Py Project

Introduction

This project is about exploring factors affecting liver disease.

Dataset Description

The dataset contains information related to hepatitis, with 155 instances and 20 attributes, including the class attribute. The data was donated by G.Gong from Carnegie-Mellon University via Bojan Cestnik of Jozef Stefan Institute.

Variable Types and Statistical Measures

Categorical Variables: Type of variables that represent categories or groups. These variables can take on a limited, fixed number of distinct values or labels, and there is no inherent order or numerical significance among these categories. Class (DIE, LIVE), SEX (male, female), STEROID (no, yes), ANTIVIRALS (no, yes), FATIGUE (no, yes), MALAISE (no, yes), ANOREXIA (no, yes), LIVER BIG (no, yes), LIVER FIRM (no, yes), SPLEEN PALPABLE (no, yes), SPIDERS (no, yes), ASCITES (no, yes), VARICES (no, yes), HISTOLOGY (no, yes)

Ordinal Variables: Type of categorical variable that, in addition to having distinct categories, also have a meaningful order or ranking among them. However, the intervals between the categories are not necessarily uniform or measurable. AGE (10, 20, 30, 40, 50, 60, 70, 80)

Continuous Variables: Quantitative variables that can take on an infinite number of values within a given range. These variables are typically measured on a continuous scale and can include decimal values. BILIRUBIN, ALK PHOSPHATE, SGOT, ALBUMIN,PROTIME

Graphical Displays and Statistical Measures:

Categorical Variables: Bar charts to show the distribution of each category. Class distribution can be visualized using a bar chart.

Ordinal Variables: Histogram to show the distribution of ages.

Continuous Variables: Box plots to identify outliers. Histograms for a visual representation of continuous variable distributions.

Dataset:

In [6]:
```
import pandas as pd
df = pd.read_csv("hepatitis.txt")

print(df.info())
print(df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 154 entries, 0 to 153
Data columns (total 20 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2       154 non-null    int64
 1   30      153 non-null    object
 2   2.1     154 non-null    int64
 3   1       152 non-null    object
 4   2.2     154 non-null    int64
 5   2.3     153 non-null    object
 6   2.4     154 non-null    object
 7   2.5     154 non-null    object
 8   1.1     154 non-null    object
 9   2.6     154 non-null    object
 10  2.7     153 non-null    object
 11  2.8     154 non-null    object
 12  2.9     154 non-null    object
 13  2.10    154 non-null    object
 14  1.00    152 non-null    object
 15  85      153 non-null    object
 16  18      154 non-null    object
 17  4.0     153 non-null    object
 18  ?       154 non-null    object
 19  1.2     153 non-null    float64
dtypes: float64(1), int64(3), object(16)
memory usage: 24.2+ KB
None
                2         2.1         2.2         1.2
count  154.000000  154.000000  154.000000  153.000000
mean     1.792208    1.097403    1.844156    1.457516
std      0.407051    0.297473    0.363891    0.499828
min      1.000000    1.000000    1.000000    1.000000
25%      2.000000    1.000000    2.000000    1.000000
50%      2.000000    1.000000    2.000000    1.000000
75%      2.000000    1.000000    2.000000    2.000000
max      2.000000    2.000000    2.000000    2.000000
```

Data Cleaning:

Data cleaning plays a pivotal role in the data preparation phase as it entails the identification and rectification of errors or inconsistencies within a dataset. This essential process guarantees the accuracy, comprehensiveness, and analysis readiness of the data. Typical data cleaning activities involve addressing missing values, eliminating duplicates, rectifying data types, and getting rid of irrelevant or redundant information. The primary objective is to improve the dataset's quality, alleviate potential biases or inaccuracies, and establish a more dependable basis for meaningful analysis and interpretation.

In [7]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

column_names = ["Class", "AGE", "SEX", "STEROID", "ANTIVIRALS", "FATIGUE", "MALAISE", "ANOREXIA",
                "LIVER BIG", "LIVER FIRM", "SPLEEN PALPABLE", "SPIDERS", "ASCITES", "VARICES",
                "BILIRUBIN", "ALK PHOSPHATE", "SGOT", "ALBUMIN", "PROTIME", "HISTOLOGY"]

missing_values = {"BILIRUBIN": ["?"], "ALK PHOSPHATE": ["?"], "SGOT": ["?"], "ALBUMIN": ["?"], "PROTIME": ["?"]}

df = pd.read_csv("hepatitis.txt", names=column_names, na_values=missing_values)
df.dropna(inplace=True)
df["BILIRUBIN"] = df["BILIRUBIN"].astype(float)

