



PRAKTIKUM ARTIFICIAL INTELLIGENCE

Kelompok 21

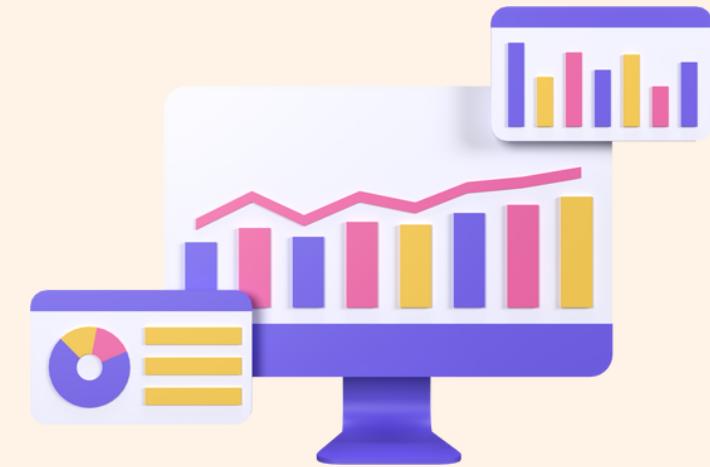
13522013 - Denise Felicia Tiowanni

13522053 - Erdianti Wiga Putri Andini

WHAT'S INSIDE?



Exploratory Data Analysis



**Splitting, Cleaning, and
Preprocessing**

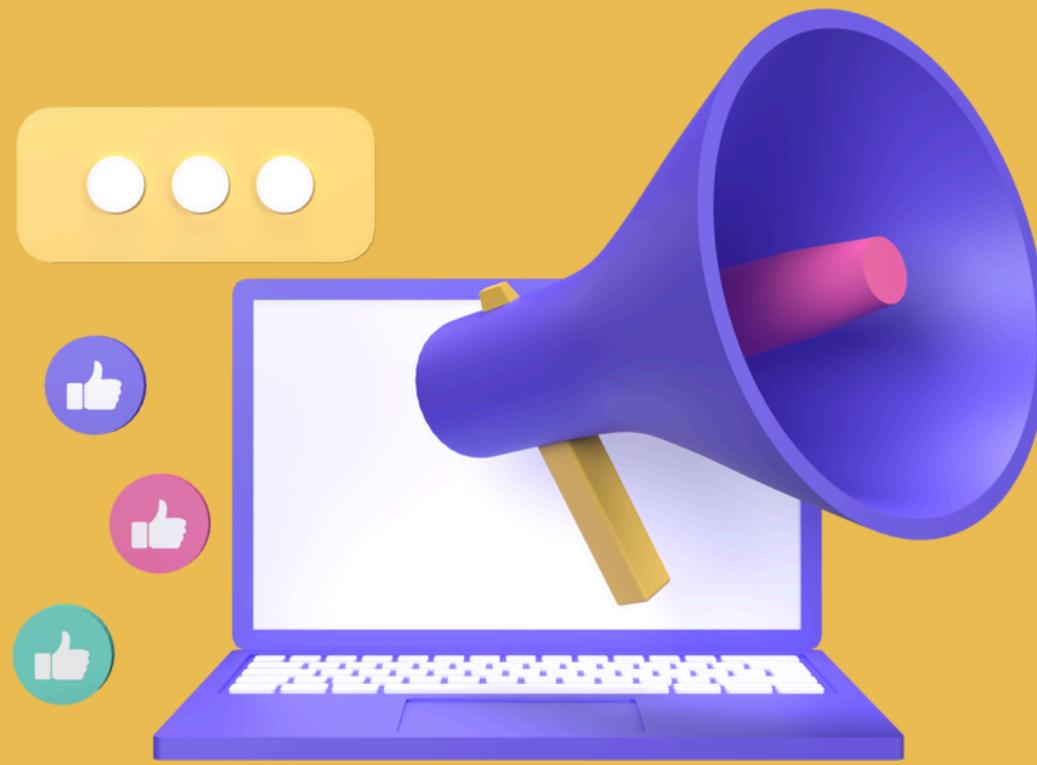


Modeling and Validation



Error Analysis

EXPLORATORY DATA ANALYSIS



DATA SIZE AND TYPES

----- DATA SIZE -----

Instances Features

0 15000 20

Total data dari dataset **train** ini adalah 15000 data dengan 20 atribut

----- DATA TYPES -----	
0	int64
N_Days	float64
Drug	object
Age	float64
Sex	object
Ascites	object
Hepatomegaly	object
Spiders	object
Edema	object
Bilirubin	float64
Cholesterol	float64
Albumin	float64
Copper	float64
Alk_Phosphatase	float64
SGOT	float64
Tryglicerides	float64
Platelets	float64
Prothrombin	float64
Stage	float64
Status	object

Berdasarkan metadata pada spesifikasi, terdapat **8 kolom** dari 20 kolom yang merupakan *categorical data* yaitu Drug, Sex, Ascites, Hepatomegaly, Spiders, Edema, Stage, dan Status. Kolom sisanya adalah *numerical data*.

CATEGORICAL DATA

---- UNIQUE COUNTS (CATEGORICAL) ----		
	Category	Unique Counts
0	Drug	2
1	Sex	2
2	Ascites	2
3	Hepatomegaly	3
4	Spiders	2
5	Edema	3
6	Stage	4
7	Status	3

Jumlah *unique value* pada masing-masing kolom categorical

DRUG	Count	Proportion (%)
Drug		
D-penicillamine	4515	53.431953
Placebo	3935	46.568047
ASCITES	Count	Proportion (%)
Ascites		
N	8027	94.960369
Y	426	5.039631
SPIDERS	Count	Proportion (%)
Spiders		
N	6598	78.166094
Y	1843	21.833906
SEX	Count	Proportion (%)
Sex		
F	14405	96.033333
M	595	3.966667
Stage		
3.0	5530	36.866667
4.0	5235	34.900000
2.0	3456	23.040000
1.0	779	5.193333
Hepatomegaly		
N	4652	55.066288
Y	3795	44.921875
158.0	1	0.011837
Edema		
N	13874	92.493333
S	737	4.913333
Y	389	2.593333
Status		
C	10117	67.446667
D	4525	30.166667
CL	358	2.386667

Proporsi tiap *unique value* pada masing-masing kolom categorical

NUMERICAL DATA

----- SUMMARY STATISTICS (NUMERICAL) -----											
	N_Days	Age	Bilirubin	Cholesterol	Albumin	Copper	Alk_Phosphat	SGOT	Tryglicerides	Platelets	Prothrombin
min	10.800000	174.000000	0.300000	120.000000	1.960000	0.900000	3.410000	0.900000	11.000000	0.900000	4.160000
25%	1092.000000	16463.000000	0.700000	242.000000	3.290000	31.000000	721.000000	71.000000	80.000000	189.000000	10.100000
median	1783.000000	19577.000000	0.900000	280.000000	3.580000	52.000000	1072.000000	97.650000	99.000000	251.000000	10.600000
75%	2615.000000	22388.000000	1.800000	360.000000	3.770000	89.000000	1689.000000	130.200000	133.000000	312.000000	11.000000
max	25976.000000	91894.000000	28.000000	1790.000000	4.700000	604.000000	18102.000000	12285.000000	608.000000	3150.000000	18.000000
mean	1969.116920	19298.427133	1.862907	322.805916	3.520585	76.395725	1676.897656	108.214094	110.670418	254.266294	10.624837
std	1374.093322	3706.122475	2.763713	163.041117	0.373807	76.662546	1882.904216	152.099011	49.828478	96.081587	0.729726

Analisis nilai minimum, q1, median, q2, nilai maksimum, nilai rata-rata, dan nilai standar deviasi untuk masing-masing kolom *numerical data*

MISSING VALUES

----- Missing Values -----:		
	Missing Values	Percentage
id	0	0.000000
N_Days	0	0.000000
Drug	6550	43.666667
Age	0	0.000000
Sex	0	0.000000
Ascites	6547	43.646667
Hepatomegaly	6552	43.680000
Spiders	6559	43.726667
Edema	0	0.000000
Bilirubin	0	0.000000
Cholesterol	8374	55.826667
Albumin	0	0.000000
Copper	6660	44.400000
Alk_Phosphatase	6556	43.706667
SGOT	6559	43.726667
Tryglicerides	8425	56.166667
Platelets	584	3.893333
Prothrombin	16	0.106667
Stage	0	0.000000
Status	0	0.000000

Jumlah *missing values* pada tiap kolom, baik yang *numerical* maupun yang *categorical* beserta persentasenya dibandingkan dengan keseluruhan data pada masing-masing kolom.

