



Statistical Consulting Project

**Addressing Class Imbalance in Random
Forest Classifier Used to Identify
Potential Pancreatic Cancer Cases**

Lori Kolaczkowski
Department of Statistics, Texas A&M University

Project Context

- Completed for a consulting course (Statistics, MS program)
- Client is fictional
- Students imagined client problems to address with a real analysis

Topic Selection

Elements desired:

- Has application in the oncology field
- Makes use of machine learning methods for classification learned in the previous semester
- Requires handling class imbalance

Dataset Selection

From the following study publication:

Debernardi S, O'Brien H, Algahmdi AS, Malats N, Stewart GD, Plješa-Ercegovac M, et al. (2020) A combination of urinary biomarker panel and PancRISK score for earlier detection of pancreatic cancer: A case–control study. PLoS Med 17(12): e1003489. <https://doi.org/10.1371/journal.pmed.1003489>

Focus of the above study:

Development of a non-invasive diagnostic test for more proactive pancreatic cancer detection, which uses several urinary biomarkers as predictors in classifying patients as high or low risk for the cancer.

Some Background

Pancreatic ductal adenocarcinoma (PDAC)

- The type of pancreatic cancer examined in the Debernardi, et al. study
- Most common, most aggressive, form of pancreatic cancer
- Across all cancers, one of the most lethal
- Very poor survival rate (~ 9% beyond 5 years)
- Silent killer: once symptoms of PDAC develop, it is far less treatable

Impact?

Early detection could increase 5 year survival rate to 70%.
Early detection requires frequent diagnostic screening for PDAC indicators, before symptoms develop.

Historical Biomarkers	Debernardi, et al. Biomarkers
<ul style="list-style-type: none">• Invasive collection (e.g., plasma)• Lower biomarker concentration• Infrequent screening	<ul style="list-style-type: none">• Completely non-invasive urine sample collection• Much higher biomarker concentration• Frequent screening is practical

The Data & Modifications

Debernardi, et al. Dataset	Modified Dataset
<ul style="list-style-type: none">• Response: diagnosis 1: healthy control (n=183) 2: benign pancreas condition (n=208) 3: PDAC (n=199)• Predictors: age (years) creatinine (mmol/L) LYVE1 (pg/ml) REG1B (pg/ml) TFF1 (pg/ml)	<ul style="list-style-type: none">• Response: diagnosis “no”: no PDAC (n=391) “yes”: PDAC (n=44)• Predictors: sex (M/F) age (years) creatinine (mmol/L) LYVE1 (pg/ml) REG1B (pg/ml) TFF1 (pg/ml)

Urine Panel Predictors

Why do these indicate higher risk of PDAC?

creatinine - High levels are associated with acute pancreatitis, a known risk factor for PDAC

LYVE1 - (lymphatic vessel endothelial hyaluronan receptor 1) is a protein present in lymphatic vessels when pancreatic cancer is invading

REG1B - (regenerating family member 1 beta) is a glycoprotein seen in patients with pancreatitis

TFF1 - (trefoil factor 1) is a gastrointestinal secretory peptide which is more highly expressed during development of a variety of cancer types

Setting: Client Study & Goals

The client seeks to

- develop a random forest classifier as the procedure for a simple non-invasive urine test for early detection of PDAC
- use the data from the Debernardi, et al. study to validate the model
- use predictors age, sex, creatinine, and urinary biomarkers LYVE1, REG1B, and TFF1 to classify samples as “no” = low risk, or “yes” = high risk for PDAC

Performance Requirements

True positive rate
 ≥ 0.90

False positive rate
 ≤ 0.25

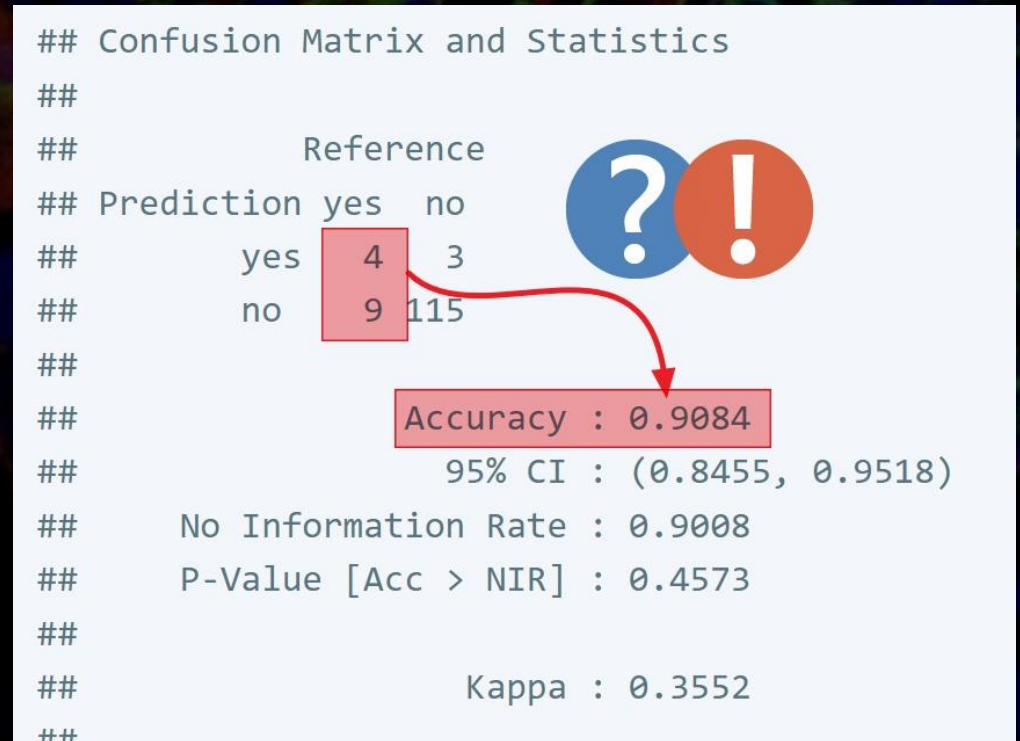
Setting: Client Problems

The test accuracy shows 0.90, yet the true positive rate is only 4/13 ...?

The classifier is classifying true cancer cases as low risk for the cancer almost 70% of the time. But our priority is detecting cancer (or risk) when it is there!

Not sure how to see which predictors are the strongest.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction yes  no
## yes         4   3
## no          9 115
##
## Accuracy : 0.9084
##          95% CI : (0.8455, 0.9518)
## No Information Rate : 0.9008
## P-Value [Acc > NIR] : 0.4573
##
##          Kappa : 0.3552
##
```



Setting: Client Questions

Q1

Why is the model output showing high accuracy, when it has such a high misclassification rate for cancer cases?

Q2

What is causing the classifier to perform poorly with respect to identifying cancer as cancer, and how do you recommend this problem be addressed in order to meet the goals of the study?

Q3

How can one get a sense of which variables are most important in predicting PDAC risk?

Q1 Q2 Q3

Accuracy Paradox

**Class B is classified
correctly 100% of the
time**

**Class A is classified
correctly 0% of the time**

Accuracy is 0.90

10%

Class A

90%

Class B

Q1 Q2 Q3

Accuracy: Not the performance metric for these goals.

