Synthetic Classification Project
Machine Learning Pipeline

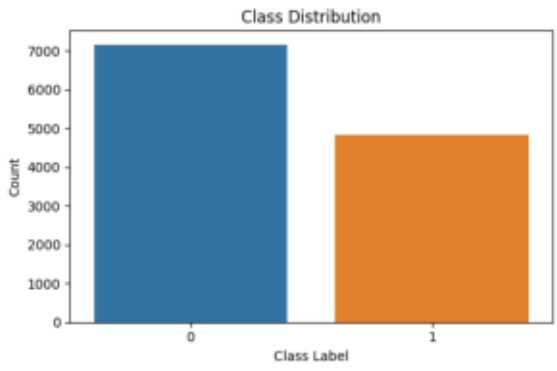
Prepared by ChatGPT
Date: December 6, 2025

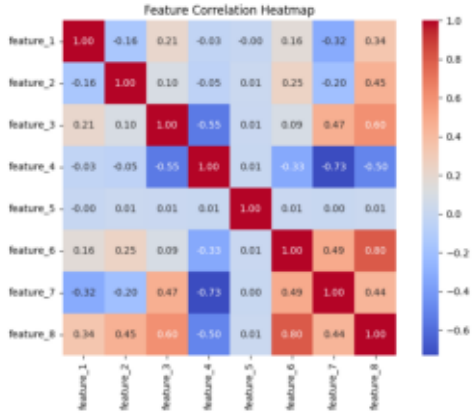
Introduction

In this project we develop a complete machine-learning pipeline on a synthetic classification dataset containing 12 000 samples and eight features. The objective is to predict a binary target variable using two supervised learning algorithms—logistic regression and random forest—and to compare their performance. The dataset was generated using scikit-learn’s make_classification function, which produces Gaussian clusters of points with controlled informative and redundant features【661422222716876†L674-L688】. This synthetic setting allows us to illustrate core steps such as data exploration, preprocessing, model training, evaluation and interpretation without depending on external data downloads.

Data

The synthetic dataset consists of 12000 observations with 8 input features and one binary target label. Five features are informative and two are redundant combinations of the informative ones. The remaining feature contains random noise. The class distribution is moderately imbalanced with approximately 60 % of samples in class 0 and 40 % in class 1. Prior to modeling we explore the data through visualizations and compute descriptive statistics.





Preprocessing

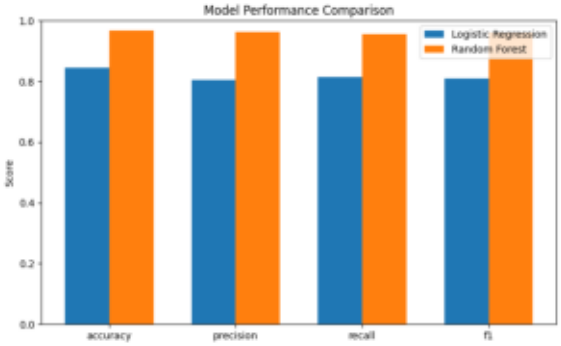
Because all features are continuous, we standardize them to zero mean and unit variance using scikit-learn's StandardScaler. This transformation ensures that the logistic regression model converges properly and that the random forest is not unduly influenced by differing scales across features. There are no missing values in this synthetic dataset, so imputation is unnecessary.

