Classification Presentation Slides

A Model to Help Prospective Patients Estimate the Likelihood of Mortality Following an Elective Surgery

George Pappy - 10 November 2021

Introduction

Cleveland Clinic performs thousands of elective surgeries per year:

Introduction

Cleveland Clinic performs thousands of elective surgeries per year:

- Knee and Hip Replacements

Introduction

Cleveland Clinic performs thousands of elective surgeries per year:

- Knee and Hip Replacements
- Nasal Procedures

Introduction

Cleveland Clinic performs thousands of elective surgeries per year:

- Knee and Hip Replacements
- Nasal Procedures
- Hernia Repairs

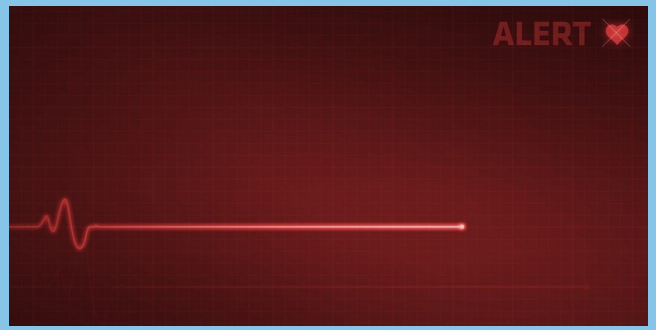
Introduction

Cleveland Clinic performs thousands of elective surgeries per year:

- Knee and Hip Replacements
- Nasal Procedures
- Hernia Repairs
- Many Others of Varying Severity

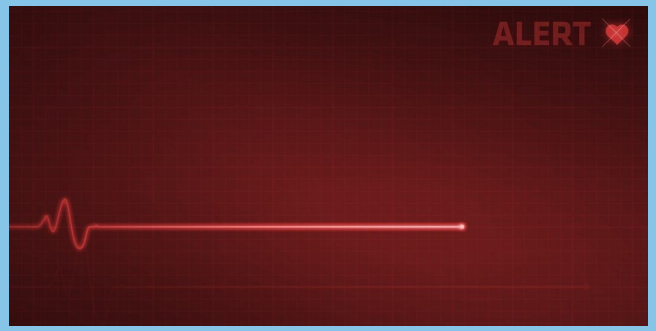
Introduction (con’t.)

A small percentage (≈ 0.414 %) of these surgeries result in patient morality within 30 days



Introduction (con’t.)

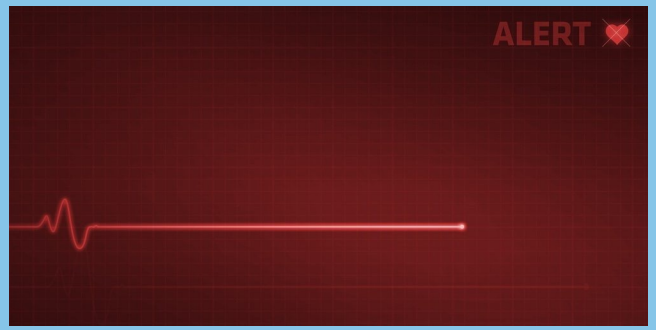
A small percentage (≈ 0.414 %) of these surgeries result in patient morality within 30 days



Patients have a right to know their own risk of post-surgical mortality

Introduction (con’t.)

A small percentage (≈ 0.414 %) of these surgeries result in patient morality within 30 days



Patients have a right to know their own risk of post-surgical mortality

⇒ **May influence their decision to proceed with a surgery that’s not strictly necessary**

Introduction (con't.)

Goal: Cleveland Clinic wants a predictive model that can give patients a sense of their elective surgery mortality risk

Introduction (con't.)

Goal: Cleveland Clinic wants a predictive model that can give patients a sense of their elective surgery mortality risk

This risk can be classified:

Introduction (con't.)

Goal: Cleveland Clinic wants a predictive model that can give patients a sense of their elective surgery mortality risk

This risk can be classified:

- No Elevated Risk of Mortality

Introduction (con't.)

Goal: Cleveland Clinic wants a predictive model that can give patients a sense of their elective surgery mortality risk

This risk can be classified:

- No Elevated Risk of Mortality
- Elevated Risk of Mortality

Introduction (con't.)

Goal: Cleveland Clinic wants a predictive model that can give patients a sense of their elective surgery mortality risk

This risk can be classified:

- No Elevated Risk of Mortality
- Elevated Risk of Mortality
  - Can then predict relative level of Elevated Risk

Introduction (con't.)

Goal: Cleveland Clinic wants a predictive model that can give patients a sense of their elective surgery mortality risk

This risk can be classified:

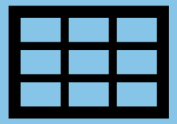
- No Elevated Risk of Mortality
- Elevated Risk of Mortality
  - Can then predict relative level of Elevated Risk
  - Risk (example): {*Low*, *Moderate*, *High*}

Methodology

- Data Set: 5 ½ years of elective surgeries at Cleveland Clinic

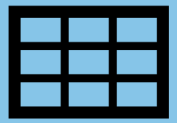
Methodology

- Data Set: 5 ½ years of elective surgeries at Cleveland Clinic
  - 28,287 rows (each is one surgical encounter)



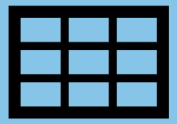
Methodology

- Data Set: 5 ½ years of elective surgeries at Cleveland Clinic
  - 28,287 rows (each is one surgical encounter)
  - 23 predictors (patient characteristics, diagnoses, surgical procedure, etc.)

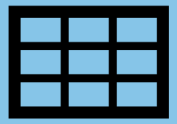


Methodology

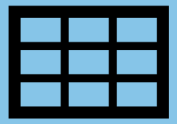
- Data Set: 5 ½ years of elective surgeries at Cleveland Clinic
  - 28,287 rows (each is one surgical encounter)
  - 23 predictors (patient characteristics, diagnoses, surgical procedure, etc.)
  - Target variable: Mortality within 30 days of surgery (1 = yes, 0 = no)



Methodology

- Data Set: 5 ½ years of elective surgeries at Cleveland Clinic
  - 28,287 rows (each is one surgical encounter) 
  - 23 predictors (patient characteristics, diagnoses, surgical procedure, etc.)
  - Target variable: Mortality within 30 days of surgery (1 = yes, 0 = no)
    - Highly imbalanced 

Methodology

- Data Set: 5 ½ years of elective surgeries at Cleveland Clinic
  - 28,287 rows (each is one surgical encounter) 
  - 23 predictors (patient characteristics, diagnoses, surgical procedure, etc.)
  - Target variable: Mortality within 30 days of surgery (1 = yes, 0 = no)
    - Highly imbalanced 
    - 99.586% negatives (0's) TO 0.414% positives (1's)

Methodology (con't.)

Methods & Tools

- **Pandas**: clean, explore, engineer features and generate final modeling data

Methodology (con't.)

Methods & Tools

- **Pandas**: clean, explore, engineer features and generate final modeling data
- **scikit-learn**: build classification models as well as to perform cross validation, variable selection and regularization

Methodology (con't.)

Methods & Tools

- **Pandas**: clean, explore, engineer features and generate final modeling data
- **scikit-learn**: build classification models as well as to perform cross validation, variable selection and regularization
- **Matplotlib/Seaborn**: visualizing data exploration, modeling and final results

Methodology (con't.)

Methods & Tools

- **Pandas**: clean, explore, engineer features and generate final modeling data
- **scikit-learn**: build classification models as well as to perform cross validation, variable selection and regularization
- **Matplotlib/Seaborn**: visualizing data exploration, modeling and final results
- **Python 3.8**: to run all of the above

Results

Six individual models and three ensemble models built:

Results

Six individual models and three ensemble models built:

1. **Logistic Regression** (regularization optimized for AUC metric)

Results

Six individual models and three ensemble models built:

1. **Logistic Regression** (regularization optimized for AUC metric)
2. **Logistic Regression** (regularization optimized for log-loss metric)

Results

Six individual models and three ensemble models built:

1. **Logistic Regression** (regularization optimized for AUC metric)
2. **Logistic Regression** (regularization optimized for log-loss metric)
3. **Random Forest Classifier** (hyperparameters optimized for AUC)

Results

Six individual models and three ensemble models built:

1. **Logistic Regression** (regularization optimized for AUC metric)
2. **Logistic Regression** (regularization optimized for log-loss metric)
3. **Random Forest Classifier** (hyperparameters optimized for AUC)
4. **Random Forest Classifier** (hyperparameters optimized for log-loss)

Results

Six individual models and three ensemble models built:

1. **Logistic Regression** (regularization optimized for AUC metric)
2. **Logistic Regression** (regularization optimized for log-loss metric)
3. **Random Forest Classifier** (hyperparameters optimized for AUC)
4. **Random Forest Classifier** (hyperparameters optimized for log-loss)
5. **XGBoost Classifier** (hyperparameters optimized for AUC)