```
## Accuracy : 0.9084
## 95% CI : (0.8455, 0.9518)
## No Information Rate : 0.9008
## P-Value [Acc > NIR] : 0.4573
##
## Kappa : 0.3552
##
## McNemar's Test P-Value : 0.1489
##
## Sensitivity : 0.30769
## Specificity : 0.97458
## Pos Pred Value : 0.57143
```

**True positive rate
 ≥ 0.90**

**False positive rate
 ≤ 0.25**



**Sensitivity
 ≥ 0.90**

**1 - Specificity
 ≤ 0.25**

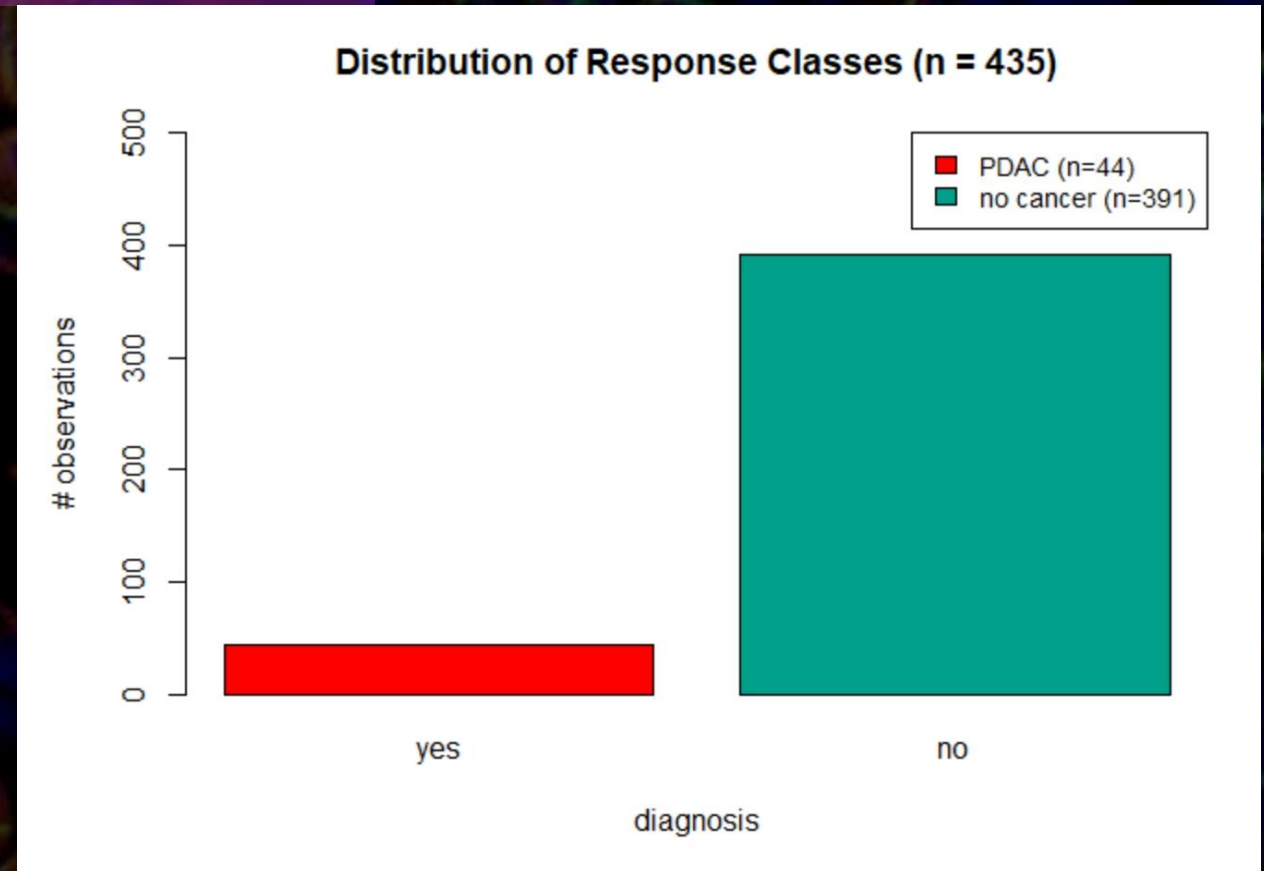
**OR Specificity
 ≥ 0.75**

Q1 Q2 Q3

The reason: class imbalance

Typical classification algorithms work to minimize error rate, without accounting for the class distribution.

They are heavily biased towards the majority class.



Q1 Q2 Q3

What to do about it?

- Collect more PDAC samples!
But this study won't allow for it, so instead try the following:
- Create train and test sets using random sampling, stratified by class
- Integrate a method for balancing the classes into the model training and validation process. Try different balancing methods and compare performances
- Try different classification algorithms appropriate for the study

Q1 Q2 Q3

Analysis: Objective

- Build a set of candidate models which include various combinations of modifications (algorithms and balancing methods) to enhance performance
- Compare them and tune probability cutoff (τ) of best performing models to reach or further optimize metrics for client performance goals

Q1 Q2 Q3

Analysis: Tools

Software:

R (version 4.0.5 for Windows)

Packages:

caret – machine learning models

DMwR – SMOTE balancing

ROSE – ROSE balancing

pROC – ROC plots

