Module 6: Linear Models for Classification

ISE-529

Content based on ISLR Chapter 4.1 – 4.3

Outline

- Classification – Basic Theory
  - Logistic regression basic theory
  - Parameter estimation
  - Confounding
  - Classification from logistic regression models
- Assessment
  - Misclassification Rate
  - Sensitivity/Specificity
  - Receiver Operating Characteristics (ROC) Charts and optimizing the classification threshold
  - Lift and Lift Curves

Data Mining Techniques Overview

Classification Analysis

The process of predicting the “class” (or label or group membership) of newly encountered entities based on analyzing known class membership of other similar types of entities.

- Belongs to the group of techniques referred to as “*supervised*” learning
- If classes are binary (pass/fail, purchase/no purchase, fraudulent/not fraudulent, it is referred to as “*binary classification*”

We will only be addressing binary classification in this module

Classification Examples

Credit Scoring

Can credit score and home ownership predict load default?

Predictor Variables:

- Credit Score: 300–850
- Home Ownership: Yes/No/Rent

⬇

Response Variable:

- Loan Default: Yes/No

Classification Examples

Biostatistics

Are alcohol and smoking related to heart disease?

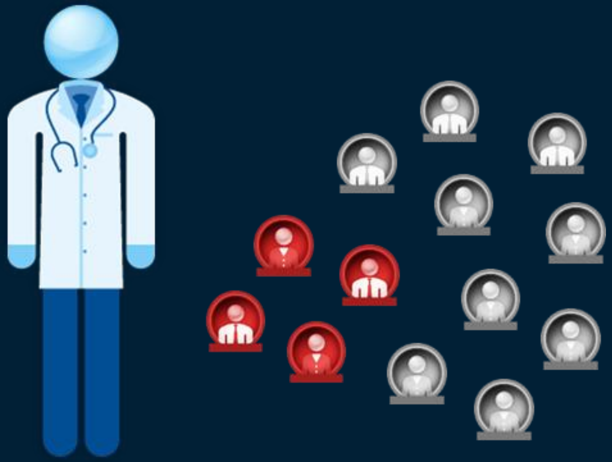
Predictor Variables:

- Alcohol: ounces per day
- Smoking: cigarettes per day

↓

Response Variable:

- Heart Disease: 1/0



Classification Examples

Campaign Marketing

Does a customer make a purchase based on past behavior?

Predictor Variables:

- Purchases: total spent in past 90 days
- Age Group: four levels
- Gender: M/F

Response Variable:

- Campaign: Yes/No

Classification

- Similar to regression, objective is to develop a mathematical model of the relationship of one or more **independent variables** (or **predictors** or **regressors** or **explanatory variables**) to one **dependent variable** (or **response variable**).
- However, in classification problems the **dependent variable** is a categorical attribute.
- Common special case is where the dependent variable takes one of two possible values (true/false, pass/fail, sick/healthy, etc.) – referred to as **binary classification**
- Generally, we are more interested in estimating the probabilities that the output belongs into each category level.
    - For example, it is more valuable to have an estimate of the probability that an insurance claim is fraudulent than just a classification of “Fraudulent”

Classification Example

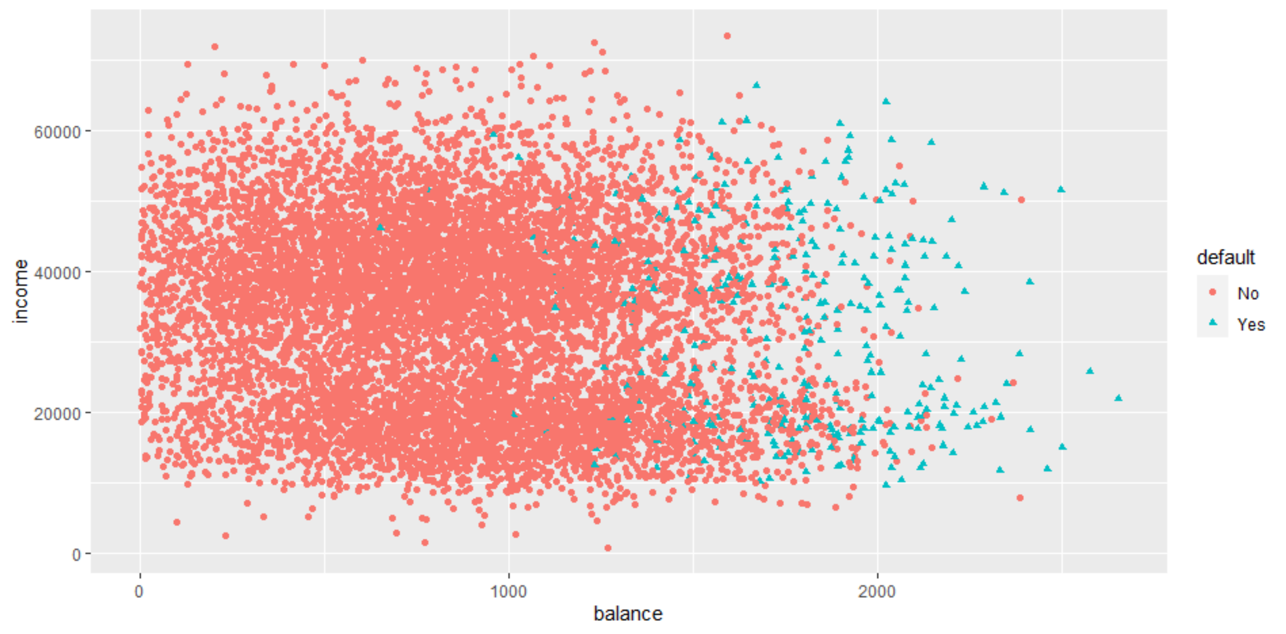
Credit Card Default

	A	B	C	D	E	F
1		default	student	balance	income	
2	1	No	No	$ 729.53	$ 44,361.63	
3	2	No	Yes	$ 817.18	$ 12,106.13	
4	3	No	No	$ 1,073.55	$ 31,767.14	
5	4	No	No	$ 529.25	$ 35,704.49	
6	5	No	No	$ 785.66	$ 38,463.50	
7	6	No	Yes	$ 919.59	$ 7,491.56	
8	7	No	No	$ 825.51	$ 24,905.23	
9	8	No	Yes	$ 808.67	$ 17,600.45	
10	9	No	No	$ 1,161.06	$ 37,468.53	
11	10	No	No	$ -	$ 29,275.27	
12	11	No	Yes	$ -	$ 21,871.07	
13	12	No	Yes	$ 1,220.58	$ 13,268.56	
14	13	No	No	$ 237.05	$ 28,251.70	
15	14	No	No	$ 606.74	$ 44,994.56	
16	15	No	No	$ 1,112.97	$ 23,810.17	
17	16	No	No	$ 286.23	$ 45,042.41	
18	17	No	No	$ -	$ 50,265.31	
19	18	No	Yes	$ 527.54	$ 17,636.54	
20	19	No	No	$ 485.94	$ 61,566.11	
21	20	No	No	$ 1,095.07	$ 26,464.63	
22	21	No	No	$ 228.95	$ 50,500.18	
23	22	No	No	$ 954.26	$ 32,457.51	
24	23	No	No	$ 1,055.96	$ 51,317.88	
25	24	No	No	$ 641.98	$ 30,466.10	
26	25	No	No	$ 773.21	$ 34,353.31	
27	26	No	No	$ 855.01	$ 25,211.33	
28	27	No	No	$ 643.00	$ 41,473.51	
29	28	No	No	$ 1,454.86	$ 32,189.09	
30	29	No	No	$ 615.70	$ 39,376.39	
31	30	No	Yes	$ 1,119.57	$ 16,556.07	
32	31	No	No	$ 494.82	$ 54,384.78	
33	32	No	Yes	$ 448.88	$ 15,799.47	
34	33	No	Yes	$ 584.90	$ 22,429.94	

Classification Example

Credit Card Default Data.csv

	default	student	balance	income
1	No	No	$729.53	$44,361.63
2	No	Yes	$817.18	$12,106.13
3	No	No	$1,073.55	$31,767.14
4	No	No	$529.25	$35,704.49
5	No	No	$785.66	$38,463.50
6	No	Yes	$919.59	$7,491.56
7	No	No	$825.51	$24,905.23
8	No	Yes	$808.67	$17,600.45
9	No	No	$1,161.06	$37,468.53
10	No	No	$-	$29,275.27
11	No	Yes	$-	$21,871.07
12	No	Yes	$1,220.58	$13,268.56
13	No	No	$237.05	$28,251.70
14	No	No	$606.74	$44,994.56
15	No	No	$1,112.97	$23,810.17
16	No	No	$286.23	$45,042.41
17	No	No	$-	$50,265.31
18	No	Yes	$527.54	$17,636.54
19	No	No	$485.94	$61,566.11
20	No	No	$1,095.07	$26,464.63
21	No	No	$228.95	$50,500.18

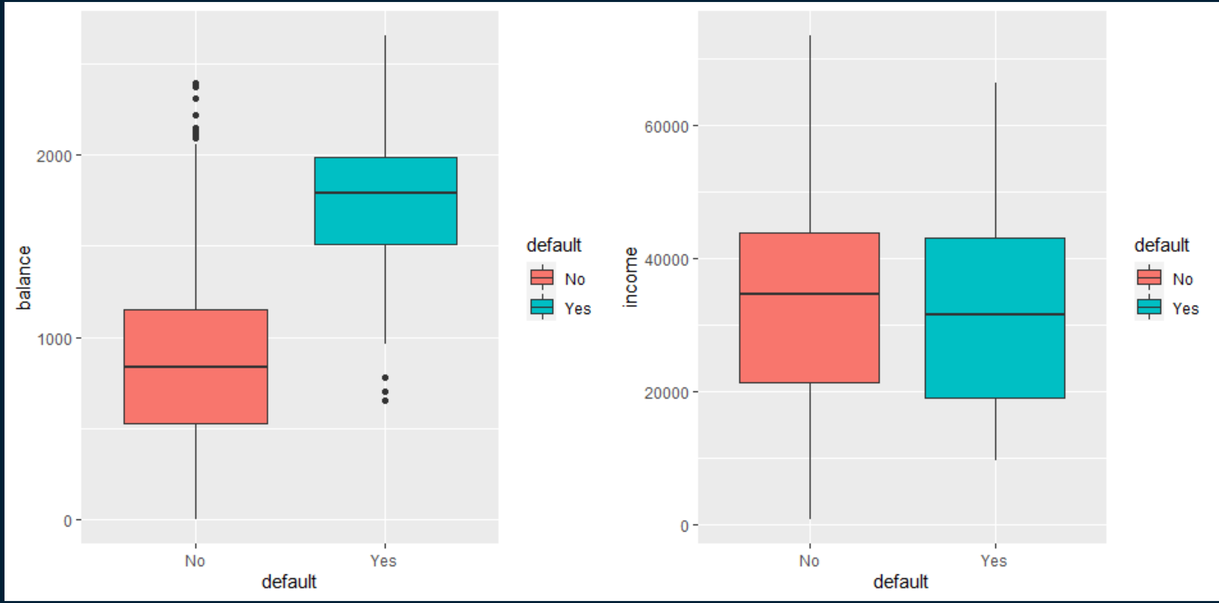


Which factor (balance or income) has the stronger relation to default?

