Welcome to
Lab2: Evaluation

Date: 11.02.2026

Who I am: Ceren Gok - PhD student working on Continual Learning and TA for MLE course 2026!

What is these Labs are about: Guided-coding exercises, Try to remember what we learned from the lectures, Sharing our opinions with our peers, aaannd competing with fun way !

The Structure of Slides: Historical Facts, Fun Facts, Revisiting Theories,
Interview Questions, Live Coding Exercise for you.

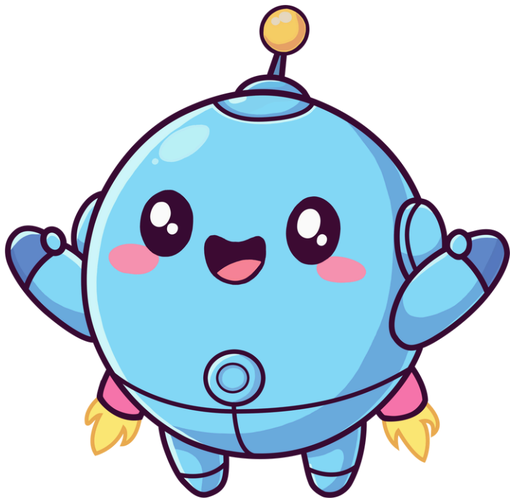
Welcome to
Lab2: Evaluation

Intended Learning Outcomes:

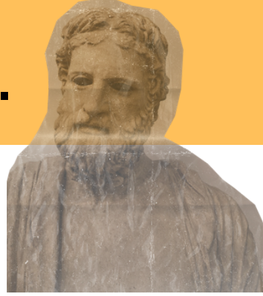
- Remember model evaluation techniques
- Implement and evaluate a classification problem
- Understand different evaluation metrics
- Learn to calibrate your model with cost function

Welcome to
Lab2: Evaluation

Lets start with a Game and see how much we remember from lecture notes!



Historical Fact: Evaluation isn't academic. It's about real consequences.



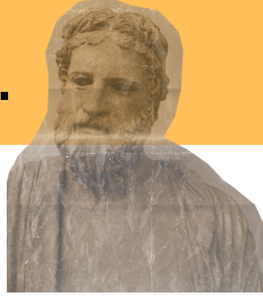
Apple Watch (High Recall Low Precision)

In 2018, Apple added a feature to the Apple Watch that warns users if it detects an abnormal heart rhythm.

The watch sent a lot of alerts, but most of them turned out to be **false alarms**.

Only about **11 out of 100** people who went to the doctor because of an alert actually had a real heart condition.The other **89** people were fine.

Historical Fact: Evaluation isn't academic. It's about real consequences.



Apple Watch (High Recall Low Precision)

In 2018, Apple added a feature to the Apple Watch that warns users if it detects an abnormal heart rhythm.

The watch sent a lot of alerts, but most of them turned out to be **false alarms**.

Only about **11 out of 100** people who went to the doctor because of an alert actually had a real heart condition.The other **89** people were fine.

Recall vs Precision Tradeoff

1.Catching everything (Recall / Sensitivity)

The watch was very good at not missing people who truly had a heart problem. This is important for safety.

2.Too many false alarms (Low Precision)

Because it was so cautious, it also warned many healthy people.

Today's Mission: Build an AI Assistant for Breast Cancer Diagnosis

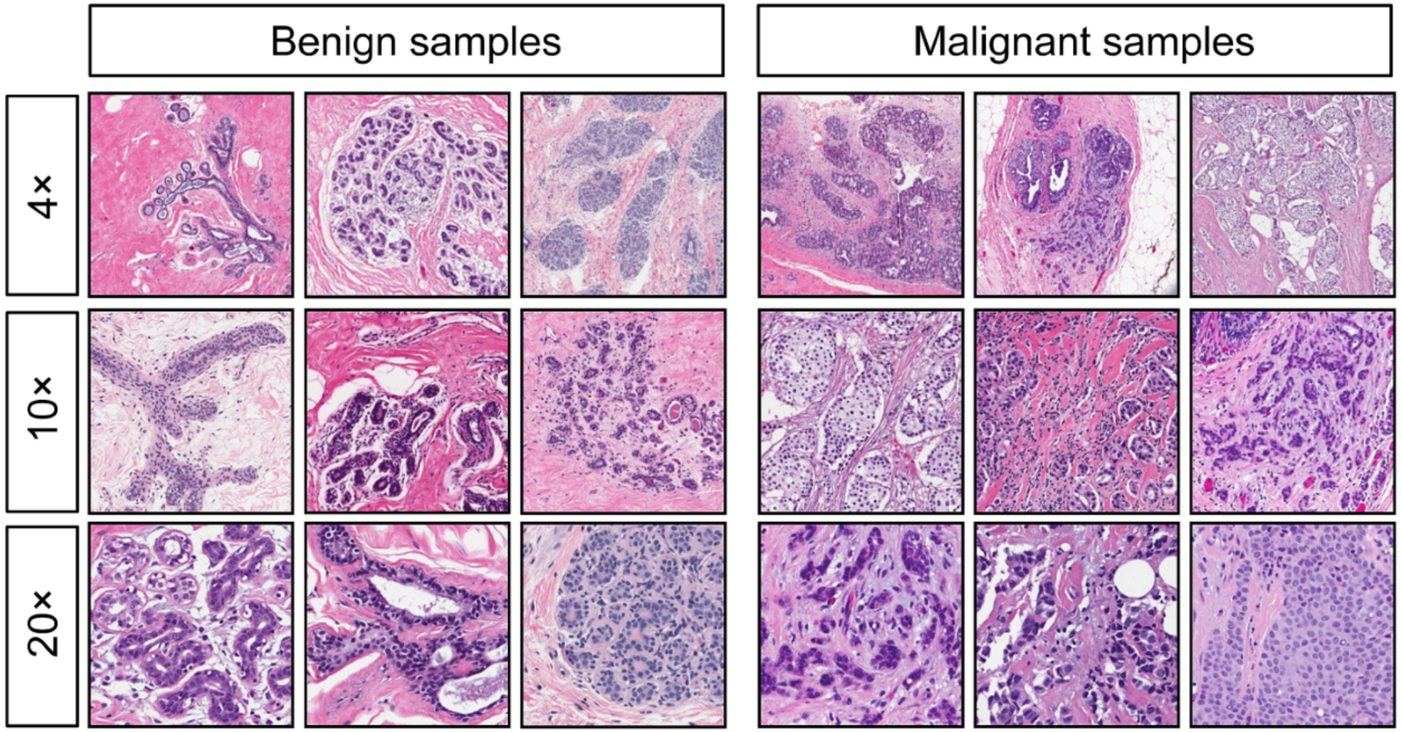
You are in the interview process as data scientist role at St. Catherina Hospital

The hospital board has questions:

- ❓ Which model should we deploy?
- ❓ How do we know your model works?
- ❓ What mistakes will it make?
- ❓ Can we trust it with patients' lives?

Your job: Answer these questions with DATA.

Know Your Data: Wisconsin Breast Cancer Dataset



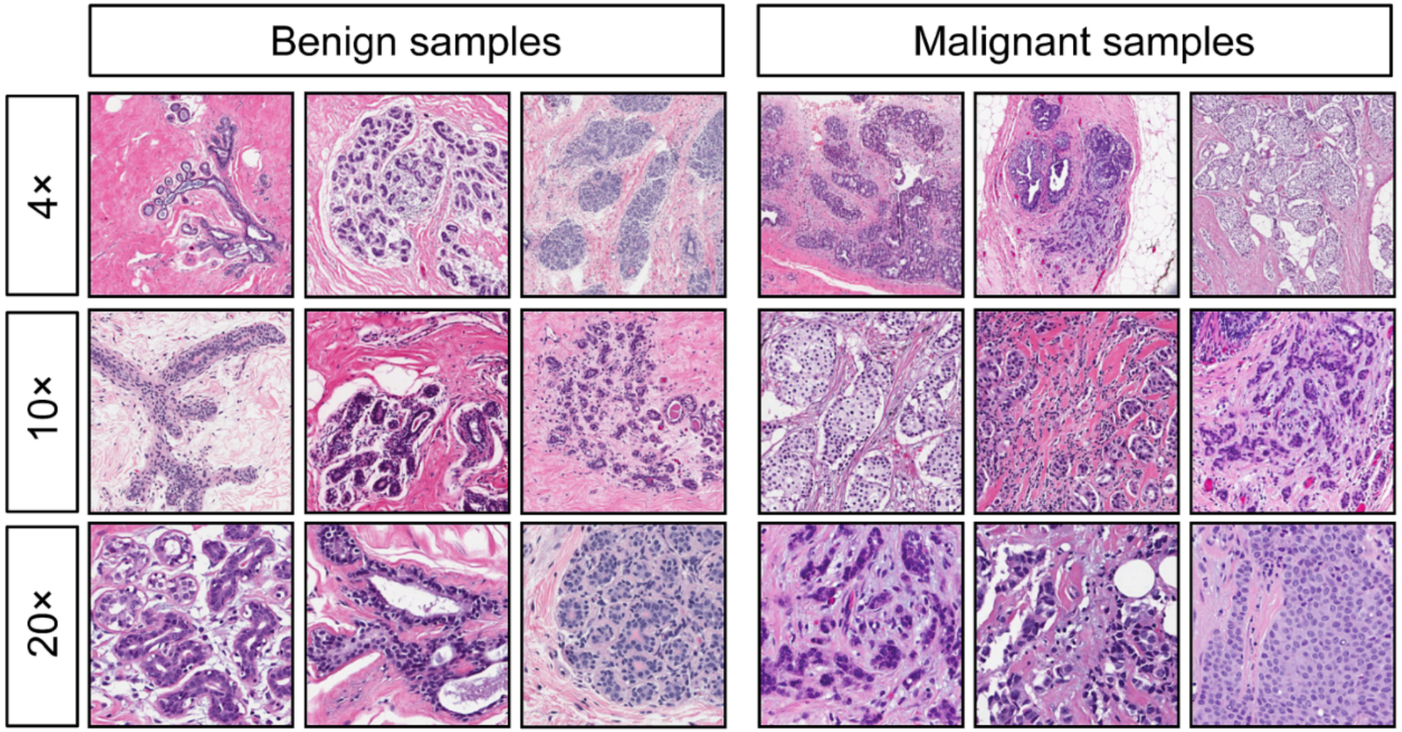
Who: University of Wisconsin Hospital

- Dr. William Wolberg, (MD)
- Prof. Olvi Mangasarian (computer scientist)

Why: Improve the accuracy and speed of cancer diagnosis

What: 569 patient samples
30 features
2 classes (malignant, benign)

Know Your Data: Wisconsin Breast Cancer Dataset



Who: University of Wisconsin Hospital

- Dr. William Wolberg, (MD)
- Prof. Olvi Mangasarian (computer scientist)

Why: Improve the accuracy and speed of cancer diagnosis

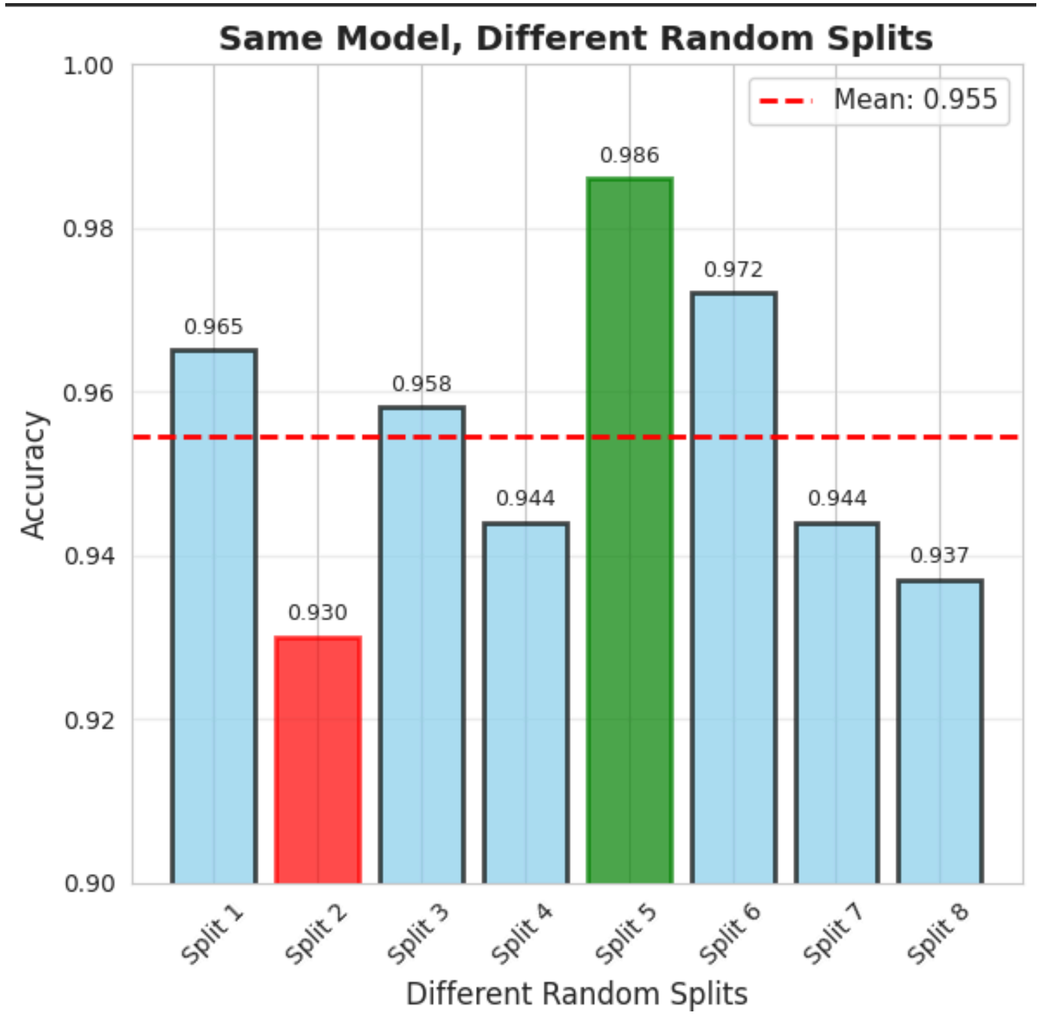
What: 569 patient samples
30 features
2 classes (malignant, benign)

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...

Actual data is in tabular form.

Interview Moment: "Lucky Split" Problem

You tried out different random seeds to split the data and plot the accuracy result:



Which result would you show to the board?

Interview Moment: "Lucky Split" Problem

I am aware that I **can't** just **pick** the **best random seed**.

That's cherry-picking.

The accuracy varies from 93% to 98% which shows the **model** is **unstable**.

I should use **cross-validation** instead to get a more **reliable** score.

Good Approach

Interview Moment: "Lucky Split" Problem

I am aware that I **can't** just **pick** the **best random seed**.

That's cherry-picking.

The accuracy varies from 93% to 98% which shows the **model** is **unstable**.

I should use **cross-validation** instead to get a more **reliable** score.

Good Approach

What I am seeing here is called '**selection bias**'.

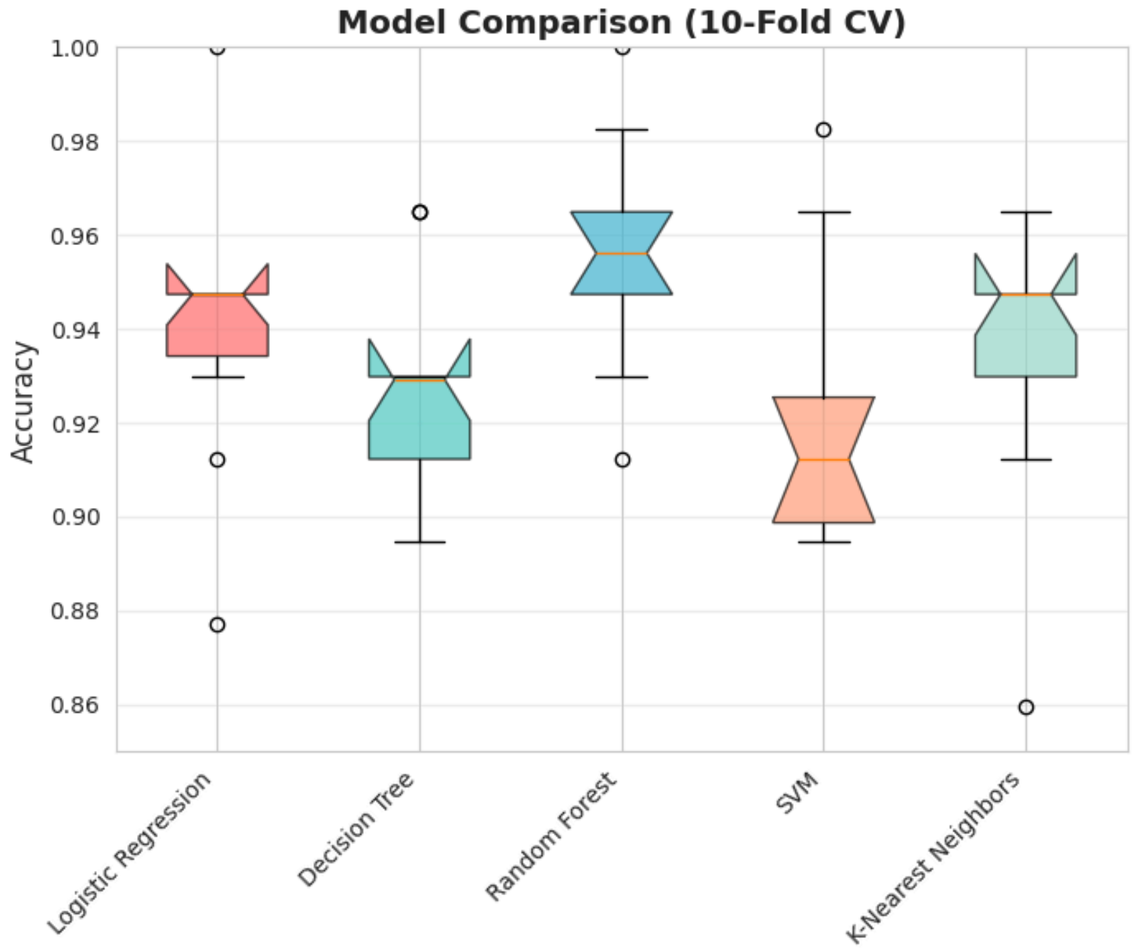
I know that the 98% is an illusion: Random seed 123 didn't give me a 'better model', it gave me an **easier test set**. Maybe that test set happened to have: more obvious cases.

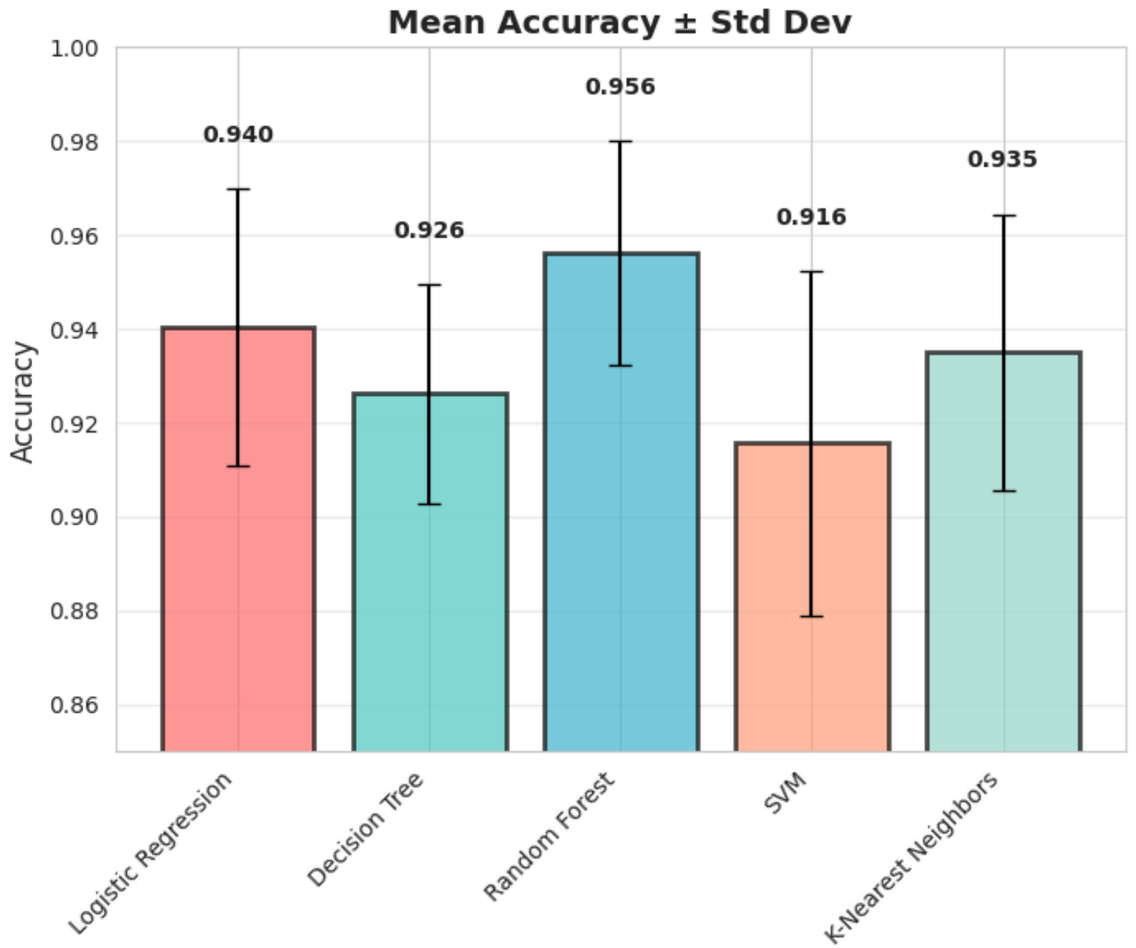
In production, we don't get to choose which patients walk through the door. I know that a **patient's life** might **depend on** which **random number** I chose!

So I decided to implement **cross validation** and **report mean accuracy** with **confidence intervals**.

FAANG-level Approach

Now we have more reliable metric with confidence interval





The Lesson: In healthcare, consistency matters as much as performance!

Accuracy tells us we're right 96% of the time 🎉

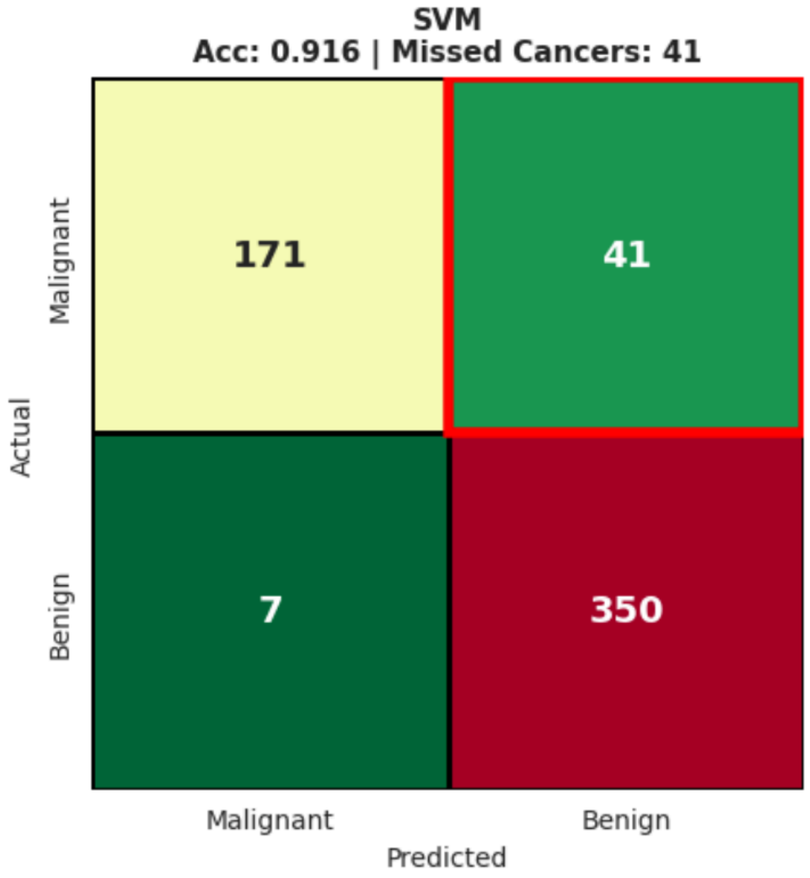
But is it enough metric to deploy in production, especially for the healthcare system?

We need to check where the model fails and think about the consequences!

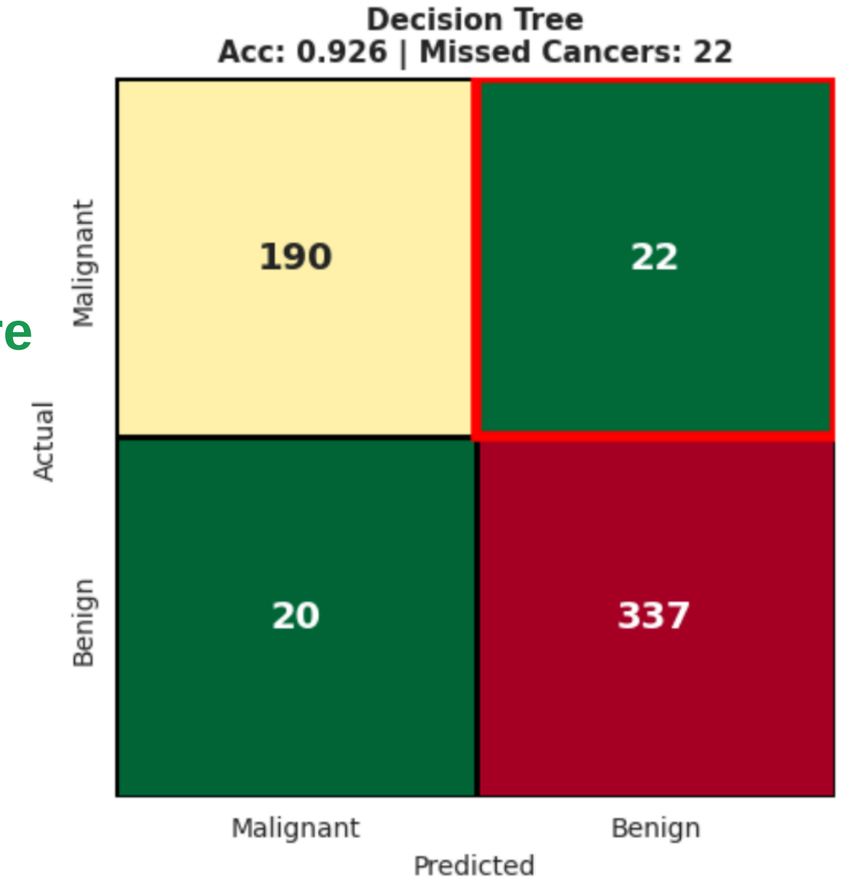
Accuracy tells us we're right 96% of the time 🎉

But is it enough metric to deploy in production, especially for the healthcare system?

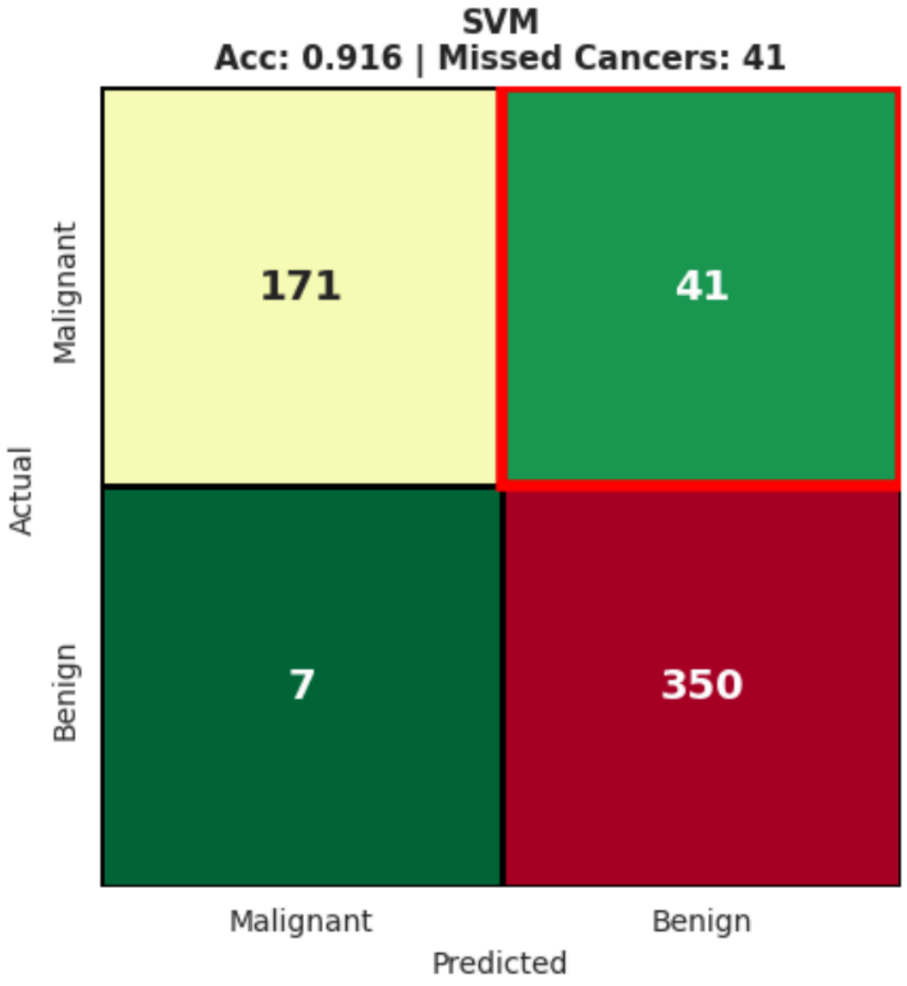
We need to check where the model fails and think about the consequences!



The accuracies are very close but what about where they fail?



Lets talk about consequences!



Missed cancers

Delayed treatment

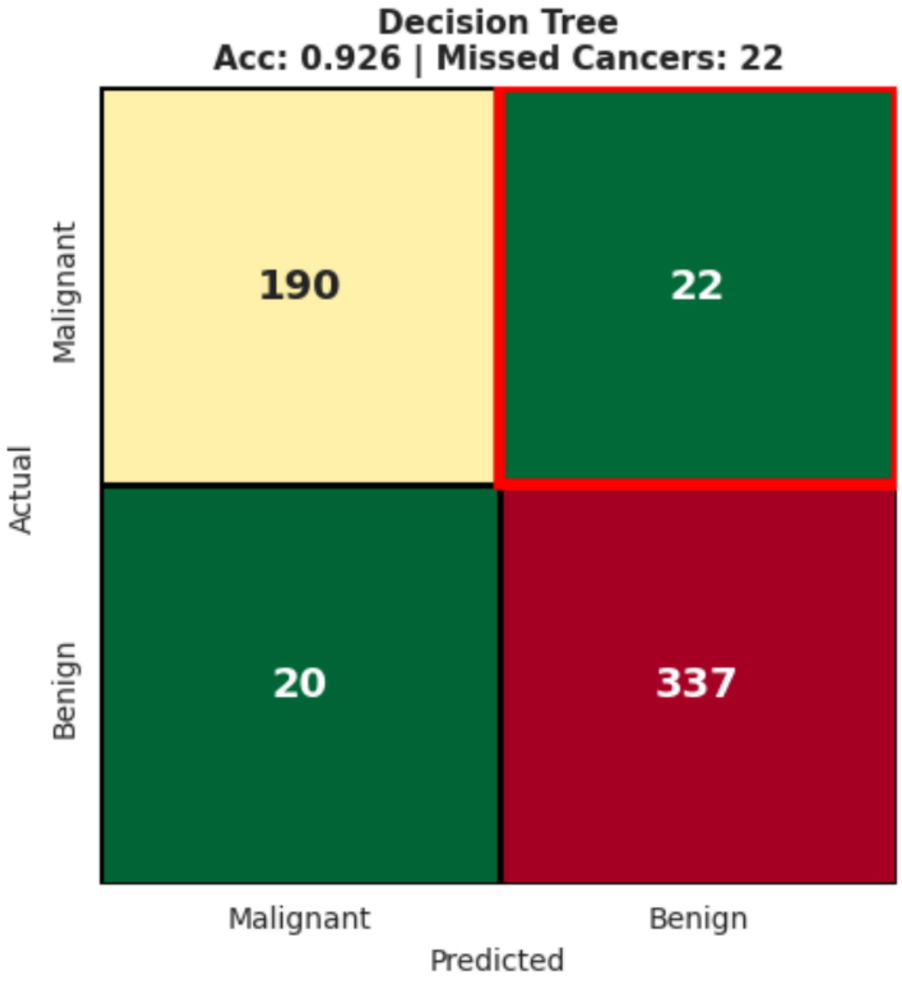
Cancer spreads

Potential death

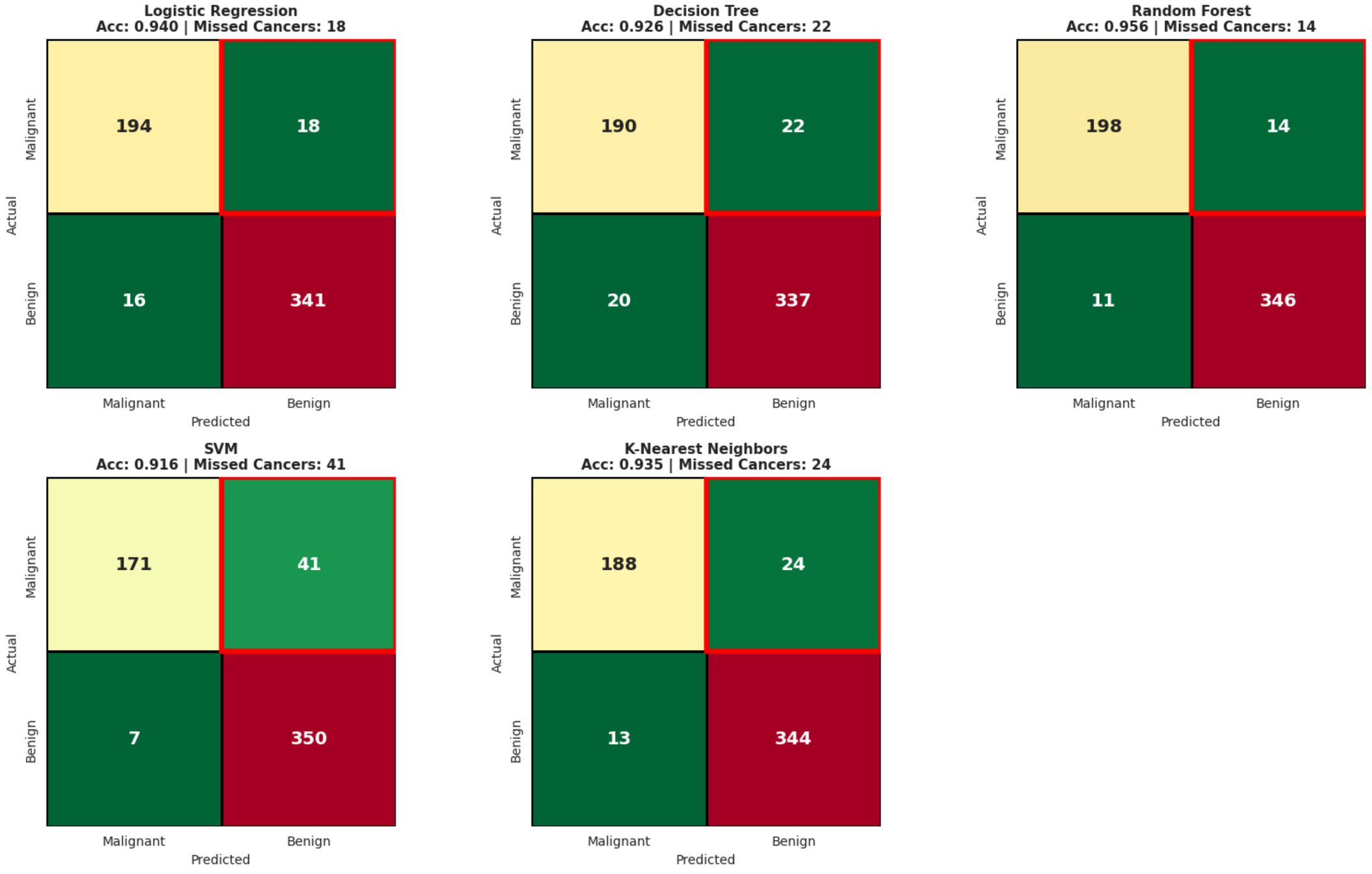
False Alarms

Stress

Unnecessary biopsy



Accuracy tells you almost nothing in medicine.


Confusion Matrices: Which Model Misses Fewer Cancers?
Logistic Regression
Acc: 0.940 | Missed Cancers: 18
194
18
16
341
Decision Tree
Acc: 0.926 | Missed Cancers: 22
190
22
20
337
Random Forest
Acc: 0.956 | Missed Cancers: 14
198
14
11
346
SVM
Acc: 0.916 | Missed Cancers: 41
171
41
7
350
K-Nearest Neighbors
Acc: 0.935 | Missed Cancers: 24
188
24
13
344
Actual
Malignant
Benign
Predicted
Malignant
Benign

We need other metrics!

Precision $\frac{TP}{TP + FP}$

Recall $\frac{TP}{TP + FN}$

Model	Accuracy	Precision	Recall	F1-Score	Missed Cancers
Logistic Regression	0.94	0.92	0.91	0.91	18
Decision Tree	0.92	0.90	0.89	0.90	22
Random Forest	0.95	0.94	0.93	0.94	14
SVM	0.91	0.96	0.80	0.87	41
KNN	0.93	0.93	0.88	0.91	24

How ROC curve comes into play

ROC (Receiver Operating Characteristic) curves are everywhere in classification tasks because they’re intuitive and give a sense of how well your model can separate positive from negative classes across different thresholds.

It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings:

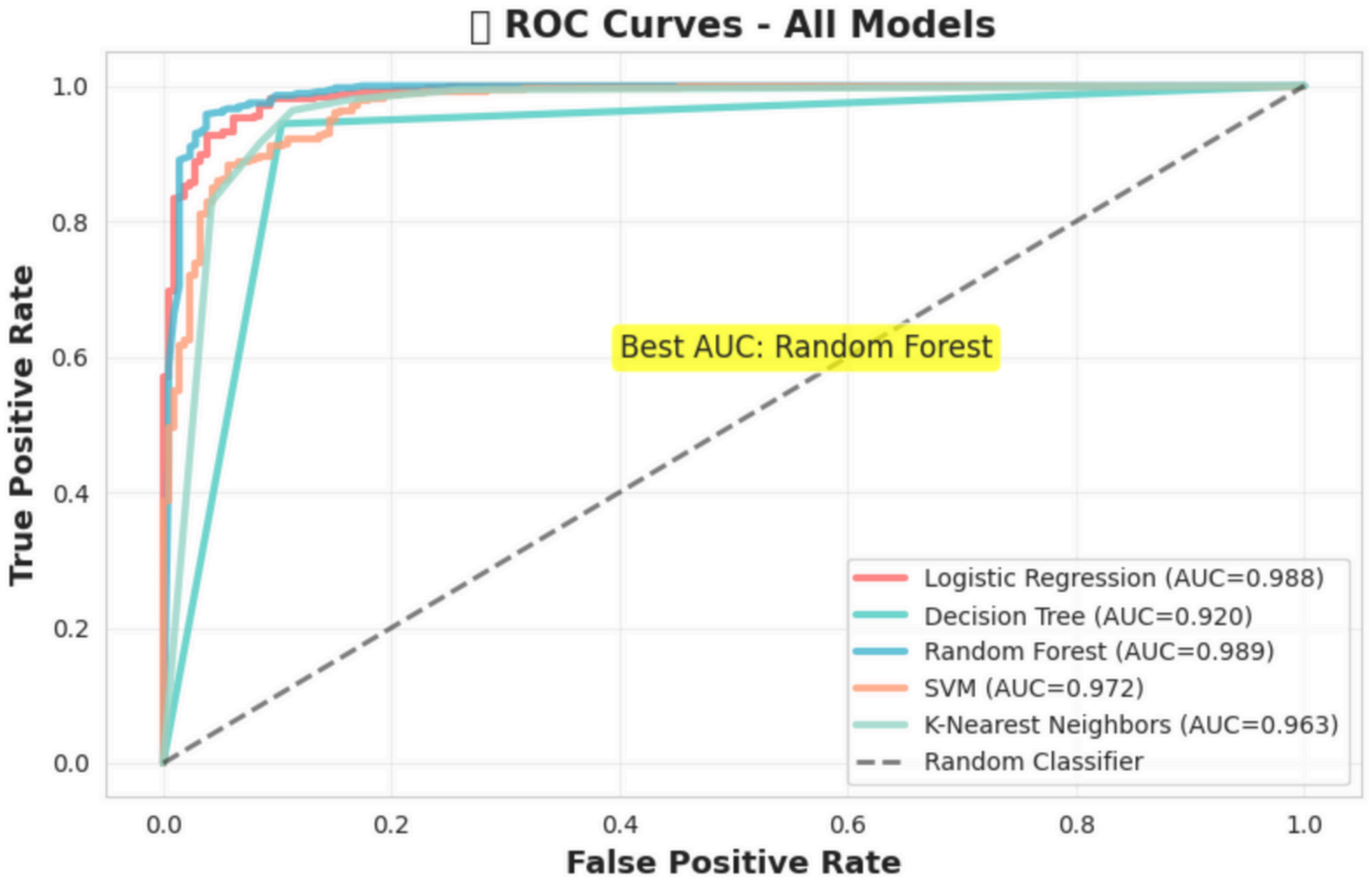
$$TPR = Recall = TP / (TP + FN)$$
$$FPR = FP / (FP + TN)$$

Each point on the curve represents a different classification threshold.

The is under the curve (AUC-ROC) shows the model’s ability to distinguish between classes: with 1.0 being perfect and 0.5 being random guessing.

Very short description: It’s a measure of separability.

How do we read the ROC curve?



We look at **how much** the **curve bows upward** away from the diagonal line.

A Good Model (The "Bowed" Curve): A good model lets you get a **high TPR before** the **FPR** starts **climbing**. Look at the "**elbow**" of the curve.

A great model might give you 0.90 TPR while the FPR is still at only 0.10.

Interview Moment: Catch with ROC Curve

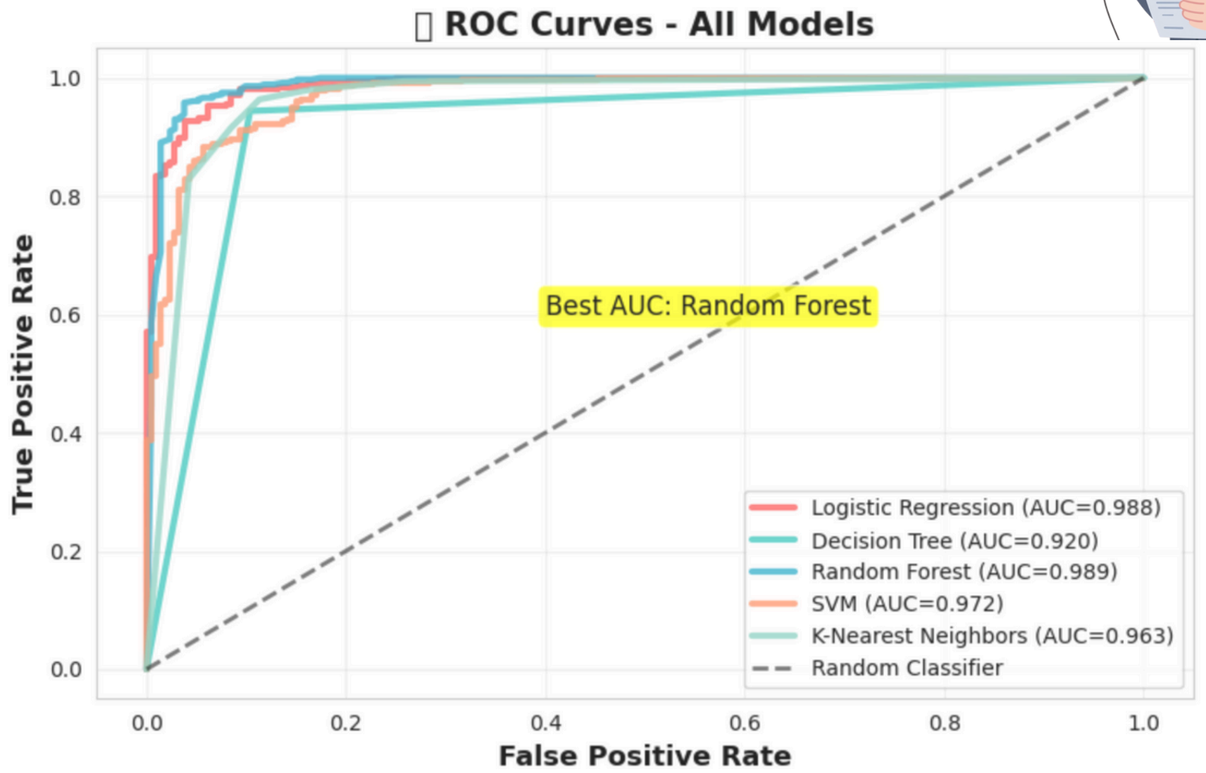
Up until this point you have analyzed different algorithms, reported accuracy results along with other metrics like precision and recall.

Finally, you have also plotted the ROC curve and calculated AUC scores.

At this point you realized something.

When you look at ROC-AUC, SVM seems doing great, but when you only look at the Recall score it is not the best choice.

How would you interpret this result, what would be your comments?



Model	Accuracy	Precision	Recall	F1-Score	Missed Cancers
SVM	0.91	0.96	**0.80**	0.87	41

Interview Moment: Catch with ROC Curve

ROC-AUC measures overall performance across different **thresholds**, while **recall** is **measured** at one **specific threshold** (0.5).

SVM might have better recall at other thresholds even though it's worse at 0.5.

So both metrics are correct, they're just **measuring different** things.

Threshold: Probability cutoff above which your model classifies a case as positive

if threshold = 0.5, then any prediction ≥ 0.5 is classified as positive, anything below is negative

Good Answer

Interview Moment: Catch with ROC Curve

ROC-AUC measures overall performance across different **thresholds**, while **recall** is **measured** at one **specific threshold** (0.5).

SVM might have better recall at other thresholds even though it's worse at 0.5.

So both metrics are correct, they're just **measuring different** things.

Threshold: Probability cutoff above which your model classifies a case as positive

if threshold = 0.5, then any prediction ≥ 0.5 is classified as positive, anything below is negative

Good Answer

SVM's AUC = 0.96 means:
The model has learned to **separate** the **two classes** well in probability space.

But at threshold 0.5, it classified many cancer patients as healthy (**low recall**).

High ROC AUC means our **model has** the **potential** for **good recall** - we just need to find the **right threshold** for our use case. The low recall is a **threshold problem**, **not** necessarily a **model quality** problem.

Random Forest has both higher AUC (0.98) and high recall (93.4%), so it's safer choice to calibrate further.

FAANG-level Answer

The Cost of Mistakes

When we are dealing with healthcare datasets we need to be careful with precision vs recall tradeoff as the both cases could bring different costs. We can play with the threshold to decide optimum threshold.

Threshold	Accuracy	Precision	Recall	Missed Cancers	False Alarms
0.3	0.942	0.994	0.92	31	2
0.5	0.958	0.972	0.961	14	10
0.7	0.944	0.927	0.982	6	26
0.9	0.891	0.835	0.99	3	59

Key Insight:

Lower threshold → Fewer false alarms (better precision), but miss some cancers

Higher threshold → Catch more cancers (better recall), but more false alarms

Where would YOU set the threshold ?

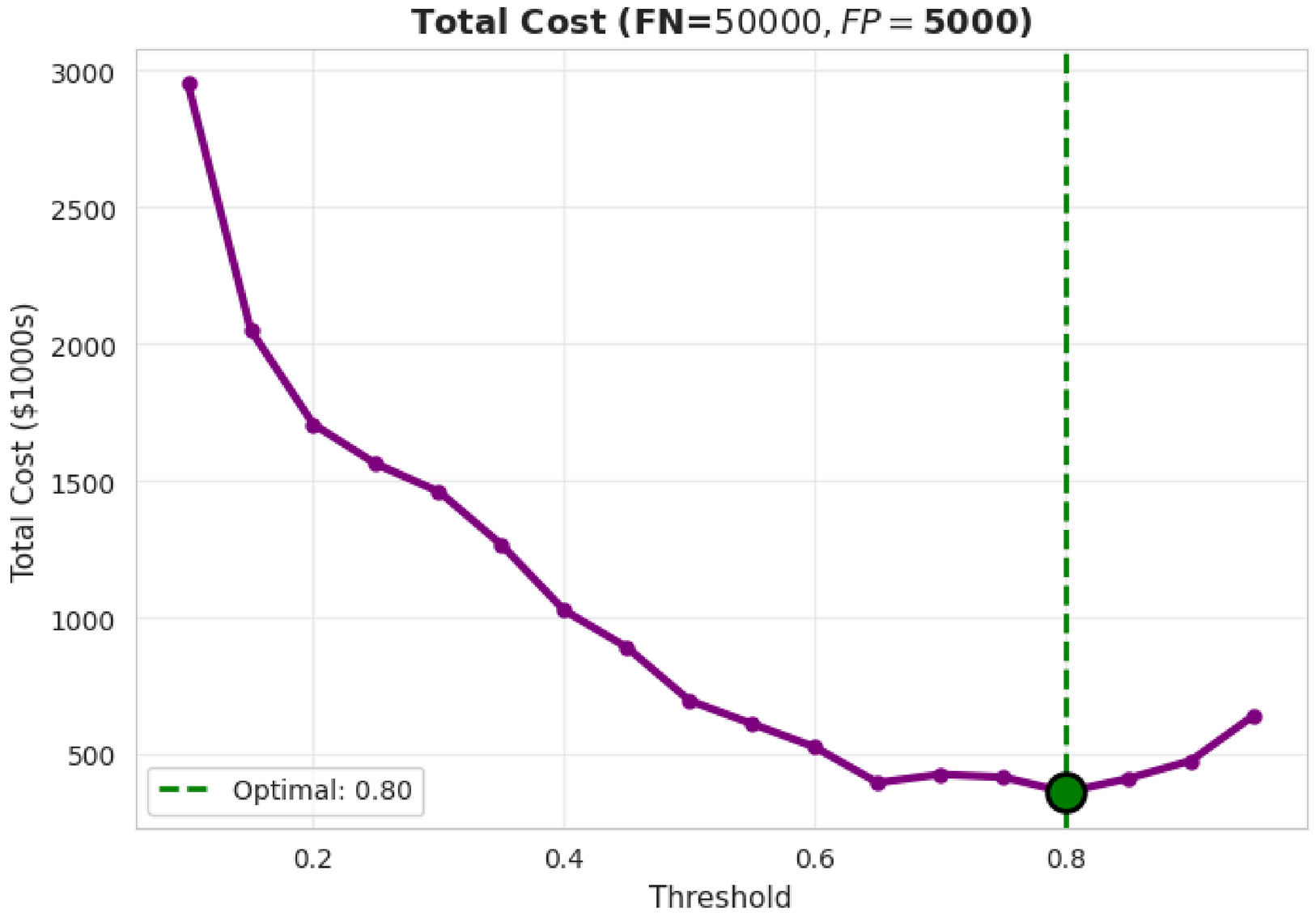
The Cost of Mistakes

Let's assume the costs of mistakes.

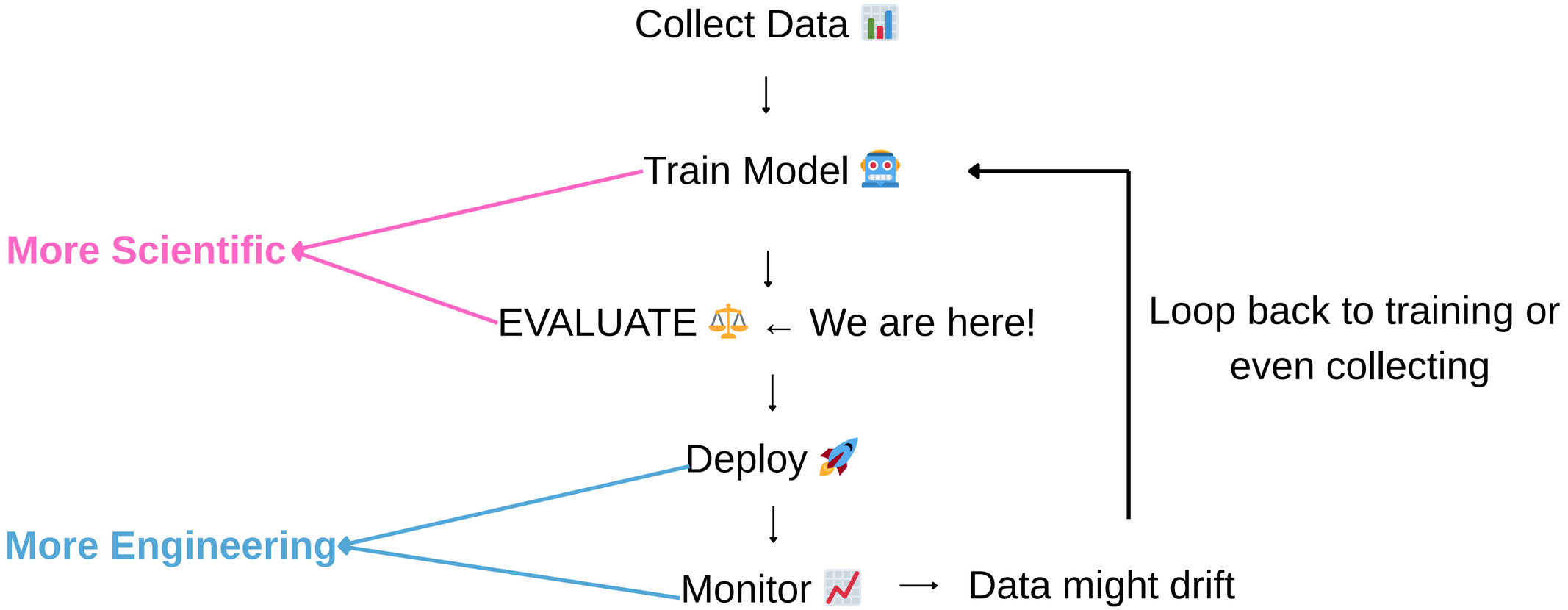
- Cost of False Negative (Saying a patient that they dont have the cancer where they actually have) is 50,000 $ for the hospital. Patient might sue the hospital delayed treatment might cost more money etc.

- Cost of a False Positive (saying a patient has cancer when they don’t) is $5,000, because the hospital will do unnecessary biopsy and extra work.

The Cost of Mistakes



The Loop in Data Scientist’s Life



PLEASE Dont Be That Person !

❌ Using test set for hyperparameter tuning

PLEASE Dont Be That Person !

❌ Using test set for hyperparameter tuning

✅ Use nested cross-validation

PLEASE Dont Be That Person !

❌ Using test set for hyperparameter tuning

✅ Use nested cross-validation

❌ Reporting only accuracy

PLEASE Dont Be That Person !

❌ Using test set for hyperparameter tuning

✅ Use nested cross-validation

❌ Reporting only accuracy

✅ Show confusion matrix, precision, recall

PLEASE Dont Be That Person !

❌ Using test set for hyperparameter tuning

✅ Use nested cross-validation

❌ Reporting only accuracy

✅ Show confusion matrix, precision, recall

❌ Ignoring model variance

PLEASE Dont Be That Person !

❌ Using test set for hyperparameter tuning

✅ Use nested cross-validation

❌ Reporting only accuracy

✅ Show confusion matrix, precision, recall

❌ Ignoring model variance

✅ Report mean ± std from CV

PLEASE Dont Be That Person !

❌ Using test set for hyperparameter tuning

✅ Use nested cross-validation

❌ Reporting only accuracy

✅ Show confusion matrix, precision, recall

❌ Ignoring model variance

✅ Report mean ± std from CV

❌ Choosing models blindly

PLEASE Dont Be That Person !

❌ Using test set for hyperparameter tuning

✅ Use nested cross-validation

❌ Reporting only accuracy

✅ Show confusion matrix, precision, recall

❌ Ignoring model variance

✅ Report mean ± std from CV

❌ Choosing models blindly

✅ Understand tradeoffs for your application

PLEASE Dont Be That Person !

❌ Using test set for hyperparameter tuning

✅ Use nested cross-validation

❌ Reporting only accuracy

✅ Show confusion matrix, precision, recall

❌ Ignoring model variance

✅ Report mean ± std from CV

❌ Choosing models blindly

✅ Understand tradeoffs for your application

❌ Forgetting about real-world costs

PLEASE Dont Be That Person !

❌ Using test set for hyperparameter tuning

✅ Use nested cross-validation

❌ Reporting only accuracy

✅ Show confusion matrix, precision, recall

❌ Ignoring model variance

✅ Report mean ± std from CV

❌ Choosing models blindly

✅ Understand tradeoffs for your application

❌ Forgetting about real-world costs

✅ Incorporate domain knowledge

Now is your turn

You can work in Teams

Total given time: 30 minutes

You need to do analysis as well and report the best model parameters