MINOR PROJECT

SMART BLOOD TEST INTERPRETER

Project Guide - Mr. Himanshu Ranjan

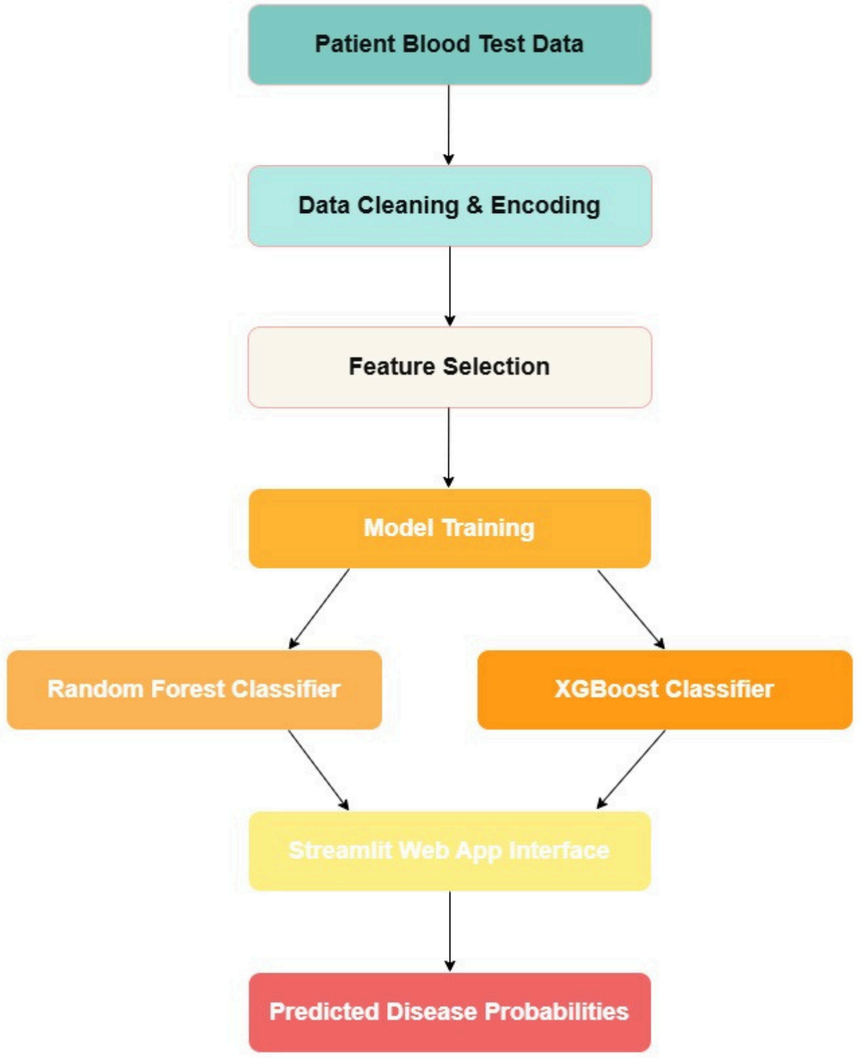
Our Team Members:

Prakhar Choyal	22051867
Pranjal Agarwal	22051868
Sayan Das	22051885
Tushar Agarwal	22051905
Suyash Pandey	22052075
Vinayak Puranik	22052083

Problem Statement

- Manual interpretation of blood test results is time-consuming and subject to human error.
- With increasing diagnostic demand, automation can enhance decision-making in clinical settings.
- Objective: Build a machine learning model to predict multiple diseases from routine blood test data.
- Also, provide a user-friendly web interface for healthcare use.

Smart Blood Test Interpreter – Workflow



- Patient blood test data is collected as input with multiple health parameters.
- Data cleaning and encoding steps are applied to handle missing values and format categorical features.
- Feature selection identifies the most relevant blood indicators for disease prediction.
- Two machine learning models – Random Forest and XGBoost – are trained on the selected features.
- The trained models are connected to a user-friendly Streamlit web application interface.
- The app takes user input and returns predicted probabilities for multiple diseases in real time.