Classification Example

Credit Card Default Data.csv

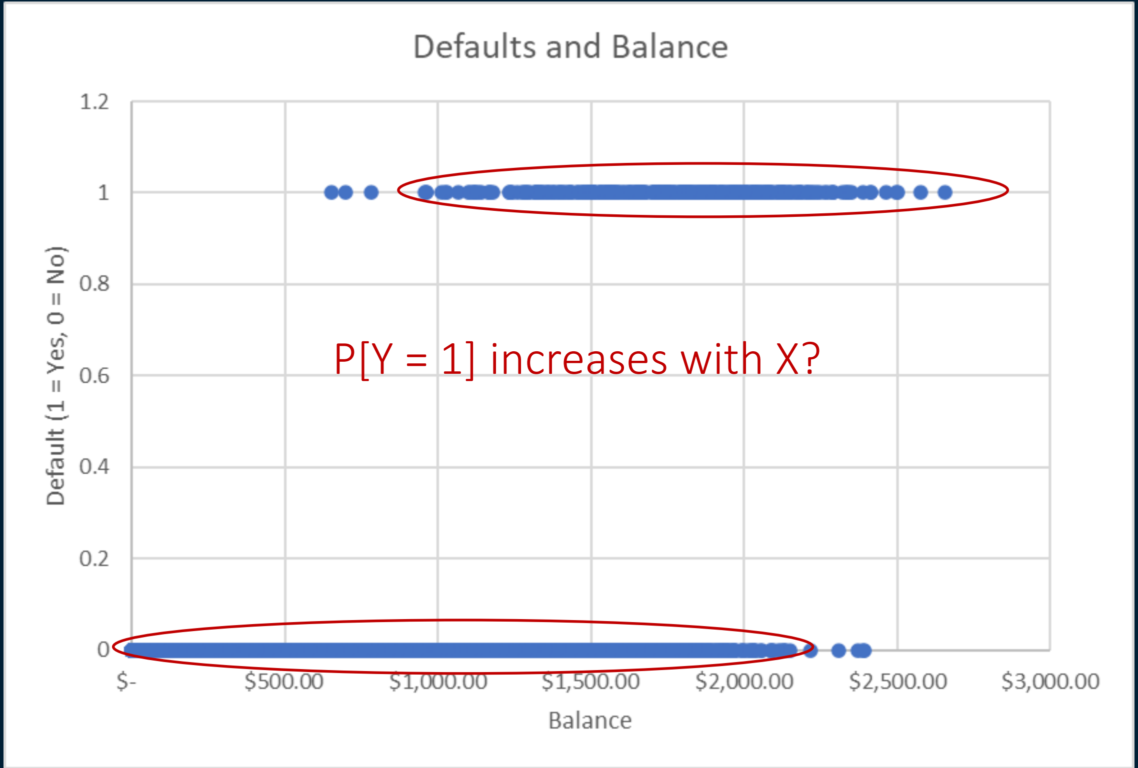
A	B	C	D	E
	default	student	balance	income
1	No	No	$729.53	$44,361.63
2	No	Yes	$817.18	$12,106.13
3	No	No	$1,073.55	$31,767.14
4	No	No	$529.25	$35,704.49
5	No	No	$785.66	$38,463.50
6	No	Yes	$919.59	$7,491.56
7	No	No	$825.51	$24,905.23
8	No	Yes	$808.67	$17,600.45
9	No	No	$1,161.06	$37,468.53
10	No	No	$-	$29,275.27
11	No	Yes	$-	$21,871.07
12	No	Yes	$1,220.58	$13,268.56
13	No	No	$237.05	$28,251.70
14	No	No	$606.74	$44,994.56
15	No	No	$1,112.97	$23,810.17
16	No	No	$286.23	$45,042.41
17	No	No	$-	$50,265.31
18	No	Yes	$527.54	$17,636.54
19	No	No	$485.94	$61,566.11
20	No	No	$1,095.07	$26,464.63
21	No	No	$228.95	$50,500.18



Which factor (balance or income) has the stronger relation to default?

Classification Example

Scatterplot



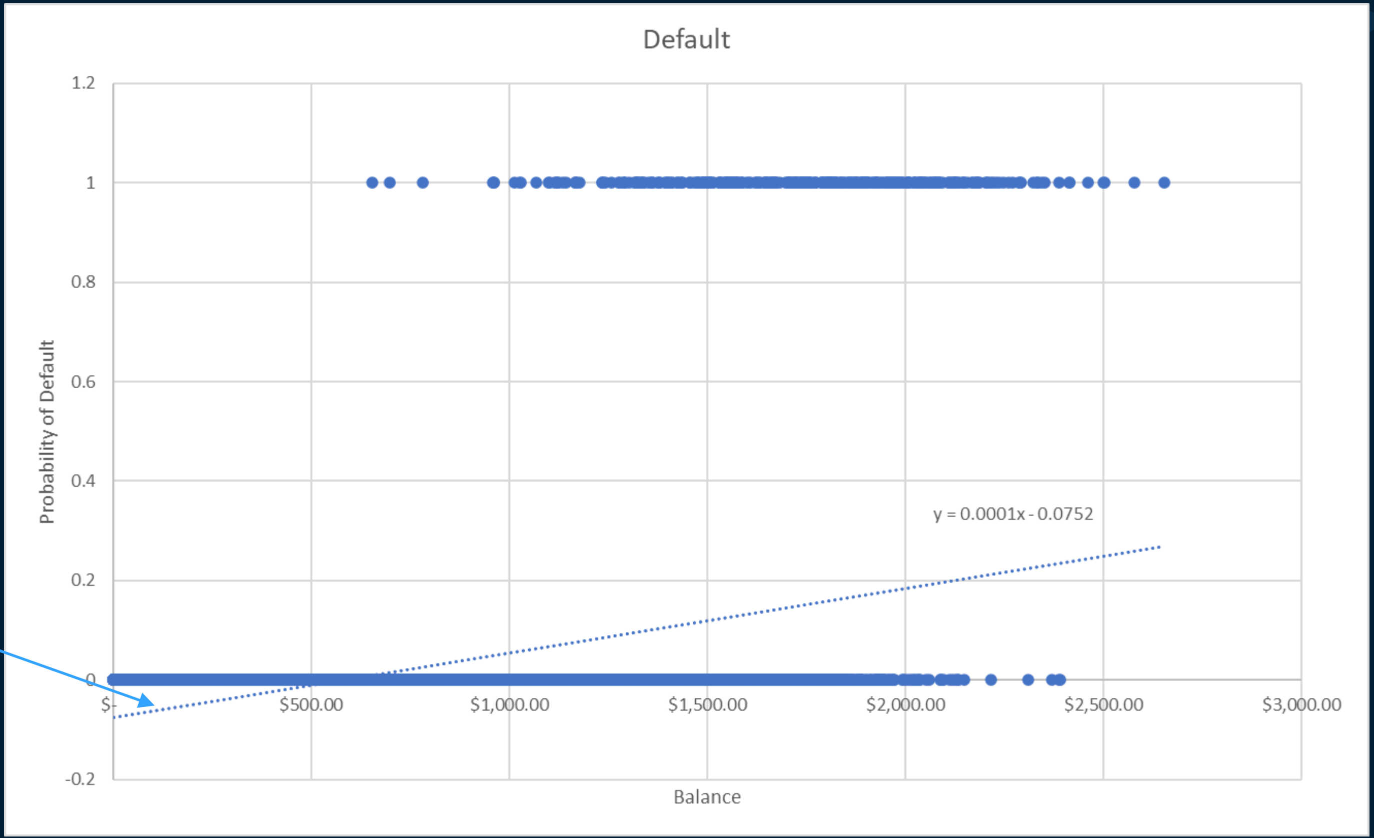
Can’t We Just Use Linear Regression?

- Suppose we code the dependent variable for an insurance claim as fraudulent as follows:

$$Y = \begin{cases} 0 \; if \; No \\ 1 \; if \; Yes \end{cases}$$

- Can’t we simply perform a linear regression of Y on X and classify as Yes if $\hat{Y} > 0.5$??

Issues With Using Linear Regression For Classification



Negative probability of default is meaningless!

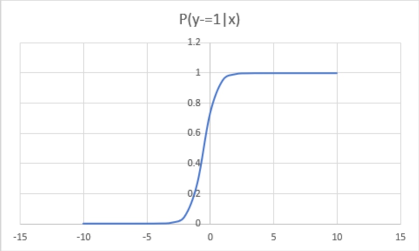
The Logistic Regression Model

- Logistic regression is based on the assumption that a good mathematical model for a binary variable (with a single regressor) is:

$$p(X) \equiv P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}} = \frac{e^{(\beta_0 + \beta_1 X)}}{1 + e^{(\beta_0 + \beta_1 X)}}$$

B5 =1/(1+EXP(-(B0+B1_*A5)))

	A	B
1	B0	1
2	B1	2
3		
4	x	P(y=1|x)
5	-10	5.6028E-09
6	-9	4.13994E-08
7	-8	3.05902E-07
8	-7	2.26032E-06
9	-6	1.67014E-05
10	-5	0.000123395
11	-4	0.000911051
12	-3	0.006692851
13	-2	0.047425873
14	-1	0.268941421
15	0	0.731058579
16	1	0.952574127
17	2	0.993307149
18	3	0.999088949
19	4	0.999876605
20	5	0.999983299
21	6	0.99999774
22	7	0.999999694
23	8	0.999999959
24	9	0.999999994
25	10	0.999999999



The Logistic Regression Model

- A bit of re-arrangement of $p(X)$ yields the log odds or logit function:

$$\underbrace{\log\left(\overbrace{\frac{p(X)}{1-p(X)}}^{\text{Odds}}\right)}_{\text{Logit Function}} = \beta_0 + \beta_1 X$$

The logit function is referred to as the link function for logistic regressions.

The Logistic Regression Model

- A bit of re-arrangement of $p(X)$ yields the log odds or logit function:

$$\underbrace{\log\left(\overbrace{\frac{p(X)}{1-p(X)}}^{\text{Odds}}\right)}_{\text{Logit Function}} = \beta_0 + \beta_1 X$$

The logit function is referred to as the link function for logistic regressions.

Parameter Estimation

Parameter Estimation

- Recall that for linear regression, we estimated the parameters based on minimizing the residuals sum of squares:

$$RSS = \sum_{i=1}^{n} {\epsilon_i}^2 = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

- However, we cannot use that method for estimating the parameters of many models, including for logistic regressions
  - WHY???
  - Hint: remember that for linear regression, we are modeling the actual value of the output attribute. What are we modeling for logistic regression?

Parameter Estimation

- In logistic regression, we are not modeling the value of an output attribute, we are modeling the probability that the output attribute takes on one of two possible states.
  - If we have enough data, we could observe the actual probabilities by counting the percentage of states for each given value of X
- In practice, we generally use an alternate technique called Maximum Likelihood Estimation (MLE)

Parameter Estimation

Maximum Likelihood Estimation

- In OLS (Ordinary Least Squares), we are trying to find the parameters of a line (or curve) that minimizes the sum of the squares of the differences between the estimation (given by the regression equation) and the actual data.
- In MLE, we are trying to find the parameters of an equation that maximizes the probability of having actually observed the data that we observe (our training data)

Maximum Likelihood Estimation

- Parameter estimation is performed by Maximum Likelihood Estimation
  - Calculate the probability of observing the data given estimated parameters
  - Find the parameter estimates that maximize that probability

$$\ell(\beta_0, \beta_1, \dots) \equiv p(\beta_0, \beta_1, \dots | X) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i))$$

  - For computational efficiency, we generally maximize the logarithm of this likelihood (referred to as the log-likelihood)

$$ll(\beta_0, \beta_1, \dots) = \sum_{i:y_i=1} \ln(p(x_i)) \sum_{i:y_i=0} \ln(1 - p(x_i))$$

Maximum Likelihood Estimation

=1/(1+EXP(-(B0+B1_*D6)))   =IF(B6="Yes",F6,(1-F6))

	A	B	C	D	E	F	G	H
1	B0	-10.65						
2	B1	0.005499						
3	ll(B0, B1)	-798.23						
4								
5		default	student	balance	income	p(Y=1|X)	likelihood(X)	ll(X)
6	1	No	No	$ 729.53	$ 44,361.63	0.001307	0.998692505	-0.00130835
7	2	No	Yes	$ 817.18	$ 12,106.13	0.002116	0.997884455	-0.00211779
8	3	No	No	$ 1,073.55	$ 31,767.14	0.008607	0.991393153	-0.0086441
9	4	No	No	$ 529.25	$ 35,704.49	0.000435	0.999564966	-0.00043513
10	5	No	No	$ 785.66	$ 38,463.50	0.001779	0.998220565	-0.00178102
11	6	No	Yes	$ 919.59	$ 7,491.56	0.003709	0.996290651	-0.00371625
12	7	No	No	$ 825.51	$ 24,905.23	0.002215	0.997785479	-0.00221698
13	8	No	Yes	$ 808.67	$ 17,600.45	0.002019	0.997981011	-0.00202103
14	9	No	No	$ 1,161.06	$ 37,468.53	0.013852	0.986147536	-0.0139493
15	10	No	No	$ -	$ 29,275.27	0.000024	0.9999763	-2.3701E-05
16	11	No	Yes	$ -	$ 21,871.07	0.000024	0.9999763	-2.3701E-05
17	12	No	Yes	$ 1,220.58	$ 13,268.56	0.019114	0.980885527	-0.01929952
18	13	No	No	$ 237.05	$ 28,251.70	0.000087	0.999912736	-8.7267E-05
19	14	No	No	$ 606.74	$ 44,994.56	0.000666	0.999333979	-0.00066624
20	15	No	No	$ 1,112.97	$ 23,810.17	0.010668	0.989332057	-0.01072525
21	16	No	No	$ 286.23	$ 45,042.41	0.000114	0.999885636	-0.00011437
22	17	No	No	$ -	$ 50,265.31	0.000024	0.9999763	-2.3701E-05
23	18	No	Yes	$ 527.54	$ 17,636.54	0.000431	0.999569037	-0.00043106
24	19	No	No	$ 485.94	$ 61,566.11	0.000343	0.999657136	-0.00034292
25	20	No	No	$ 1,095.07	$ 26,464.63	0.009678	0.990322193	-0.00972494
26	21	No	No	$ 228.95	$ 50,500.18	0.000083	0.999916534	-8.3469E-05

Maximum Likelihood Estimation

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

Interpreting standard regression coefficient statistics:

- Std. Error – sample standard deviation (adjusted for sample size)
  - Roughly 95% of the observations fall within 2 Standard Errors of the estimated value
- Z-statistic – Coefficient divided by the Std. Error
  - Number of standard errors the coefficient is away from zero
  - Looking for a number greater than 2 or 3 for significance of the coefficient
- P-value – probability that the true coefficient is zero
  - Looking for a value less than 0.05 or 0.01

Making Predictions

- Once the parameters have been estimated, it is straightforward to make a prediction on a new case by plugging the values of the parameters and independent variable(s) into the logistic regression equation:

$$p(X) \equiv P(Y = 1|X) = \frac{e^{(\beta_0+\beta_1 X)}}{1 + e^{(\beta_0+\beta_1 X)}}$$

