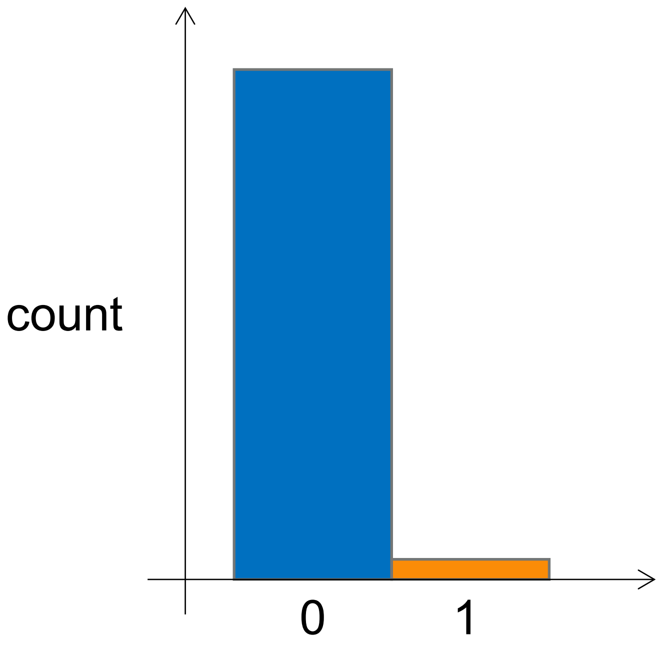
Logistic Regression

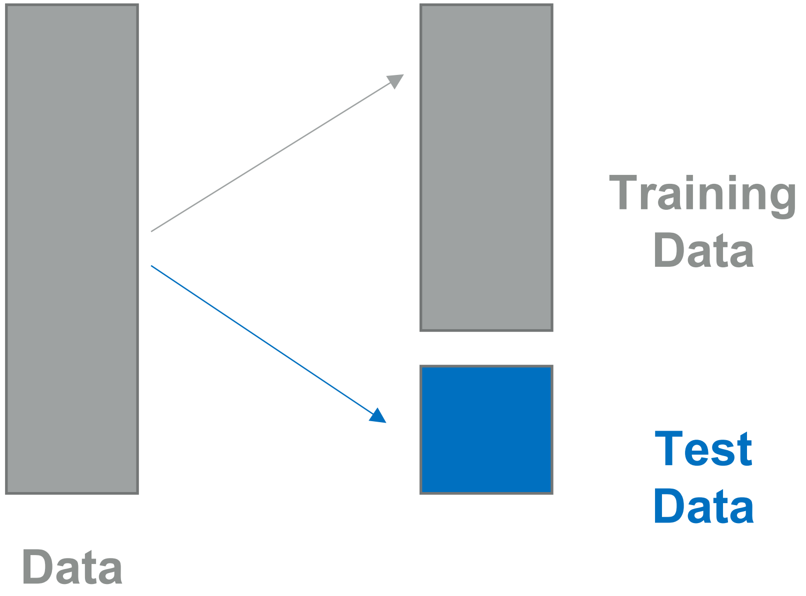
John Rios

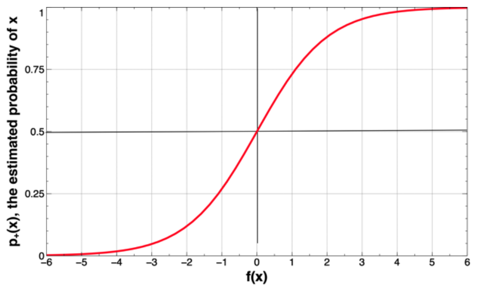
Business Intelligence

Terry College of Business
UNIVERSITY OF GEORGIA




```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  926 171
       Yes 106 202

               Accuracy : 0.8028
                 95% CI : (0.7811, 0.8234)
    No Information Rate : 0.7345
    P-Value [Acc > NIR] : 1.364e-09

                  Kappa : 0.4647

 Mcnemar's Test P-Value : 0.0001204

              Precision : 0.6558
                 Recall : 0.5416
                     F1 : 0.5932
             Prevalence : 0.2655
         Detection Rate : 0.1438
   Detection Prevalence : 0.2192
      Balanced Accuracy : 0.7194

       'Positive' Class : Yes
```



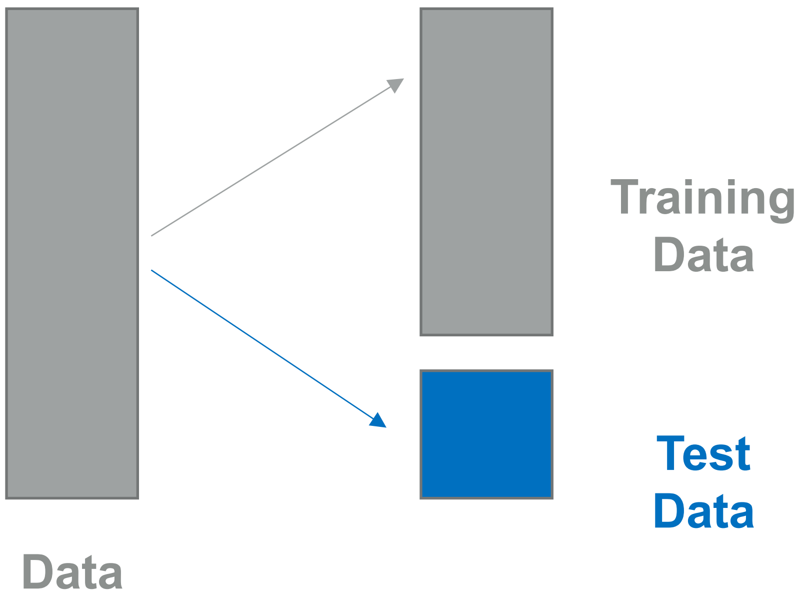


Machine Learning Use

The goal is to predict the target using a new dataset where we have values for predictors but not the target

Evaluate based on prediction error

- Build the model using training data
- Assess performance on test (hold-out) data



Model Evaluation

How well the model predicts new data (*not* how well it fits the data it was trained with)

- Key component of most measures is difference between actual outcome and predicted outcome (i.e., error)

Model Evaluation (*Regression*)

Error for data record = predicted (p) minus actual (a)

RMSE: Root Mean Squared Error
MAE: Mean Absolute Error
MAPE: Mean Absolute Percentage Error
Total SSE: Total Sum of Squared Errors

When the target is ***numeric*!**

Last Class…

Model Evaluation (*Classification*)

Accuracy = (true positives + true negatives) / total

Precision = true positives / (true positives + false positives)
Recall = true positives / (true positives + false negatives)

F1-measure = (2 * precision * recall) / (precision + recall)

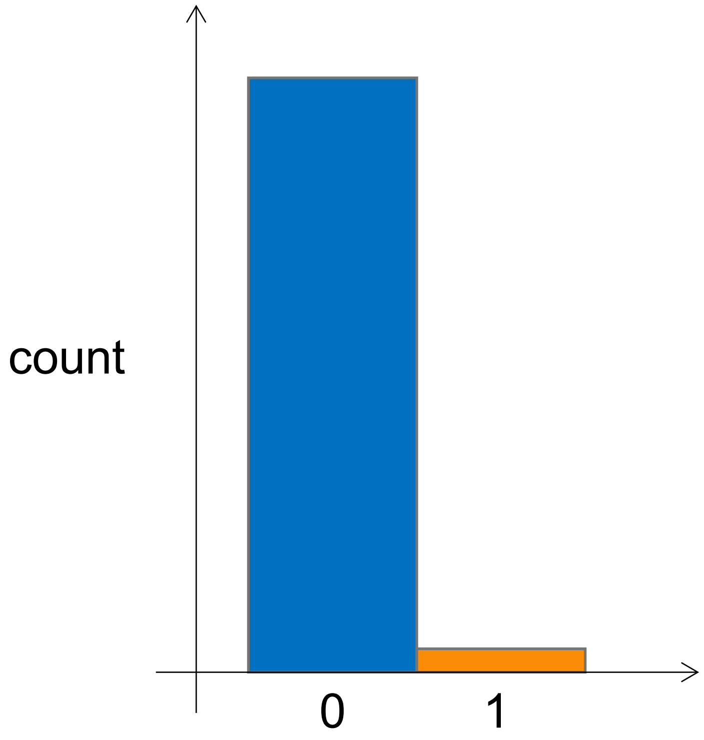
When the target is
a class!

