



Course: AI with Deep Learning

Liver Disease Prediction

Evaluating prediction algorithms in an effort to reduce burden on doctors

Students: Dendić Damjan & Jevtić Neda

Presentation date: 27th May, 2024

Course coordinators:

Tome Eftimov, Nada Lavrač, Uroš Godnov

Jožef Stefan Institute

SEB UL



Presentation Overview

1. Introduction
2. Data explanation and EDA
3. Modelling
4. Evaluation of ML models
5. Summary & Proposals



INTRODUCTION

Context

Patients with liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food and drugs. Thus, we wanted to use the chosen dataset to evaluate prediction algorithms in an effort to reduce burden on doctors.

Content

This data set contains 416 liver patient records and 167 non liver patient records collected from North East of Andhra Pradesh, India. The "Dataset" column is a class label used to divide groups into liver patient (liver disease) or not (no disease). This data set contains 441 male patient records and 142 female patient records.

INTRODUCTION

Features

- Age
- Gender
- Total Bilirubin (high levels indicate liver dysfunction)
- Direct Bilirubin
- Alkaline Phosphotase (enzyme, high levels may be a sign of a liver problem)
- Alamine Aminotransferase
- Aspartate Aminotransferase
- Total Proteins
- Albumin (protein in blood plasma, low albumin levels might be the result of liver disease)
- Albumin and Globulin Ratio (High A/G ratio: can be a sign of disease in liver, kidney, or intestines)
- Dataset - class label (liver disease (1) & no liver disease (0))

Encoding of categorical variable

Data explanation & EDA

```
Data columns (total 11 columns):
```

#	Column	Non-Null Count	Dtype
0	Age	583 non-null	int64
1	Gender	583 non-null	object
2	Total_Bilirubin	583 non-null	float64
3	Direct_Bilirubin	583 non-null	float64
4	Alkaline_Phosphotase	583 non-null	int64
5	Alamine_Aminotransferase	583 non-null	int64
6	Aspartate_Aminotransferase	583 non-null	int64
7	Total_Protiens	583 non-null	float64
8	Albumin	583 non-null	float64
9	Albumin_and_Globulin_Ratio	579 non-null	float64
10	Dataset	583 non-null	int64

dtypes: float64(5), int64(5), object(1)

Class label (1- liver disease; 0- no disease)

```
df.Dataset.value_counts()
```

```
Dataset
1    416
0    167
Name: count, dtype: int64
```

```
# Categorical value Handling
def convertgender(x):
    if x== 'Male':
        return 0
    else:
        return 1
df['Gender'] = df['Gender'].map(convertgender)
```

Original class distribution: Counter({1: 402, 0: 162})



Classification baseline accuracy: 0.71
Resampled class distribution: Counter({1: 281, 0: 281})