Missing values ini berupa NaN.

OUTLIERS (NUMERICAL)

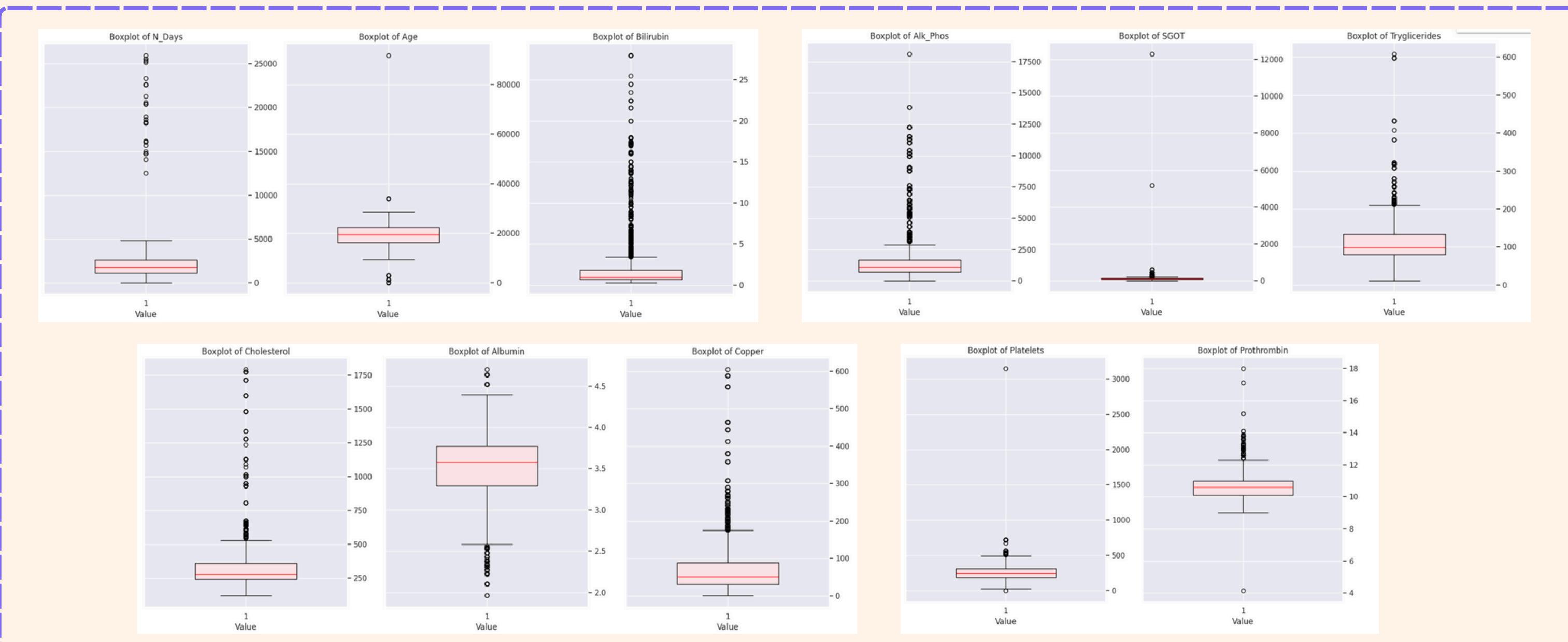
----- NUMERICAL DATA OUTLIERS BY IQR (True indicates an outlier) -----	
0	
N_Days	True
Age	True
Bilirubin	True
Cholesterol	True
Albumin	True
Copper	True
Alk_Phosphatase	True
SGOT	True
Tryglicerides	True
Platelets	True
Prothrombin	True

Identifikasi apakah kolom *numerical data* mengandung data yang berupa outlier atau tidak

----- JUMLAH OUTLIER PER KOLOM NUMERICAL -----	
0	
N_Days	35
Age	16
Bilirubin	1577
Cholesterol	384
Albumin	353
Copper	695
Alk_Phosphatase	845
SGOT	208
Tryglicerides	291
Platelets	136
Prothrombin	392

Jumlah data yang termasuk outlier pada masing-masing kolom *numerical*.

OUTLIERS (NUMERICAL)



Penggunaan boxplot untuk memvisualisasikan outlier pada masing-masing kolom *numerical*

OUTLIERS (CATEGORICAL)

RARE CATEGORIES IN NON-NUMERICAL DATA:

No rare categories detected in Drug that fall below the 1.0% threshold.

No rare categories detected in Sex that fall below the 1.0% threshold.

No rare categories detected in Ascites that fall below the 1.0% threshold.

Rare categories in HEPATOMEGALY (less than 1.0%):

proportion

Hepatomegaly

158.0	0.000118
-------	----------

dtype: float64

No rare categories detected in Spiders that fall below the 1.0% threshold.

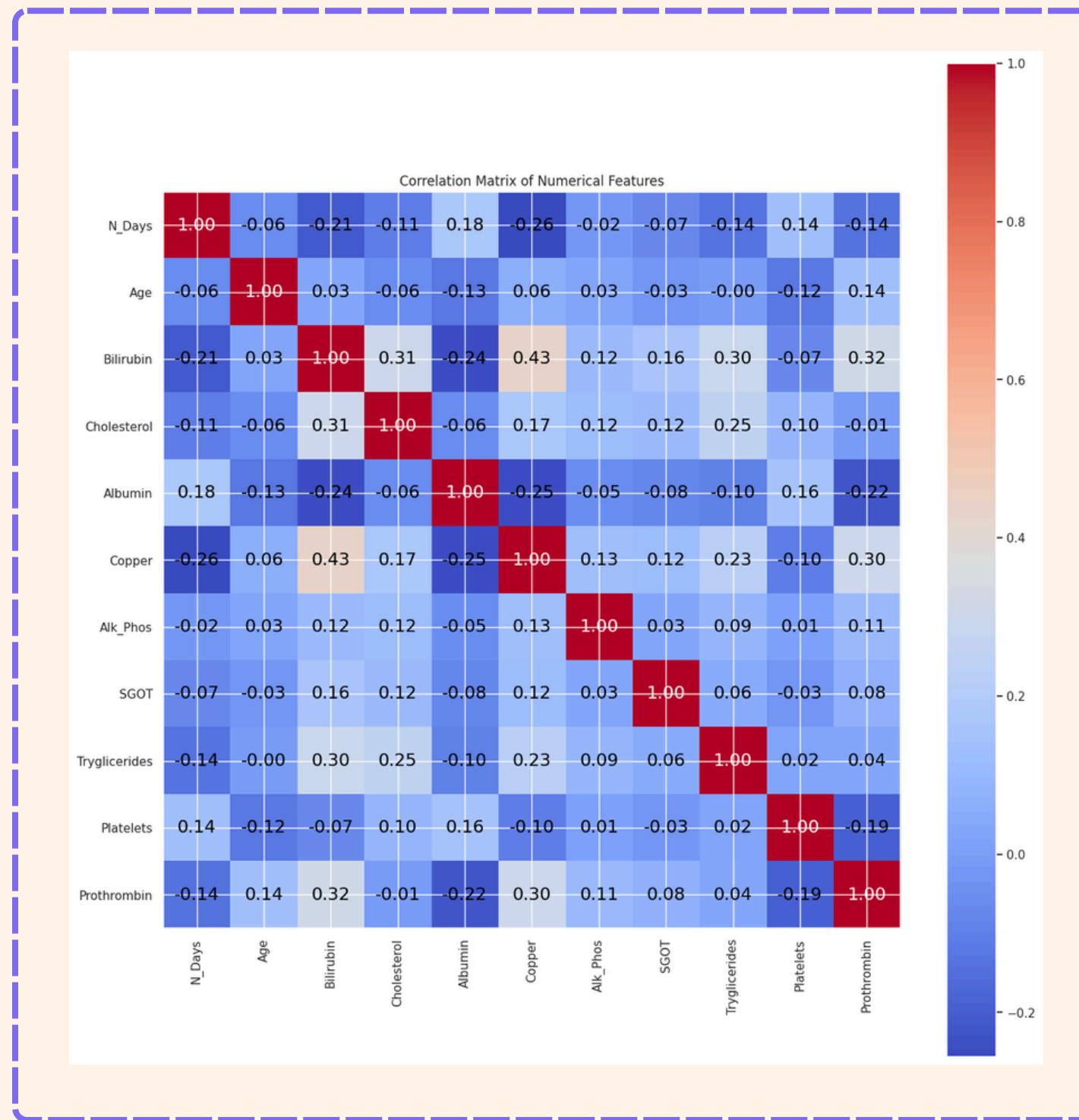
No rare categories detected in Edema that fall below the 1.0% threshold.