dplyr – data manipulation

ggplot2 – some plots

Q1 Q2 Q3

Analysis: Choice of algorithms

Classification

Linear
SVM

Decision trees

Non-parametric

Logistic
Regression

Kernel SVM

k-NN

Decision
trees

Naïve
Bayes

Deep neural networks

Random Forest

Simple
neural
networks

Gradient
Boosting
Machines

Kernel
regression

LASSO & Ridge Regression

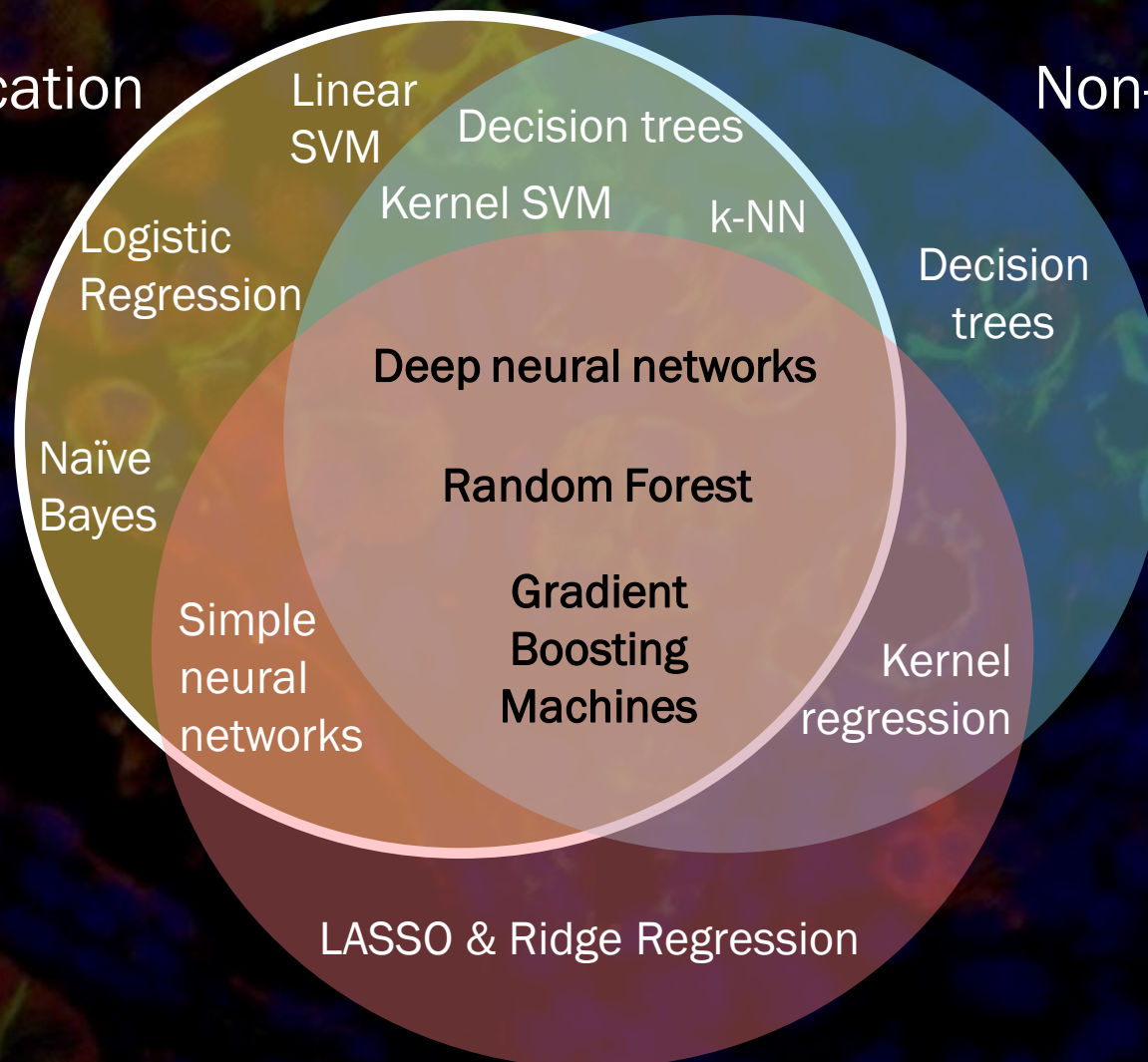
Higher Average Predictive
Performance

Q1 Q2 Q3

Analysis: Choice of algorithms

- Classification problem

Classification



Higher Average Predictive
Performance

Q1 Q2 Q3

Analysis: Choice of algorithms

- Classification problem
- Prediction focused

Classification

Linear
SVM

Decision trees

Non-parametric

Kernel SVM

k-NN

Logistic
Regression

Decision
trees

Naïve
Bayes

Deep neural networks

Random Forest

Simple
neural
networks

Gradient
Boosting
Machines

Kernel
regression

LASSO & Ridge Regression

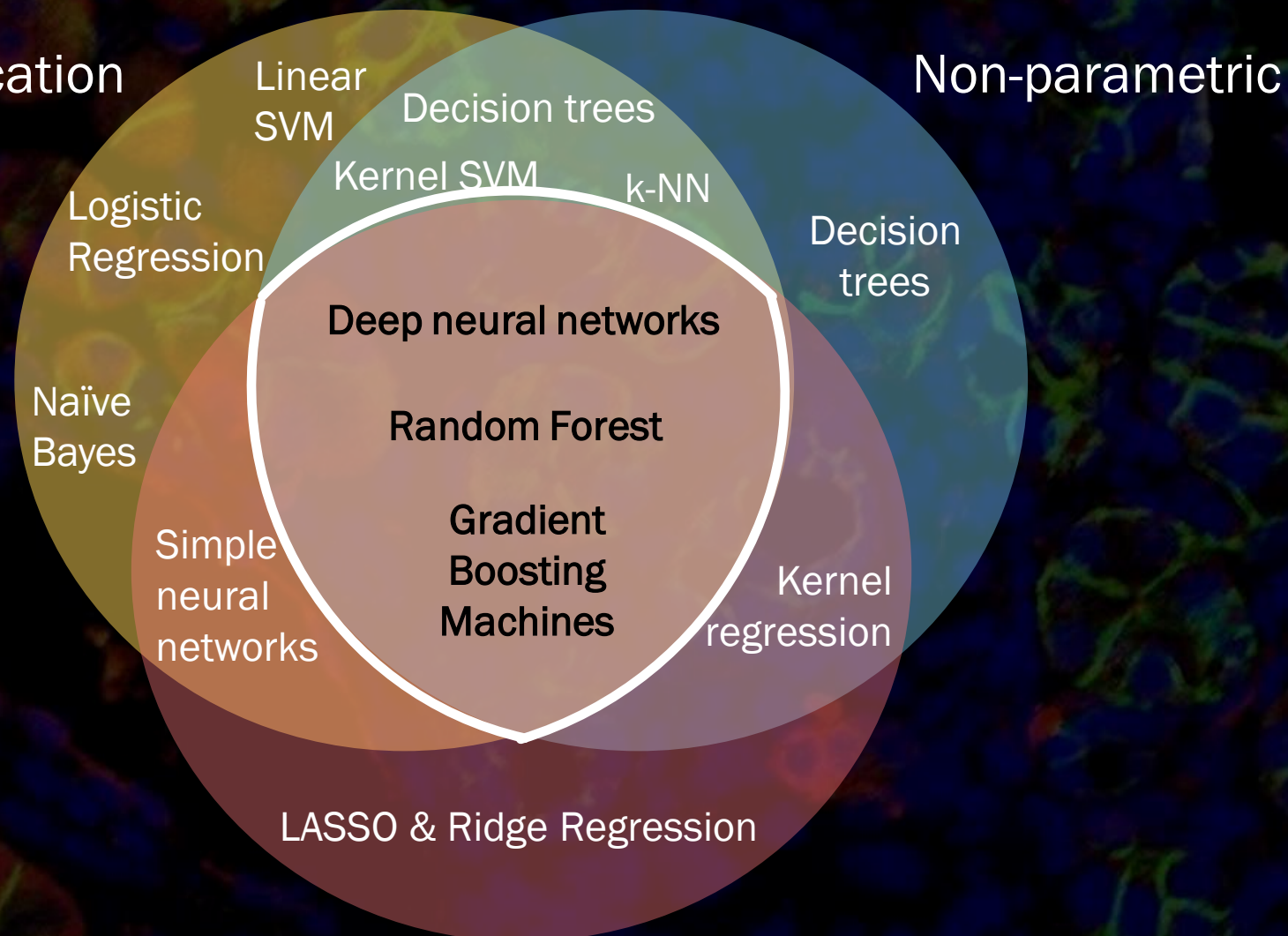
Higher Average Predictive
Performance

Q1 Q2 Q3

Analysis: Choice of algorithms

- Classification problem
- Prediction focused
- Non-parametric does not carry heavy assumptions

Classification



Higher Average Predictive
Performance

Q1 Q2 Q3

Analysis: Choice of algorithms

- Classification problem
- Prediction focused
- Non-parametric does not carry heavy assumptions
- Stick with tree-based

Classification

Linear
SVM

Decision trees

Non-parametric

Kernel SVM

k-NN

Logistic
Regression

Decision
trees

Deep neural networks

Naïve
Bayes

Random Forest

Simple
neural
networks

Gradient
Boosting