Today!

Accuracy

Inappropriate for unbalanced (or skewed) classes

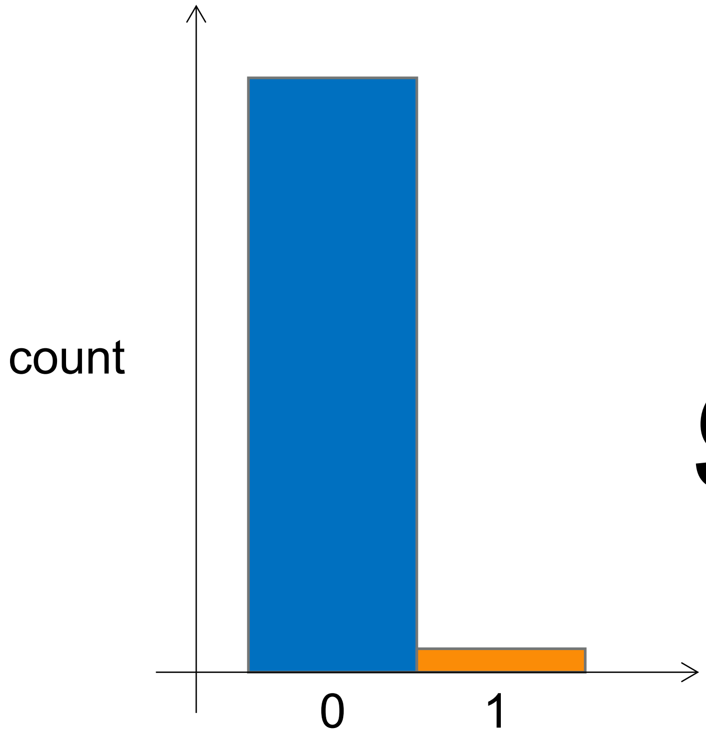
0 = no fraud
1 = yes fraud



Accuracy

Inappropriate for unbalanced (or skewed) classes

0 = no fraud
1 = yes fraud



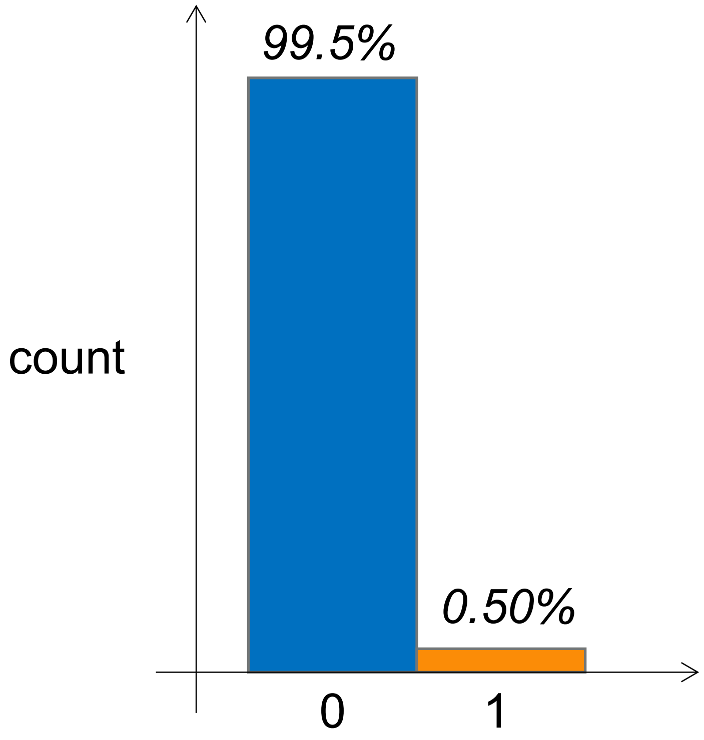
Train a logistic model and find that you have 0.8% error on test set

99.2% accurate!

Accuracy

Inappropriate for unbalanced (or skewed) classes

0 = no fraud
1 = yes fraud



Only 0.50% of transactions are fraudulent!

```
predict_fraud <- function(x){
  return(0)
}
```

99.5% accurate!

Precision / Recall

		Actual	
		Positive	**Negative**
Predicted	**Positive**	**True Positives**	**False Positives**
	Negative	**False Negatives**	**True Negatives**

Precision = True positives / *Predicted* positives

Recall = True positives / *Actual* positives

Precision (of all transactions where we predicted fraud, what fraction actually was fraud?)

Recall (of all transactions that actually were fraud, what fraction did we correctly detect as being fraud?)

Precision / Recall

		Actual	
		Positive	**Negative**
Predicted	**Positive**	True Positives (**45**)	False Positives (**75**)
	Negative	False Negatives (**5**)	True Negatives (**9875**)

Precision = True positives / *Predicted* positives

Recall = True positives / *Actual* positives

Precision (of all transactions where we predicted fraud, what fraction actually was fraud?)
45(TP) / [45(TP) + 75(FP)] = 0.375

Recall (of all transactions that actually were fraud, what fraction did we correctly detect as being fraud?)
45(TP) / [45(TP) + 5(FN)] = 0.9

F1-measure: (2 * 0.375 * 0.9) / (0.375 + 0.9) = 0.529

Accuracy: (45+9875)/10000 = 0.992

Precision / Recall

		Actual	
		Positive	**Negative**
Predicted	**Positive**	True Positives (0)	False Positives (0)
	Negative	False Negatives (**50**)	True Negatives (**9950**)

Precision = True positives / *Predicted* positives

Recall = True positives / *Actual* positives

Precision: 0(TP) / [0(TP) + 0(FP)] = undefined

Recall: 0(TP) / [0(TP) + 50(FN)] = 0

F1-measure: undefined

Accuracy: (9950)/10000 = 0.995

Precision / Recall

Useful metrics for evaluating performance when what we want to predict is rare (e.g., fraudulent transaction)

If the model has **high precision** and **high recall**, then we can be confident that the model is doing well even if we have very skewed classes