No rare categories detected in Status that fall below the 1.0% threshold.

No rare categories detected in Stage that fall below the 1.0% threshold.

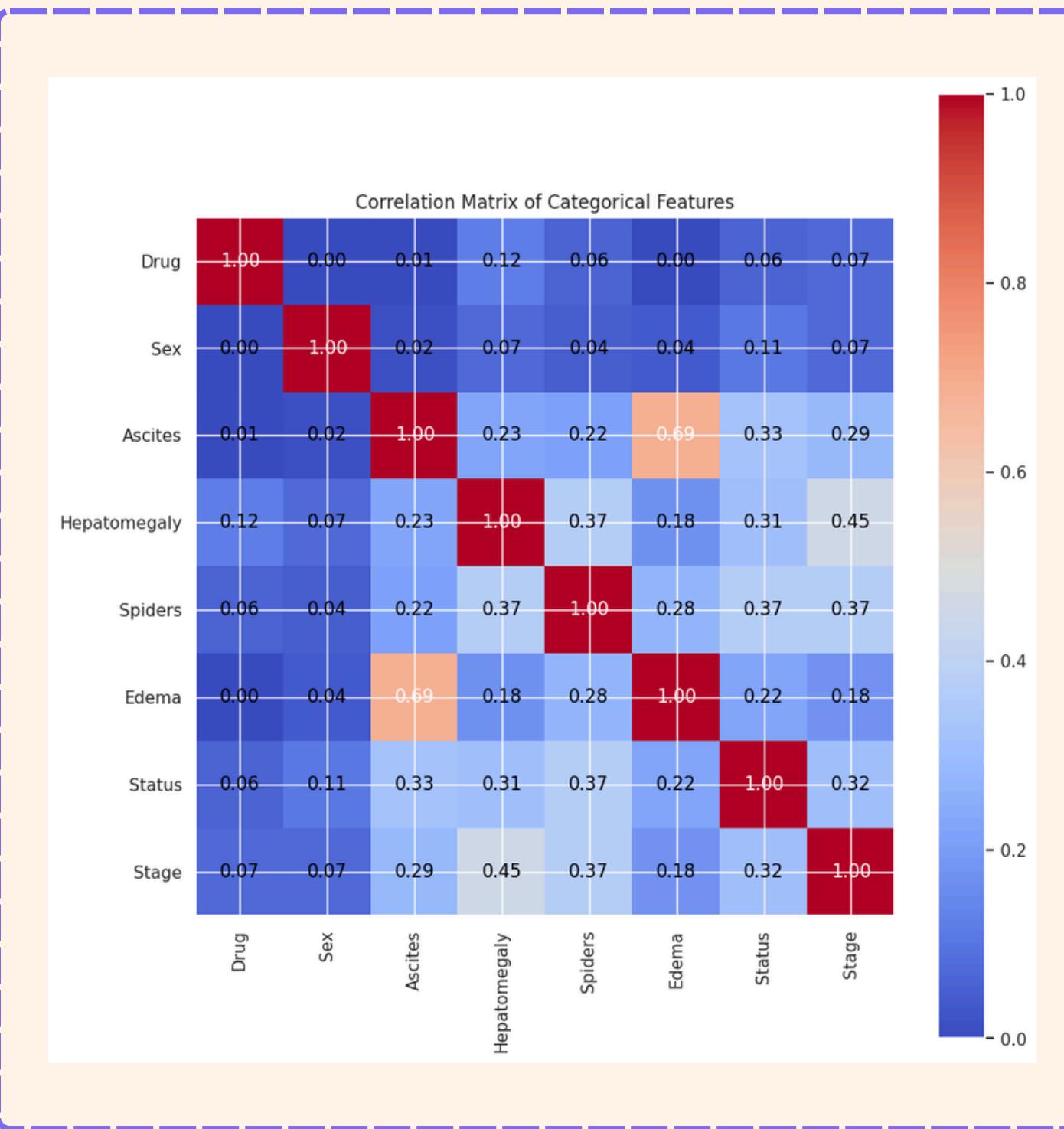
Mendeteksi *rare categories* pada *categorical data* dimana data ini adalah data-data yang berada di bawah treshold 1%

CORRELATION (NUMERICAL)



