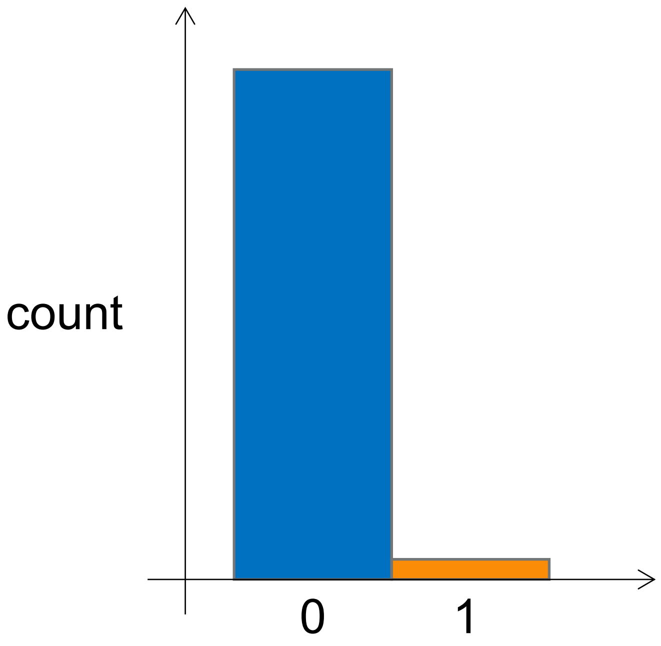
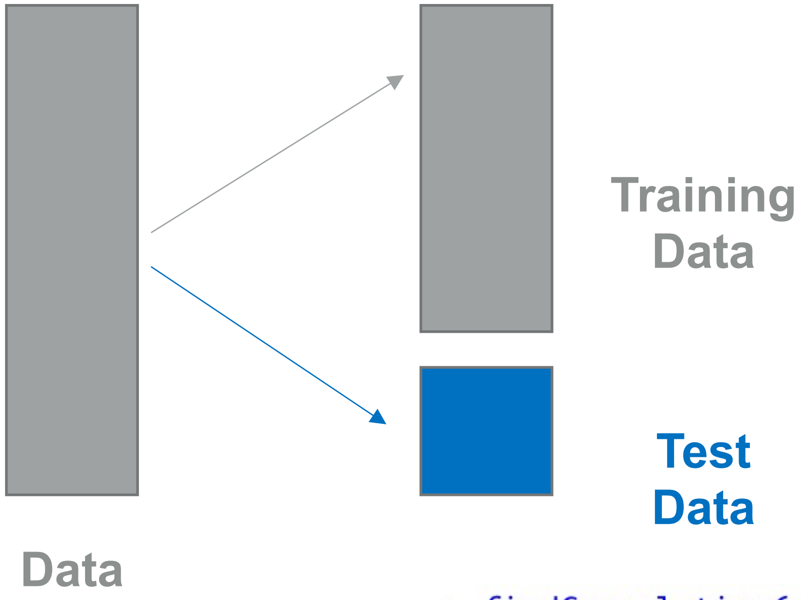
# Model Fitting III

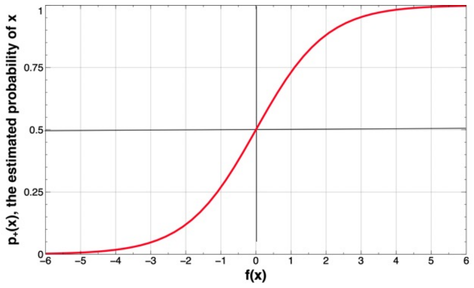
Carolina A. de Lima Salge
Assistant Professor
Terry College of Business
University of Georgia

*Business Intelligence*
*Spring 2021*




```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  925 176
       Yes 107 197

               Accuracy : 0.7986
                 95% CI : (0.7766, 0.8193)
    No Information Rate : 0.7345
    P-Value [Acc > NIR] : 1.342e-08

                  Kappa : 0.4511

 Mcnemar's Test P-Value : 5.296e-05

              Precision : 0.6480
                 Recall : 0.5282
                     F1 : 0.5820
             Prevalence : 0.2655
         Detection Rate : 0.1402
   Detection Prevalence : 0.2164
      Balanced Accuracy : 0.7122

       'Positive' Class : Yes
```




```
> findCorrelation(predCor, cutoff =  .7, names = TRUE)
[1] "TotalCharges"    "MonthlyCharges"
```

# Machine Learning Use

Predictive modeling

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

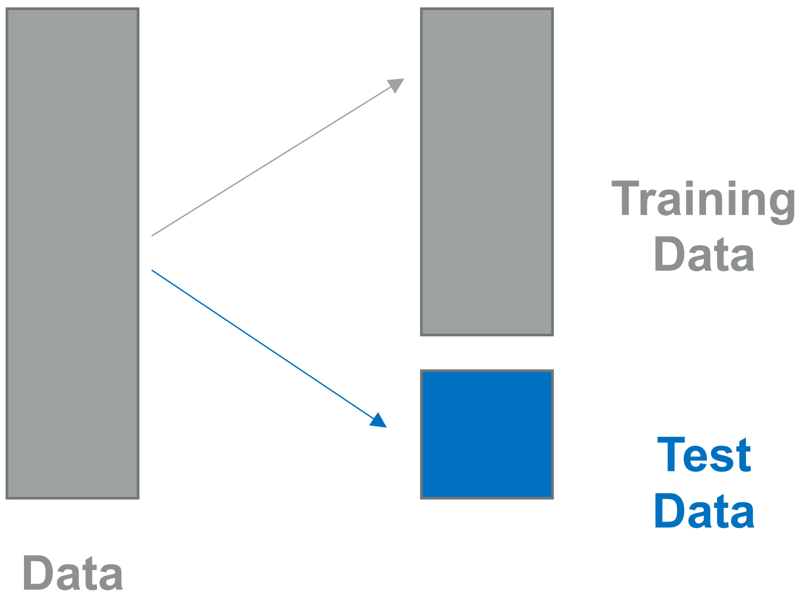


Provost, F. & Fawcett, T. (2013). Data Science for Business.