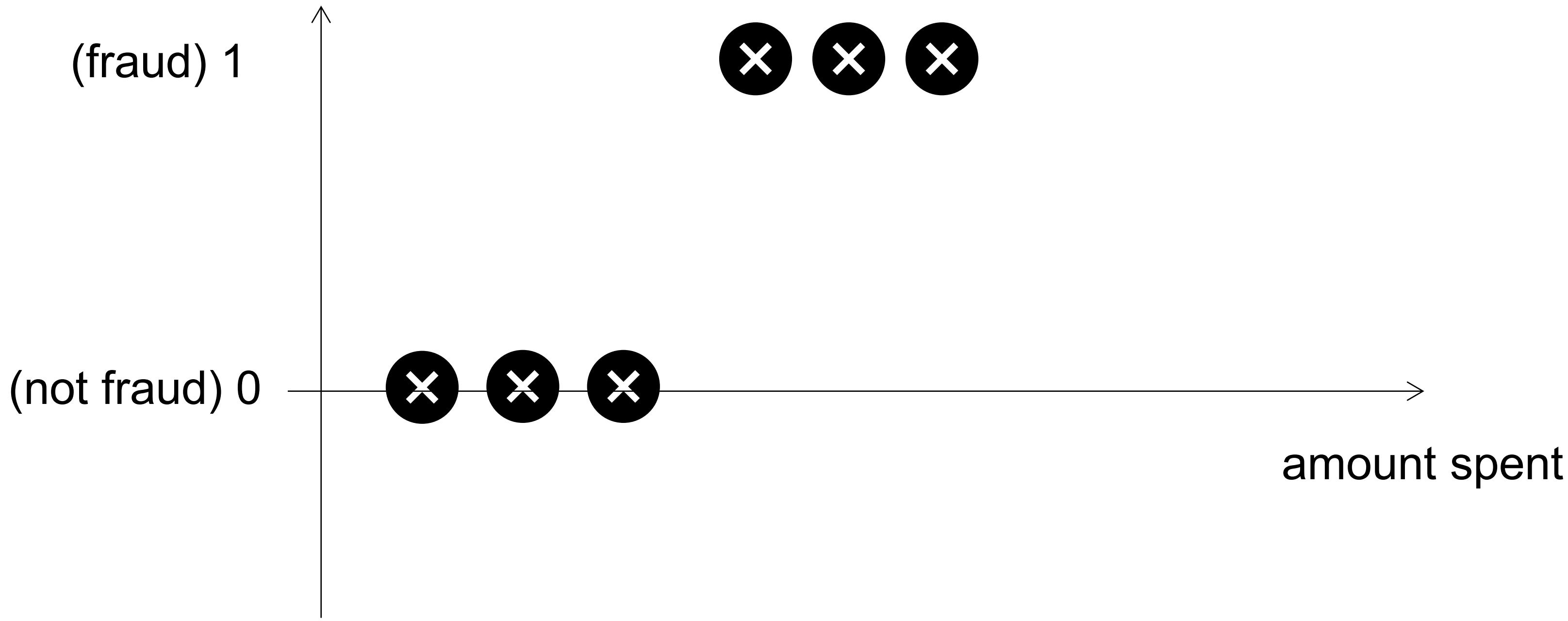
Method (or Model)

For regression, we started with a linear regression model and then experimented with random forest next

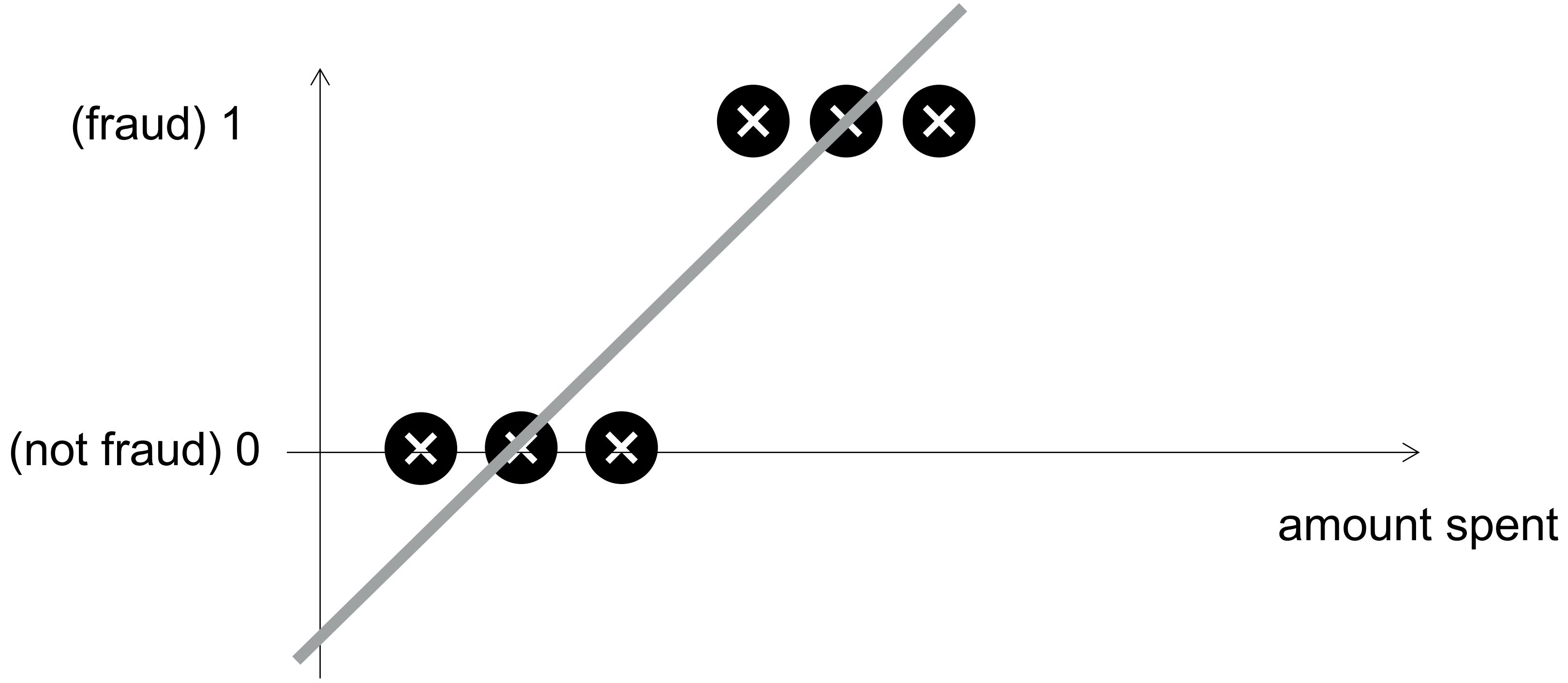
Can we do the same for **classification**?

We could, but it is not a good idea to start with a linear regression!