Results

Six individual models and three ensemble models built:

1. **Logistic Regression** (regularization optimized for AUC metric)
2. **Logistic Regression** (regularization optimized for log-loss metric)
3. **Random Forest Classifier** (hyperparameters optimized for AUC)
4. **Random Forest Classifier** (hyperparameters optimized for log-loss)
5. **XGBoost Classifier** (hyperparameters optimized for AUC)
6. **XGBoost Classifier** (hyperparameters optimized for log-loss)

Results

Six individual models and three ensemble models built:

1. **Logistic Regression** (regularization optimized for AUC metric)
2. **Logistic Regression** (regularization optimized for log-loss metric)
3. **Random Forest Classifier** (hyperparameters optimized for AUC)
4. **Random Forest Classifier** (hyperparameters optimized for log-loss)
5. **XGBoost Classifier** (hyperparameters optimized for AUC)
6. **XGBoost Classifier** (hyperparameters optimized for log-loss)
7. **Ensemble: Voting Classifier** (combining all 6 models above)

Results

Six individual models and three ensemble models built:

1. **Logistic Regression** (regularization optimized for AUC metric)
2. **Logistic Regression** (regularization optimized for log-loss metric)
3. **Random Forest Classifier** (hyperparameters optimized for AUC)
4. **Random Forest Classifier** (hyperparameters optimized for log-loss)
5. **XGBoost Classifier** (hyperparameters optimized for AUC)
6. **XGBoost Classifier** (hyperparameters optimized for log-loss)
7. **Ensemble: Voting Classifier** (combining all 6 models above)
8. **Ensemble: Voting Classifier** (Logistic Regression + Random Forest + XGBoost, AUC-optimized)

Results

Six individual models and three ensemble models built:

1. **Logistic Regression** (regularization optimized for AUC metric)
2. **Logistic Regression** (regularization optimized for log-loss metric)
3. **Random Forest Classifier** (hyperparameters optimized for AUC)
4. **Random Forest Classifier** (hyperparameters optimized for log-loss)
5. **XGBoost Classifier** (hyperparameters optimized for AUC)
6. **XGBoost Classifier** (hyperparameters optimized for log-loss)
7. **Ensemble: Voting Classifier** (combining all 6 models above)
8. **Ensemble: Voting Classifier** (Logistic Regression + Random Forest + XGBoost, AUC-optimized)
9. **Ensemble: Voting Classifier** (Logistic Regression + XGBoost, AUC-optimized)

Results (con't.)

Goal: Maximize Recall (True Positive Rate, or TPR)

$$\text{Recall } (TPR) \equiv \frac{TP}{TP + FN}$$

Results (con't.)

Goal: Maximize Recall (True Positive Rate, or TPR)

$$\text{Recall } (TPR) \equiv \frac{TP}{TP + FN}$$

Minimize False Negatives (FN) so the model **misses as few Elevated Risk patients as possible**

Results (con't.)

Goal: Maximize Recall (True Positive Rate, or TPR)

$$\text{Recall } (TPR) \equiv \frac{TP}{TP + FN}$$

Minimize False Negatives (FN) so the model **misses as few Elevated Risk patients as possible**

- As FN decreases, TPR increases

Results (con't.)

Goal: Maximize Recall (True Positive Rate, or TPR)

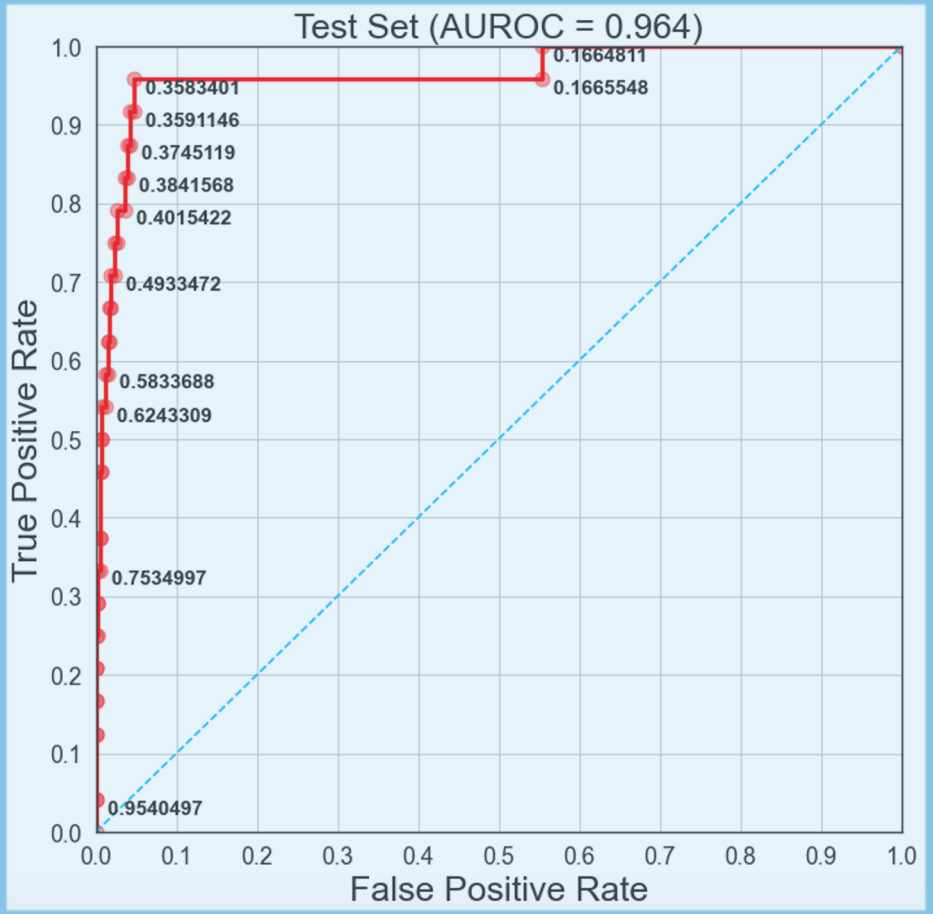
$$\text{Recall } (TPR) \equiv \frac{TP}{TP + FN}$$

Minimize False Negatives (FN) so the model **misses as few Elevated Risk patients as possible**

- As FN decreases, TPR increases
- But as FN decreases, False Positive Rate increases…

Results (con't.)

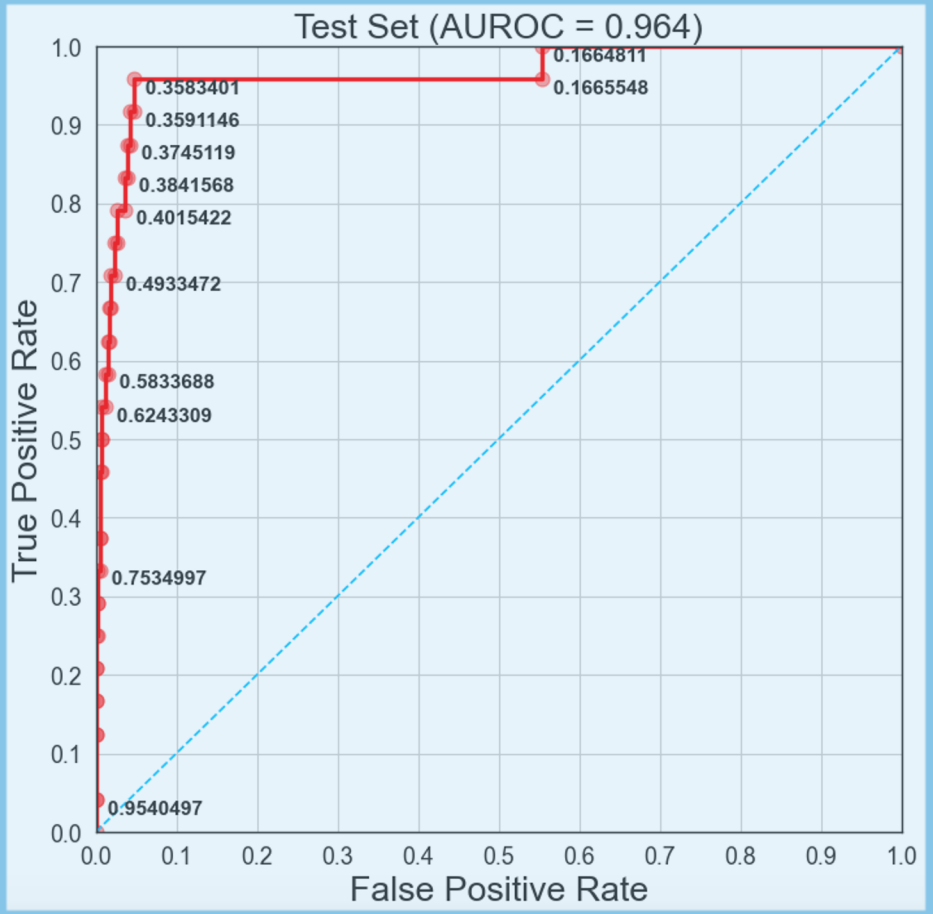
Best Model: Voting Classifier (soft)
(Logistic Regression + XGBoost, AUC-optimized)



Results (con't.)

