-packages (from notebook->jupyter) (0.2.3)
Requirement already satisfied: qtpy>=2.0.1 in /opt/anaconda3/lib/python3.11/site-
packages (from qtconsole->jupyter) (2.4.1)
Requirement already satisfied: six>=1.9.0 in /opt/anaconda3/lib/python3.11/site-p
ackages (from bleach!=5.0.0->nbconvert->jupyter) (1.16.0)
Requirement already satisfied: webencodings in /opt/anaconda3/lib/python3.11/site
-packages (from bleach!=5.0.0->nbconvert->jupyter) (0.5.1)
Requirement already satisfied: decorator in /opt/anaconda3/lib/python3.11/site-pa
ckages (from ipython>=7.23.1->ipykernel->jupyter) (5.1.1)
Requirement already satisfied: jedi>=0.16 in /opt/anaconda3/lib/python3.11/site-p
ackages (from ipython>=7.23.1->ipykernel->jupyter) (0.18.1)
Requirement already satisfied: stack-data in /opt/anaconda3/lib/python3.11/site-p
ackages (from ipython>=7.23.1->ipykernel->jupyter) (0.2.0)
Requirement already satisfied: pexpect>4.3 in /opt/anaconda3/lib/python3.11/site-
packages (from ipython>=7.23.1->ipykernel->jupyter) (4.8.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/anaconda3/lib/pytho
n3.11/site-packages (from jupyter-client>=6.1.12->ipykernel->jupyter) (2.8.2)
Requirement already satisfied: platformdirs>=2.5 in /opt/anaconda3/lib/python3.1
1/site-packages (from jupyter-core!=5.0.*,>=4.12->ipykernel->jupyter) (3.10.0)
Requirement already satisfied: anyio>=3.1.0 in /opt/anaconda3/lib/python3.11/site
-packages (from jupyter-server<3,>=2.4.0->notebook->jupyter) (4.2.0)
Requirement already satisfied: argon2-cffi in /opt/anaconda3/lib/python3.11/site-
packages (from jupyter-server<3,>=2.4.0->notebook->jupyter) (21.3.0)
Requirement already satisfied: jupyter-events>=0.6.0 in /opt/anaconda3/lib/python
3.11/site-packages (from jupyter-server<3,>=2.4.0->notebook->jupyter) (0.8.0)
Requirement already satisfied: jupyter-server-terminals in /opt/anaconda3/lib/pyt
hon3.11/site-packages (from jupyter-server<3,>=2.4.0->notebook->jupyter) (0.4.4)
Requirement already satisfied: overrides in /opt/anaconda3/lib/python3.11/site-pa
ckages (from jupyter-server<3,>=2.4.0->notebook->jupyter) (7.4.0)
Requirement already satisfied: prometheus-client in /opt/anaconda3/lib/python3.1
1/site-packages (from jupyter-server<3,>=2.4.0->notebook->jupyter) (0.14.1)
Requirement already satisfied: send2trash>=1.8.2 in /opt/anaconda3/lib/python3.1
1/site-packages (from jupyter-server<3,>=2.4.0->notebook->jupyter) (1.8.2)
Requirement already satisfied: terminado>=0.8.3 in /opt/anaconda3/lib/python3.11/
site-packages (from jupyter-server<3,>=2.4.0->notebook->jupyter) (0.17.1)
Requirement already satisfied: websocket-client in /opt/anaconda3/lib/python3.11/
site-packages (from jupyter-server<3,>=2.4.0->notebook->jupyter) (0.58.0)
Requirement already satisfied: async-lru>=1.0.0 in /opt/anaconda3/lib/python3.11/
site-packages (from jupyterlab<4.1,>=4.0.2->notebook->jupyter) (2.0.4)
Requirement already satisfied: jupyter-lsp>=2.0.0 in /opt/anaconda3/lib/python3.1
1/site-packages (from jupyterlab<4.1,>=4.0.2->notebook->jupyter) (2.2.0)
```

