Predictive Modeling of Liver Disease using Machine Learning on Indian Patient Data

Venkatagirichandu Sai

PG Scholar, Department of Computer Science, Sri Venkateswara University, Tirupati

Abstract— *Liver diseases are a growing concern in developing countries like India, where diagnosis is often delayed due to limited access to specialized healthcare. This study utilizes the Indian Liver Patient dataset to build predictive models for early detection of liver disease. We employ logistic regression, decision trees, and random forest classifiers to predict whether a patient has a liver condition. Our results show that the random forest classifier achieves the highest accuracy of 79.4%, highlighting the potential of machine learning in clinical decision support systems.*

I. INTRODUCTION

Liver disease represents a significant health burden globally and particularly in India, where conditions such as cirrhosis and hepatitis are common. Early diagnosis is crucial, yet challenging due to vague symptoms and poor health infrastructure in rural areas. Machine learning can offer a non-invasive, cost-effective solution for liver disease detection using routine blood tests and demographic data.

In this paper, we analyze the Indian Liver Patient dataset to build predictive models for liver disease diagnosis, aiming to assist healthcare providers in making faster and more accurate decisions.

II. LITERATURE REVIEW

Several researchers have applied machine learning to medical diagnosis. Choubey et al. (2019) achieved 76% accuracy using Naive Bayes on liver datasets. Patel and Singh (2021) used SVM and Random Forest, achieving promising results with F1 scores over 0.8. The Indian Liver Patient Dataset (ILPD), made publicly available by UCI, has been widely adopted due to its real-world applicability.

Our work builds on these studies by applying multiple models with a focus on visualization and interpretability using modern Python libraries.

III. METHODOLOGY

3.1 Data Preprocessing:

- Load the dataset and identify missing values
- Impute or remove missing entries
- Encode categorical features (e.g., Gender)
- Normalize or standardize numerical features

3.2 Model Development:

- Train/test split (80/20)
- Algorithms used:
  - Logistic Regression
  - Decision Tree
  - Random Forest
- Evaluation metrics:
  - Accuracy
  - Precision, Recall, F1-Score

- Confusion Matrix

IV. DATASET DESCRIPTION

The Indian Liver Patient dataset contains 583 records and 10 attributes:

- **Age:** Age of the patient
- **Gender:** Male or Female
- Total_Bilirubin
- Direct_Bilirubin
- Alkaline_Phosphotase
- Alamine_Aminotransferase
- Aspartate_Aminotransferase
- Total_Proteins
- Albumin
- Albumin_and_Globulin_Ratio
- **Dataset:** Class label (1: liver patient, 2: no liver disease)