# Machine Learning Use

Evaluate based on prediction error

- Build 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)

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

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

**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

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

**Today!**

When the target is
***a class*!**

# Model Evaluation (*Classification*)

**Accuracy = true positives + true negatives / total**

When the target is ***a class*!**

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

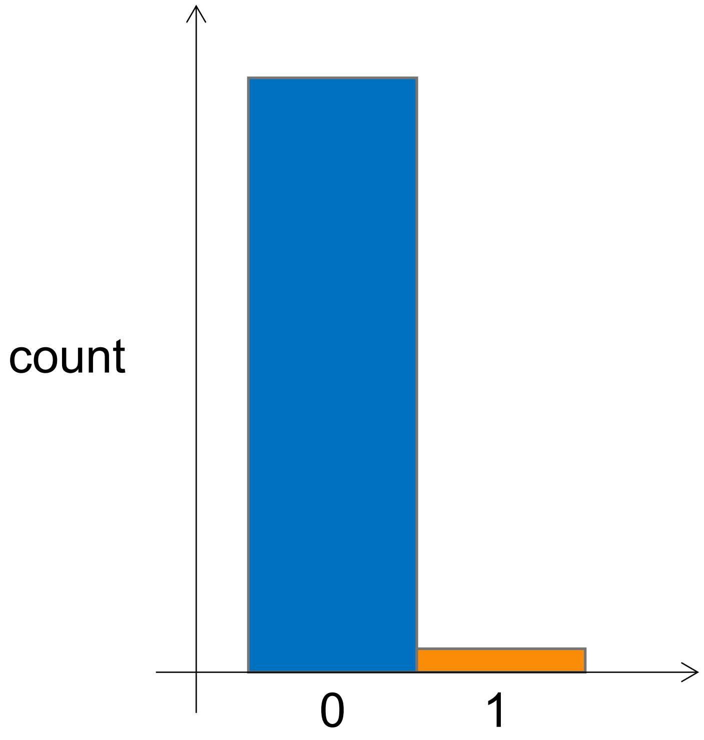
F-measure = (2 * precision * recall) / (precision + recall)

**Today!**

# Accuracy

Inappropriate for imbalanced (or skewed) classes

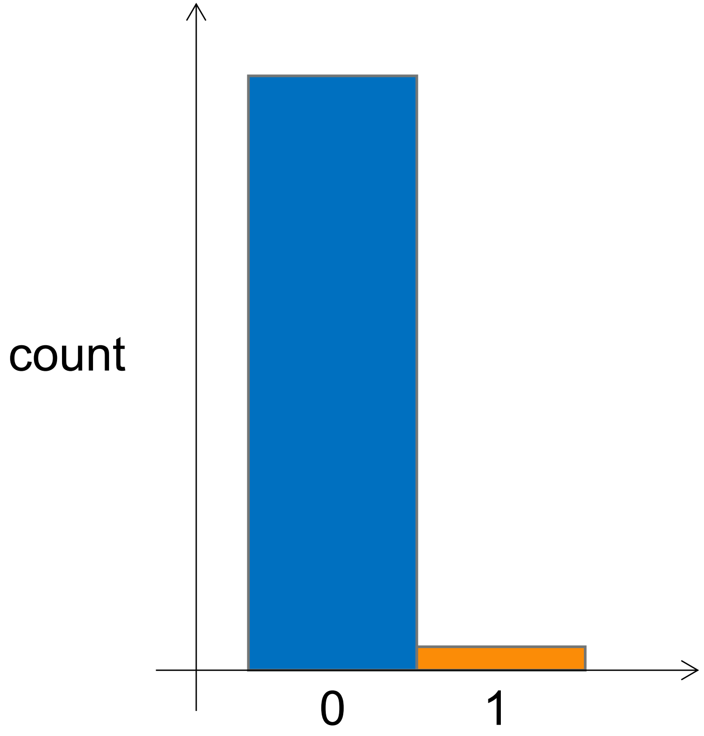
0 = no fraud
1 = yes fraud



# Accuracy

Inappropriate for imbalanced (or skewed) classes

0 = no fraud
1 = yes fraud



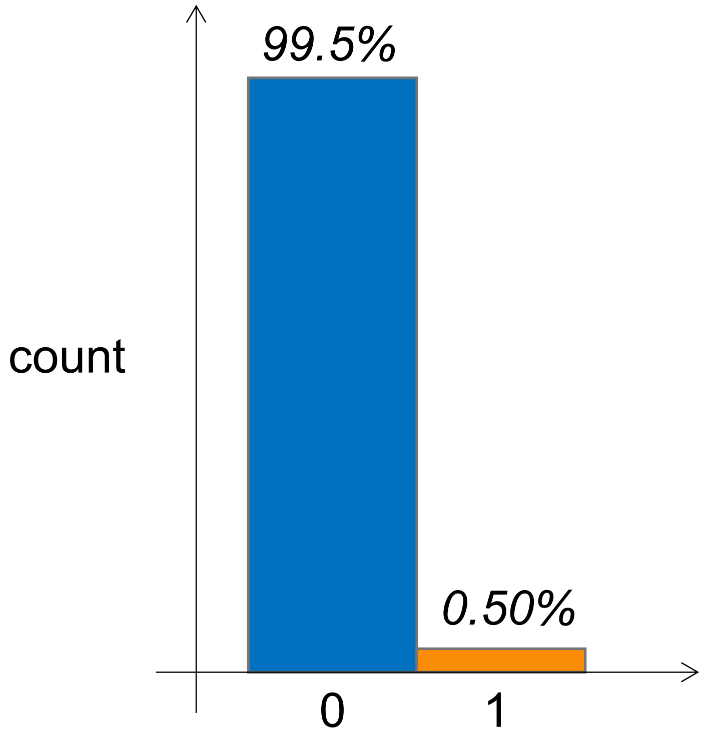

*Train a logistic model and find that you have 1% error on test set*

*99% accurate!*

# Accuracy

Inappropriate for imbalanced (or skewed) classes

0 = no fraud
1 = yes fraud




*Only 0.50% of transactions are fraudulent!*

```
function y = predictFraud(x)
    y = 0; %ignore x!
return
```

*99.5% accurate!*

# Precision / Recall

| | Actual 1 | Actual 0 |
|---|---|---|
| **Predicted 1** | *True positives* | *False positives* |
| **Predicted 0** | *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 1 | Actual 0 |
|---|---|---|
| **Predicted 1** | *True positives* ***(20)*** | *False positives* ***(1)*** |
| **Predicted 0** | *False negatives* ***(5)*** | *True negatives* ***(100)*** |

