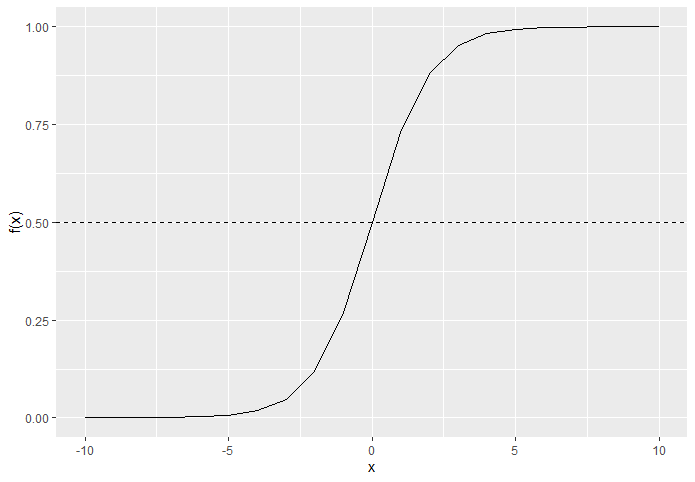
Ako su vam pojmovi logistička regresija, maximum likelihood, log odds, generalized linear models uvijek bili neshvatljivi i nije vam vilo prijatno kada naiđete na njih, onda je ovaj post za vas. Inuitivno, kroz primjer, cilj je da jasno objasnim ove pojmove, tako da sve sami možete replicirati, korak po korak. Određeni koncepti možda možda nisu rigorozno tačno prikazani, zbog pedagoške simplifikacije, ali to ne umanjuje uvide i zaključke ovoga objašnjenja.

Priču započunjemo sredinom 19. vijeka kada belgijski naučnik Pjer Fransoa Verhulst, baveći se istraživanjem rasta ljudske populacije, dolazi do logističke funkcije, kako je sam nazvao. Funkcija, kao i svaka druga matematička funkcija uzima brojeve, nešto radi sa njima i izbaca neki rezultat. Ono što logistička funkcija radi sa brojevima prikazano je sledećom formulom:

f(x) =



Precizno, ovo je standardna logistička funkcija, ali je i dalje logistička funkcija. Uzme 1 i 'izbaci' 0.7310, uzme -1 i 'izbaci' 0.2689. Koje god vrijednosti da uzme od – besonačno, do + beskonačno 'izbaciće' rezultat od 0 do 1. Tako za vrijednost x-a od -10 do 10 logistička funkcija izgleda ovako.

 Izgled logističke funkcije.

Logistička funkcija pripada sigmoidnim funkcijama, a to su funkcije koje imaju izgled latiničnog slova S.

Sada kad znamo šta je logistička funkcija, da vidimo i šta je logistička regresija. Logistička regresija se koristi u slučaju kada želimo da modeliramo ponašanje varijable koja ima samo dva ishoda (da/ne, 0/1, etc). Idemo odmah na primjer kako bi prikazali prvo rezultate, a onda pokazali i kako 'radi' logistička funkcija.

Recimo da jedna telekomunikaciona kompanija želi da predvidi odliv svojih korisnika i sakupila je određene podatke (Podaci dostupni na linku <https://www.kaggle.com/datasets/blastchar/telco-customer-churn>). Kako želimo da pokažemo mehanizam funkcionisanja logističke regresije, uzećemo samo jednu nezavisnu varijablu – -'MonthlyCharges' i vidjeti kako ona utiče na stepen odliva klijenata. Kolona 'Churn' u dataset-u ima dva ishoda 'No' i 'Yes' gdje 'No' znači da klijent nije napustio kompaniju. Importujemo podatke u R i radimo logističku regresiju:

library(dplyr)  
library(ggplot2)

read.csv("C:/Users/Nikola/Downloads/telco.csv") -> data1

data1 %>% mutate(Churn = ifelse(Churn == "No", 0,1)) -> data

glm(Churn ~ MonthlyCharges, family = binomial, data = data) -> model\_telco

ggplot(data, aes(x = MonthlyCharges, y = fitted(model\_telco))) +

geom\_line() +

labs(title = "Logistic Regression Fitted Values",

x = "MontlyCharges",

y = "Predicted Probability") +

theme\_minimal()

coef(model\_telco)[1] -> intercept

coef(model\_telco)[2] -> b1

1/(1+exp(-(intercept+b1\*data$MonthlyCharges))) -> test

head(fitted(model\_telco))   
head(test)

