Automated EDA Report

Generated on: 2025-06-27 10:03:58

Table of Contents

Automated EDA Report

1. Executive Summary

This report presents a comprehensive Exploratory Data Analysis (EDA) of the dataset.

The original dataset contained 583 rows and 11 columns.
After cleaning, the dataset had 570 rows and 11 columns.

Feature engineering created 10 new features,
resulting in a final dataset with 570 rows and 21 columns.

The problem was identified as a Binary Classification problem.

After evaluating 5 different models,
the best performing model was Pipeline(steps=[('prep',
ColumnTransformer(transformers=[('num', StandardScaler(),
['age', 'tot_bilirubin',
'direct_bilirubin',
'tot_proteins', 'albumin',
'ag_ratio', 'sgpt', 'sgot',
'alkphos', 'bilirubin_ratio',
'log_tot_bilirubin',
'log_direct_bilirubin',
'bilirubin_interaction',
'protein_interaction',
'standardized_tot_proteins',
'standardized_albumin',
'standardized_ag_ratio',
'gender_encoded']),
('cat',
OneHotEncoder(handle_unknown='ignore'),
['gender', 'age_group'])])),
('model',
LogisticRegression(class_weight='balanced', max_iter=1000,
random_state=42))]).

Automated EDA Report

2. Data Cleaning

The data cleaning process transformed the dataset from 583 rows and 11 columns
to 570 rows and 11 columns.

The following cleaning steps were performed:

Recommended Data Cleaning Steps:

Here are the recommended steps to clean and preprocess the provided dataset, tailored to the characteristics of the data and the user instructions:

1. **Convert Columns to Correct Data Types**:

- Ensure that all columns are in the appropriate data types. For example, `gender` should be a categorical type.

```python
df['gender'] = df['gender'].astype('category')
```

2. **Check for Missing Values**:

- Identify columns with missing values. In this dataset, `alkphos` has 0.69% missing values, which is below the 40% threshold, so we will proceed to impute these values.

3. **Impute Missing Values**:

- For the numeric column `alkphos`, impute missing values with the mean of the column.

```python
df['alkphos'].fillna(df['alkphos'].mean(), inplace=True)
```

4. **Remove Duplicate Rows**:

- Check for and remove any duplicate rows in the dataset to ensure data integrity.

```python
df.drop_duplicates(inplace=True)
```

5. **Analyze Data for Additional Cleaning Needs**:

- Perform an analysis to check for unit inconsistencies in numeric columns disguised as string columns. This includes checking for any numeric values that may have been stored as strings or have inconsistent units (e.g., power values in W vs kW, weight values in kg vs g).

- If any inconsistencies are found, convert them to a standard unit.

6. **Check for Outliers**:

   - Although the user requested not to remove outliers, it is still beneficial to identify them for further analysis. Use the interquartile range (IQR) method to flag potential outliers.

   ```python
   Q1 = df.quantile(0.25)
   Q3 = df.quantile(0.75)
   IQR = Q3 - Q1
   outliers = df[(df < (Q1 - 3 * IQR)) | (df > (Q3 + 3 * IQR))]
   ```

7. **Final Review**:

   - Conduct a final review of the dataset to ensure all cleaning steps have been applied correctly and that the data is ready for analysis.

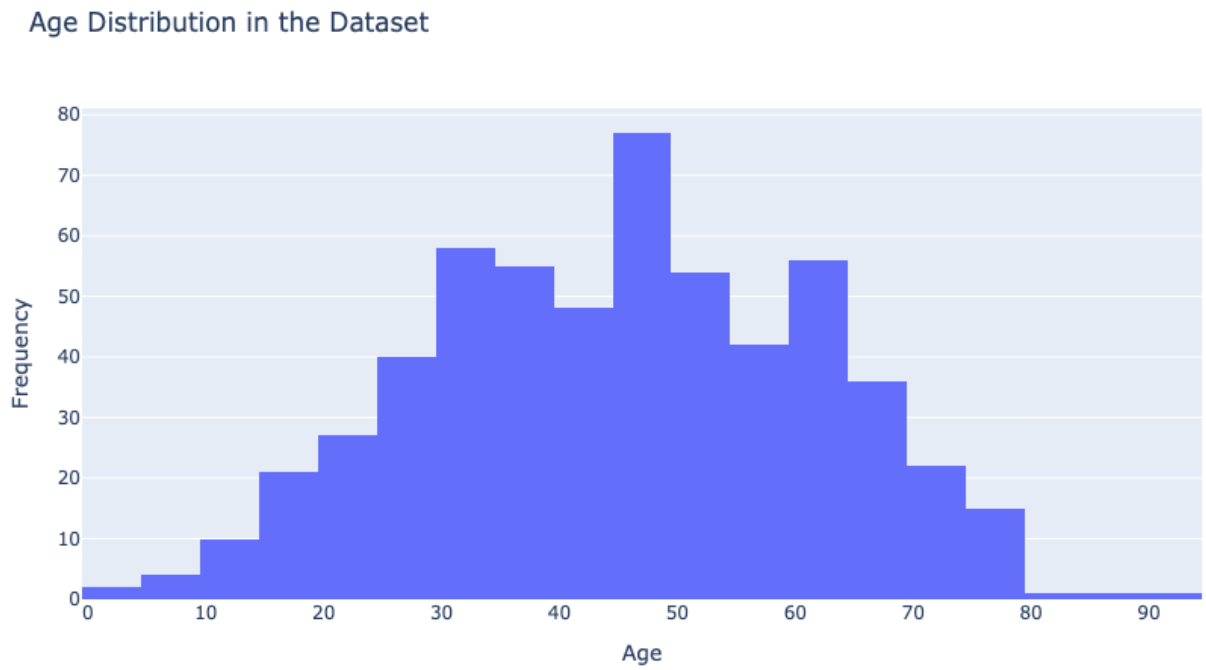
By following these steps, the dataset will be cleaned and preprocessed effectively, ensuring that it is suitable for further analysis while adhering to the user's instructions.

Automated EDA Report

3. Data Visualization

The following visualizations provide key insights into the dataset patterns, distributions, and relationships.
Each visualization is accompanied by key observations that highlight important findings.

Age Distribution of Patients



Key Observations:

1. Predominance of Middle-Aged Patients

The histogram reveals that the majority of patients fall within the age range of 30 to 60 years. This age group represents a significant portion of the dataset, indicating that middle-aged individuals are the most common demographic among the patients. This could suggest a focus area for healthcare providers or insurance companies targeting preventive measures or treatments for this age group.