**Precision = True positives / *Predicted* positives**

**Recall = True positives / *Actual* positives**

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

- 20 (TP) / 20 (TP) + 1 (FP) = **95.24%**

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

- 20 (TP) / 20 (TP) + 5 (FN) = **80%**

# 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

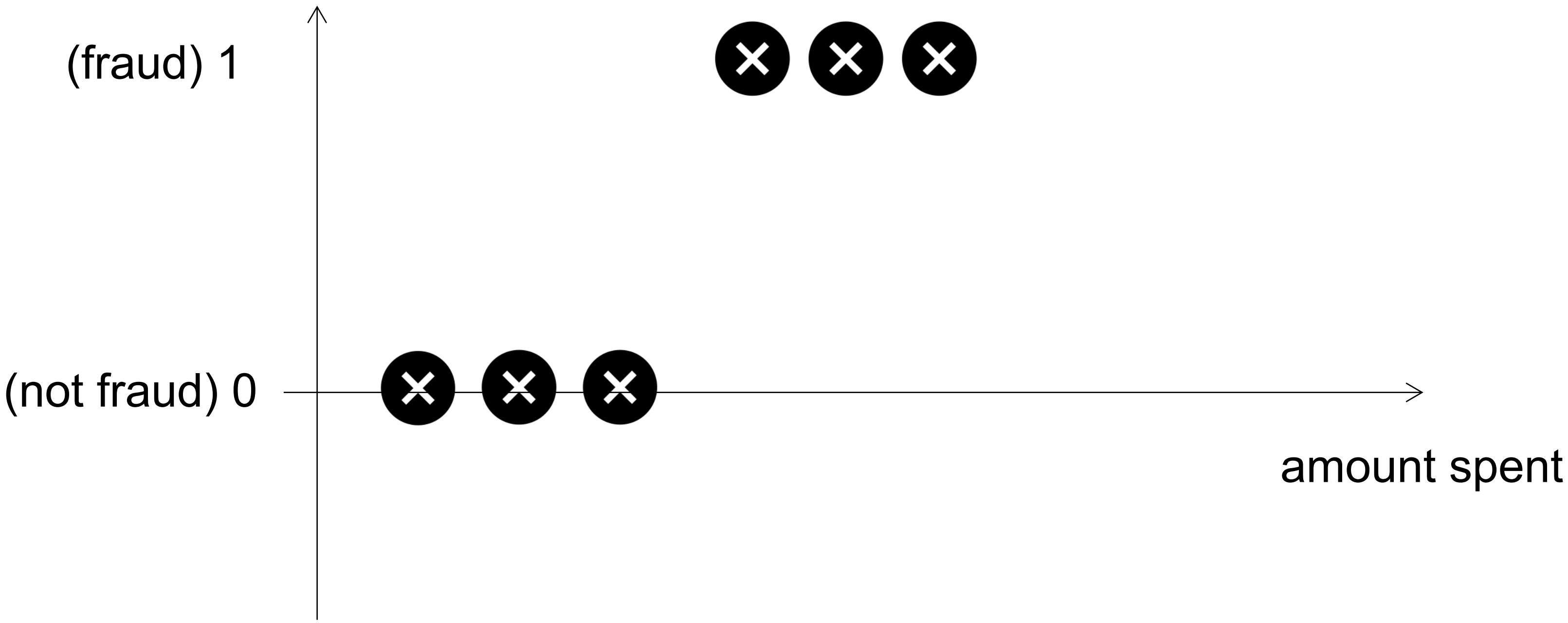
# Method (or Model)

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

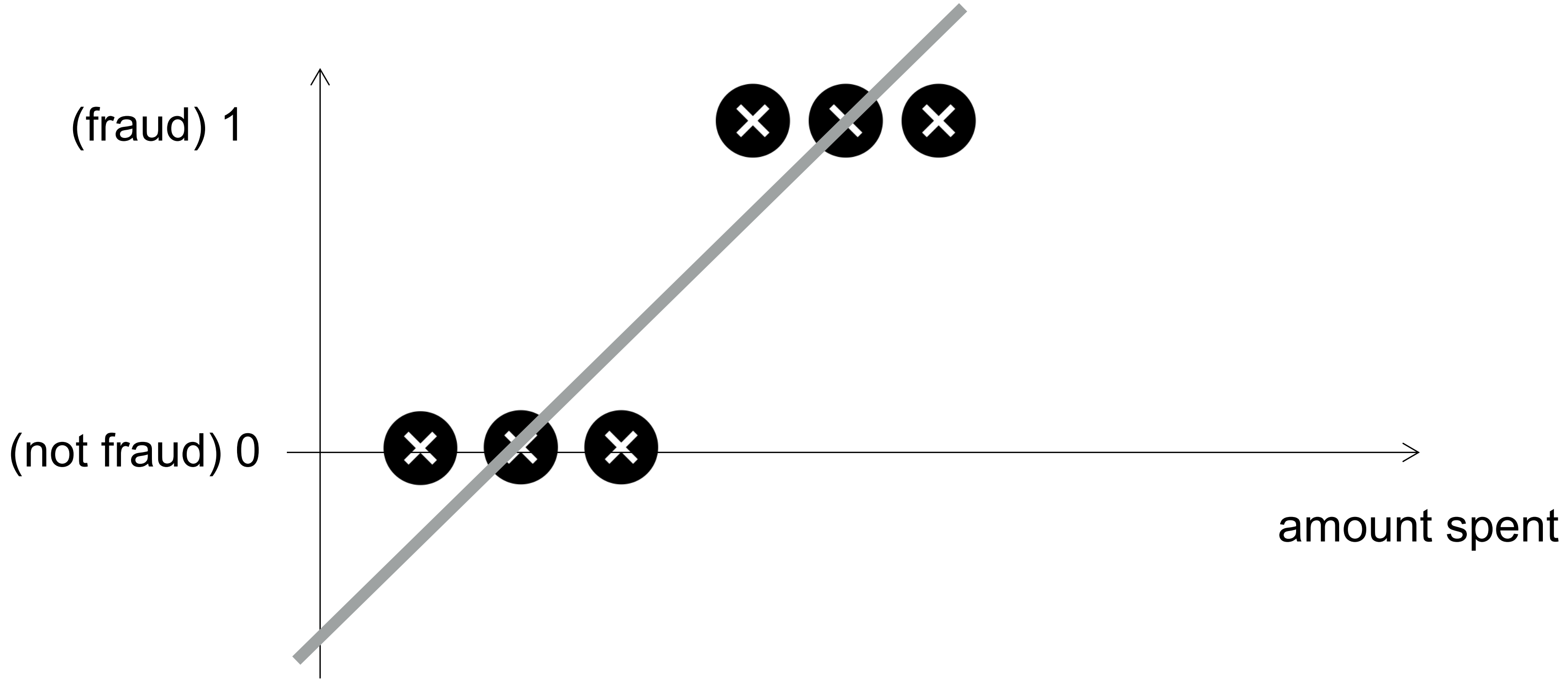
Can we do the same for **classification**?

