JSC 270 - Lecture 6
Evaluating Predictions

https://jsc270.github.io/

Announcements 1

- Talk today 5:15-6:15pm EST: Tom Wright

  From mosquitos to flying saucers: Modelling in an imperfect world
- Program exploration days for Stats and CS

  Feb 23 9-11am, 2-4pm EST *Math, Physical and Computer Sciences*

  Feb 24 9-11am, 2-4pm EST *Life Sciences*

  Feb 25 9-11am, 3-5pm EST *Social Sciences, Humanities and Rotman*
-

Announcements 2

- Thank you for responding to the survey (Please respond if you haven't yet!)
  - Post lecture recordings earlier (will do)
  - Post Perusall and RQ grades (done), as well as Assignment 2
  - Distance between assignments - we post right away so you can start thinking about them, but you can wait a couple of days to start programming, just don't wait too long!
  - Perusall - a chance to get comfortable reading scientific papers and identify the holes in your knowledge!
  - Communication - Discourse is the primary mode, if you send email (last resort or special circumstances) - send to TAs and me (I get a lot of emails and don't want to miss it!!)
- Presentations - awesome job!
  - Introduce subject/motivation before talking about the data
  - Don't copy code output and put in presentations
- Reflection Quiz 3 will be online at 2pm today, due March 1st 11:59am

TODAY

How to evaluate and pick a classifier?

Data - AIRBNB. What predicts high occupancy rate?

host_response_rate	100	88	100	90	100	100	100	100	100	100
host_listings_count	1	1	2	5	2	1	5	2	4	4
host_total_listings_count	1	1	2	5	2	1	5	2	4	4
latitude	38.98	39.00	38.91	38.91	38.91	38.91	38.85	38.83	38.84	38.84
longitude	-77.02	-77.04	-77.03	-77.02	-77.02	-77.03	-77.00	-77.01	-76.98	-76.98
accommodates	4	1	4	2	5	2	3	4	4	5
bathrooms	1.5	1	1	1	2.5	1	1	2.5	1.5	1.5
bedrooms	2	1	1	1	2	1	1	2	1	1
beds	2	1	2	1	2	1	2	2	1	1
price	97	55	150	138	283	89	55	130	94	64
weekly_price	580	299	1100	1000	1650	524	295	800	559	379
monthly_price	2100	999	3700	2494	4400	1848	650	2800	1869	1129
security_deposit	250	100	100	250	500	200	150	300	95	95
guests_included	4	1	2	1	4	1	1	1	1	1
minimum_nights	4	3	2	1	3	1	1	2	1	1
maximum_nights	1125	1125	365	365	1125	1125	1125	1125	31	90
number_of_reviews	5	1	84	47	15	3	5	1	18	115
review_scores_rating	88	100	99	92	100	93	84	100	98	94
review_scores_accuracy	9	10	10	9	10	9	10	10	10	10
review_scores_cleanliness	9	6	10	8	10	9	8	10	10	10
review_scores_checkin	10	10	10	10	10	10	10	10	10	10
review_scores_communication	10	10	10	10	10	10	9	10	10	10
review_scores_location	9	10	10	8	10	10	6	6	9	9
review_scores_value	9	10	9	9	10	10	8	10	10	9
calculated_host_listings_count	1	1	2	4	1	1	5	2	3	3
reviews_per_month	0.22	1	2.91	0.87	1.06	0.64	2.73	0.45	0.99	6.1

Let's Train 2 classifiers

Logistic Regression

Random Forest

First step?

Let's Train 2 classifiers

Logistic Regression

Random Forest

First step? Split the data into train and test
Make sure to split once – keep
same train and test for both classifiers!

Split at Random: 80% train, 20% test

LR Train Accuracy: 0.67

LR Test Accuracy: 0.65

RF train accuracy: 0.72

RF test accuracy: 0.65

Split at Random: 80% train, 20% test

LR Train Accuracy: 0.67
LR Test Accuracy: 0.65

RF train accuracy: 0.72
RF test accuracy: 0.65

... but the class frequency in the test set is 34 %, so ....

Split at Random: 80% train, 20% test

LR Train Accuracy: 0.67

LR Test Accuracy: 0.65

RF train accuracy: 0.72

RF test accuracy: 0.65

... but the class frequency in the test set is 34 %, so ....
a classifier that predicts all 0 will be 66% accurate!!!

So what do you think about this classifier?

So what do you think about this classifier?

There are many other criteria for comparison!

What metrics to compare classification models do you know?

What are the metrics to compare classification models?

- Accuracy
- False Positives
- True Positives
- False Negatives
- True Negatives
- Precision
- Recall
- Sensitivity
- Specificity
- Positive and Negative Predictive Values
- False Discovery Rate
- F1

- ROC (Receiver Operating Curve)
- AUC (Area Under ROC)
- Precision recall (PR)
- AUPR (Area Under PR)

Confusion Matrix

	Observed Positive	Observed Negative
Predicted Positive		
Predicted Negative		

Confusion Matrix

	Observed Positive	Observed Negative
Predicted Positive	True Positives (TP)	False Positives (FP)
Predicted Negative	False Negatives (FN)	True Negatives (TN)

Scenarios

When do we care about False Positives more?

When do we care about False Negatives more?

Scenarios

When do we care about False Positives more?

E.g. Predicting invasive procedure (surgery)

When do we care about False Negatives more?

E.g. Screening (we don’t care if some we thought were sick were not sick, but we def don’t want sick people to be walking around!)

Defining measures

Accuracy =

Precision =

Recall =

Sensitivity =

Specificity =

PPV =

NPV =

F1 =

Defining measures

Accuracy = (TP + TN)/ (P + N)

Precision =

Recall =

Sensitivity =

Specificity =

PPV =

NPV =

F1 =

P - all observed positives
N - all observed negatives

Defining measures

P - all observed positives
N - all observed negatives

Accuracy = (TP + TN)/ (P + N)

Precision = TP / (TP + FP)

Recall =

Sensitivity =

Specificity =

PPV = Precision = 1–FDR, FDR – False Discovery Rate

NPV =

F1 =

Defining measures

P - all observed positives
N - all observed negatives

Accuracy = (TP + TN)/ (P + N)

Precision = TP / (TP + FP)

Recall = TPR = TP/P

Sensitivity = Recall

Specificity =

PPV = Precision = 1-FDR, FDR - False Discovery Rate

NPV =

F1 =

Defining measures

Accuracy = (TP + TN)/ (P + N)

Precision = TP / (TP + FP)

Recall = TPR = TP/P

Sensitivity = Recall

Specificity = TNR = TN/N = 1–FPR, FPR = FP/N

PPV = Precision = 1–FDR

NPV =

F1 =

P - all observed positives
N - all observed negatives

Defining measures

Accuracy = (TP + TN)/ (P + N)

Precision = TP / (TP + FP)

Recall = TPR = TP/P

Sensitivity = Recall

Specificity = TNR = TN/N = 1-FPR, FPR = FP/N

PPV = Precision = 1-FDR

NPV = TN /(TN + FN)

F1 =

P - all observed positives
N - all observed negatives

Defining measures

P – all observed positives
N – all observed negatives

Accuracy = (TP + TN)/ (P + N)

Precision = TP / (TP + FP)

Recall = TPR = TP/P

Sensitivity = Recall

Specificity = TNR = TN/N = 1–FPR, FPR = FP/N

PPV = Precision = 1–FDR

NPV = TN /(TN + FN)

F1 = 2 (PPV*TPR)/(PPV+TPR) = 2TP/(2TP+FP+FN)

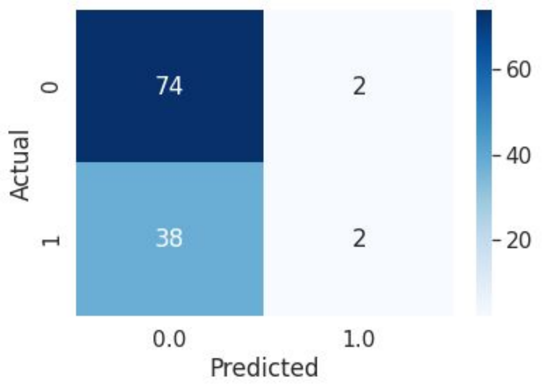
Classification

	Observed True	Observed False	
Predicted True	True Positives (TP)	False Positives (FP)	PPV,Precision, FDR
Predicted False	False Negatives (FN)	True Negatives (TN)	NPV
	TPR,Recall, Sensitivity	FPR,Specificity, TNR	Accuracy, F1

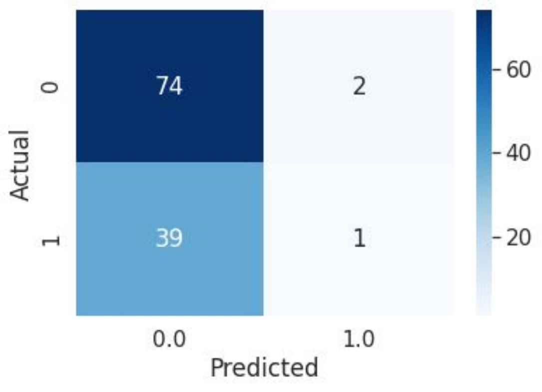
https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Comparing measures on AIRBNB data

LR



RF



	LR	RF
Accuracy	.65	.65
Sensitivity/Recall	0.05	.025
Specificity/TNR	.97	.97
Precision	0.5	.33
F1	0.09	.05

Assumption we made when computing previous measures is that we are making a class prediction directly.

What if we predicted probabilities of a class being 1 (or 0)?

Assumption we made when computing previous measures is that we are making a class prediction directly.

What if we predicted probabilities of a class being 1 (or 0)?

Then we could draw different thresholds!

ROC

ROC - Receiver Operating Characteristic curve

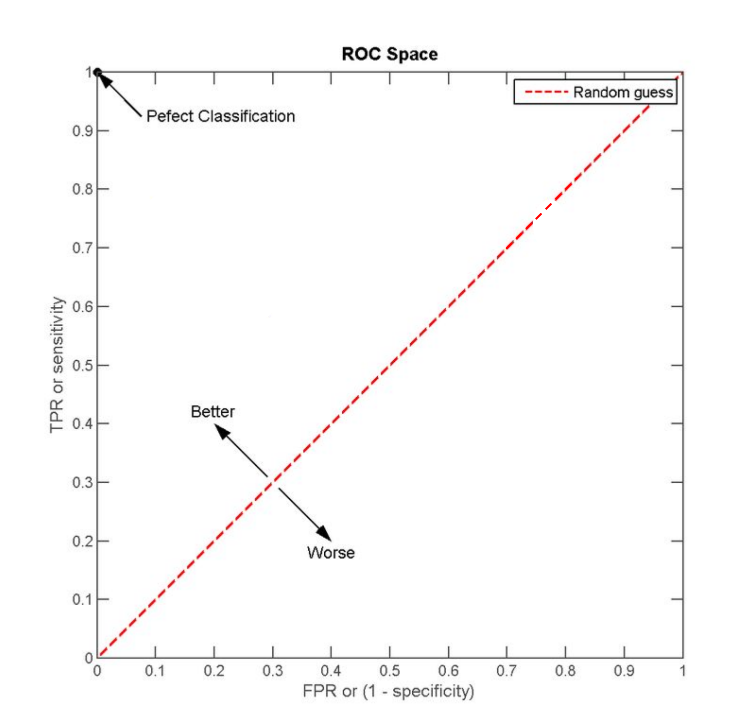
First used during World War II to analyze radar signals following the attack on Pearl Harbor 1941

Now it’s the most commonly used classifier evaluation criterion!

ROC - HOW TO COMPUTE

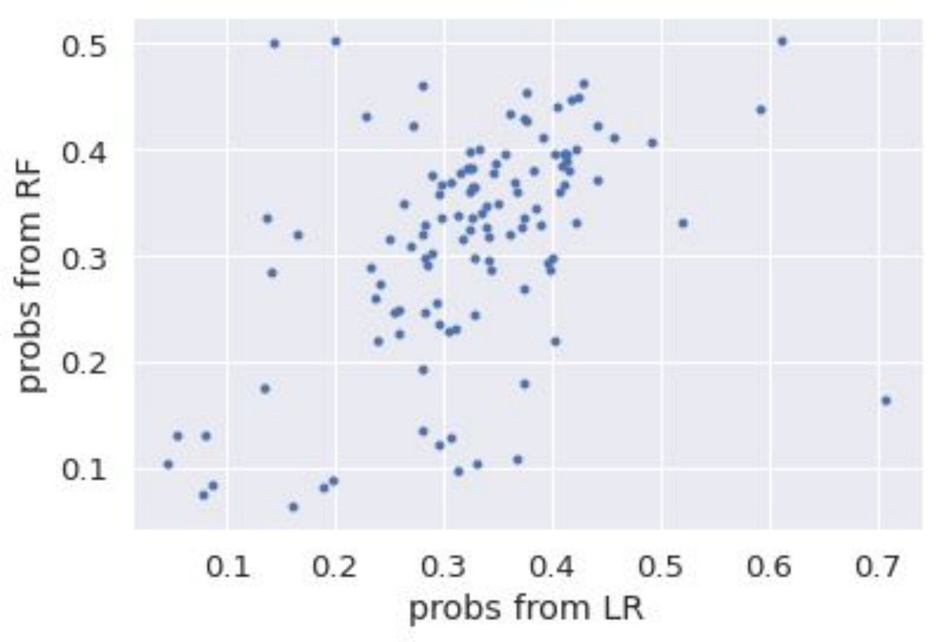
1. Order probabilities from highest to lowest

2. Start from the highest probability and draw a threshold at each point, each time checking TPR and FPR

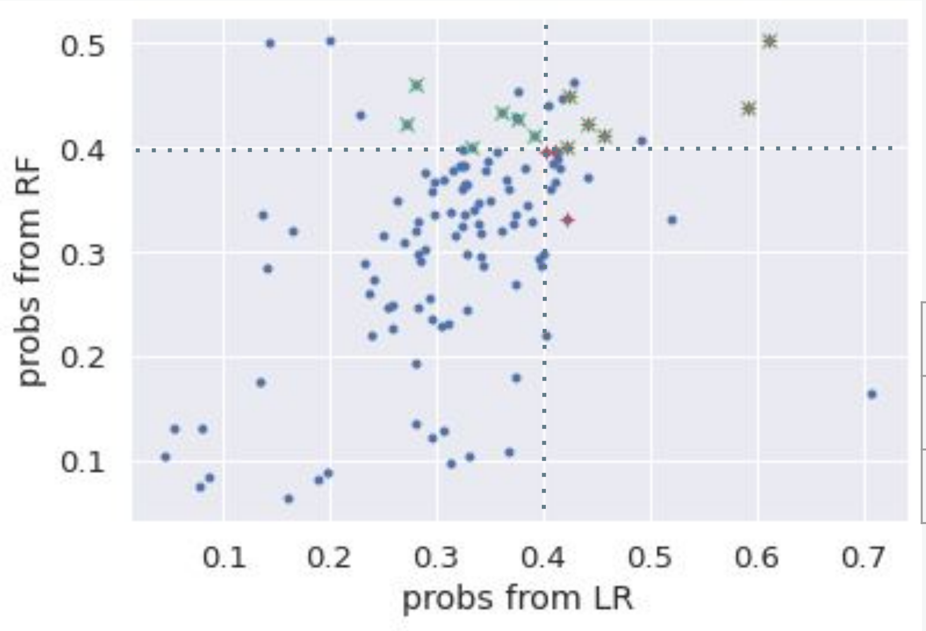


Note 1: ROC – fraction of correct predictions for positive class vs fraction of incorrect predictions for negative class

Computing ROC - Airbnb



Computing ROC - AIRBNB



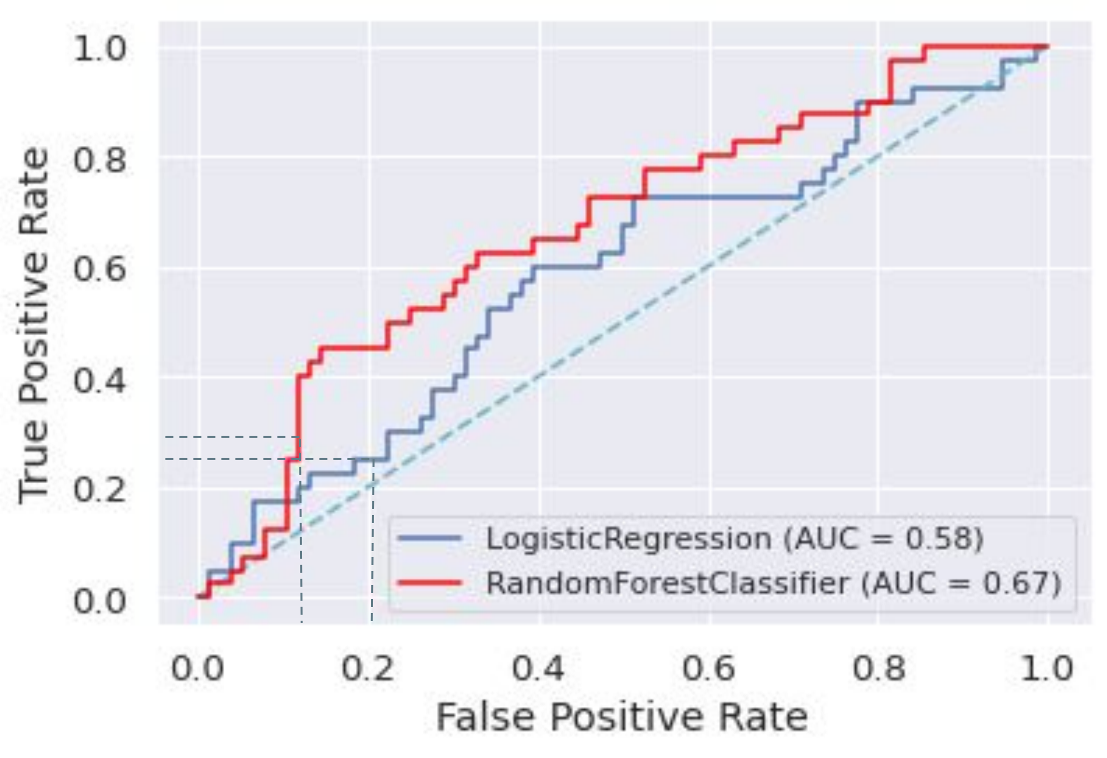
+ – LR TP

x – RF TP

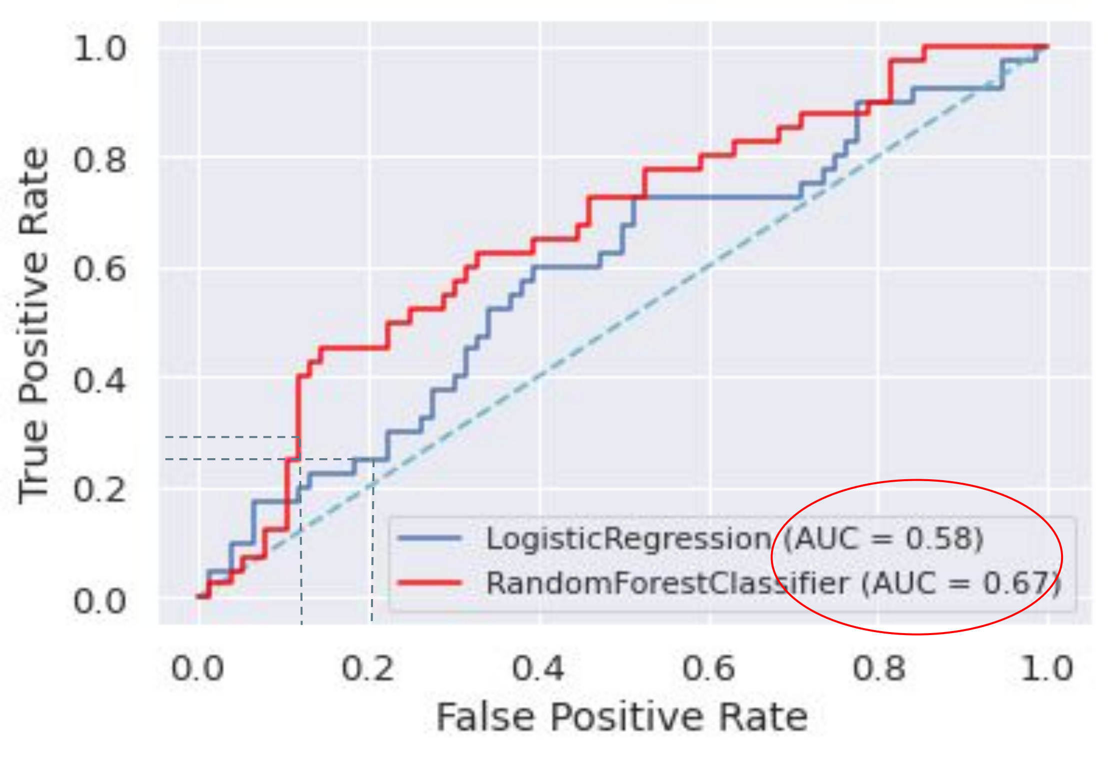
At prob threshold 0.4

	LR	RF
TPR	0.25	0.3
FPR	.2	.12

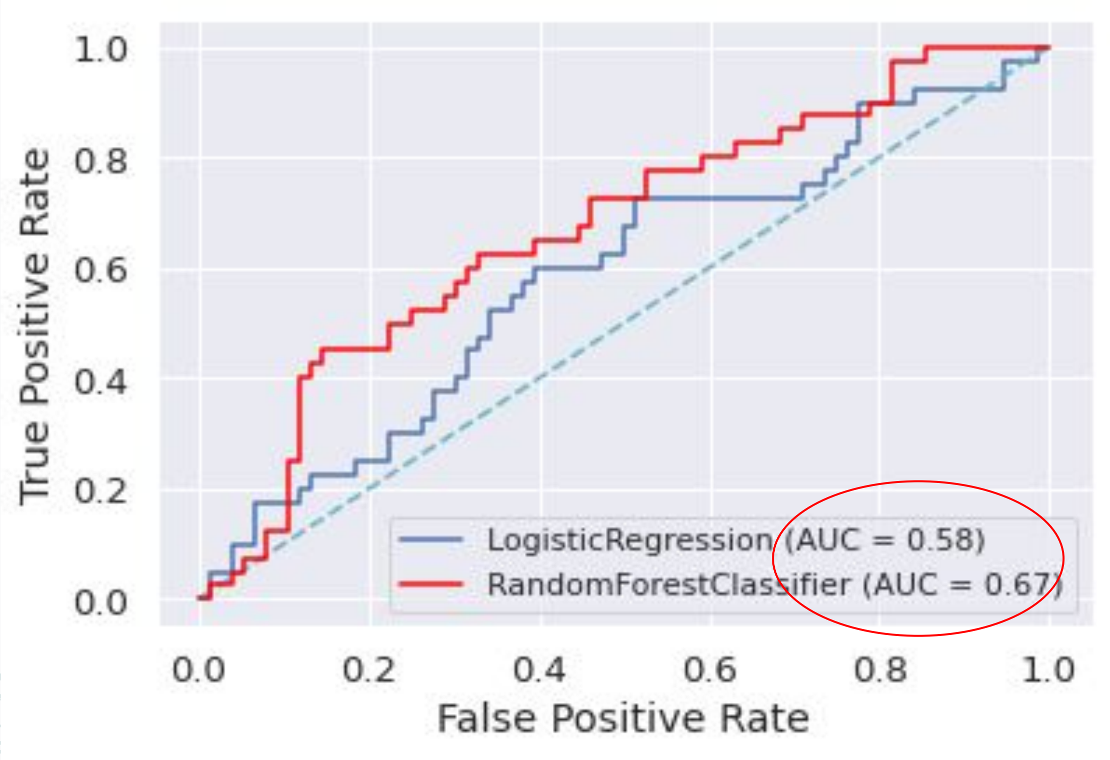
ROCs in AIRBNB example



ROCs in AIRBNB example



ROCs in AIRBNB EXAMPLE



Pick a threshold depending on the problem

ROC

Advantages

- Captures a lot of information
- Can see behavior of the model at different thresholds
- Doesn’t matter if the data is imbalanced

Disadvantages

❌ If the data is severely imbalanced, each data point will make a big difference!

Precision Recall Curve

Remember:

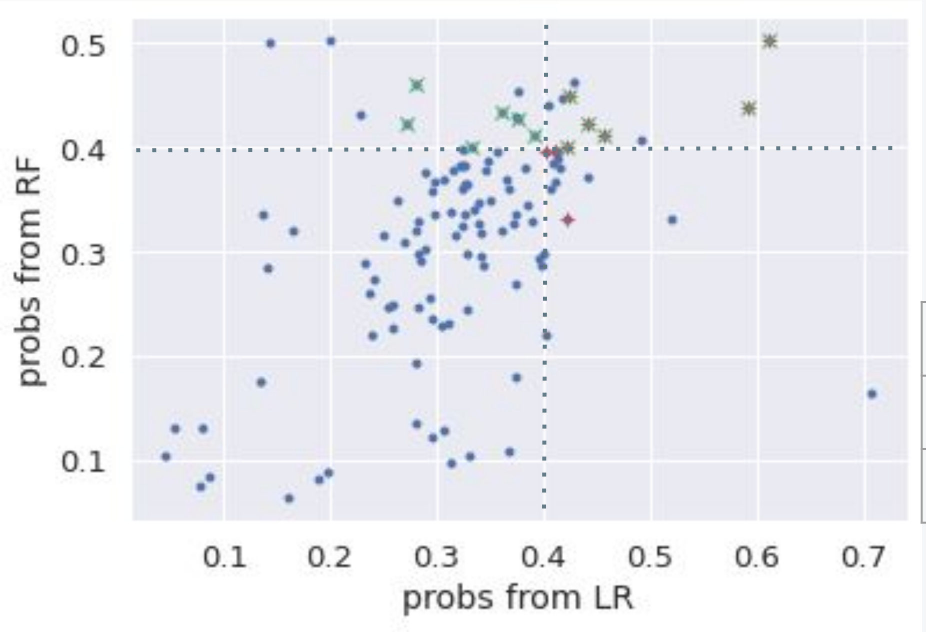
Precision = TP/(TP+FP)
- number of correctly made positive predictions out of all predictions made

Recall = TPR = TP/P
- number of correctly made positive predictions out of all that *could* be made

Compute: compute Precision and Recall for each prob threshold!

Note 1: Focus is on the minority (positive) class, majority class doesn’t matter

Computing Precision-Recall curve - AIRBNB



\+ - LR TP

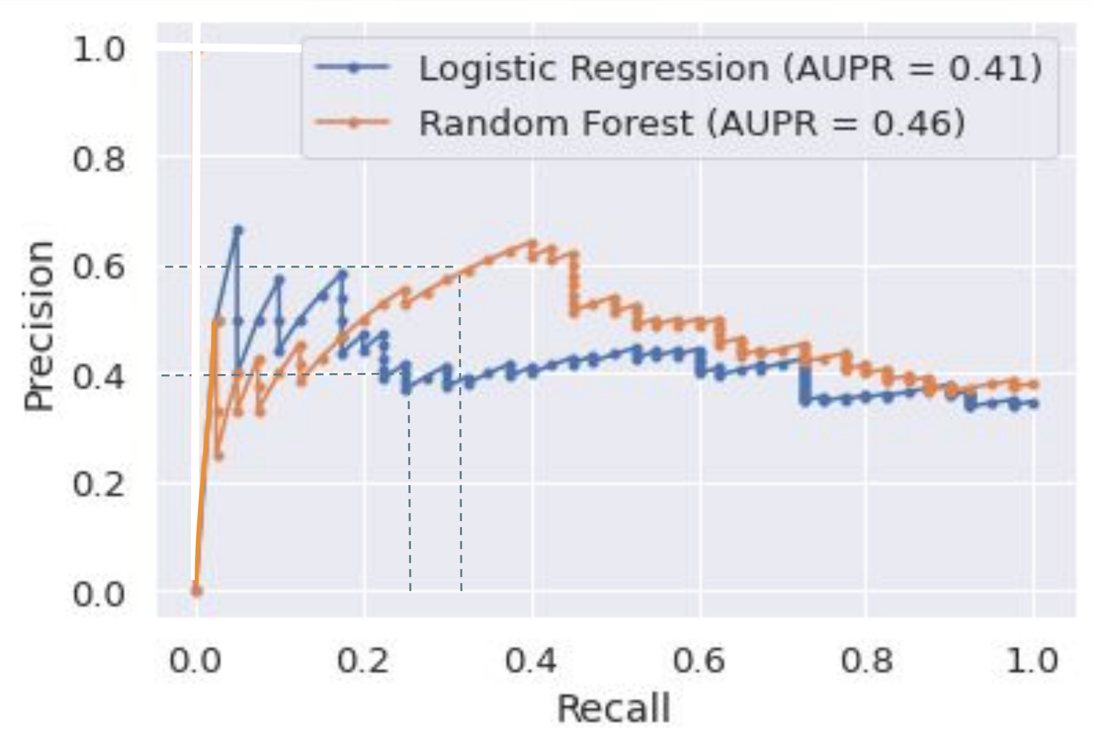
x - RF TP

At prob threshold 0.4

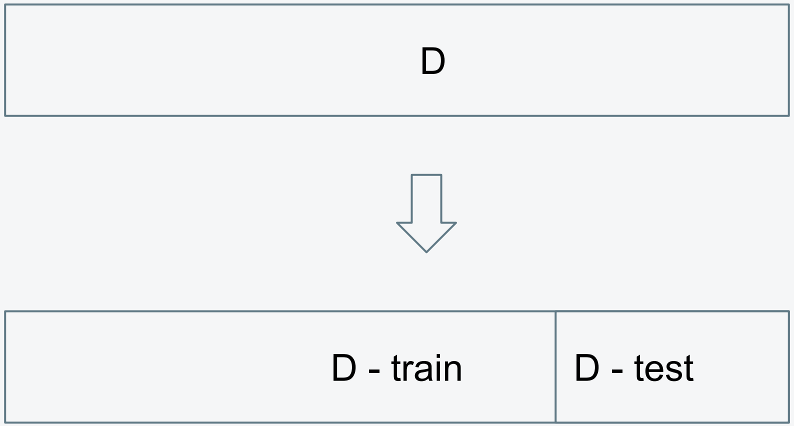
	LR	RF
Recall/TPR	0.25	0.3
Precision	0.4	0.6

Note 2: Precision may not decrease with recall

Precision Recall Curve for Airbnb



So far, we had 1 split of data into train and test...



Reminder...

The algorithm
CANNOT see this data!

D - train	D - test

The algorithm
CANNOT see this data!

D - train

D - test

What do we do if we want to select parameters, etc?
Can’t use Test data to optimize....

Cross-validation

Training and Testing

$D = D1 \cup D2 \cup \dots \cup D_k$

D1	D2	D3

The algorithm
CANNOT see this data!

D - test

Train $D1 \cup D2$, validate on D3

Train $D1 \cup D3$, validate on D2

Train $D2 \cup D3$, validate on D1

Cross-validation

Training and Testing

D = D1 ∪ D2 ∪ … ∪ Dk

D1	D2	D3

The algorithm
CANNOT see this data!

D - test

k=3
3-fold
Cross - validation

Train D1 ∪ D2, validate on D3

Train D1 ∪ D3, validate on D2

Train D2 ∪ D3, validate on D1

Cross-validation

Training and Testing

D = D1 ∪ D2 ∪ ... ∪ Dk

The algorithm
CANNOT see this data!

D1	D2	D3

D - test

k=3
3-fold
Cross - validation

Train D1 ∪ D2, validate on D3

Train D1 ∪ D3, validate on D2

Train D2 ∪ D3, validate on D1

If you are testing the same algorithm, averaging your evaluation criteria is a better indication of the performance on the held out test data!!

Hidden Dragons

Selecting Test data can be tricky!

The algorithm
CANNOT see this data!

D - train	D - test

Hidden Dragons

Selecting Test data can be tricky!

The algorithm CANNOT see this data!

D - train	D - test

- Selecting at random might not work if a process is evolving in time – need to train on past, test on future!
  (e.g. COVID data – varies wildly over time, random selection of test will inflate the results!)

- Might need to make sure that the same item/person is not in both train and test! (e.g. xray for patients over time, need to make sure that the same patient is not in train and test)

Balancing Issue

When the classes are very imbalanced (not 50%/50% in your data and in life), some algorithms will have trouble learning a good model

Potential Balancing Solutions

Potential Balancing Solutions

- Downsampling majority class (if enough samples)
- Upsampling
- Balancing metrics (e.g. when computing impurity, can weigh the classes differently)

Pros and Cons of balancing

Pros: the algorithm doesn't have to be fighting the bias in the data

Cons: if downsampling, using much less of the data to train the algorithm

Note: remember to not balance the test set – test set should represent the reality as much as possible!

Reading Material

- All measures for a given threshold: https://en.wikipedia.org/wiki/Sensitivity_and_specificity
- Multi-class precision recall https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bddc2
- Animation for building an ROC curve from probability vector: http://homepage.stat.uiowa.edu/~rdecook/stat6220/ROC_animated2.html