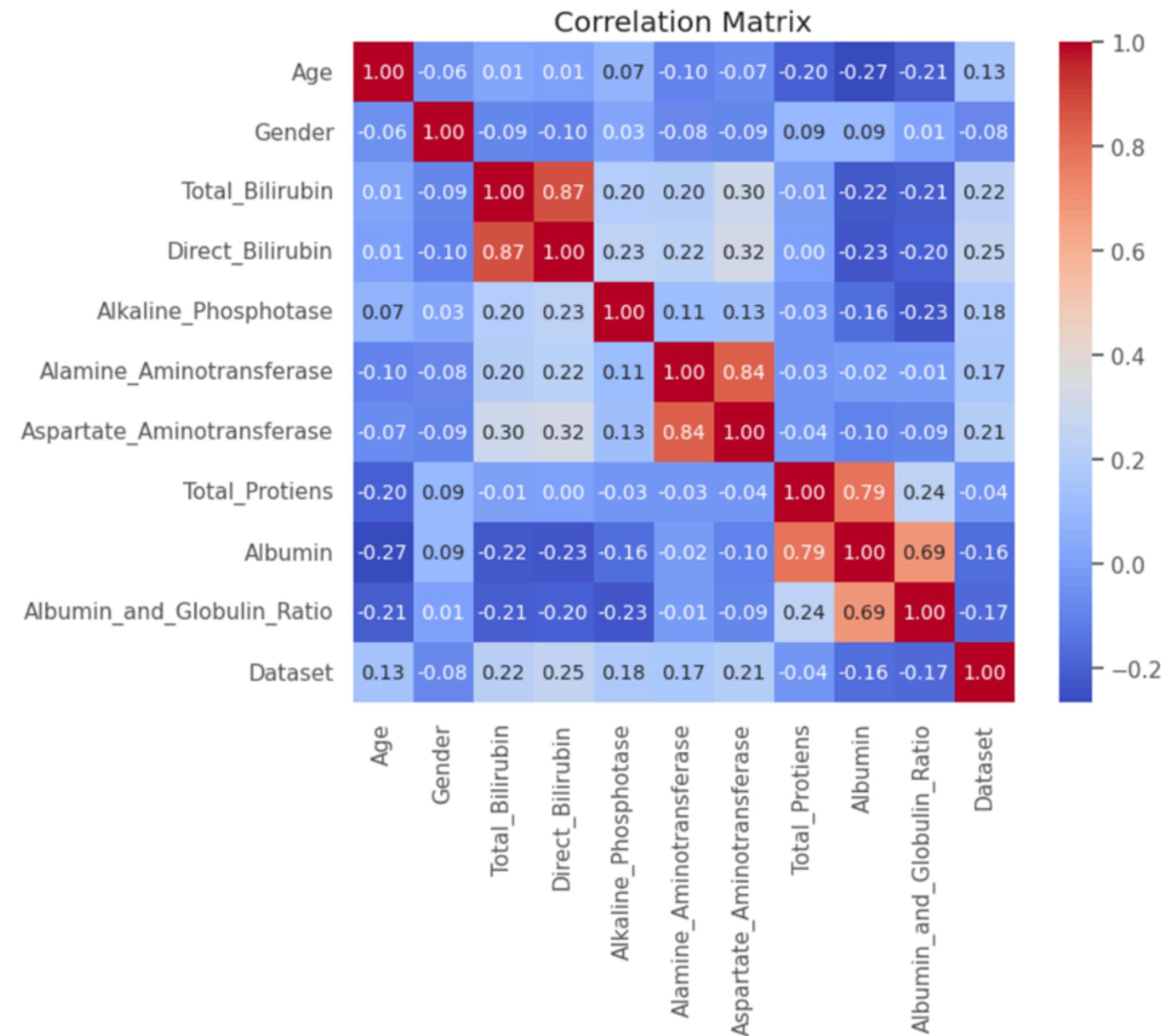
We decided to apply the upsampling technique (oversampling the minority class) to balance the dataset. This will help our model to learn from the minority class more effectively.

Data explanation & EDA

The strong correlations between certain enzymes (ALT and AST) and bilirubin levels (Total and Direct) suggest that these features together can provide a more accurate diagnosis of liver disease.

Understanding these correlations can help clinicians focus on key diagnostic tests and interpret multiple test results in context, leading to better-informed clinical decisions.

The correlation matrix reveals that **Total_Bilirubin**, **Direct_Bilirubin**, **Alamine_Aminotransferase** (ALT), **Aspartate_Aminotransferase** (AST), and **Alkaline_Phosphotase** (ALP) are among the best features for predicting liver disease. Strong relationships between these variables highlight the interdependence of liver function indicators, enhancing our understanding and improving diagnostic accuracy.



Data explanation & EDA

Removing Outliers

```
▶ df.Aspartate_Aminotransferase.sort_values(ascending=False).head()
```

```
→ 135    4929  
  117    2946  
  118    1600  
  207    1500  
  199    1050  
Name: Aspartate_Aminotransferase, dtype: int64
```

```
[33] df = df[df.Aspartate_Aminotransferase<=3000]
```

```
[35] df.Aspartate_Aminotransferase.sort_values(ascending=False).head()
```

```
→ 117    2946  
  118    1600  
  207    1500  
  119    1050  
  199    1050  
Name: Aspartate_Aminotransferase, dtype: int64
```

```
[36] df = df[df.Aspartate_Aminotransferase<=2500]
```

Dealing with missing values (there were only 4 missing values in the column “Albumin and Globulin Ratio”)

```
▶ df.isnull().sum()
```

```
→ Age          0  
  Gender       0  
  Total_Bilirubin 0  
  Direct_Bilirubin 0  
  Alkaline_Phosphotase 0  
  Alamine_Aminotransferase 0  
  Aspartate_Aminotransferase 0  
  Total_Protiens 0  
  Albumin      0  
  Albumin_and_Globulin_Ratio 4  
  Dataset       0  
  dtype: int64
```

```
[39] df = df.dropna(how='any')
```