Example

Scikit-Learn Logistic Regression

```
In [154]: import numpy as np
          import pandas as pd
          from sklearn.linear_model import LogisticRegression
          import sklearn.metrics as metrics

In [57]:  credit_default_data = pd.read_csv("Credit Card Default Data.csv")

In [95]:  credit_default_data
Out[95]:
```

	default	student	income	balance
0	Yes	Yes	9663.79	2024.66
1	Yes	Yes	10155.32	1681.48
2	Yes	Yes	10470.64	2066.70
3	Yes	Yes	10591.72	1707.91
4	Yes	Yes	11054.07	1492.96
...	...	...	...	...
9496	No	No	70700.65	1067.84
9497	No	No	71238.55	1253.18
9498	No	No	71878.77	201.81
9499	No	No	72461.30	1233.71
9500	No	No	73554.23	1593.43

9501 rows × 4 columns

Example

Scikit-Learn Logistic Regression

Consider the example of predicting loan defaults based on loan balance:

```
y = credit_default_data['default']
X = credit_default_data[['balance']]
```

```
model1 = LogisticRegression().fit(X,y)
```

```
model1.intercept_
```

```
array([-10.64979227])
```

```
model1.coef_
```

```
array([[0.005498]])
```

$$p(X) = \frac{e^{(-10.65+0.0055X)}}{1+e^{(-10.65+0.0055X)}}$$

Example

- What is the estimated probability of default for someone with a balance of $1000?

$$p(X)\,\frac{e^{(-10.65+0.0055*1000)}}{1+e^{(-10.65+0.0055*1000)}} = 0.006$$

- What is the estimated probability of default for someone with a balance of $2000?

$$p(X)\,\frac{e^{(-10.65+0.005*2000)}}{1+e^{(-10.65+0.005*2000)}} = 0.586$$

Example

Scikit-Learn Logistic Regression

Consider the example of predicting loan defaults based on student status:

Create simple logistic regression model with student as predictor

```
y = credit_default_data['default']
dummy_vars = pd.get_dummies(credit_default_data['student'])
X = dummy_vars[['Yes']]
```

```
model2 = LogisticRegression().fit(X,y)
```

```
model2.intercept_
```

```
array([-3.43705617])
```

```
model2.coef_
```

```
array([[0.35856645]])
```

$$p(X) = \frac{e^{(-3.44 + 0.359X)}}{1 + e^{(-3.44 + 0.359X)}}$$

Default Example

Using Student as Predictor

Calculate estimated probability of default for students and non-students:

```
student = 1
np.exp(model2.intercept_ + model2.coef_*student)/(1+np.exp(model2.intercept_ + model2.coef_*student))
```

```
array([[0.0440033]])
```

```
student = 0
np.exp(model2.intercept_ + model2.coef_*student)/(1+np.exp(model2.intercept_ + model2.coef_*student))
```

```
array([[0.03115723]])
```

$$P(default \mid student = Yes) = 0.044$$
$$P(default \mid student = No) = 0.031$$

Interpreting Coefficients

Interpreting Coefficients

Understanding Odds and Odds Ratios

$$\underbrace{\log\left(\overbrace{\frac{p(Y)}{1-p(Y)}}^{\text{Odds}}\right)}_{\text{Logit Function}} = \beta_0 + \beta_1 X$$

Odds Ratios

- A random event may be observed with probability π
- The odds ratio (or just "odds") is an expression of how much more likely an event is to occur than not occur
  - Closely tied with gambling where payoffs are based on odds ratios

Odds Ratios

- Mathematically:

$$Odds[A] = \frac{\pi}{1-\pi}$$

- Example, polls indicate:
  - There is a 2/3 chance of candidate A winning
  - There is a 1/3 chance of candidate B winning
- The odds of candidate A winning are “two to one”, or 2:1

Interpreting Coefficients

Odds Ratios

An odds ratio quantifies that strength of the association between two events and is defined as the ratio of the odds of Y in the presence of X to the odds of Y in the absence of X:

$$odds\ ratio = \frac{odds(Y = 1|X = 1)}{odds(Y = 1|X = 0)}$$

Thus, if the odds ratio is 2, it is twice as likely that Y=1 when X=1 than when X = 0.

Why do we care??

Interpreting Coefficients

Binary Factor Variable X

Some basic algebraic manipulations indicate that the coefficient β_j in the logistic regression is the log of the odds ratio for X_j:

$$\frac{odds(\beta_0 + \cdots + \beta_j(\overset{=1}{X_j}) + \ \ldots + \beta_p X_p)}{odds(\beta_0 + \cdots + \beta_j(\underset{=0}{X_j}) + \ \ldots + \beta_p X_p)} = \frac{e^{(\beta_0 + \cdots + \beta_j * 1 + \ \ldots + \beta_p X_p)}}{e^{(\beta_0 + \cdots + \beta_j * 0 + \ \ldots + \beta_p X_p)}} = \ e^{\beta_j}$$

Thus, the coefficient β_j in the logistic regression is the log of the odds ratio for X_j:

$$\left(\frac{odds(X_j = 1)}{odds(X_j = 0)}\right) = \ e^{\beta_j}$$

Odds and Odds Ratios

English Language Usage – Odds Ratios

- Consider the odds ratio of the odds of default if student status = 1:
  - If it were 1.5, we would simply say that the odds of default are 1.5 times higher if the borrower is a student than if the borrow is not

Interpreting the Regression Coefficient

Student Status as Predictor

$$p(X) = \frac{e^{(-3.44 + 0.359X)}}{1 + e^{(-3.44 + 0.359X)}}$$

Recall:

$$\log\left(\frac{odds(X_j=1)}{odds(X_j=0)}\right) = \beta_j = 0.359$$

$$\frac{odds(X_j = 1)}{odds(X_j = 0)} = e^{0.359} = 1.43$$

The odds of the result (default) are 1.43 times higher if the borrower is a student than if not

Interpreting Coefficients

Continuous Variable X_j

Similarly, for a continuous variable X_j if the value of X_j increases by 1:

$$\frac{odds(\beta_0 + \beta_j(X_j+1) + \ \ldots + \beta_p X_p)}{odds(\beta_0 + \beta_j X_j + \ \ldots + \beta_p X_p)} = \frac{e^{\beta_0+\beta_j(X_j+1)+\ \ldots+\beta_p X_p}}{e^{\beta_0+\beta_j X_j+\ \ldots+\beta_p X_p}} = \ e^{\beta_j}$$

Thus, the coefficient β_j in the logistic regression is the log of the odds ratio for an increase by 1 for the value X_j:

$$\frac{odds(X_j + 1)}{odds(X_j)} = \ e^{\beta_j}$$

Confounding

Multiple Logistic Regression

- Making a model for loan default based on balance, income, and student status:

Create and Assess multiple logistic model

```
y3 = credit_default_data['default'].copy()
X3 = credit_default_data.drop('default', 1).copy()
```

```
dummy_vars = pd.get_dummies(credit_default_data['student'])
X3['student'] = dummy_vars['Yes']
X3
```

	student	income	balance
0	1	9663.79	2024.66
1	1	10155.32	1681.48
2	1	10470.64	2066.70
3	1	10591.72	1707.91
4	1	11054.07	1492.96
...	...	...	...
9496	0	70700.65	1067.84
9497	0	71238.55	1253.18
9498	0	71878.77	201.81
9499	0	72461.30	1233.71
9500	0	73554.23	1593.43

9501 rows × 3 columns

Multiple Logistic Regression

- Making a model for loan default based on balance, income, and student status:

Why is the parameter on student negative here when it was positive as part of a Simple Logistic Regression??

Split dataset into training and test

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test  = train_test_split(X3,y3, test_size = 0.25, random_state = 1)
```