Best Model: Voting Classifier (soft)
(Logistic Regression + XGBoost, AUC-optimized)

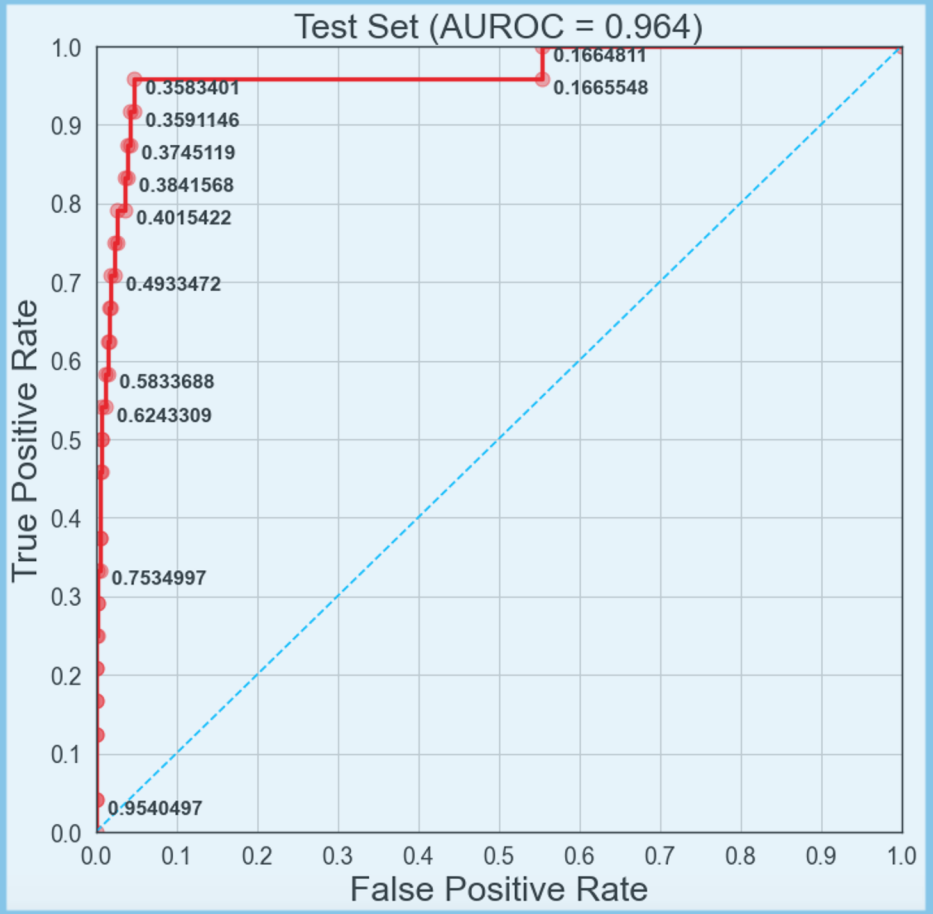
- Reasonable Performance Criteria:



Results (con't.)

Best Model: Voting Classifier (soft)
(Logistic Regression + XGBoost, AUC-optimized)

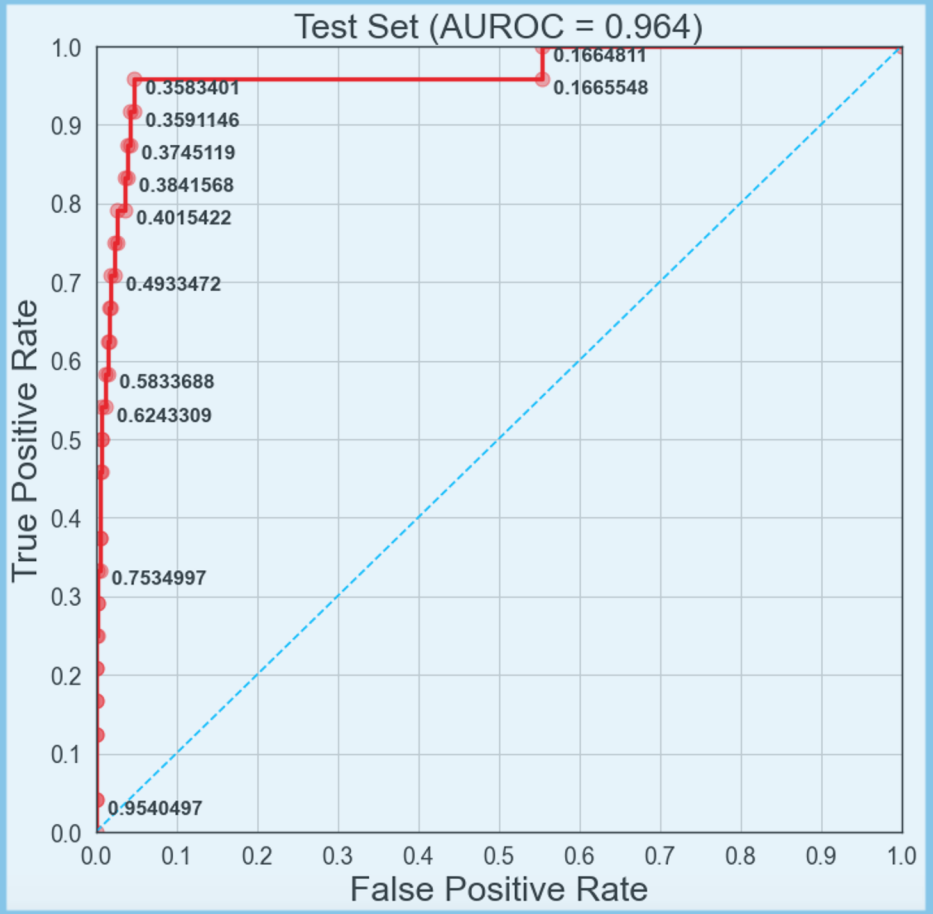
- Reasonable Performance Criteria:
  - >= 0.95 TPR



Results (con't.)

Best Model: Voting Classifier (soft)
(Logistic Regression + XGBoost, AUC-optimized)

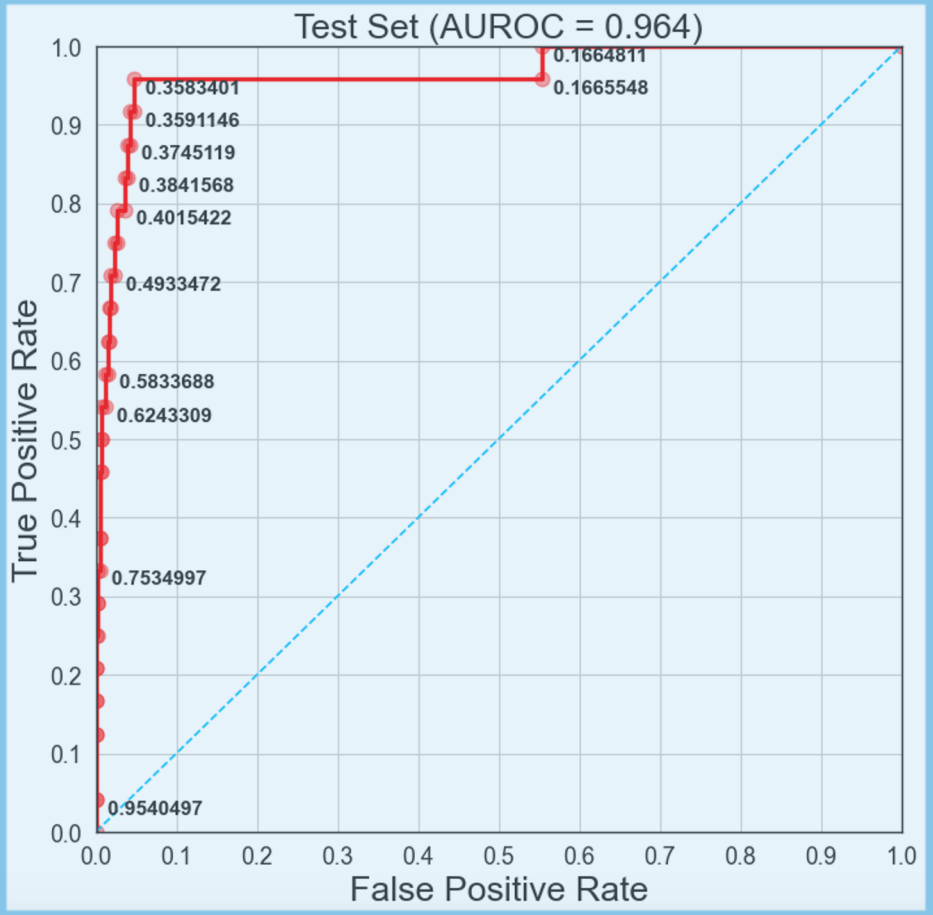
- Reasonable Performance Criteria:
  - >= 0.95 TPR
  - Minimize FPR as much as possible:



Results (con't.)