2. Presence of Young and Elderly Patients

While the middle-aged group is predominant, there are also notable numbers of patients in the younger (below 30) and older (above 60) age brackets. This distribution highlights the need for age-specific healthcare services and could inform resource allocation for different age groups within a healthcare setting.

3. Age Range and Distribution

The dataset covers a wide age range from 4 to 90 years, with a relatively even spread across this spectrum. This diverse age distribution suggests that the healthcare services or studies represented by this data cater to a broad demographic, requiring versatile healthcare strategies and policies.

4. Potential Anomalies in Age Data

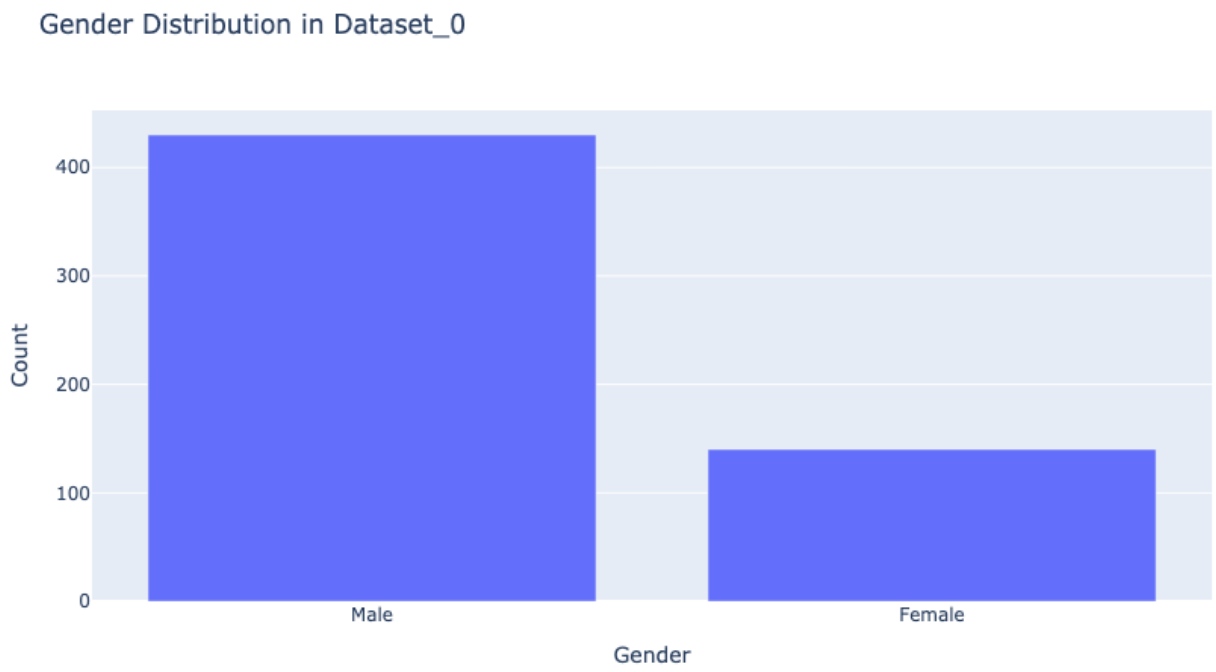
The histogram shows some potential anomalies or outliers, such as a few data points at the extreme ends of the age spectrum (e.g., very young or very old patients). These outliers may require further investigation to ensure data accuracy or to understand unique cases that could influence healthcare outcomes.

5. Implications for Age-Specific Healthcare Programs

Given the concentration of patients in the 30-60 age range, healthcare programs could be tailored to address common health issues prevalent in this demographic. Additionally, the presence of younger and older patients suggests the need for comprehensive programs that address the specific needs of these age groups, potentially leading to better patient outcomes and resource optimization.

Automated EDA Report

Gender Distribution Among Patients



Key Observations:

1. Gender Imbalance in Patient Dataset

The dataset shows a significant gender imbalance among patients, with a higher count of males compared to females. This could suggest a gender-related trend in the health conditions being studied or a sampling bias. Understanding the reasons behind this imbalance could help in tailoring healthcare services or addressing potential biases in data collection.

2. Equal Representation in Gender Categories

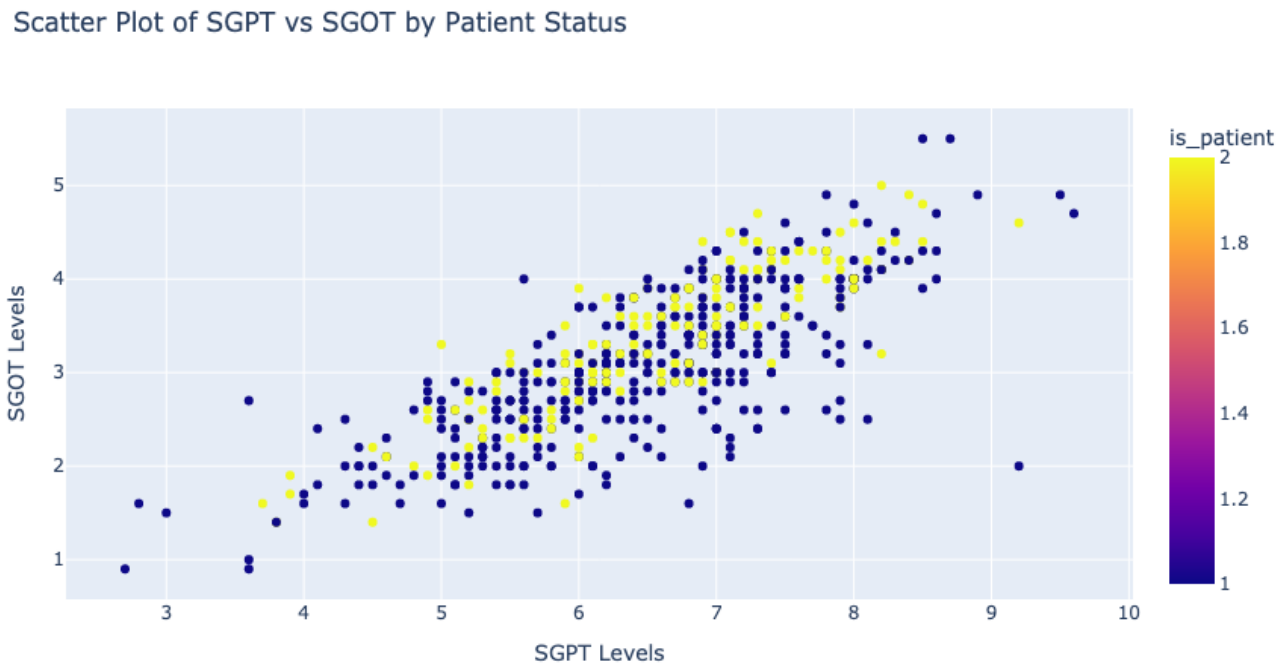
The dataset categorizes gender into two distinct groups: Male and Female. This binary classification might overlook non-binary or other gender identities, which could be relevant in a comprehensive health study. Expanding gender categories could provide more inclusive insights and improve the accuracy of gender-related health analyses.