```
model3 = LogisticRegression().fit(X_train,y_train)
```

```
model3.intercept_
```

```
array([-2.95293691])
```

```
pd.DataFrame(model3.coef_.T, columns = ['Coefficients'], index = X.columns)
```

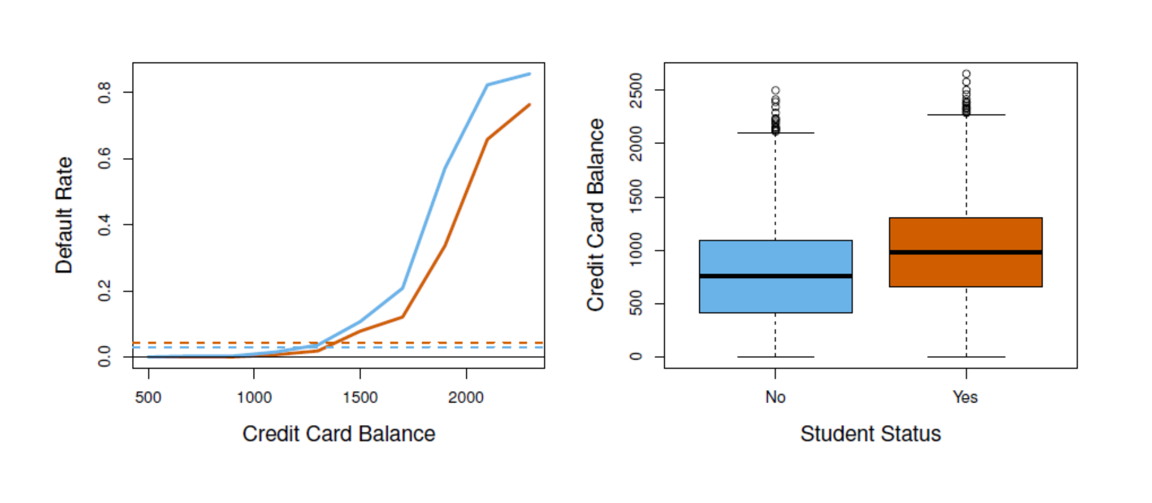
	Coefficients
student	-3.911981
income	-0.000135
balance	0.004101

Confounding

- This is an example of ***confounding***
  - Combining factors that are correlated in such a way as to distort the true relationship.

- In this example, student status is confounded with balance levels
  - Without considering other factors, students have higher chance of default
  - Considering balance as an additional factor, students have lower chance of default compared with non-students *who hold the same balance*
  - Generally, students hold a higher balance and consequently overall higher chance of default. In other words, balance and student status are positively correlated

Confounding

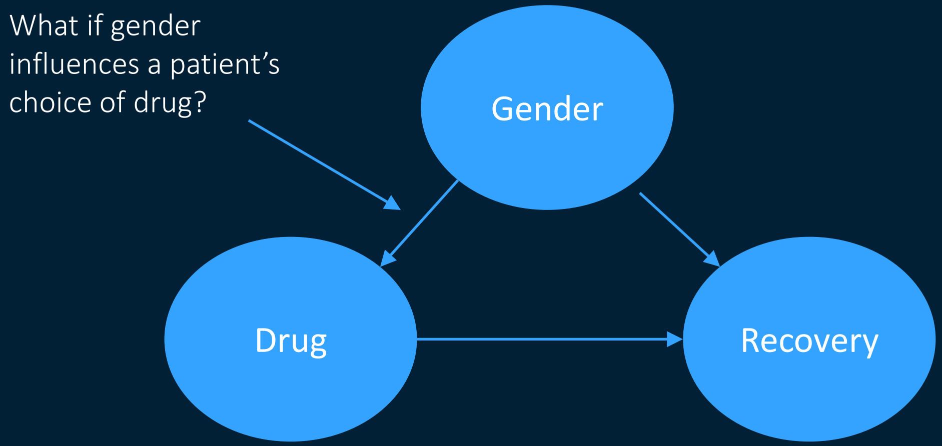


Left: Default rates are shown for students (orange) and non-students (blue). The solid lines display default rate as a function of balance, while the horizontal broken lines display the overall default rates.

Right: Boxplots of balance for students (orange) and non-students (blue) are shown.

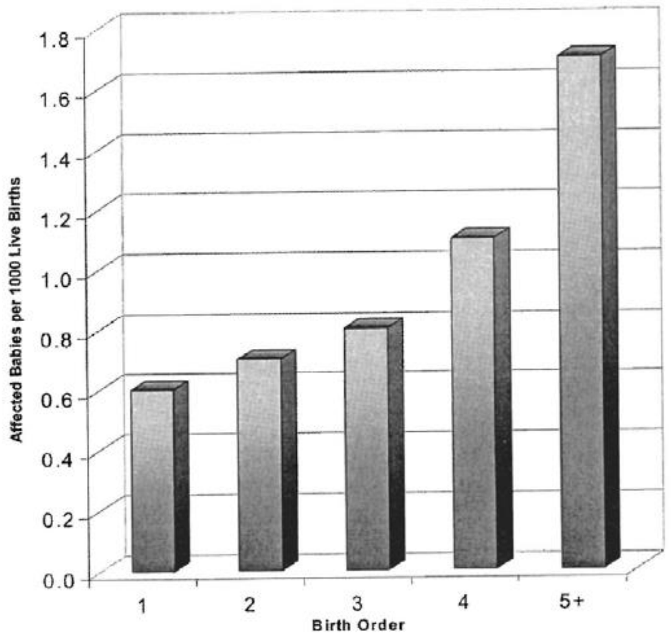
Confounding

- Confounding occurs when one predictor variable influences both another predictor variable and the response variable
- Example: predicting drug effectiveness



Confounding Example

Association Between Birth Order and Down Syndrome

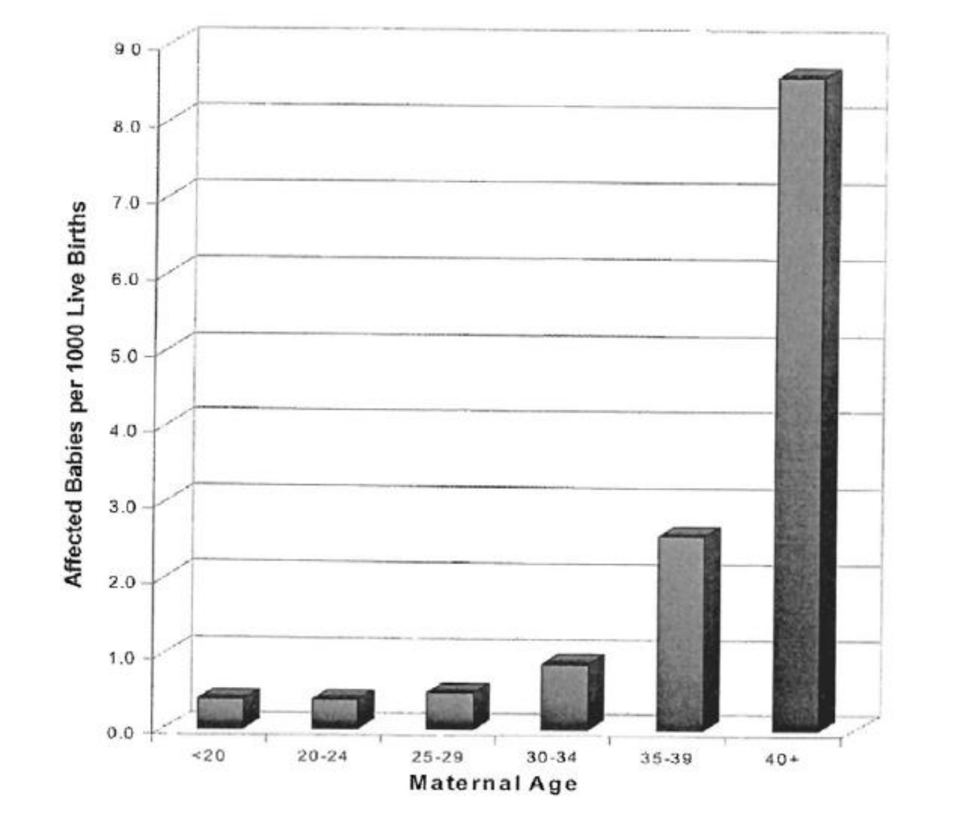


Data from Stark and Mantel (1966)

Source: Rothman 2002

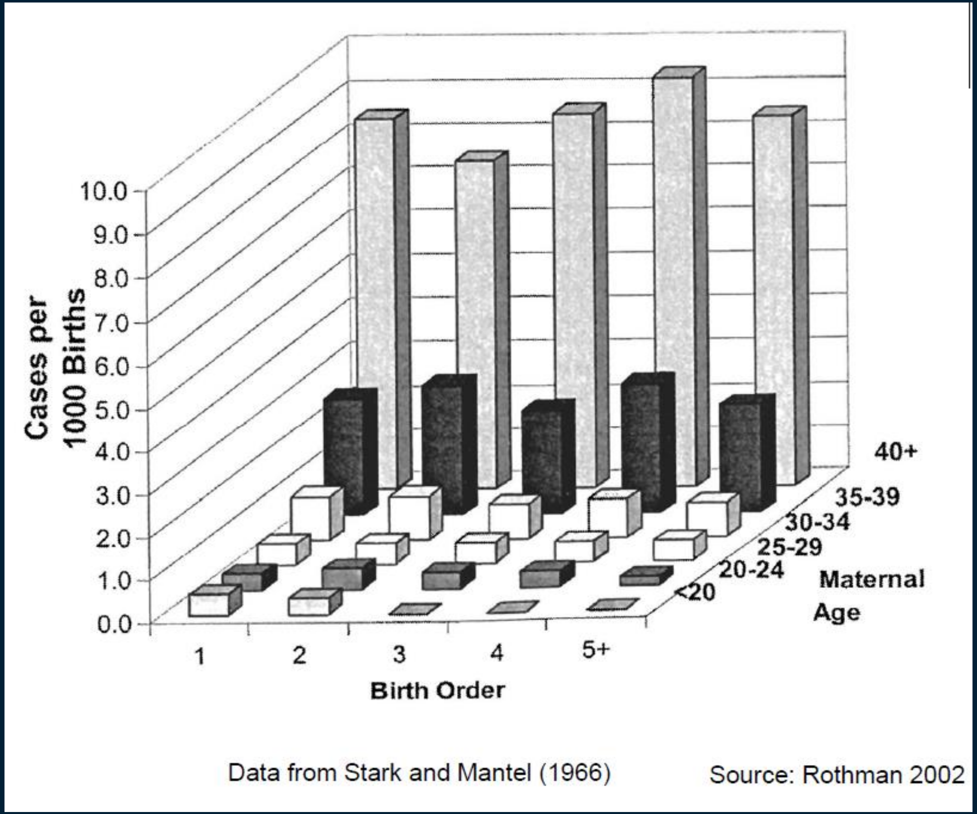
Confounding Example

Association Between Maternal Age and Down Syndrome



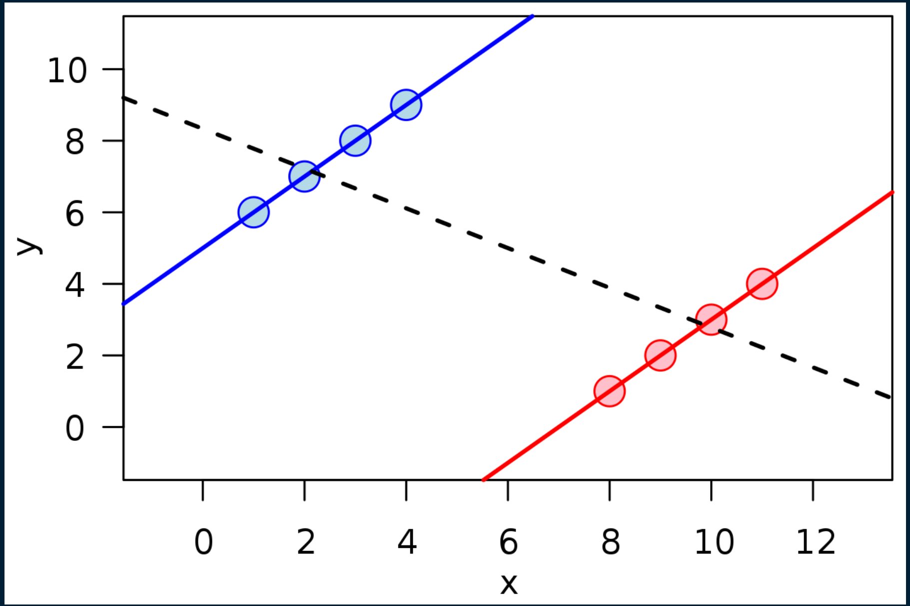
Confounding Example

Association Between Birth Order and Down Syndrome



Simpson’s Paradox

- A trend observed in an overall dataset disappears or reverses when the trend is observed by groups



Confounding

- “Confounding, the situation in which an apparent effect of an exposure on risk is explained by its association with other factors, is probably the most important cause of spurious associations in observational epidemiology”
  - BMJ Editorial: “The Scandal of Poor Epidemiological Research”, 16 October 2004
- Randomized clinical trials (RCTs) attempt to avoid or minimize confounding through randomization and matching techniques
  - As opposed to “observational studies”

Model Assessment

Assessment Measures (Fit Statistics)

Binary Targets

The techniques that are used to model fit assessment for regression models do not apply for classification models (why?)

Three types of assessment measures for binary classification models:

- Classification accuracy: Accuracy in predicting the actual category result (0/1, True/False, Churn/No Churn, etc.)
- Ranking predictions: Accuracy of the rankings of the likelihood of the event
- Estimate predictions: Accuracy of the actual probability predictions

Model Assessment Overview

Classification Accuracy

- Focus is on "misclassification rate" – what percentage of our predictions are wrong?
- Key statistics
  - "Confusion matrix"
  - Sensitivity/specificity
- Key assessment graphs
  - Receiver Operating Characteristics (ROC) curves
  - Lift curves

Classification Predictions

Confusion Matrix

		Actual Classification		
		Negative	Positive	Totals
Predicted Classification	Negative	**TN** (# true negatives)	**FN** (# false negatives)	N (# true negatives)
	Positive	**FP** (# false positives)	**TP** (# true positives)	P (# true positives)

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

Classification Predictions

- Sensitivity: what percentage of the positive outcomes do we correctly identify?
  - Does the classifier capture most of the important events?
- Specificity: what percentage of the negative outcomes do we correctly identify?
  - Does the classifier "weed out" most of the unimportant events?

Classification – Assessment Techniques

Confusion Matrix

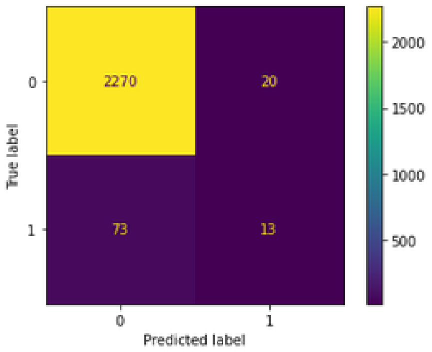
Generate Confustion Matrix

```
y_hat = model3.predict(X_test)
cnf_matrix = metrics.confusion_matrix(y_test, y_hat)
cnf_matrix
```

```
array([[2270,   20],
       [  73,   13]], dtype=int64)
```

```
metrics.ConfusionMatrixDisplay(cnf_matrix).plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x291b54d8cd0>
```



Classification – Assessment Techniques

Metrics

```python
TN = cnf_matrix[0,0]
FP = cnf_matrix[0,1]
FN = cnf_matrix[1,0]
TP = cnf_matrix[1,1]
print('True Negatives:', TN)
print('False Positives:', FP)
print('False Negatives:', FN)
print('True Positives:', TP)
```

```
True Negatives: 2270
False Positives: 20
False Negatives: 73
True Positives: 13
```

```python
Sensitivity = TP/(TP+FN)
Specificity = TN/(TN + FP)
print('Sensitivity:', Sensitivity)
print('Specificity:', Specificity)
```

```
Sensitivity: 0.1511627906976744
Specificity: 0.9912663755458515
```

Is this good performance?

Classifier

- There is generally a tradeoff between sensitivity and specificity
- After modeling the probability of an event, we must decide on the "discrimination threshold" we will use
  - Typically, 50% is used
  - Why would we want to use anything else?
- Adjusting the discrimination threshold allows the data scientist to fine tune the model to get the desired balance between sensitivity and specificity

Classifier

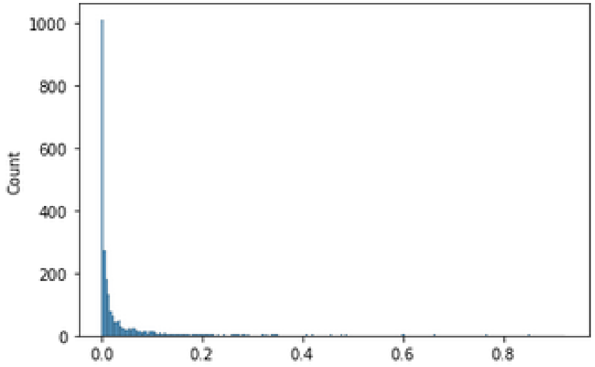
Looking at Prediction Probabilities

```
y_probs = model3.predict_proba(X_test)[:,1]
y_probs
```

```
array([0.05459464, 0.00301417, 0.00049685, ..., 0.05420965, 0.00601634,
       0.00354868])
```

```
sns.histplot(y_probs)
```

```
<AxesSubplot:ylabel='Count'>
```



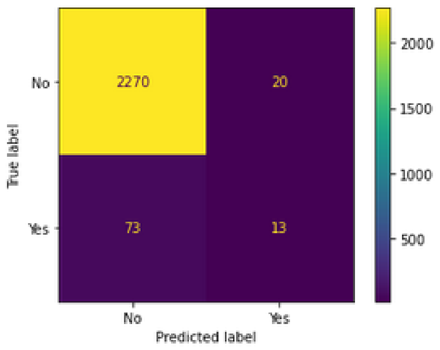
Classifier

Adjusting Discrimination Threshold

- Sklearn does not have a built-in parameter to specify a discrimination threshold, but it is easy to implement:

```python
threshold = 0.5
preds = np.where(y_probs > threshold, 'Yes', 'No')
cnf_matrix = metrics.confusion_matrix(y_test, preds)
metrics.ConfusionMatrixDisplay(cnf_matrix, display_labels = ["No", "Yes"]).plot()
```

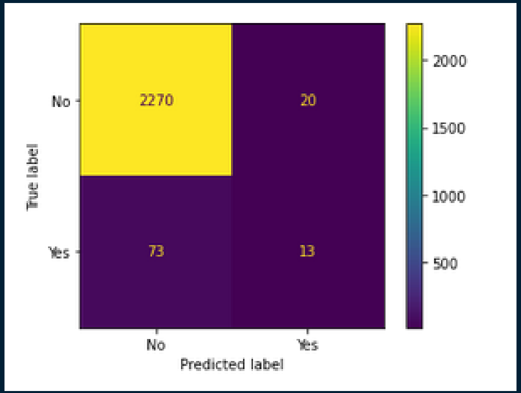
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x291b5d48730>
```