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

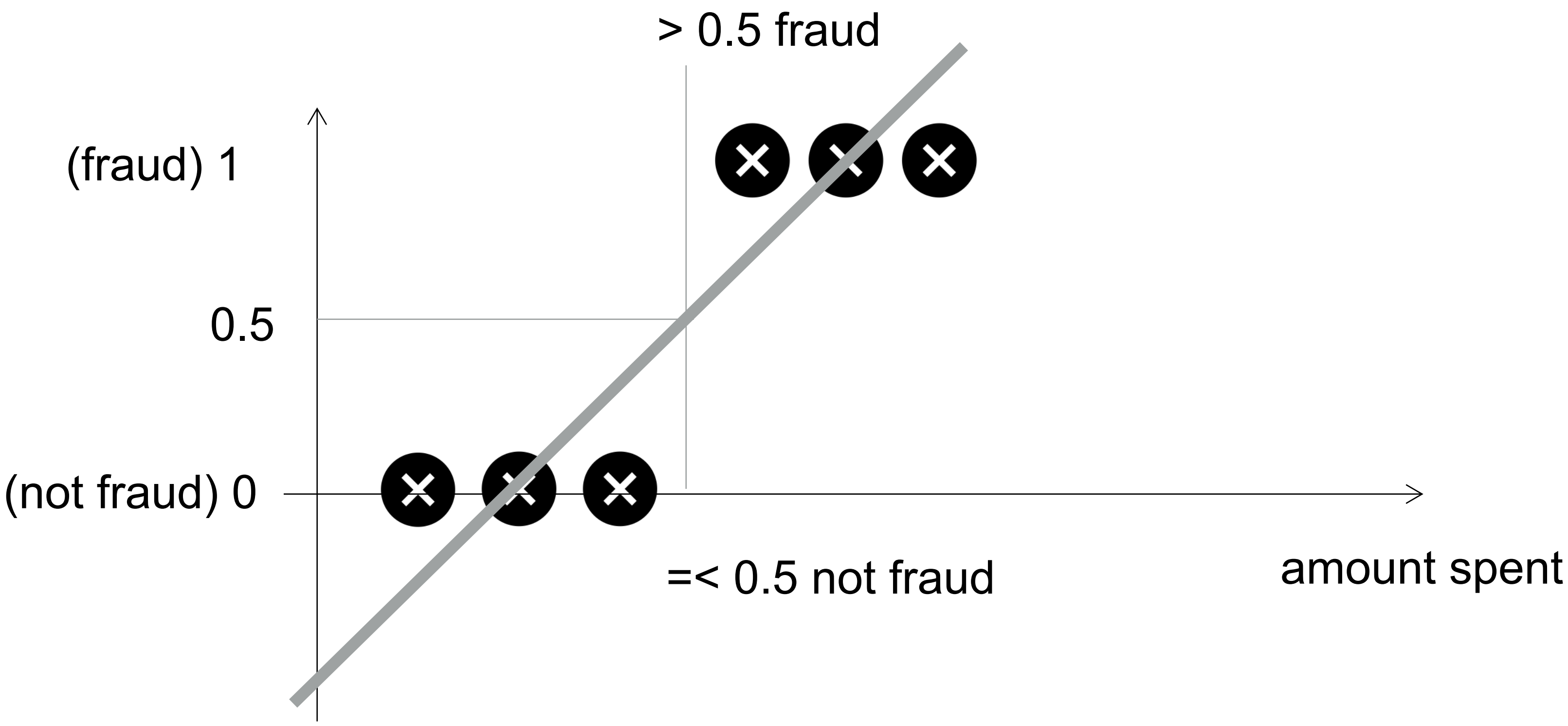
# Method (or Model)