# Save the cleaned dataset
df.to_csv("hepatitis_cleaned.csv", index=False)
print(df.info())
print(df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 77 entries, 3 to 154
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Class            77 non-null     int64
 1   AGE              77 non-null     object
 2   SEX              77 non-null     int64
 3   STEROID          77 non-null     object
 4   ANTIVIRALS       77 non-null     int64
 5   FATIGUE          77 non-null     object
 6   MALAISE          77 non-null     object
 7   ANOREXIA         77 non-null     object
 8   LIVER BIG        77 non-null     object
 9   LIVER FIRM       77 non-null     object
 10  SPLEEN PALPABLE  77 non-null     object
 11  SPIDERS          77 non-null     object
 12  ASCITES          77 non-null     object
 13  VARICES          77 non-null     object
 14  BILIRUBIN        77 non-null     float64
 15  ALK PHOSPHATE    77 non-null     object
 16  SGOT             77 non-null     float64
 17  ALBUMIN          77 non-null     object
 18  PROTIME          77 non-null     object
 19  HISTOLOGY        77 non-null     float64
dtypes: float64(3), int64(3), object(14)
memory usage: 12.6+ KB
None
           Class        SEX  ANTIVIRALS  BILIRUBIN        SGOT  HISTOLOGY
count  77.000000  77.000000   77.000000  77.000000   77.000000  77.000000
mean    1.805195   1.103896    1.727273   1.277922   90.220779   1.428571
std     0.398648   0.307127    0.448282   0.937329   88.040305   0.498117
min     1.000000   1.000000    1.000000   0.300000   14.000000   1.000000
25%     2.000000   1.000000    1.000000   0.700000   31.000000   1.000000
50%     2.000000   1.000000    2.000000   1.000000   60.000000   1.000000
75%     2.000000   1.000000    2.000000   1.300000  118.000000   2.000000
max     2.000000   2.000000    2.000000   4.800000  528.000000   2.000000
```

Data Analysis:

Hypthesis Testing

In [8]:
```
import pandas as pd
from scipy.stats import ttest_ind, chi2_contingency

df = pd.read_csv("hepatitis_cleaned.csv")
selected_factors = ['AGE', 'SEX', 'ANTIVIRALS', 'BILIRUBIN', 'SGOT', 'HISTOLOGY']


class_1_data = df[df['Class'] == 1]
class_2_data = df[df['Class'] == 2]


continuous_columns = ['BILIRUBIN', 'SGOT']

for column in continuous_columns:
    t_statistic, p_value = ttest_ind(class_1_data[column].dropna(), class_2_data[column].dropna())
    print(f'Test for {column}:')
    print(f'T-statistic: {t_statistic}')
    print(f'P-value: {p_value}')
    if p_value < 0.05:
        print('Reject the null hypothesis. There is a significant difference.\n')
    else:
        print('Fail to reject the null hypothesis. There is no significant difference.\n')


categorical_columns = ['SEX', 'STEROID', 'ANTIVIRALS', 'FATIGUE', 'MALAISE', 'ANOREXIA', 'LIVER BIG', 'LIVER FIRM', 'SPLEEN PALPABLE', 'SPIDERS', 'ASCITES', 'VARICES', 'HISTOLOGY']