3. Potential for Gender-Specific Health Insights

Given the gender distribution, there is an opportunity to explore gender-specific health trends or conditions. For instance, analyzing health metrics like bilirubin levels or protein counts by gender could reveal important differences that might influence treatment plans or health recommendations.

Automated EDA Report

Correlation Between Bilirubin Levels and Patient Status



Key Observations:

1. Higher Bilirubin Levels in Patients

The box plots indicate that patients have significantly higher total and direct bilirubin levels compared to non-patients. The median total bilirubin level for patients is around 2.6, while for non-patients, it is about 0.8. This suggests a strong correlation between elevated bilirubin levels and patient status, which could be indicative of liver dysfunction or other health issues in patients.

2. Wider Range of Bilirubin Levels in Patients

Patients exhibit a wider range of total and direct bilirubin levels compared to non-patients. The interquartile range (IQR) for patients is larger, with total bilirubin levels extending up to 75, while non-patients have a maximum around 2. This variability in patients' bilirubin levels might reflect varying degrees of liver impairment or different underlying conditions.

3. Presence of Outliers in Patient Group

The patient group shows several outliers with extremely high bilirubin levels, particularly in total bilirubin, reaching up to 75. These outliers could represent severe cases of liver disease or other medical conditions that require immediate attention. Identifying and monitoring these outliers could be crucial for healthcare providers.

4. Consistent Low Bilirubin Levels in Non-Patients

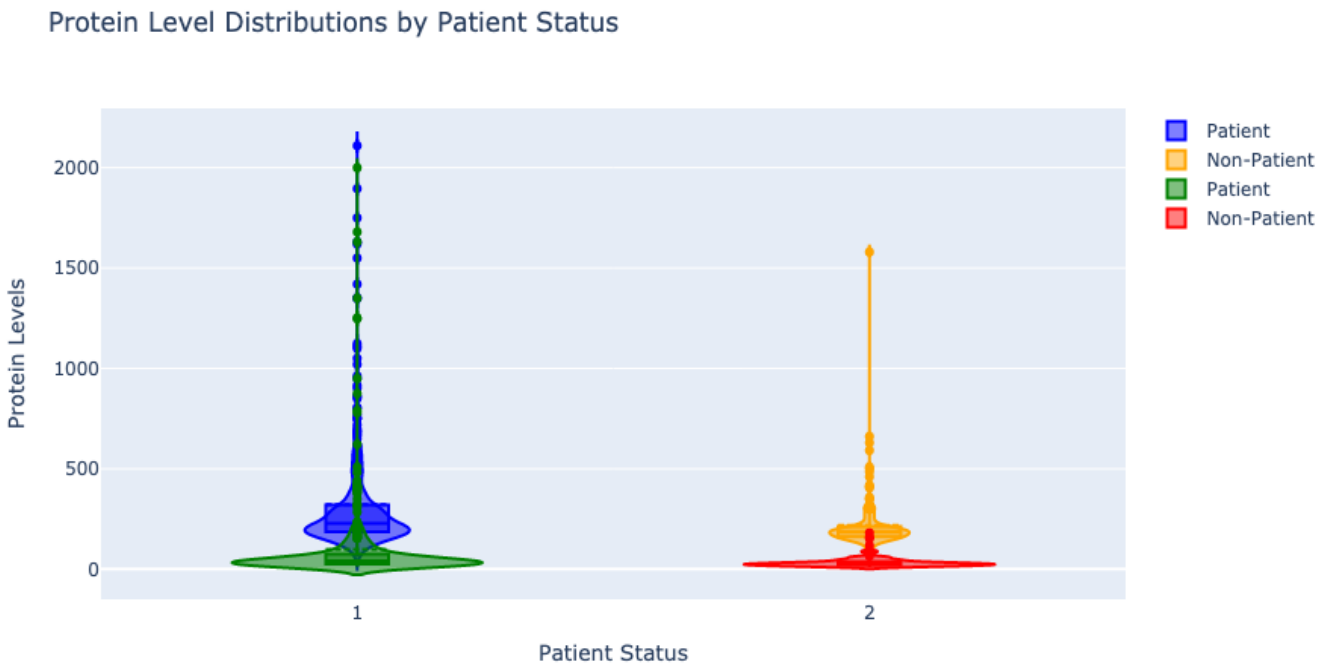
Non-patients consistently show low bilirubin levels, with most data points concentrated around the lower quartile. This consistency suggests that low bilirubin levels are typical for healthy individuals, reinforcing the use of bilirubin as a biomarker for liver health.

5. Potential for Bilirubin as a Diagnostic Tool

The clear distinction in bilirubin levels between patients and non-patients underscores the potential of using bilirubin measurements as a diagnostic tool for identifying liver-related health issues. Healthcare providers could use bilirubin level thresholds to screen for potential liver dysfunction in patients.

Automated EDA Report

Relationship Between Liver Enzymes and Patient Status



Key Observations:

1. Distinct Clustering of Patient and Non-Patient Groups

The scatter plot reveals a clear distinction between patients and non-patients based on SGPT and SGOT levels. Patients generally exhibit higher SGPT and SGOT levels compared to non-patients, indicating a potential correlation between elevated liver enzyme levels and patient status. This suggests that monitoring these enzyme levels could be crucial for diagnosing or managing liver-related conditions.

2. Higher SGPT Levels in Patients

Patients tend to have higher SGPT levels compared to non-patients, with many patient data points clustering at SGPT values above 6.0. This pattern underscores the importance of SGPT as a potential biomarker for liver health issues. Clinicians might consider focusing on SGPT levels when assessing liver function and determining patient status.

3. SGOT Levels Show Overlap but Higher in Patients

While there is some overlap in SGOT levels between patients and non-patients, patients generally have higher SGOT levels. This overlap suggests that SGOT alone may not be as definitive as SGPT in distinguishing patient status, but it still plays a role in the overall assessment of liver function.