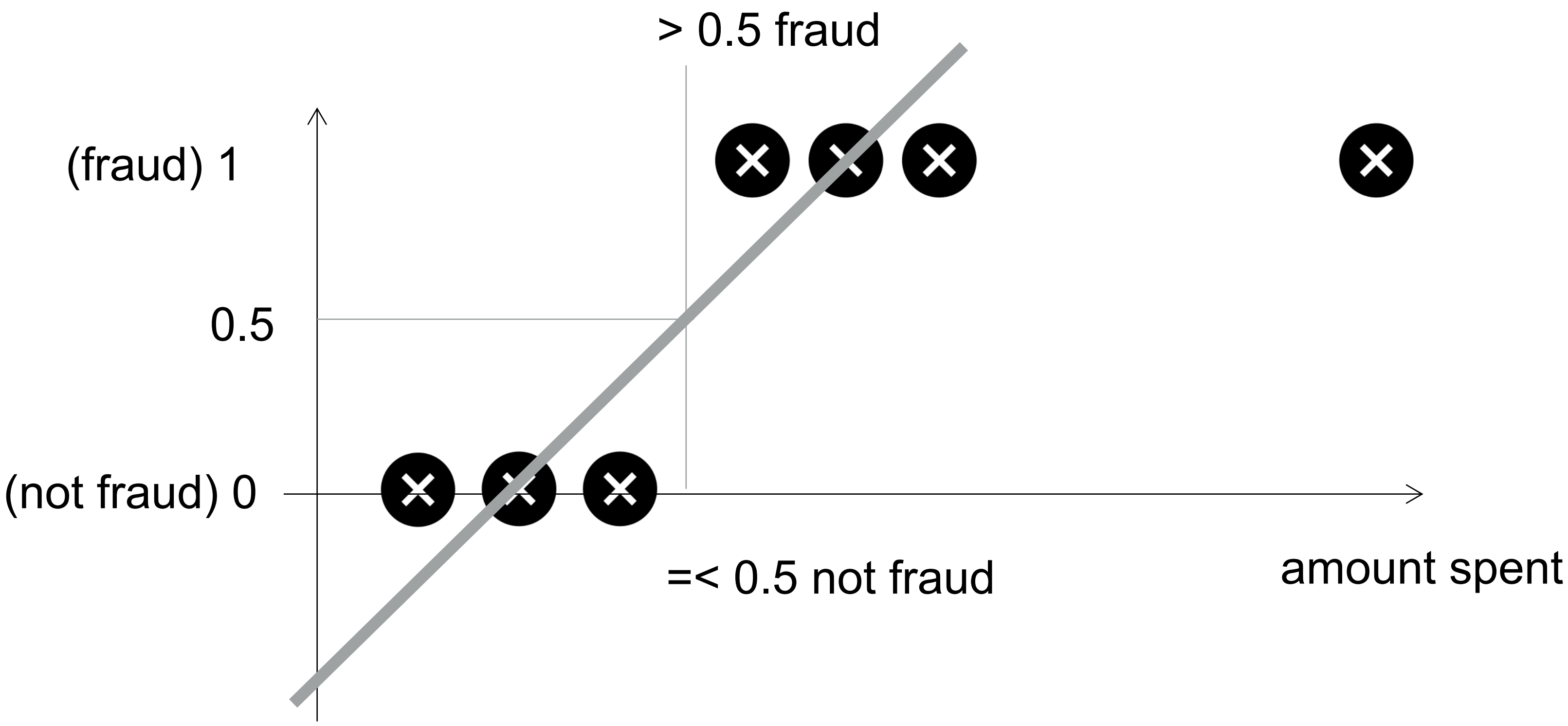
# Method (or Model)



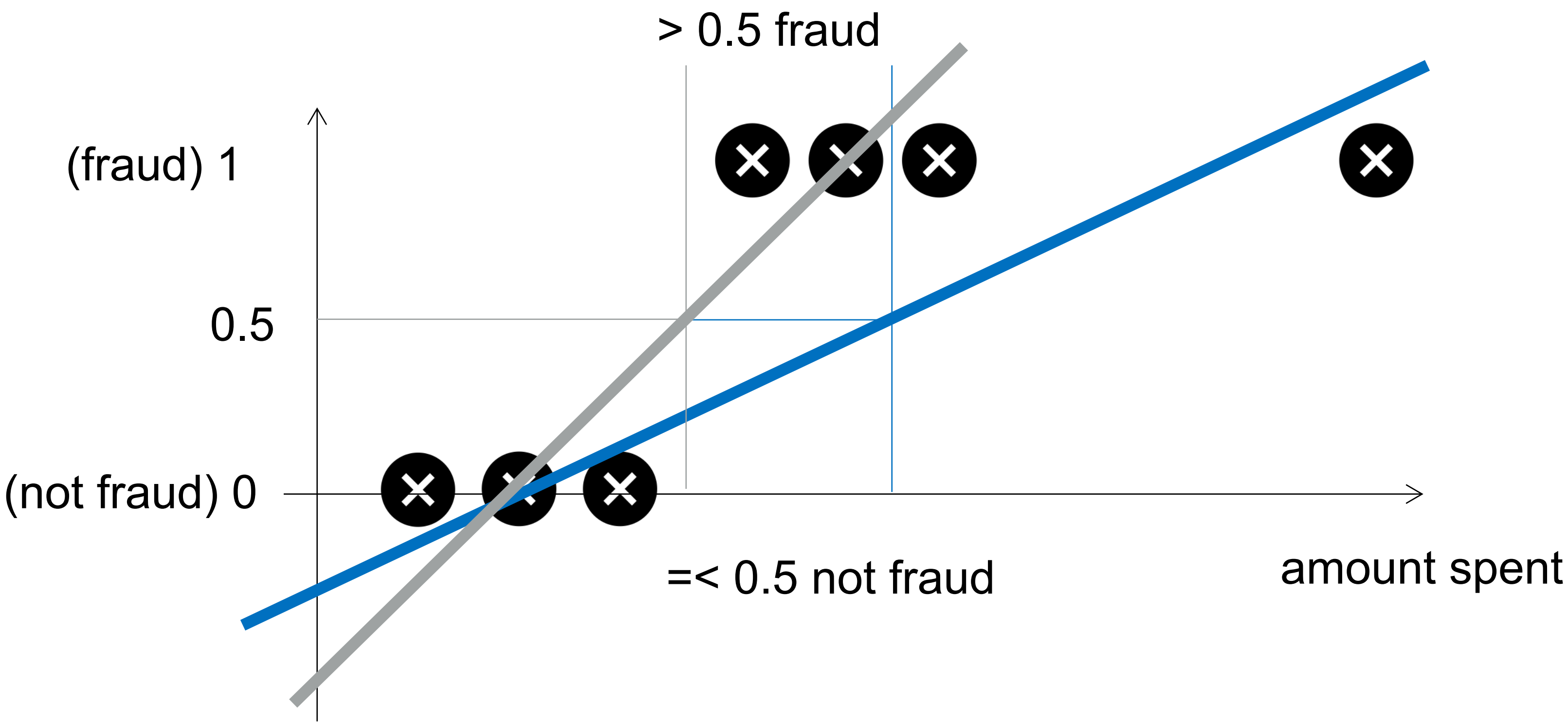
# Method (or Model)



