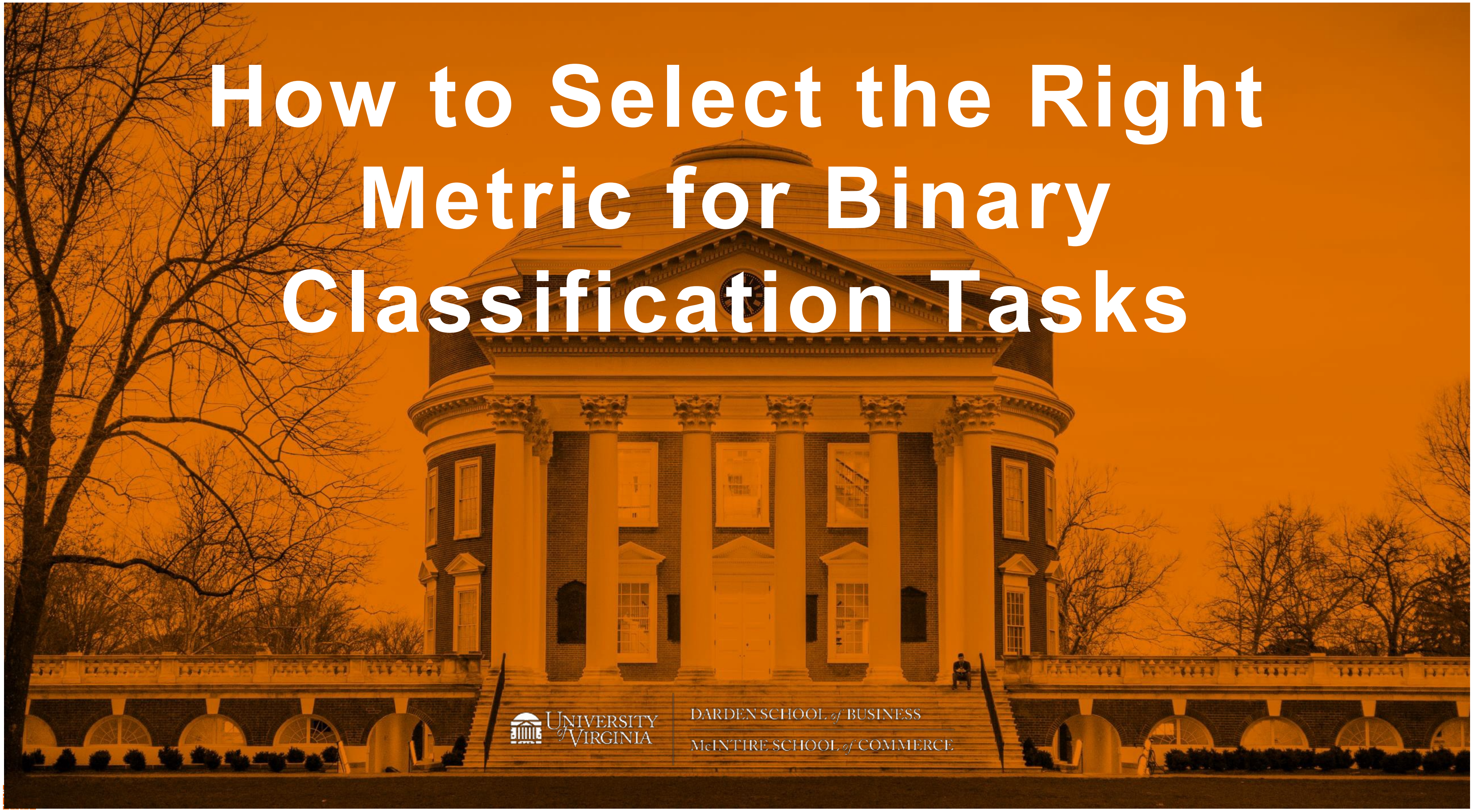
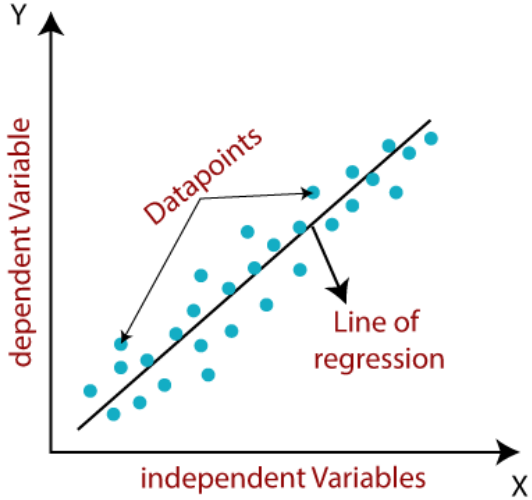
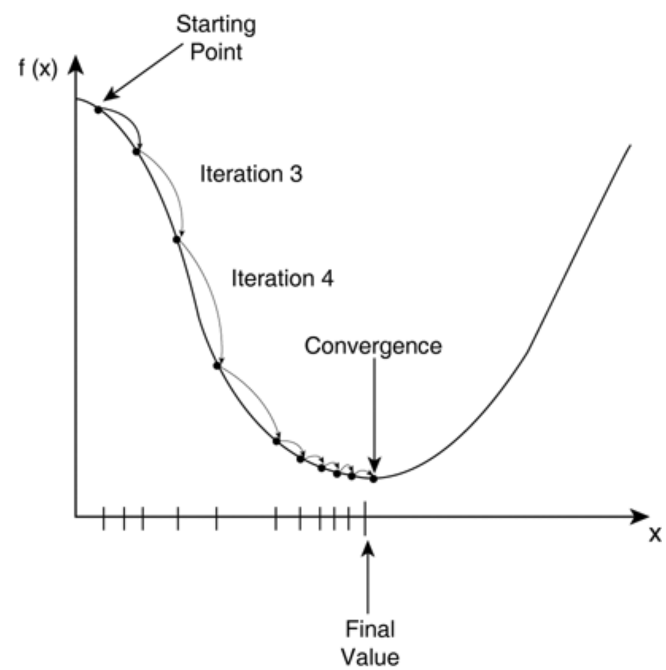
How to Select the Right Metric for Binary Classification Tasks



Objective Functions in Machine Learning

An objective function is a function that the algorithm uses to find the best values for the parameter(s) in a model.





The Objective Function in Classification (Cross Entropy)

Cross-entropy is a measure of the difference between two probability distributions for a given random variable or set of events.

ID	Actual	Predicted probabilities
ID6	1	0.94
ID1	1	0.90
ID7	1	0.78
ID8	0	0.56
ID2	0	0.51
ID3	1	0.47
ID4	1	0.32
ID5	0	0.10

ID	Actual	Predicted probabilities	Corrected Probabilities
ID6	1	0.94	0.94
ID1	1	0.90	0.90
ID7	1	0.78	0.78
ID8	0	**0.56**	**0.44**
ID2	0	**0.51**	**0.49**
ID3	1	0.47	0.47
ID4	1	0.32	0.32
ID5	0	**0.10**	**0.90**

The Objective Function in Classification

Now, we take the log of the corrected probabilities (Log(Corrected probabilities)):

ID	Actual	Predicted probabilities	Corrected Probabilities	Log
ID6	1	0.94	0.94	-0.0268721464
ID1	1	0.90	0.90	-0.0457574906
ID7	1	0.78	0.78	-0.1079053973
ID8	0	0.56	0.44	-0.3565473235
ID2	0	0.51	0.49	-0.30980392
ID3	1	0.47	0.47	-0.3279021421
ID4	1	0.32	0.32	-0.4948500217
ID5	0	0.10	0.90	-0.0457574906

Log Loss

The negative average of corrected probabilities is the Log loss or Binary cross-entropy:

$$- \frac{1}{N} \sum_{i=1}^{N} (\log(p_i))$$

ID	Actual	Predicted probabilities	Corrected Probabilities	Log
ID6	1	0.94	0.94	-0.0268721464
ID1	1	0.90	0.90	-0.0457574906
ID7	1	0.78	0.78	-0.1079053973
ID8	0	0.56	0.44	-0.3565473235
ID2	0	0.51	0.49	-0.30980392
ID3	1	0.47	0.47	-0.3279021421
ID4	1	0.32	0.32	-0.4948500217
ID5	0	0.10	0.90	-0.0457574906

→ Take the average and multiply by - → Log Loss

Log Loss Example- Model A

Observation	Actual Training Label/Class	Prediction Score	Corrected Prediction Score	Log(Corrected Prediction Score)
1	Yes	0.9	0.9	-0.046
2	No	0.2	0.8	-0.097
3	Yes	0.8	0.8	-0.097
4	Yes	0.1	0.1	-1
5	Yes	0.7	0.7	-0.155
6	Yes	0.3	0.3	-0.522
7	No	0.4	0.6	-0.222
8	No	0.3	0.7	-0.155
9	No	0.4	0.6	-0.222
10	No	0.6	0.4	-0.398

Log Loss = 0.291

Log Loss Exercise- Model B

Calculate Log Loss for Model B and compare with that of Model A. Which model is better?

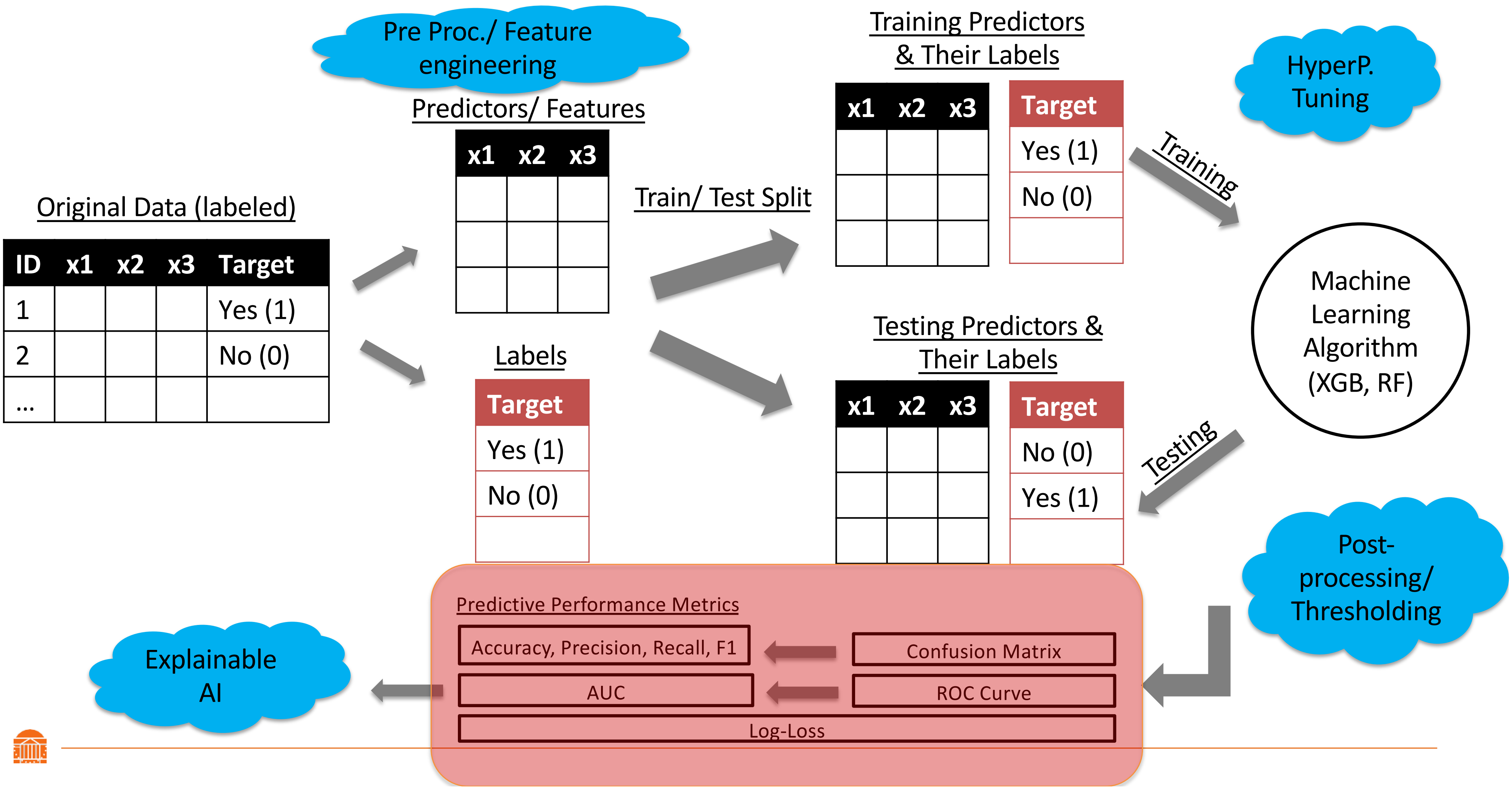
Observation	Actual Training Label/Class	Prediction Score	Corrected Prediction Score	Log(Corrected Prediction Score)
1	Yes	1.0		
2	No	0.1		
3	Yes	0.9		
4	Yes	0.4		
5	Yes	0.8		
6	Yes	0.4		
7	No	0.3		
8	No	0.2		
9	No	0.3		
10	No	0.5		

A Note on Log-Loss:

So far, we learned that Log-Loss is a metric that some of the classification algorithms such as XGBoost, LightGBM, and ANNs (Deep learning models) internally use to fit the data.

However, similar to AUC and accuracy, Log-Loss can also be used to compare the predictive performances of multiple models. For instance, if we have an XGBoost model and we want to compare its predictive performance with a RandomForest model, we can compute the Log-Loss for each model and select the model with a lower Log-Loss.

How to Develop a Classifier?



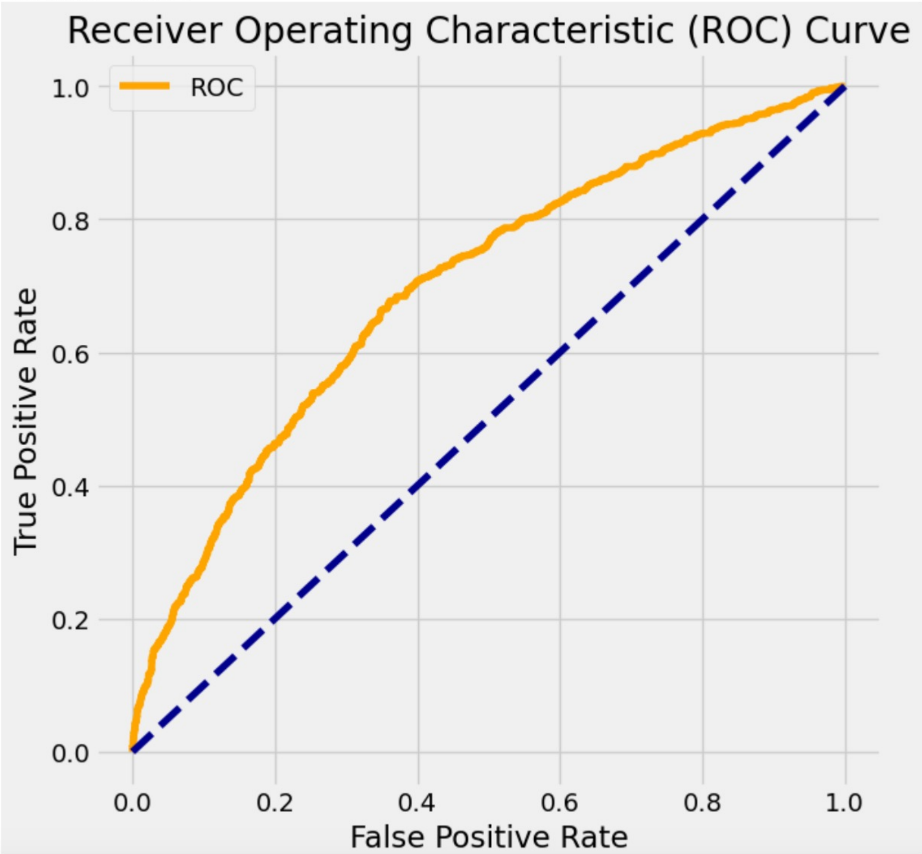
Confusion Matrix Metrics

Predicted Class	Actual Class		
		Class = Yes	Class = No
	Class = Yes	3	1
	Class = No	2	4

```
Accuracy: 74.83
PrecisionNegative: 79.49
PrecisionPositive: 48.09
RecallNegative: 89.79
RecallPositive: 29.00
F1 Score: 0.7483333333333333
```

ROC AUC

AUC Score = 69.658



Log-Loss

```
from sklearn.metrics import *
```

```
positiveProbabilities = predictedProbabilities[:,1]
```

```
# Calculate Log Loss
logloss = log_loss(testLabels, positiveProbabilities)

# Print Log Loss
print(f"Log Loss: {logloss}")
```

```
Log Loss: 0.03818363201196248
```

AUC vs. Log-Loss

Let's look at AUC_Log-Loss.ipynb notebook:

Discussion on AUC and Log-Loss

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import log_loss
plt.style.use('fivethirtyeight')
from custom_functions import plot_conf_mat, plot_roc_curve, plot_featur
```

Which Metric To Use?

Based on the model development stage and the way we use the outputs of our models in subsequent business processes, we should decide which metric (e.g., confusion matrix-based metrics, AUC, or Log-Loss) to use to compare models. In what follows, we elaborate on this.

Confusion Matrix vs. Others

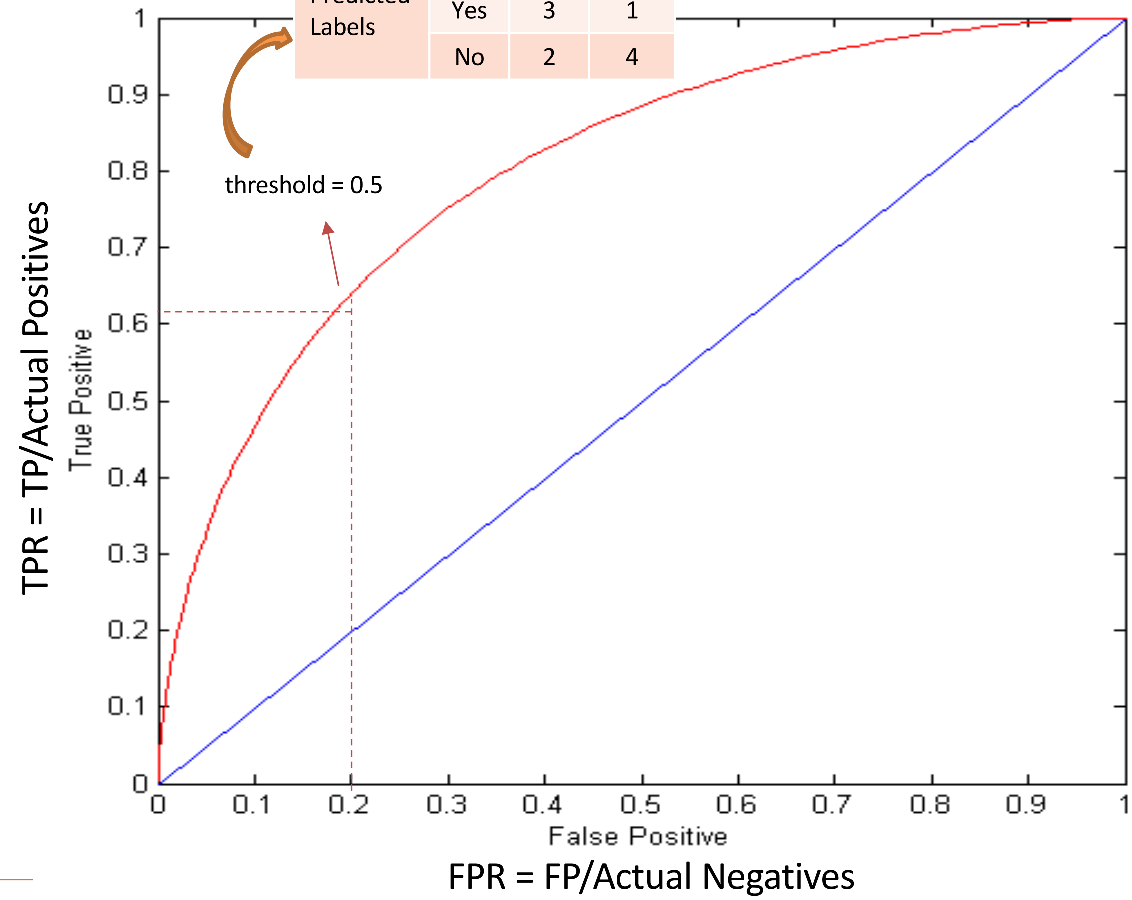
- Confusion matrix gives us a snapshot of the performance of the model for a specific threshold. Therefore, it is not great for examining the overall performance of the model across different thresholds.

- ***Rule 1: Hence, during the model development stage (e.g., choosing between different algorithms/ parameter values) use either AUC or Log-Loss.***

- Once you selected your best model based on AUC/ Log-Loss, you can use the confusion matrix to decide what threshold works best for your business problem. For instance, you can use precision and recall to determine the best threshold for your model. Essentially, use the confusion matrix to determine at which threshold you should use your best model.

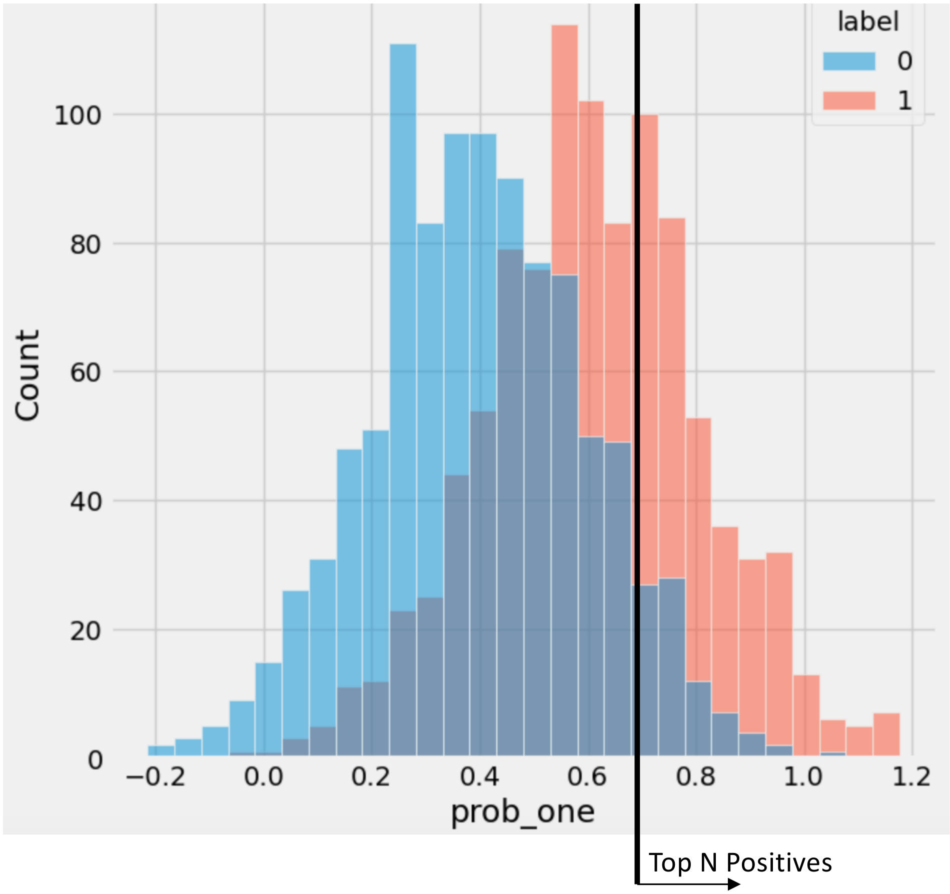
		Actual Labels	
		Yes	No
Predicted Labels	Yes	3	1
	No	2	4

FPR = 1/5 = 0.2
TPR = 3/5 = 0.6



AUC vs. Log-Loss

- AUC measures the model's ability to separate the two classes of samples (positives and negatives).
- Log-Loss measures the model's ability to give very high prob 1 score to positive samples and a very low prob 1 score to negative samples.
- ***Rule 2: Use AUC if all of the samples that the model predicted as positive will be used in the subsequent task.***
- ***Rules 3: Use Log-Loss if a portion of the samples that the model predicted as positive will be used in the subsequent task.***



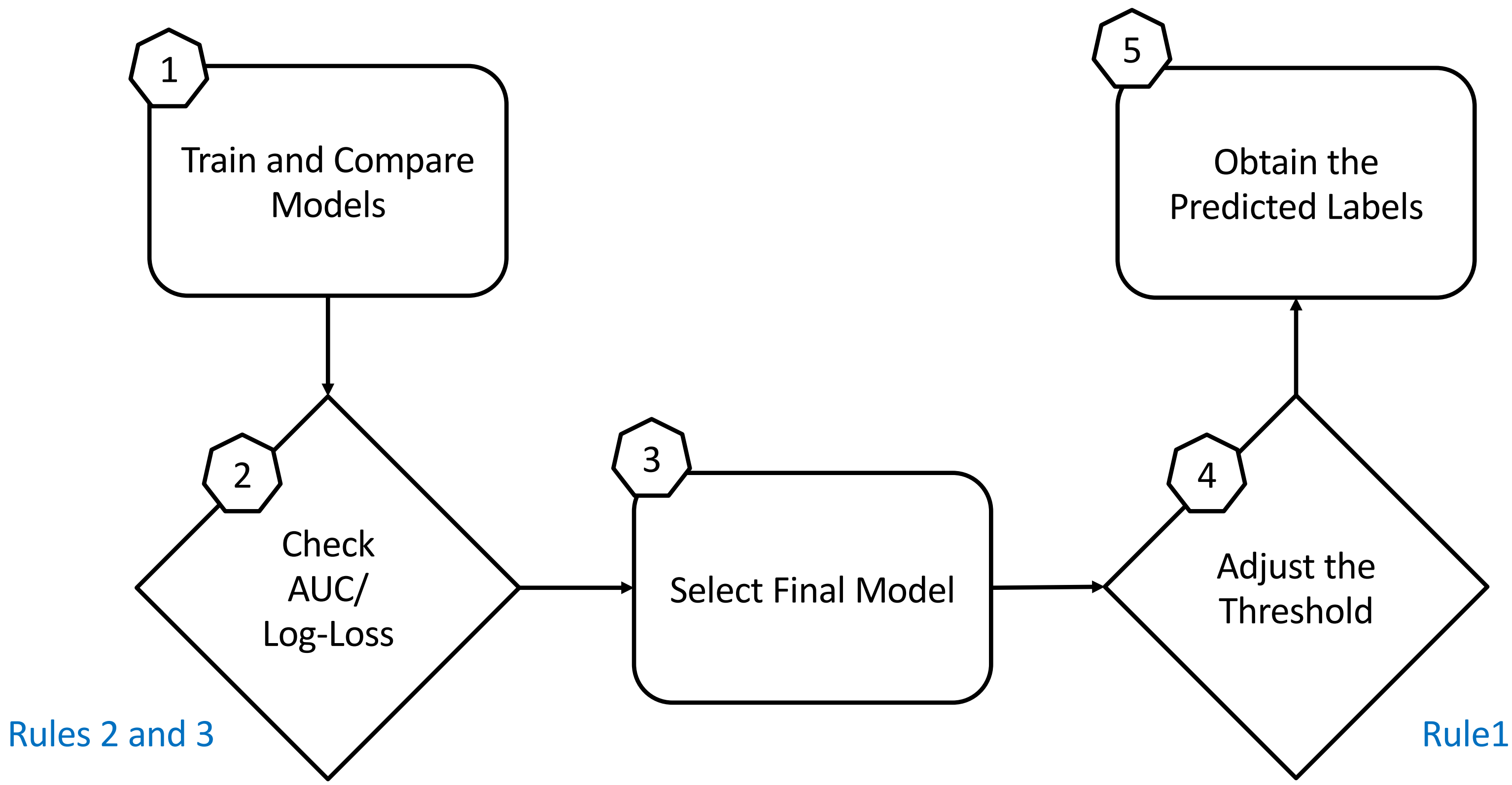
Exercise

In the following cases, determine whether you should use AUC or Log-Loss to compare models that you (or your data science teams) are developing:

1- Mortgage **Pre-Approval** Decision: You are building a model that takes factors related to the applicant, the property, and the local market conditions as inputs and determines whether a mortgage application should be pre-approved or not. The model should inform the applicant about the pre-approval decision.

2- Time & Expense (T&E) Audit: You are building a model that uses employees' T&E data (e.g., transaction amount, transaction description, vendor's location, employee's past transactions, employee's role, …) as inputs and determines whether a transaction submitted by an employee is non-compliant (e.g., personal expense instead of business expense). Once the model returns the probability scores for each sample, the top 1,000 samples based on the prob 1 score will be sent to an audit team for manual investigation. This 1,000-sample cap is imposed because of the audit team's limited resources.

To evaluate the models you (or your teams) are building in each case, would you use AUC or Log-Loss? Why?


1
Train and Compare Models
2
Check AUC/ Log-Loss
3
Select Final Model
4
Adjust the Threshold
5
Obtain the Predicted Labels
Rules 2 and 3
Rule1

?