Eltecon Data Science Course by Emarsys

Introduction to ML

Levente Otti

September 22, 2021

Goal of the lesson

- Understand what *prediction* is, and how it differs from *causal inference*
- Try basic `R` commands for fitting & evaluating simple regression and classification models

What is prediction?

What is prediction

You have an assumed relationship:

$$Y \approx f(X) + \epsilon$$

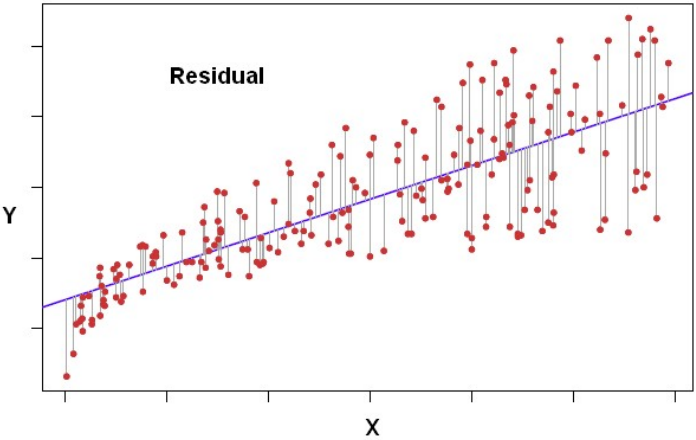
where:

- Y is your target variable
- X are your predictors
- $f()$: is the relationship between X and Y
- ϵ: is the irreducible error

Prediction is:

- estimating $f()$ based on the available observations (X) ...
- **... to minimize the error of Y vs $\hat{Y}$**

Residuals



Error metric: RMSE

- Root Mean Square Error (RMSE)
- It is the standard deviation of the residuals i.e. prediction errors
- RMSE penalizes the model for large errors

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Bike rental - the dataset

- Source: Kaggle
- Goal: Predict the total count of bikes rented during each hour

count	season_1	season_2	workingday	holiday	hour	weather_1	weather_2	temp	atemp	humidity	windspeed
16	TRUE	FALSE	0	0	0	TRUE	FALSE	9.84	14.395	81	0.0000
40	TRUE	FALSE	0	0	1	TRUE	FALSE	9.02	13.635	80	0.0000
32	TRUE	FALSE	0	0	2	TRUE	FALSE	9.02	13.635	80	0.0000
13	TRUE	FALSE	0	0	3	TRUE	FALSE	9.84	14.395	75	0.0000
1	TRUE	FALSE	0	0	4	TRUE	FALSE	9.84	14.395	75	0.0000
1	TRUE	FALSE	0	0	5	FALSE	TRUE	9.84	12.880	75	6.0032

Bike rental - variables

field	description
season	$1 = $ spring, $2 = $ summer, $3 = $ fall, $4 = $ winter
holiday	whether the day is considered a holiday
workingday	whether the day is neither a weekend nor holiday
weather	$1 \sim$ Clear, $2 \sim$ Cloudy, $3 \sim$ Light Rain, $4 \sim$ Heavy Rain
temp	temperature in Celsius
atemp	feels like temperature in Celsius
humidity	relative humidity
windspeed	wind speed
count	number of total rentals

Benchmark "model"

```
calculateRMSE <- function(actual, predictions) {
  sqrt(mean((actual - predictions) ^ 2))
}
```

```
predictions_benchmark_model <- rep(
  bike_sharing_train[, mean(count)], bike_sharing_train[, .N]
)

bike_sharing_benchmark_model_rmse <- calculateRMSE(
  actual = bike_sharing_train$count,
  predictions = predictions_benchmark_model
)

bike_sharing_benchmark_model_rmse
```

```
## [1] 170.2384
```

Regression

Linear regression

When to use it:

- We want to estimate a numerical target (e.g. price)
- Assuming an approximate linear relationship between X and Y

Simple linear regression formula:

$$\hat{Y} = \beta_0 + \beta_1 X$$

The OLS estimation of β will conveniently minimize RMSE for given X!

(So what is Machine Learning?)

"A computer program is said to ***learn from experience*** *E with respect to some class of tasks T and performance measure P,* ***if its performance*** *at tasks in T, as measured by P,* ***improves with experience*** *E."* (Tom Mitchell) Performance is the error metric, ie. RMSE; Experience is the training data we provide. So it means, with more data the ML will provide better predictions. ## Bike rental - minimal model