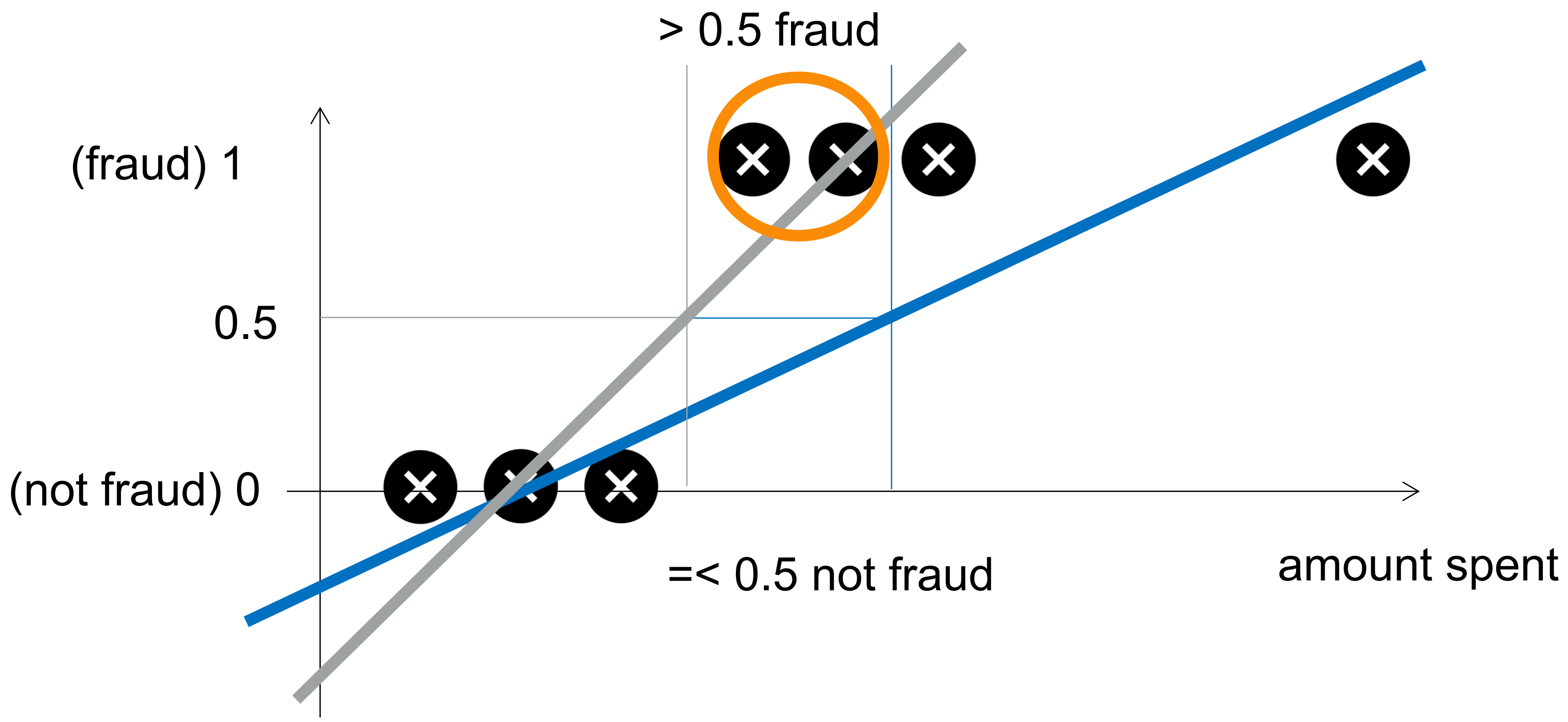
# Method (or Model)



# 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?**

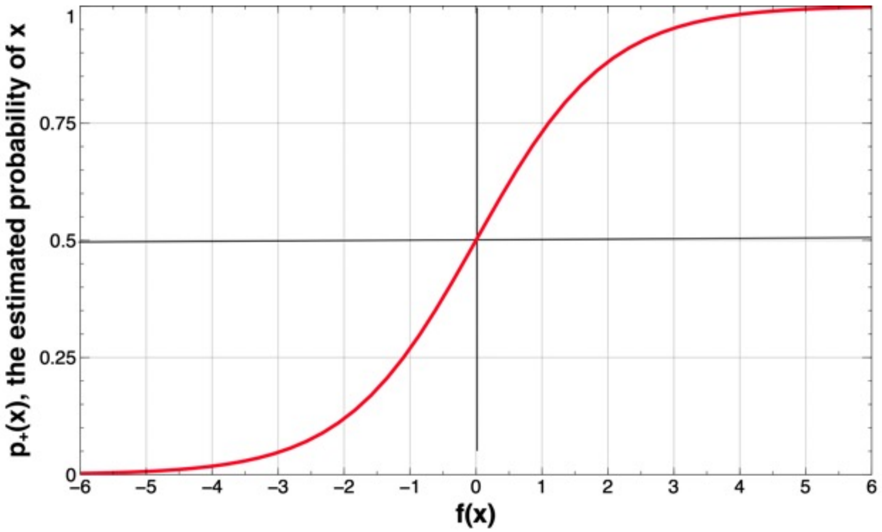
# Logistic Regression

Start with logistic regression, a very popular model 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

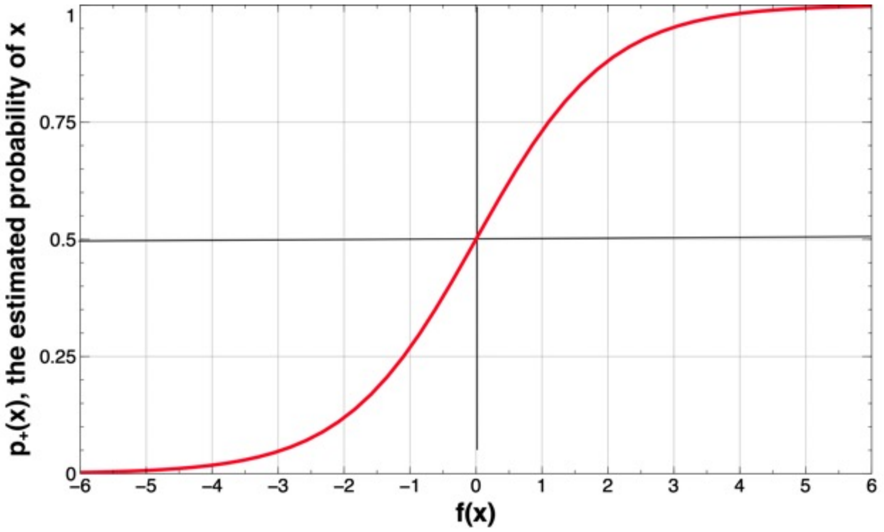
$$\log\left(\frac{p}{1-p}\right) = \boldsymbol{\alpha} + \boldsymbol{\beta x}$$