Data preparation, calculating the baseline for majority class, SMOTE

```
[44] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)
```

```
[45] def calculate_classification_baseline(y_train):
    majority_class = y_train.mode()[0]
    majority_class_prob = (y_train == majority_class).mean()
    return majority_class, majority_class_prob
```

```
majority_class, majority_class_prob = calculate_classification_baseline(y_train)
print(f"Classification baseline accuracy: {majority_class_prob:.2f}")
```

→ Classification baseline accuracy: 0.71

```
[92] # Handle Class Imbalance using SMOTE
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)
print(f"Resampled class distribution: {Counter(y_train_resampled)}")
```

→ Resampled class distribution: Counter({1: 281, 0: 281})

Modeling - Logistic Regression, Random Forest, Decision Tree, KNeighbour Classifier, SVC (Support Vector Classifier), Gradient Boosting Classifier

Logistic Regression

Cross-validated accuracy: 0.69 (Baseline: 0.71)

Training accuracy: 0.70

Validation accuracy: 0.69

Test set accuracy: 0.63

Test set precision: 0.72

Test set recall: 0.63

Test set F1-score: 0.65

Robustness check across multiple splits:

Mean accuracy: 0.66 ± 0.03

Mean precision: 0.74 ± 0.02

Mean recall: 0.66 ± 0.03

Mean F1-score: 0.68 ± 0.03

Random Forest

Cross-validated accuracy: 0.77 (Baseline: 0.71)

Training accuracy: 1.00

Validation accuracy: 0.77

Test set accuracy: 0.74

Test set precision: 0.74

Test set recall: 0.74

Test set F1-score: 0.74

Robustness check across multiple splits:

Mean accuracy: 0.68 ± 0.03

Mean precision: 0.69 ± 0.03

Mean recall: 0.68 ± 0.03

Mean F1-score: 0.68 ± 0.03

Decision Tree

Cross-validated accuracy: 0.72 (Baseline: 0.71)

Training accuracy: 1.00

Validation accuracy: 0.72

Test set accuracy: 0.66

Test set precision: 0.67

Test set recall: 0.66

Test set F1-score: 0.67

Robustness check across multiple splits:

Mean accuracy: 0.64 ± 0.03

Mean precision: 0.66 ± 0.03

Mean recall: 0.64 ± 0.03

Mean F1-score: 0.65 ± 0.03

KNeighbourClassifier

Cross-validated accuracy: 0.67 (Baseline: 0.71)

Training accuracy: 0.85

Validation accuracy: 0.67

Test set accuracy: 0.57

Test set precision: 0.65

Test set recall: 0.57

Test set F1-score: 0.59

Robustness check across multiple splits:

Mean accuracy: 0.59 ± 0.04

Mean precision: 0.67 ± 0.03

Mean recall: 0.59 ± 0.04

Mean F1-score: 0.61 ± 0.03

SVC

Cross-validated accuracy: 0.68 (Baseline: 0.71)

Training accuracy: 0.83

Validation accuracy: 0.68

Test set accuracy: 0.64

Test set precision: 0.70

Test set recall: 0.64

Test set F1-score: 0.66

Robustness check across multiple splits:

Mean accuracy: 0.63 ± 0.02

Mean precision: 0.72 ± 0.03

Mean recall: 0.63 ± 0.02

Mean F1-score: 0.64 ± 0.02

Gradient Boosting Classifier

Cross-validated accuracy: 0.70 (Baseline: 0.71)

Training accuracy: 1.00

Validation accuracy: 0.70

Test set accuracy: 0.69

Test set precision: 0.69

Test set recall: 0.69

Test set F1-score: 0.69

Robustness check across multiple splits:

Mean accuracy: 0.68 ± 0.03

Mean precision: 0.69 ± 0.03

Mean recall: 0.68 ± 0.03

Mean F1-score: 0.68 ± 0.03

Summary & Proposals

In summary, the choice between accuracy, precision, and recall depends on the specific requirements and implications of the prediction task. For liver disease prediction, while accuracy is generally important, precision and recall may be more crucial depending on the relative costs and consequences of false positives and false negatives in the given context.

For instance, in medical diagnosis, missing a positive case (low recall) could be more critical than mistakenly diagnosing a healthy person as sick (low precision).

By incorporating this predictive model into regular health check-ups, medical professionals can identify individuals who are at high risk of developing liver disease before symptoms become severe.

Healthcare facilities can use the model to prioritize patients who need immediate attention and monitoring.

Health authorities can use aggregated data from the model to monitor liver disease trends across different regions and populations, identifying hotspots and demographic groups at higher risk.

**Thank you
for your
attention!**