Dataset Overview

- Dataset Name : Final_Dataset.csv (Post-cleaning and preprocessing)
- Rows : 15,175 patient entries
- Columns : 40 total (22 features + 18 disease labels)
- Input Features Include : Age, Gender, Hemoglobin, WBC Count, Platelet Count, Neutrophils, Monocytes, RDW- CV, MCHC, etc.
- Target Labels (18 diseases) : Include Dengue, Malaria, Leukemia, Anemia, Hypothyroidism, General Infection, Multiple Myeloma, and others.
- Multi-label format : A patient may have 0, 1, or more diseases.

Data Cleaning Process

1. Implemented in Data_Cleaning.ipynb
2. Steps Taken:

- Removed duplicate records and handled missing/null values.
- Standardized column names and corrected inconsistent entries.
- Dropped irrelevant columns and verified feature consistency.
- Ensured clean, structured data for further preprocessing steps.

3. Final clean dataset: Shape (15,175 × 40) saved to CSV for future use.

Data Preprocessing

1. Performed in Data_Preprocessing.ipynb
2. Feature Engineering:

- One-Hot Encoding on nominal categorical features.
- Top 20 features selected using XGBoost's feature importance ranking.

3. Data Augmentation:

- Added Gaussian noise to numeric columns to enhance model robustness and reduce overfitting.

4. Normalization:

- StandardScaler applied to all numeric features for consistent model learning.

5. Splitting:

- Train-Test split = 80:20 (X_train: 12,140 rows; X_test: 3,035 rows)

Machine Learning Models Used

1. Designed for Multi-Label Classification where multiple diseases may be predicted simultaneously.
2. Algorithms Used:

- Random Forest Classifier - Ensemble of decision trees, bagging-based, interpretable.
- XGBoost Classifier - Gradient Boosting Trees, known for high performance.

3. Wrapper: MultiOutputClassifier used to wrap both models to handle multilabel outputs.
4. Each model predicts presence/absence of each of the 18 diseases for a given blood profile.

VISUAL REPRESENTATION

	precision	recall	f1-score	support
No Major Condition Detected	0.96	0.97	0.96	1097
Iron Deficiency Anemia	0.94	0.94	0.94	205
Hemolytic Anemia	0.97	0.96	0.97	951
Vitamin B12 & Folate Deficiency	0.96	0.96	0.96	27
Chronic Kidney Disease	0.99	0.99	0.99	277
Thalassemia	0.91	0.95	0.93	42
Sepsis	1.00	0.99	0.99	83
Liver Disease	1.00	0.99	0.99	287
Dengue	1.00	0.98	0.99	45
Malaria	1.00	0.99	0.99	287
Aplastic Anemia	0.97	0.97	0.97	29
Leukemia	1.00	0.94	0.97	69
Multiple Myeloma	1.00	0.97	0.98	60
Myelodysplastic Syndrome	0.98	0.96	0.97	50
Pernicious Anemia	0.97	0.99	0.98	69
General Infection	1.00	1.00	1.00	1092
Hypothyroidism	0.98	0.97	0.98	546
Possible Autoimmune Disease	1.00	0.97	0.98	64
micro avg	0.98	0.98	0.98	5280
macro avg	0.98	0.97	0.97	5280
weighted avg	0.98	0.98	0.98	5280
samples avg	0.97	0.98	0.97	5280

Random Forest – Evaluation Metrics

1. Model trained on 12,140 entries; tested on 3,035 entries.
2. Performance Metrics:

- Training Accuracy: 98.45%
- Testing Accuracy: 93.12%
- Hamming Loss: 0.0213 (lower is better)
- F1 Score (Micro Avg): 0.9417
- F1 Score (Macro Avg): 0.9032

3. Observations:

- Performs well on most frequent disease classes.
- May misclassify rare conditions due to class imbalance.