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

$p$ = probability of class membership
$\alpha$ = log odds of positive class when all predictors are zero
$\beta$ = 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 $\beta$ value for each predictor variable indicates the effect of that predictor on the odds ratio. For example, if the $\beta$ for the flue shot is negative, then getting a flu shot decreases the 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

# 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")
```

# 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 | Contract | PaperlessBilling | PaymentMethod | MonthlyCharges | TotalCharges | Churn |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 7590-VHVEG | Female | 0 | Yes | No | 1 | No | No phone service | DSL | No | Yes | No | No | No | No | Month-to-month | Yes | Electronic check | 29.85 | 29.85 | No |
| 2 | 5575-GNVDE | Male | 0 | No | No | 34 | Yes | No | DSL | Yes | No | Yes | No | No | No | One year | No | Mailed check | 56.95 | 1889.50 | No |
| 3 | 3668-QPYBK | Male | 0 | No | No | 2 | Yes | No | DSL | Yes | Yes | No | No | No | No | Month-to-month | Yes | Mailed check | 53.85 | 108.15 | Yes |
| 4 | 7795-CFOCW | Male | 0 | No | No | 45 | No | No phone service | DSL | Yes | No | Yes | Yes | No | No | One year | No | Bank transfer (automatic) | 42.30 | 1840.75 | No |
| 5 | 9237-HQITU | Female | 0 | No | No | 2 | Yes | No | Fiber optic | No | No | No | No | No | No | Month-to-month | Yes | Electronic check | 70.70 | 151.65 | Yes |
| 6 | 9305-CDSKC | Female | 0 | No | No | 8 | Yes | Yes | Fiber optic | No | No | Yes | No | Yes | Yes | Month-to-month | Yes | Electronic check | 99.65 | 820.50 | Yes |
| 7 | 1452-KIOVK | Male | 0 | No | Yes | 22 | Yes | Yes | Fiber optic | No | Yes | No | No | Yes | No | Month-to-month | Yes | Credit card (automatic) | 89.10 | 1949.40 | No |
| 8 | 6713-OKOMC | Female | 0 | No | No | 10 | No | No phone service | DSL | Yes | No | No | No | No | No | Month-to-month | No | Mailed check | 29.75 | 301.90 | No |
| 9 | 7892-POOKP | Female | 0 | Yes | No | 28 | Yes | Yes | Fiber optic | No | No | Yes | Yes | Yes | Yes | Month-to-month | Yes | Electronic check | 104.80 | 3046.05 | Yes |
| 10 | 6388-TABGU | Male | 0 | No | Yes | 62 | Yes | No | DSL | Yes | Yes | No | No | No | No | One year | No | Bank transfer (automatic) | 56.15 | 3487.95 | No |
| 11 | 9763-GRSKD | Male | 0 | Yes | Yes | 13 | Yes | No | DSL | Yes | No | No | No | No | No | Month-to-month | Yes | Mailed check | 49.95 | 587.45 | 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 | Two year | No | Credit card (automatic) | 18.95 | 326.80 | No |
| 13 | 8091-TTVAX | Male | 0 | Yes | No | 58 | Yes | Yes | Fiber optic | No | No | Yes | No | Yes | Yes | One year | No | Credit card (automatic) | 100.35 | 5681.10 | No |
| 14 | 0280-XJGEX | Male | 0 | No | No | 49 | Yes | Yes | Fiber optic | No | Yes | Yes | No | Yes | Yes | Month-to-month | Yes | Bank transfer (automatic) | 103.70 | 5036.30 | Yes |
| 15 | 5129-JLPIS | Male | 0 | No | No | 25 | Yes | No | Fiber optic | Yes | No | Yes | Yes | Yes | Yes | Month-to-month | Yes | Electronic check | 105.50 | 2686.05 | No |
| 16 | 3655-SNQYZ | Female | 0 | Yes | Yes | 69 | Yes | Yes | Fiber optic | Yes | Yes | Yes | Yes | Yes | Yes | Two year | No | Credit card (automatic) | 113.25 | 7895.15 | No |
| 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 | One year | No | Mailed check | 20.65 | 1022.95 | No |
| 18 | 9959-WOFKT | Male | 0 | No | Yes | 71 | Yes | Yes | Fiber optic | Yes | No | Yes | No | Yes | Yes | Two year | No | Bank transfer (automatic) | 106.70 | 7382.25 | No |
| 19 | 4190-MFLUW | Female | 0 | Yes | Yes | 10 | Yes | No | DSL | No | No | Yes | Yes | No | No | Month-to-month | No | Credit card (automatic) | 55.20 | 528.35 | Yes |
| 20 | 4183-MYFRB | Female | 0 | No | No | 21 | Yes | No | Fiber optic | No | Yes | Yes | No | No | Yes | Month-to-month | Yes | Electronic check | 90.05 | 1862.90 | No |
| 21 | 8779-QRDMV | Male | 1 | No | No | 1 | No | No phone service | DSL | No | No | Yes | No | No | Yes | Month-to-month | Yes | Electronic check | 39.65 | 39.65 | 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 | One year | No | Bank transfer (automatic) | 19.80 | 202.25 | No |
| 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 | Month-to-month | No | Mailed check | 20.15 | 20.15 | Yes |
| 24 | 3638-WEABW | Female | 0 | Yes | No | 58 | Yes | Yes | DSL | No | Yes | No | Yes | No | No | Two year | Yes | Credit card (automatic) | 59.90 | 3505.10 | No |
| 25 | 6322-HRPFA | Male | 0 | Yes | Yes | 49 | Yes | No | DSL | Yes | Yes | No | Yes | No | No | Month-to-month | No | Credit card (automatic) | 59.60 | 2970.30 | No |
| 26 | 6865-JZNKO | Female | 0 | No | No | 30 | Yes | No | DSL | Yes | Yes | No | No | No | No | Month-to-month | Yes | Bank transfer (automatic) | 55.30 | 1530.60 | No |
| 27 | 6467-CHFZW | Male | 0 | Yes | Yes | 47 | Yes | Yes | Fiber optic | No | Yes | No | No | Yes | Yes | Month-to-month | Yes | Electronic check | 99.35 | 4749.15 | Yes |
| 28 | 8665-UTDHZ | Male | 0 | Yes | Yes | 1 | No | No phone service | DSL | No | Yes | No | No | No | No | Month-to-month | No | Electronic check | 30.20 | 30.20 | Yes |
| 29 | 5248-YGIJN | Male | 0 | Yes | No | 72 | Yes | Yes | DSL | Yes | Yes | Yes | Yes | Yes | Yes | Two year | Yes | Credit card (automatic) | 90.25 | 6369.45 | No |
| 30 | 8773-HHUOZ | Female | 0 | No | Yes | 17 | Yes | No | DSL | No | No | No | No | Yes | Yes | Month-to-month | Yes | Mailed check | 64.70 | 1093.10 | Yes |
| 31 | 3841-NFECX | Female | 1 | Yes | No | 71 | Yes | Yes | Fiber optic | Yes | Yes | Yes | Yes | No | No | Two year | Yes | Credit card (automatic) | 96.35 | 6766.95 | No |

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

## Selecting Predictors

The goal is to find a parsimonious model – i.e., a simple model that performs well

- Correlation between predictors
- Correlation between predictors and target

# Selecting Predictors

To compute the correlation, we need numeric values