Machines

Kernel
regression

LASSO & Ridge Regression

Higher Average Predictive
Performance

Q1 Q2 Q3

Analysis: Choice of balancing methods

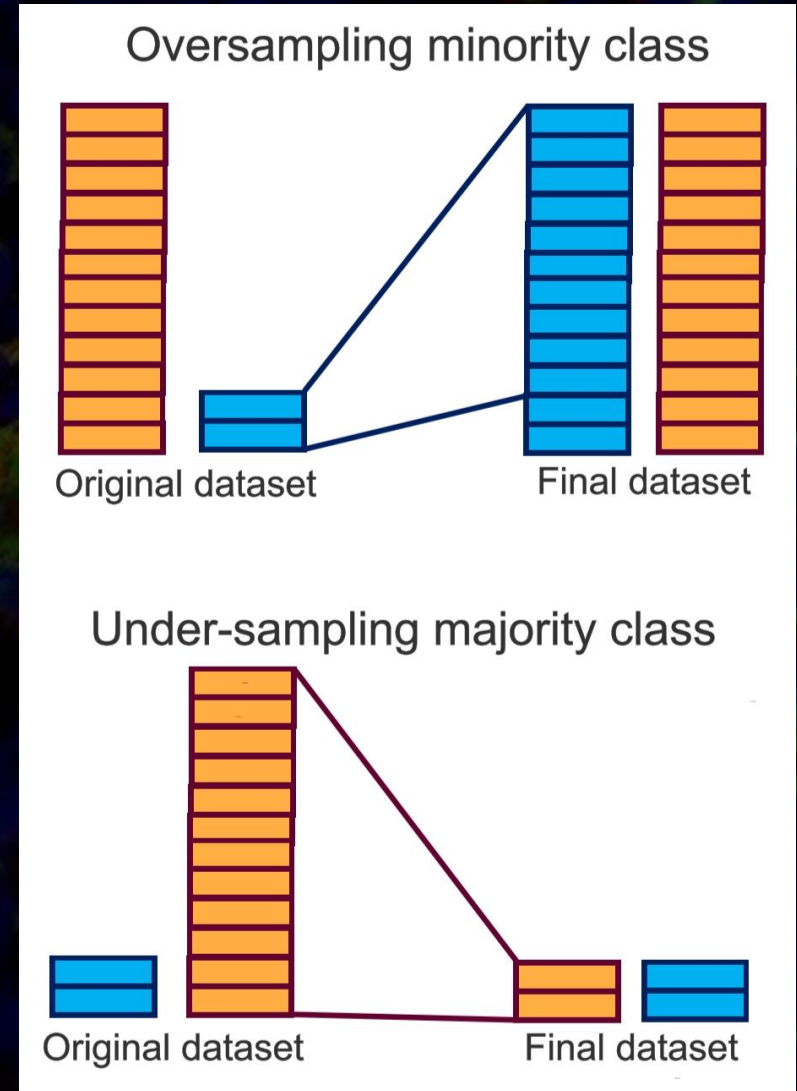
Chosen:

Under-sampling

Oversampling via resampling

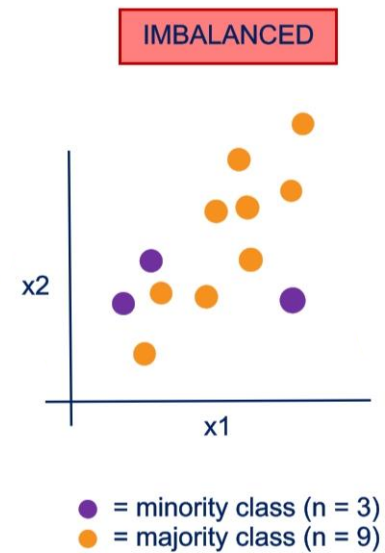
SMOTE (Synthetic Minority Oversampling TEchnique)

ROSE (Random Over-Sampling Examples)



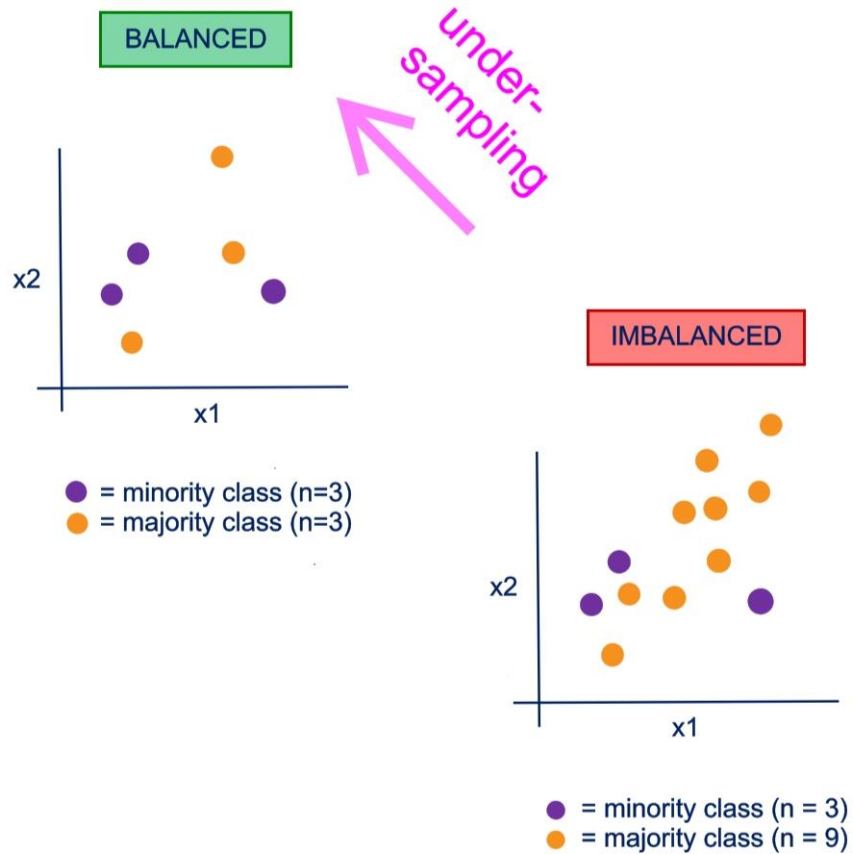
Q1 Q2 Q3

Balancing methods explained



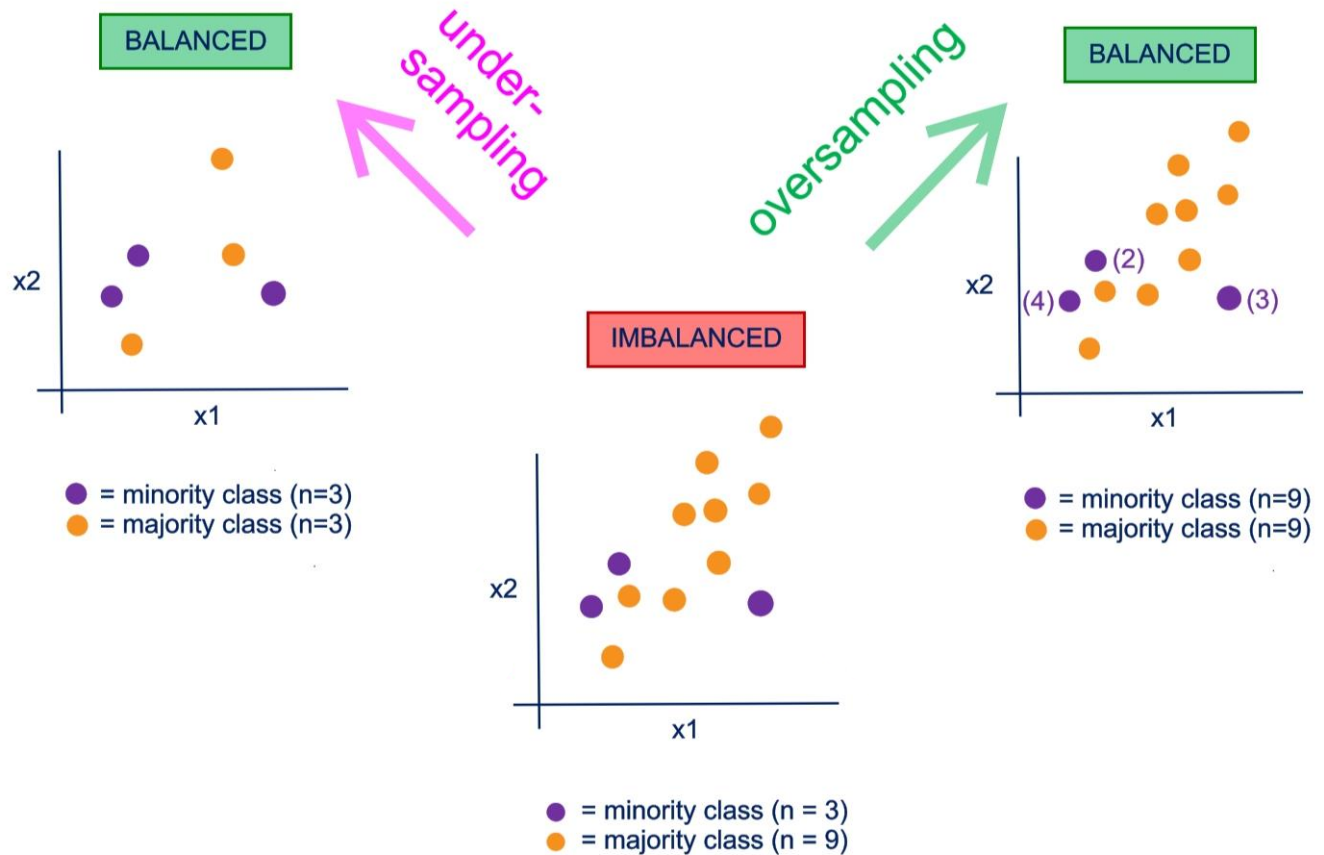
Q1 Q2 Q3

Balancing methods explained



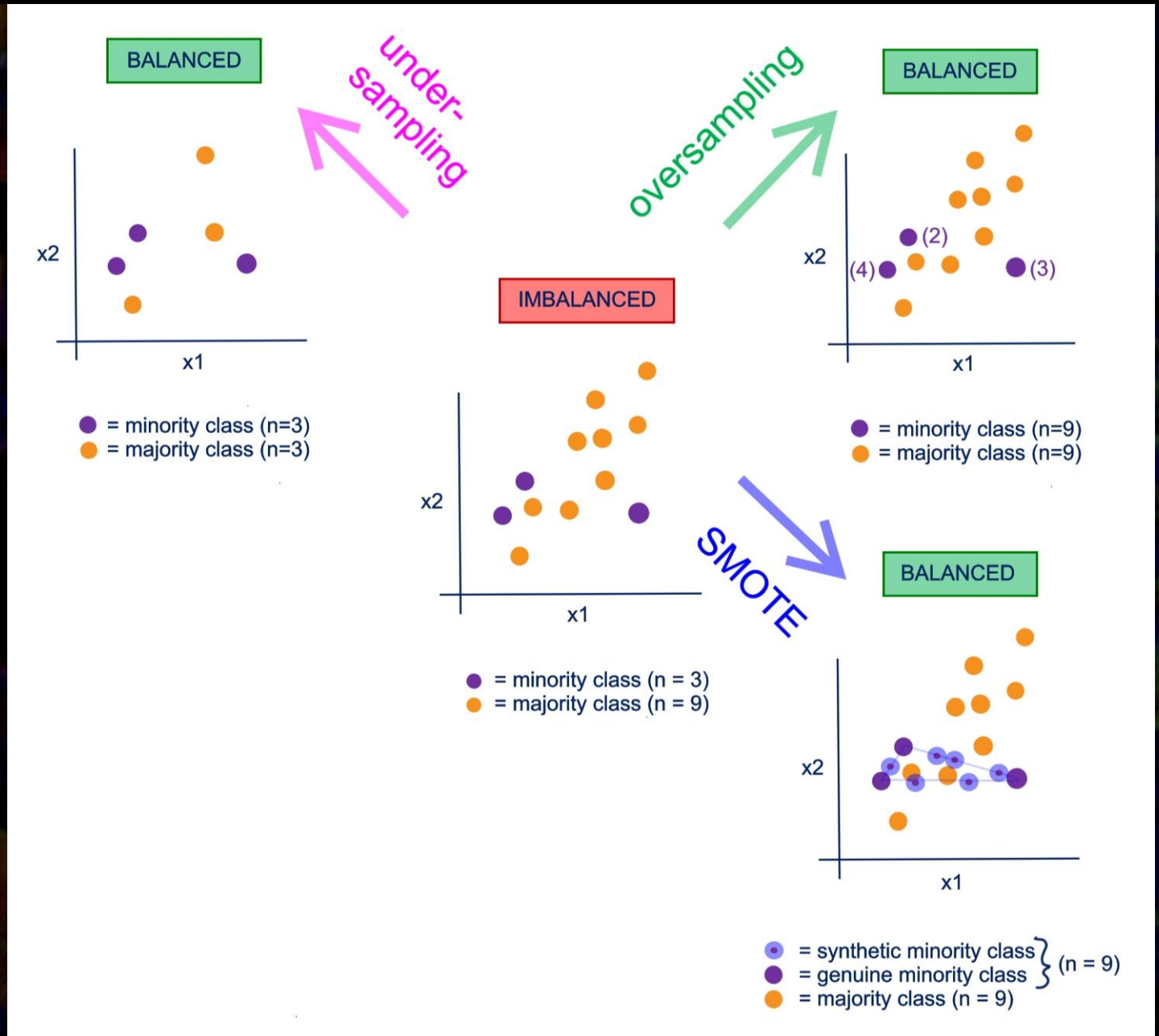
Q1 Q2 Q3

Balancing methods explained



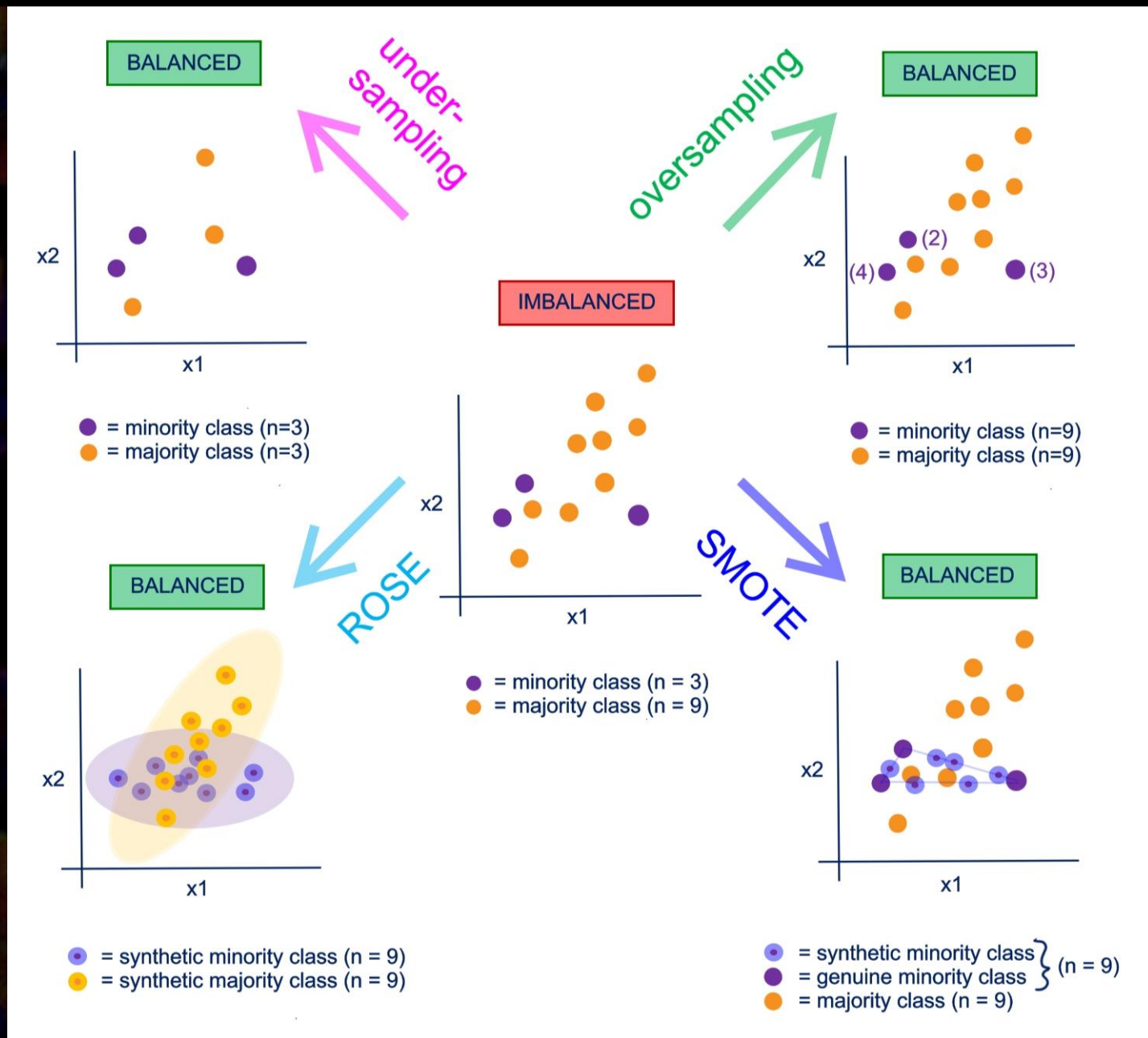
Q1 Q2 Q3

Balancing methods explained



Q1 Q2 Q3

Balancing methods explained



Q1 Q2 Q3

Analysis: Models

Performance of these 10 models were compared:

Algorithms:

RF = random forest

GBM = gradient
boosting machines

RF/imbalanced (client model)

RF/balanced (under-sampling)

RF/balanced (oversampling)

RF/balanced (SMOTE)

RF/balanced (ROSE)

GBM/imbalanced

GBM/balanced (under-sampling)

GBM/balanced (oversampling)

GBM/balanced (SMOTE)

GBM/balanced (ROSE)

Q1 Q2 Q3

Analysis: Things Kept Consistent With Client Model

- Used the caret package in R for modeling
- Set the same seed as client model, across all models, to ensure the same randomization
- Used a train/test split (70/30) of the original dataset
- Used cross-validation (CV) with 5 folds and 10 repeats
- Allowed the caret package to choose the search grid for all model hyperparameters in CV
