

# Phase 5c: XGBoost Model

Peter Mangoro

2025-12-07

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Load Data</b>	<b>1</b>
<b>3</b>	<b>Train/Test Split</b>	<b>1</b>
<b>4</b>	<b>Data Preparation for XGBoost</b>	<b>2</b>
<b>5</b>	<b>Build XGBoost Model</b>	<b>2</b>
<b>6</b>	<b>Variable Importance</b>	<b>4</b>
<b>7</b>	<b>Model Evaluation</b>	<b>5</b>
<b>8</b>	<b>ROC Curve</b>	<b>5</b>
<b>9</b>	<b>Summary</b>	<b>7</b>

## 1 Introduction

This document presents Phase 5c: XGBoost Model. We build an extreme gradient boosting model, analyze variable importance, and evaluate model performance.

## 2 Load Data

Dataset: 24996 observations, 18 variables

## 3 Train/Test Split

Training set: 17498 observations (70%)

Testing set: 7498 observations (30%)

## 4 Data Preparation for XGBoost

Data prepared for XGBoost:

Training features: 17

Training samples: 17498

Test samples: 7498

## 5 Build XGBoost Model

```
# Set parameters
params <- list(
  objective = "binary:logistic",
  eval_metric = "auc",
  max_depth = 6,
  eta = 0.1,
  subsample = 0.8,
  colsample_bytree = 0.8,
  min_child_weight = 1
)

# Train model
model_xgb <- xgb.train(
  params = params,
  data = dtrain,
  nrounds = 200,
  evals = list(train = dtrain, test = dtest),
  early_stopping_rounds = 20,
  verbose = 1
)
```

Multiple eval metrics are present. Will use test\_auc for early stopping.  
Will train until test\_auc hasn't improved in 20 rounds.

```
[1] train-auc:0.658557  test-auc:0.637749
[2] train-auc:0.661825  test-auc:0.640261
[3] train-auc:0.672553  test-auc:0.652316
[4] train-auc:0.676902  test-auc:0.655880
[5] train-auc:0.679827  test-auc:0.655335
[6] train-auc:0.681499  test-auc:0.654926
[7] train-auc:0.682365  test-auc:0.655623
[8] train-auc:0.683948  test-auc:0.656352
[9] train-auc:0.685574  test-auc:0.656026
[10]   train-auc:0.687852  test-auc:0.656963
[11]   train-auc:0.689678  test-auc:0.657575
[12]   train-auc:0.690788  test-auc:0.658325
[13]   train-auc:0.692323  test-auc:0.658499
[14]   train-auc:0.693297  test-auc:0.658487
```

```
[15] train-auc:0.695013 test-auc:0.658992
[16] train-auc:0.697169 test-auc:0.659924
[17] train-auc:0.698525 test-auc:0.660018
[18] train-auc:0.699297 test-auc:0.660365
[19] train-auc:0.700256 test-auc:0.660409
[20] train-auc:0.700876 test-auc:0.660358
[21] train-auc:0.702418 test-auc:0.660344
[22] train-auc:0.703845 test-auc:0.660396
[23] train-auc:0.704658 test-auc:0.660384
[24] train-auc:0.705393 test-auc:0.660452
[25] train-auc:0.706910 test-auc:0.660059
[26] train-auc:0.707772 test-auc:0.660464
[27] train-auc:0.709126 test-auc:0.660738
[28] train-auc:0.710278 test-auc:0.660682
[29] train-auc:0.711353 test-auc:0.660409
[30] train-auc:0.713046 test-auc:0.660781
[31] train-auc:0.714558 test-auc:0.660443
[32] train-auc:0.715305 test-auc:0.660236
[33] train-auc:0.715757 test-auc:0.660504
[34] train-auc:0.716299 test-auc:0.660482
[35] train-auc:0.717727 test-auc:0.659965
[36] train-auc:0.718438 test-auc:0.660345
[37] train-auc:0.719661 test-auc:0.660181
[38] train-auc:0.720361 test-auc:0.660109
[39] train-auc:0.721356 test-auc:0.660780
[40] train-auc:0.722926 test-auc:0.661245
[41] train-auc:0.724267 test-auc:0.661124
[42] train-auc:0.724796 test-auc:0.661087
[43] train-auc:0.726153 test-auc:0.661412
[44] train-auc:0.726631 test-auc:0.661651
[45] train-auc:0.727568 test-auc:0.661790
[46] train-auc:0.729230 test-auc:0.661777
[47] train-auc:0.729642 test-auc:0.661752
[48] train-auc:0.731424 test-auc:0.661925
[49] train-auc:0.732575 test-auc:0.662412
[50] train-auc:0.733726 test-auc:0.662489
[51] train-auc:0.734395 test-auc:0.662350
[52] train-auc:0.735071 test-auc:0.661931
[53] train-auc:0.735961 test-auc:0.661683
[54] train-auc:0.736955 test-auc:0.661889
[55] train-auc:0.738275 test-auc:0.661480
[56] train-auc:0.740071 test-auc:0.661352
[57] train-auc:0.740794 test-auc:0.661421
[58] train-auc:0.741573 test-auc:0.661306
[59] train-auc:0.742724 test-auc:0.661557
[60] train-auc:0.743523 test-auc:0.661663
[61] train-auc:0.744492 test-auc:0.661851
[62] train-auc:0.745666 test-auc:0.662080
[63] train-auc:0.746991 test-auc:0.662114
[64] train-auc:0.748398 test-auc:0.662259
[65] train-auc:0.749381 test-auc:0.662172
[66] train-auc:0.749877 test-auc:0.661537
[67] train-auc:0.750661 test-auc:0.661420
[68] train-auc:0.751955 test-auc:0.661157
```

```
[69]    train-auc:0.753128  test-auc:0.661163
Stopping. Best iteration:
[70]    train-auc:0.754549  test-auc:0.660863

[70]    train-auc:0.754549  test-auc:0.660863

cat("\nXGBoost model fitted successfully!\n")
```