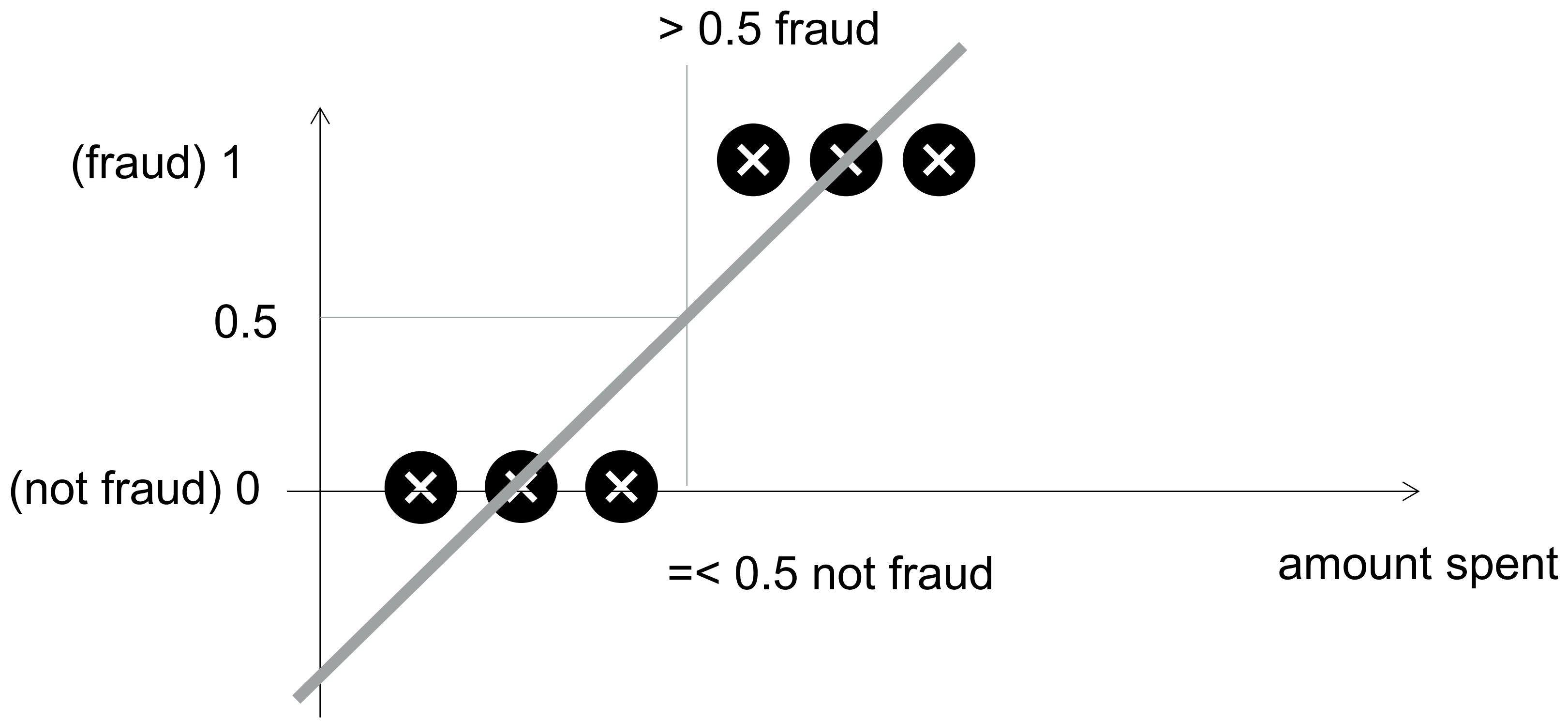
Method (or Model)



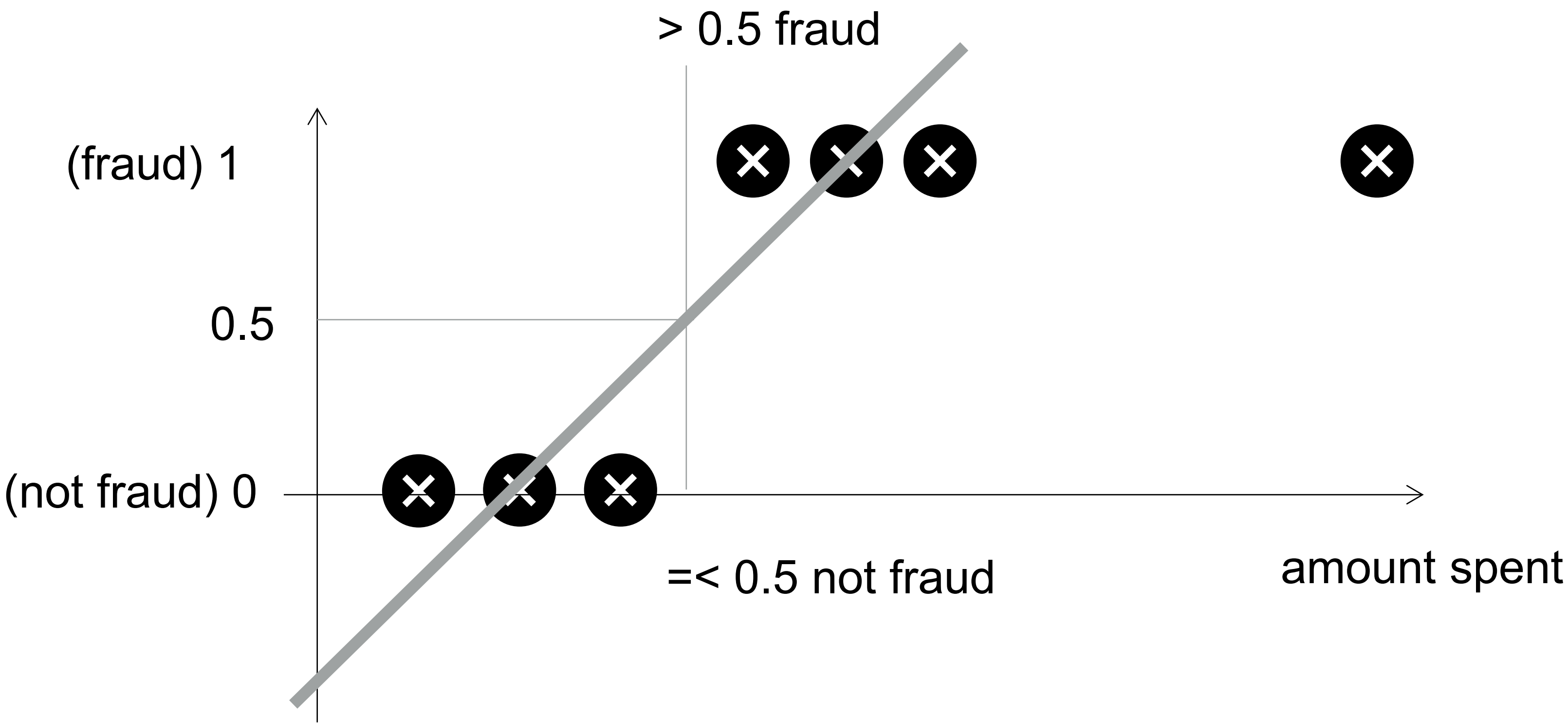
Method (or Model)