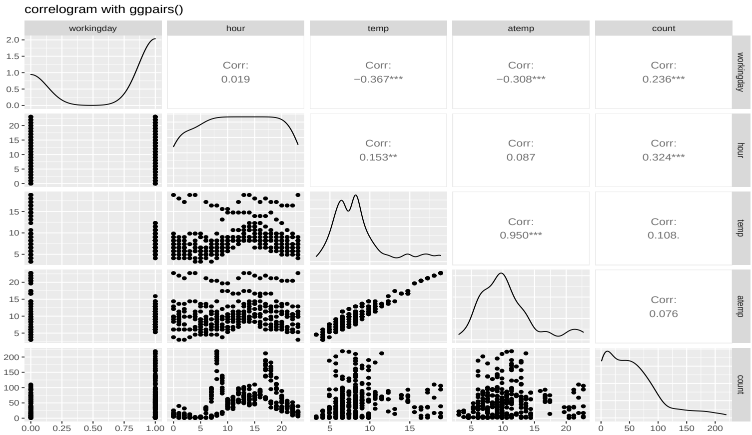
```
bike_sharing_min_model <- glm(
  formula = count ~ atemp, #target variable ~ feature variables
  family = gaussian,
  data = bike_sharing_train
)
```

Discovering basic patterns in the data

```
library(GGally)
data <- as.data.frame(bike_sharing_train[sample(300), .(workingday, hour, temp
```

Discovering basic patterns in the data

```
ggpairs(data, title="correlogram with ggpairs()")
```



Bike rental - minimal model

```r
summary(bike_sharing_min_model)
```

```
##
## Call:
## glm(formula = count ~ atemp, family = gaussian, data = bike_sharing_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -282.73  -103.86   -27.63    72.96   709.89
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -19.5412     4.6755   -4.18 2.95e-05 ***
## atemp         8.3048     0.1841   45.10  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 23649.64)
##
##     Null deviance: 261264747  on 9014  degrees of freedom
## Residual deviance: 213154213  on 9013  degrees of freedom
## AIC: 116378
##
## Number of Fisher Scoring iterations: 2
```

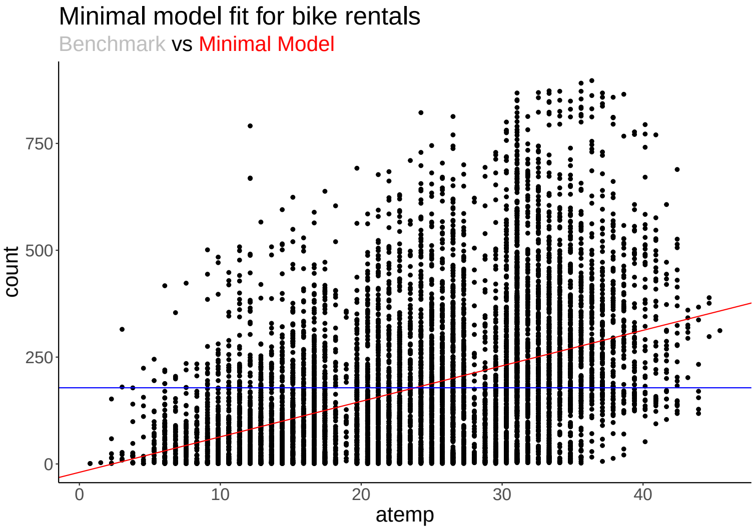
Bike rental - minimal model - coefficients

```r
str(bike_sharing_min_model$coefficients)
```

```
##  Named num [1:2] -19.5 8.3
##  - attr(*, "names")= chr [1:2] "(Intercept)" "atemp"
```

```r
intercept <- bike_sharing_min_model$coefficients[1]
slope <- bike_sharing_min_model$coefficients[2]
```

Bike rental - minimal model - predictive fit



Bike rental - minimal model - prediction error

```
predictions_min_model <- predict.glm(
  bike_sharing_min_model, newdata = bike_sharing_train
)

predictions_min_model[1:5]
```

```
##         1         2         3         4         5
## 100.00620  93.69456  93.69456 100.00620 100.00620
```

```
bike_sharing_train[1:5, count]
```

```
## [1] 16 40 32 13  1
```

Bike rental - minimal model - prediction error

```
calculateRMSE <- function(actual, predictions) {
  sqrt(mean((actual - predictions) ^ 2))
}
```

```
bike_sharing_min_model_rmse <- calculateRMSE(
  actual = bike_sharing_train[, count],
  predictions = predictions_min_model
)

bike_sharing_min_model_rmse
```

```
## [1] 153.7673
```

Bike rental - improving predictions

```r
bike_sharing_2nd_model <- glm(
  formula = count ~ atemp + humidity,
  data = bike_sharing_train
)

predictions_2nd_model <- predict.glm(
  bike_sharing_2nd_model, newdata = bike_sharing_train
)

calculateRMSE(
  actual = bike_sharing_train[, count],
  predictions = predictions_2nd_model
)
```

```
## [1] 144.5333
```

Practice time

- Task: improve the model to be as accurate as possible!
- Share your regression formula + achieved RMSE in Socrative!
- You have 20 minutes - feel free to take a break if needed.