```r
# 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
    )) %>%
  mutate(InternetServiceN = case_when(
    InternetService == "Fiber optic" ~ 2,
    InternetService == "DSL" ~ 1,
  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,
```

# Selecting Predictors

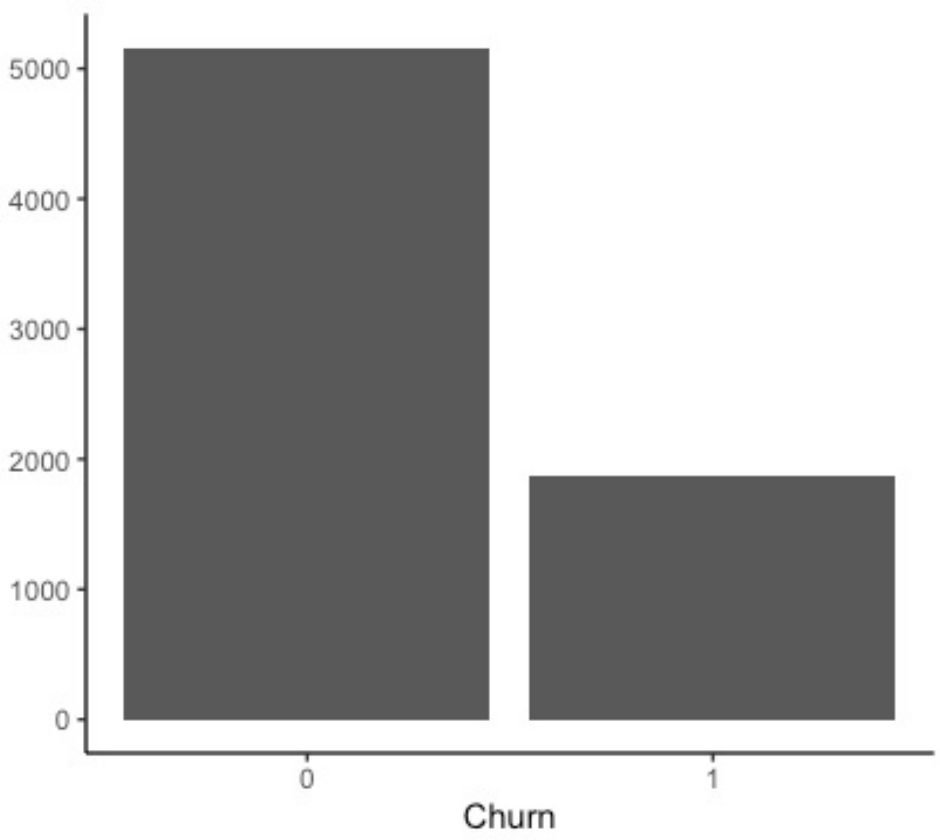
To compute the correlation, we need numeric values

```
# 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 starting 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 correlation between predictors
predCor <- cor(churn_train[,3:21])

# which variables to remove to avoid multicollinearity?
findCorrelation(predCor, cutoff =  .7, names = TRUE)
```

```
> findCorrelation(predCor, cutoff =  .7, names = TRUE)
[1] "TotalCharges"   "MonthlyCharges"
```

# Selecting Predictors

To compute the correlation, we need numeric values

```
churn_train <- churn_train %>%
  dplyr::select(Churn, ChurnN, SeniorCitizen, tenure,
genderN:PaymentN)

# compute correlation between predictors and the target
predTargetCor <- cor(churn_train[,2:19])
```

| | ChurnN |
|---|---|
| ContractN | -0.403106687 |
| tenure | -0.357595735 |
| PaymentN | -0.208914546 |
| OnlineSecurityN | -0.171331455 |
| DependentsN | -0.165798719 |
| TechSupportN | -0.165196333 |
| PartnerN | -0.151886551 |
| OnlineBackupN | -0.086484892 |
| DeviceProtectionN | -0.059665761 |
| genderN | -0.007478973 |
| PhoneServiceN | 0.023936107 |
| MultipleLinesN | 0.042687277 |
| StreamingMoviesN | 0.059091113 |
| StreamingTVN | 0.064058803 |
| SeniorCitizen | 0.151257781 |
| PaperlessN | 0.187102484 |
| InternetServiceN | 0.319796385 |
| ChurnN | 1.000000000 |

# Model Induction and Testing

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

```
model <- train(Churn ~ InternetServiceN + PaperlessN + SeniorCitizen +
                 PartnerN + TechSupportN + DependentsN + OnlineSecurityN +
                 PaymentN + tenure + ContractN,
               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")
```

- Of all customers where we predicted churn, ~65% actually churned
- Of all customers that actually churned, we only correctly predicted about half (~53%)

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  925 176
       Yes 107 197

               Accuracy : 0.7986
                 95% CI : (0.7766, 0.8193)
    No Information Rate : 0.7345
    P-Value [Acc > NIR] : 1.342e-08

                  Kappa : 0.4511

 Mcnemar's Test P-Value : 5.296e-05

              Precision : 0.6480
                 Recall : 0.5282
                     F1 : 0.5820
             Prevalence : 0.2655
         Detection Rate : 0.1402
   Detection Prevalence : 0.2164
      Balanced Accuracy : 0.7122

       'Positive' Class : Yes
```

# 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

# *Thank You!*