- Specified “ROC” as the metric for caret’s cross-validation model selection (Uses AUC)
- Used a probability threshold (τ) = 0.5, for initial model comparison

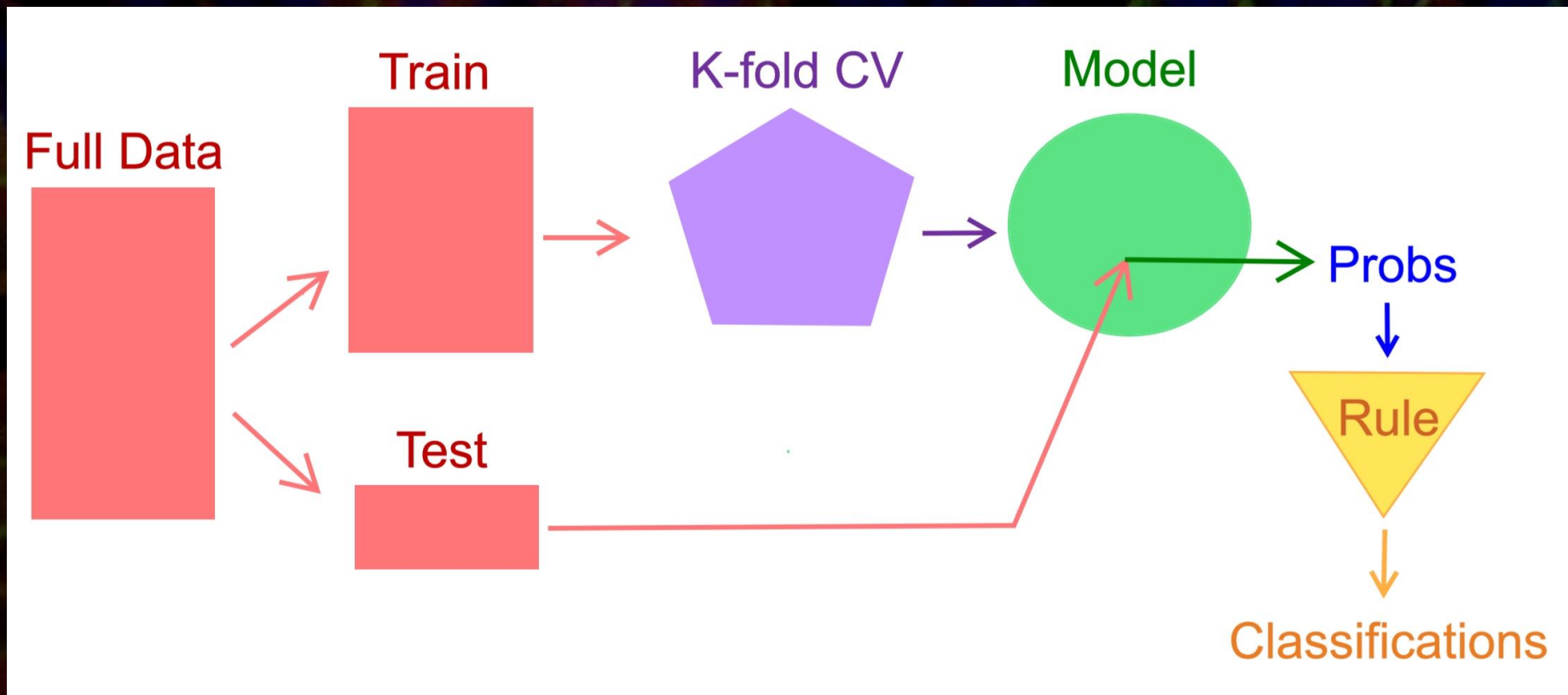
Next slides:

Modifications to data splitting and cross-validation, and where exactly balancing occurs in the modeling process

Q1 Q2 Q3

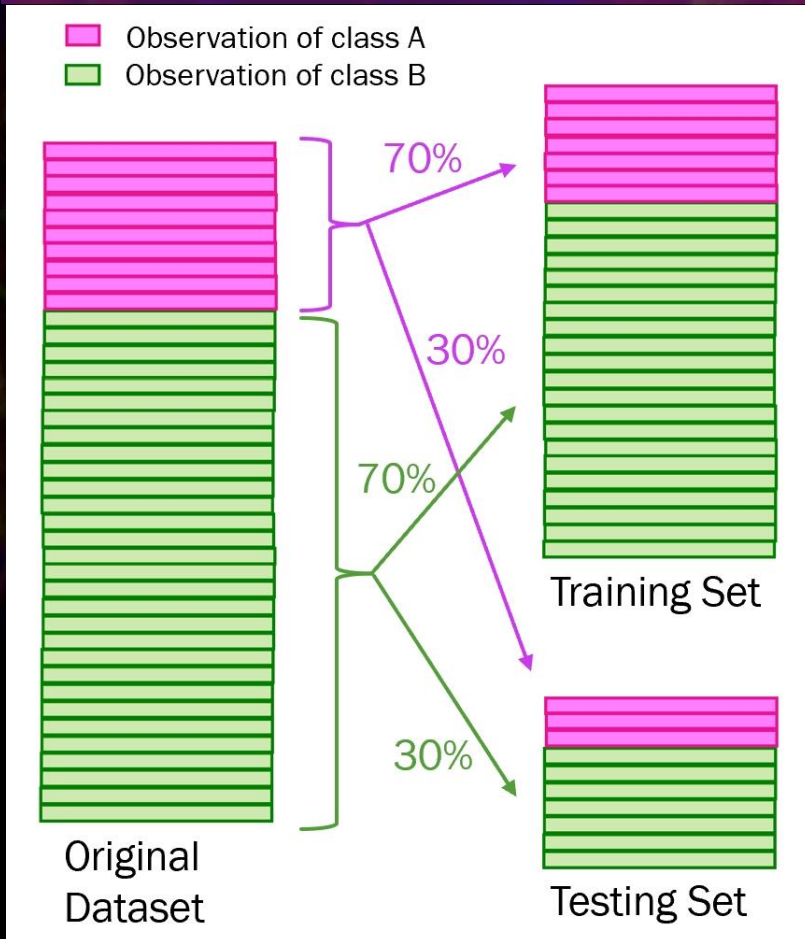
Analysis: Modeling Overview

This entire process is carried out for each of the 10 models previously mentioned, using the same train & test set.



Q1 Q2 Q3

Analysis: Train/Test Split

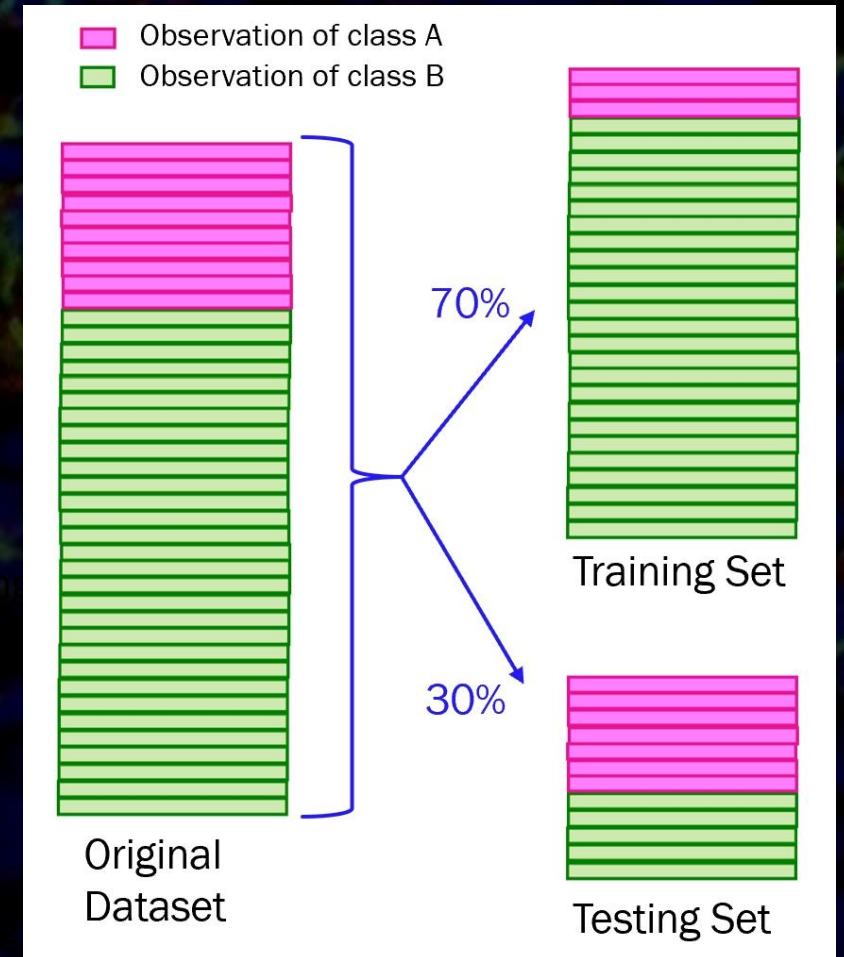


Observation of class A
Observation of class B

Stratified

vs.

Unstratified

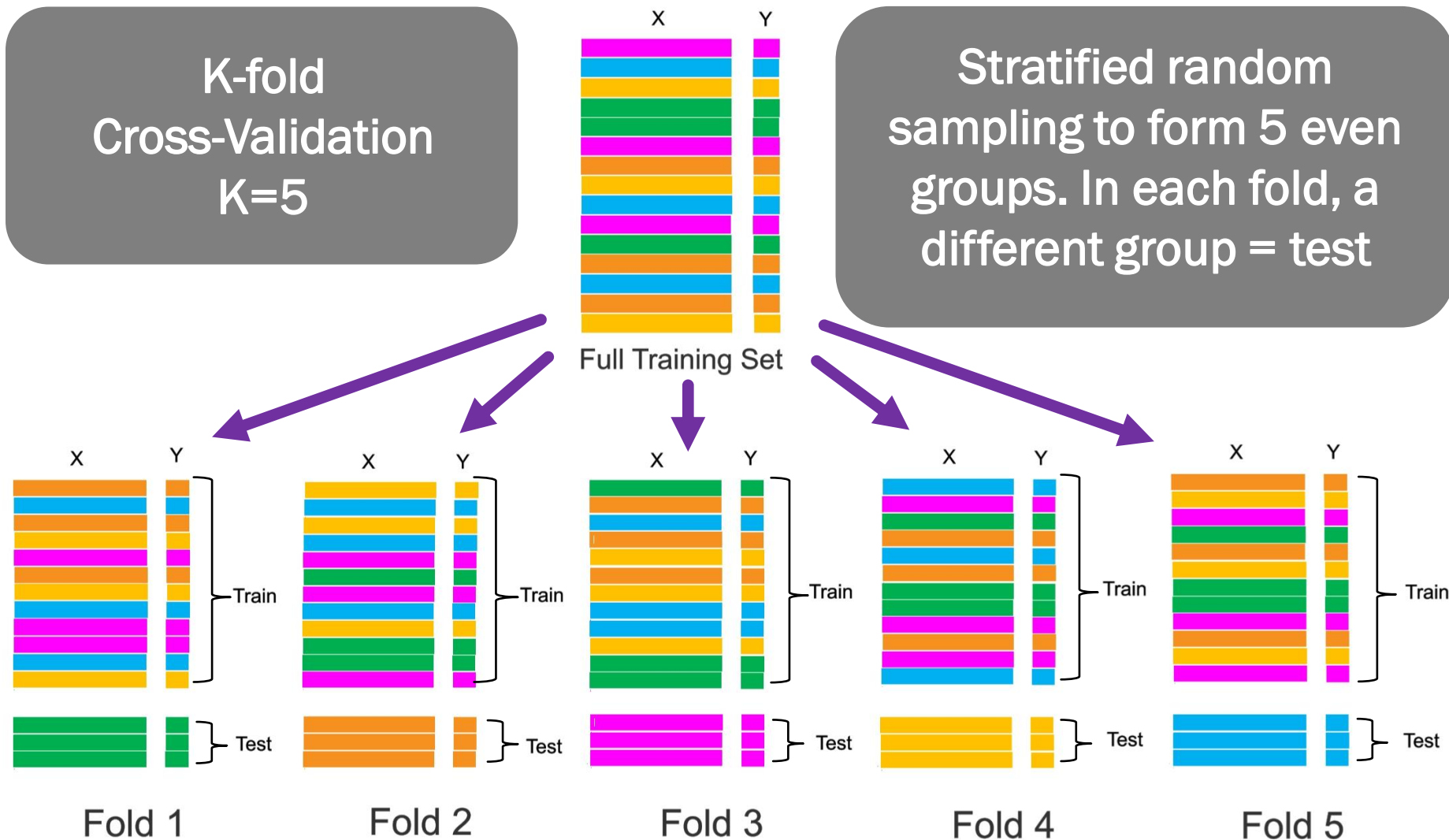


Q1 Q2 Q3

Analysis: Model Validation

K-fold
Cross-Validation
 $K=5$

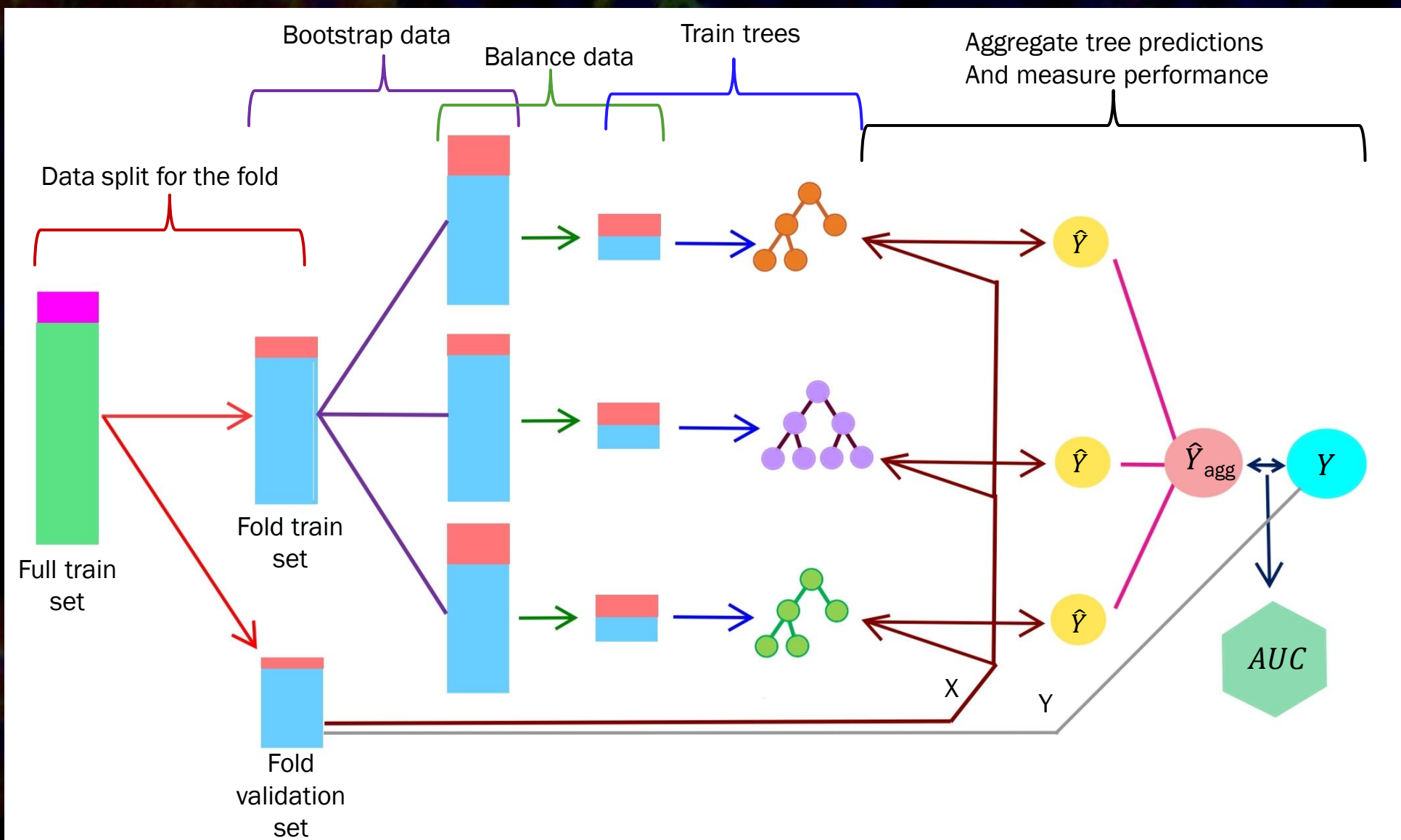
Stratified random
sampling to form 5 even
groups. In each fold, a
different group = test



Q1 Q2 Q3

Analysis: Model Validation

Process
within each
of the 5 folds
for cross-
validation



Analysis: Model Testing

After Stratified K-fold cross-validation chooses the parameters which yield the most robust model:

- 1) Test dataset (just X) is fed to this model and probabilities are generated:

The probability that each observation belongs to the “no” (low risk) class.

- 2) These probabilities are converted to class predictions using this rule:

If probability $> \tau \rightarrow$ class prediction = “no”
(otherwise, class prediction = “yes”)
Initially, $\tau = 0.5$

- 3) The predicted classes are then compared to the true classes (Y of test set), and the output gives:

The true positive rate (sensitivity)
The true negative rate (specificity)

(1 – specificity gives the false positive rate)

Q1 Q2 Q3

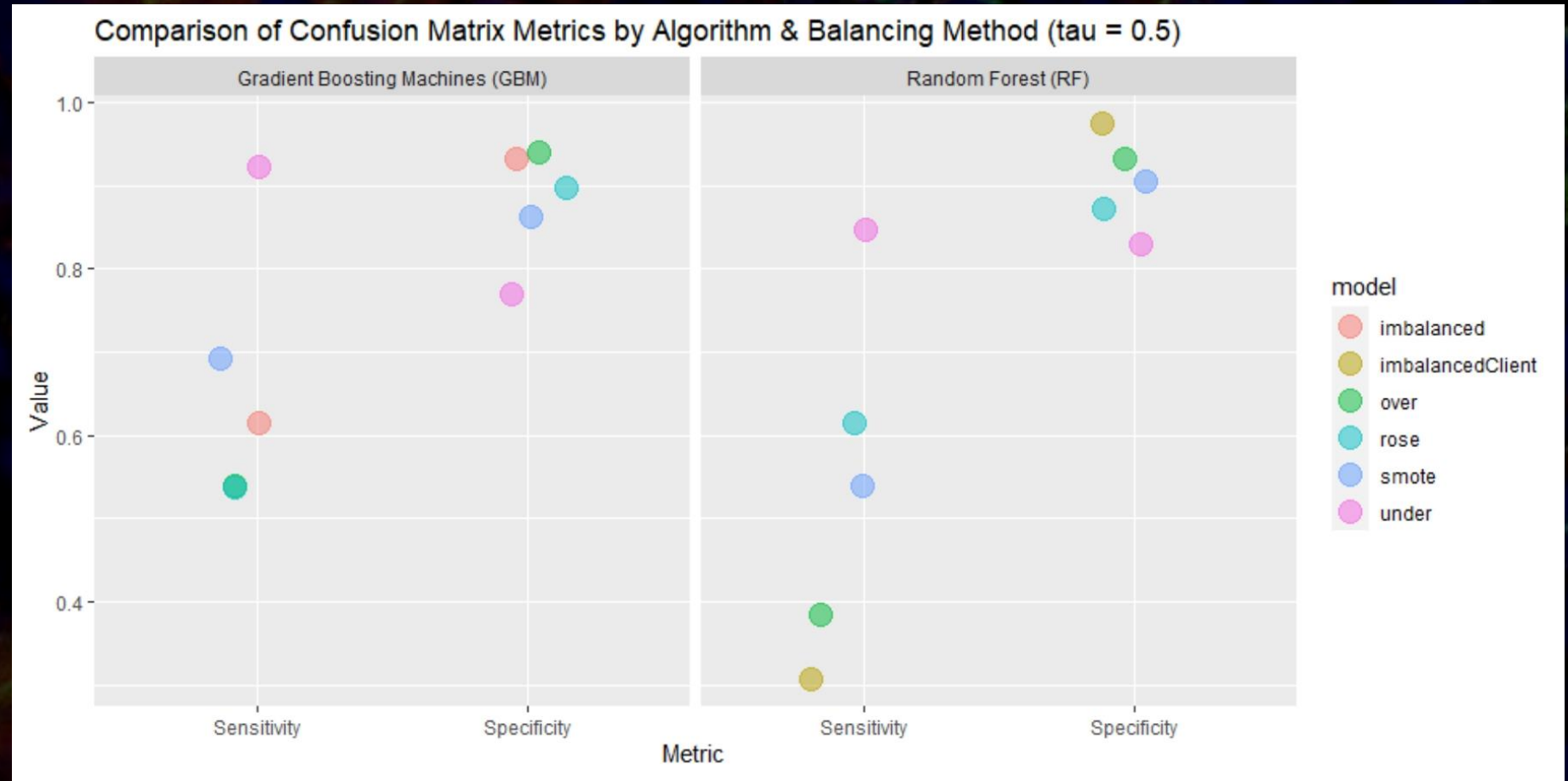
Analysis: Results

Goals: **Sensitivity**
 ≥ 0.90 **Specificity**
 ≥ 0.75

GBM/under
appears to
make both
goals.

RF/under might
with tuning.

Let's tune both.



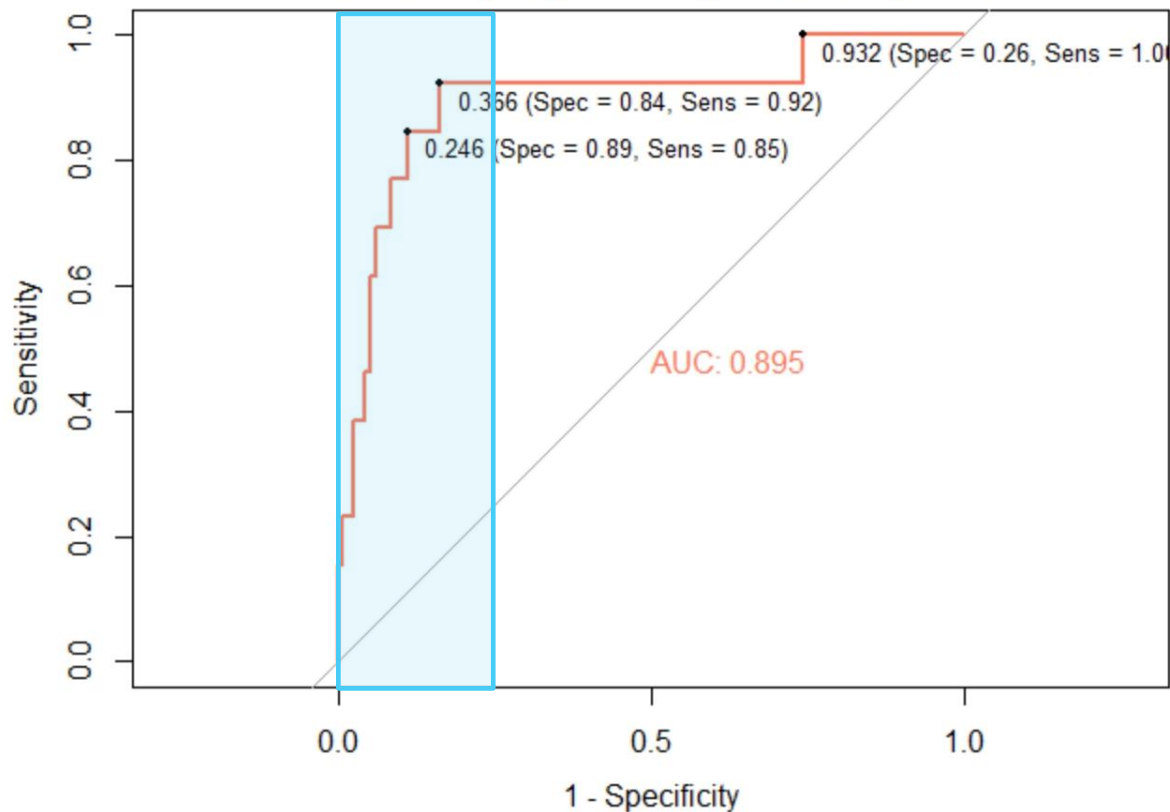
Q1 Q2 Q3

Analysis: Results

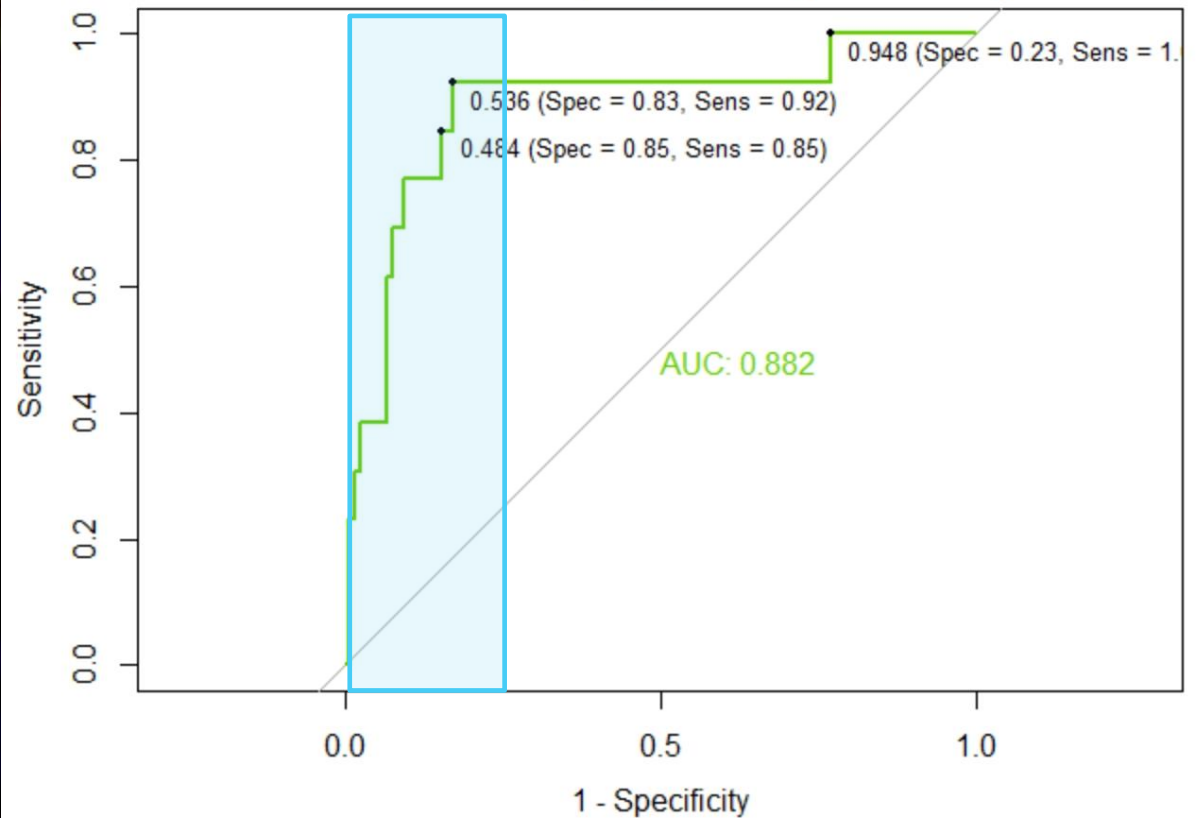
Tuning: How do they compare
at various thresholds (τ)?

Goal range for $1 - \text{specificity}$ (≤ 0.25)

GBM/under-sampling: Tuning Probability Threshold



Random Forest/under-sampling: Tuning Probability Threshold



Q1 Q2 Q3

Analysis: Results

RF/under:

Optimal at $\tau = 0.536$

Sensitivity: 0.923

1 - specificity: 0.171

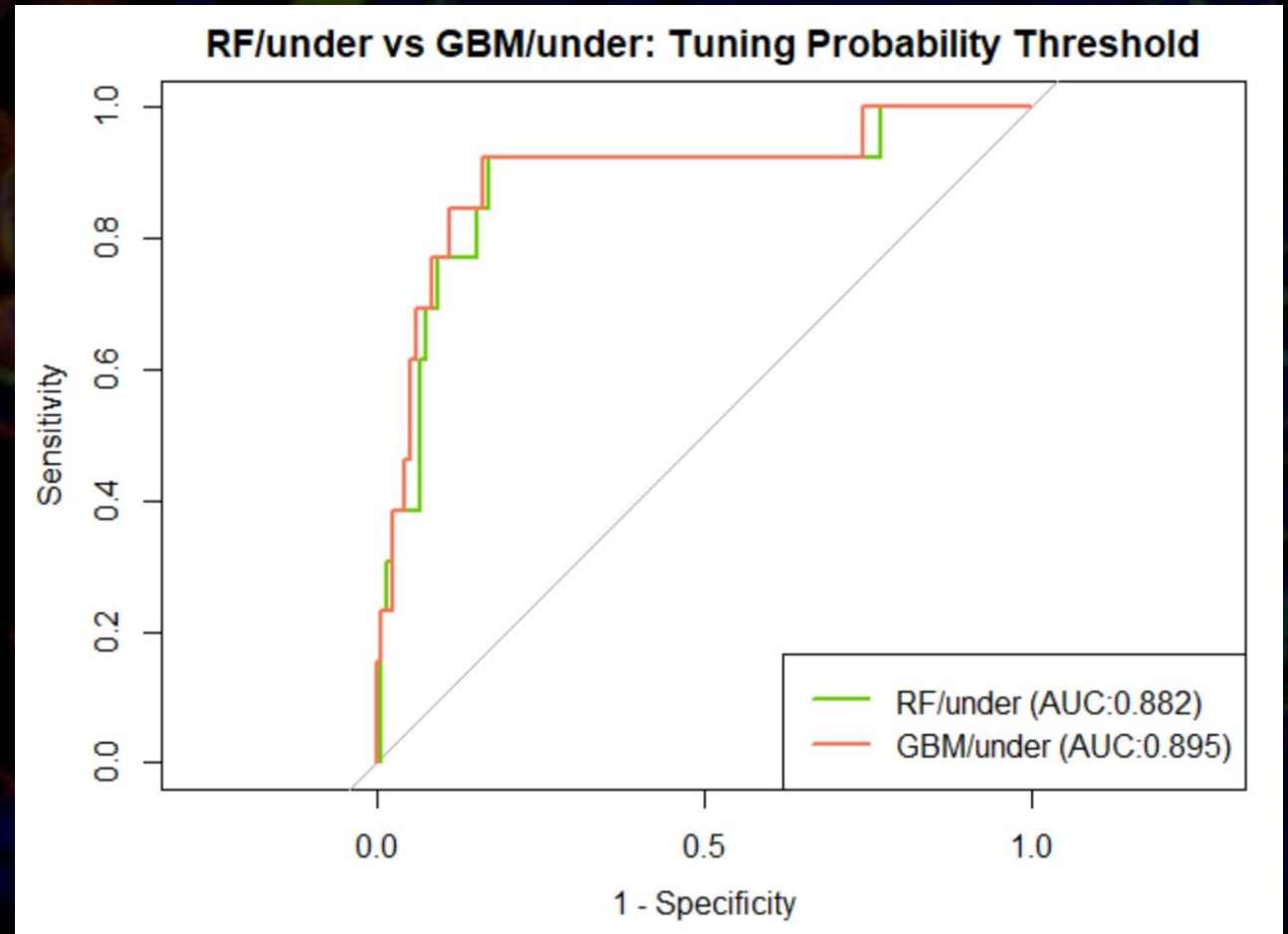
GBM/under:

Optimal at $\tau = 0.366$

Sensitivity: 0.923

1 - specificity: 0.162

GBM/under has slightly better performance.



Q1 Q2 Q3

Recommendation

Procedure:

GBM algorithm

(n.trees = 50, interaction.depth = 3, shrinkage = 0.1,
n.minobsinnode = 10)

**using stratified data splitting and
under-sampling with classification
cutoff**

$\tau = 0.366$

Performance:

**Sensitivity
(true positive rate)
= 0.923**

**1 – Specificity
(false positive rate)
= 0.162**

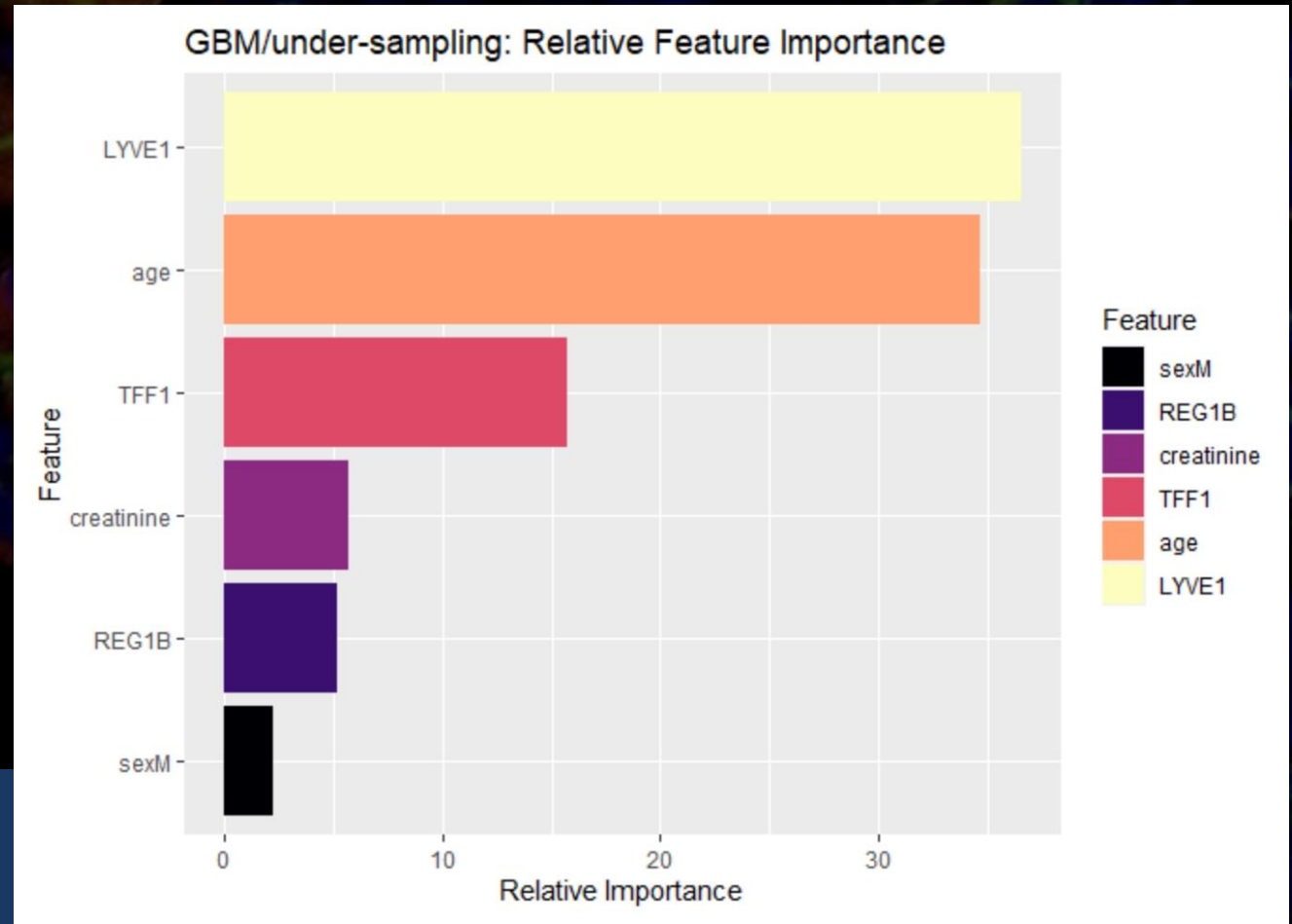
Q1 Q2 Q3

Variable Importance

GBM: The caret package in R can output the strength of each variable in predicting the response by inserting your model into the function `summary()`.

R Code for the plot:

```
# GBM/Under
RI <- summary(modelgbm_us, plot = FALSE)
RI <- transform(RI, var = reorder(var, rel.inf))
ggplot2::ggplot(RI, aes(rel.inf, var, fill = var)) +
  geom_col(aes()) +
  labs(title="GBM/under-sampling: Relative Feature Importance") +
  labs(x="Relative Importance", y="Feature") +
  scale_fill_viridis_d(option="magma", name = "Feature")
```



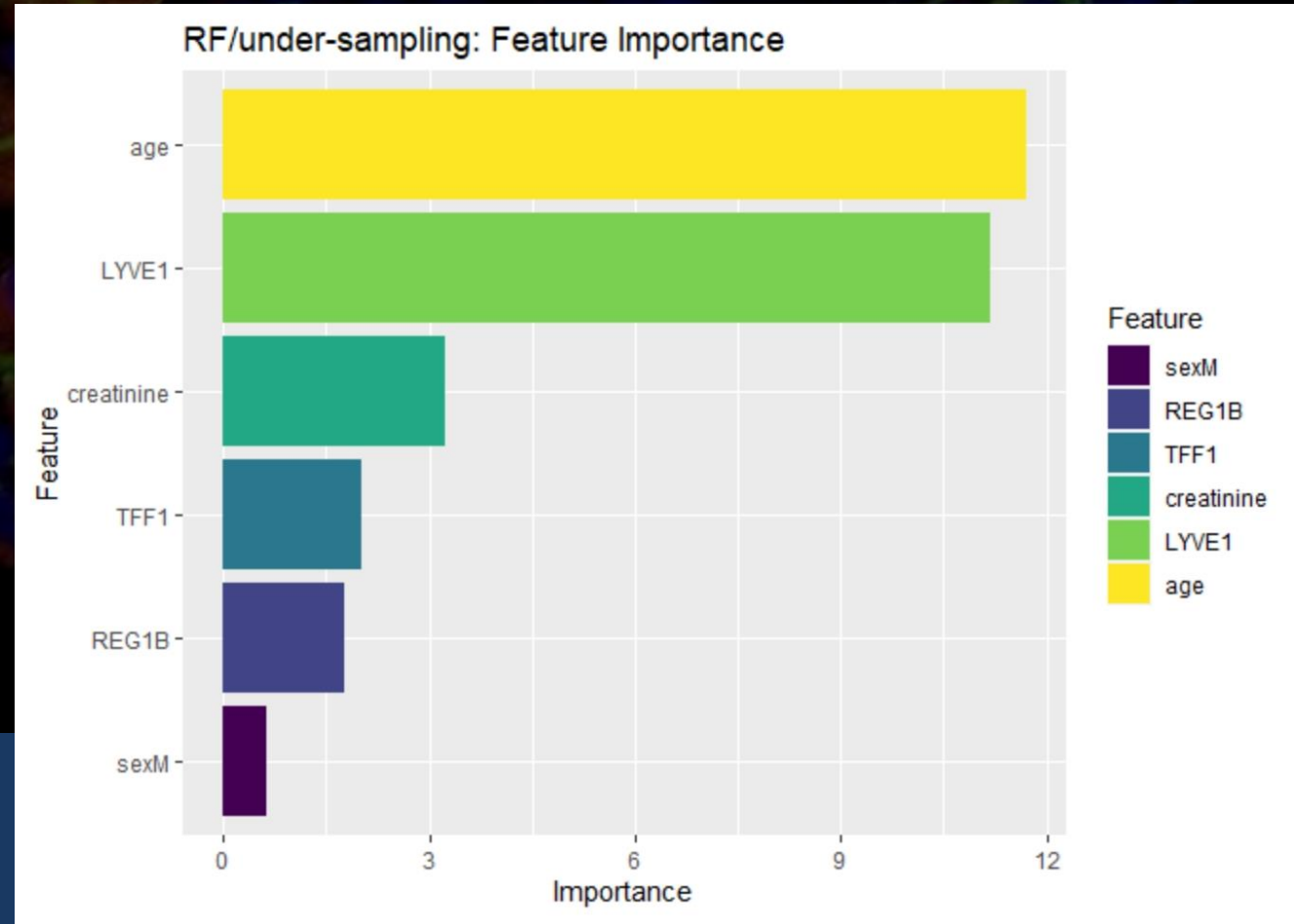
Q1 Q2 Q3

Variable Importance

Random Forest: The caret package in R can output the strength of each variable in predicting the response by inserting your model into the function varImp().

R Code for the plot:

```
# RF/Under
VI <- varImp(modelrf_us, scale=FALSE)
VI <- VI$importance
VI$feat <- row.names(VI)
VI <- transform(VI, feat = reorder(feat, overall))
ggplot2::ggplot(VI, aes(overall, feat, fill = feat)) +
  geom_col(aes()) +
  labs(title="RF/under-sampling: Feature Importance") +
  labs(x="Importance",y="Feature") +
  scale_fill_viridis_d(option="viridis", name = "Feature")
```



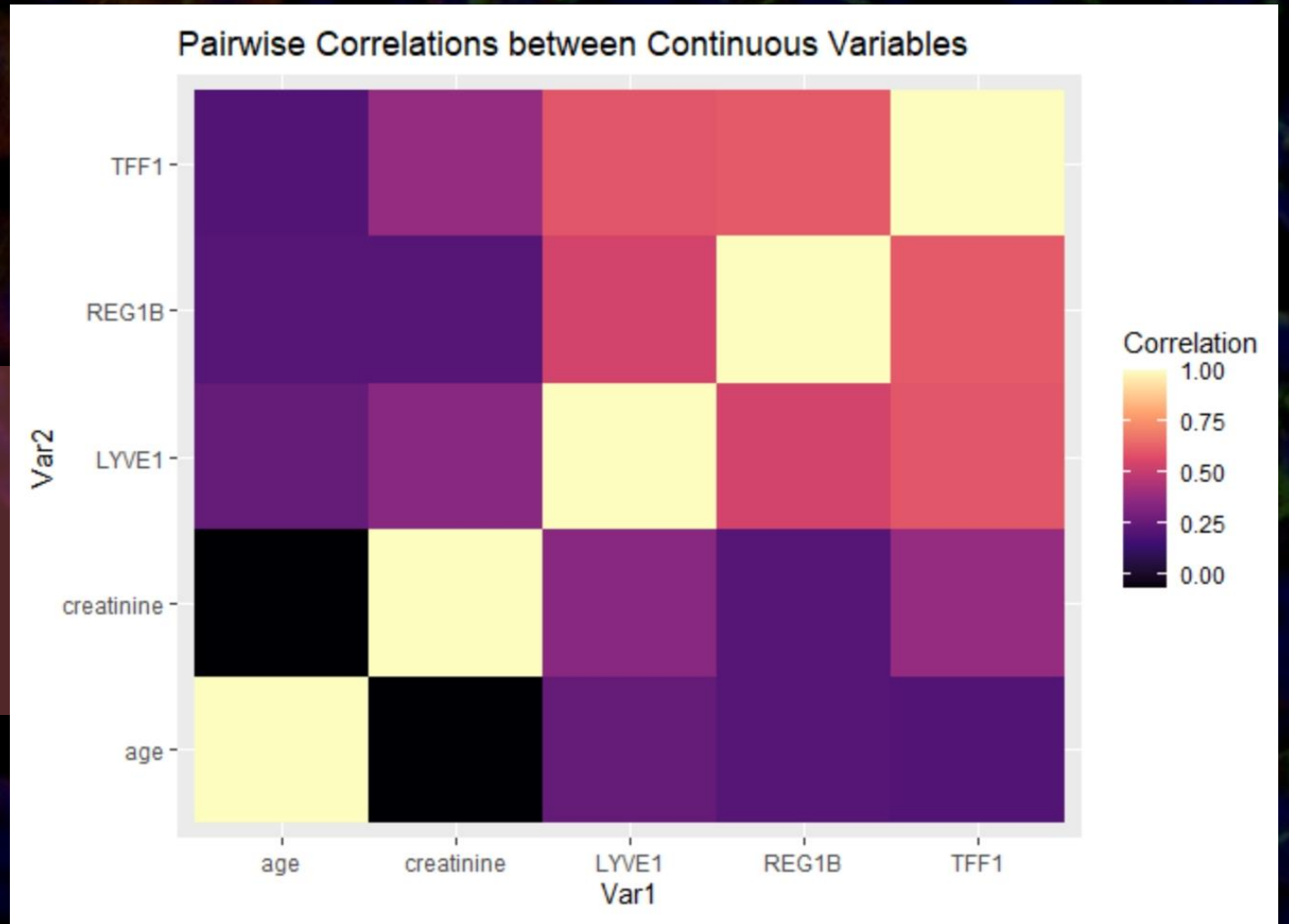
Q1 Q2 Q3

Variable Importance

Warning:

Can be misleading if correlation between any 2 variables exceeds about 0.7

Highest correlation between any two predictors is 0.61, so we can trust the variable importance plots.



Takeaways for the Client

- Q1** Performance discrepancy: accuracy paradox
Don't look at accuracy. Find the metrics for your goals.
True positive rate = Sensitivity False positive rate = 1 - Specificity
- Q2** Poor classification cause: class imbalance
This recommended procedure corrects for this and meets study goals:
GBM algorithm using under-sampling with probability cutoff $\tau = 0.366$
- Q3** Use `varImp()` or `summary()` on your model and plot with `ggplot2` in R to see variable importance.

Conclusion

It can be done! Even with imbalanced data, this less invasive screening test for PDAC risk can perform as needed, given the right procedure.

It is clinically practical enough to be put to use in the real world, increasing frequency of screening and increasing survival rates of those with PDAC.

Sources

1. Debernardi S, O'Brien H, Algahmdi AS, Malats N, Stewart GD, Plješa-Ercegovac M, et al. (2020) A combination of urinary biomarker panel and PancRISK score for earlier detection of pancreatic cancer: A case–control study. PLoS Med 17(12): e1003489. <https://doi.org/10.1371/journal.pmed.1003489>
2. Data download: <https://www.kaggle.com/johnjdavisiv/urinary-biomarkers-for-pancreatic-cancer/version/1>
3. Nitesh V. Chawla (2005) Data Mining for Imbalanced Datasets: An Overview. Data Mining and Knowledge Discovery Handbook, pp. 853–867
4. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall and W. Philip Kegelmeyer (2002) SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, Vol. 16, pp. 321–357
5. Nicola Lunardon, Giovanna Menardi and Nicola Torelli (2014) ROSE: A Package for Binary Imbalanced Learning. R Journal, Vol. 6 Issue 1, pp. 79–89
6. <https://towardsdatascience.com/accuracy-paradox-897a69e2dd9b>
7. Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, pp. 1137–1143.
8. Slide background image: credit to Michael Feigin, Tuveson Lab, CSHL

A fluorescence microscopy image showing a dense population of cells. The cells are stained with three different dyes: a blue dye (likely DAPI) that highlights the nuclei, a red dye that stains the cytoplasm or certain organelles, and a green dye that appears to outline the cell membranes or specific internal structures. The overall image has a dark background, making the brightly stained cells stand out. A semi-transparent purple rectangular box is centered over the image, containing the text 'Thank you!'.

Thank you!