Best Model: Voting Classifier (soft)
(Logistic Regression + XGBoost, AUC-optimized)

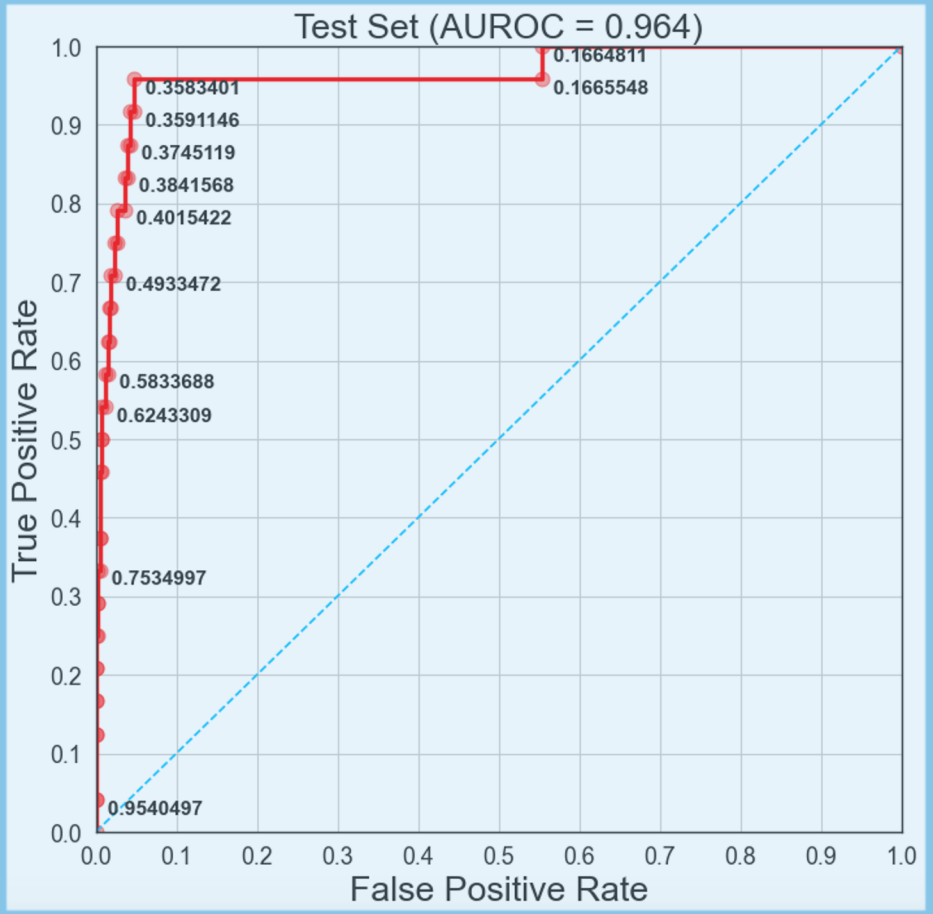
- Reasonable Performance Criteria:
  - >= 0.95 TPR
  - Minimize FPR as much as possible:
    - > 0.0 is acceptable



Results (con't.)

Best Model: Voting Classifier (soft)
(Logistic Regression + XGBoost, AUC-optimized)

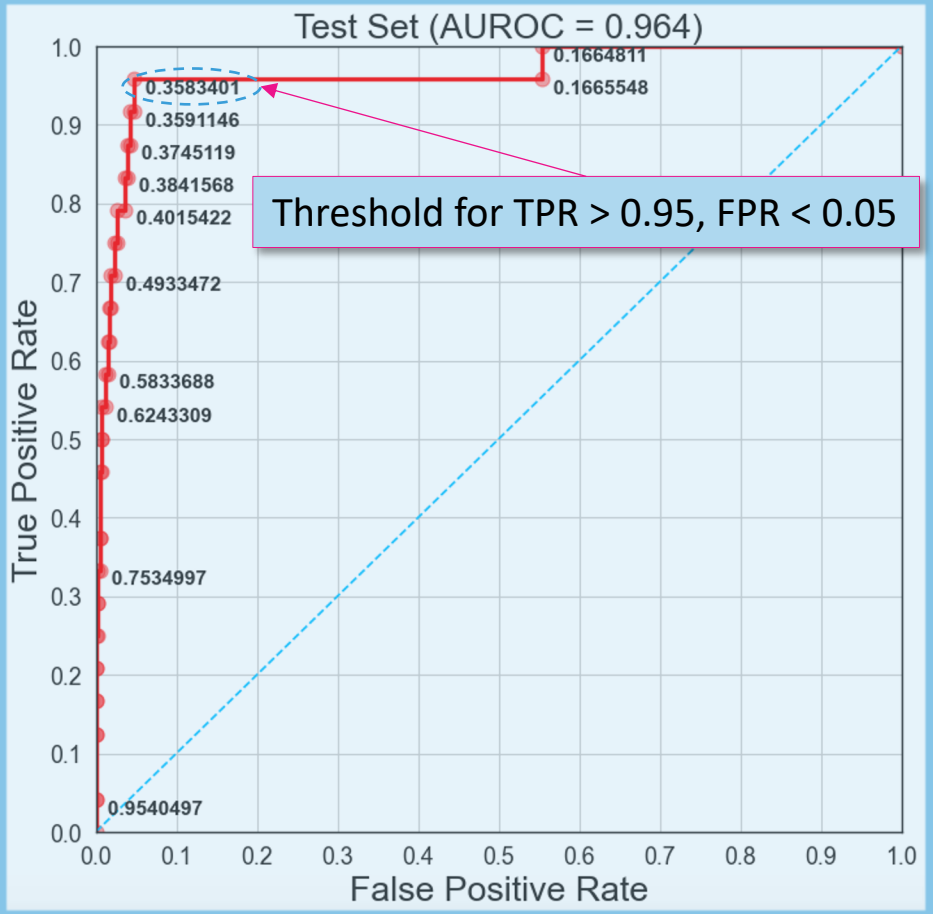
- Reasonable Performance Criteria:
  - >= 0.95 TPR
  - Minimize FPR as much as possible:
    - > 0.0 is acceptable
    - > 0.10 is undesirable



Results (con't.)

Best Model: Voting Classifier (soft)
(Logistic Regression + XGBoost, AUC-optimized)

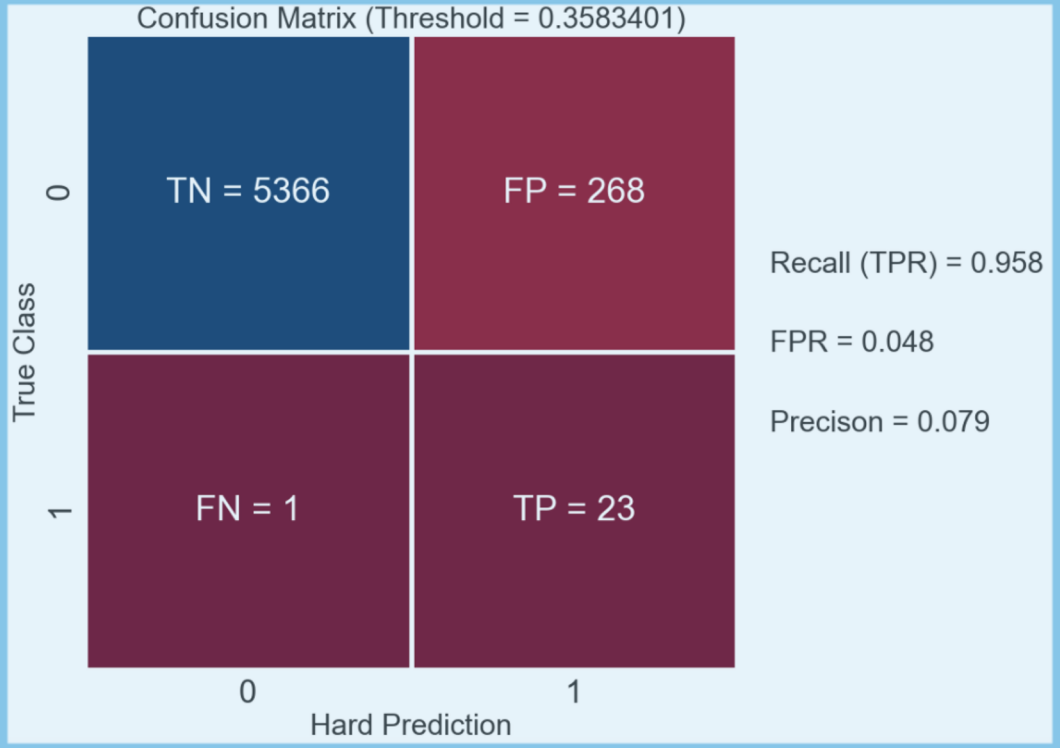
- Reasonable Performance Criteria:
  - >= 0.95 TPR
  - Minimize FPR as much as possible:
    - > 0.0 is acceptable
    - > 0.10 is undesirable



Results (con't.)