# Perform chi-square tests for categorical variables
for column in categorical_columns:
    contingency_table = pd.crosstab(df['Class'], df[column])
    chi2_stat, p_value, _, _ = chi2_contingency(contingency_table)
    print(f'Test for {column}:')
    print(f'Chi-square statistic: {chi2_stat}')
    print(f'P-value: {p_value}')
    if p_value < 0.05:
        print('Reject the null hypothesis. There is a significant difference.\n')
    else:
        print('Fail to reject the null hypothesis. There is no significant difference.\n')
```

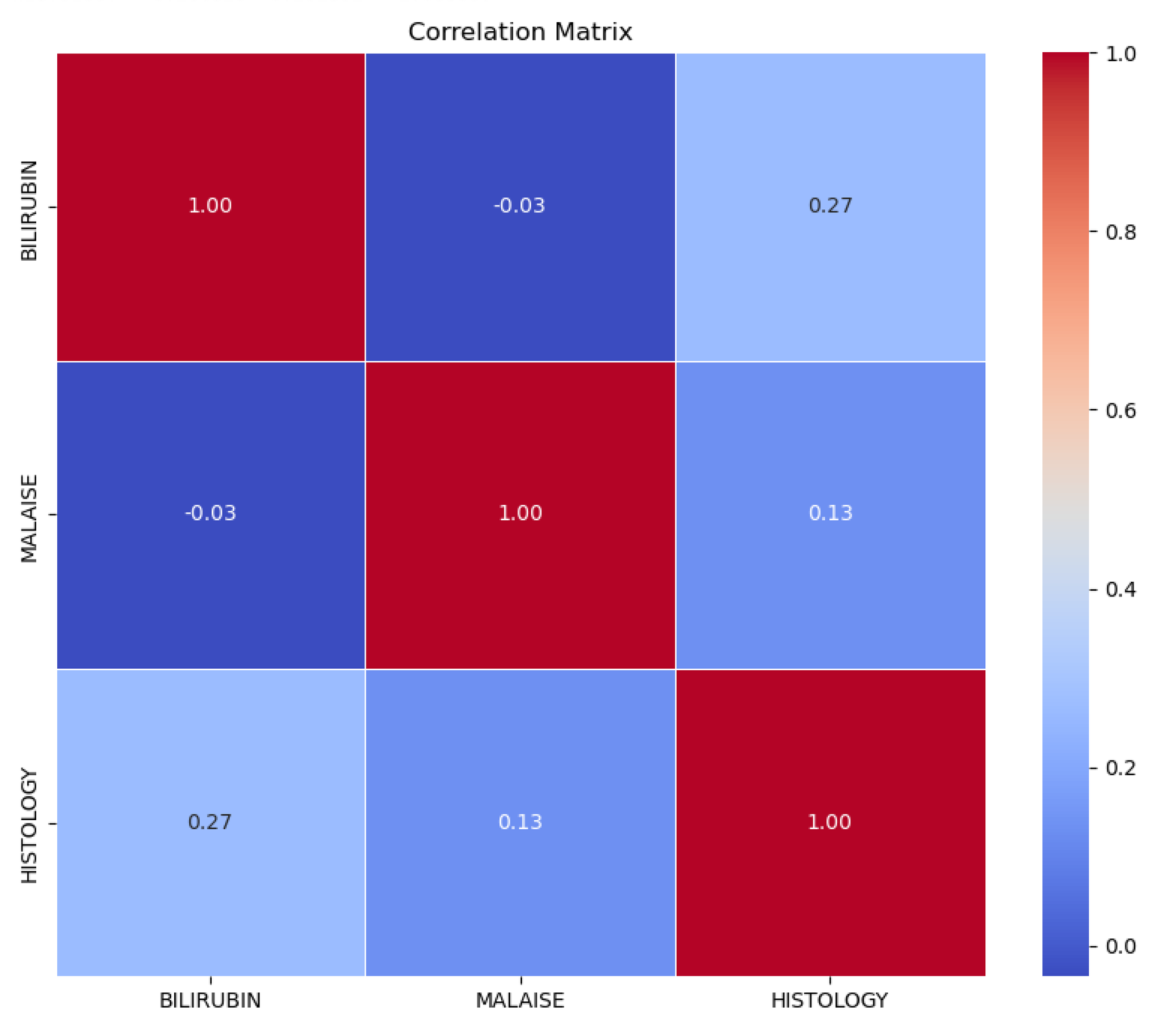
```
Test for BILIRUBIN:
T-statistic: 3.768475950186287
P-value: 0.00032500395046467645
Reject the null hypothesis. There is a significant difference.

Test for SGOT:
T-statistic: 0.3369026016254193
P-value: 0.7371312695067664
Fail to reject the null hypothesis. There is no significant difference.

Test for SEX:
Chi-square statistic: 0.9962857448963689
P-value: 0.3182109210444769
Fail to reject the null hypothesis. There is no significant difference.

Test for STEROID:
Chi-square statistic: 6.17730217810863
P-value: 0.1032956023235858
Fail to reject the null hypothesis. There is no significant difference.

Test for ANTIVIRALS:
Chi-square statistic: 1.0565076164874556
P-value: 0.30401307029693575
Fail to reject the null hypothesis. There is no significant difference.

Test for FATIGUE:
Chi-square statistic: 2.7694394663480684
P-value: 0.09607980135369354
Fail to reject the null hypothesis. There is no significant difference.

Test for MALAISE:
Chi-square statistic: 8.521843758430649
P-value: 0.014109289352254913
Reject the null hypothesis. There is a significant difference.

Test for ANOREXIA:
Chi-square statistic: 0.2794354838709676
P-value: 0.5970714594639474
Fail to reject the null hypothesis. There is no significant difference.