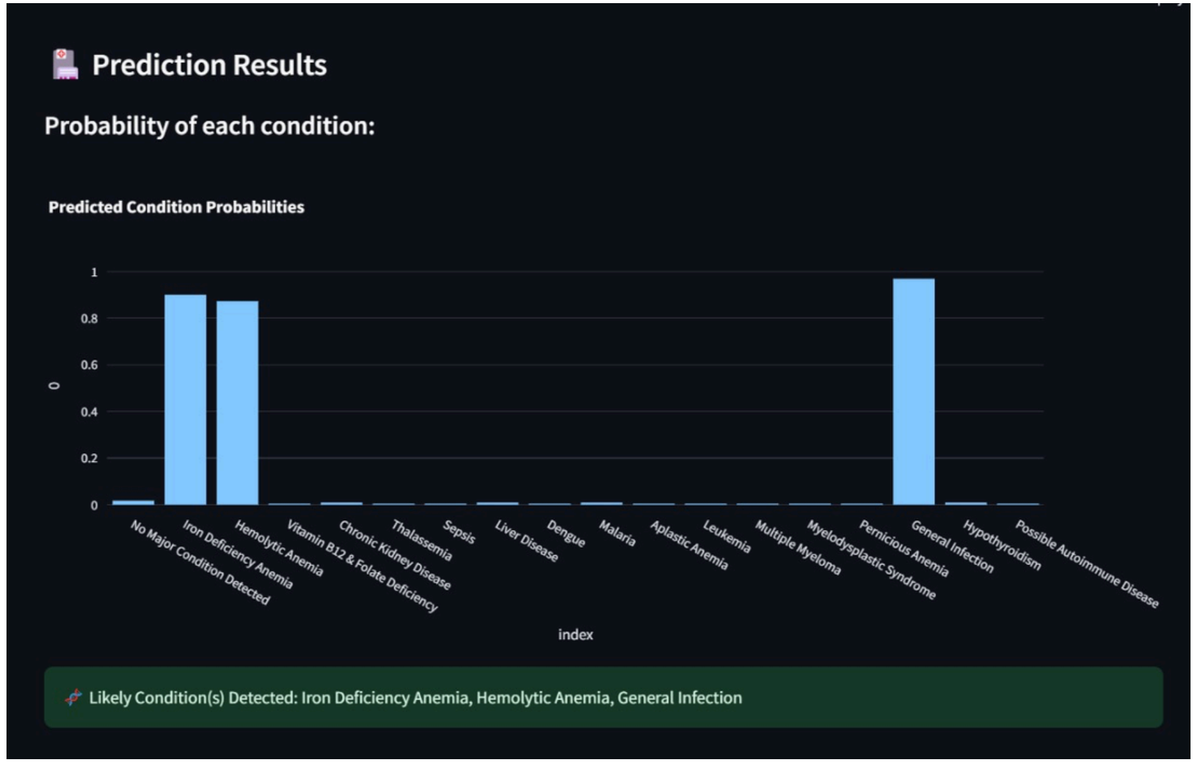
XGBoost – Evaluation Metrics

1. Advanced boosting technique, better at generalizing on unseen data.
2. Performance Metrics:

- Training Accuracy: 97.37%
- Testing Accuracy: 94.99%
- Hamming Loss: 0.0043
- F1 Score (Micro Avg): 0.9779
- F1 Score (Macro Avg): 0.9749

3. Observations:

- Outperformed Random Forest in every metric.
- Better precision and recall even on less frequent diseases.
- Ideal candidate for deployment due to superior performance.