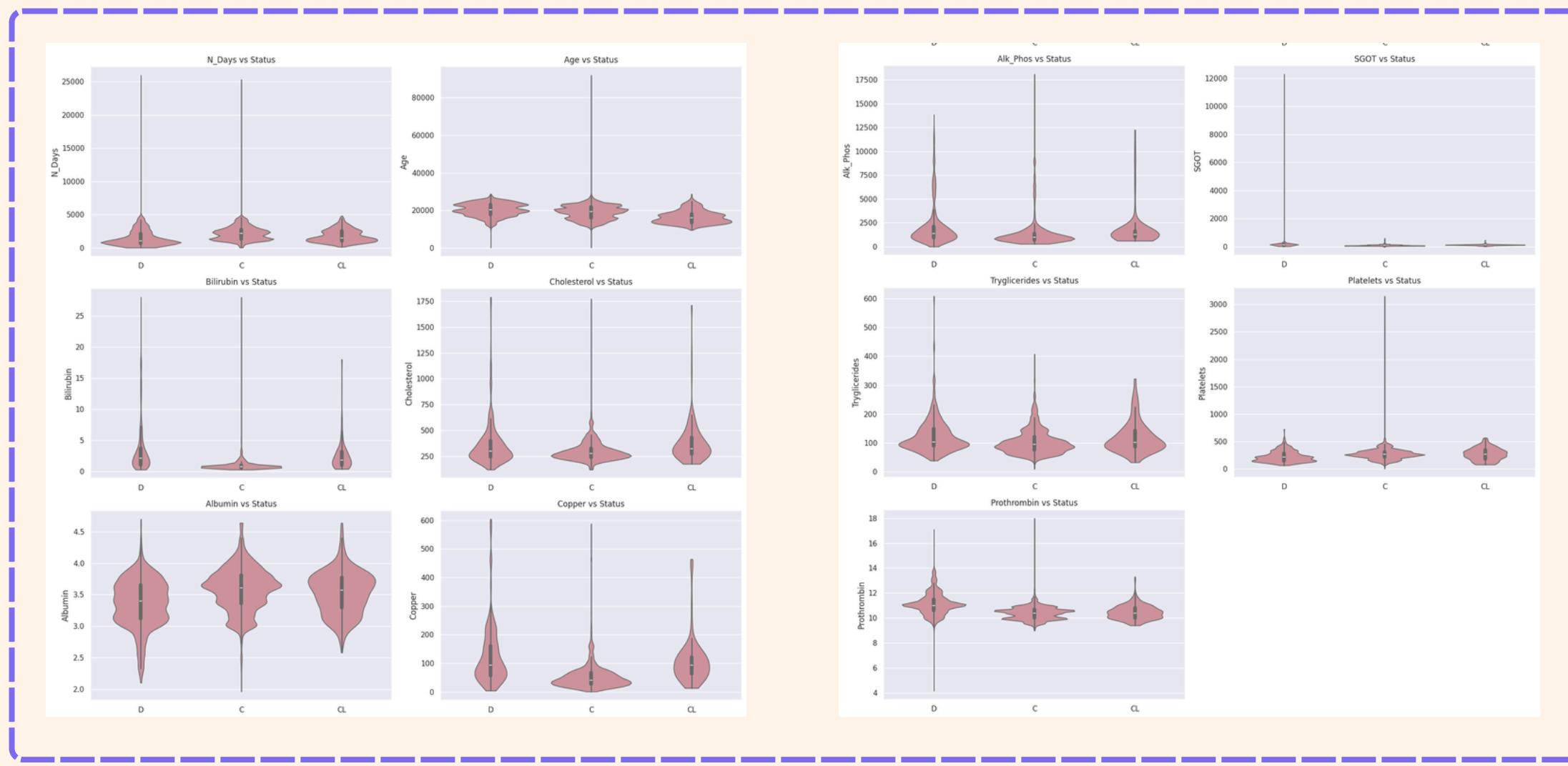
Correlation matrix ini menggambarkan hubungan antara berbagai fitur **numerical**, di mana warna **merah tua (1.0)** menunjukkan korelasi **positif kuat**, **biru tua (-1.0)** menunjukkan korelasi **negatif kuat**, dan warna **pudar** menandakan korelasi **lemah**. Korelasi **terkuat** terlihat antara **Bilirubin-Copper (0.43)** dan **Prothrombin-Bilirubin (0.32)**, sementara mayoritas variabel menunjukkan korelasi yang lemah (ditandai warna biru muda atau putih) yang mengindikasikan bahwa kebanyakan fitur bersifat independen satu sama lain, dengan diagonal selalu bernilai 1.0 (merah) karena menunjukkan korelasi variabel dengan dirinya sendiri.

CORRELATION (CATEGORICAL)



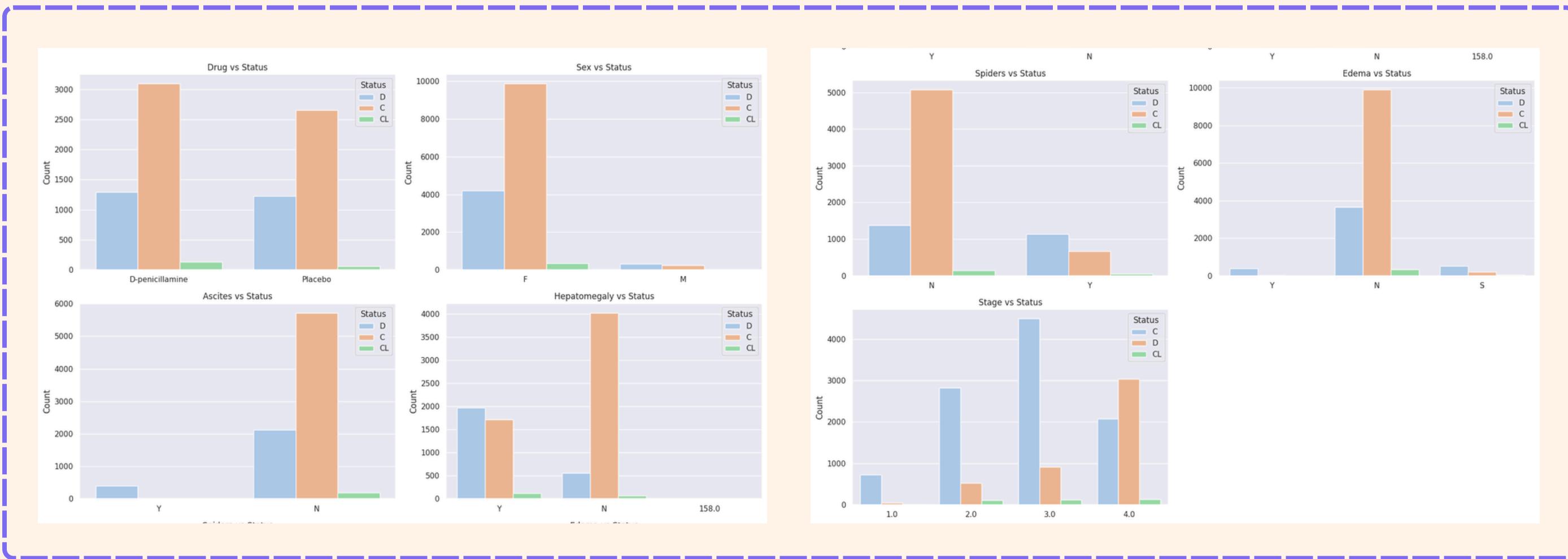
Correlation matrix ini menunjukkan hubungan antara fitur-fitur **categorical** yang tampaknya terkait dengan kondisi medis, di mana korelasi **terkuat** terlihat antara **Ascites-Edema (0.69)** dan **Hepatomegaly-Stage (0.45)**, sementara variabel **Drug** dan **Sex** menunjukkan korelasi yang **sangat lemah** dengan variabel lainnya (hampir semua **di bawah 0.1**), dengan pola warna yang sama seperti matriks sebelumnya dimana merah menunjukkan korelasi positif kuat, biru menunjukkan korelasi lemah, dan diagonal bernilai 1.0 karena merupakan korelasi dengan diri sendiri.

