FINAL REPORT — Project 3
Nitishwar Vasantha Kumar

Introduction and Overview

In this project, I tackled a binary-classification problem with the goal of building and comparing predictive models. After thoroughly inspecting, cleaning, and preparing the dataset, I developed a baseline Logistic Regression model. I then analyzed more advanced models Random Forest, Gradient Boosting, XGBoost, LightGBM, and a Multi-Layer Perceptron (MLP) and applied hyperparameter tuning to XGBoost for potential performance gains.

The final objective was to contrast logistic and non-logistic models, analyze their behavior, and conclude which model is most effective, including how I'd communicate that choice to a non-technical business audience.

Data Cleaning and Preparation

I began by reviewing the dataset's structure, including data types, missing values, and the balance between classes. The binary target variable was reasonably balanced, which is beneficial for model training. I removed any constant or low-variance features that did not contribute information. For missing data, I used median imputation for numeric features and mode imputation for categorical ones, and dropped rows only when appropriate. Categorical variables were encoded, and numeric features were scaled using StandardScaler, which is important for algorithms sensitive to feature magnitude like Logistic Regression and MLP. I also created visualizations such as correlation heatmaps, class-wise distributions, and pairwise feature relationships to gain insights into the data. Finally, I performed a train-test split to fairly evaluate the model's generalization performance.

Part A — Logistic Regression Model

What is Logistic Regression?

Logistic Regression is a linear model for binary classification. It estimates the probability of a class using the logistic function.

- Advantages: Simple, interpretable coefficients, fast to train, outputs probabilities, robust when relationships are linear.
- Disadvantages: Limited to linear relations, cannot model feature interactions or complex patterns, may underperform on complex data.

Feature Selection and Importance

I examined feature importances using absolute coefficient magnitudes and performed feature selection (removing redundant or weak predictors). I also used SHAP values for a more granular view of each feature's impact.

Feature Engineering

Discretization (binning) was used for certain continuous variables to help expose nonlinear effects.

Cross-validation and Tuning

Stratified k-fold cross-validation (k=5) was used for stable evaluation, and regularization was tuned by searching for optimal penalty strength.

Results — Logistic Regression

Metric	**Value**
Training ROC AUC	0.9264
Validation ROC AUC	0.9247
Test Accuracy	0.8740
Test F1 Score	0.8512
Test ROC AUC	0.9267

Classification Report — Test Set:

Class	**Precision**	**Recall**	**F1 Score**
0	0.88	0.90	0.89
1	0.87	0.84	0.85

The ROC curve and confusion matrix indicate good separation, but there's noticeable room for improvement.

Part B — Non-Logistic Models

Model Choice & Advantages/Disadvantages

I analyzed Random Forest, Gradient Boosting, XGBoost, LightGBM, and MLP (neural network). These models are well-suited to capturing nonlinear relationships and feature interactions:

- Advantages: Handle nonlinearity, mixed data types, outliers, missing data; robust performance.

- Disadvantages: Less interpretable than Logistic Regression, longer training time, need tuning to avoid overfitting.

Hyperparameter Tuning and Validation
Hyperparameters were tuned (max_depth, learning rate, n_estimators, subsample, etc.) for each model using RandomizedSearchCV (3-fold cross-validation for XGBoost).

Comparative Results (Test Set)

Model	**Accuracy**	**F1 Score**	**AUC**
Tuned XGBoost	0.9631	0.9565	0.9947
XGBoost	0.9621	0.9554	0.9941
LightGBM	0.9619	0.9551	0.9941
Gradient Boosting	0.9616	0.9548	0.9940
MLP	0.9563	0.9486	0.9924
Random Forest	0.9329	0.9216	0.9825
Logistic Regression	0.8740	0.8512	0.9267

Tuned XGBoost — Final Model Results

Class	**Precision**	**Recall**	**F1 Score**

0	0.96	0.98	0.97
1	0.97	0.94	0.96

Test Accuracy: 0.9631
Test F1 Score: 0.9565
Test ROC AUC: 0.9947

The ROC curve for tuned XGBoost indicated exceptional predictive power with almost no misclassifications.

Training Challenges
Non-logistic models, especially XGBoost, took longer to tune and needed careful regularization to prevent overfitting.

Discussion and Model Comparison
The accuracy jump from 87% (logistic) to 96% (tuned XGBoost) is significant. Likewise, AUC improves from ~0.93 to ~0.995. The dramatic improvement is because boosted trees model feature interactions and complex decision boundaries that logistic regression cannot.

Performance Comparison Table

Metric Type	**Logistic Regression**	**Tuned XGBoost**
Training AUC	0.9264	0.9987
Validation AUC	0.9247	0.9947
Test AUC	0.9267	0.9947
Accuracy	87.40%	96.31%

F1 Score	85.12%	95.65%

Business-Friendly Summary

If I were describing this to a business leader, I would say:

Think of the Logistic Regression model as a straightforward checklist. It carefully reviews the data and makes decisions based on a fixed set of simple rules. For example, if one factor is high and another is low, it predicts ‘Yes.’ It’s dependable and easy to understand, and it gets the right answer about 88% of the time.

The XGBoost model, on the other hand, works more like a panel of experts. Instead of a single checklist, it builds hundreds of small decision trees, where each one focuses on correcting the mistakes made by the previous trees. By learning from its own errors, it captures complex subtle patterns and exceptions that the checklist misses.

Thanks to this self-improving ‘committee’ approach, XGBoost achieves accuracy of about 96%. While the checklist offers a solid baseline, the expertise of the committee significantly boosts the ability to identify challenging cases that really matter.

Conclusion

Through systematic data cleaning, exploratory data analysis, baseline and advanced modeling, and tuning, I identified Tuned XGBoost as the best model: highly accurate, robust, and generalizing well. It is well-suited for real-world deployment and represents a substantial upgrade from traditional logistic regression