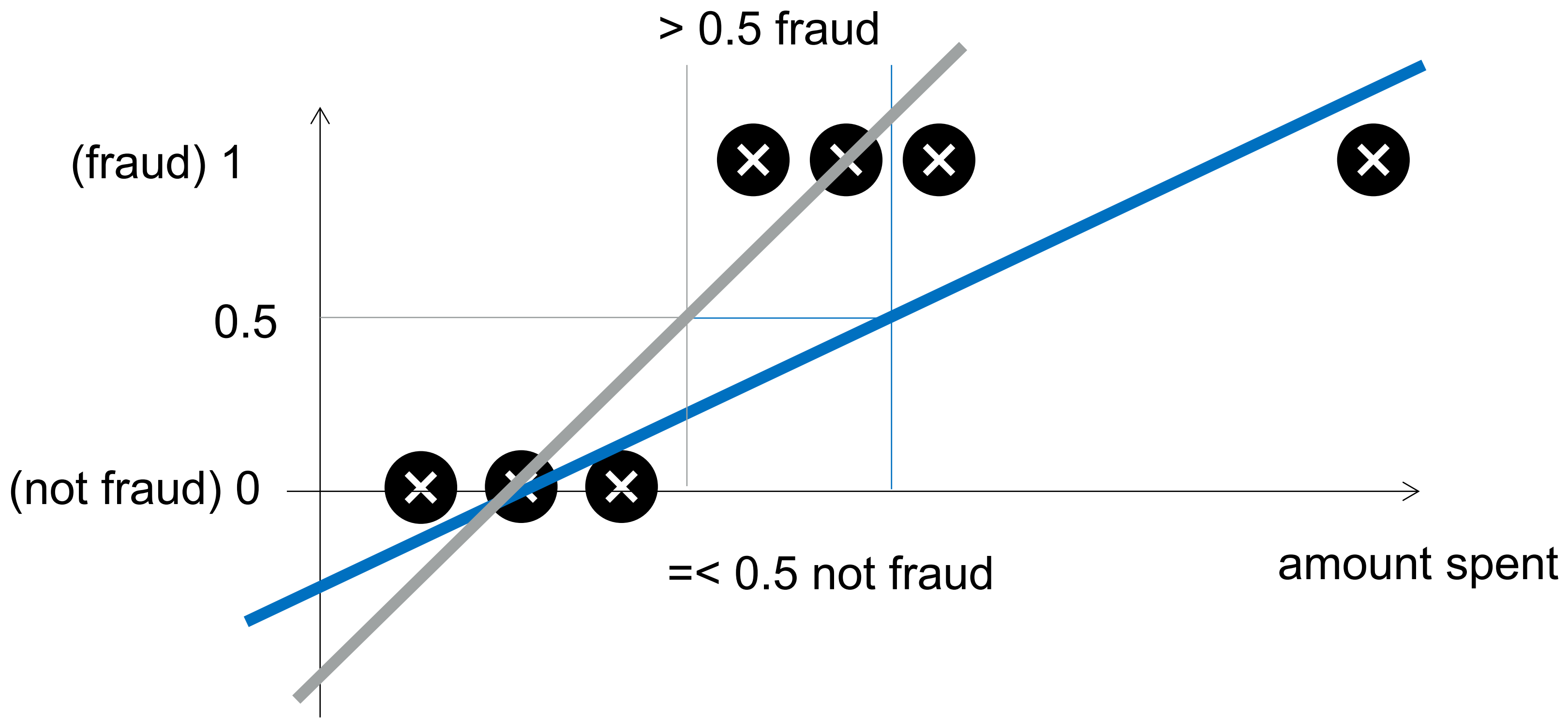
Method (or Model)



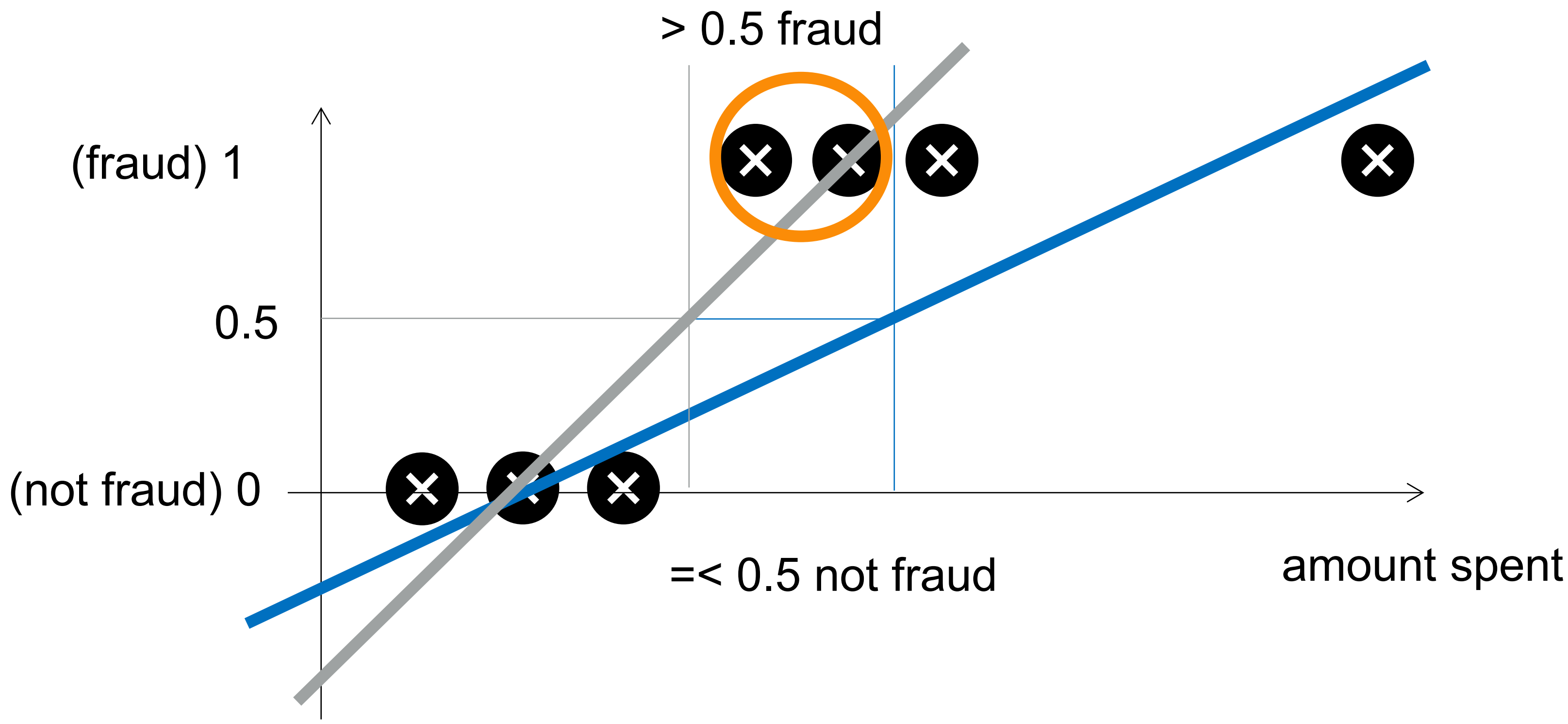
Method (or Model)



Method (or Model)



Method (or Model)



Another Note

We know that a linear regression can output values > 1 or < 0

But it is kind of weird to have such possibility when we know that the target is either 1 or 0

What to do?