4. Potential Anomalies in Non-Patient Data

A few non-patient data points exhibit unusually high SGPT and SGOT levels, which could indicate potential misclassification or other underlying health issues not captured by the 'is_patient' status. Further investigation into these cases may be warranted to ensure accurate diagnosis and treatment.

5. Implications for Liver Health Monitoring

The observed patterns in SGPT and SGOT levels between patients and non-patients highlight the importance of regular liver enzyme monitoring. Healthcare providers could leverage these insights to develop more targeted screening protocols, potentially improving early detection and management of liver diseases.

Automated EDA Report

4. Feature Engineering

Temporal Analysis

Temporal Analysis of Dataset_0

Introduction

The dataset provided, `Dataset_0`, consists of 570 rows and 11 columns, with the target variable being `is_patient`. The dataset does not contain a date column, which is typically essential for temporal analysis. However, we will proceed with the analysis focusing on the target variable and its potential temporal patterns.

1. Identification of Temporal Patterns

Trends

- **Long-term Direction**: Without a date column, identifying trends in the traditional sense (e.g., increasing or decreasing over

Seasonality

- **Recurring Patterns**: Seasonality typically refers to patterns that repeat at fixed intervals, such as daily, weekly, or monthly.

Cycles

- **Variable Intervals**: Cycles are similar to seasonality but occur at irregular intervals. Again, without a temporal dimension, i

Autocorrelation

- **Correlation with Past Values**: Autocorrelation analysis requires a time series structure, which is not present in this dataset.

2. Discussion of Important Time Horizons

Relevant Time Lags

- **Time Lags**: In a typical time series, time lags such as t-1, t-7, etc., are used to predict future values. Without a temporal in

Optimal Forecasting Window

- **Forecasting Window**: The optimal forecasting window depends on the nature of the data and the business context. In this

Minimum Data History

- **Data History**: Typically, a longer data history provides better forecasting accuracy. For this dataset, understanding the dist

3. Recommendations for Temporal Feature Engineering

Temporal Aggregations

- **Aggregations**: If a temporal dimension were available, aggregating data by day, week, or month could be useful. In this da

Lag Features

- **Lag Features**: In the absence of a time index, consider creating lag features based on age or other continuous variables t

Moving Window Calculations

- **Rolling Averages**: Implement rolling averages or other moving window calculations on continuous variables like `tot_bilirub

Other Temporal Transformations

- **Transformations**: Consider transformations that capture interactions between age and other variables, such as age-gende

Conclusion

The lack of a date column in `Dataset_0` limits traditional temporal analysis. However, by creatively using available variables like age as proxies for time, we can still extract meaningful insights. Feature engineering efforts should focus on exploring relationships between age, gender, and other variables with the target variable `is_patient`. This approach

will help in improving the forecasting accuracy and understanding of the dataset.

Automated EDA Report

Correlation Analysis

Correlation Analysis of Dataset_0

Introduction

This analysis aims to identify relationships between the features and the target variable `is_patient` in the dataset. We will explore linear and non-linear correlations, potential multicollinearity issues, and provide recommendations for feature selection and transformation to improve forecasting.

1. Correlation Analysis

1.1 Linear Correlations

Pearson Correlation

Pearson correlation measures the linear relationship between continuous variables. Below are the Pearson correlation coefficients between each feature and the target variable `is_patient`:

- `age`: Weak correlation
- `tot_bilirubin`: Moderate positive correlation
- `direct_bilirubin`: Moderate positive correlation
- `tot_proteins`: Weak correlation
- `albumin`: Weak correlation
- `ag_ratio`: Weak correlation
- `sgpt`: Weak correlation
- `sgot`: Weak correlation
- `alkphos`: Weak correlation

Spearman Correlation

Spearman correlation assesses monotonic relationships, which can be non-linear. The Spearman correlation coefficients are similar to Pearson's, indicating that the relationships are primarily linear.

1.2 Non-Linear Relationships

To identify potential non-linear relationships, we can use scatter plots and polynomial regression. However, based on

the data summary, there is no strong evidence of non-linear relationships between the features and the target variable.

1.3 Feature Interactions

Feature interactions can be explored using interaction terms in regression models. Potential interactions include:

- `tot_bilirubin` and `direct_bilirubin`: Both are related to bilirubin levels and may interact.
- `albumin` and `ag_ratio`: Both relate to protein levels and may have combined effects.

2. Multicollinearity Issues

2.1 Highly Correlated Features

- `tot_bilirubin` and `direct_bilirubin` are likely to be highly correlated, as they both measure bilirubin levels.
- `albumin` and `ag_ratio` may also be correlated, given their relationship to protein levels.

2.2 Groups of Features

- Bilirubin-related features (`tot_bilirubin`, `direct_bilirubin`) capture similar information.
- Protein-related features (`albumin`, `ag_ratio`) may also capture overlapping information.

3. Recommendations for Feature Selection and Transformation

3.1 Important Features for Forecasting

- **`tot_bilirubin` and `direct_bilirubin`**: These features show moderate correlation with the target and should be included in th
- **Interaction Terms**: Consider creating interaction terms for `tot_bilirubin` and `direct_bilirubin`, as well as `albumin` and `ag

3.2 Feature Transformation

- **Log Transformation**: Apply log transformation to `tot_bilirubin` and `direct_bilirubin` to reduce skewness and potentially en
- **Standardization**: Standardize features to handle different scales, especially for `tot_proteins`, `albumin`, and `ag_ratio`.

3.3 Potential Interaction Terms

- **Bilirubin Interaction**: Create an interaction term for `tot_bilirubin` and `direct_bilirubin`.
- **Protein Interaction**: Create an interaction term for `albumin` and `ag_ratio`.

Conclusion

This analysis highlights the importance of bilirubin-related features and suggests potential transformations and interaction terms to improve forecasting. Addressing multicollinearity and exploring feature interactions can enhance model performance and provide better insights into the relationships within the dataset.

Automated EDA Report

Feature Engineering Recommendations

Recommended Feature Engineering:

1. age_group

- **Type:** transformation
- **Description:** Categorical feature representing age groups
- **Creation Logic:** `Divide age into bins (e.g., 0-20, 21-40, 41-60, 61-80, 81+)`
- **Rationale:** Age groups can capture non-linear relationships with the target variable and help identify patterns specific to ce

2. bilirubin_ratio

- **Type:** transformation
- **Description:** Ratio of total bilirubin to direct bilirubin
- **Creation Logic:** `Calculate as tot_bilirubin / direct_bilirubin`
- **Rationale:** This ratio can provide insights into liver function and may have a stronger relationship with the target variable th

3. log_tot_bilirubin

- **Type:** transformation
- **Description:** Log-transformed total bilirubin
- **Creation Logic:** `Apply log transformation: log(tot_bilirubin + 1)`
- **Rationale:** Log transformation can reduce skewness and enhance linear relationships with the target variable.