Best Model: Voting Classifier (soft)
(Logistic Regression + XGBoost, AUC-optimized)

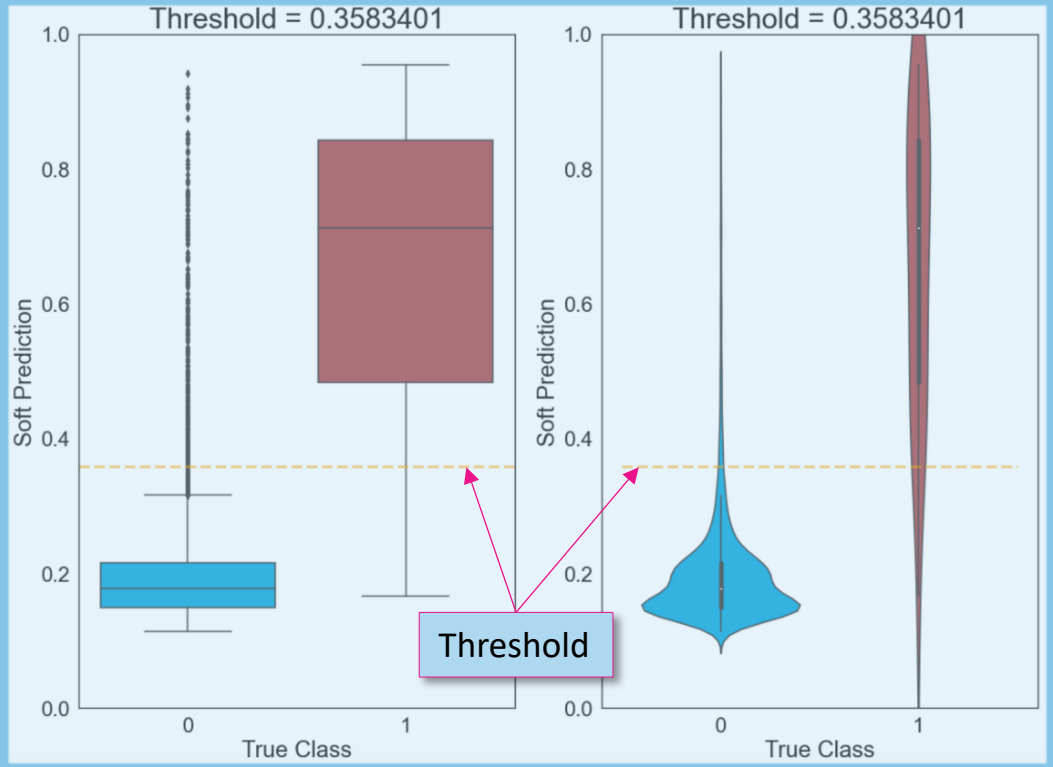
- Reasonable Performance Criteria:
  - >= 0.95 TPR
  - Minimize FPR as much as possible:
    - > 0.0 is acceptable
    - > 0.10 is undesirable



Results (con't.)

Best Model: Voting Classifier (soft)
(Logistic Regression + XGBoost, AUC-optimized)

- Reasonable Performance Criteria:
  - >= 0.95 TPR
  - Minimize FPR as much as possible:
    - > 0.0 is acceptable
    - > 0.10 is undesirable



Conclusions

Recommendations

- Deploy this model (Voting Classifier w/ Logistic Regression & XGBoost)
- Develop an explicit ranking scheme for the model's soft predictions with "Elevated Risk" patients

  → Example: {*Low, Moderate, High*}

Conclusions (con't.)

Future Work

- Track the deployed model's performance over time:

  → Does it maintain TPR > 0.95 and FPR ≈ 0.05 (i.e., current performance)?

- Retrain the model periodically as more data accrues (new surgeries)
- Build Deep Learning models for potentially better performance

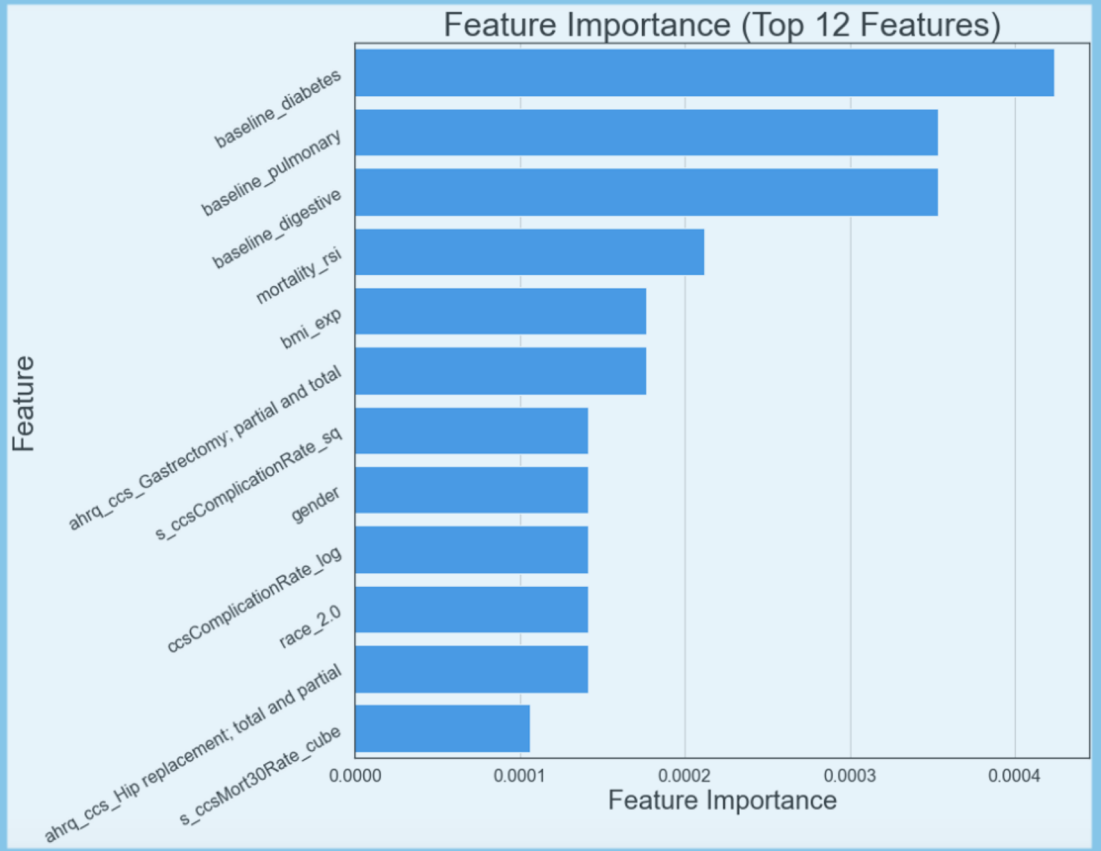
Appendix

Feature Importance

Best Model: Voting Classifier

(Logistic Regression + XGBoost, AUC-optimized)

- Interestingly, this model finds the most importance in much different features than the underlying models do (see the next few slides)
- Diabetes, pulmonary diagnoses and digestive disorders appear to play a significant role in mortality outcomes as judged by this model
- Body mass index and mortality_rsi (risk stratification index) also play non-trivial roles)
- Gastrectomy and Hip Replacement procedures appear to play outsized roles relative to other procedures
- Reliable interpretability may be an inherent weakness of this model