Classifier

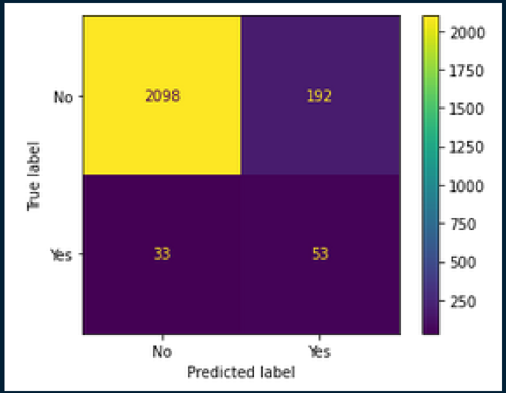
- There is generally a tradeoff between sensitivity and specificity
- Adjusting the discrimination threshold allows the data scientist to fine tune the model to get the desired balance between sensitivity and specificity

Threshold = 0.5



With a 0.5 classification threshold, we "miss" 73 true positives

Threshold = 0.1

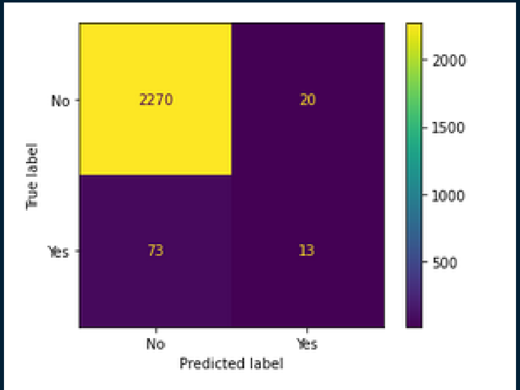


With a 0.1 classification threshold, we capture an additional 40 true positives, but at the cost of having to look at 192 false positives

Classifier

Comparing Thresholds

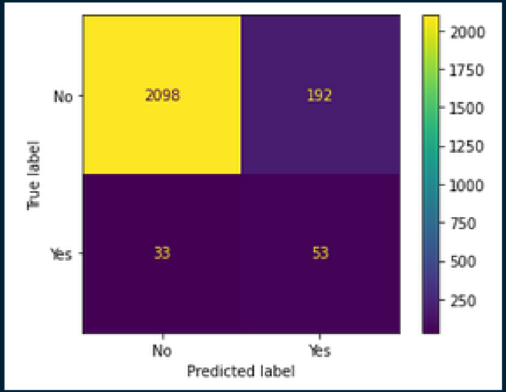
Threshold = 0.5



With a 0.5 classification threshold, we "miss" 73 true positives

```
Sensitivity: 0.1511627906976744
Specificity: 0.9912663755458515
```

Threshold = 0.1



With a 0.1 classification threshold, we capture an additional 40 true positives, but at the cost of having to look at 192 false positives

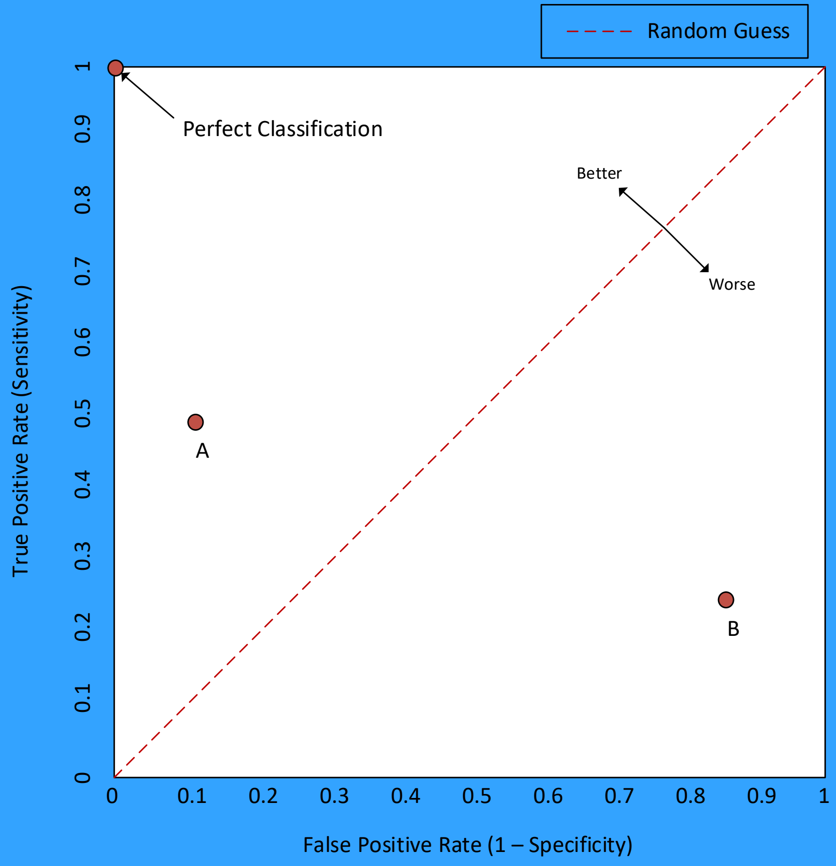
```
Sensitivity: 0.6162790697674418
Specificity: 0.9161572052401746
```

Classification Assessment Visualizations

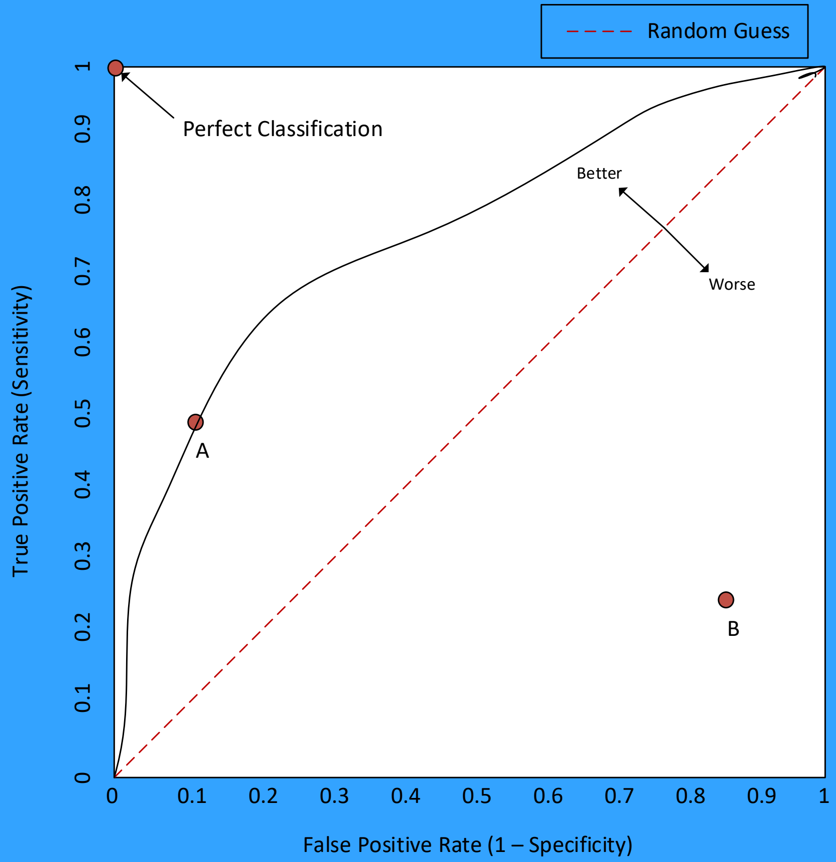
ROC Chart

- Graphical plot to assist in optimizing the sensitivity/specificity tradeoff
- Plots the True Positive Rate (sensitivity) against the False Positive Rate (1-specificity) at each possible value of the discrimination threshold.

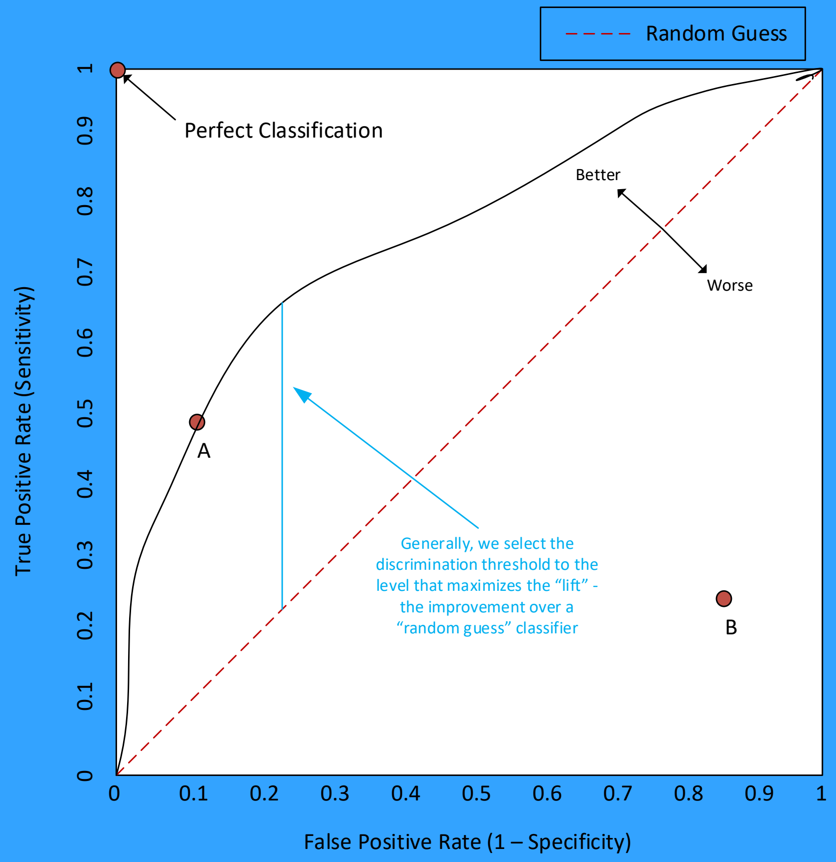
ROC Curve


Random Guess
Perfect Classification
Better
Worse
A
B
True Positive Rate (Sensitivity)
False Positive Rate (1 – Specificity)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

ROC Curve


Random Guess
Perfect Classification
Better
Worse
A
B
True Positive Rate (Sensitivity)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
False Positive Rate (1 – Specificity)

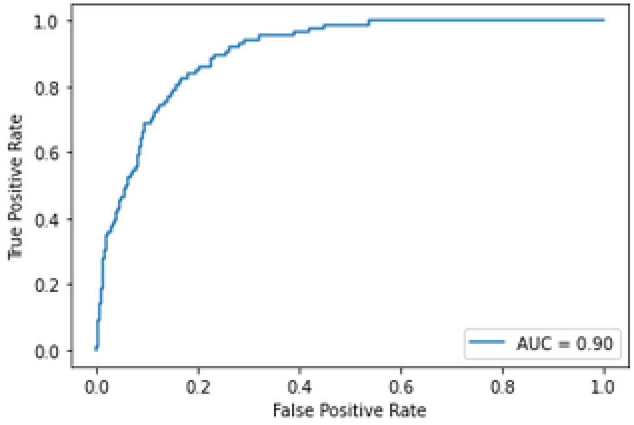
ROC Curve


Random Guess
Perfect Classification
Better
Worse
A
B
Generally, we select the discrimination threshold to the level that maximizes the "lift" - the improvement over a "random guess" classifier
True Positive Rate (Sensitivity)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
False Positive Rate (1 – Specificity)