Practice time

DEMO

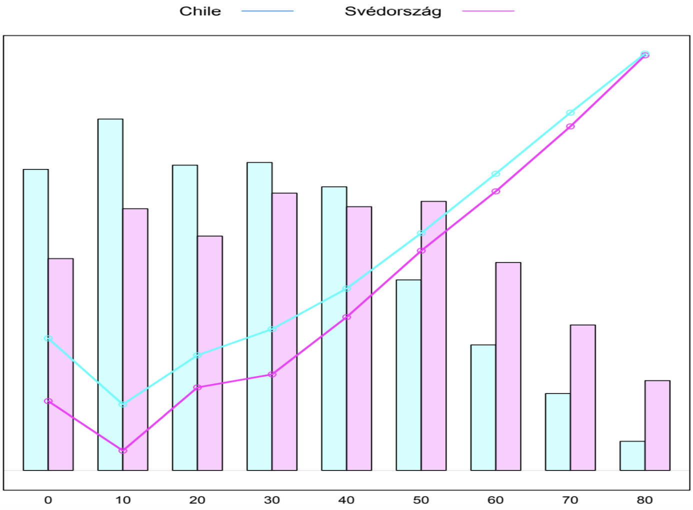
Prediction vs causality (Confounding)

In Sweden, in 2005 91709 people were dead. The population was 9 010 729, so the mortality rate is 10,2/1000 person / year.

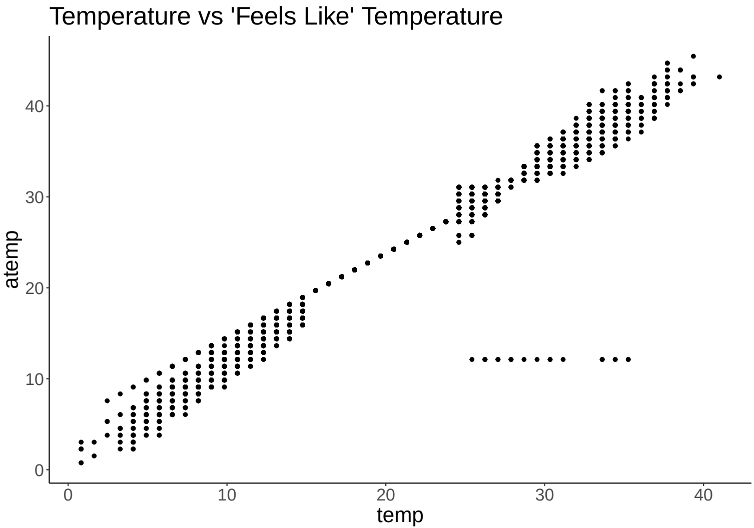
In the same year in Chile there was 86100 deaths, while the population was 15 519 347, so the mortality rate was 5,5/1000 people/year.

Conclusion: in Sweden the mortality rate is double compared to Chile???

Prediction vs causality - Mortality by Age



Don't worry about Confounding!



Why should you still care about model inputs?

- Model explainability is often desirable in business (and other applications)
- Example: Amazon Sexist Hiring AI
- In Emarsys: all models are retrained every 30 days
- This ensures that any shifts in user behavior are promptly captured
- E.g. before / after Black Friday

Classification

Binary classification

- Binary: target can take on two values (0 or 1)
- Typical example is predicting if an event is happening or not
- Examples:
- Patient has a medical condition or not
- Loan will be repaid in full
- User will make a purchase in the next 30 days (BPS - Buying Probability Score)

Question time

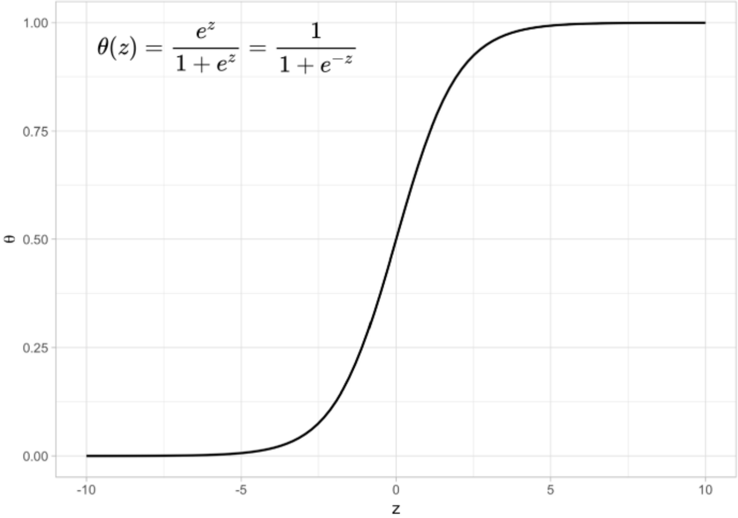
- **Why should we not use linear regression for binary classification**?
- Please record your answers in Socrative!
- You have 5 minutes