Test for LIVER BIG:
Chi-square statistic: 10.697204301075269
P-value: 0.004754792840798936
Reject the null hypothesis. There is a significant difference.

Test for LIVER FIRM:
Chi-square statistic: 8.784623655913977
P-value: 0.012372093980572119
Reject the null hypothesis. There is a significant difference.

Test for SPLEEN PALPABLE:
Chi-square statistic: 7.459625390218524
P-value: 0.023997330233102146
Reject the null hypothesis. There is a significant difference.

Test for SPIDERS:
Chi-square statistic: 10.62284439034287
P-value: 0.013950313877031165
Reject the null hypothesis. There is a significant difference.

Test for ASCITES:
Chi-square statistic: 22.833944342164948
P-value: 4.3732663366086055e-05
Reject the null hypothesis. There is a significant difference.

Test for VARICES:
Chi-square statistic: 13.162731147131726
P-value: 0.0013859553779612228
Reject the null hypothesis. There is a significant difference.

Test for HISTOLOGY:
Chi-square statistic: 16.90584677419355
P-value: 3.9280405089975144e-05
Reject the null hypothesis. There is a significant difference.
```

In summary, the tests indicate significant differences in several variables(BILIRUBIN, MALAISE, LIVER BIG, LIVER FIRM, SPLEEN PALPABLE, SPIDERS, ASCITES, VARICES, HISTOLOGY), suggesting meaningful associations or patterns in the dataset for those specific factors. Other factors (SGOT, SEX, STEROID, ANTIVIRALS, FATIGUE, ANOREXIA) do not have any significant differnece.

Correlation Analysis

In [10]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv("hepatitis_cleaned.csv")
selected_columns = ['BILIRUBIN', 'MALAISE', 'LIVER BIG', 'LIVER FIRM', 'SPLEEN PALPABLE', 'SPIDERS', 'ASCITES', 'VARICES', 'HISTOLOGY']
selected_data = df[selected_columns]

correlation_matrix = selected_data.corr()
print(correlation_matrix)

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Matrix')
plt.show()
```

```
           BILIRUBIN   MALAISE  HISTOLOGY
BILIRUBIN   1.000000 -0.034376   0.268529
MALAISE    -0.034376  1.000000   0.132362
HISTOLOGY   0.268529  0.132362   1.000000
```