ROC Curve

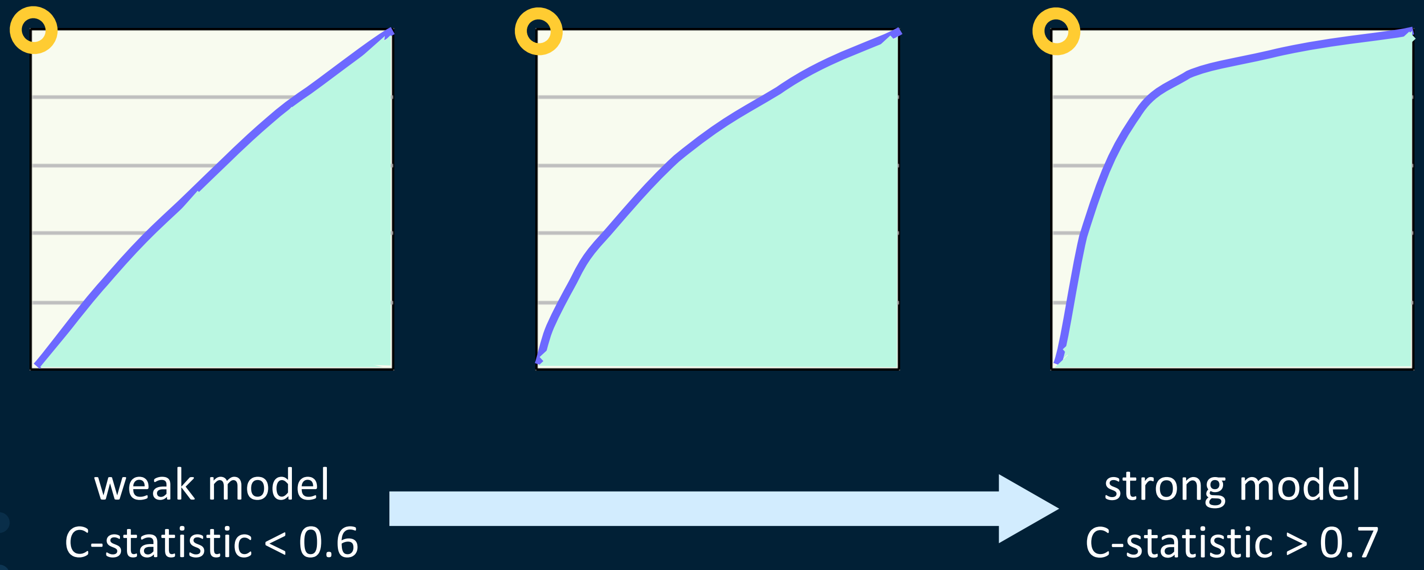
Loan Default Example

```python
fpr, tpr, thresholds = metrics.roc_curve(y_test, y_probs, pos_label = 'Yes')
roc_auc = metrics.auc(fpr, tpr)
display = metrics.RocCurveDisplay(fpr=fpr, tpr=tpr, roc_auc=roc_auc).plot()
```



Ranking Predictions

ROC Index / C Statistic



Lift

- Background: many classification applications involve looking for relatively low probability, but important events, for example:
  - Fraudulent tax returns
  - Diseases
  - People likely to default on a loan
- We want to identify a group of cases for more in-depth investigation, but we are limited in our ability to perform a large number of investigations. Therefore, we want our classifier to select a group with a high number of "true positive" outcomes

Model Lift

- Another measure of the performance of a classification model (or an association rule model) that measures the "lift" or "improvement" a model provides over randomly selecting observations.
- Example:
  - Your dataset of 1000 cases has 100 "true positives"
  - Randomly selecting 100 cases could thus be expected to return 10 positives
  - If your classifier takes its 100 "highest probability" cases and returns 40 positives, the lift would be 4.
- Lift curves display a curve of the performance of a classifier for increasing size "bins" – your top 5%, 10%, of cases, etc.

Lift Curve – Model Lift

	A	B	C	D	E
1	P(Y)	Event?			
2	0.08	0		Number of observations:	100
3	0.06	0		Number of events:	25
4	0.88	0			
5	0.13	0			
6	0.53	0			
7	0.47	0			
8	0.89	1			
9	0.15	0			
10	0.17	0			
11	0.02	0			
12	0.22	0			
13	0.74	0			
14	0.41	0			
15	0.25	0			
16	0.66	0			
17	0.49	0			
18	0.21	1			
19	0.11	0			
20	0.66	0			
21	0.64	0			
22	0.42	1			
23	0.7	1			
24	0.27	0			
25	0.6	0			
26	0.51	1			
27	0.16	0			
28	0.02	0			
29	0.84	0			
30	0.11	0			
31	0.81	0			
32	0.95	1			
33	0.89	1			
34	0.38	0			
35	0.45	0			
36	0.76	0			
37	0.8	0			
38	0.7	0			

Sort by decreasing model probability

A	B
P(Y)	Event?
0.98	1
0.98	1
0.98	1
0.96	1
0.96	1
0.95	1
0.95	1
0.94	1
0.94	1
0.93	0
0.93	1
0.92	1
0.91	1
0.9	0
0.89	1
0.89	1
0.89	1
0.88	0
0.88	0
0.88	1
0.86	0
0.86	1
0.84	0
0.82	0
0.81	0
0.8	0
0.79	0
0.78	1
0.77	0
0.76	0
0.74	0
0.73	0
0.72	0
0.7	1
0.7	0
0.69	0
0.67	0
0.66	0

# of events in top 5% high probability observations: 5

# of events (on average) in a random sample of 5% of observations? 1.25

Model lift at 5%: 5/1.25 = 4

Lift Curve – Model Lift

	A	B	C	D	E
1	P(Y)	Event?			
2	0.08	0		Number of observations:	100
3	0.06	0		Number of events:	25
4	0.88	0			
5	0.13	0			
6	0.53	0			
7	0.47	0			
8	0.89	1			
9	0.15	0			
10	0.17	0			
11	0.02	0			
12	0.22	0			
13	0.74	0			
14	0.41	0			
15	0.25	0			
16	0.66	0			
17	0.49	0			
18	0.21	1			
19	0.11	0			
20	0.66	0			
21	0.64	0			
22	0.42	1			
23	0.7	1			
24	0.27	0			
25	0.6	0			
26	0.51	1			
27	0.16	0			
28	0.02	0			
29	0.84	0			
30	0.11	0			
31	0.81	0			
32	0.95	1			
33	0.89	1			
34	0.38	0			
35	0.45	0			
36	0.76	0			
37	0.8	0			
38	0.7	0			

Sort by decreasing model probability

A	B
P(Y)	Event?
0.98	1
0.98	1
0.98	1
0.96	1
0.96	1
0.95	1
0.95	1
0.94	1
0.94	1
0.93	0
0.93	1
0.92	1
0.91	1
0.9	0
0.89	1
0.89	1
0.89	1
0.88	0
0.88	0
0.88	1
0.86	0
0.86	1
0.84	0
0.82	0
0.81	0
0.8	0
0.79	0
0.78	1
0.77	0
0.76	0
0.74	0
0.73	0
0.72	0
0.7	1
0.7	0
0.69	0
0.67	0
0.66	0

# of events in top 10% high probability observations: 9

# of events (on average) in a random sample of 5% of observations? 2.5

Model lift at 5%: 9/2.5 = 3.6

“Best” Lift

- Performs the same calculation, as model lift except that it is calculated as though a sorting by probability ends up with all the positives (1s) on the top and all the negatives (0s) on the bottom

Lift Curve – Best Lift

P(Y)	Event?	Best Model
0.98	1	1
0.98	1	1
0.98	1	1
0.96	1	1
0.96	1	1
0.95	1	1
0.95	1	1
0.94	1	1
0.94	1	1
0.93	0	1
0.93	1	1
0.92	1	1
0.91	1	1
0.9	0	1
0.89	1	1
0.89	1	1
0.89	1	1
0.88	0	1
0.88	0	1
0.88	1	1
0.86	0	1
0.86	1	1
0.84	0	1
0.82	0	1
0.81	0	1
0.8	0	0
0.79	0	0
0.78	1	0
0.77	0	0
0.76	0	0
0.74	0	0
0.73	0	0
0.72	0	0
0.7	1	0
0.7	0	0
0.69	0	0
0.67	0	0

# of events in first 5% observations: 5

# of events (on average) in a random sample of 5% of observations? 1.25

Best lift at 5%: 5/1.25 = 4

Lift Curve – Best Lift

A	B	C
P(Y)	Event?	Best Model
0.98	1	1
0.98	1	1
0.98	1	1
0.96	1	1
0.96	1	1
0.95	1	1
0.95	1	1
0.94	1	1
0.94	1	1
0.93	0	1
0.93	1	1
0.92	1	1
0.91	1	1
0.9	0	1
0.89	1	1
0.89	1	1
0.89	1	1
0.88	0	1
0.88	0	1
0.88	1	1
0.86	0	1
0.86	1	1
0.84	0	1
0.82	0	1
0.81	0	1
0.8	0	0
0.79	0	0
0.78	1	0
0.77	0	0
0.76	0	0
0.74	0	0
0.73	0	0
0.72	0	0
0.7	1	0
0.7	0	0
0.69	0	0
0.67	0	0

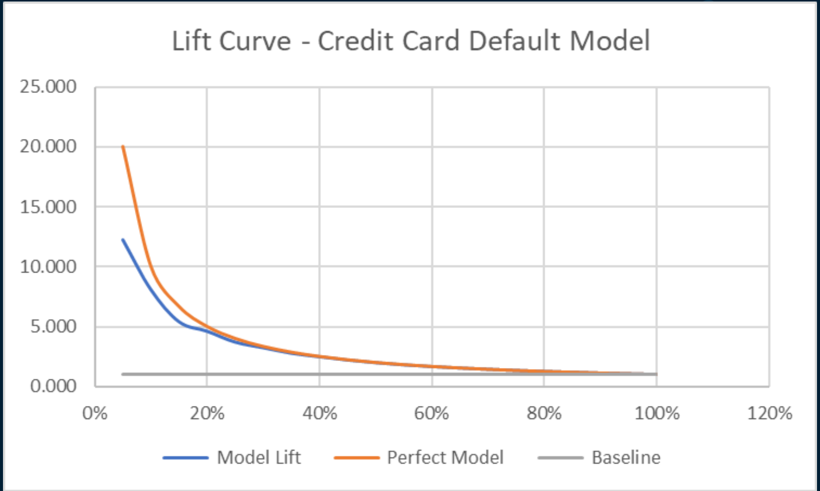
# of events in first 10% observations: 10

# of events (on average) in a random sample of 10% of observations? 2.5

Best lift at 10%: 10/2.5 = 4

Lift Curve – Credit Card Default Model

Percentile	Random	Model	Model Lift	Perfect Model	Baseline
5%	16.65	204	12.252	20	1
10%	33.3	269	8.078	10	1
15%	49.95	270	5.405	6.666666667	1
20%	66.6	307	4.610	5	1
25%	83.25	308	3.700	4	1
30%	99.9	323	3.233	3.333333333	1
35%	116.55	323	2.771	2.857142857	1
40%	133.2	330	2.477	2.5	1
45%	149.85	330	2.202	2.222222222	1
50%	166.5	330	1.982	2	1
55%	183.15	330	1.802	1.818181818	1
60%	199.8	332	1.662	1.666666667	1
65%	216.45	332	1.534	1.538461538	1
70%	233.1	333	1.429	1.428571429	1
75%	249.75	333	1.333	1.333333333	1
80%	266.4	333	1.250	1.25	1
85%	283.05	333	1.176	1.176470588	1
90%	299.7	333	1.111	1.111111111	1
95%	316.35	333	1.053	1.052631579	1
100%	333	333	1.000	1	1



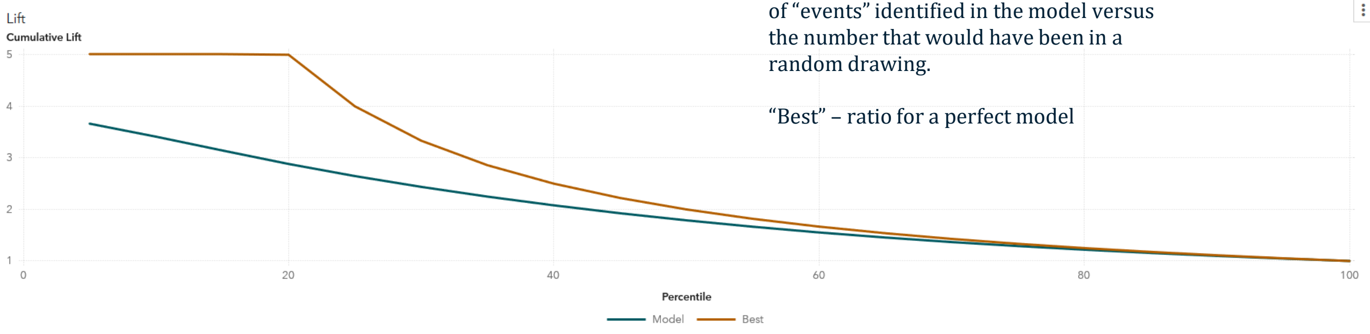
Logistic Regression Results: Lift

Logistic Regression **tgt Binary New Product** (event=1) R-Square **0.2325** Observations Used **1,060,038** Create pipeline

Fit Summary | Residual | Assessment

“Model” lift – ratio between the number of “events” identified in the model versus the number that would have been in a random drawing.