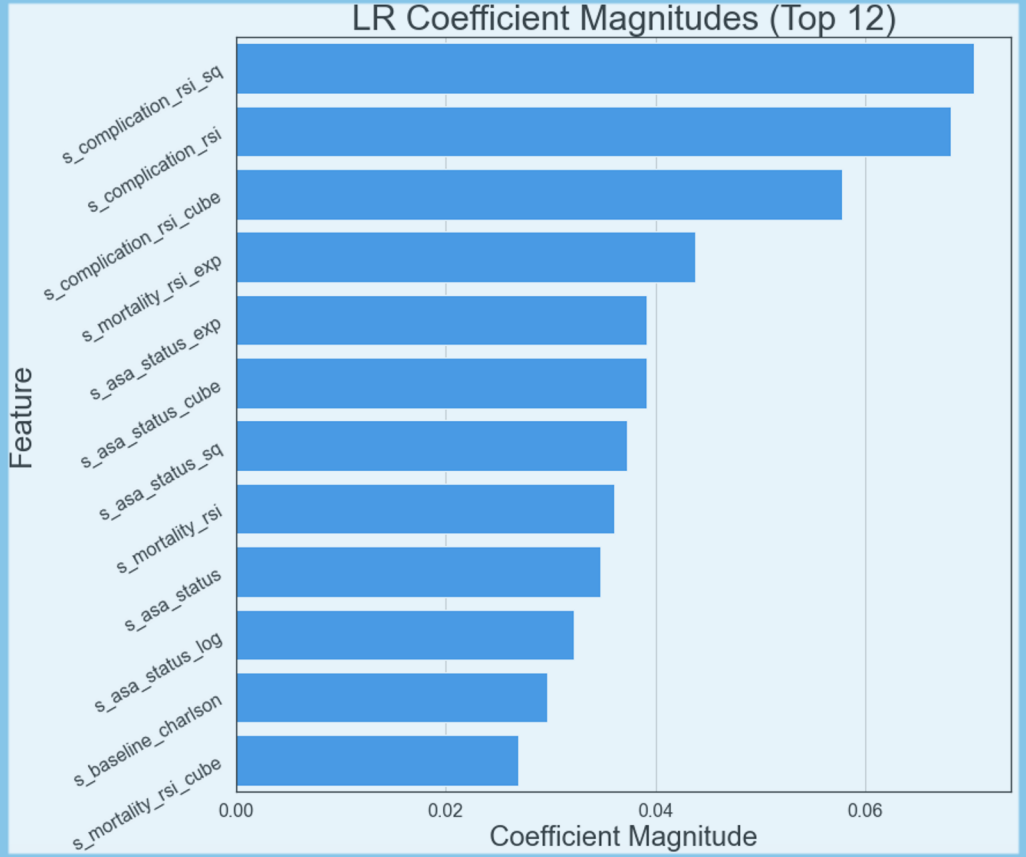


Feature Importance (con't.)

Best Model: Voting Classifier

(Logistic Regression + XGBoost, AUC-optimized)

- The underlying Logistic Regression model appears largely dependent upon features different than the most important ones found in the overall voting classifier
- This model depends most upon on multiple engineered features based on the same original variable (complication_rsi, mortality_rsi, asa_status)
- The one area of overlap between this model and the overall ensemble seems to be mortality_rsi (not surprising)

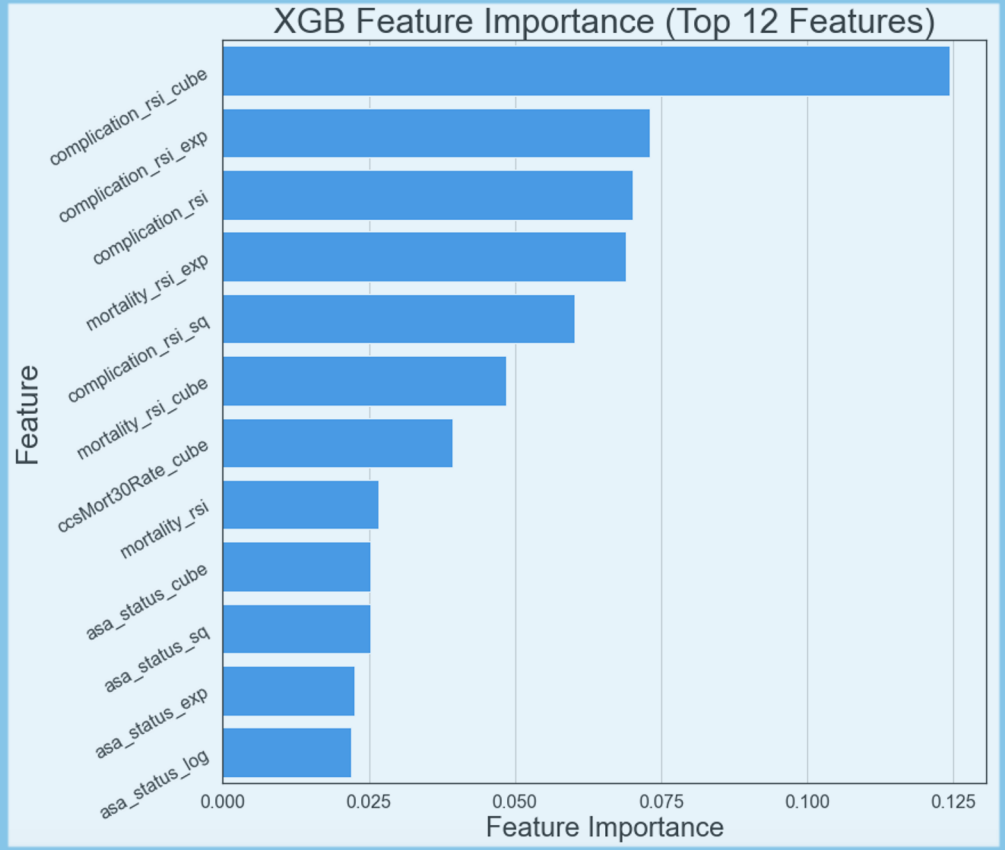


Feature Importance (con't.)

Best Model: Voting Classifier

(Logistic Regression + XGBoost, AUC-optimized)

- The underlying XGBoost model also appears largely dependent upon features different than the most important ones found in the overall voting classifier
- However, this model seems to agree with the underlying Logistic Regression in large part about the most significant features (complication_rsi, mortality_rsi, asa_status)
- The one area of overlap between this model and the overall ensemble seems to be mortality_rsi (just as with the Logistic Regression model)

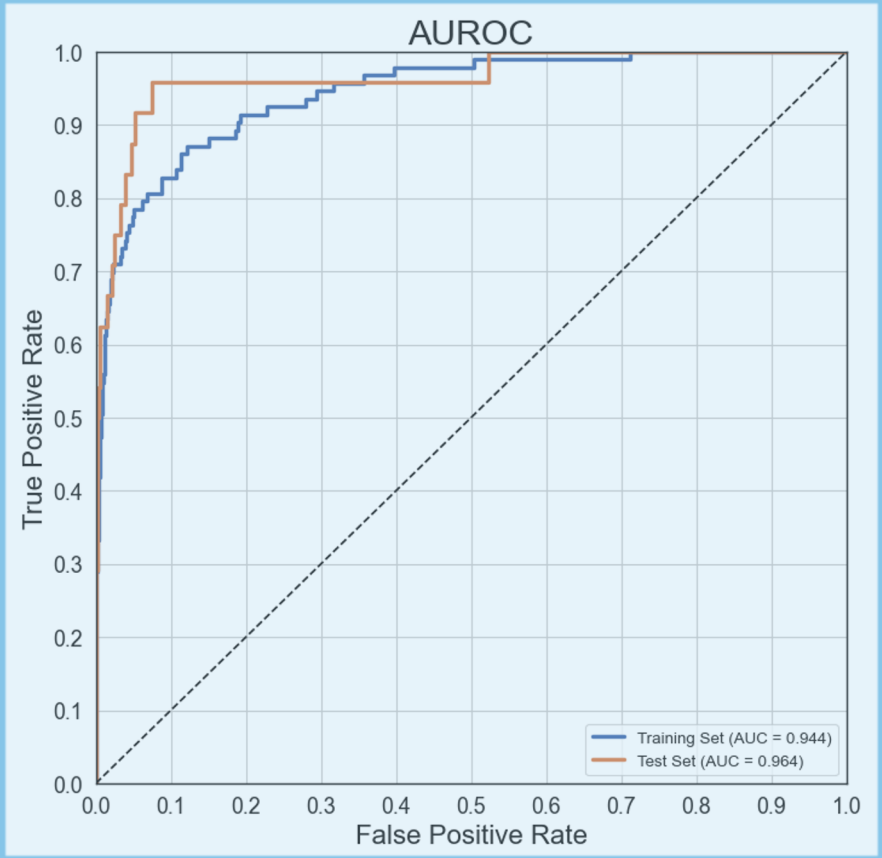


First Individual Model Developed

Logistic Regression

(Regularization parameters optimized to AUC)

- TPR = 0.958 achievable (same as final model)
- FPR = 0.074 for TPR = 0.958
  → Not as good as the final model (FPR = 0.048)

