

Model Building

1. Train-Test Split

Split the dataset to evaluate model performance.

```
from sklearn.model_selection import train_test_split
X = df.drop('Cirrhosis', axis=1)
y = df['Cirrhosis']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

2. Model Selection

Try multiple algorithms and compare:

- ☒ Logistic Regression
- ☒ Random Forest
- ☒ XGBoost
- ☒ Support Vector Machine (SVM)
- ☒ K-Nearest Neighbors (KNN)

3. Training Models

Example: Random Forest

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
```

Example: XGBoost

```
import xgboost as xgb
xgb_model = xgb.XGBClassifier(use_label_encoder=False, eval_metric='logloss')
xgb_model.fit(X_train, y_train)
```

4. Model Evaluation

Use metrics like:

- **Accuracy**
- **Precision, Recall, F1-Score**
- **Confusion Matrix**
- **ROC-AUC Score**

```
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score
y_pred = rf.predict(X_test)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
print("ROC AUC Score:", roc_auc_score(y_test, rf.predict_proba(X_test)[:,:1]))
```

5. Hyperparameter Tuning (Optional but Recommended)

```
from sklearn.model_selection import GridSearchCV
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [4, 6, 8]
}
grid = GridSearchCV(RandomForestClassifier(), param_grid, cv=5)
grid.fit(X_train, y_train)
```

6. Model Comparison Table (Example Output)

| Model | Accuracy | ROC AUC | F1-Score |
|--------------------|----------|---------|----------|
| LogisticRegression | 82% | 0.85 | 0.79 |
| RandomForest | 89% | 0.91 | 0.86 |
| XGBoost | 90% | 0.93 | 0.88 |
| SVM | 85% | 0.87 | 0.82 |

Final Step: Model Selection & Save Best Model

```
import joblib
joblib.dump(xgb_model, 'best_model_liver.pkl')
```