“Best” – ratio for a perfect model



Model Information | Dimensions | Response Profile | Iteration History | Convergence | Fit Statistics | Parameter Estimates | Type III Test | Selection Info | Selection Summary | Lift | Misclassification | ROC | Assessment Statistics

Percentile	Observations	Events	Model	Best	Cumulative Model	Cumulative Best
5.00	53002	38741	3.6633	5.0118	3.6633	5.0118
10.00	53002	33413	3.1595	5.0118	3.4114	5.0118
15.00	53002	27583	2.6082	5.0118	3.1437	5.0118
20.00	53002	22173	2.0966	4.9646	2.8819	5.0000
25.00	53002	18051	1.7069	0.0000	2.6469	4.0000
30.00	53002	14715	1.3914	0.0000	2.4377	3.3333
35.00	53002	11776	1.1135	0.0000	2.2485	2.8571
40.00	53002	9485	0.8969	0.0000	2.0795	2.5000
45.00	53002	7352	0.6952	0.0000	1.9257	2.2222

In this example in the first bin of 53002 observations which the model assigned the highest probability to:

- 38,741 of the 53,002 observations contained the event
- In a random draw of 53,002 observations, 10,575 observations would have contained the event (based

In the second bin (10%) of 53002 observations which the model assigned the highest probability to:

- 33,413 of the 53,002 observations contained the event
- Thus, in the first two bins cumulatively, 72,154 observations would have contained the event

Other Classification Model Assessment Statistics

Assessment Measures (Fit Statistics)

Binary Targets

Prediction Type	Fit Statistic	
Decisions	Accuracy/Misclassification KS Youden	Maximum distance from ROC curve to diagonal
Rankings	ROC Index Gini Coefficient	Area under ROC curve Concordance statistic
Estimates	Average Squared Error RMSE/SBC/AIC/Likelihood	

Ranking Predictions

Concordance Statistic / Gini Index

- Essentially a correlation coefficient between the predicted and actual order
  - Not always able to determine the actual order

Estimate Prediction

Average Squared Error

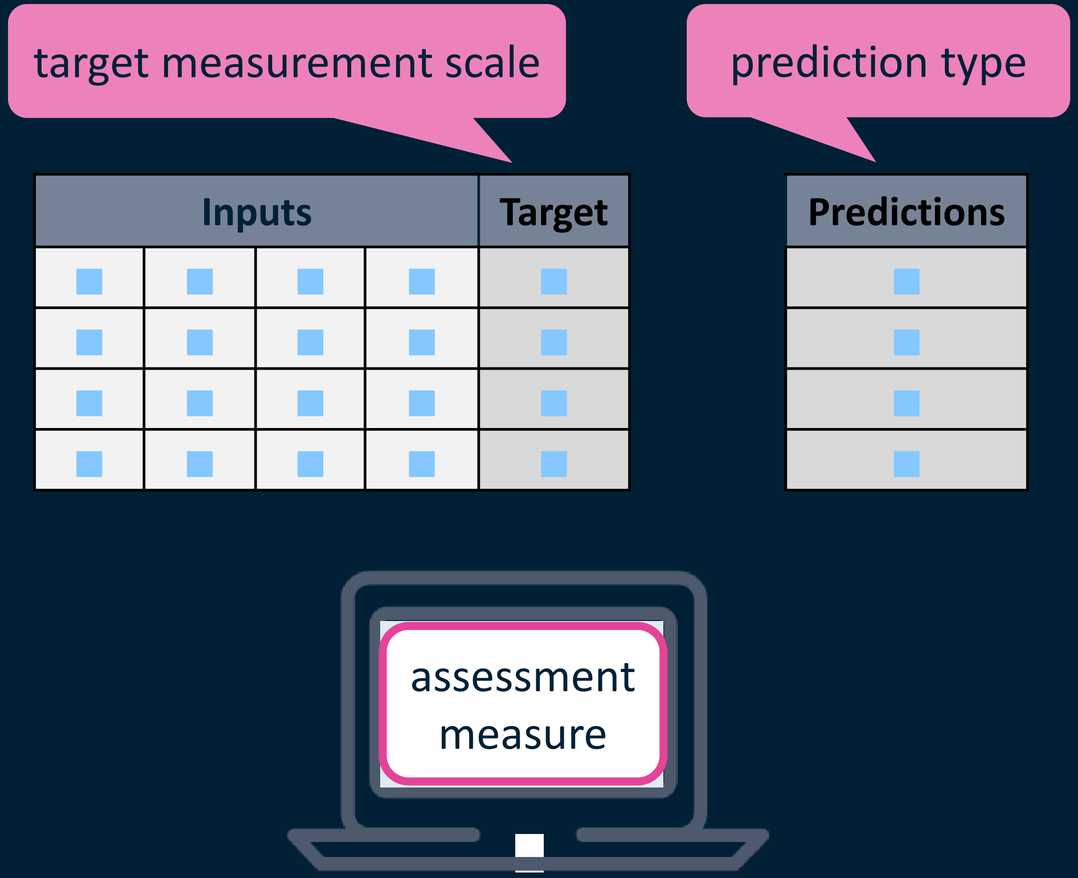
$$ASE_{cat} = \sum_{i=1}^{N}\sum_{j=1}^{J}\frac{(\delta_{ij} - \hat{p}_{ij})^2}{JN}$$

J: Number of target values (classes)

δ_{ij}: Equals 1 if value j occurs in observation i, 0 if not

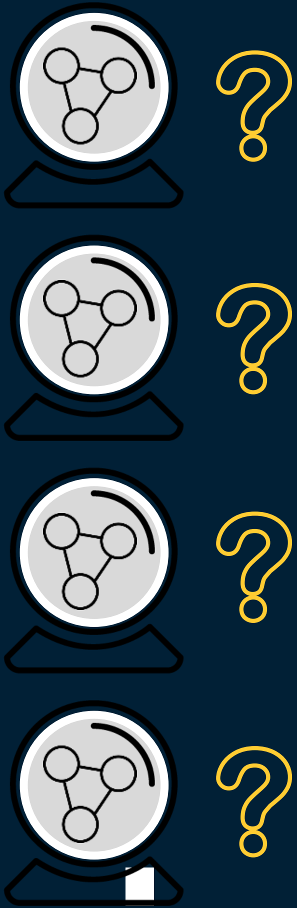
$\hat{p}_{ij}$: Predicted probability of nominal target value j for observation i

Which Assessment Measure Should You Use?



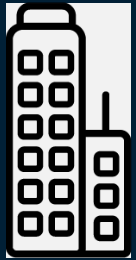
Evaluation of Model Performance

Assessment measures and statistical graphics of performance



Business needs

- Speed of training
- Speed of scoring
- Feasibility of deployment
- Noise tolerance
- Interpretability



Model Assessment Statistics

Classification Models

Accuracy	A measure of how many observations are correctly classified for each value of the response variable. It is the number of event and non-event cases classified correctly, divided by all cases.
Area under the curve (C statistic)	A measure of goodness of fit for binary outcome. It is the concordance rate and it is calculated as the area under the curve.
Average squared error	The sum of squared errors (SSE) divided by the number of observations.
Captured response	The number of events in each bin divided by the total number of events.

Model Assessment Statistics

Classification Models

Cumulative captured response	The cumulative value of the captured response rate.
Cumulative lift	Cumulative lift up to and including the specified percentile bin of the data, sorted in descending order of the predicted event probabilities.
F1 score	The weighted average of precision (positive predicted value) and recall (sensitivity). It is also known as the *F-score* or *F-measure*.
False discovery rate	The expected proportion of type error I – incorrectly reject the null hypothesis (false positive rate).
False positive rate	The number of positive cases misclassified (as negative).

Model Assessment Statistics

Classification Models

Gain	Similar to a lift chart. It equals the expected response rate using the predictive model divided by the expected response rate from using no model at all.
Gini	A measure of the quality of the model. It has values between -1 and 1. Closer to 1 is better. It is also known as Somer's D.
Kolmogorov-Smirnov statistic (KS)	A goodness-of-fit statistic that represents the maximum separation between the model ROC curve and the baseline ROC curve.
KS (Youden)	A goodness-of-fit index that represents the maximum separation between the model ROC curve and the baseline ROC curve.
Lift	A measure of the advantage (or lift) of using a predictive model to improve on the target response versus not using a model. It is a measure of the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model. The higher the lift in the lower percentiles of the chart, the better the model is.

Model Assessment Statistics

Classification Models

Misclassification (Event)	Considers only the classification of the event level versus all other levels. Thus, a non-event level classified as another non-event level does not count in the misclassification. For binary targets, these two measures are the same. It is computed in the context of the ROC report. That is, at each cutoff value, this measure is calculated.
Misclassification (MCE)	A measure of how many observations are incorrectly classified for each value of the response variable. This is the true misclassification rate. That is, every observation where the observed target level is predicted to be a different level counts in the misclassification rate.
Multiclass log loss	The loss function applied to multinomial target. It is the negative log-likelihood of the true labels given a probabilistic classifier's prediction.
ROC separation	The area under the ROC curve is the accuracy. The ROC separation enables you to change the ROC-based cutoff and evaluate the model's performance under different ranges of accuracy.
Root average squared error	It is the square root of the average differences between the prediction and the actual observation.

Dataset Partitioning Considerations

Addressing Rare Events

- Special handling is required when the target of interest is a rare event relative to the total number of samples
  - For example, detecting fraudulent activity
- "Fitting a model without accounting for the extreme imbalance in the occurrence of the event gives you a model that is extremely accurate at telling you absolutely nothing of value"

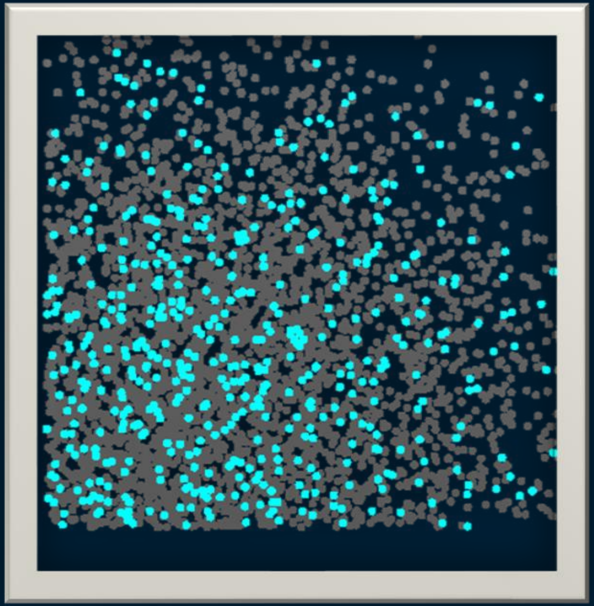
Addressing Rare Events

- A common practice is to build models from a sample with a primary outcome proportion different from the true population ("Event-Based Sampling")
- It can be shown that you can obtain a model of similar predictive power with a smaller overall case count
  - The amount of information in a data set with a categorical outcome is determined not by the total number of cases in the data set, but by the number of cases in the rarest outcome category

Addressing Rare Events

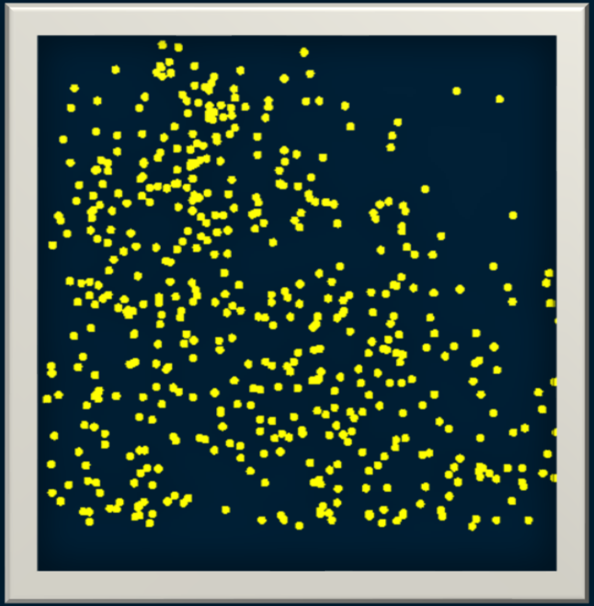
Event-Based Sampling

Secondary Outcome



Select some cases

Primary Outcome

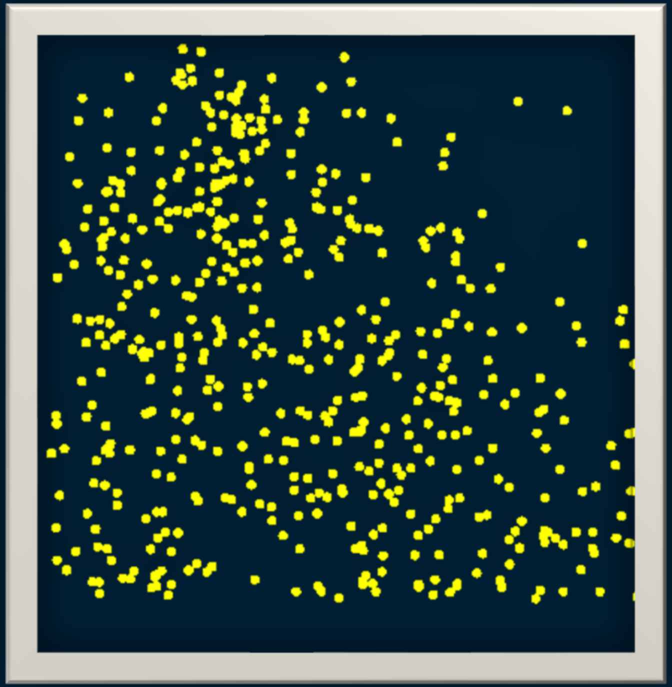


Select all cases

Addressing Rare Events

Event-Based Sampling

- Similar predictive power with smaller case count
- Must adjust assessment measures and graphics
- Must adjust prediction estimates for bias
- Model Studio automatically adjusts for event-based sampling



Dataset Partition Strategies

- Partition data / address rare events
  - Stratified sampling: ensure partitions have same percentages (of a category) as the overall population
  - Event-based sampling: Over/under sample to get specified percentages of each category of observations in each partition