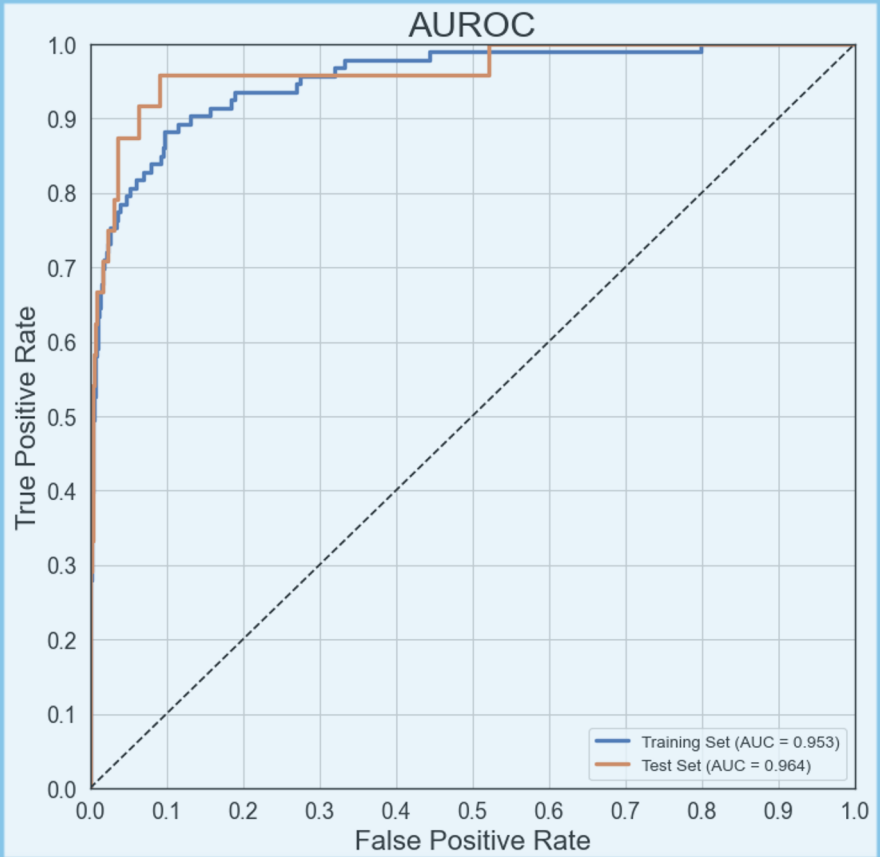


Second Individual Model Developed

Logistic Regression

(Regularization parameters optimized to log-loss)

- TPR = 0.958 achievable (same as final model)
- FPR = 0.091 for TPR = 0.958
  → Not as good as the final model (FPR = 0.048)

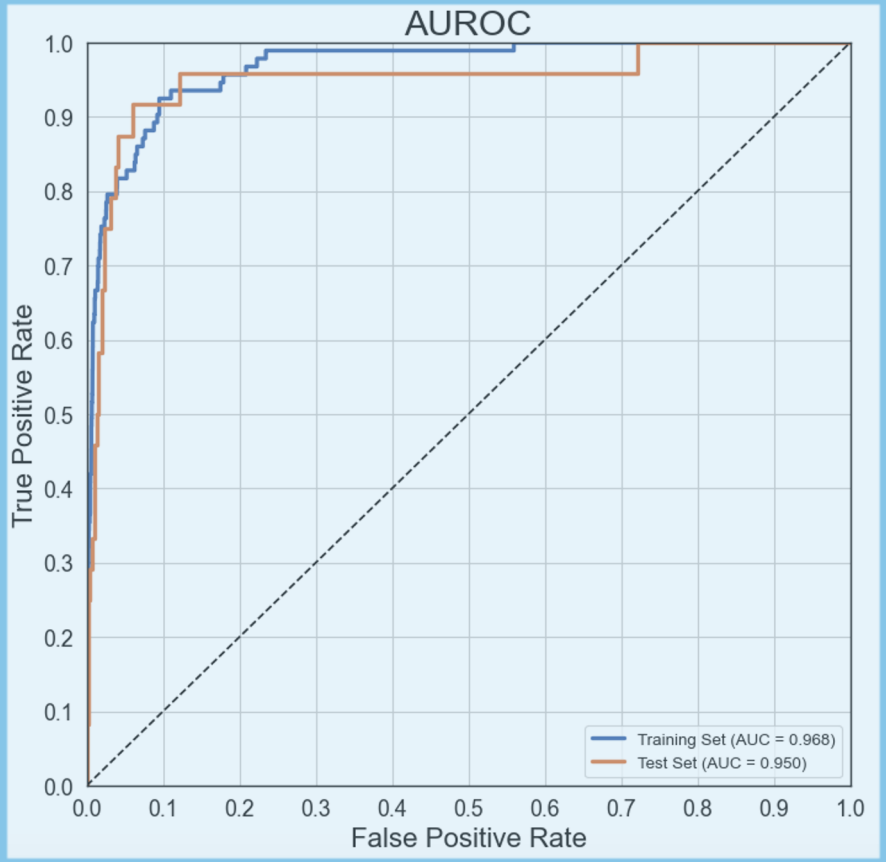


Third Individual Model Developed

Random Forest

(Regularization parameters optimized to AUC)

- TPR = 0.958 achievable (same as final model)
- FPR = 0.122 for TPR = 0.958
  → Not as good as the final model (FPR = 0.048)

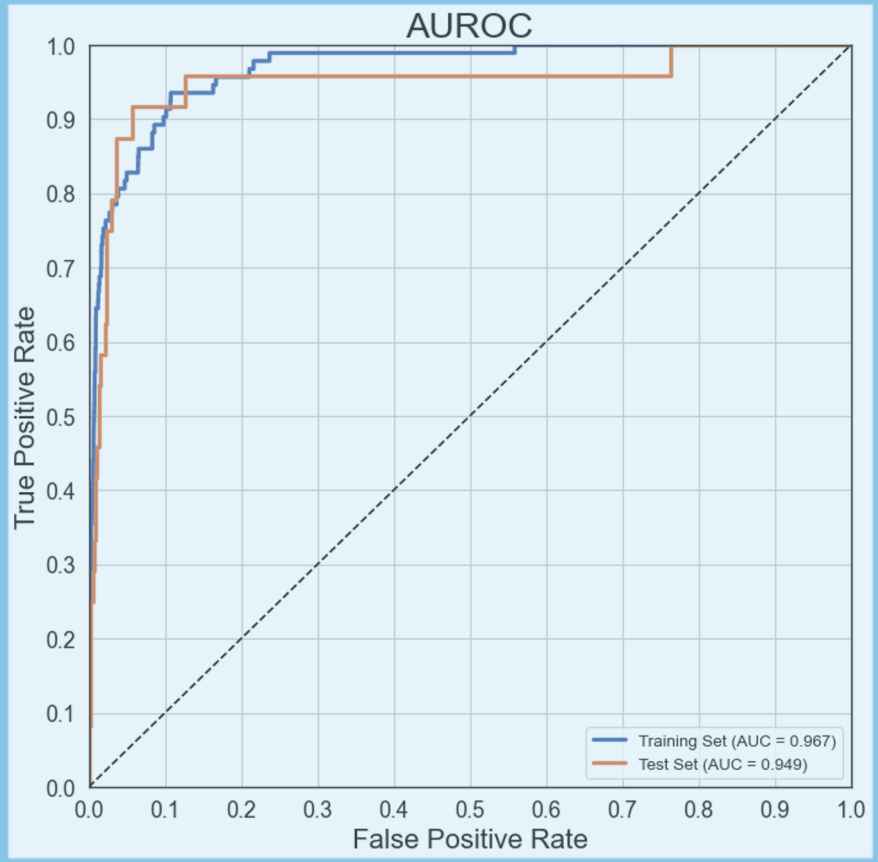


Fourth Individual Model Developed

Random Forest

(Regularization parameters optimized to log-loss)

- TPR = 0.958 achievable (same as final model)
- FPR = 0.126 for TPR = 0.958
  → Not as good as the final model (FPR = 0.048)

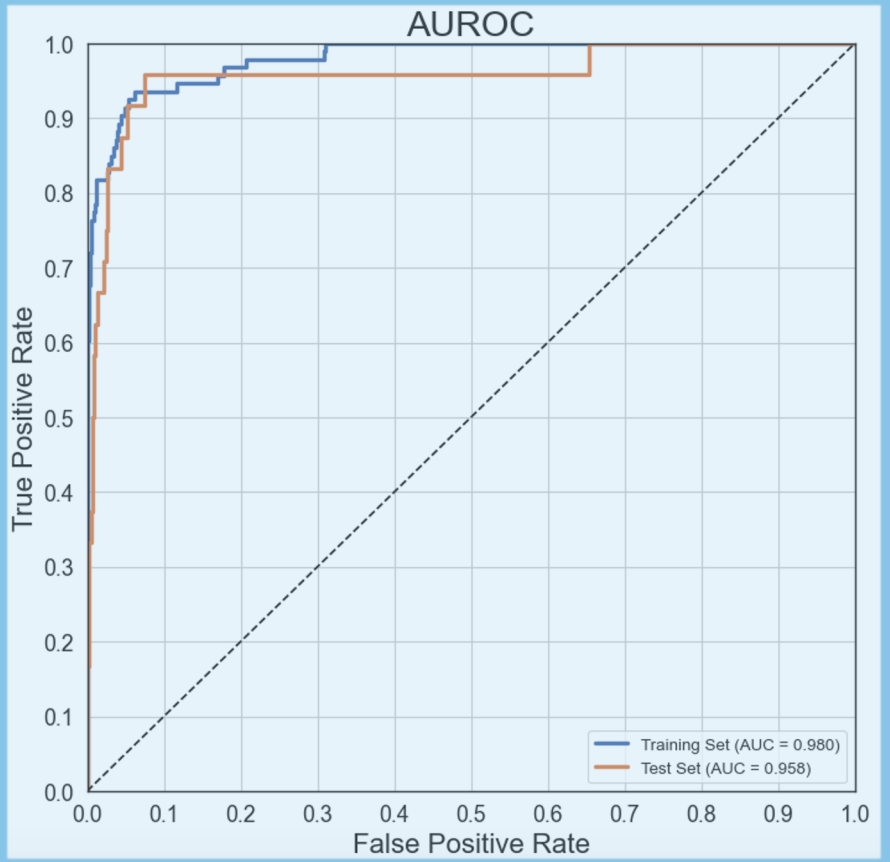


Fifth Individual Model Developed

XGBoost

(Regularization parameters optimized to log-loss)