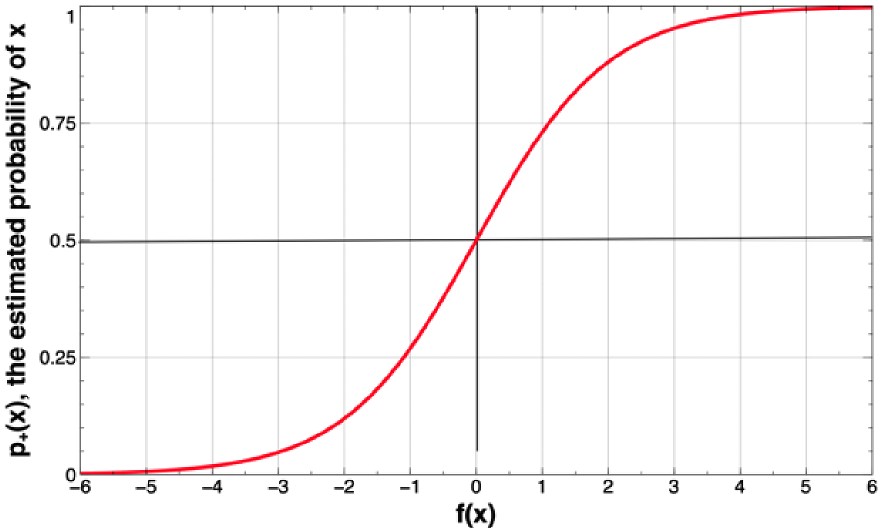
Logistic Regression

Start with logistic regression, a very popular model(simple and fast) that will produce output values (predicted scores) between 0 and 1

Don’t be confused by terminology, logistic regression has the term “regression” in it for historical reasons, but it is used in ML for **classification**

Logistic Regression

Constructed to maximize the probability of correct classification



The model function

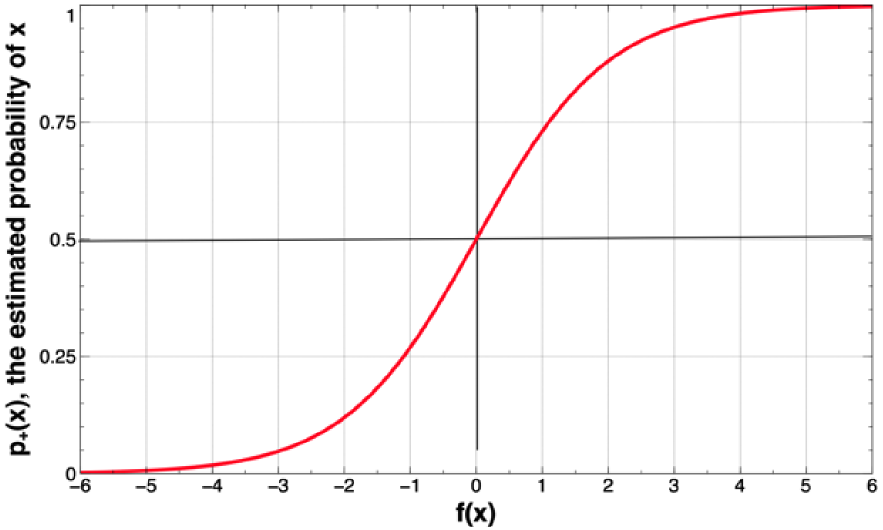
$$\log\left(\frac{p}{1-p}\right) = \boldsymbol{\alpha} + \boldsymbol{\beta x}$$

$$\log(odds\ ratio) = \boldsymbol{\alpha} + \boldsymbol{\beta x}$$

p = probability of class membership
α = log odds of positive class when all predictors are zero
β = the effect of the predictor on the odds ratio
x = predictor

Logistic Regression

Probability	Odds ratio	log(odds ratio)
0.5	50:50 or 1	0
0.9	90:10 or 9	2.19
0.999	999:1 or 999	6.9
0.01	1:99 or 0.0101	-4.6
0.001	1:999 or 0.001001	-6.9



The odds ratio is the relative chance of an event taking place (OR > 1 more likely, OR < 1 less likely, OR = 1 equally likely)

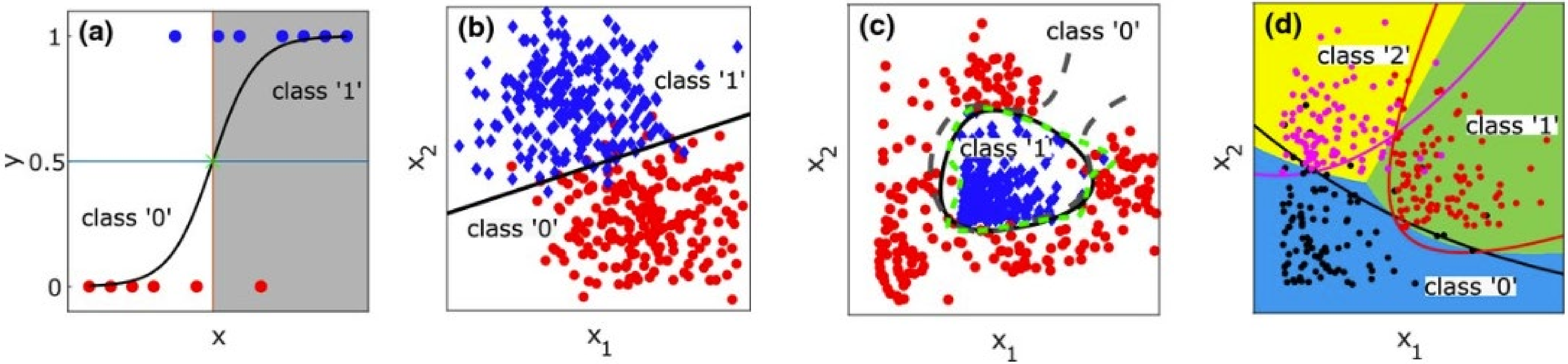
Example

the β value for each predictor variable indicates the effect of that predictor on the odds ratio. For example, if the β for the flu shot is negative, then getting a flu shot decreases the estimated probability of getting sick,

Flu Shot	Vitamin C intake	Sleep	Sick?
1	1000	7	0
1	500	5	1
0	700	8	1
0	1100	8.5	0
1	600	7	0
0	500	6	1
			…
1	800	6	0

predictor

target



De Paula, et. al., (2021). Mitigation of nonlinear phase noise in single-channel coherent 16-QAM systems employing logistic regression. Optical and Quantum Electronics 53(9)

ML Classification in R

Use the the **`caret`** package

Telco Customer Churn – recall, *the goal is to predict the target using a new dataset as best as we can*

```
library(tidyverse)
library(caret)

churn <- read_csv("churn.csv")
```

Customer Churn Rate

A communications company has seen an increase in its rate of attrition or customer churn. As a BI analyst in the company, you have been asked to predict the probability that a customer discontinues their subscription. Your manager shared a dataset with you, and you need to do the following:

- Transform the text variables into numerical variables (e.g., dummy variables)
- Use machine learning and logistic regression to predict the probability that a customer may stop his/her subscription.

Churn Data

Dataset

1494

Telco Customer Churn

Focused customer retention programs

BlastChar • updated 3 years ago (Version 1)

Data | Tasks (1) | Code (564) | Discussion (11) | Activity | Metadata

Download (955 KB) | New Notebook

Usability 8.8 | **License** Data files © Original Authors | **Tags** business

Description

Context

"Predict behavior to retain customers. You can analyze all relevant customer data and develop focused customer retention programs." [IBM Sample Data Sets]

Content

Each row represents a customer, each column contains customer's attributes described on the column Metadata.