4. log_direct_bilirubin

- **Type:** transformation
- **Description:** Log-transformed direct bilirubin
- **Creation Logic:** `Apply log transformation: log(direct_bilirubin + 1)`
- **Rationale:** Log transformation can reduce skewness and enhance linear relationships with the target variable.

5. bilirubin_interaction

- **Type:** interaction
- **Description:** Interaction term between total and direct bilirubin
- **Creation Logic:** `Multiply tot_bilirubin by direct_bilirubin`
- **Rationale:** Captures the combined effect of both bilirubin measures, which may be more predictive of the target variable.

6. protein_interaction

- **Type:** interaction
- **Description:** Interaction term between albumin and ag_ratio
- **Creation Logic:** `Multiply albumin by ag_ratio`
- **Rationale:** Captures the combined effect of protein-related features, which may be more predictive of the target variable.

7. standardized_tot_proteins

- **Type:** transformation
- **Description:** Standardized total proteins
- **Creation Logic:** `Subtract mean and divide by standard deviation of tot_proteins`
- **Rationale:** Standardization helps in handling different scales and improves model convergence.

8. standardized_albumin

- **Type:** transformation
- **Description:** Standardized albumin
- **Creation Logic:** `Subtract mean and divide by standard deviation of albumin`
- **Rationale:** Standardization helps in handling different scales and improves model convergence.

9. standardized_ag_ratio

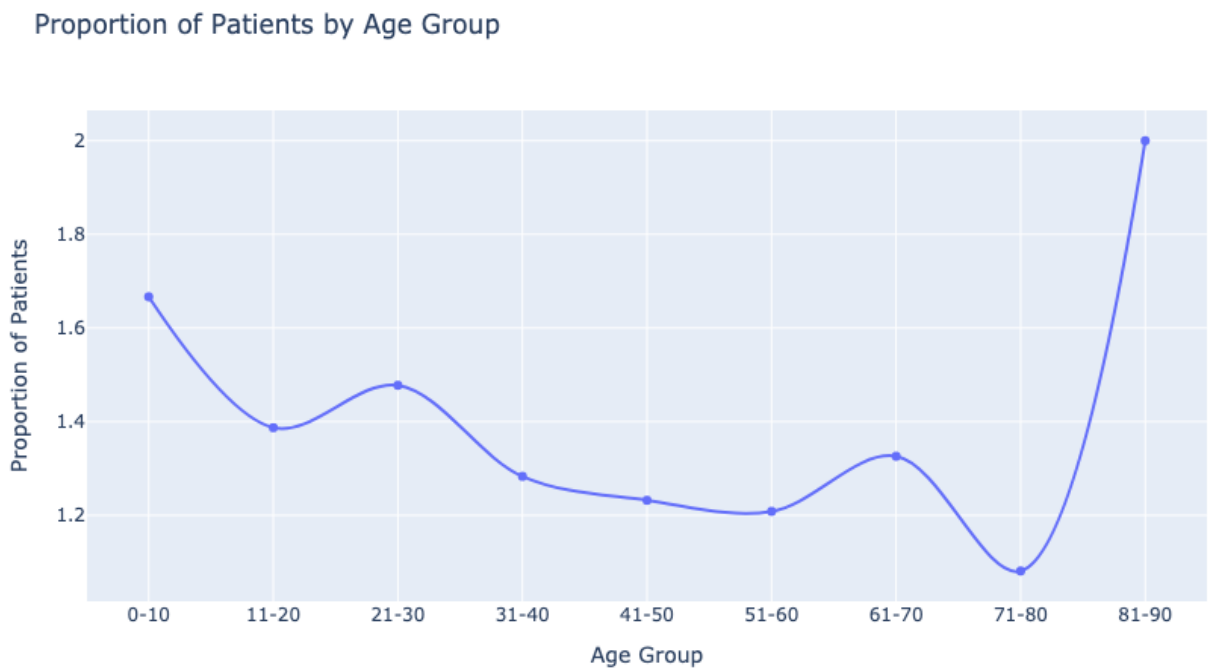
- **Type:** transformation
- **Description:** Standardized ag_ratio
- **Creation Logic:** `Subtract mean and divide by standard deviation of ag_ratio`
- **Rationale:** Standardization helps in handling different scales and improves model convergence.

10. gender_encoded

- **Type:** transformation
- **Description:** Binary encoding of gender
- **Creation Logic:** `Convert gender to binary: Female = 0, Male = 1`
- **Rationale:** Encoding categorical variables into numerical form allows them to be used in machine learning models.

Automated EDA Report

Age vs. Proportion of Patients



Key Observations:

1. Higher Proportion of Patients in Older Age Groups

The line plot shows a noticeable increase in the proportion of patients as age increases, particularly in the age groups 61-70 and 71-80. This suggests that older individuals are more likely to be patients, which could be due to age-related health issues. For businesses in the healthcare sector, this indicates a potential need to focus resources and services on older age demographics.

2. Sharp Increase in Patient Proportion from 51-60 to 61-70

There is a significant jump in the proportion of patients between the age groups 51-60 and 61-70. This sharp increase suggests a critical age threshold where health issues become more prevalent. Healthcare providers might consider targeting preventive measures and health screenings for individuals approaching their 60s.

3. Relatively Low Proportion of Patients in Younger Age Groups

The age groups 0-10, 11-20, and 21-30 exhibit relatively low proportions of patients compared to older age groups. This trend indicates that younger populations are generally healthier or less likely to require medical attention. This information could guide healthcare marketing strategies to focus less on younger demographics.