XGBoost model fitted successfully!

```
best_iter_val <- ifelse(is.null(model_xgb$best_iteration) || length(model_xgb$best_iteration) == 0,
                        200, as.numeric(model_xgb$best_iteration[1]))
cat("Best iteration: ", best_iter_val, "\n")
```

Best iteration: 200

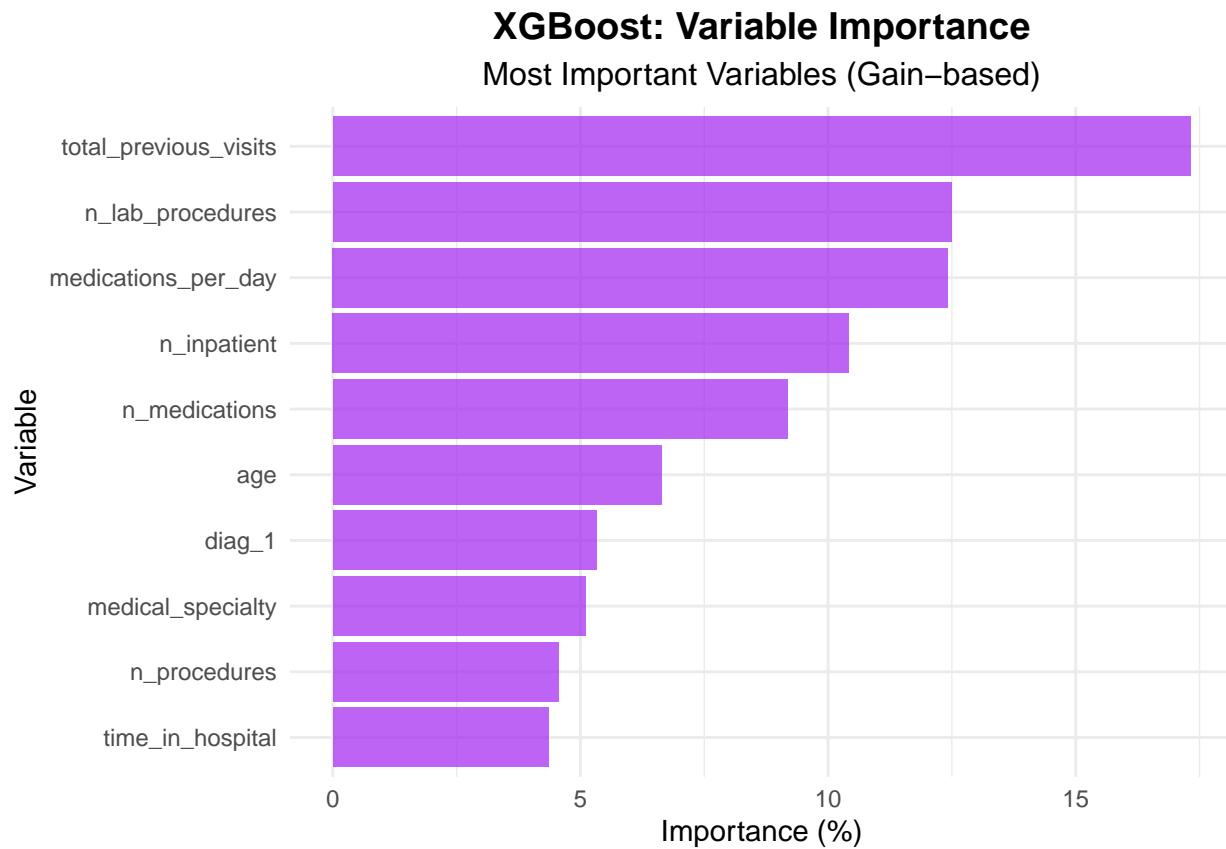
```
# Get best score from evaluation log
if(!is.null(model_xgb$evaluation_log) && nrow(model_xgb$evaluation_log) > 0 && best_iter_val <= nrow(model_xgb$evaluation_log)) {
  best_auc <- model_xgb$evaluation_log$test_auc[best_iter_val]
  cat("Best AUC: ", round(best_auc, 4), "\n")
} else {
  cat("Model training completed\n")
}
```