The data set includes information about:

- Customers who left within the last month – the column is called Churn
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies

ML Classification in R

Filter

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies
1	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	No	No	No
2	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No	No	No
3	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No	No
4	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	Yes	No	No
5	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No	No	No
6	9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No	No	Yes	No	Yes	Yes
7	1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	Yes	No	No	Yes	No
8	6713-OKOMC	Female	0	No	No	10	No	No phone service	DSL	Yes	No	No	No	No	No
9	7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	No	Yes	Yes	Yes	Yes
10	6388-TABGU	Male	0	No	Yes	62	Yes	No	DSL	Yes	Yes	No	No	No	No
11	9763-GRSKD	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	No	No	No	No	No
12	7469-LKBCI	Male	0	No	No	16	Yes	No	No	No internet service	No internet service	No internet service	No internet service	No internet service	No internet service
13	8091-TTVAX	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No	No	Yes	No	Yes	Yes
14	0280-XJGEX	Male	0	No	No	49	Yes	Yes	Fiber optic	No	Yes	Yes	No	Yes	Yes
15	5129-JLPIS	Male	0	No	No	25	Yes	No	Fiber optic	Yes	No	Yes	Yes	Yes	Yes
16	3655-SNQYZ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes
17	8191-XWSZG	Female	0	No	No	52	Yes	No	No	No internet service	No internet service	No internet service	No internet service	No internet service	No internet service
18	9959-WOFKT	Male	0	No	Yes	71	Yes	Yes	Fiber optic	Yes	No	Yes	No	Yes	Yes
19	4190-MFLUW	Female	0	Yes	Yes	10	Yes	No	DSL	No	No	Yes	Yes	No	No
20	4183-MYFRB	Female	0	No	No	21	Yes	No	Fiber optic	No	Yes	Yes	No	No	Yes
21	8779-QRDMV	Male	1	No	No	1	No	No phone service	DSL	No	No	Yes	No	No	Yes
22	1680-VDCWW	Male	0	Yes	No	12	Yes	No	No	No internet service	No internet service	No internet service	No internet service	No internet service	No internet service
23	1066-JKSGK	Male	0	No	No	1	Yes	No	No	No internet service	No internet service	No internet service	No internet service	No internet service	No internet service
24	3638-WEABW	Female	0	Yes	No	58	Yes	Yes	DSL	No	Yes	No	Yes	No	No
25	6322-HRPFA	Male	0	Yes	Yes	49	Yes	No	DSL	Yes	Yes	No	Yes	No	No
26	6865-JZNKO	Female	0	No	No	30	Yes	No	DSL	Yes	Yes	No	No	No	No
27	6467-CHFZW	Male	0	Yes	Yes	47	Yes	Yes	Fiber optic	No	Yes	No	No	Yes	Yes
28	8665-UTDHZ	Male	0	Yes	Yes	1	No	No phone service	DSL	No	Yes	No	No	No	No
29	5248-YGIJN	Male	0	Yes	No	72	Yes	Yes	DSL	Yes	Yes	Yes	Yes	Yes	Yes
30	8773-HHUOZ	Female	0	No	Yes	17	Yes	No	DSL	No	No	No	No	Yes	Yes
31	3841-NFECX	Female	1	Yes	No	71	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	No	No

	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
1	Month-to-month	Yes	Electronic check	29.85	29.85	No
2	One year	No	Mailed check	56.95	1889.50	No
3	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
4	One year	No	Bank transfer (automatic)	42.30	1840.75	No
5	Month-to-month	Yes	Electronic check	70.70	151.65	Yes
6	Month-to-month	Yes	Electronic check	99.65	820.50	Yes
7	Month-to-month	Yes	Credit card (automatic)	89.10	1949.40	No
8	Month-to-month	No	Mailed check	29.75	301.90	No
9	Month-to-month	Yes	Electronic check	104.80	3046.05	Yes
10	One year	No	Bank transfer (automatic)	56.15	3487.95	No
11	Month-to-month	Yes	Mailed check	49.95	587.45	No
12	Two year	No	Credit card (automatic)	18.95	326.80	No
13	One year	No	Credit card (automatic)	100.35	5681.10	No
14	Month-to-month	Yes	Bank transfer (automatic)	103.70	5036.30	Yes
15	Month-to-month	Yes	Electronic check	105.50	2686.05	No
16	Two year	No	Credit card (automatic)	113.25	7895.15	No
17	One year	No	Mailed check	20.65	1022.95	No
18	Two year	No	Bank transfer (automatic)	106.70	7382.25	No
19	Month-to-month	No	Credit card (automatic)	55.20	528.35	Yes
20	Month-to-month	Yes	Electronic check	90.05	1862.90	No
21	Month-to-month	Yes	Electronic check	39.65	39.65	Yes
22	One year	No	Bank transfer (automatic)	19.80	202.25	No
23	Month-to-month	No	Mailed check	20.15	20.15	Yes
24	Two year	Yes	Credit card (automatic)	59.90	3505.10	No
25	Month-to-month	No	Credit card (automatic)	59.60	2970.30	No
26	Month-to-month	Yes	Bank transfer (automatic)	55.30	1530.60	No
27	Month-to-month	Yes	Electronic check	99.35	4749.15	Yes
28	Month-to-month	No	Electronic check	30.20	30.20	Yes
29	Two year	Yes	Credit card (automatic)	90.25	6369.45	No
30	Month-to-month	Yes	Mailed check	64.70	1093.10	Yes
31	Two year	Yes	Credit card (automatic)	96.35	6766.95	No

Showing 1 to 31 of 7,043 entries, 21 total columns

Transform the following variables to numeric

Add an N at the end of the new variable name to identify that the variable is numeric, e.g., PartnerN

- Partner (1, 0)
- Dependents (1, 0)
- PhoneService (1, 0)
- MultipleLines (1, 0)
- InternetService (dummy variables)
- OnlineSecurity (1,0)
- OnlineBackup (1, 0)
- DeviceProtection (1,0)
- TechSupport (1, 0)
- StreamingTV (1, 0)
- StreamingMovies (1, 0)
- Contract (1, 0)
- PaperlessBilling (1, 0)
- PaymentMethod (1, 0)
- Churn (1, 0)

Data Manipulation

To run the model, we need numeric values.

```r
library(fastDummies)
# transform categories to numbers
churn <- churn %>%
  mutate(genderN = case_when(
    gender == "Male" ~ 1,
    gender == "Female" ~ 0
  )) %>%
  mutate(PartnerN = case_when(
    Partner == "Yes" ~ 1,
    Partner == "No" ~ 0
  )) %>%
  mutate(DependentsN = case_when(
    Dependents == "Yes" ~ 1,
    Dependents == "No" ~ 0
  )) %>%
  mutate(PhoneServiceN = case_when(
    PhoneService == "Yes" ~ 1,
    PhoneService == "No" ~ 0
  )) %>%
  mutate(MultipleLinesN = case_when(
    MultipleLines == "Yes" ~ 1,
    MultipleLines == "No" ~ 0,
    MultipleLines == "No phone service" ~ 0
  )) %>%
  dummy_cols(., select_columns =
               'InternetService'
  )%>%
  mutate(OnlineSecurityN = case_when(
    OnlineSecurity == "Yes" ~ 1,
    OnlineSecurity == "No" ~ 0,
    OnlineSecurity == "No internet service" ~ 0
```

```r
mutate(DeviceProtectionN = case_when(
  DeviceProtection == "Yes" ~ 1,
  DeviceProtection == "No" ~ 0,
  DeviceProtection == "No internet service" ~ 0
)) %>%
mutate(TechSupportN = case_when(
  TechSupport == "Yes" ~ 1,
  TechSupport == "No" ~ 0,
  TechSupport == "No internet service" ~ 0
)) %>%
mutate(StreamingTVN = case_when(
  StreamingTV == "Yes" ~ 1,
  StreamingTV == "No" ~ 0,
  StreamingTV == "No internet service" ~ 0
)) %>%
mutate(StreamingMoviesN = case_when(
  StreamingMovies == "Yes" ~ 1,
  StreamingMovies == "No" ~ 0,
  StreamingMovies == "No internet service" ~ 0
)) %>%
mutate(ContractN = case_when(
  Contract == "Month-to-month" ~ 0,
  Contract == "One year" ~ 1,
  Contract == "Two year" ~ 1
)) %>%
mutate(PaperlessN = case_when(
  PaperlessBilling == "Yes" ~ 1,
  PaperlessBilling == "No" ~ 0
)) %>%
mutate(PaymentN = case_when(
  PaymentMethod == "Electronic check" ~ 0,
  PaymentMethod == "Mailed check" ~ 0,
  PaymentMethod == "Bank transfer (automatic)" ~ 1,
```