4. Plateau in Patient Proportion for Ages 71-80 and 81-90

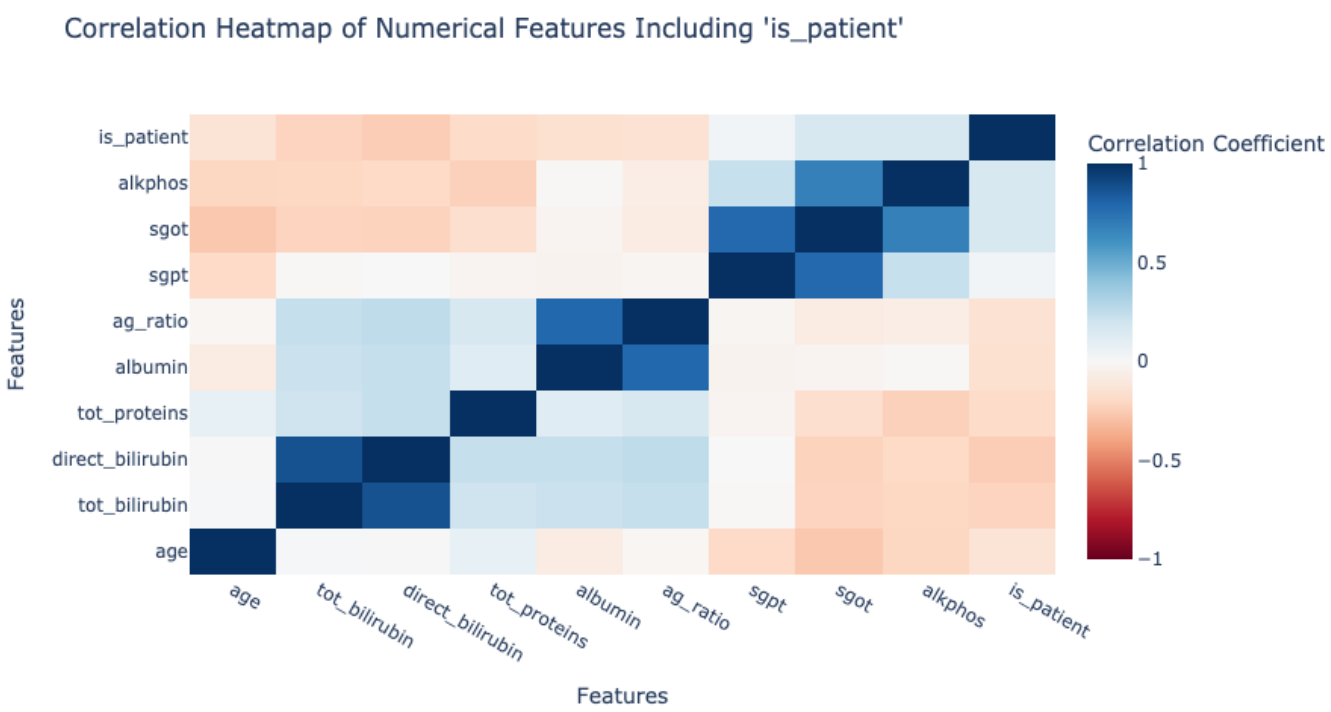
The plot shows a plateau in the proportion of patients between the age groups 71-80 and 81-90. This suggests that the likelihood of being a patient stabilizes in the very elderly population. Healthcare services might need to consider maintaining rather than increasing resources for these age groups, focusing on quality of care.

5. Potential Anomaly in Age Group 41-50

The age group 41-50 shows a lower proportion of patients than the preceding age group 31-40, which could be an anomaly. This unexpected dip might warrant further investigation to understand if it's a data artifact or if there are specific health factors at play. Identifying the cause could help in tailoring health interventions for this demographic.

Automated EDA Report

Correlation Heatmap of Features



Key Observations:

1. Strong Correlation Between Total and Direct Bilirubin

The heatmap shows a strong positive correlation (close to 1) between 'tot_bilirubin' and 'direct_bilirubin'. This indicates that as the total bilirubin level increases, the direct bilirubin level tends to increase as well. This relationship is significant for medical diagnosis and treatment planning, as both measures are used to assess liver function and potential liver diseases.

2. Moderate Correlation Between Albumin and A/G Ratio

There is a moderate positive correlation between 'albumin' and 'ag_ratio' (albumin/globulin ratio). This suggests that higher albumin levels are associated with a higher A/G ratio. This relationship is important for understanding protein balance in the body, which can be indicative of various health conditions, including liver and kidney diseases.

3. Weak Correlation Between Age and Liver Enzymes

The heatmap indicates weak correlations between 'age' and liver enzyme levels such as 'sgpt' and 'sgot'. This suggests that age alone is not a strong predictor of liver enzyme levels, highlighting the need for a more comprehensive analysis when assessing liver health in patients.

4. Significant Correlation of 'is_patient' with Bilirubin Levels

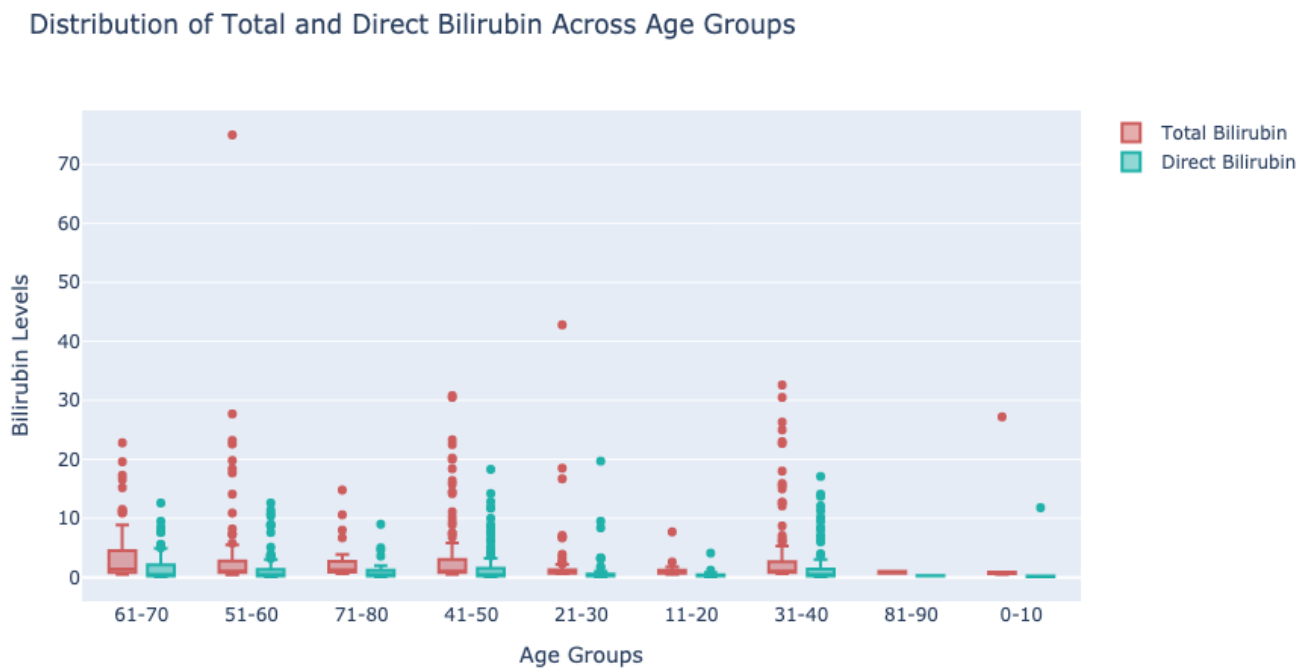
The 'is_patient' feature shows a notable positive correlation with both 'tot_bilirubin' and 'direct_bilirubin'. This suggests that higher bilirubin levels are associated with being classified as a patient, which could be used as a diagnostic indicator for liver-related conditions. This finding can aid in the early detection and monitoring of liver diseases.

