Data collection and preparation

1.Data Collection

Dataset to Use:

Indian Liver Patient Dataset (ILPD)

- **Source:** UCI ML Repository
- **Alternate:** Search "Liver Patient Dataset" on Kaggle for updated versions or enhanced features.

Dataset Summary:

Feature	Description
Age	Age of the patient
Gender	Male/Female
Total_Bilirubin	Liver function indicator
Direct_Bilirubin	More specific liver marker
Alkaline_Phosphotase	Enzyme related to bile ducts
Alamine_Aminotransferase	Enzyme level in liver
Aspartate_Aminotransferase	Enzyme involved in metabolism
Total_Proteins	Overall protein level
Albumin	Protein produced by the liver
Albumin_and_Globulin_Ratio	Protein ratio affected by liver health
Dataset (Target)	1 = Liver patient, 2 = Not a liver patient

2. Data Preparation Steps

Step 1: Load Dataset

```
import pandas as pd
df = pd.read_csv('indian_liver_patient.csv')
df.head()
```

Step 2: Data Cleaning

a)Check for Null Values

```
df.isnull().sum()
df['Albumin_and_Globulin_Ratio'].fillna(df['Albumin_and_Globulin_Ratio'].median(),
inplace=True)
```

b) Fix Column Names

```
df.columns = df.columns.str.replace(' ', '_')
```

Step 3: Target Column Mapping

Convert Dataset values to binary (1 = Liver Disease, 0 = No Disease):

```
df['Target'] = df['Dataset'].map({1: 1, 2: 0})
df.drop('Dataset', axis=1, inplace=True)
```

Step 4: Exploratory Data Analysis (Optional)

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(x='Target', data=df)
plt.title('Class Distribution')
plt.show()
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
```

Step 5: Feature Engineering

a) Convert Categorical to Numeric

```
df['Gender'] = df['Gender'].map({'Male': 1, 'Female': 0})
```

b) Feature Scaling (optional for some models)

```
from sklearn.preprocessing import StandardScaler
features = df.drop('Target', axis=1)
target = df['Target']
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)
```

Step 6: Train-Test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(scaled_features, target, test_size=0.2,
random_s
```