CORRELATION BETWEEN FEATURES (NUMERICAL) AND TARGET



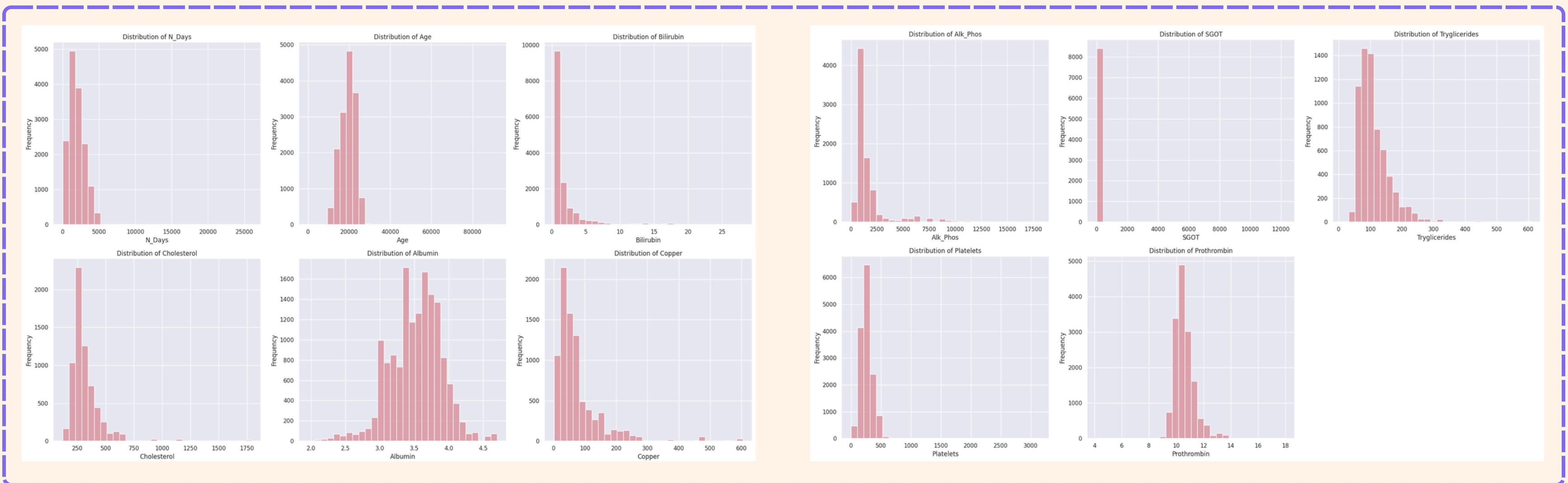
Serangkaian violin plot ini yang memperlihatkan korelasi berbagai fitur numerik (seperti N_Days, Age, Bilirubin, Cholesterol, dll.) dengan target Status pasien yang dibagi menjadi tiga kategori (D, C, dan CL). Bentuk "violin" menunjukkan kepadatan distribusi data, di mana bagian yang lebih lebar menandakan lebih banyak data pada nilai tersebut, dengan garis hitam di tengah menunjukkan median - beberapa pola menarik terlihat seperti Bilirubin yang memiliki distribusi lebih tinggi pada status D dan CL dibanding C, sementara Albumin relatif stabil di semua status, dan SGOT menunjukkan beberapa nilai ekstrem terutama pada status D.

CORRELATION BETWEEN FEATURES (CATEGORICAL) AND TARGET



Gambar ini menampilkan korelasi fitur-fitur kategorikal (Drug, Sex, Ascites, Hepatomegaly, Spiders, Edema, dan Stage) dengan status pasien (D, C, CL) dalam bentuk diagram batang, dimana setiap plot menunjukkan frekuensi atau jumlah kejadian untuk setiap kategori - misalnya, untuk Drug menunjukkan distribusi antara D-penicillamine dan Placebo, Sex menunjukkan mayoritas pasien adalah perempuan (F), Ascites dan Hepatomegaly lebih banyak yang tidak memiliki kondisi tersebut (N), dan Stage menunjukkan distribusi tingkat keparahan penyakit dari 1.0 hingga 4.0, dengan proporsi yang berbeda-beda untuk setiap Status.

VISUALISASI DATA (NUMERICAL)



Distribusi frekuensi dari 11 fitur **numerical** dalam dataset medis melalui **histogram**, dimana setiap plot menunjukkan sebaran data untuk parameter yang berbeda. Bentuk distribusi yang berbeda-beda ini memberikan gambaran tentang karakteristik setiap parameter dalam populasi pasien.

VISUALISASI DATA (CATEGORICAL)



Distribusi data **categorical** dalam bentuk **diagram batang**, dimana terlihat bahwa penggunaan D-penicillamine sedikit lebih tinggi dibanding Placebo, mayoritas pasien adalah perempuan (F), kebanyakan pasien tidak memiliki Ascites (N), distribusi Hepatomegaly lebih tinggi pada kondisi tidak ada (N), Spiders lebih banyak yang tidak memiliki (N), Edema dominan pada kondisi tidak ada (N), dan Stage penyakit menunjukkan peningkatan jumlah pasien dari stage 1 hingga mencapai puncak di stage 3-4. Semua visualisasi ini membantu memahami karakteristik populasi pasien dalam dataset.

INSIGHTS



- Terdapat 8 kolom yang termasuk data **categorical**. Terbagi menjadi 3 jenis yaitu **Ordinal** (Stage dan Status), **Binary** (Drug, Sex, Ascites, Hepatomegaly, dan Spiders), serta **Nominal** (Edema).
- Dataset memiliki masalah **missing values** yang signifikan terutama pada variabel Tryglicerides (56.17%), Cholesterol (55.83%), dan beberapa variabel lain yang memiliki sekitar 43% missing values, sementara beberapa variabel penting seperti Status, Stage, N_Days, Age, Sex, dan Bilirubin tidak memiliki missing values sama sekali.
- Variabel Bilirubin memiliki jumlah **outlier** terbanyak (1577 data), diikuti oleh Alk_Phosphatase (845) dan Copper (695), sementara N_Days (35) dan Age (16) memiliki outlier yang relatif sedikit.
- Dari kedua matriks korelasi terlihat bahwa baik fitur **numerical** maupun **categorical** umumnya memiliki **korelasi yang lemah** satu sama lain (majoritas nilai <0.4), kecuali hubungan antara Ascites-Edema (0.69) di fitur kategorikal dan Bilirubin-Copper (0.43) di fitur **numerical**, menunjukkan bahwa kebanyakan variabel cukup independen dalam menjelaskan kondisi pasien.

DATA SPLITTING, CLEANING, AND PREPROCESSING



DATA SPLIT -> TRAIN AND TEST

Splitting bertujuan untuk membagi dataset menjadi dua bagian, yaitu **training set** untuk melatih model dan **validation set** untuk mengevaluasi performa model pada data yang belum pernah dilihat.

Pembagian dilakukan dengan proporsi 80% data digunakan untuk training, dan 20% untuk validasi. Pembagian ini dilakukan untuk mengatasi sekiranya dataset tidak seimbang (imbalanced data) antara training dan test, sehingga model pun dilatih dan diuji pada representasi yang proporsional dari kelas-kelas target.

Splitting yang kami gunakan adalah **Hold-Out Validation**.

```
Training Set Size: (12000, 19), Test Set Size: (3000, 19)
```

```
Class distribution in original set:
```

Status	
C	0.674467
D	0.301667
CL	0.023867

```
Name: proportion, dtype: float64
```

```
Class distribution in training set:
```

Status	
C	0.674500
D	0.301667
CL	0.023833

```
Name: proportion, dtype: float64
```

```
Class distribution in test set:
```

Status	
C	0.674333
D	0.301667
CL	0.024000

```
Name: proportion, dtype: float64
```

DATA CLEANING

Dari data missing values yang telah diperoleh pada tahap EDA, baik itu missing values pada data numerik maupun kategorikal, kita akan menentukan sebaiknya data tersebut digantikan dengan apa. Dalam hal ini, kami memutuskan untuk mengganti **missing values pada data numerik dengan median** dan data **kategorikal dengan nilai konstan, yaitu string 'missing'** dengan memanfaatkan pipeline.

```
numerical_pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='median')),
    ('power_transform', PowerTransformer(standardize=True)),
    ('scaler', RobustScaler())
])

categorical_pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='constant', fill_value='missing')),
    ('onehot', OneHotEncoder(handle_unknown='ignore', sparse_output=False,
min_frequency=0.01))
])
```

PREPROCESSING

Selanjutnya, pipeline yang telah dibuat sebelumnya akan di **compile menjadi sebuah column transformer** yang akan digunakan untuk data **training** dan **test**.

```
preprocessor = ColumnTransformer([
    ('num', numerical_pipeline, num_columns),
    ('cat', categorical_pipeline, cat_columns)
])

X_train_processed =
preprocessor.fit_transform(x_train)
X_val_processed = preprocessor.transform(x_val)
```