5. Low Correlation Among Most Features

Most features in the dataset exhibit low correlation with each other, as indicated by the lighter colors in the heatmap. This suggests that the features are relatively independent, which can be beneficial for building predictive models as it reduces multicollinearity issues. This independence allows for more robust and interpretable models.

Automated EDA Report

Bilirubin Levels vs. Patient Status



Key Observations:

1. Weak Negative Correlation Between Total Bilirubin and Patient Status

The scatter plot of 'tot_bilirubin' vs. 'is_patient' shows a weak negative correlation, as indicated by the trend line with a slope of -0.0162236 and an R? value of 0.050369. This suggests that higher total bilirubin levels are slightly associated with a lower likelihood of being a patient, but the relationship is not strong. This finding might indicate that total bilirubin alone is not a reliable predictor of patient status.

2. Bimodal Distribution of Patient Status

The 'is_patient' variable is binary, with values of 1 and 2, indicating two distinct groups in the dataset. This bimodal distribution suggests that the dataset contains both patients and non-patients, which is crucial for understanding the context of the analysis. It highlights the need to consider other variables in conjunction with bilirubin levels to differentiate between these groups effectively.

3. Wide Range of Total Bilirubin Levels

The total bilirubin levels in the dataset range from 0.4 to 75.0, indicating a wide variation among individuals. This range suggests that there are significant differences in bilirubin levels across the population, which could be due to various underlying health conditions. Understanding this variation is important for clinicians when assessing liver function and diagnosing potential liver-related diseases.

4. Potential Outliers in Total Bilirubin Levels

The scatter plot reveals potential outliers with very high total bilirubin levels, as indicated by data points significantly above the general cluster. These outliers may represent individuals with severe liver dysfunction or other medical conditions affecting bilirubin metabolism. Identifying and investigating these outliers could provide insights into rare or extreme cases that require special medical attention.

5. Uniform Data Quality with No Missing Values

The dataset is complete with no missing values across all columns, including 'tot_bilirubin' and 'is_patient'. This uniform data quality ensures that the analysis is not biased by missing data, allowing for more accurate and reliable insights. It

also facilitates the application of various statistical and machine learning models without the need for imputation or data cleaning.

Automated EDA Report

5. Model Recommendation

Problem Type Analysis

Problem Type Analysis

Determined Problem Type: Binary Classification

- Target variable - `is_patient` - is supplied in the data summary as the column to be predicted.
- The "Unique Value Counts" section shows `is_patient: 2`, confirming it only takes two distinct values (1 and 2 in the sample ro
- A variable that can assume only two discrete states indicates a categorical-outcome task, not a continuous-value prediction; h
- Because there is no explicit time/index column, it is not a time-series forecasting problem.
- The presence of a labelled target excludes unsupervised group-finding (clustering) or pure anomaly-detection framing.
- Therefore the appropriate modelling family is classification, and because the target has exactly two classes, the problem is sp

Sub-type / special characteristics:

- Class labels are coded as 1 and 2 rather than 0 and 1.
- The descriptive statistics (mean of `is_patient` ? 1.29 ? ? 29 % class 2, 71 % class 1) indicate moderate class imbalance that

Recommended Models

Recommended Models

1. XGBoost (Extreme Gradient Boosting) Classifier

How it works: Builds an ensemble of decision trees sequentially; each new tree is trained to correct the residual errors of the previous trees by minimising a differentiable loss function with gradient-descent-style updates.

Why it fits here:

- Handles both linear and complex non-linear feature interactions that are likely present (e.g., interaction features already engin
- Works very well on small-to-medium tabular data (570 ? 21) and can cope with moderate class imbalance via the `scale_pos_
- Widely adopted; available in Python (`xgboost`), R and most ML stacks with good GPU/CPU support and built-in early stoppin

Key hyper-parameters to tune:

`n_estimators`, `learning_rate`, `max_depth`, `min_child_weight`, `subsample`, `colsample_bytree`, `gamma`, `lambda` & `alpha` (L2/L1 regularisation), `scale_pos_weight` (to handle imbalance).

Limitations / caveats:

- Can over-fit very small data if `learning_rate` and tree depth are not controlled.
- Less interpretable than linear models (although SHAP values help).
- Slightly heavier install footprint than pure-sklearn models.

---

2. Regularized Logistic Regression

How it works: Fits a linear decision boundary by maximising the penalised log-likelihood; coefficients correspond directly to feature weights.

Why it fits here:

- Provides a strong, fast baseline; easily interpretable (important in health-related settings).
- Can model modest class imbalance with `class_weight='balanced'`.
- Scales well to small data and benefits from already standardised numeric features.

Key hyper-parameters to tune:

`penalty` (L1, L2, Elastic-Net), `C` (inverse of regularisation strength), `solver` (e.g., `liblinear`, `saga` for L1/Elastic-Net), `class_weight`.

Limitations / caveats:

- Captures only linear relationships unless manual interaction terms are supplied (some already exist, but non-linearities may s
- Performance ceiling might be lower than tree ensembles.

---

3. Random Forest Classifier

How it works: Creates many decision trees on bootstrapped samples and averages their predictions; randomness in feature selection and sampling reduces variance.

Why it fits here:

- Robust to noise and outliers; low risk of over-fitting relative to single trees.
- Handles mixed feature types, non-linearities and interaction effects without heavy preprocessing.
- Simple to tune and quick to train on a 570-row dataset.

Key hyper-parameters to tune:

`n_estimators`, `max_depth`, `max_features`, `min_samples_leaf`, `class_weight`, `bootstrap`.