Data Manipulation

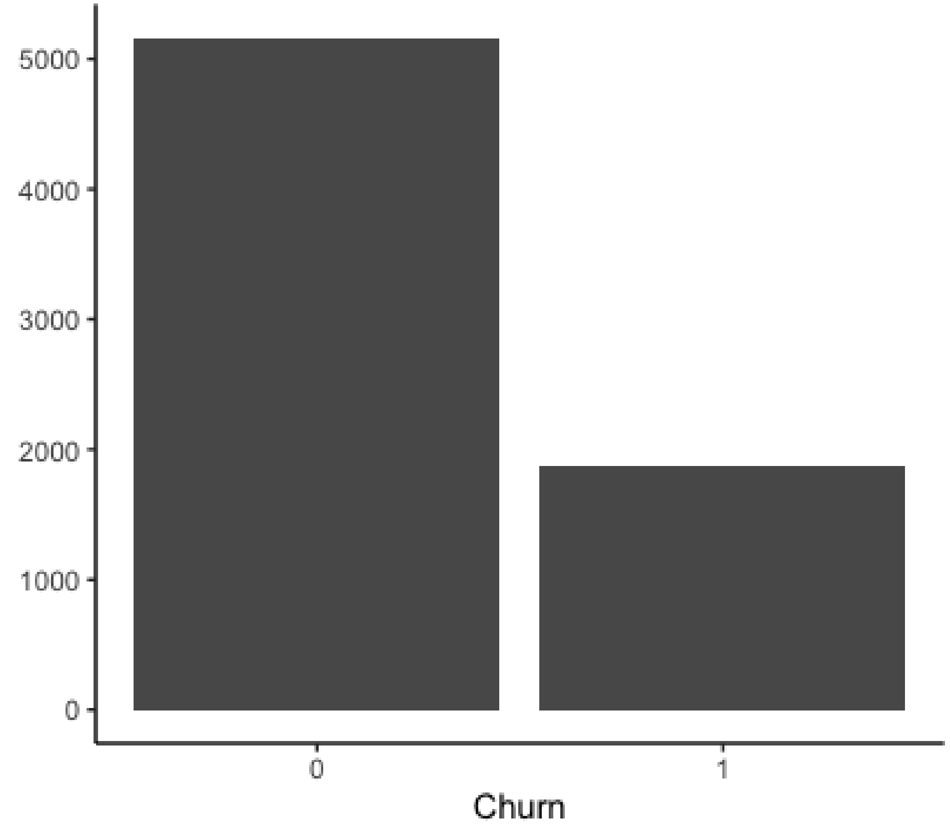
To run the model, we need numeric values.

```
# only select numeric variables
df <- churn %>% dplyr::select(Churn, ChurnN, SeniorCitizen, tenure,
                              MonthlyCharges, TotalCharges, genderN:PaymentN)

# drop missing values NAs
df1 <- drop_na(df)
```

Check: Class Distribution

Is the target skewed?

```
# is the target skewed?
ggplot(df1, aes(ChurnN)) +
  geom_bar() +
  theme_classic() +
  labs(x = "Churn", y = NULL) +
  scale_x_continuous(breaks = c(0,1))
```



Splitting Data

Set a seed value so that results are reproducible
Split the data into training and testing

```
# transform target into a factor
df1$Churn <- as.factor(df1$Churn)

set.seed(12L) # set a starting seed to be able to get reproducible results

# partition data
trainIndex <- createDataPartition(df1$Churn, # target variable
                                  p = 0.8, # percentage that goes to training
                                  list = FALSE, # results will not be in a list
                                  times = 1) # number of partitions to create

churn_train <- df1[trainIndex, ] # data frame for training
churn_test <- df1[-trainIndex, ] # data frame for testing
```

Selecting Predictors

To compute the correlation, we need numeric values

```
# compute the correlation between predictors and the target
predTargetCor <- cor(churn_train[,2:23])
```

	ChurnN
ContractN	-0.412521352
tenure	-0.364702823
InternetService_No	-0.229404000
PaymentN	-0.216936773
TotalCharges	-0.210396458
OnlineSecurityN	-0.171199823
TechSupportN	-0.170782623
DependentsN	-0.154602867
PartnerN	-0.145129210
InternetService_DSL	-0.115749460
OnlineBackupN	-0.098013503
DeviceProtectionN	-0.071942497
genderN	0.006471311
PhoneServiceN	0.007138488
MultipleLinesN	0.036112029
StreamingMoviesN	0.057989626
StreamingTVN	0.059047981
SeniorCitizen	0.143192896
PaperlessN	0.185933833
MonthlyCharges	0.186248242
InternetService_Fiber optic	0.300759774
ChurnN	1.000000000

Model Induction and Testing

Use training set to build model, then predict churn using the test set

```
model <- train(Churn ~ SeniorCitizen + PaperlessN + MonthlyCharges +

                 ContractN + tenure + InternetService_No + PaymentN +
                 TotalCharges + OnlineSecurityN + TechSupportN +
                 DependentsN + PartnerN + InternetService_DSL,

               data = churn_train, # use training set
               method = "glm") # simple additive logistic regression

# now predict outcomes in test set
p <- predict(model, churn_test, type = 'raw')

# add predictions to initial dataset
churn_test$pred_churn <- p
```

Model Performance

Use training set to build model, then predict churn using the test set

```
# how did we do? confusion matrix
confusionMatrix(data = churn_test$pred_churn,
                reference = churn_test$Churn,
                mode = "prec_recall",
                positive = "Yes")
```

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  926 171
       Yes 106 202

               Accuracy : 0.8028
                 95% CI : (0.7811, 0.8234)
    No Information Rate : 0.7345
    P-Value [Acc > NIR] : 1.364e-09

                  Kappa : 0.4647

 Mcnemar's Test P-Value : 0.0001204

              Precision : 0.6558
                 Recall : 0.5416
                     F1 : 0.5932
             Prevalence : 0.2655
         Detection Rate : 0.1438
   Detection Prevalence : 0.2192
      Balanced Accuracy : 0.7194

       'Positive' Class : Yes
```

- Of all customers where we predicted churn, ~66% actually churned
- Of all customers that actually churned, we only correctly predicted about half (~54%)

At-home exercise

- Experiment with different models to check and see if your model performance changes. A couple of popular options to try out are:
  - *k*-Nearest neighbors
  - Decision Trees
  - Support Vector Machines
  - Naïve Bayes

Summary

- Classification ML is when the target is a class (e.g., “yes" or “no”). Here, start with logistic regression rather than linear regression to try and maximize the probability of correct classification

- If the class distribution of the target is skewed (e.g., a lot more 0s than 1s), look for precision and recall in addition to accuracy in order to evaluate the performance of the model

- Other rules still apply: transform data, split sample, select features, train the model, and test performance