MODELING AND VALIDATION

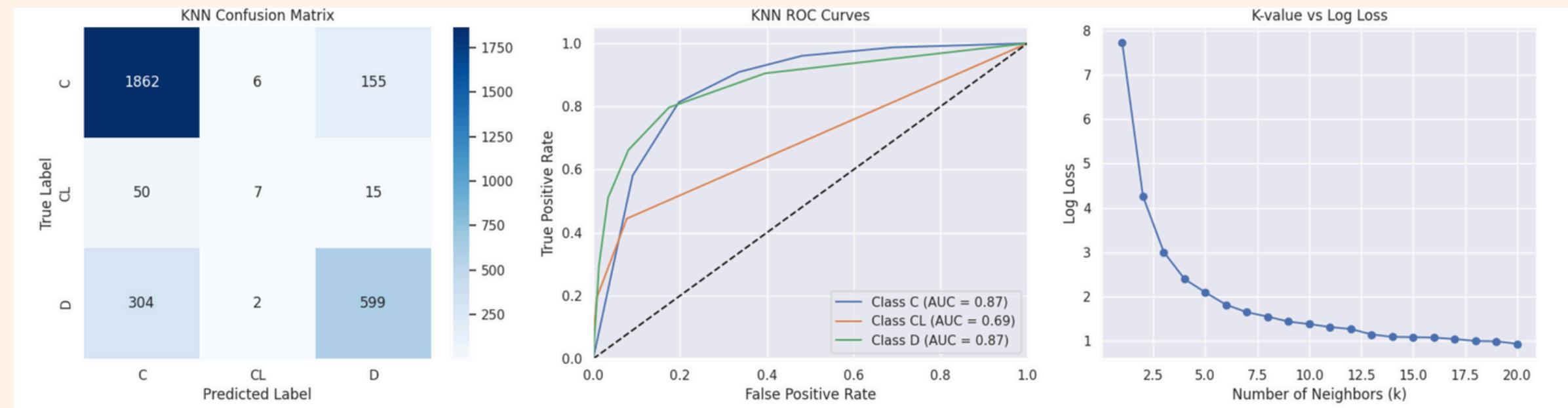


KNN

Berikut adalah kode untuk model kNN.

```
knn_model = KNeighborsClassifier()  
knn_model.fit(X_train_processed, y_train)  
y_val_pred_knn =  
knn_model.predict_proba(X_val_processed)  
knn_log_loss = log_loss(y_val,  
y_val_pred_knn)
```

Dari sana, didapatkan nilai log loss untuk kNN adalah sebesar **2.0836**.



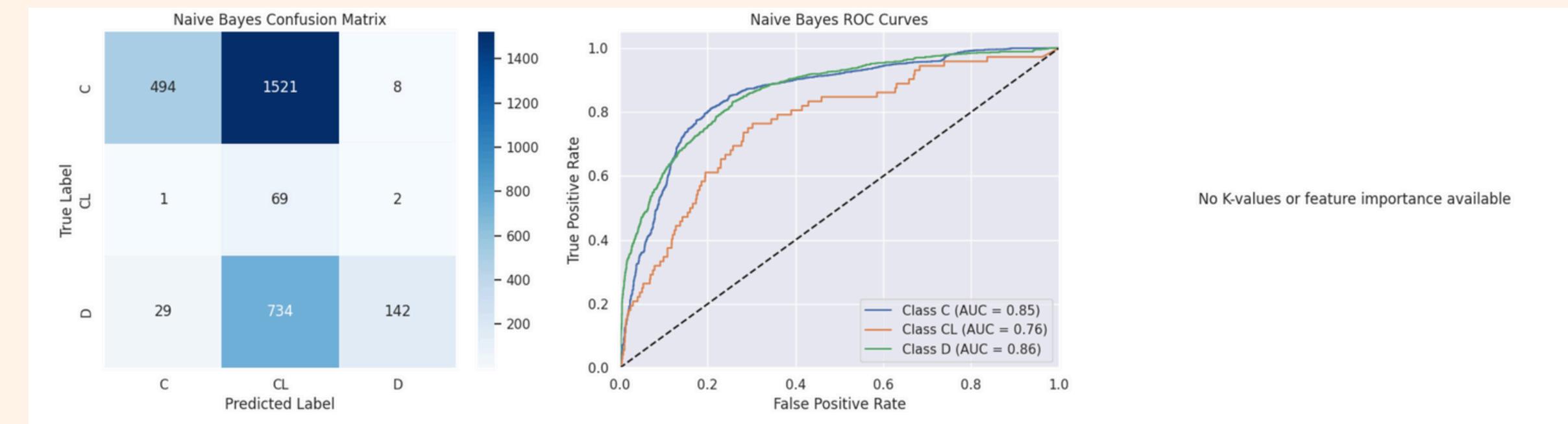
- › Grafik confusion matrix menunjukkan bahwa kelas 0 dan kelas 2 memiliki performa yang baik, namun kelas 1 memiliki performa yang buruk, terlihat dari jumlah prediksi yang salah lebih tinggi dibandingkan prediksi yang benar.
- › Dari grafik ROC, didapatkan bahwa kelas CL adalah yang paling sulit dipisahkan (AUC rendah), sementara kelas C dan D lebih mudah diprediksi dengan AUC tinggi.
- › Dari grafik log loss vs k value, didapatkan bahwa saat nilai k meningkat, nilai log loss cenderung menurun, menandakan prediksi model menjadi lebih stabil. K yang optimal mungkin berada di antara 10 sampai 15.

NAIVE BAYES

Berikut adalah kode untuk model Naive Bayes.

```
nb_model = GaussianNB()  
nb_model.fit(X_train_processed, y_train)  
y_val_pred_nb =  
nb_model.predict_proba(X_val_processed)  
nb_log_loss = log_loss(y_val, y_val_pred_nb)
```

Dari sana, didapatkan nilai log loss untuk Naive Bayes adalah sebesar **11.6159**.



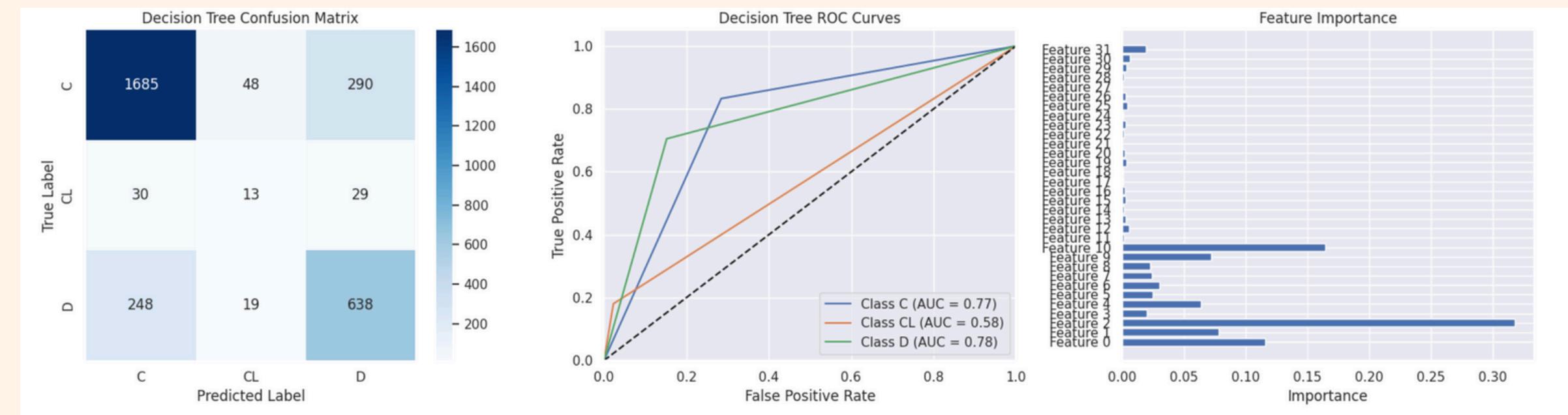
- > Grafik confusion matrix menunjukkan bahwa Kelas sebanyak 1521 sampel kelas C diklasifikasikan salah sebagai kelas CL, sebagian kecil salah diklasifikasikan sebagai kelas D (8 sampel). Sebagian besar kelas CL diklasifikasikan benar, hanya 1 salah sebagai kelas C dan 2 ke kelas D. Sementara itu, sebagian besar kelas D salah diprediksi ke kelas CL, yaitu 734.
- > Dari grafik ROC, didapatkan bahwa kelas CL adalah yang paling sulit dipisahkan (AUC rendah), sementara kelas C dan D lebih mudah diprediksi dengan AUC tinggi.

ID3

Berikut adalah kode untuk model ID3.

```
dt_model =  
DecisionTreeClassifier(random_state=42)  
dt_model.fit(X_train_processed, y_train)  
y_val_pred_dt =  
dt_model.predict_proba(X_val_processed)  
dt_log_loss = log_loss(y_val, y_val_pred_dt)
```

Dari sana, didapatkan nilai log loss untuk ID3 adalah sebesar **7.9296**.



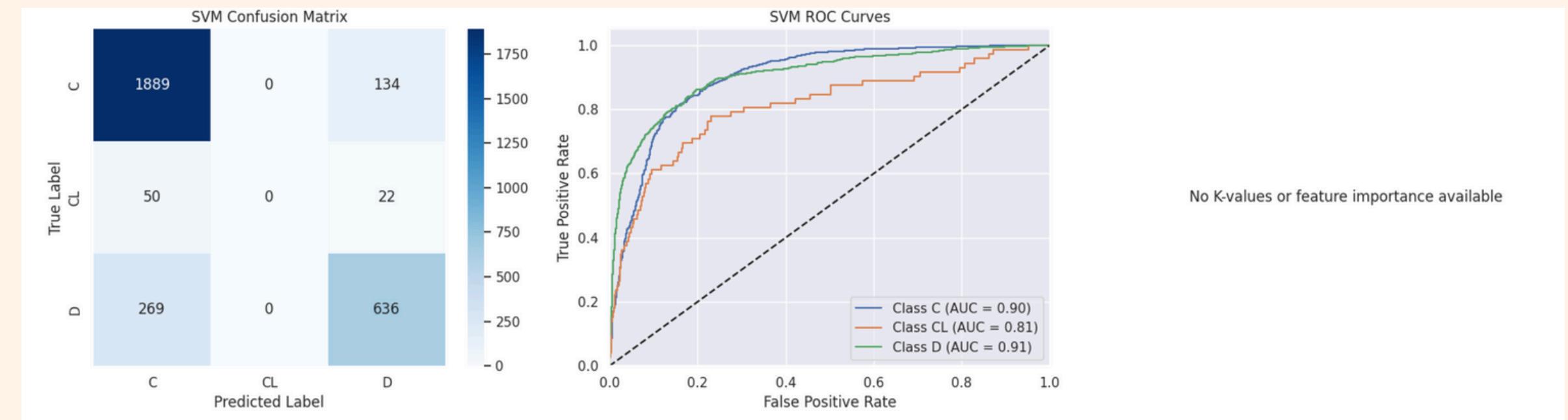
- > Grafik confusion matrix menunjukkan bahwa Kelas sebanyak 1521 sampel kelas C diklasifikasikan benar, hanya sebagian kecilnya yang salah diprediksi ke kelas CL dan D. Sebagian besar kelas CL diklasifikasikan salah, hanya 13 yang benar salah. Sementara itu, sebagian besar kelas D benar diprediksi, dan umum salah diklasifikasikan ke kelas C.
- > Dari grafik ROC, didapatkan bahwa kelas CL adalah yang paling sulit dipisahkan (AUC rendah), sementara kelas C dan D lebih mudah diprediksi dengan AUC tinggi.
- > Dari grafik feature importance, didapatkan informasi bahwa fitur yang paling penting dalam pengklasifikasian adalah fitur 2.

SVM

Berikut adalah kode untuk model SVM.

```
svm_model = SVC(probability=True,  
random_state=42)  
svm_model.fit(X_train_processed, y_train)  
y_val_pred_svm =  
svm_model.predict_proba(X_val_processed)  
svm_log_loss = log_loss(y_val,  
y_val_pred_svm)
```

Dari sana, didapatkan nilai log loss untuk SVM adalah sebesar **0.4263**.



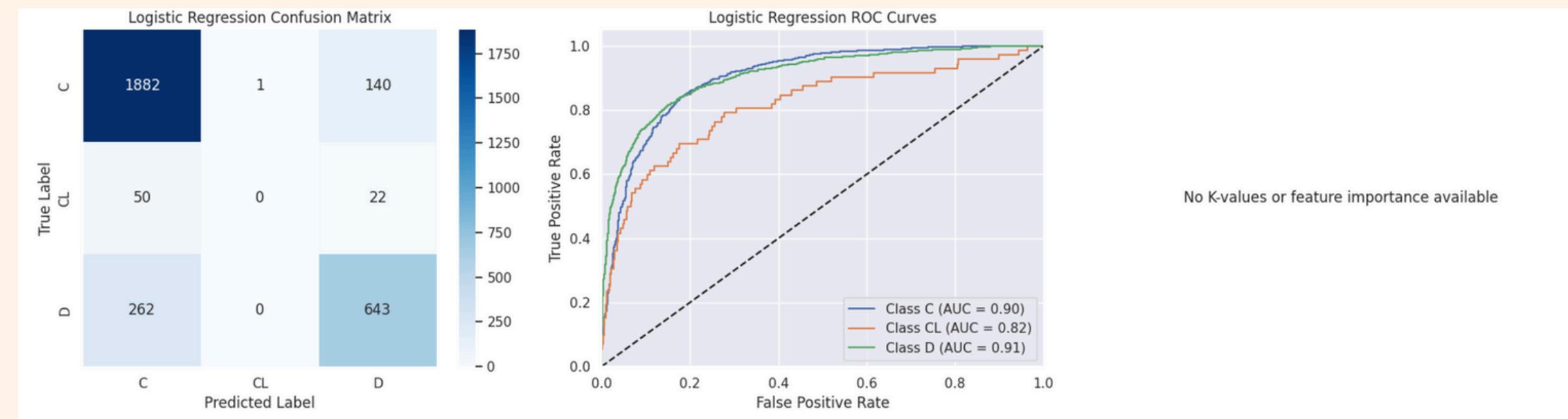
- › Grafik confusion matrix menunjukkan bahwa Kelas sebanyak 1889 sampel kelas C diklasifikasikan benar, hanya sebagian kecilnya yang salah diprediksi ke kelas D. Seluruh kelas CL diklasifikasikan salah. Sementara itu, sebagian besar kelas D benar diprediksi, dan apabila ada salah, diklasifikasikan ke kelas C.
- › Dari grafik ROC, didapatkan bahwa kelas CL adalah yang paling sulit dipisahkan (AUC rendah), sementara kelas C dan D lebih mudah diprediksi dengan AUC tinggi.

LOGISTIC REGRESSION

Berikut adalah kode untuk model Logistic Regression.

```
lr_model = LogisticRegression(max_iter=1000,  
random_state=42)  
lr_model.fit(X_train_processed, y_train)  
y_val_pred_lr =  
lr_model.predict_proba(X_val_processed)  
lr_log_loss = log_loss(y_val, y_val_pred_lr)
```

Dari sana, didapatkan nilai log loss untuk Logistic Regression adalah sebesar **0.4206**.



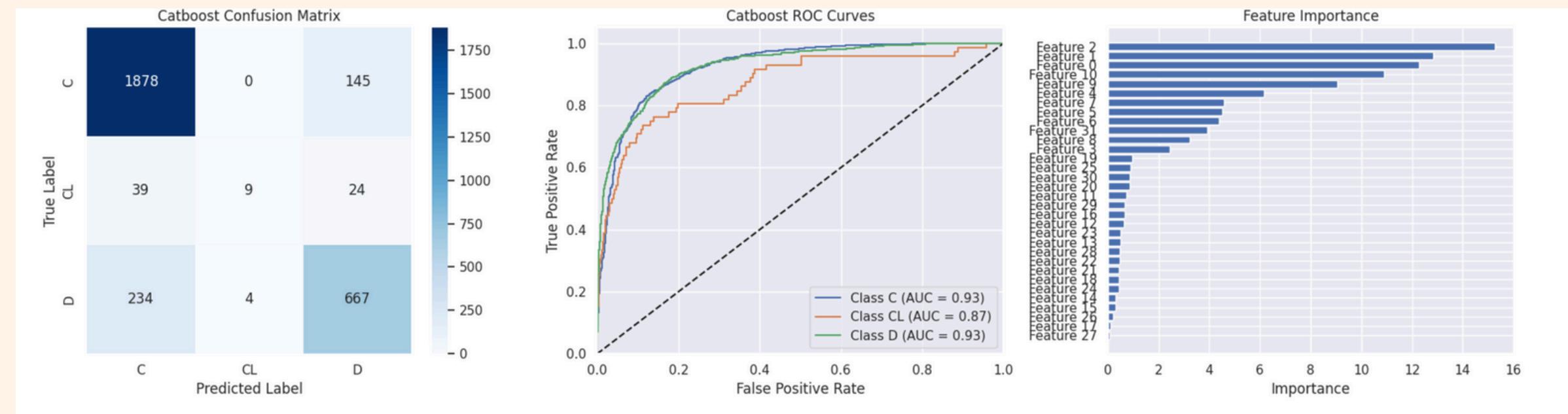
- > Grafik confusion matrix menunjukkan bahwa Kelas sebanyak 1882 sampel kelas C diklasifikasikan benar, hanya sebagian kecilnya yang salah diprediksi ke kelas CL dan D. Seluruh kelas CL diklasifikasikan salah. Sementara itu, sebagian besar kelas D benar diprediksi, dan apabila ada salah, diklasifikasikan ke kelas C.
- > Dari grafik ROC, didapatkan bahwa kelas CL adalah yang paling sulit dipisahkan (AUC rendah), sementara kelas C dan D lebih mudah diprediksi dengan AUC tinggi.

CATBOOST

Berikut adalah kode untuk model Catboost.

```
if len(np.unique(y_train)) == 2:  
    catboost_model = CatBoostClassifier(  
        iterations=1000,  
        learning_rate=0.1,  
        random_state=42,  
        loss_function='Logloss',  
        eval_metric='Logloss',  
        verbose=100  
    )  
else:  
    catboost_model = CatBoostClassifier(  
        iterations=1000,  
        learning_rate=0.1,  
        random_state=42,  
        loss_function='MultiClass', # for multiclass  
        classification  
        eval_metric='MultiClass',  
        verbose=100  
    )  
catboost_model.fit(  
    X_train_processed,  
    y_train,  
    eval_set=[(X_val_processed, y_val)],  
    early_stopping_rounds=50,  
    verbose=100  
)  
y_val_pred_catboost =  
catboost_model.predict_proba(X_val_processed)  
catboost_log_loss = log_loss(y_val,  
y_val_pred_catboost)
```

Darisana, didapatkan nilai log loss untuk Catboost adalah sebesar **0.3716**.



- > Grafik confusion matrix menunjukkan bahwa Kelas sebanyak 1878 sampel kelas C diklasifikasikan benar, hanya sebagian kecilnya yang salah diprediksi ke kelas D. Hampir seluruh kelas CL diklasifikasikan salah. Sementara itu, sebagian besar kelas D benar diprediksi.
- > Dari grafik ROC, didapatkan bahwa kelas CL adalah yang paling sulit dipisahkan (AUC rendah), sementara kelas C dan D lebih mudah diprediksi dengan AUC tinggi.
- > Dari grafik feature importance, didapatkan pula bahwa fitur yang paling penting dalam menentukan klasifikasi adalah fitur kedua.

ERROR ANALYSIS



DATA IMBALANCE

Data **imbalance** merupakan **ketidakseimbangan proporsi data**, seperti misalnya dari 3 status yang ada (C, CL, dan D), salah satu atau beberapaanya memiliki jumlah yang jauh lebih banyak dibandingkan kelas yang lain (dalam hal ini, kelas C jauh lebih banyak dibandingkan CL dan D), sehingga mempengaruhi performa model secara signifikan, khususnya dalam hal kemampuan model untuk mengenali kelas dengan jumlah data yang lebih sedikit. Model menjadi "**terbiasa**" dengan **kelas mayoritas**, sehingga menjadi **bias**.

```
----- VALUE COUNTS (STATUS) -----
Status
C      10117
D      4525
CL     358
Name: count, dtype: int64
```

MISSING VALUES

----- Missing Values -----:		
	Missing Values	Percentage
id	0	0.000000
N_Days	0	0.000000
Drug	6550	43.666667
Age	0	0.000000
Sex	0	0.000000
Ascites	6547	43.646667
Hepatomegaly	6552	43.680000
Spiders	6559	43.726667
Edema	0	0.000000
Bilirubin	0	0.000000
Cholesterol	8374	55.826667
Albumin	0	0.000000
Copper	6660	44.400000
Alk_Phosphatase	6556	43.706667
SGOT	6559	43.726667
Tryglicerides	8425	56.166667
Platelets	584	3.893333
Prothrombin	16	0.106667
Stage	0	0.000000
Status	0	0.000000

Dataset memiliki banyak **missing values**, yang jika tidak ditangani dengan benar, model dapat **kehilangan informasi penting** yang dibutuhkan untuk belajar dari data. Model mungkin mempelajari **pola yang tidak lengkap** atau bias jika nilai yang hilang tidak digantikan atau diproses dengan benar.

Hal yang kami lakukan pertama adalah mencoba untuk menghapus data-data *missing values* ini, namun ternyata hasilnya malah lebih buruk. Terdapat beribu-ribu data yang hilang dan membuat data **tidak akurat**.

Akhirnya kami memutuskan untuk **mengisi nilai yang hilang** tersebut dengan **median** (untuk *numerical*) dan mengisi dengan **nilai konstan 'missing'** (untuk *categorical*) pada pipeline.

OUTLIERS

----- NUMERICAL DATA OUTLIERS BY IQR (True indicates an outlier) -----	
	0
N_Days	True
Age	True
Bilirubin	True
Cholesterol	True
Albumin	True
Copper	True
Alk_Phosphat	True
SGOT	True
Tryglicerides	True
Platelets	True
Prothrombin	True

----- JUMLAH OUTLIER PER KOLOM NUMERICAL -----	
	0
N_Days	35
Age	16
Bilirubin	1577
Cholesterol	384
Albumin	353
Copper	695
Alk_Phosphat	845
SGOT	208
Tryglicerides	291
Platelets	136
Prothrombin	392

RARE CATEGORIES IN NON-NUMERICAL DATA:

No rare categories detected in Drug that fall below the 1.0% threshold.

No rare categories detected in Sex that fall below the 1.0% threshold.

No rare categories detected in Ascites that fall below the 1.0% threshold.

Rare categories in HEPATOMEGALY (less than 1.0%):

proportion	
Hepatomegaly	158.0 0.000118

dtype: float64

No rare categories detected in Spiders that fall below the 1.0% threshold.

No rare categories detected in Edema that fall below the 1.0% threshold.

No rare categories detected in Status that fall below the 1.0% threshold.

No rare categories detected in Stage that fall below the 1.0% threshold.

Penanganan outliers sebenarnya ada bermacam-macam. Untuk *numerical data* dapat diganti dengan **mean**, **median**, ataupun **modus**. Selain itu juga dapat menggunakan **metode clipping** dan metode untuk **mengganti value outliers ke value yang paling dekat** dengan outliers tersebut. Namun setelah kami mencoba semua hal tersebut, hasil model lebih baik apabila outliers tidak diganti. Maka dari itu kami memutuskan untuk membiarkan nilai-nilai outliers tersebut.



**THANK
YOU!**