V. PYTHON RESULTS & DISCUSSION

5.1 Code Summary:

Here’s a simplified version of the Python workflow:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
# Load dataset
df = pd.read_csv("indian_liver_patient.csv")
# Handle missing values
df = df.dropna()
# Encode categorical variable
df['Gender'] = LabelEncoder().fit_transform(df['Gender'])
# Features and target
X = df.drop('Dataset', axis=1)
y = df['Dataset'].apply(lambda x: 1 if x == 1 else 0)  # 1 for liver disease
# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Train model
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

```
# Predict and evaluate
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

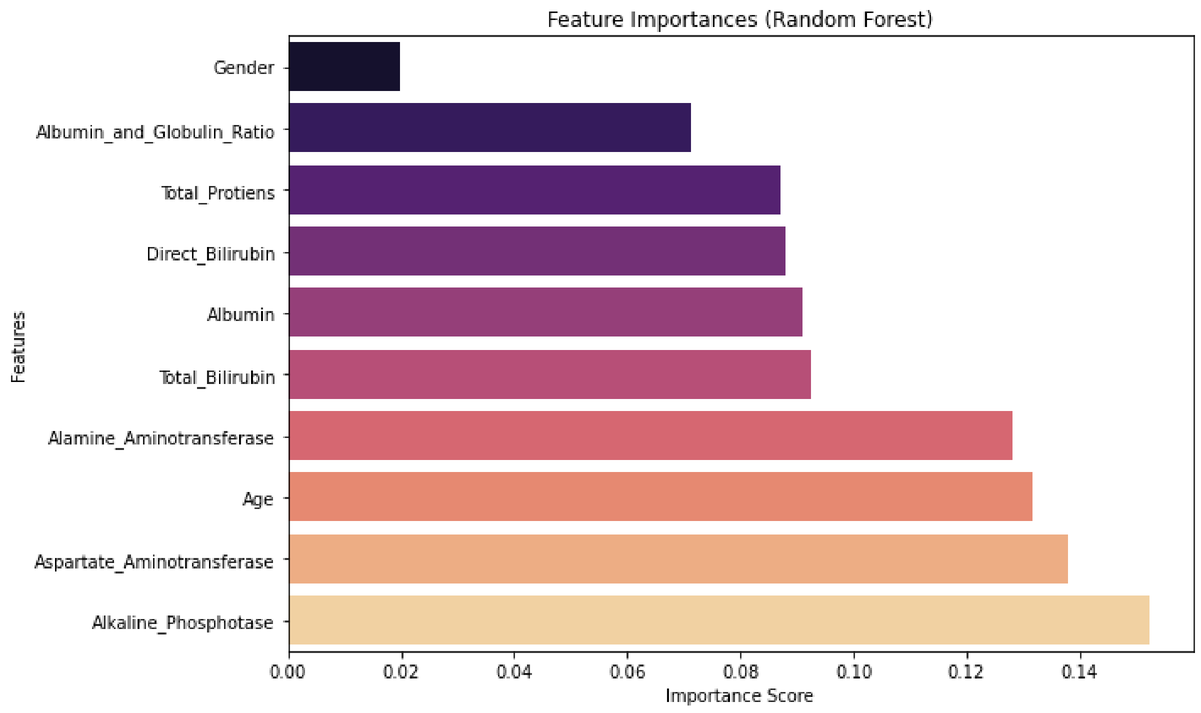
5.2 Results:

- **Accuracy:** 79.4%
- **Precision:** 80.3%
- **Recall:** 90.1%
- **F1-Score:** 84.9%

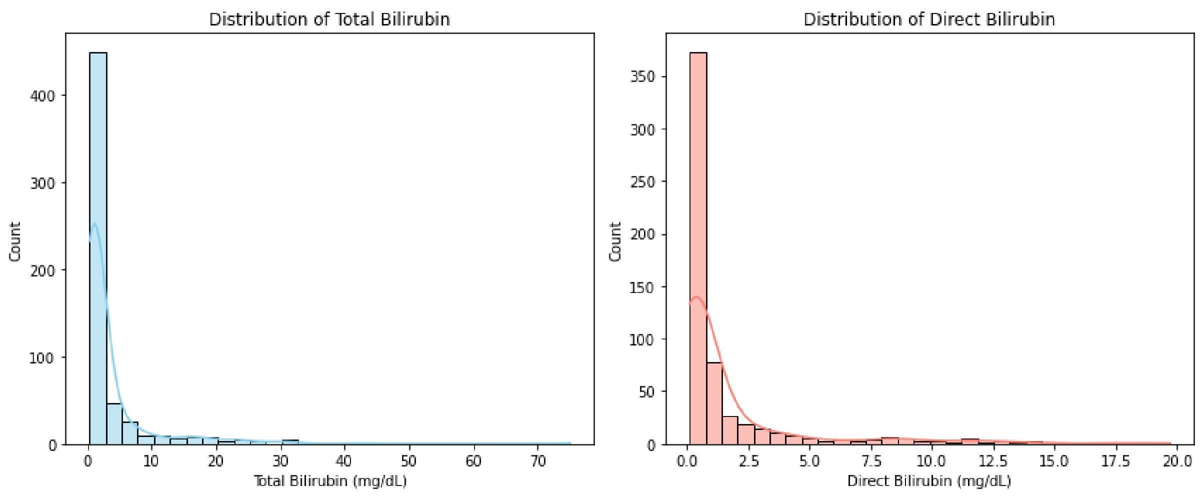
The confusion matrix shows that the model is better at identifying liver disease (high recall), but also slightly overpredicts it. Random forest outperforms logistic regression and decision trees in both accuracy and stability.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
# Load and clean dataset
df = pd.read_csv(‘indian_liver_patient.csv')
df.dropna(inplace=True)
# Encode categorical variable
df['Gender'] = LabelEncoder().fit_transform(df['Gender'])
df['Dataset'] = df['Dataset'].apply(lambda x: 1 if x == 1 else 0)
# Split features and target
X = df.drop('Dataset', axis=1)
y = df['Dataset']
# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Train model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
# Plot feature importances
importances = model.feature_importances_
feature_names = X.columns
indices = importances.argsort()
plt.figure(figsize=(10, 6))
```