```
Requirement already satisfied: babel>=2.10 in /opt/anaconda3/lib/python3.11/site-
packages (from jupyterlab-server<3,>=2.22.1->notebook->jupyter) (2.11.0)
Requirement already satisfied: json5>=0.9.0 in /opt/anaconda3/lib/python3.11/site
-packages (from jupyterlab-server<3,>=2.22.1->notebook->jupyter) (0.9.6)
Requirement already satisfied: jsonschema>=4.18.0 in /opt/anaconda3/lib/python3.1
1/site-packages (from jupyterlab-server<3,>=2.22.1->notebook->jupyter) (4.19.2)
Requirement already satisfied: requests>=2.31 in /opt/anaconda3/lib/python3.11/si
te-packages (from jupyterlab-server<3,>=2.22.1->notebook->jupyter) (2.31.0)
Requirement already satisfied: fastjsonschema in /opt/anaconda3/lib/python3.11/si
te-packages (from nbformat>=4.2.0->ipywidgets->jupyter) (2.16.2)
Requirement already satisfied: wcwidth in /opt/anaconda3/lib/python3.11/site-pack
ages (from prompt-toolkit>=3.0.30->jupyter-console->jupyter) (0.2.5)
Requirement already satisfied: soupsieve>1.2 in /opt/anaconda3/lib/python3.11/sit
e-packages (from beautifulsoup4->nbconvert->jupyter) (2.5)
Requirement already satisfied: idna>=2.8 in /opt/anaconda3/lib/python3.11/site-pa
ckages (from anyio>=3.1.0->jupyter-server<3,>=2.4.0->notebook->jupyter) (3.4)
Requirement already satisfied: sniffio>=1.1 in /opt/anaconda3/lib/python3.11/site
-packages (from anyio>=3.1.0->jupyter-server<3,>=2.4.0->notebook->jupyter) (1.3.
0)
Requirement already satisfied: pytz>=2015.7 in /opt/anaconda3/lib/python3.11/site
-packages (from babel>=2.10->jupyterlab-server<3,>=2.22.1->notebook->jupyter) (20
23.3.post1)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /opt/anaconda3/lib/python3.
11/site-packages (from jedi>=0.16->ipython>=7.23.1->ipykernel->jupyter) (0.8.3)
Requirement already satisfied: attrs>=22.2.0 in /opt/anaconda3/lib/python3.11/sit
e-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.22.1->notebook->jupy
ter) (23.1.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /opt/anaco
nda3/lib/python3.11/site-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>
=2.22.1->notebook->jupyter) (2023.7.1)
Requirement already satisfied: referencing>=0.28.4 in /opt/anaconda3/lib/python3.
11/site-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.22.1->notebook
->jupyter) (0.30.2)
Requirement already satisfied: rpds-py>=0.7.1 in /opt/anaconda3/lib/python3.11/si
te-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.22.1->notebook->jup
yter) (0.10.6)
Requirement already satisfied: python-json-logger>=2.0.4 in /opt/anaconda3/lib/py
thon3.11/site-packages (from jupyter-events>=0.6.0->jupyter-server<3,>=2.4.0->not
ebook->jupyter) (2.0.7)
Requirement already satisfied: pyyaml>=5.3 in /opt/anaconda3/lib/python3.11/site-
packages (from jupyter-events>=0.6.0->jupyter-server<3,>=2.4.0->notebook->jupyte
r) (6.0.1)
Requirement already satisfied: rfc3339-validator in /opt/anaconda3/lib/python3.1
1/site-packages (from jupyter-events>=0.6.0->jupyter-server<3,>=2.4.0->notebook->
jupyter) (0.1.4)
Requirement already satisfied: rfc3986-validator>=0.1.1 in /opt/anaconda3/lib/pyt
hon3.11/site-packages (from jupyter-events>=0.6.0->jupyter-server<3,>=2.4.0->note
book->jupyter) (0.1.1)
Requirement already satisfied: ptyprocess>=0.5 in /opt/anaconda3/lib/python3.11/s
ite-packages (from pexpect>4.3->ipython>=7.23.1->ipykernel->jupyter) (0.7.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/anaconda3/lib/pyt
hon3.11/site-packages (from requests>=2.31->jupyterlab-server<3,>=2.22.1->noteboo
k->jupyter) (2.0.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/anaconda3/lib/python3.1
1/site-packages (from requests>=2.31->jupyterlab-server<3,>=2.22.1->notebook->jup
yter) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /opt/anaconda3/lib/python3.1
1/site-packages (from requests>=2.31->jupyterlab-server<3,>=2.22.1->notebook->jup
yter) (2024.2.2)
Requirement already satisfied: argon2-cffi-bindings in /opt/anaconda3/lib/python
```

```
3.11/site-packages (from argon2-cffi->jupyter-server<3,>=2.4.0->notebook->jupyte
r) (21.2.0)
Requirement already satisfied: executing in /opt/anaconda3/lib/python3.11/site-pa
ckages (from stack-data->ipython>=7.23.1->ipykernel->jupyter) (0.8.3)
Requirement already satisfied: asttokens in /opt/anaconda3/lib/python3.11/site-pa
ckages (from stack-data->ipython>=7.23.1->ipykernel->jupyter) (2.0.5)
Requirement already satisfied: pure-eval in /opt/anaconda3/lib/python3.11/site-pa
ckages (from stack-data->ipython>=7.23.1->ipykernel->jupyter) (0.2.2)
Requirement already satisfied: fqdn in /opt/anaconda3/lib/python3.11/site-package
s (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.6.0->jupyter-server<
3,>=2.4.0->notebook->jupyter) (1.5.1)
Requirement already satisfied: isoduration in /opt/anaconda3/lib/python3.11/site-
packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.6.0->jupyter-
server<3,>=2.4.0->notebook->jupyter) (20.11.0)
Requirement already satisfied: jsonpointer>1.13 in /opt/anaconda3/lib/python3.11/
site-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.6.0->jup
yter-server<3,>=2.4.0->notebook->jupyter) (2.1)
Requirement already satisfied: uri-template in /opt/anaconda3/lib/python3.11/site
-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.6.0->jupyter
-server<3,>=2.4.0->notebook->jupyter) (1.3.0)
Requirement already satisfied: webcolors>=1.11 in /opt/anaconda3/lib/python3.11/s
ite-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.6.0->jupy
ter-server<3,>=2.4.0->notebook->jupyter) (1.13)
Requirement already satisfied: cffi>=1.0.1 in /opt/anaconda3/lib/python3.11/site-
packages (from argon2-cffi-bindings->argon2-cffi->jupyter-server<3,>=2.4.0->noteb
ook->jupyter) (1.16.0)
Requirement already satisfied: pycparser in /opt/anaconda3/lib/python3.11/site-pa
ckages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->jupyter-server<3,>=
2.4.0->notebook->jupyter) (2.21)
Requirement already satisfied: arrow>=0.15.0 in /opt/anaconda3/lib/python3.11/sit
e-packages (from isoduration->jsonschema[format-nongpl]>=4.18.0->jupyter-events>=
0.6.0->jupyter-server<3,>=2.4.0->notebook->jupyter) (1.2.3)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:

```python
# Importing necessary library
from nbconvert import HTMLExporter

# Instantiating the HTMLExporter
html_exporter = HTMLExporter()

# Converting the notebook to HTML
(output_html, resources) = html_exporter.from_filename('ML_Phase_1.ipynb')

# Writing the HTML output to a file
with open('ML_Phase_1.html', 'w') as f:
    f.write(output_html)
```

In [ ]: