Project Document

Project Title:

Revolutionizing Liver Care: Predicting Liver Cirrhosis using Advanced Machine Learning Techniques

---

1. Introduction

This project aims to empower early detection of liver cirrhosis by utilizing machine learning. By analyzing patient data, it predicts cirrhosis risk—enabling timely intervention and improved patient care.

---

2. Data Pipeline & Preparation

- **Data collection & cleaning**: Collected dataset, handled missing values, and performed exploratory data analysis (EDA).
- **Encoding & splitting**: Categorical data was encoded; dataset divided into training and test sets for model evaluation.

---

3. Machine Learning Models

Tested multiple algorithms:

- Random Forest
- Support Vector Classifier (SVC)
- Logistic Regression
- XGBoost
- K-Nearest Neighbors (KNN)

After assessment, **KNN emerged as the best-performing model**.

---

4. Evaluation Metrics

Models were evaluated using:

- Accuracy
- Precision
- Recall
- F1-score

- Confusion matrices

These helped identify the most effective algorithm for prediction.

---

5. Deployment

- Built a **Flask web app** so users can input data and receive predictions easily.
- Includes interactive confusion matrix visualizations and performance metrics for end-users.

---

6. Summary of Key Outcomes

- **Best model**: KNN, based on comprehensive comparison.
- **Interactive tool**: A deployable application that supports prediction and performance analysis.
- Supports early detection efforts, scalable for future improvements.

---

7. Technologies Used

- **Backend**: Python, Flask
- **ML & Analysis**: Scikit-learn (for KNN, Random Forest, SVC, Logistic Regression), XGBoost
- **Visualization**: Libraries for plotting confusion matrices and performance graphs

---

8. Usage Instructions

1. Clone the repo.
2. Install requirements (e.g., pip install -r requirements.txt).
3. Run the Flask app: python app.py.
4. Use the web interface to input patient data and receive **cirrhosis** predictions.

---

9. Future Improvements

- Increase dataset size and diversity.
- Hyperparameter tuning for further accuracy.
- Add advanced algorithms or ensemble models.
- Implement user authentication, logging, and analytics in the web app.
- Consider packaging as a container (e.g., Docker) for easy deployment.

---

10. Contributors

- **K Mayuri**