Multinomial Logistic Regression

Multinomial Logistic Regression

Overview

- Extension of logistic regression for multiple categories
- Select a single class to serve as the baseline (here, we select K):

$$P(Y = k|X = x) = \frac{e^{(\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p)}}{1 + e^{(\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p)}}$$

for k = 1, …, K-1, and

$$P(Y = K|X = x) = \frac{1}{1 + e^{(\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p)}}$$

Module 7

Generalized Linear Models

Generalized Linear Models

Overview

Covers cases where the response variable Y is neither qualitative or quantitative

- For example, count variables

Example

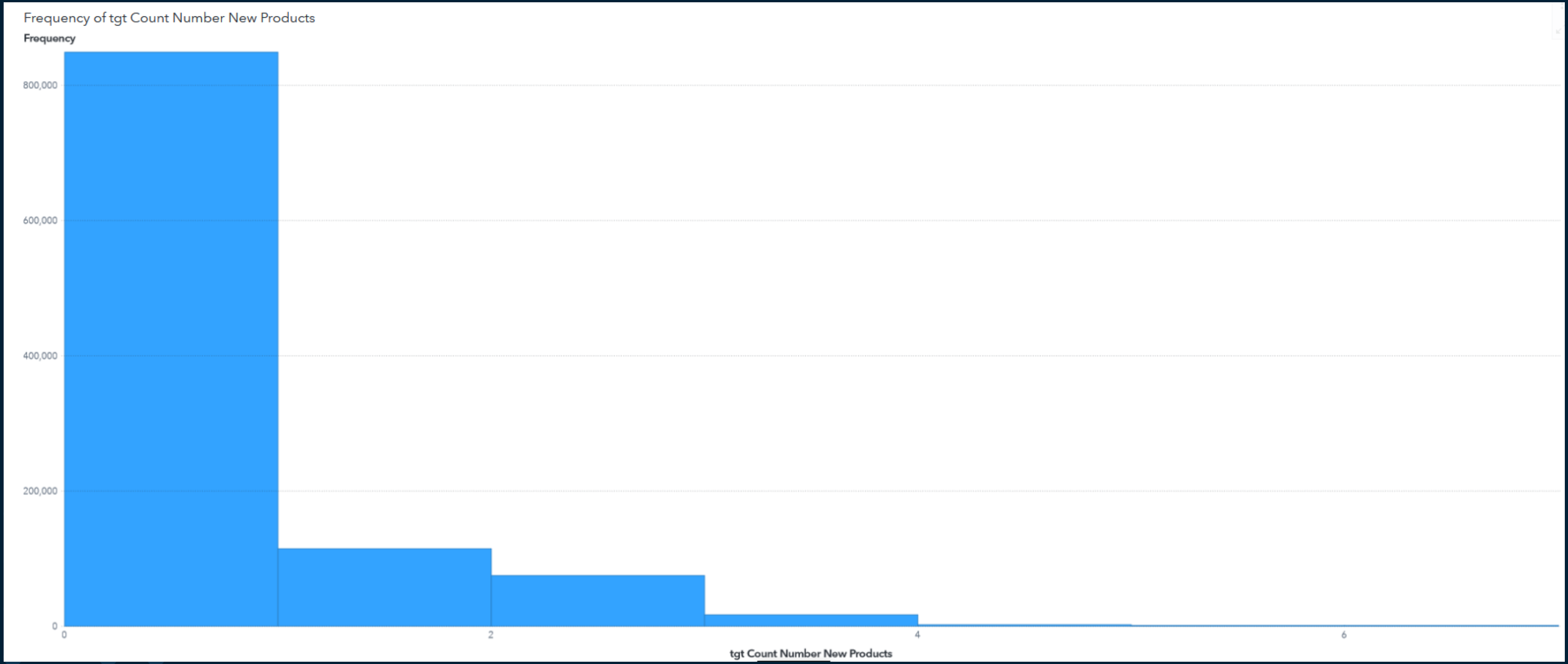
VS_Bank Campaign Response Data

List Table ⋮ Tgt Count Num New Products Linear Regression GLM +

Account ID	category 1 Account Activity Level	category 2 Customer Value Level	logi_rfm1 Average Sales Past 3 Years	logi_rfm10 Count Total Promos Past Year	logi_rfm11 Count Direct Promos Past Year	logi_rfm12 Customer Tenure	logi_rfm2 Average Sales Lifetime	tgt Count Number New Products ▲
100000001	X	A	1.903598951	3.0445224377	2.302585093	4.5325994932	1.8500283774	.
100020117	Z	B	2.7725887222	2.3978952728	1.6094379124	2.8903717579	2.7725887222	0
100008192	X	B	2.6390573296	2.3978952728	1.6094379124	3.4011973817	2.6390573296	0
100022680	Z	A	3.258096538	2.5649493575	1.7917594692	2.8332133441	3.258096538	0
100020744	X	A	3.258096538	2.6390573296	1.9459101491	4.7874917428	2.6581594315	0
100020118	Z	B	3.0445224377	2.5649493575	1.7917594692	3.3322045102	3.0445224377	0
100004096	X	A	2.8332133441	2.7080502011	1.9459101491	3.8286413965	2.6946271808	0
100023030	X	A	2.4570214463	3.0910424534	1.9459101491	4.7449321284	2.5160822673	0
100022390	X	A	3.258096538	2.6390573296	1.9459101491	4.4067192473	2.2159372863	0
100021590	X	A	2.9704144656	2.7725887222	2.0794415417	4.3820266347	2.7245795031	0
100021018	X	A	2.7568403653	2.6390573296	1.9459101491	4.3820266347	2.4484155412	0
100020745	X	A	3.7534960972	2.5649493575	2.0794415417	4.5432947823	3.4397768636	0
100010217	Z	A	3.0445224377	2.3978952728	1.6094379124	3.0910424534	3.0445224377	0
100020119	Z	B	3.258096538	2.4849066498	1.7917594692	2.8332133441	3.258096538	0
100026873	X	C	2.7725887222	1.9459101491	1.3862943611	4.2046926194	2.5257286443	0
100023692	X	A	2.7725887222	2.4849066498	1.6094379124	4.7361984484	2.1882959466	0
100023368	X	A	2.6026896854	2.3978952728	1.6094379124	3.4965075615	2.6026896854	0
100023031	X	A	4.6151205168	3.0445224377	1.9459101491	4.248495242	4.189654742	0
100022715	Z	A	3.258096538	2.6390573296	1.9459101491	3.1780538303	3.258096538	0
100011128	X	D	2.4423470354	2.1972245773	1.6094379124	4.8040210447	2.0055258587	0
100021914	X	A	3.0445224377	2.6390573296	1.7917594692	4.4998096703	3.1023420086	0
100021591	X	A	3.6571307558	2.7080502011	2.0794415417	4.8283137373	2.8903717579	0
100021294	X	A	3.5115454388	2.7080502011	2.0794415417	4.6728288345	2.9871959425	0
100021019	X	A	3.295836866	2.4849066498	1.9459101491	4.4067192473	2.8390784635	0
100020880	X	A	2.6026896854	2.5649493575	1.7917594692	4.189654742	2.2407096893	0
100010373	X	D	2.6390573296	2.4849066498	1.7917594692	3.7612001157	2.427454075	0
100020593	X	A	2.76000994	2.7080502011	1.9459101491	3.7376696183	2.6511270537	0

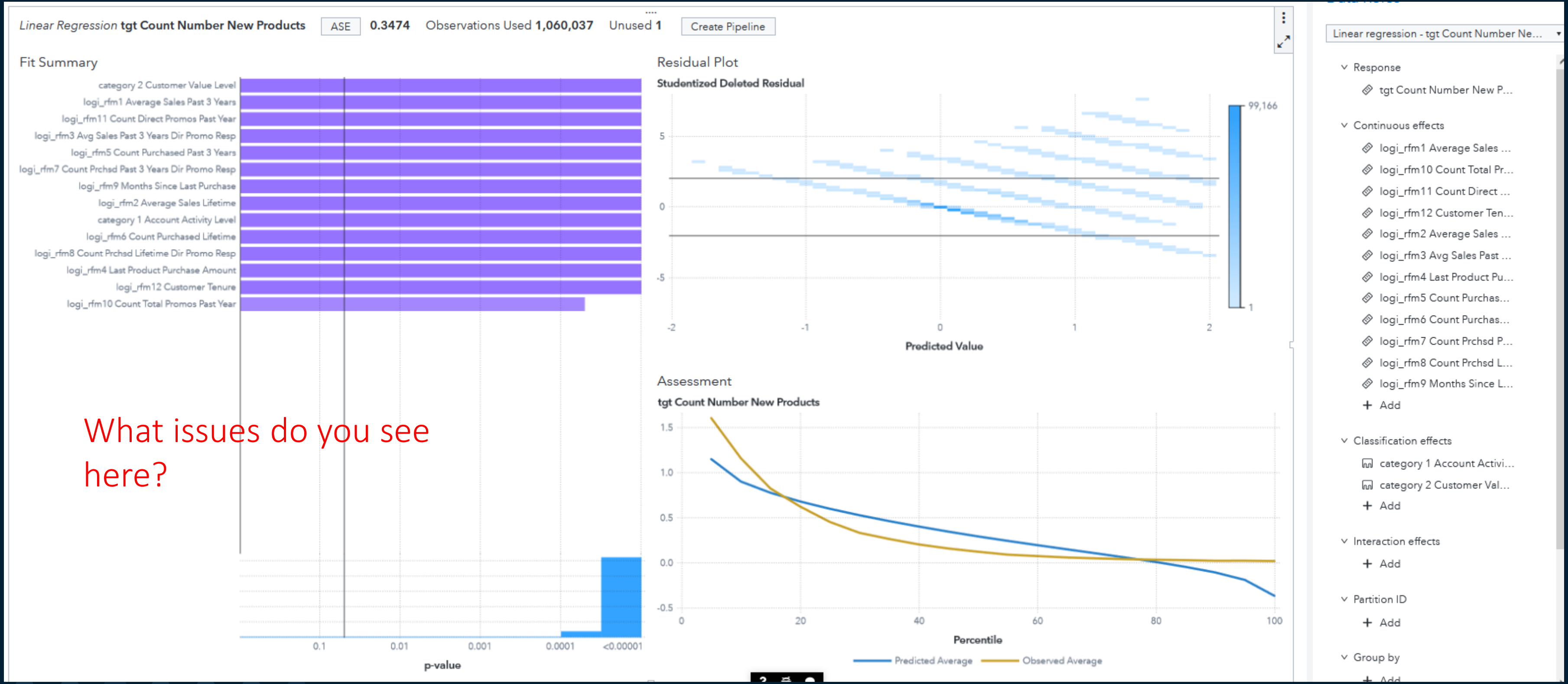
Example

Count Number New Products



Example

Linear Model



What issues do you see here?

Generalized Linear Models

- Extends the linear regression model by incorporating a "link function" and associated probability distribution to change the distribution of the response variable:

$$g(E(Y|X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

  - where $g(\cdot)$ can be a variety of functions with a variety of probability distributions

- One way of dealing with response variables that are not normally distributed with respect to the predictors

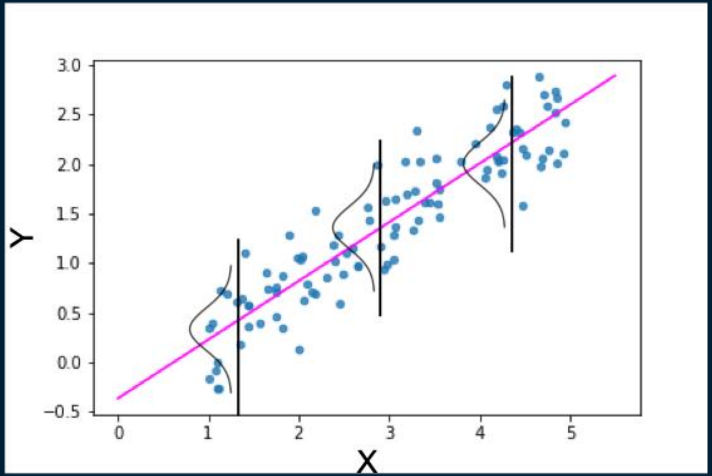
Generalized Linear Models

Linear Regression

Models the mean of a continuous response variable Y

- $g(\cdot)$ is distributed normally with an identity link function:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$



Generalized Linear Models

Binary Logistic Regression

Models the odds of "success" for a binary response variable Y

- $g(\cdot)$ is the logit function with a binomial distribution:

$$logit(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Generalized Linear Models

Poisson Regression

Models the mean of a discrete (count) response variable Y

- $g(\cdot)$ is the log link function with a Poisson distribution:

$$log\lambda_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Generalized Linear Models

Common Linear Distributions

Distribution	Available Link Functions (default listed first)
Beta	Logit, Probit, Log-log, C-log-log
Binary	Logit, Probit, Log-log, C-log-log
Exponential	Log, Identity
Gamma	Log, Identity, Reciprocal
Geometric	Log, Identity
Inverse Gaussian	Power(-2), Log, Identity
Negative Binomial	Log, Identity
Normal (default)	Identity, Log
Poisson	Log, Identity
Tweedie	Identity, Log

Examples of Popular GLMs

Response Variable	Distribution	Link Function	Variance Function
Continuous	Normal	Identity	σ^2
Binary	Binomial	Logit	$\mu(1-\mu)$
Count	Poisson	Log	μ