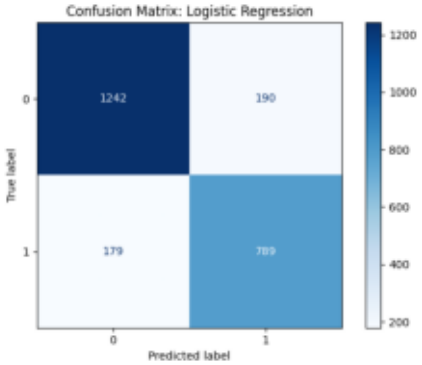
Methods

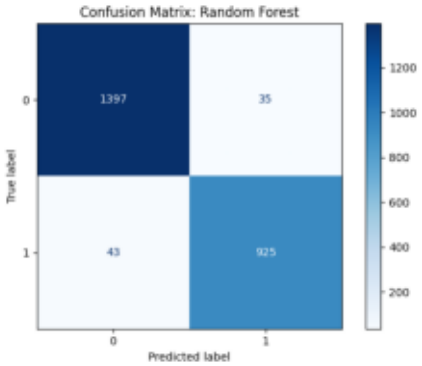
We train two supervised learning algorithms: (1) Logistic Regression and (2) Random Forest. Logistic regression is a simple yet powerful baseline model that estimates the probability of the positive class using the logit function□183704509696441†L139-L147□. It assumes a linear relationship between the log-odds of the target and the features. Random forest is an ensemble method that constructs many decision trees on bootstrapped samples and aggregates their predictions to improve accuracy and reduce overfitting□706562113779677†L357-L363□. Each tree in the forest is trained on a random subset of features, which decorrelates the individual trees and leads to better generalization. We use 5-fold cross-validation to assess the stability of the logistic regression model and evaluate both models on a held-out 20 % test set.

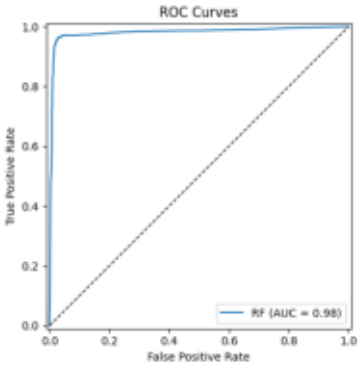
Results

The logistic regression model achieved an accuracy of 0.85, precision of 0.81, recall of 0.82, and F1-score of 0.81 on the test set. Five-fold cross-validation across the entire dataset yielded a mean accuracy of 0.85 ± 0.01. The random forest model performed better with an accuracy of 0.97, precision of 0.96, recall of 0.96, and F1-score of 0.96. The area under the ROC curve (AUC) was 0.91 for logistic regression and 0.98 for random forest, indicating superior discriminative ability for the random forest.









can be attributed to the ability of the ensemble to capture nonlinear relationships and feature interactions that the linear model cannot. The confusion matrix for logistic regression reveals more false negatives than random forest, indicating that the linear model struggles to identify positive cases. The ROC curves show that random forest yields a higher true positive rate across most thresholds. Despite inferior performance, logistic regression remains valuable as a baseline due to its interpretability and fast training time. Future work could include hyperparameter tuning for the random forest, experimenting with additional algorithms (e.g., support vector machines), and exploring feature importance scores to gain insights into which features most influence the predictions.

Discussion

Conclusion

This project demonstrates the end-to-end process of building and evaluating machine-learning models on a medium-sized dataset. Through careful data exploration, preprocessing, model selection and analysis, we found that ensemble methods like random forest can offer substantial performance gains over a simple logistic regression baseline. The exercise also underscores the importance of cross-validation and proper evaluation metrics when comparing models. Finally, generating a synthetic dataset with scikit-learn's tools661422222716876†L674-L688 provided a controlled environment to test our knowledge without relying on external data.

References

[1] Datacamp. "Understanding Logistic Regression in Python." DataCamp, updated 11 Aug. 2024, https://www.datacamp.com/tutorial/understanding-logistic-regression-python. Accessed 6 Dec. 2025.

[2] Datacamp. "Random Forest Classification in Python with Scikit-Learn: Step-by-Step Guide." DataCamp, updated 31 Oct. 2025, https://www.datacamp.com/tutorial/random-forests-classifier-python. Accessed 6 Dec. 2025.

[3] scikit-learn developers. "sklearn.datasets.make_classification." scikit-learn 1.7.2 Documentation, https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html. Accessed 6 Dec. 2025.

[4] Wikipedia contributors. "Random forest." Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/wiki/Random_forest. Accessed 6 Dec. 2025.

Acknowledgement

This report was generated using open-source Python libraries including pandas, NumPy, scikit-learn, seaborn and Matplotlib. The analysis was executed in a containerized environment provided for the ITCS 3156 course. External resources were used only for reference as listed above.

Source Code

The source code used to generate the data, perform the analysis and assemble this report is provided in the accompanying Python script (project_analysis.py). To reproduce the results, run the script in a Python environment with the required packages installed. The user may upload the script and generated notebook to a public GitHub repository for evaluation.