Logistic regression

- A linear model makes continuous predictions that are unbounded.
- In classification, we are interested in the probability of an outcome occurring
- So we want **predictions that are bounded between 0 and 1.**

$$Pr(Y_i = 1|X_i) = \frac{exp(\beta_0 + \beta_1 X_i)}{1 + exp(\beta_0 + \beta_1 X_i)}$$

The sigmoid function



The Titanic dataset

```
install.packages("titanic")
library(titanic)
head(titanic_train)
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500		S
6	0	3	Moran, Mr. James	male	NA	0	0	330877	8.4583		Q

Example - binary model

```
model <- glm(
    Survived ~ Fare,
    data = titanic_train,
    family = binomial(link = "logit")
)
```

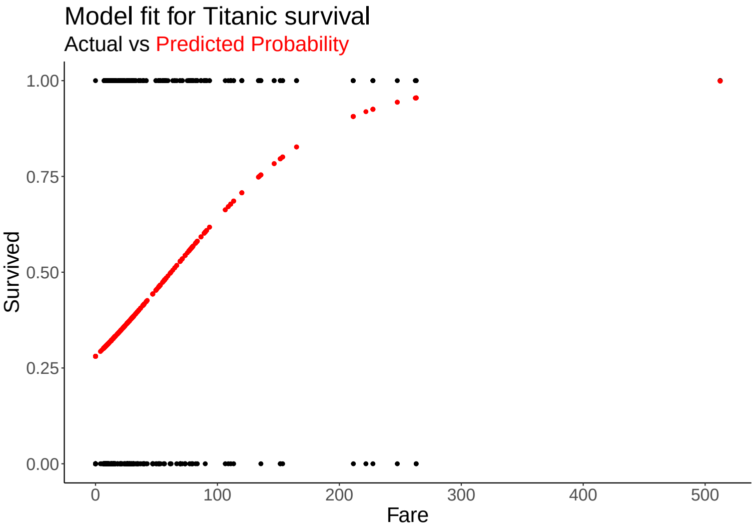
Making predictions

```
predicted_prob <- predict.glm(
  model,
  newdata = titanic_train,
  type = "response"
)

predicted_prob[1:5]
```

```
##         1         2         3         4         5
## 0.3034014 0.5354287 0.3055738 0.4664564 0.3059770
```

Predictive fit



Converting probabilities to predictions

```
titanic_train[1:5, "Survived"]
```

```
## [1] 0 1 1 1 0
```

Using `cutoff = 0.5`:

```
predicted_class <- ifelse(predicted_prob > 0.5, 1, 0)
predicted_class[1:5]
```

```
## 1 2 3 4 5
## 0 1 0 0 0
```

Evaluating binary models - Accuracy

```r
calculateAccuracy <- function(actual, predicted) {
    return( sum(actual == predicted) / length(actual) )
}
```

```r
calculateAccuracy(titanic_train$Survived, predicted_class)
```

```
## [1] 0.6655443
```

Evaluating binary models - Confusion Matrix

```
table(
  predicted_class,
  titanic_train$Survived,
  dnn = c( "predicted", "actual")
)
```

```
##          actual
## predicted   0   1
##         0 511 260
##         1  38  82
```

Confusion Matrix - Precision & Recall

Predicted Label \ Real Label	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

$$\text{Precision} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FN}}$$

$$\text{Accuracy} = \frac{\sum \text{TP} + \text{TN}}{\sum \text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

Practice time

- Task: improve the model to be as accurate as possible!
- Share your regression formula + achieved Accuracy & Confusion Matrix in Socrative!
- You have 20 minutes - feel free to take a break if needed.

Practice time

DEMO

Generalization performance

Why do we care?

- It's easy to predict something we already know. . .
- Actually, it would be silly to build predictive models to predict what we already know!
- What we are after is **out-of-sample** performance

Example on the Bike Sharing dataset

```
simple_model <- glm(
  count ~ hour + temp + workingday,
  data = bike_sharing_train
)
```

Accuracy on our training data

```r
simple_model_predictions <- predict.glm(
  simple_model,
  newdata = bike_sharing_train
)

calculateRMSE(bike_sharing_train$count, simple_model_predictions)
```

```
## [1] 141.4175
```

Accuracy out-of-sample

```
bike_sharing_test <- fread("bike_sharing_test.csv")

simple_model_predictions <- predict.glm(
  simple_model,
  newdata = bike_sharing_test
)

calculateRMSE(bike_sharing_test$count, simple_model_predictions)
```

```
## [1] 208.0325
```

Homework

Homework - prediction

- Find an interesting example for Prediction vs causality issue
- Find the relevant data and visualise it