Model training completed

## 6 Variable Importance

Table 1: XGBoost: Most Important Variables

Variable	Gain	Cover	Frequency	Importance_Percent
total_previous_visits	0.1732	0.0851	0.0481	17.3179
n_lab_procedures	0.1249	0.1877	0.1724	12.4880
medications_per_day	0.1242	0.1248	0.1634	12.4222
n_inpatient	0.1042	0.0668	0.0297	10.4208
n_medications	0.0918	0.1303	0.1153	9.1827
age	0.0664	0.0632	0.0701	6.6356
diag_1	0.0533	0.0424	0.0681	5.3286
medical_specialty	0.0510	0.0621	0.0666	5.0974
n_procedures	0.0455	0.0405	0.0636	4.5549
time_in_hospital	0.0436	0.0348	0.0663	4.3603



## 7 Model Evaluation

Table 2: XGBoost: Confusion Matrix

	Not_Readmitted	Readmitted
Not_Readmitted	2944	1768
Readmitted	1029	1757

Table 3: XGBoost: Performance Metrics

Metric	Value	Percentage
Accuracy	0.63	62.70
Precision	0.63	63.07
Recall (Sensitivity)	0.50	49.84
Specificity	0.74	74.10
F1-Score	0.56	55.68

## 8 ROC Curve

Area Under the Curve (AUC): 0.6625

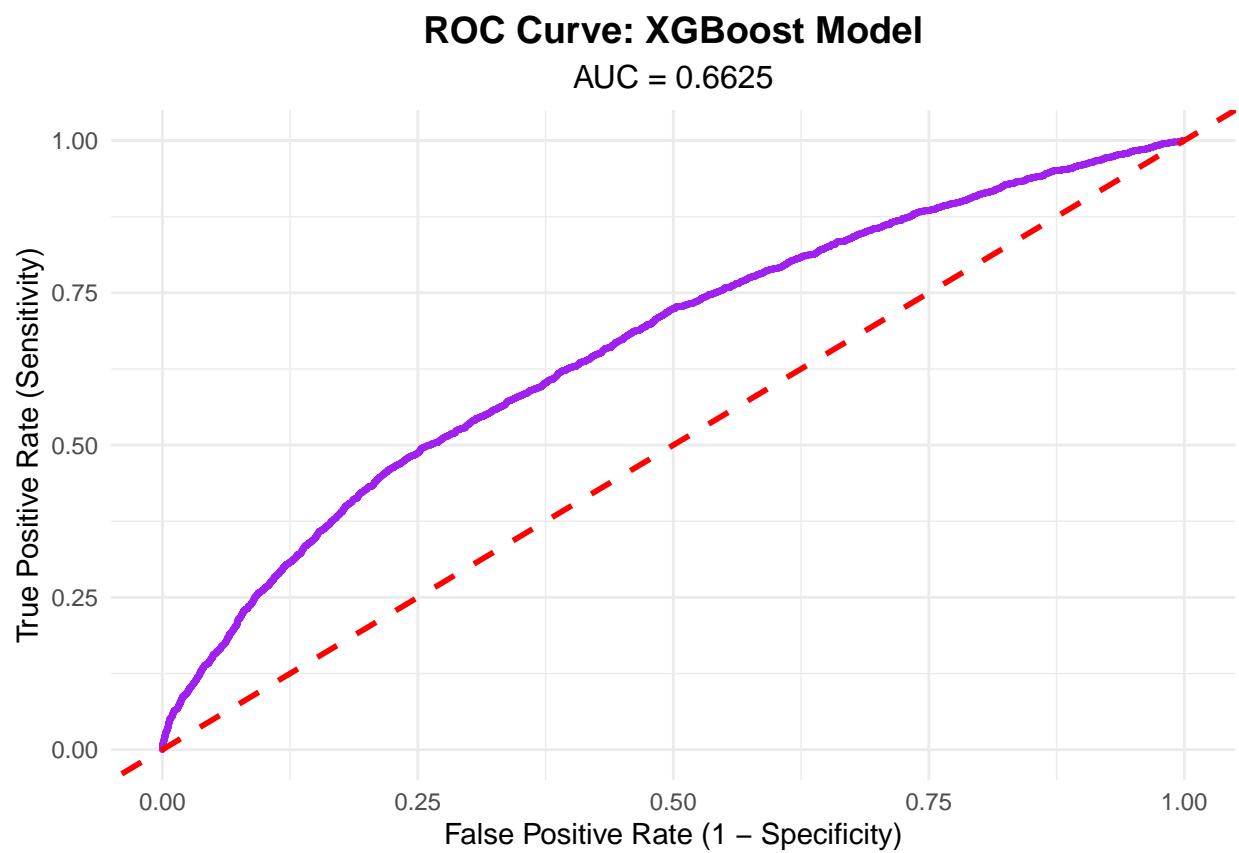


Figure 1: ROC Curve: XGBoost Model

## 9 Summary

This phase successfully built and evaluated the XGBoost model:

- **Accuracy:** 62.7%
- **AUC:** 0.662
- **Best iteration:**
- **Max depth:** 6
- **Learning rate (eta):** 0.1
- **Top predictor:** total previous visits (17.32% importance)
- **Interpretability:** Lower (gradient boosting ensemble)