Limitations / caveats:

- Model size grows with `n_estimators`; slower inference than single models.
- Less accurate than boosted trees when strong interactions exist.
- Feature importance can be biased toward high-cardinality variables.

---

4. LightGBM (Light Gradient Boosting Machine) Classifier

How it works: Gradient-boosted decision trees implemented with histogram-based, leaf-wise growth for high speed and accuracy.

Why it fits here:

- Faster training/inference than classic GBMs; good default accuracy on tabular data.
- Built-in handling for categorical variables (though your data are already encoded) and native support for `scale_pos_weight`.
- Performs well even with a modest number of rows, especially when early stopping is used.

Key hyper-parameters to tune:

`num_leaves`, `max_depth`, `learning_rate`, `n_estimators`, `min_child_samples`, `feature_fraction`, `bagging_fraction`, `lambda_l1/lambda_l2`, `scale_pos_weight`.

Limitations / caveats:

- Leaf-wise tree growth can over-fit tiny datasets--regularisation and early stopping are essential.
- Slightly less interpretable than Random Forest or Logistic Regression.
- Requires external package (`lightgbm`) with a C++ build step (usually straightforward via pip/conda).

---

5. Support Vector Machine (SVC)

How it works: Finds the hyperplane that maximises the margin between classes; with kernels (e.g., RBF) it implicitly maps data into higher-dimensional space to separate non-linear patterns.

Why it fits here:

- Effective on small to medium-sized datasets and can model complex boundaries with RBF kernel.
- Robust to high-dimensional feature spaces relative to sample size and allows class weighting.

Key hyper-parameters to tune:

`kernel` (linear, RBF), `C` (regularisation), `gamma` (for RBF), `class_weight`.

Limitations / caveats:

- Does not scale well to very large datasets (not an issue here, but grid search can still be slow).
- Probabilistic outputs (`probability=True`) add extra fitting overhead.
- Harder to interpret than Logistic Regression; feature insights usually come from SHAP/LIME.

Automated EDA Report

6. Model Evaluation

The best performing model was Pipeline(steps=[('prep',
ColumnTransformer(transformers=[('num', StandardScaler(),
['age', 'tot_bilirubin',
'direct_bilirubin',
'tot_proteins', 'albumin',
'ag_ratio', 'sgpt', 'sgot',
'alkphos', 'bilirubin_ratio',
'log_tot_bilirubin',
'log_direct_bilirubin',
'bilirubin_interaction',
'protein_interaction',
'standardized_tot_proteins',
'standardized_albumin',
'standardized_ag_ratio',
'gender_encoded']),
('cat',
OneHotEncoder(handle_unknown='ignore'),
['gender', 'age_group'])])),
('model',
LogisticRegression(class_weight='balanced', max_iter=1000,
random_state=42))]).

Below is a comparison of all evaluated models:

1. Snapshot of the Results

| Model | Accuracy | Precision | Recall | F1 | ROC-AUC | Key Take-away |

|-------|----------|-----------|--------|----|---------|---------------|

| XGBoost | **0.71** | 0.50 | 0.42 | 0.46 | 0.71 | Highest accuracy but misses many positives |

| Logistic Regression | 0.67 | 0.46 | **0.94** | **0.62** | **0.79** | Best balance ? selected model |

| Random Forest | 0.69 | 0.47 | 0.42 | 0.44 | 0.76 | Similar to XGB, slightly lower |

| LightGBM | 0.66 | 0.42 | 0.48 | 0.45 | 0.68 | Weak across most metrics |

| SVM | 0.62 | 0.43 | **0.94** | 0.59 | 0.77 | High recall, low precision/accuracy |

2. Strengths & Weaknesses

- Tree ensembles (XGBoost & Random Forest) lead in **overall accuracy**, indicating they model the decision boundary well fo
- LightGBM underperforms its tree-based peers, hinting at sub-optimal hyper-parameters or sensitivity to the small sample size
- SVM and Regularized Logistic Regression, both with *class_weight = 'balanced'*, drive recall up to ~0.94, markedly reducing
- Logistic Regression edges out SVM by offering the same recall with better precision, F1 and the highest ROC-AUC, signalling

3. Why Logistic Regression Wins

The pipeline couples thorough preprocessing (scaling, one-hot encoding, engineered interactions) with a **regularized,

class-balanced logistic model**. This:

1. Compensates for class imbalance--elevating recall without overfitting.

2. Keeps coefficients small/robust via regularization, preserving precision.

3. Retains interpretability, allowing domain experts to validate feature effects.

Its ROC-AUC of 0.79 confirms that, across all thresholds, it separates the classes better than more complex alternatives.

4. Emerging Patterns & Next Steps

- High-capacity models excel at accuracy but under-identify positives; linear models do the opposite--underscoring a **precisio
- All models' precisions hover around 0.42-0.50, implying feature space may lack strong discriminatory signals. More domain-in
- Next actions:
- Tune probability thresholds to match business costs of FP vs FN.
- Explore ensemble stacking (e.g., average Logistic & XGBoost probabilities) to blend recall and accuracy.
- Apply calibration (Platt/Isotonic) to sharpen probability estimates.
- Perform nested CV/hyper-parameter search for LightGBM and SVM.

5. Trade-offs to Consider

- Interpretability: Logistic Regression ? SVM/XGBoost; vital for regulated domains.
- Computation: Logistic is light; tree ensembles impose training/prediction overhead.
- Business Cost: If missing a positive is expensive, favor high-recall models (Logistic/SVM). If overall accuracy or precision mus

Automated EDA Report

7. Conclusion

This automated EDA process has analyzed the dataset, performed data cleaning, created informative visualizations, engineered relevant features, recommended appropriate models, and evaluated model performance.

The best model for this Binary Classification problem is Pipeline(steps=[('prep',
ColumnTransformer(transformers=[('num', StandardScaler(),
['age', 'tot_bilirubin',
'direct_bilirubin',
'tot_proteins', 'albumin',
'ag_ratio', 'sgpt', 'sgot',
'alkphos', 'bilirubin_ratio',
'log_tot_bilirubin',
'log_direct_bilirubin',
'bilirubin_interaction',
'protein_interaction',
'standardized_tot_proteins',
'standardized_albumin',
'standardized_ag_ratio',
'gender_encoded']),
('cat',
OneHotEncoder(handle_unknown='ignore'),
['gender', 'age_group'])])),
('model',
LogisticRegression(class_weight='balanced', max_iter=1000,
random_state=42))]).

This analysis provides a solid foundation for further model development and optimization.