Streamlit Web Application

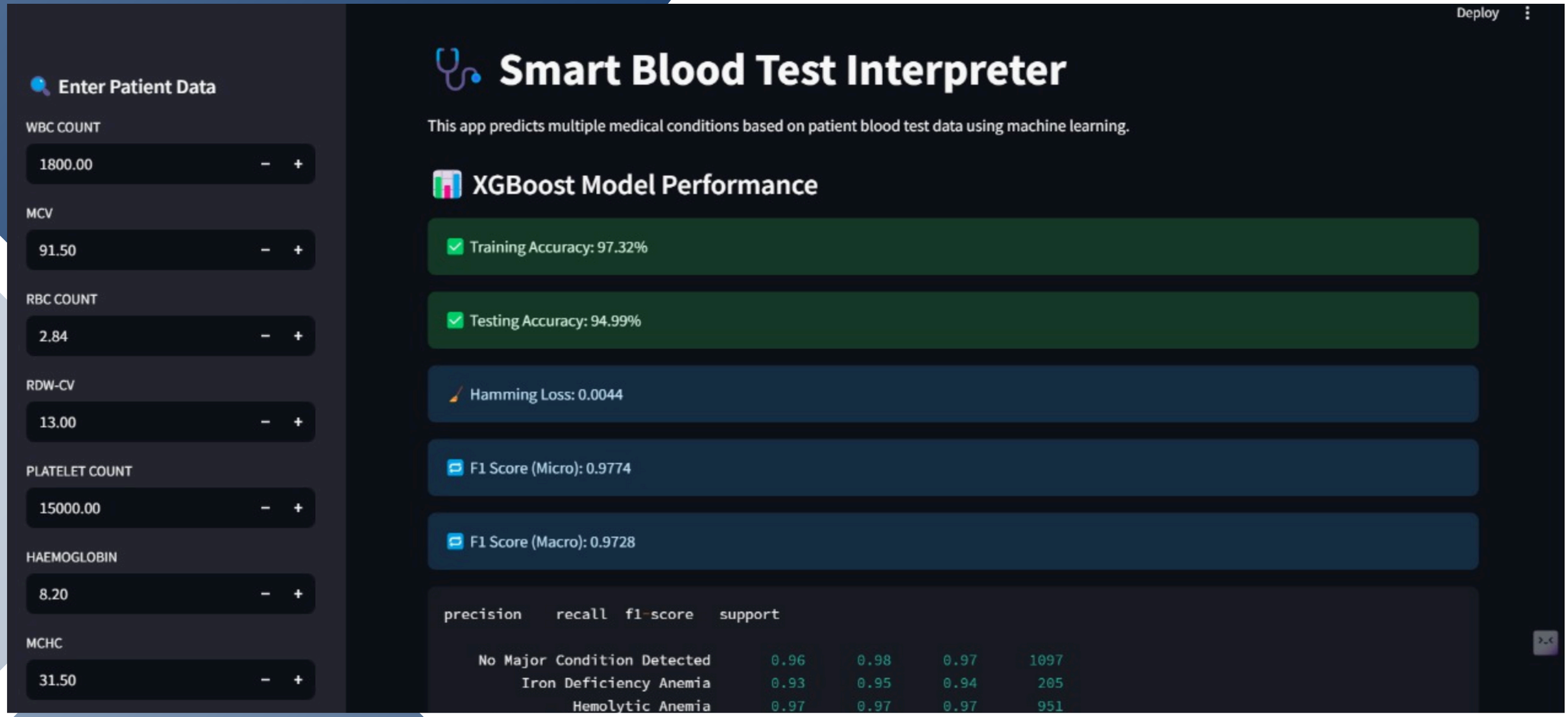
1. Frontend: Streamlit – lightweight, fast, Python-based GUI.
2. Functionality:

- Sidebar to input values for 20 blood parameters.
- Predict button runs real-time inference using the trained XGBoost model.
- Diseases with predicted probability ≥ 0.5 are marked as “Likely Present.”
- Interactive bar chart displays probability of all diseases.

3. Technologies Used:

- Plotly for charts, Scikit-learn for model loading, Streamlit widgets for UI/UX.
- App includes confetti balloons and warnings for user feedback.

VISUAL REPRESENTATION



Visual Output Example

1. Input: User enters values like Hemoglobin, Platelet Count, Neutrophil %, etc.
2. Output:

- Bar chart of probabilities for 18 diseases.
- Each bar shows likelihood (0 to 1) for a specific disease.
- Values > 0.5 marked with “⚠ Warning” and highlighted in red.

3. Interactivity:

- Uses Plotly dynamic visualization.
- Allows doctors to quickly focus on top predicted conditions without scanning raw numbers.

Conclusion and Future Directions

1. Conclusion:

- Successfully implemented a predictive system for multi-disease detection using blood test data.
- Achieved over 94% test accuracy using XGBoost.
- Developed a working, interactive web interface for real-time usage.

2. Future Work:

- Handle imbalanced data more efficiently (e.g., SMOTE, ensemble stacking).
- Add contextual features like patient history, vitals, symptoms.
- Expand to support lab integrations (e.g., HL7, FHIR standards).
- Deploy cloud-based version with patient profile storage and doctor login.

THANK YOU!