- TPR = 0.958 achievable (same as final model)
- FPR = 0.074 for TPR = 0.958
  → Not as good as the final model (FPR = 0.048)

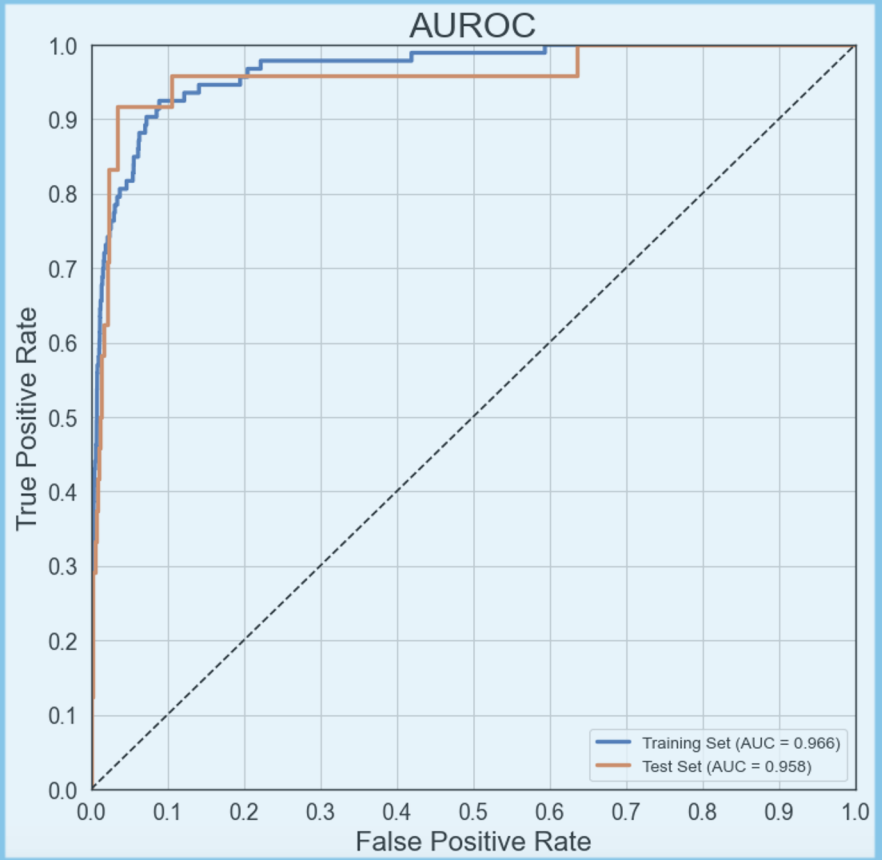


Sixth Individual Model Developed

XGBoost

(Regularization parameters optimized to AUC)

- TPR = 0.958 achievable (same as final model)
- FPR = 0.105 for TPR = 0.958
  → Not as good as the final model (FPR = 0.048)

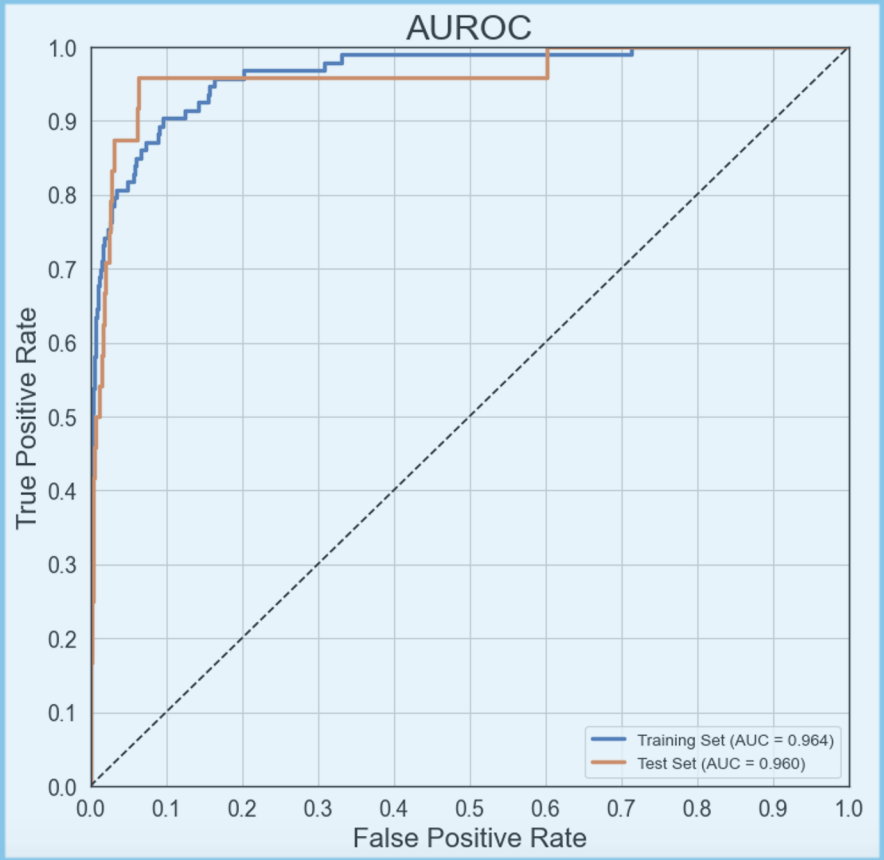


First Ensemble Model Developed

Voting Classifier

(Includes all six individual models)

- TPR = 0.958 achievable (same as final model)
- FPR = 0.063 for TPR = 0.958
  → Not as good as the final model (FPR = 0.048)

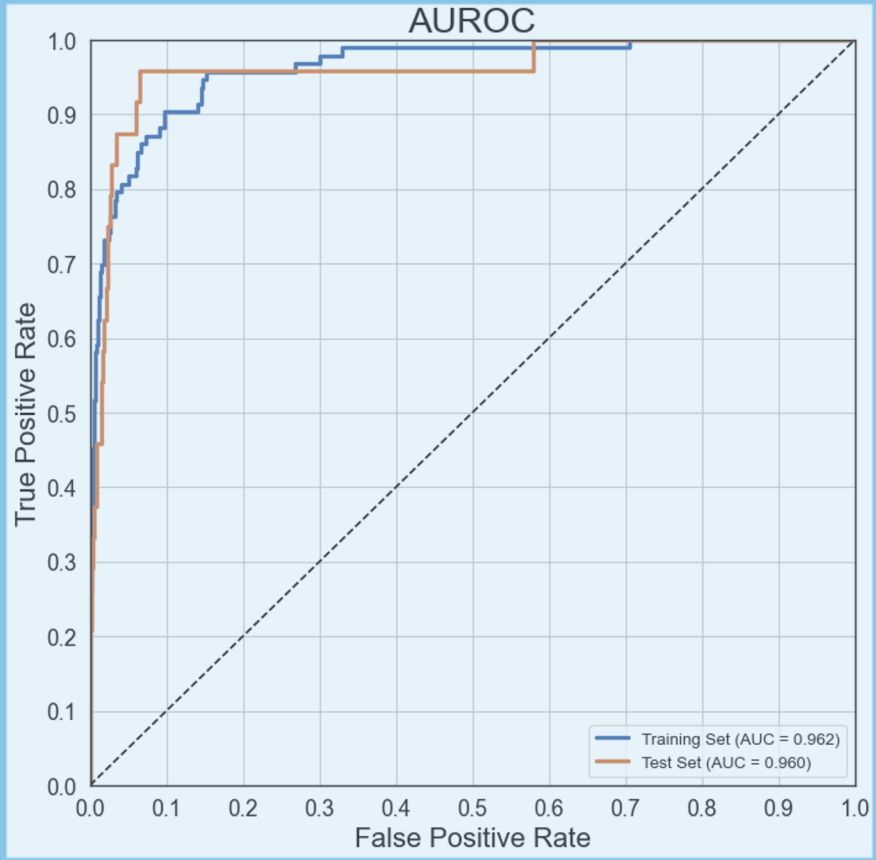


Second Ensemble Model Developed

Voting Classifier

(LR + RF + XGBoost models, optimized to AUC)

- TPR = 0.958 achievable (same as final model)
- FPR = 0.065 for TPR = 0.958
  → Not as good as the final model (FPR = 0.048)



Third Ensemble Model Developed

Voting Classifier

(LR + XGBoost models, optimized to AUC)

- TPR = 0.958
- FPR = 0.048

  → **Best Model**