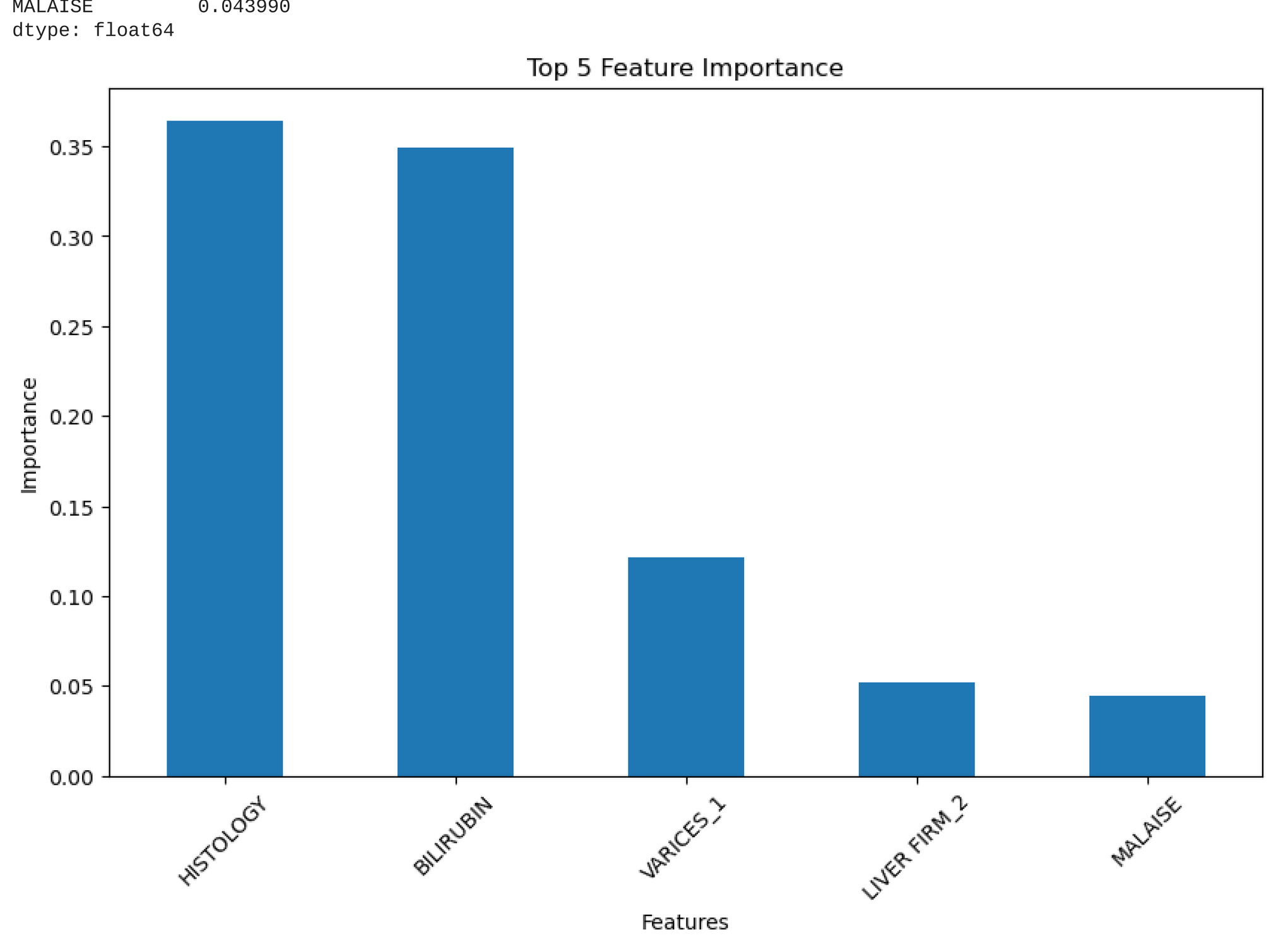


BILIRUBIN and MALAISE have a very weak negative correlation (-0.034376). BILIRUBIN and HISTOLOGY have a moderate positive correlation (0.268529). MALAISE and HISTOLOGY have a weak positive correlation (0.132362).

Model

In [14]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix

df = pd.read_csv("hepatitis_cleaned.csv")
df = df.dropna()

selected_features = ['BILIRUBIN', 'MALAISE', 'LIVER BIG', 'LIVER FIRM', 'SPLEEN PALPABLE', 'SPIDERS', 'ASCITES', 'VARICES', 'HISTOLOGY']
X = df[selected_features]
y = df['Class']

X = pd.get_dummies(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

# Print the evaluation metrics
print(f"Accuracy: {accuracy:.3f}")
print(f"Precision: {precision:.3f}")
print(f"Recall: {recall:.3f}")
print(f"F1 Score: {f1:.3f}")
print("Confusion Matrix:")
print(conf_matrix)

# 8. Feature Importance
n_top_features = 5

feature_importance = pd.Series(model.feature_importances_, index=X_train.columns)
sorted_importance = feature_importance.sort_values(ascending=False).head(n_top_features)
print(sorted_importance)

plt.figure(figsize=(10, 6))
sorted_importance.plot(kind='bar', rot=45)
plt.title('Top {} Feature Importance'.format(n_top_features))
plt.xlabel('Features')
plt.ylabel('Importance')
plt.show()
```

```
Accuracy: 0.812
Precision: 0.333
Recall: 0.500
F1 Score: 0.400
Confusion Matrix:
[[ 1  1]
 [ 2 12]]
HISTOLOGY       0.364583
BILIRUBIN       0.348911
VARICES_1       0.121497
LIVER FIRM_2    0.052137
MALAISE         0.043990
dtype: float64
```



The model's accuracy is decent at 81.2%, but other metrics should be considered, especially if there's an imbalance in the dataset. The precision is relatively low, indicating that when the model predicts positive, it has a high chance of being incorrect. The recall is moderate, suggesting that the model is reasonably good at capturing actual positive instances. The F1 score balances precision and recall, and a score of 0.4 suggests room for improvement.

HISTOLOGY and BILIRUBIN are the most important features for predicting the outcome, contributing significantly to the model's decision-making process. VARICES_1, LIVER FIRM_2, and MALAISE also contribute to the model's predictions, albeit to a lesser extent.

Conclusion:

Based on the analyses, it is evident that certain clinical and laboratory features, such as BILIRUBIN levels, histological conditions, and the presence of specific symptoms (e.g., MALAISE), play crucial roles in predicting the outcome. The model, while showing decent accuracy, requires further refinement, especially in reducing false positives and improving recall.

In [ ]: