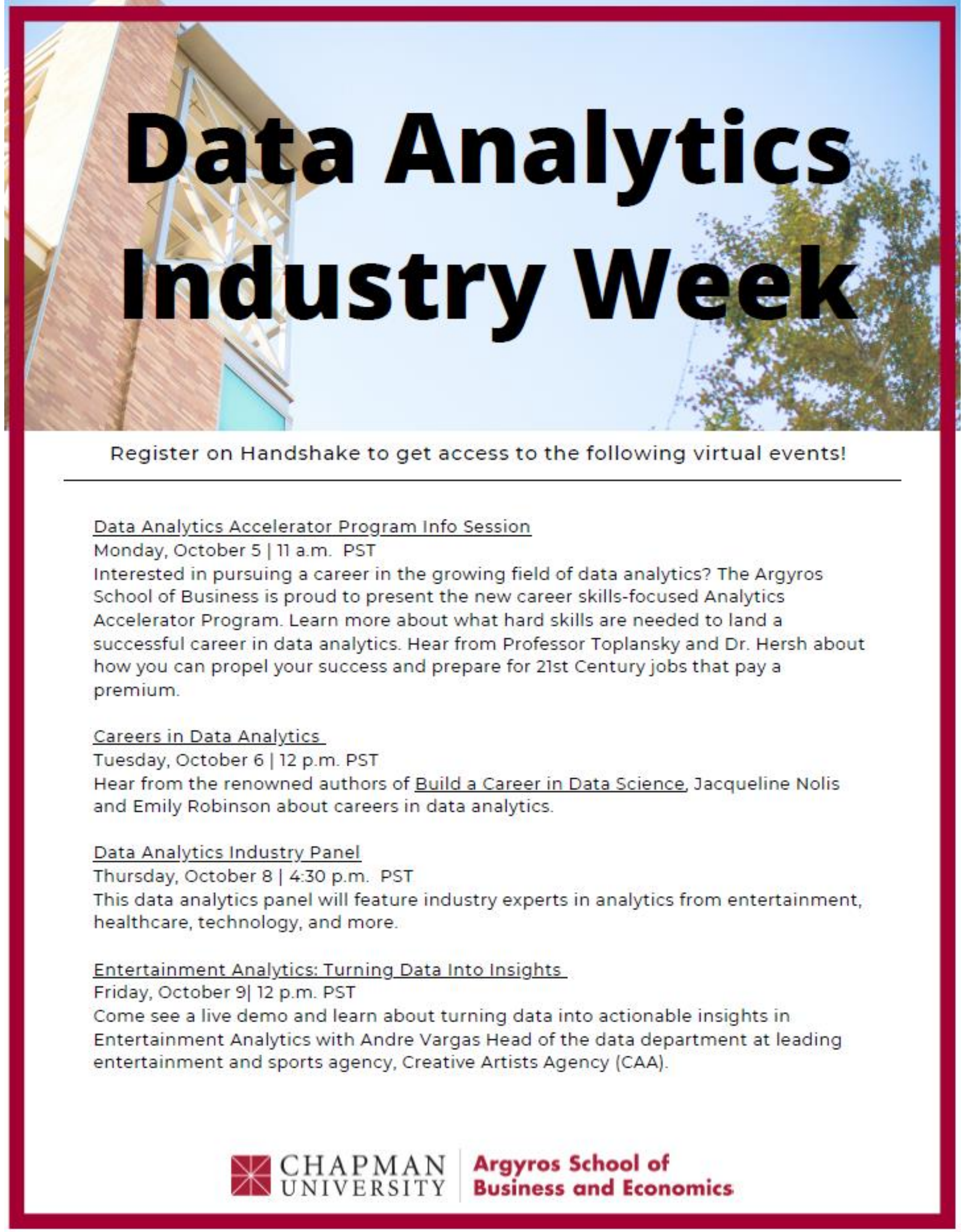
Class 11: Classification 3

MGSC 310

Prof. Jonathan Hersh

Class 11: Announcements

1. Quiz 3 posted tonight, due Thursday @ midnight
2. Problem Set 3 posted, Due Oct 13
   - Late problem sets docked 10% per day unless extenuating circumstances
3. Data Analytics Week!



Data Analytics Industry Week

Register on Handshake to get access to the following virtual events!

Data Analytics Accelerator Program Info Session
Monday, October 5 | 11 a.m. PST
Interested in pursuing a career in the growing field of data analytics? The Argyros School of Business is proud to present the new career skills-focused Analytics Accelerator Program. Learn more about what hard skills are needed to land a successful career in data analytics. Hear from Professor Toplansky and Dr. Hersh about how you can propel your success and prepare for 21st Century jobs that pay a premium.

Careers in Data Analytics
Tuesday, October 6 | 12 p.m. PST
Hear from the renowned authors of Build a Career in Data Science, Jacqueline Nolis and Emily Robinson about careers in data analytics.

Data Analytics Industry Panel
Thursday, October 8 | 4:30 p.m. PST
This data analytics panel will feature industry experts in analytics from entertainment, healthcare, technology, and more.

Entertainment Analytics: Turning Data Into Insights
Friday, October 9| 12 p.m. PST
Come see a live demo and learn about turning data into actionable insights in Entertainment Analytics with Andre Vargas Head of the data department at leading entertainment and sports agency, Creative Artists Agency (CAA).

CHAPMAN UNIVERSITY | Argyros School of Business and Economics

Class 11: Outline

1. Review:
   - False/True Positives

     False/True Negatives
   - Confusion Matrices
2. ROC Curves and AUC
3. Classification Lab
4. Calibration Plots
5. Severe Class Imbalance
   1. Up-sampling
   2. Down-sampling

Remember: Logisitic Model Gives Probabilities For Class Predictions We Must Choose Threshold

Outcome

$$y_i = \begin{cases} 0 \\ 1 \end{cases}$$

Here we are using an arbitrary cutoff of 0.5

Data:

1. **Training:** Train model to estimate parameters (betas)
2. **Scoring:** Use parameters to estimate probabilities
3. **Class assignment**: Use threshold to estimate class

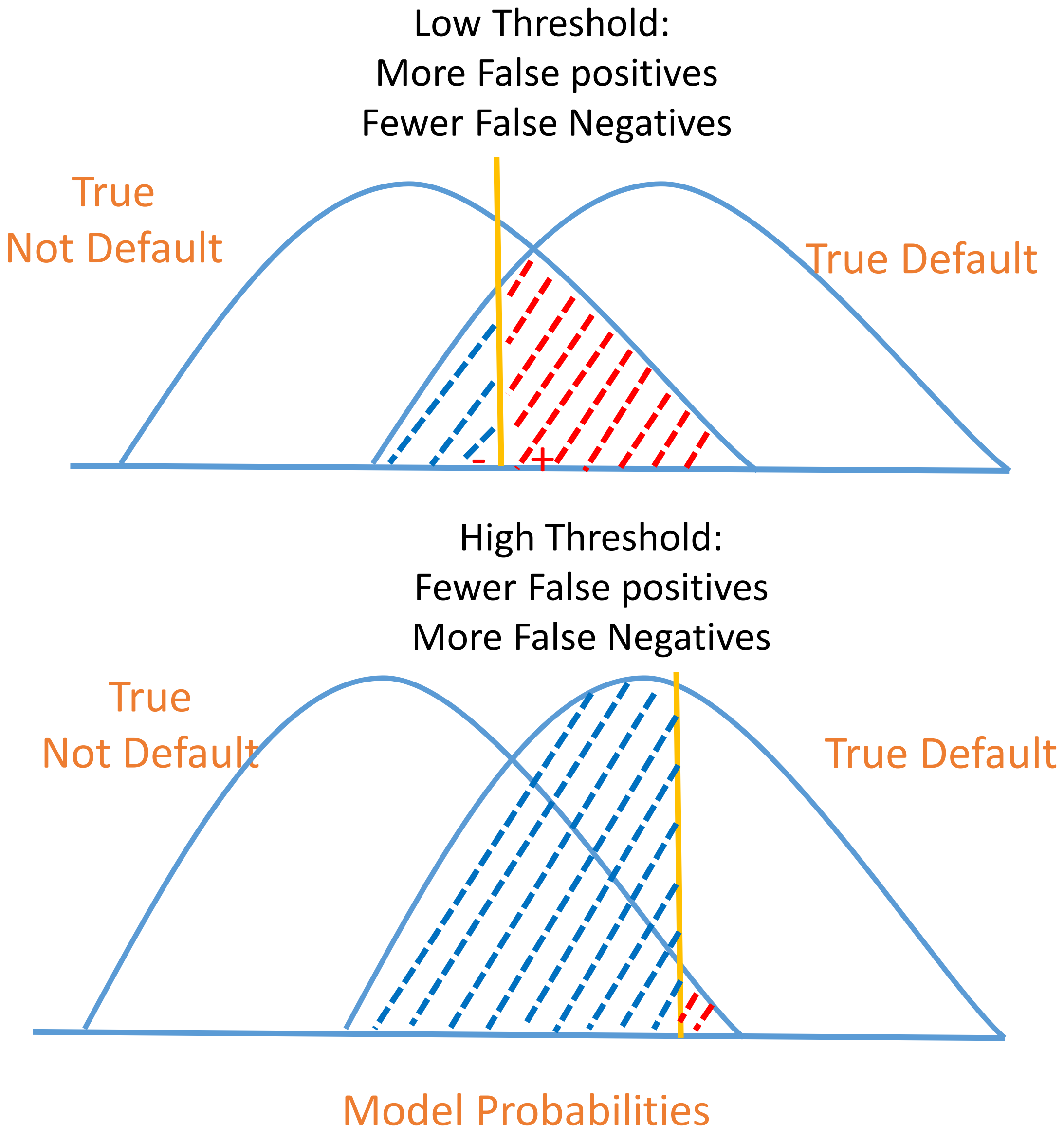
y	X_1	…	X_k	$\hat{p}$ "scores"	Class prediction (p cutoff > 0.5)
0	24		2000	0.05	0
0	12		4000	0.51	1
1	49		4999	0.55	1
0	33		100	0.01	0

Logit Model:

$$Pr(Y=1|X) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_k x_k}} = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x_1+\cdots+\hat{\beta}_k x_k}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 x_1+\cdots+\hat{\beta}_k x_k}}$$

$\hat{}$ means we estimated the parameter/trained the model using some data

Which Probability Threshold To Use?



- Threshold you choose should depend on relative costs of FPs and FN
  - e.g. screening at airport (cost of false neg high)
  - e.g. direct mail advertisement (cost of false positive low)
- Some common choices
  - **Maximize Accuracy** (equal weighting of FPs and FNs)
  - **Threshold p_hat** > 0.5
  - **Minimize cost**: TC = costFP *FPs + costFN * FNs

Sensitivity and Specificity, Confusion Matrix at P Cutoff > 0.5

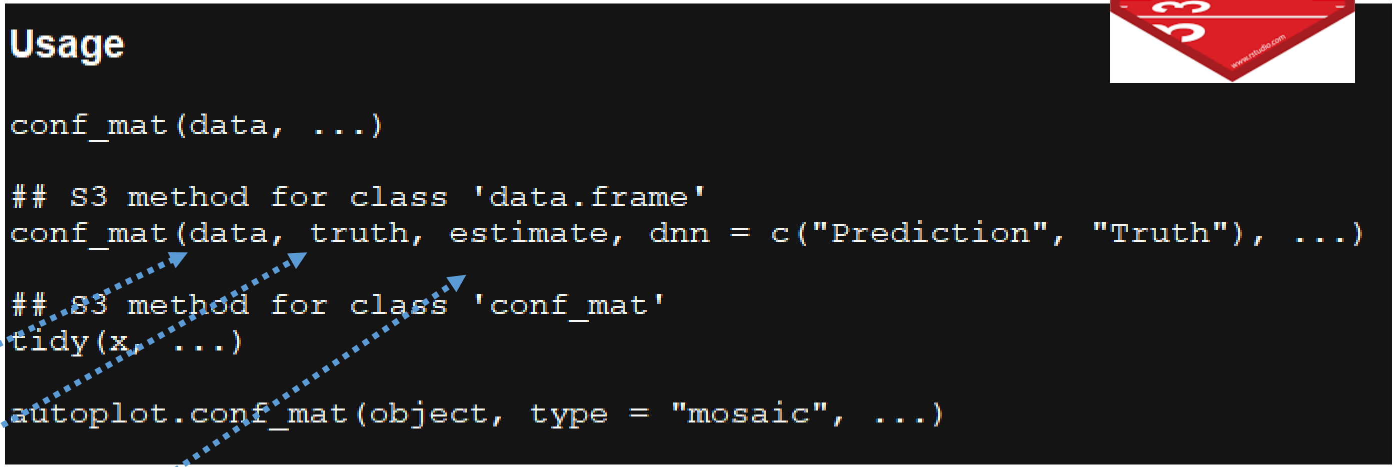
		True default status		
		No	**Yes**	
Predicted default status (cutoff p>0.5)	**No**	TN = 484	FN = 11	N^* = 495
	Yes	FP = 2	TP = 3	P^*= 5
		N = 486	P = 14	

- **Sensitivity:** True positive rate (aka 1 – power or recall)
  - TP/P = 3 / 14 = 21.4%
- **Specificity:** True negative rate
  - TN/N = 484 / 486= 99.5%

- **False positive rate** (aka Type I error, 1 - Specificity)
  - FP/N = 2/486= 0.004%

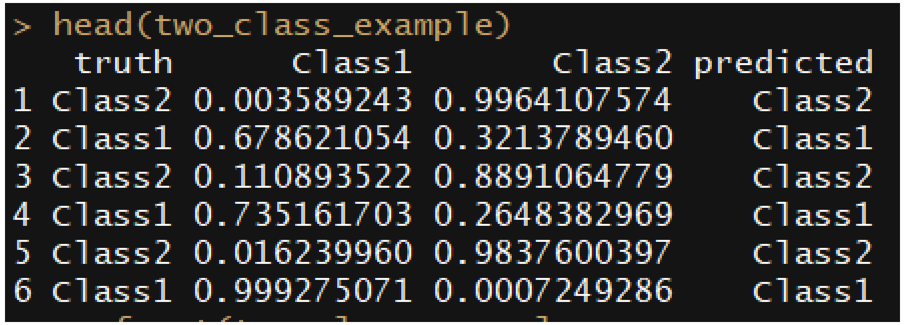
Generating Confusion Matrices in R

- To produce a confusion matrix in R we will use the yardstick package
- The function conf_mat() produces confusion matrices but we must format our data correctly
- We need to specify a data frame with
- Actual event (Y = 1) values
- Our estimated probabilities (scores)
- This example data frame shows how we need to structure our results data frame



Usage

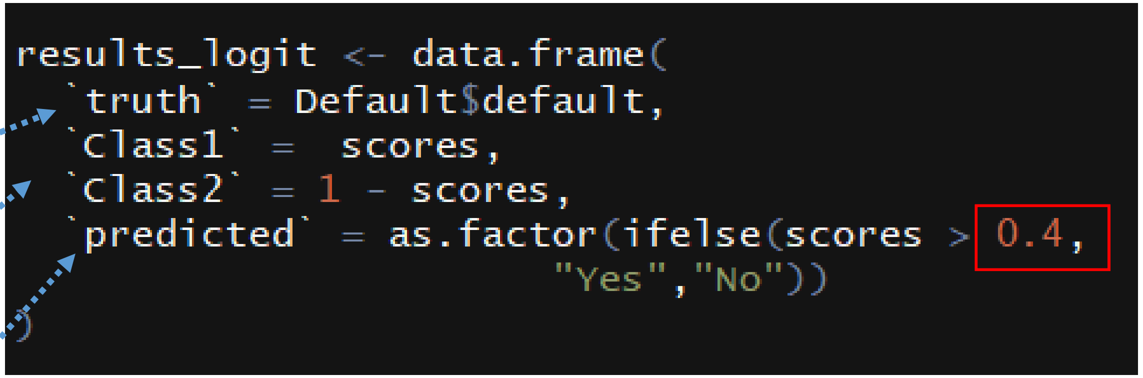
```
conf_mat(data, ...)

## S3 method for class 'data.frame'
conf_mat(data, truth, estimate, dnn = c("Prediction", "Truth"), ...)

## S3 method for class 'conf_mat'
tidy(x, ...)

autoplot.conf_mat(object, type = "mosaic", ...)
```



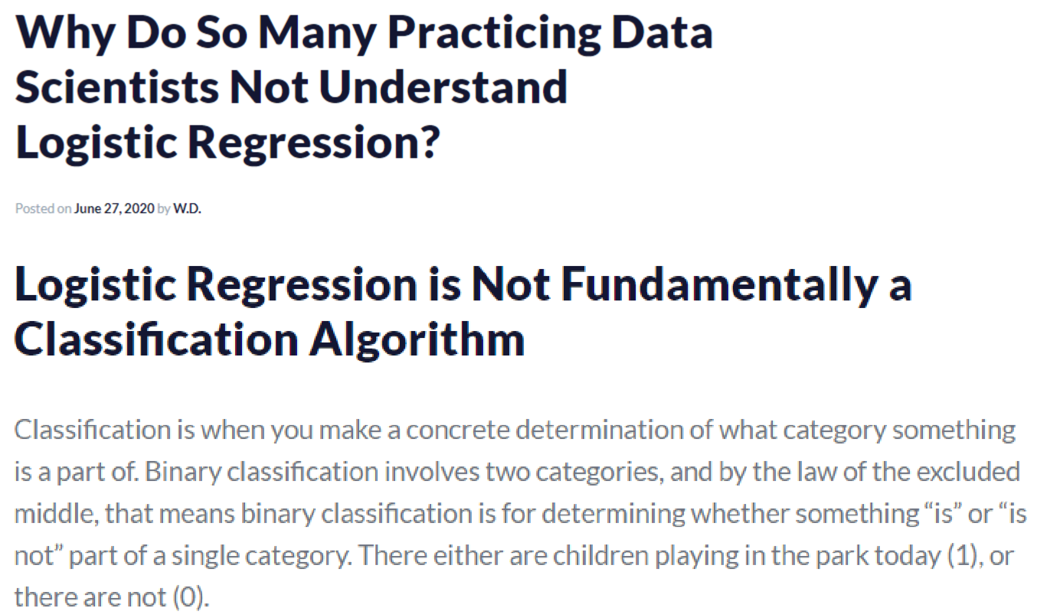
```
> head(two_class_example)
   truth      Class1       Class2 predicted
1 Class2 0.003589243 0.9964107574    Class2
2 Class1 0.678621054 0.3213789460    Class1
3 Class2 0.110893522 0.8891064779    Class2
4 Class1 0.735161703 0.2648382969    Class1
5 Class2 0.016239960 0.9837600397    Class2
6 Class1 0.999275071 0.0007249286    Class1
```

Formatting Results Matrix for Confusion Matrix

- Let’s store the model results in a data frame
- We must specify the actual default behavior
- And the probability of class1 (default) as well as probability of class2 (not default)
- We *must* specify a cutoff above which probabilities are classified as “class1” (or having the event) and below which they are not
- The cutoff probability is determined by the relative cost of false positives and false negatives! Do not use rules of thumb!



```
results_logit <- data.frame(
  `truth` = Default$default,
  `Class1` =  scores,
  `Class2` = 1 - scores,
  `predicted` = as.factor(ifelse(scores > 0.4,
                          "Yes","No"))
)
```



Why Do So Many Practicing Data Scientists Not Understand Logistic Regression?

Posted on June 27, 2020 by W.D.

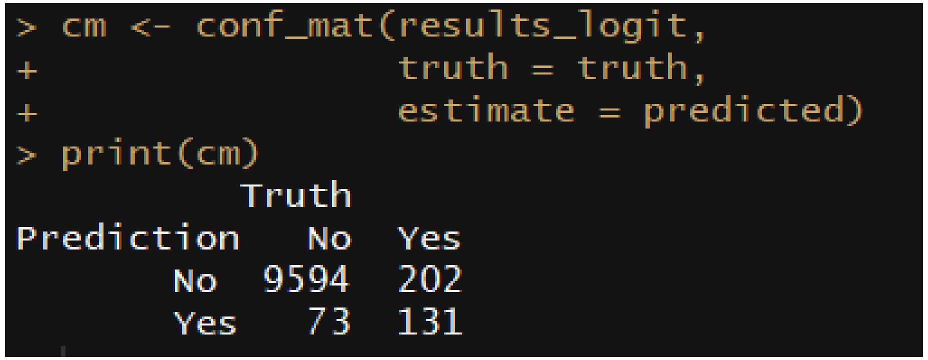
Logistic Regression is Not Fundamentally a Classification Algorithm

Classification is when you make a concrete determination of what category something is a part of. Binary classification involves two categories, and by the law of the excluded middle, that means binary classification is for determining whether something “is” or “is not” part of a single category. There either are children playing in the park today (1), or there are not (0).

https://ryxcommar.com/2020/06/27/why-do-so-many-practicing-data-scientists-not-understand-logistic-regression/

Producing Confusion Matrix Using Formatted Results Data

- The conf_mat() function shows the confusion matrix
- If we summarize the conf_mat() object we see more binary metrics of classification (don’t need to know all of these)
- **Sensitivity** is the true positive rate (TP/P) and here we identify of the true positives = 131/333 = 39.3%
- We may need to lower our threshold of cutoff probability

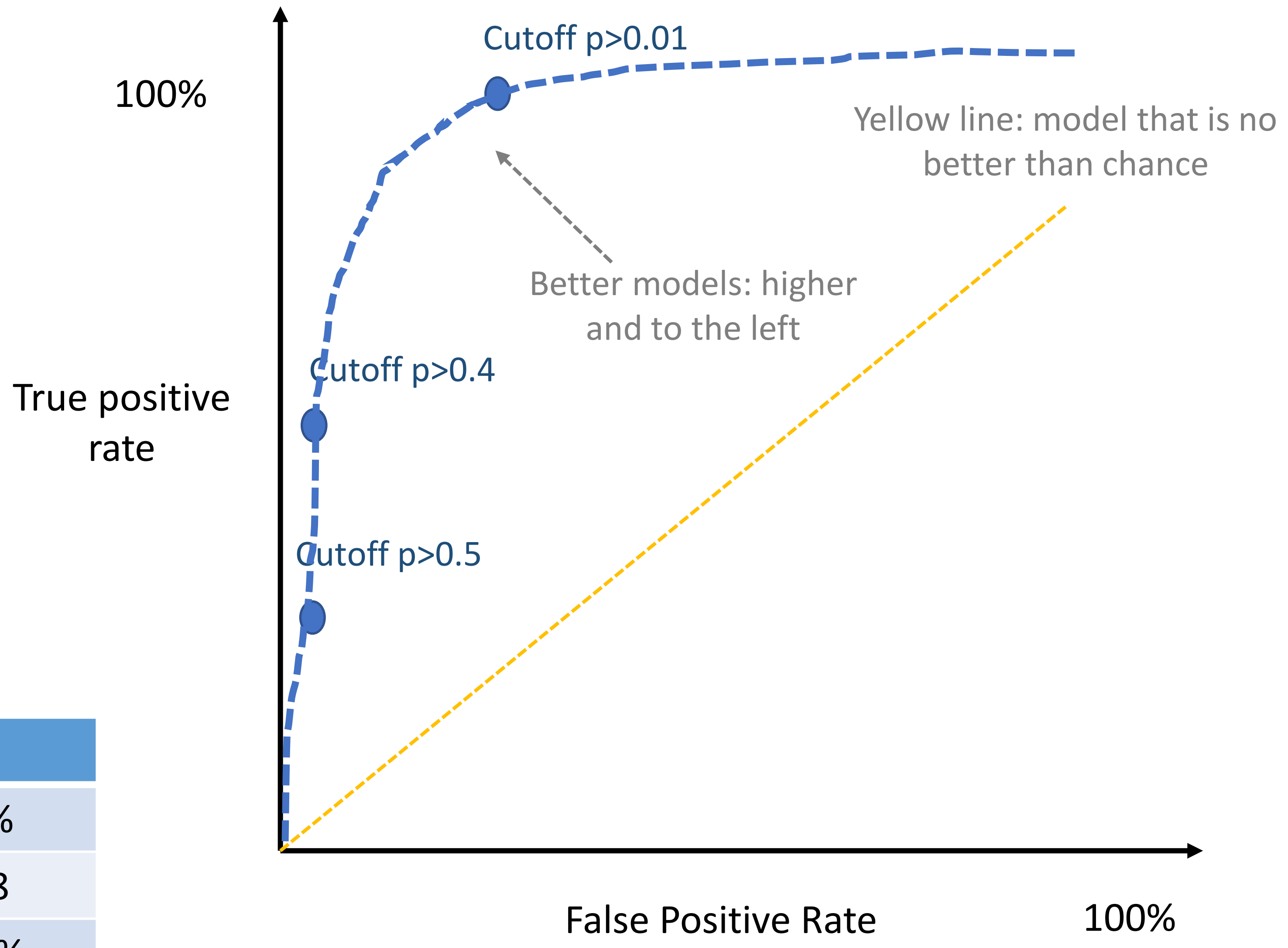


```
> cm <- conf_mat(results_logit,
+                truth = truth,
+                estimate = predicted)
> print(cm)
          Truth
Prediction   No  Yes
       No  9594  202
       Yes   73  131
```

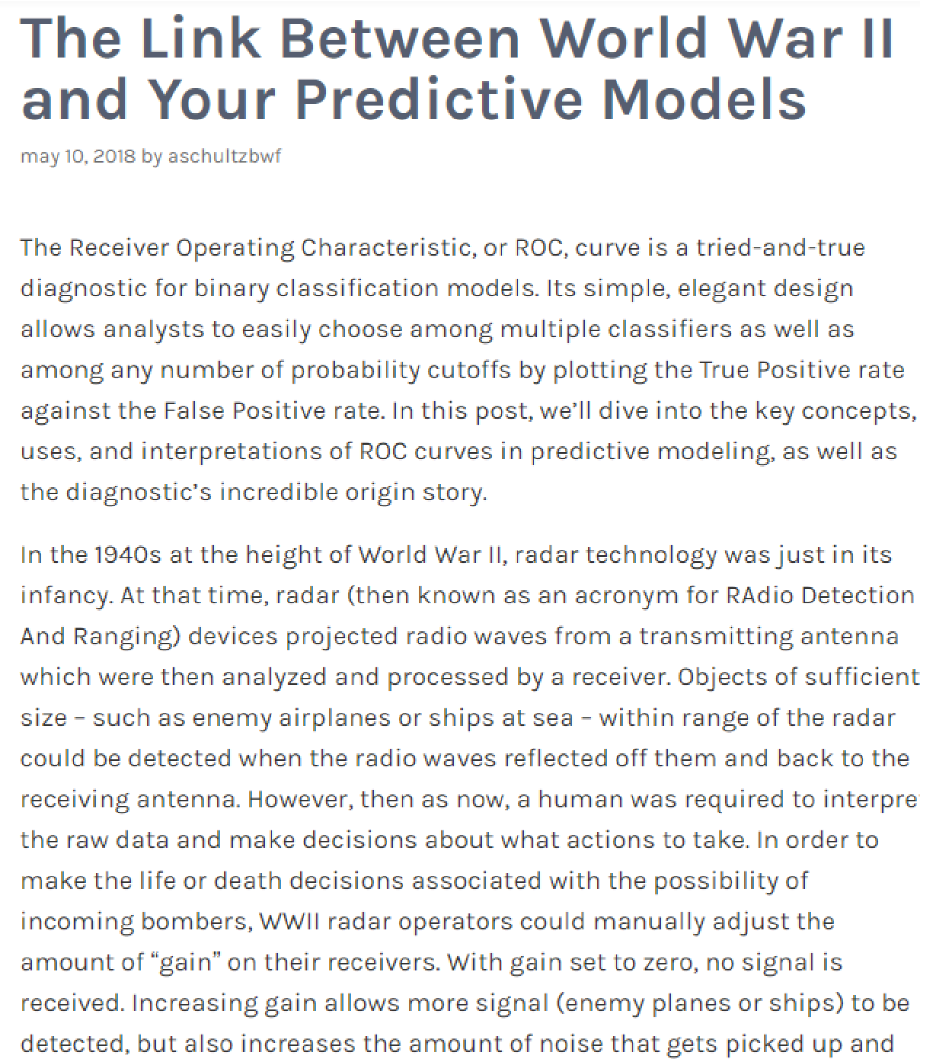
Continuous Cutoff: ROC Curve

- Can we show consequences of FPs and FNs as we vary the cutoff probability to assign classes?
- That is a ROC (Receiver Operator Curve) plot

Cutoff	TPR	FPR
0.01	100%	22.6%
0.4	57%	0.008
0.5	21.4%	0.004%
0.6	21.4%	0.002%



ROC Curves Tracked Airplanes on Radar During WWII



The Link Between World War II and Your Predictive Models

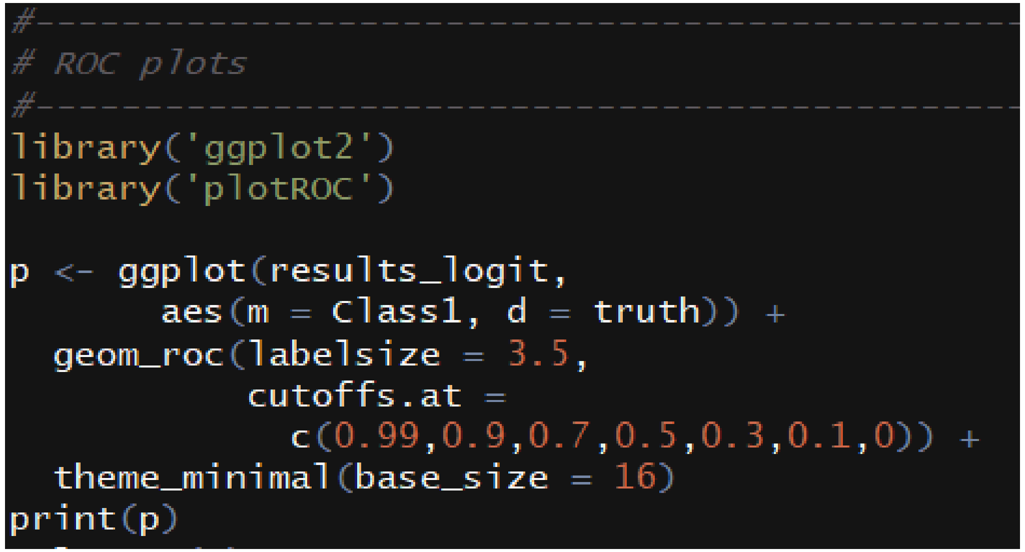
may 10, 2018 by aschultzbwf

The Receiver Operating Characteristic, or ROC, curve is a tried-and-true diagnostic for binary classification models. Its simple, elegant design allows analysts to easily choose among multiple classifiers as well as among any number of probability cutoffs by plotting the True Positive rate against the False Positive rate. In this post, we'll dive into the key concepts, uses, and interpretations of ROC curves in predictive modeling, as well as the diagnostic's incredible origin story.

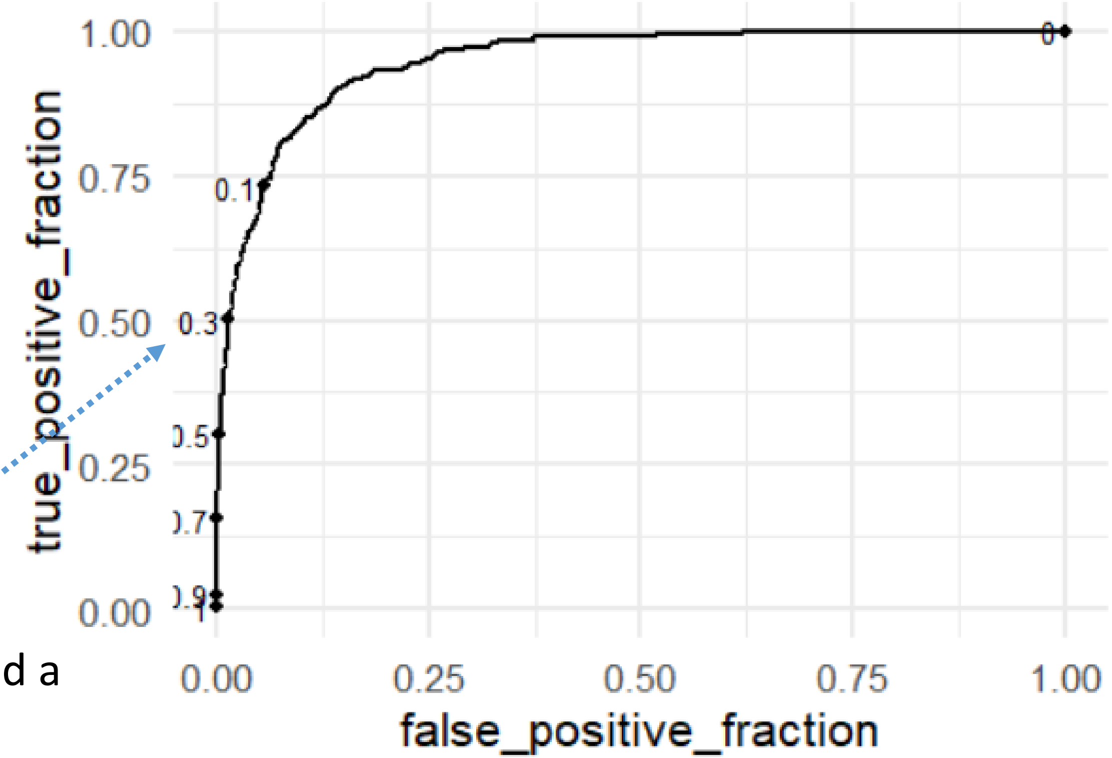
In the 1940s at the height of World War II, radar technology was just in its infancy. At that time, radar (then known as an acronym for RAdio Detection And Ranging) devices projected radio waves from a transmitting antenna which were then analyzed and processed by a receiver. Objects of sufficient size – such as enemy airplanes or ships at sea – within range of the radar could be detected when the radio waves reflected off them and back to the receiving antenna. However, then as now, a human was required to interpre the raw data and make decisions about what actions to take. In order to make the life or death decisions associated with the possibility of incoming bombers, WWII radar operators could manually adjust the amount of "gain" on their receivers. With gain set to zero, no signal is received. Increasing gain allows more signal (enemy planes or ships) to be detected, but also increases the amount of noise that gets picked up and

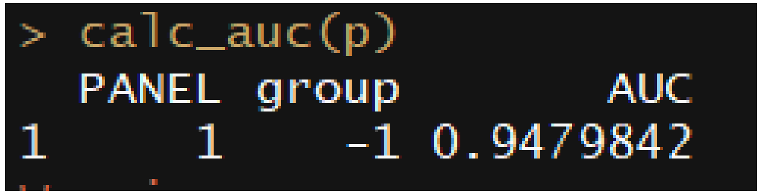
https://bwfinsight.blog/2018/05/10/the-link-between-world-war-ii-and-your-predictive-models/

ROC Curves in R



```
#-------------------------------------------------------------
# ROC plots
#-------------------------------------------------------------
library('ggplot2')
library('plotROC')

p <- ggplot(results_logit,
       aes(m = Class1, d = truth)) +
  geom_roc(labelsize = 3.5,
           cutoffs.at =
             c(0.99,0.9,0.7,0.5,0.3,0.1,0)) +
  theme_minimal(base_size = 16)
print(p)
```

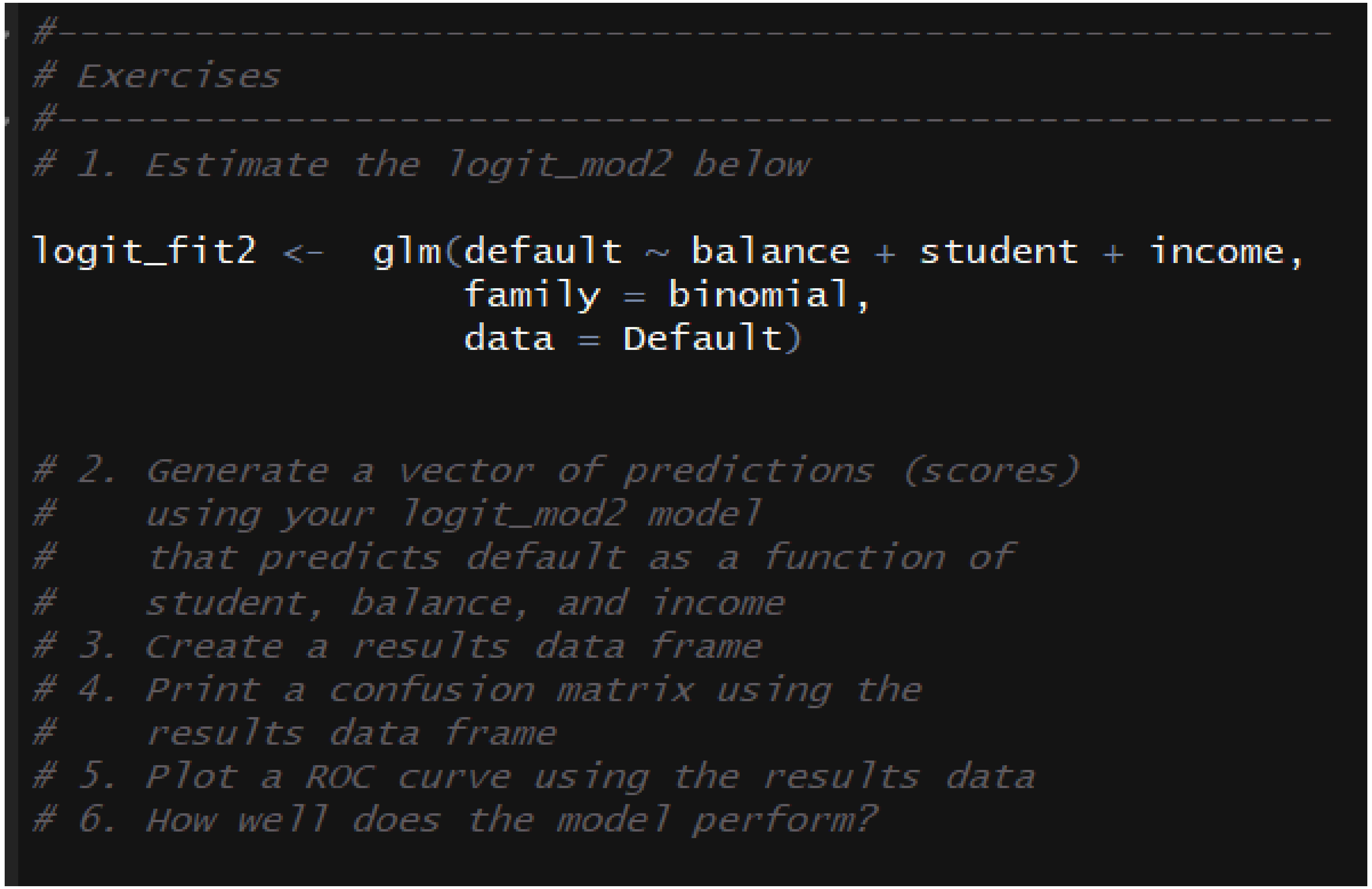




```
> calc_auc(p)
  PANEL group        AUC
1     1    -1 0.9479842
```

- At a cutoff of 0.3, we get a true positive fraction of 0.5 and a false positive fraction of a very low number
- Better models lie up and to the left in the ROC plot
- AUC calculates how much total area is under a particular curve
- AUC of 0.947 is pretty good

Lab Time!



```
#-----------------------------------------------------------
# Exercises
#-----------------------------------------------------------
# 1. Estimate the logit_mod2 below

logit_fit2 <-  glm(default ~ balance + student + income,
                   family = binomial,
                   data = Default)



# 2. Generate a vector of predictions (scores)
#    using your logit_mod2 model
#    that predicts default as a function of
#    student, balance, and income
# 3. Create a results data frame
# 4. Print a confusion matrix using the
#    results data frame
# 5. Plot a ROC curve using the results data
# 6. How well does the model perform?
```

Class 11: Outline

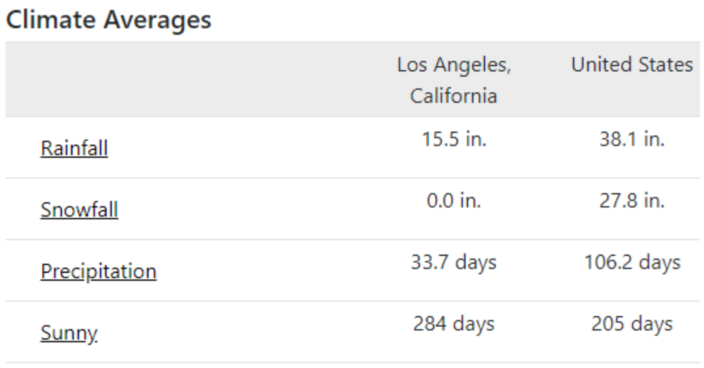
1. Review:
   - False/True Positives

     False/True Negatives
   - Confusion Matrices
2. ROC Curves and AUC
3. Classification Lab
4. Calibration Plots
5. Severe Class Imbalance
   1. Up-sampling
   2. Down-sampling

Chuck Kloserman on Probability

> “Life is chock full of lies, but the biggest lie is math. That’s particularly clear in the discipline of probability, a field od study that's completely and wholly fake. When push comes to shove - when you truly get down to the core essence of existence - there is only one mathematical possibility: Everything is 50-50. Either something will happen or it won’t”
>
> *- Chuck Kloserman, being wrong about probability*

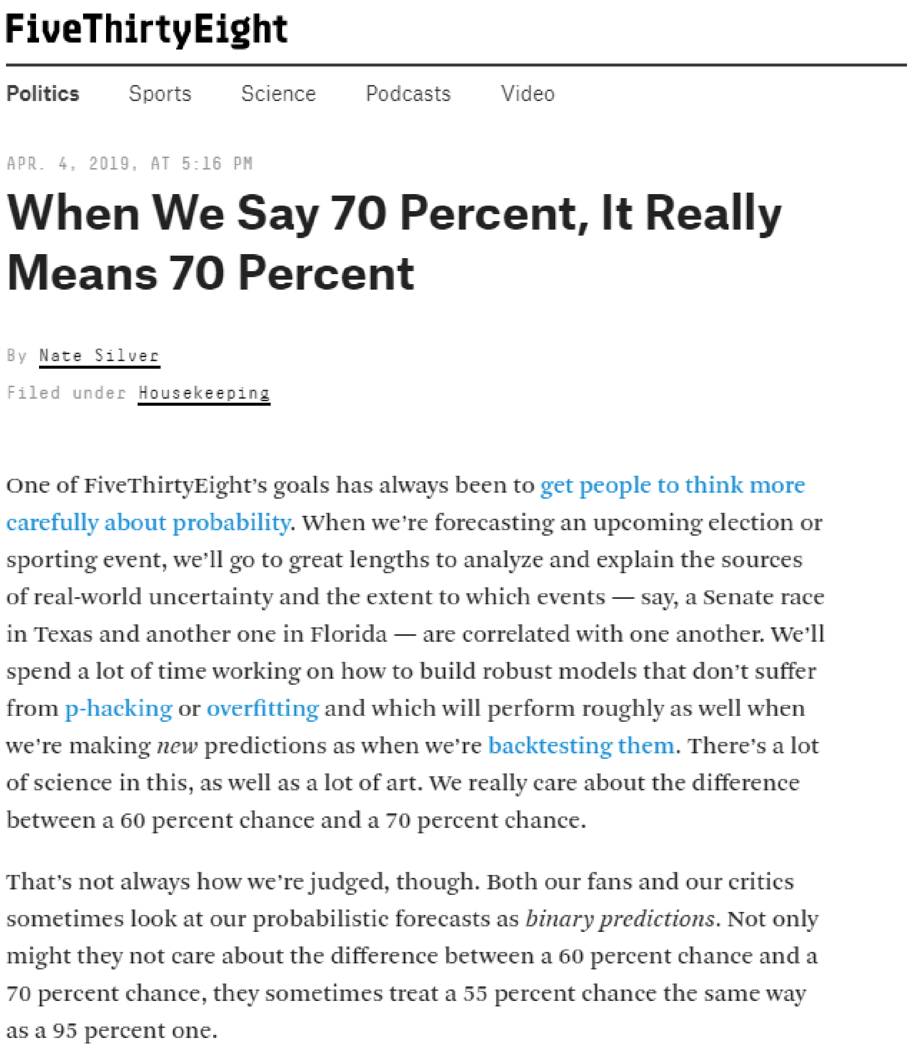
- This is an incorrect understanding of probability. Obviously something can only happen or not happen
- Ex-post (after the fact) we can say odds were 50-50. But ex-ante, odds are clearly not 50-50
- There is not a 50% chance it snows tomorrow in LA. How do we know? History tells us.



Climate Averages

	Los Angeles, California	United States
Rainfall	15.5 in.	38.1 in.
Snowfall	0.0 in.	27.8 in.
Precipitation	33.7 days	106.2 days
Sunny	284 days	205 days

So What Does it Mean to Predict Binary Events Well?



FiveThirtyEight

Politics Sports Science Podcasts Video

APR. 4, 2019, AT 5:16 PM

When We Say 70 Percent, It Really Means 70 Percent

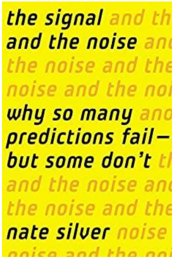
By Nate Silver

Filed under Housekeeping

One of FiveThirtyEight's goals has always been to get people to think more carefully about probability. When we're forecasting an upcoming election or sporting event, we'll go to great lengths to analyze and explain the sources of real-world uncertainty and the extent to which events — say, a Senate race in Texas and another one in Florida — are correlated with one another. We'll spend a lot of time working on how to build robust models that don't suffer from p-hacking or overfitting and which will perform roughly as well when we're making *new* predictions as when we're backtesting them. There's a lot of science in this, as well as a lot of art. We really care about the difference between a 60 percent chance and a 70 percent chance.

That's not always how we're judged, though. Both our fans and our critics sometimes look at our probabilistic forecasts as *binary predictions*. Not only might they not care about the difference between a 60 percent chance and a 70 percent chance, they sometimes treat a 55 percent chance the same way as a 95 percent one.

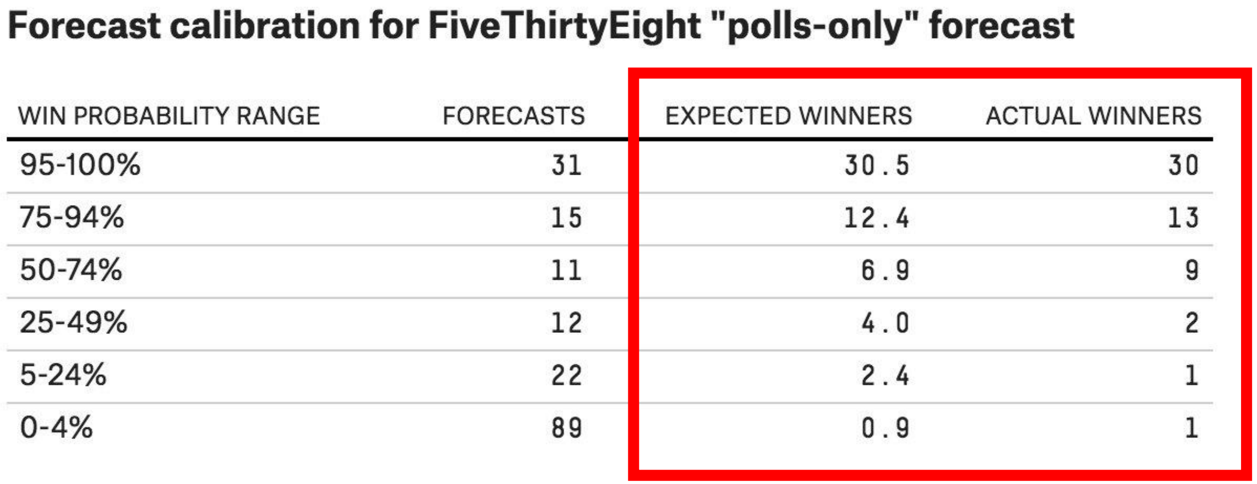
- One person you should read if you care about prediction is Nate Silver



Calibration measures whether, over the long run, events occur about as often as you say they're going to occur. For instance, of all the events that you forecast as having an 80 percent chance of happening, they should indeed occur about 80 out of 100 times; that's good calibration. If these events happen only 60 out of 100 times, you have problems — your forecasts aren't well-calibrated and are *overconfident*. **But it's just as bad if they occur 98 out of 100 times, in which case your forecasts are *underconfident*.**

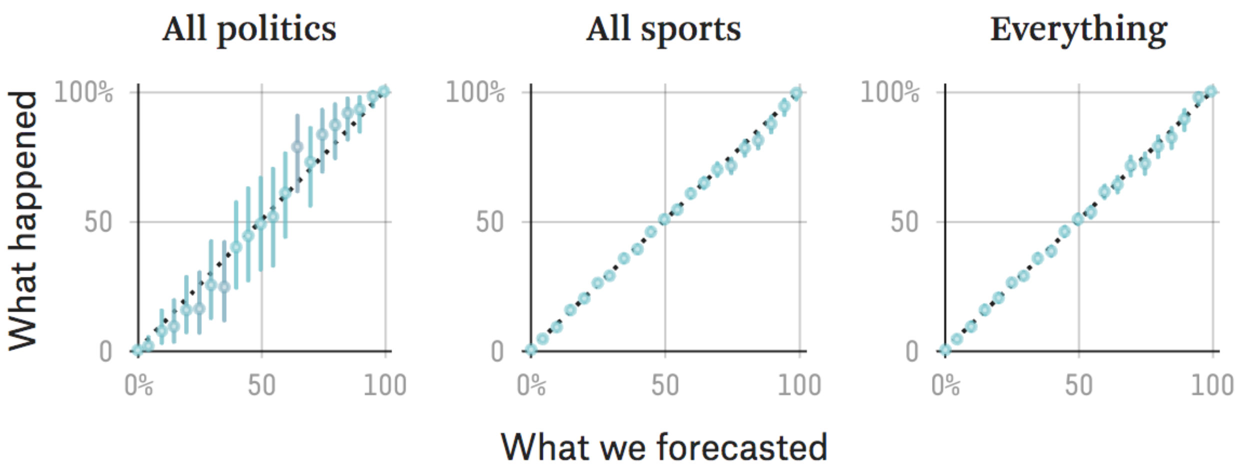
https://fivethirtyeight.com/features/when-we-say-70-percent-it-really-means-70-percent/

Are Nate Silver’s Forecasts Well Calibrated?



Forecast calibration for FiveThirtyEight "polls-only" forecast

WIN PROBABILITY RANGE	FORECASTS	EXPECTED WINNERS	ACTUAL WINNERS
95-100%	31	30.5	30
75-94%	15	12.4	13
50-74%	11	6.9	9
25-49%	12	4.0	2
5-24%	22	2.4	1
0-4%	89	0.9	1



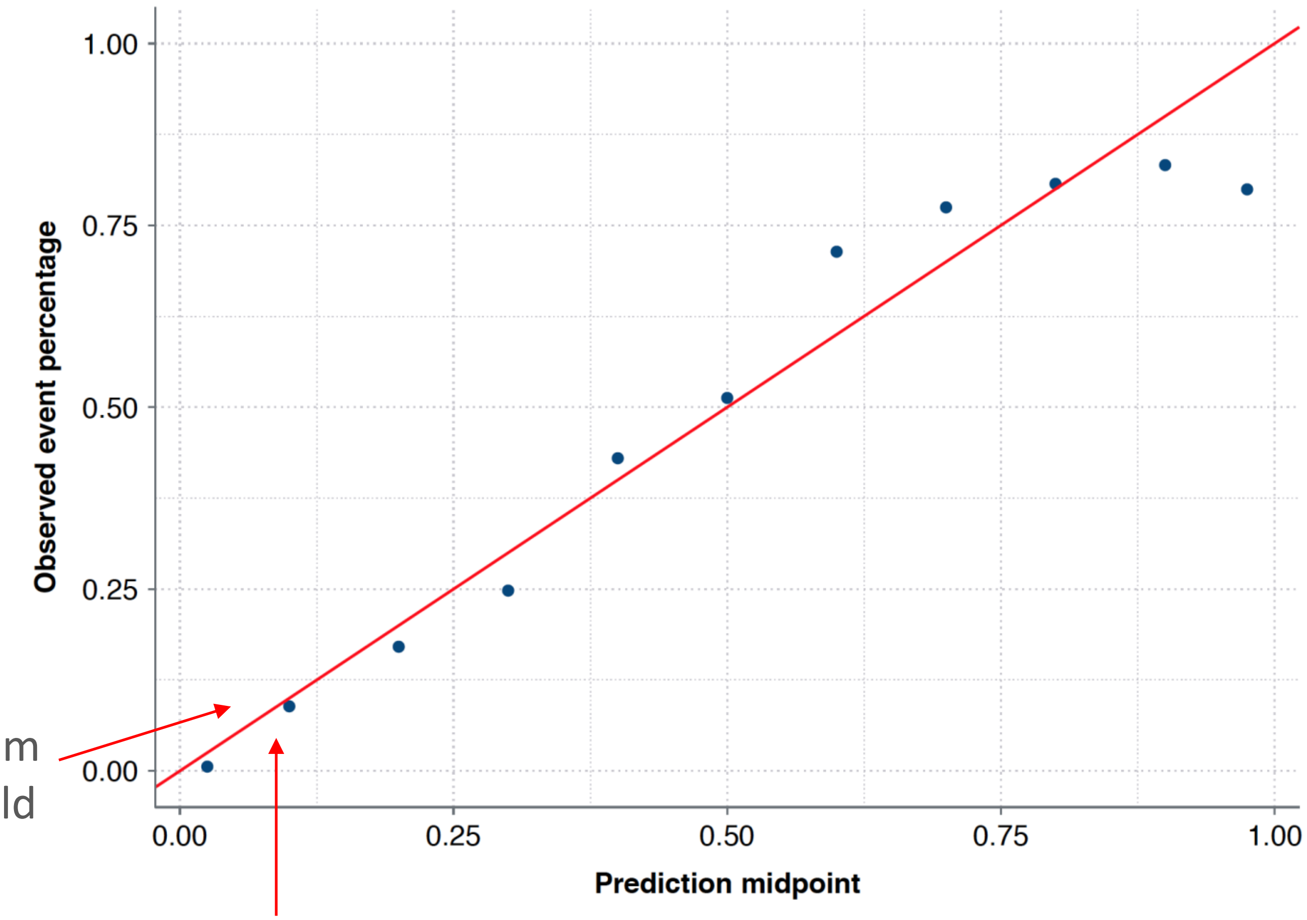
- 95% probability of win means out of 20 you forecast with 95% probability, 19 should win!
- 10% probability of win means out of 20 forecasts with 10% probability 2/20 should win!
- Overall Nate Silver is well calibrated, but politics is more noisy!

https://fivethirtyeight.com/features/when-we-say-70-percent-it-really-means-70-percent/

Calibration plot

- Idea of a calibration plot:

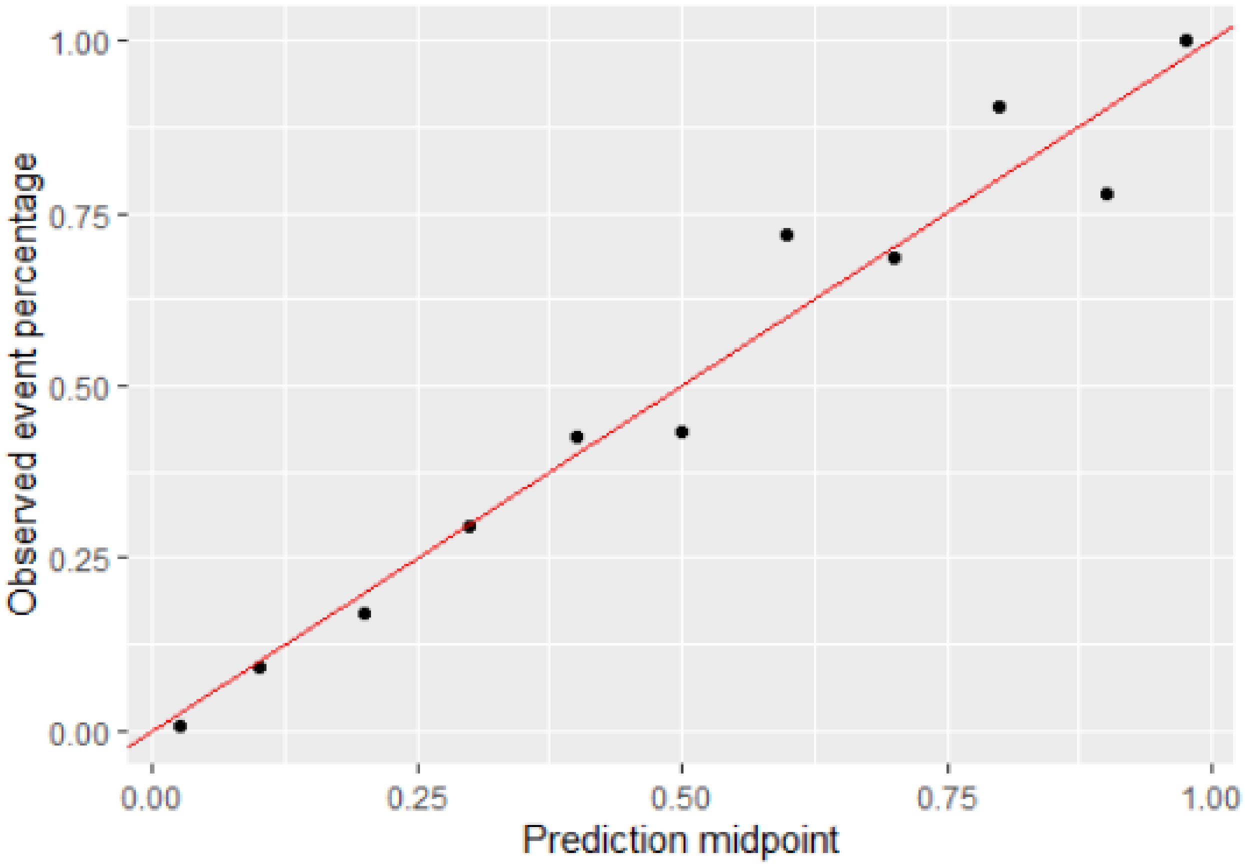
- A well-calibrated prediction is one where if we give an X% probability of an event occurring, X/100 of the time the event happens.

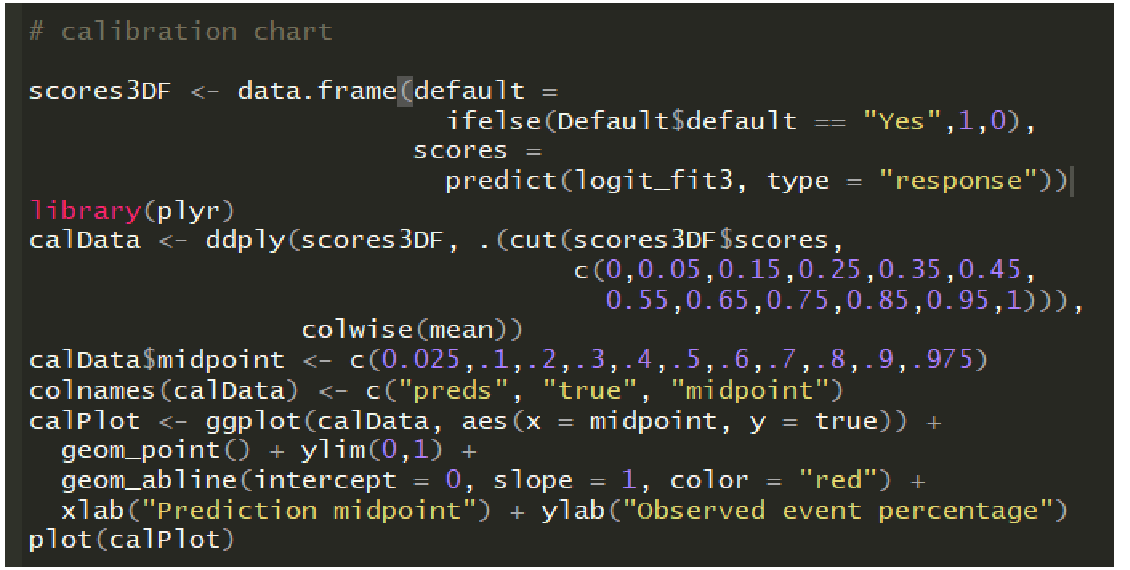


For 1/10 of them the event should actually occur

All obs where we predict 10% chance of event occurring

Calibration plot





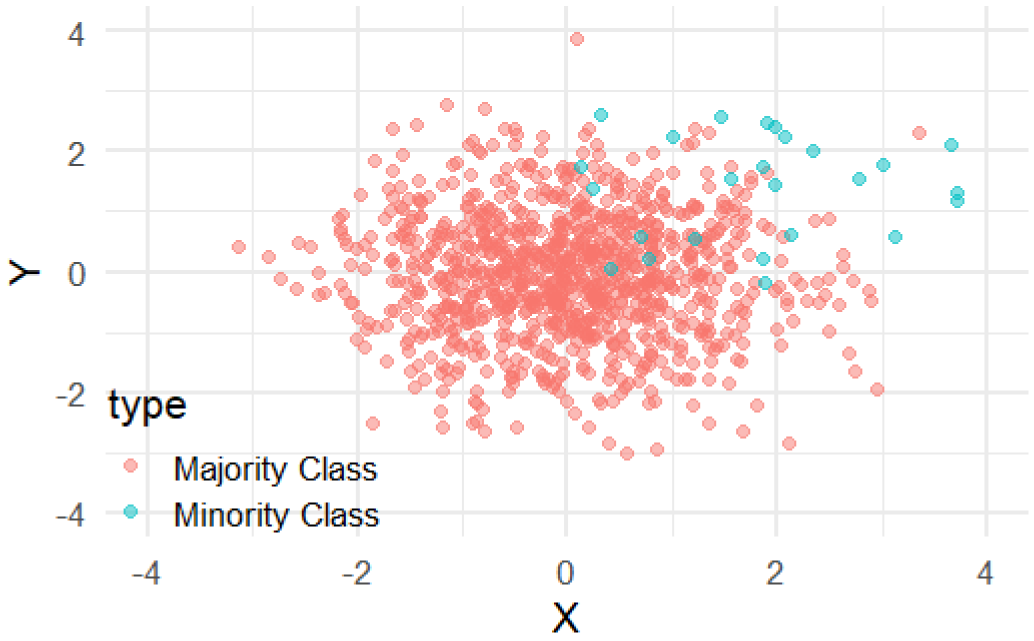
```
# calibration chart

scores3DF <- data.frame(default =
                            ifelse(Default$default == "Yes",1,0),
                          scores =
                            predict(logit_fit3, type = "response"))
library(plyr)
calData <- ddply(scores3DF, .(cut(scores3DF$scores,
                                     c(0,0.05,0.15,0.25,0.35,0.45,
                                       0.55,0.65,0.75,0.85,0.95,1))),
                 colwise(mean))
calData$midpoint <- c(0.025,.1,.2,.3,.4,.5,.6,.7,.8,.9,.975)
colnames(calData) <- c("preds", "true", "midpoint")
calPlot <- ggplot(calData, aes(x = midpoint, y = true)) +
  geom_point() + ylim(0,1) +
  geom_abline(intercept = 0, slope = 1, color = "red") +
  xlab("Prediction midpoint") + ylab("Observed event percentage")
plot(calPlot)
```

Class 11: Outline

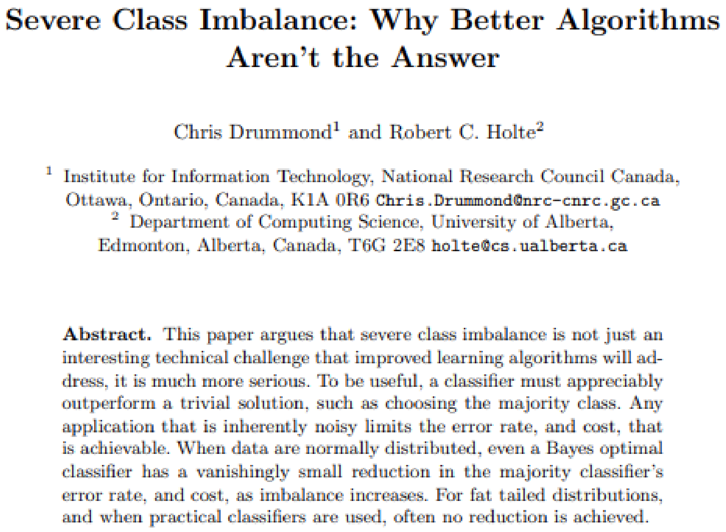
1. Review:
   - False/True Positives

     False/True Negatives
   - Confusion Matrices
2. ROC Curves and AUC
3. Classification Lab
4. Calibration Plots
5. **Severe Class Imbalance**
   1. Up-sampling
   2. Down-sampling

What Is Class Imbalace?



- Class Imbalance occurs with classification models where one class severely outnumbers the other class(es)
- What is significant imbalance? Anything less than ~10-5%
- This creates problems for **any** classification algorithm. Even AI isn't immune!



Severe Class Imbalance: Why Better Algorithms Aren't the Answer

Chris Drummond[1] and Robert C. Holte[2]

[1] Institute for Information Technology, National Research Council Canada,
Ottawa, Ontario, Canada, K1A 0R6 Chris.Drummond@nrc-cnrc.gc.ca
[2] Department of Computing Science, University of Alberta,
Edmonton, Alberta, Canada, T6G 2E8 holte@cs.ualberta.ca

Abstract. This paper argues that severe class imbalance is not just an interesting technical challenge that improved learning algorithms will address, it is much more serious. To be useful, a classifier must appreciably outperform a trivial solution, such as choosing the majority class. Any application that is inherently noisy limits the error rate, and cost, that is achievable. When data are normally distributed, even a Bayes optimal classifier has a vanishingly small reduction in the majority classifier's error rate, and cost, as imbalance increases. For fat tailed distributions, and when practical classifiers are used, often no reduction is achieved.

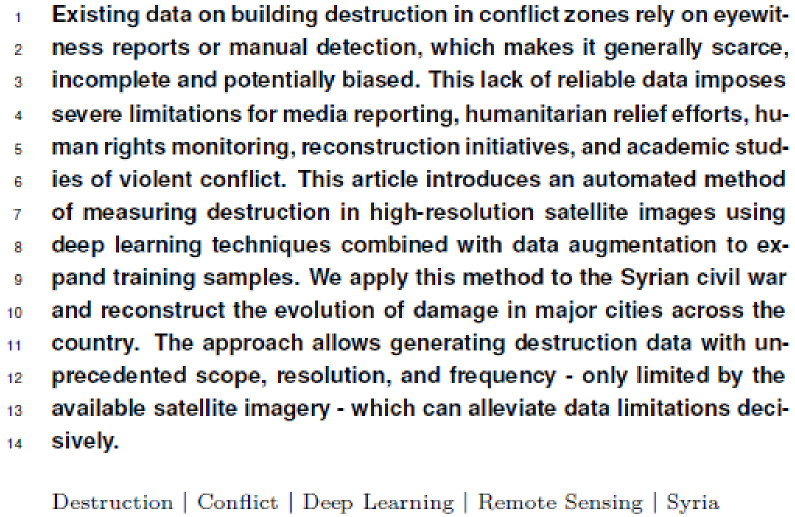
Monitoring War Destruction from Space: A Machine Learning Approach

Hannes Mueller[a,b,1], Andre Groeger[b,c,1], Jonathan Hersh[d,2], Andrea Matranga[d,e,2], and Joan Serrat[f,1]

[a]Institute of Economic Analysis (IAE-CSIC), 08193 Bellaterra, Barcelona, Spain; [b]Barcelona Graduate School of Economics (BGSE), 08005 Ba
Economics and Economic History, Universitat Autònoma de Barcelona (UAB), 08193 Bellaterra, Spain; [d]Argyros School of Business, Chapma
USA; [e]Smith Institute for Political Economy and Philosophy, Orange, CA 92868 USA; [f]Computer Science Department and Computer Vision C
Barcelona (UAB), 08193 Bellaterra, Spain

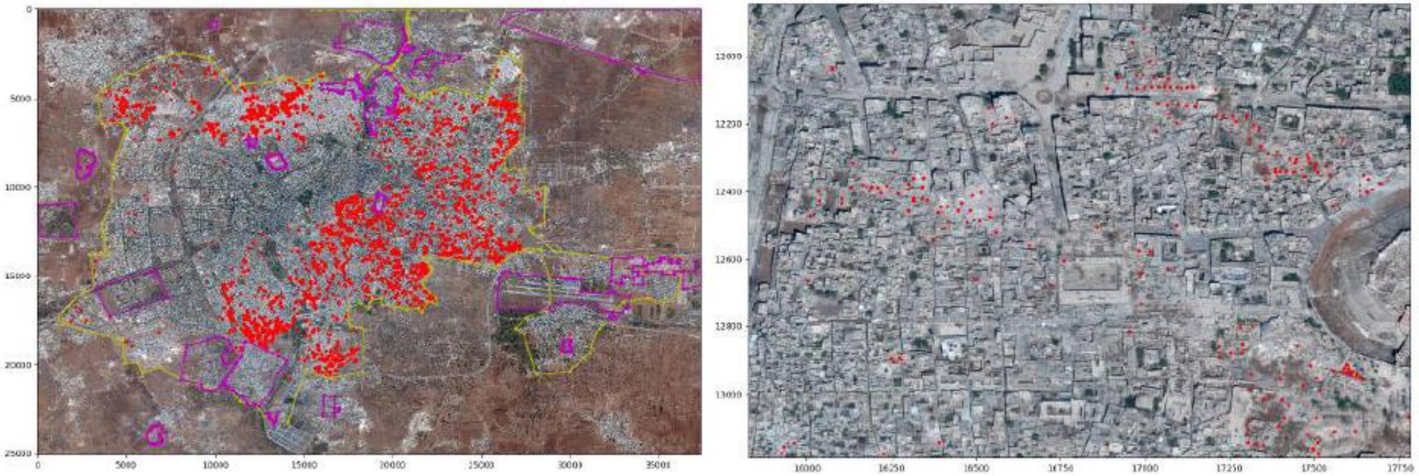
This manuscript was compiled on October 2, 2020

- My own work: finding which buildings have been bombed using machine learning applied to satellite imagery is hard because most buildings are not bombed, even in Syria!



Existing data on building destruction in conflict zones rely on eyewitness reports or manual detection, which makes it generally scarce, incomplete and potentially biased. This lack of reliable data imposes severe limitations for media reporting, humanitarian relief efforts, human rights monitoring, reconstruction initiatives, and academic studies of violent conflict. This article introduces an automated method of measuring destruction in high-resolution satellite images using deep learning techniques combined with data augmentation to expand training samples. We apply this method to the Syrian civil war and reconstruct the evolution of damage in major cities across the country. The approach allows generating destruction data with unprecedented scope, resolution, and frequency - only limited by the available satellite imagery - which can alleviate data limitations decisively.

Destruction | Conflict | Deep Learning | Remote Sensing | Syria

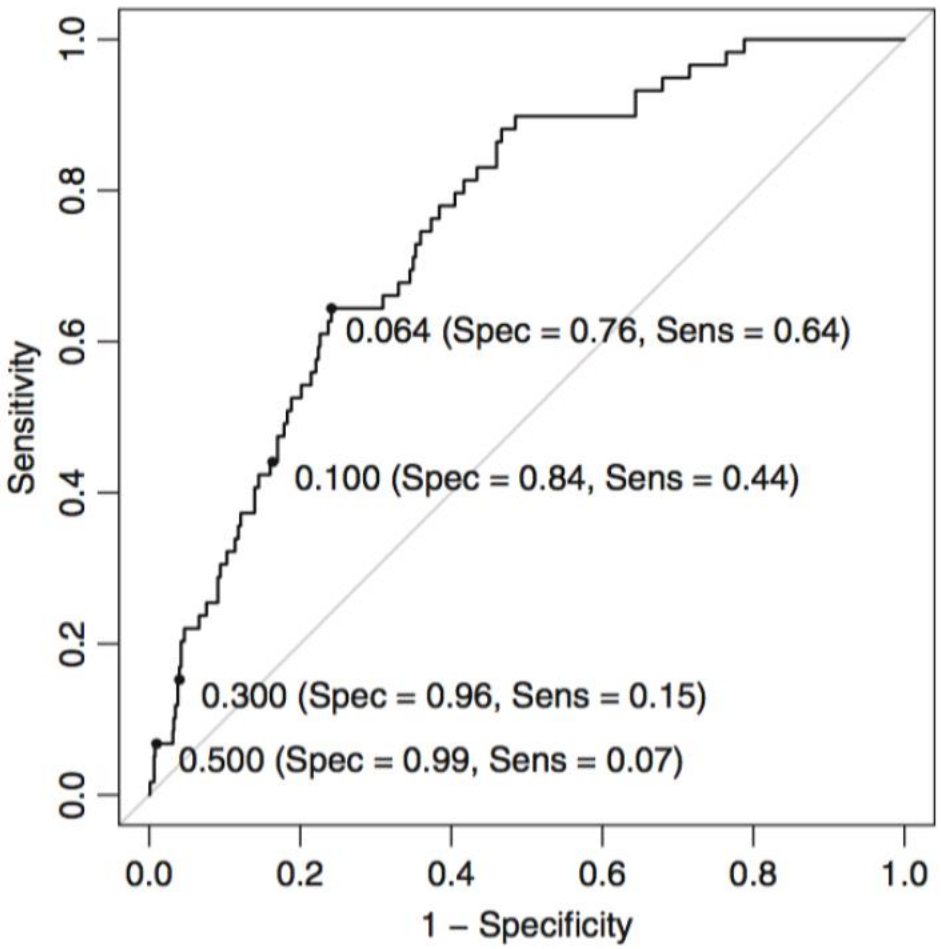


(a) Full Extent of Aleppo, Syria.

(b) Zoom Showing Destruction is Sparse

Fig. 1. Imagery of Aleppo 09/18/2016. Red dots indicate UNOSAT annotations as *destroyed*. The regions enclosed in magenta are *no analysis zones*, excluded from the UNOSAT damage assessment due to being non-civilian. The yellow lines indicates the boundary of populated areas in Aleppo under analysis. Sources: Google Earth/Maxar Technologies and UNITAR/UNOSAT.

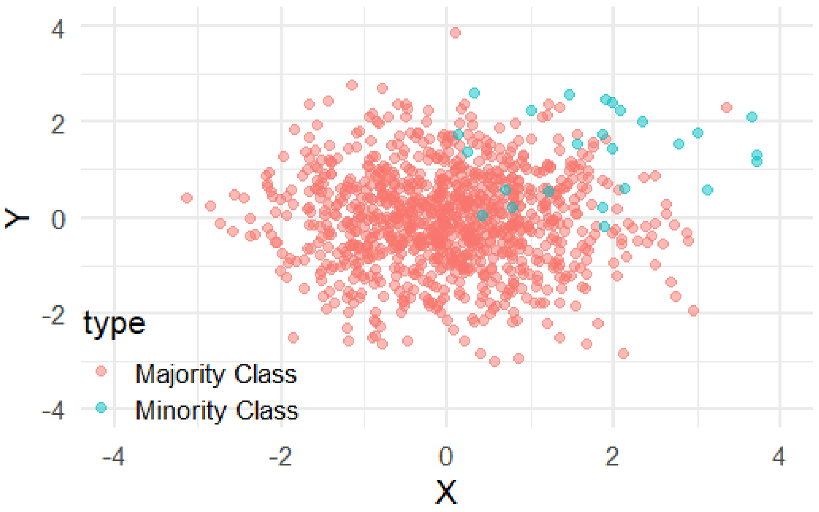
Remedy 1: Alternative Class Threshold



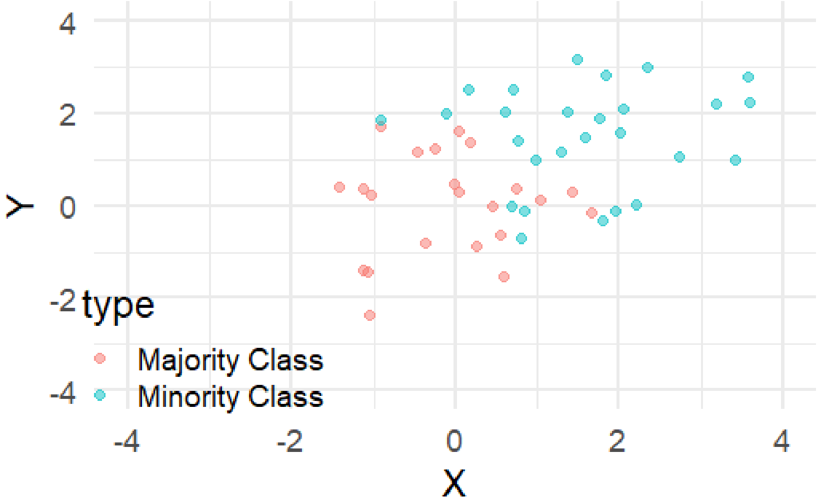
- First solution may be to choose a different probability threshold
- Each threshold from the ROC plot shows a different resulting specificity or sensitivity
- For applications we care more about sensitivity (low FNs), for others we care are specificity (low FPs)

Other Remedies: Up-Sampling and Down-Sampling

- Down-Sampling creates an artificial data set with equal minority and majority cases

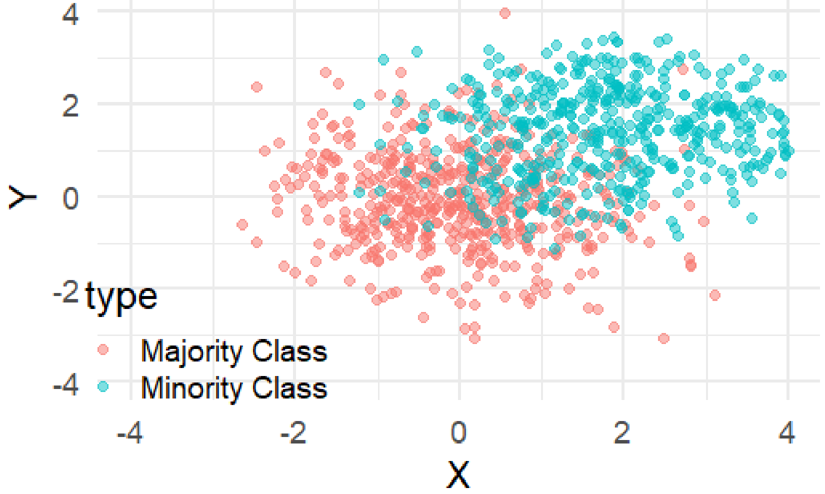


Down-Sampling

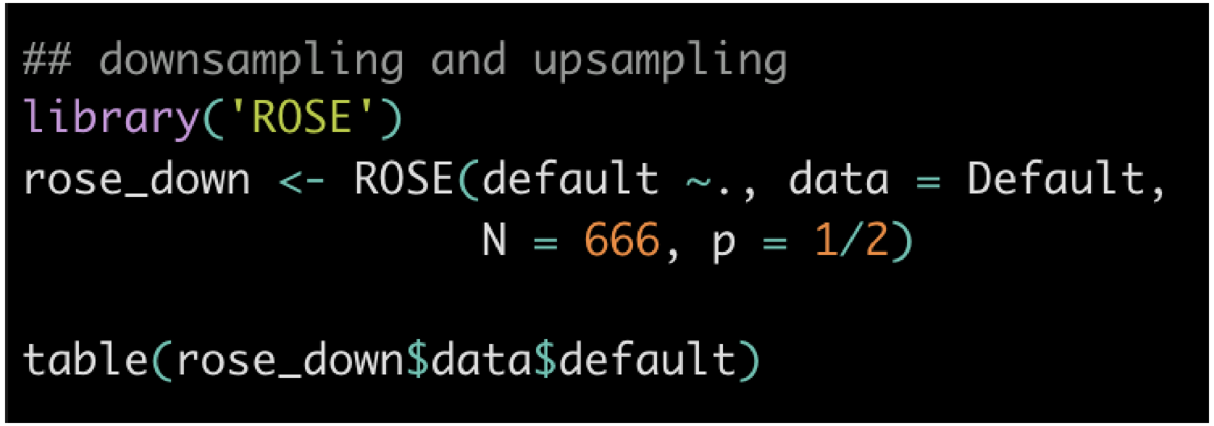


- Up-Sampling creates an artificial data set with more (repeated copies of) the minority class

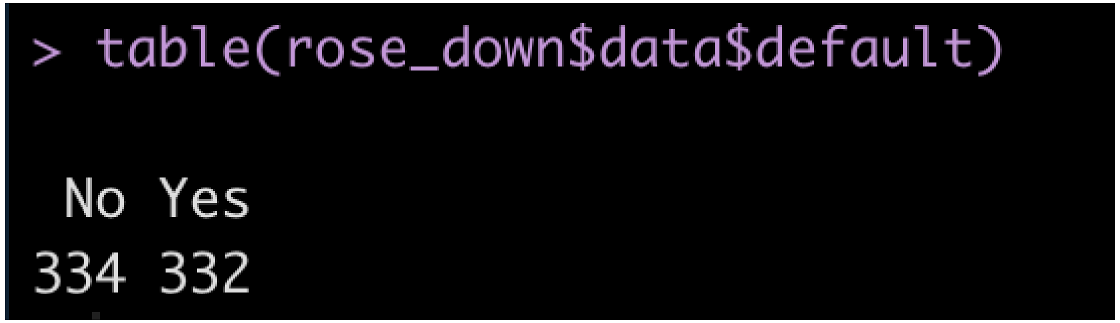
Up-Sampling



Down-Sampling using ROSE



```
## downsampling and upsampling
library('ROSE')
rose_down <- ROSE(default ~., data = Default,
                  N = 666, p = 1/2)

table(rose_down$data$default)
```

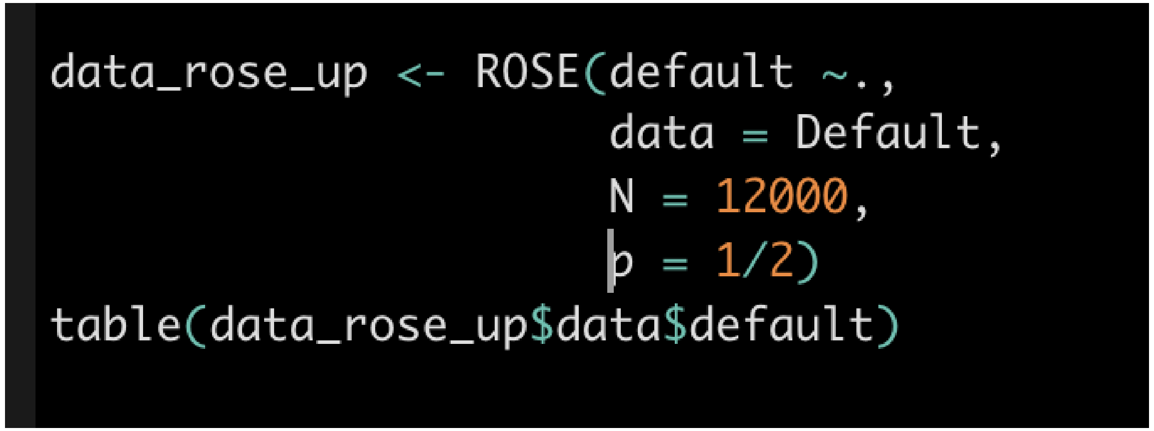


```
> table(rose_down$data$default)

 No Yes
334 332
```

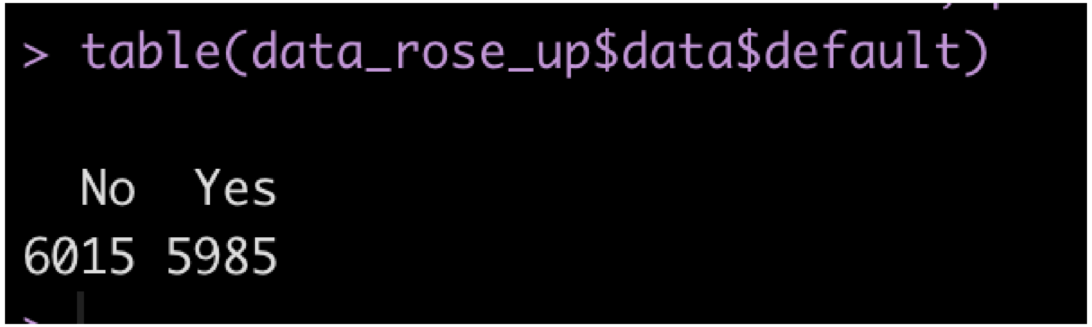
- N = size of dataset desired
- p = probability of event in desired dataset

- Set N = p*nrow(origina datset) to down sample

Up-Sampling using ROSE



```
data_rose_up <- ROSE(default ~.,
                     data = Default,
                     N = 12000,
                     p = 1/2)
table(data_rose_up$data$default)
```

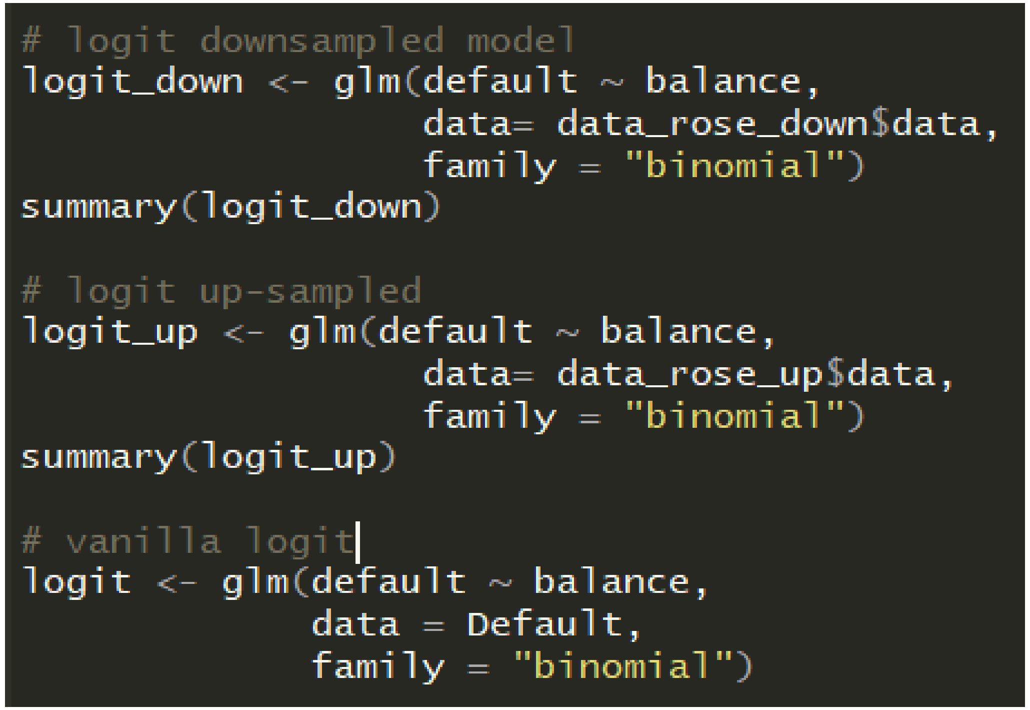


```
> table(data_rose_up$data$default)

  No  Yes
6015 5985
```

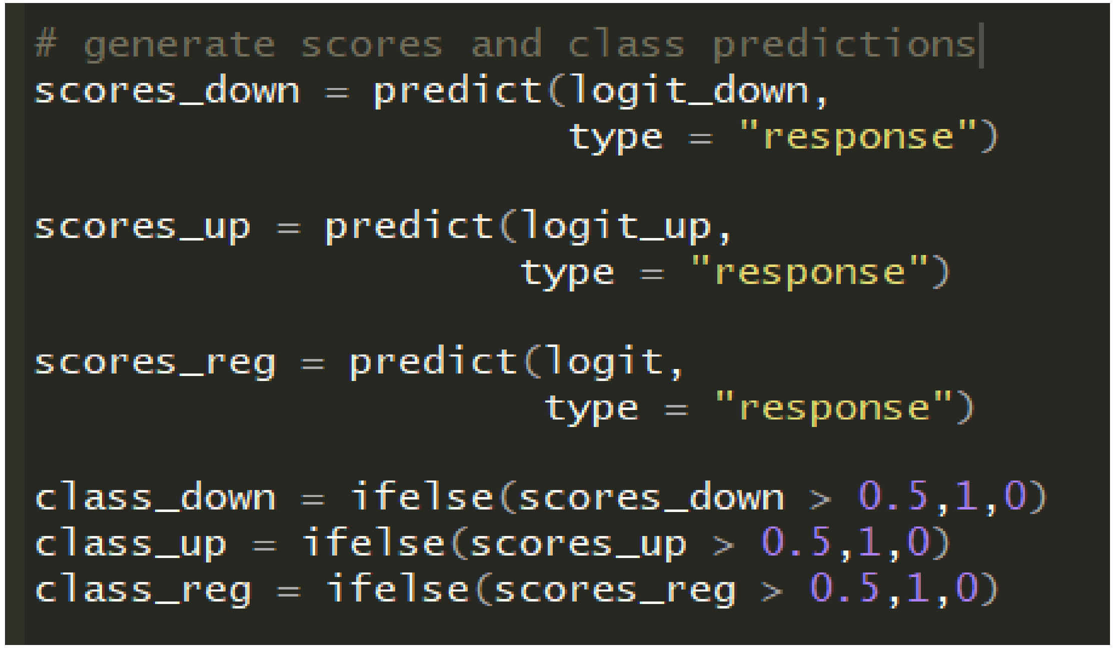
- N = size of dataset desired
- p = probability of event in desired dataset

- Set N > nrow(origina datset) to up sample

Fit Models on Upsampled/Downsampled Data



```
# logit downsampled model
logit_down <- glm(default ~ balance,
                  data= data_rose_down$data,
                  family = "binomial")
summary(logit_down)

# logit up-sampled
logit_up <- glm(default ~ balance,
                  data= data_rose_up$data,
                  family = "binomial")
summary(logit_up)

# vanilla logit
logit <- glm(default ~ balance,
             data = Default,
             family = "binomial")
```



```
# generate scores and class predictions
scores_down = predict(logit_down,
                      type = "response")

scores_up = predict(logit_up,
                    type = "response")

scores_reg = predict(logit,
                     type = "response")

class_down = ifelse(scores_down > 0.5,1,0)
class_up = ifelse(scores_up > 0.5,1,0)
class_reg = ifelse(scores_reg > 0.5,1,0)
```

Model Diagnostics on Up-Sampled/Down-Sampled Models

	Up-Sampled Logit	Down-Sampled Logit	Regular Logit
Accuracy	0.872	0.869	0.972
True Positives	307	5282	100
True Negs	274	5149	9625
Sensitivity	0.9	0.88	0.3
Specificity	0.843	0.859	0.996
False Pos Rate	0.157	0.141	0.004

Class 11 Summary

- Confusion matrices show the true/false positives/negatives.
- ROC plots measure the consequence on true positive fraction and false positive fraction for different cutoff probabilities
- Higher AUC scores mean a better ROC plot indicating a better model
- Calibration curves show the actual event % for each group of predicted probabilities
- Well-calibrated models interpret probability correctly. If we give an X% probability of something occurring, it will occur X/100 of the time.

- Severe class imbalance is when one class is dominant
- Better algorithms won’t help
- Down-sampling and up-sampling are two methods, choosing a different probability threshold is another. Many other methods ☺