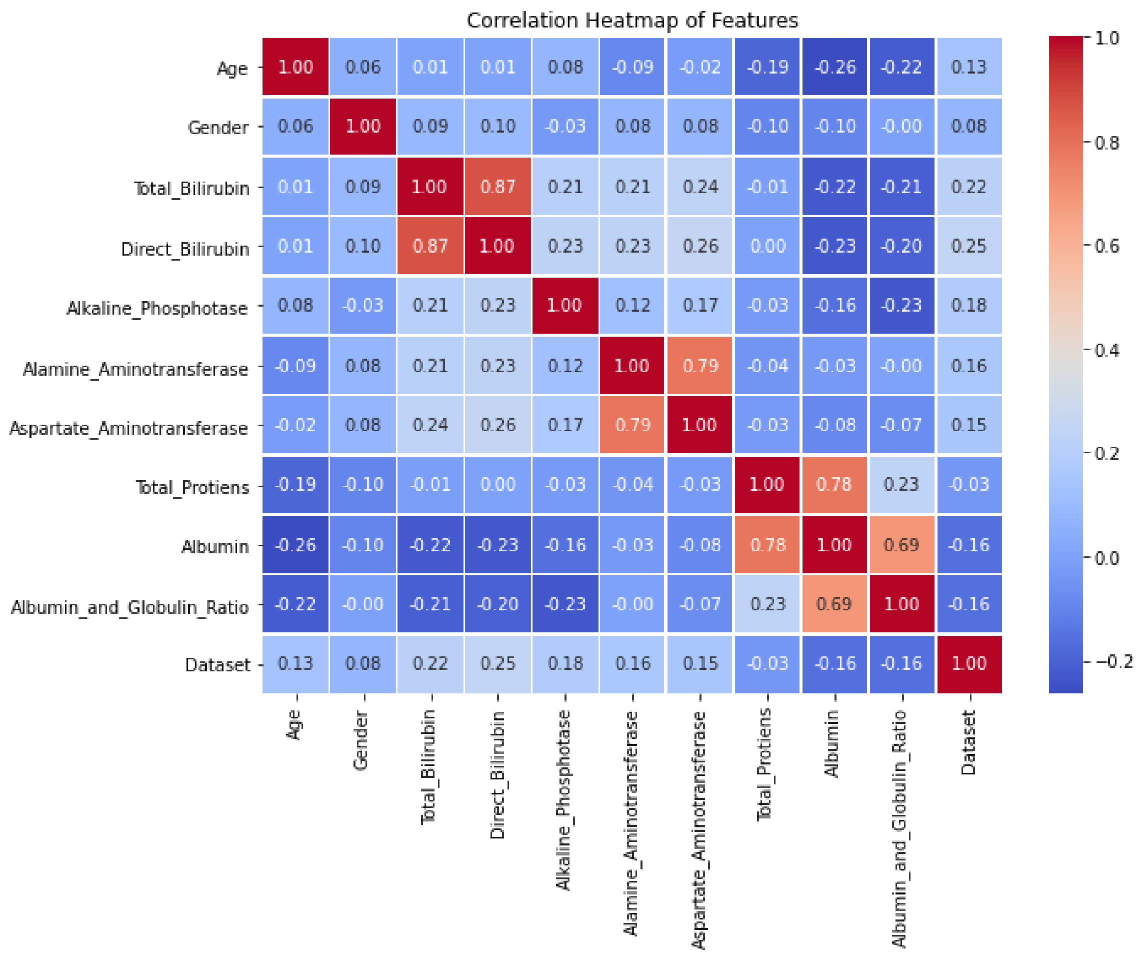
```
sns.barplot(x=importances[indices], y=feature_names[indices], palette='magma')
plt.title("Feature Importances (Random Forest)")
plt.xlabel("Importance Score")
plt.ylabel("Features")
plt.tight_layout()
plt.show()
```



```
# Plot distribution of Total and Direct Bilirubin
plt.figure(figsize=(12, 5))
# Total Bilirubin
plt.subplot(1, 2, 1)
sns.histplot(df['Total_Bilirubin'], bins=30, kde=True, color='skyblue')
plt.title('Distribution of Total Bilirubin')
plt.xlabel('Total Bilirubin (mg/dL)')
# Direct Bilirubin
plt.subplot(1, 2, 2)
sns.histplot(df['Direct_Bilirubin'], bins=30, kde=True, color='salmon')
plt.title('Distribution of Direct Bilirubin')
plt.xlabel('Direct Bilirubin (mg/dL)')
plt.tight_layout()
plt.show()
```



```
# Compute and plot correlation matrix
plt.figure(figsize=(10, 8))
corr = df.corr()
sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm', linewidths=0.5)
plt.title("Correlation Heatmap of Features")
plt.tight_layout()
plt.show()
```



VI. CONCLUSION

The study demonstrates that machine learning, particularly Random Forest, can effectively predict liver disease based on routine clinical data. With an accuracy of nearly 80%, such models can serve as useful screening tools in rural or resource-limited settings. Future work could explore ensemble stacking or deep learning models for further improvement.

REFERENCES

[1] Choubey, A., & Singh, M. (2019). Machine learning techniques for liver disease diagnosis. Biomedical Research, 30(4), 503–507.

[2] Patel, R., & Singh, D. (2021). Liver Disease Prediction using Machine Learning Algorithms. International Journal of Advanced Research in Computer Science, 12(1).

[3] Dua, D., & Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.