Glm u formuli za logističku regresiju predstavlja generalized linear model. Pretpostavke linearne regresije su linearna zavisnost izmedju zavisne i nezavisne varijable i normalan raspored nezavisne varijable. GLM omogućava modeliranje uz narušavanje ovih pretpostavki. Specifično, kod logističke regresije, kao tipa GLM-a, response varijabla nema normalan raspored, vec binominalan (može imati samo vrijednosti 0 i 1). Vratimo se na formulu logističke regresije:

y\_pred = (1)

Svaki GLM ima tri komponente:

1. Random komponentu koja se odnosi na raspored zavisne varijable.
2. Systematic komponentu koja predstavlja linearnu kombinaciju nezavisnih varijabli
3. Link funkciju koja predstavlja vezu između zavisne varijable i linearne kombinacije nezavisnih varijabli

Kod logističke regresije, kao što smo rekli, binominalan je raspored zavisne varijable, b0+b1\*x je sistemska komponenta a link funkcija je logistička funkcija.

Na ovaj način smo pokazali da se predviđena vrijednost logističke regresije dobija pomoću logističke funkcije. Dok je veza između varijabli kod linearne funkcije prikazana pravom linijom, pa otud i naziv linearna regresija, veza kod logističke regresije je prikazana logističkom funkcijom, pa otud i naziv logistička regresija. Kod linearne regresije se koeficijenti određuju metodom najmanjih kvadrata, dok kod logističke nečim što zovemo maximum likelihood method. Da vidimo šta je to i da dokažemo.

Dokaz o izgledu logističke funkcije:

simulated\_values <- runif(500, min = 0, max = 500)

1/(1+exp(-(intercept+b1\*simulated\_values))) -> test1

as.data.frame(cbind(simulated\_values, test1)) -> df1

ggplot(df1,aes(simulated\_values, test1)) + geom\_line()

Kako smo dobili koeficijente b0 i b1?

U našem datasetu, prava vrijednost y-a koju ćemo nazvati y\_actual može imati samo 2 vrijednosti 0 ili 1 (Kolona 'Churn'). Vrijednosti predviđene logističkom regresijom uzimaju numericke vrijednosti između 0 i 1. Razmotrimo sledeći izraz:

y\_actual\*y\_pred + (1-y\_actual)\*(1-y\_pred)

Kada je vrijednost y\_actual = 1 (klijent je otišao) tada izraz ima vrijednost:

1\*y\_pred + 0\*(1-y\_pred) = y\_pred. Maksimalna vrijednost ovog izraza je 1, ako nam je i y\_pred = 1, u kom slučaju smo tačno predvidjeli.

Kada je vrijednost y\_actual = 0 (klijent nije otišao) tada izraz ima vrijednost:

0\*y\_pred + 1\*(1-y\_pred) = 1 – y\_pred. Maksimalna vrijednost ovog izraza je 1, kada je y\_pred = 0, u kom slučaju smo tačno predvidjeli.   
  
Na osnovu ovog primjera, vidimo da nam je cilj da maksimiziramo izraz s početka. Krenimo sa proizvoljnom vrijednošću naših koeficijenata b0 i b1:

b0 <- -2  
b1 <- 0.1

Izračunajmo y\_predicted za ove vrijednosti koeficijenata koristeći logističku regresiju (formula 1) i naš dataset.

y\_pred <- 1/(1+exp(b0+b1\*data$MonthlyCharges))

as.data.frame(cbind(y\_actual = data$Churn, y\_pred)) -> data2

data2 %>% mutate(izraz = y\_actual\*y\_pred + (1-y\_actual)\*(1-y\_pred)) %>% summarise(sum(izraz))

Ovaj izraz se zove likelihood. Zbog komputaciono praktičnih razloga, izraz koji maksimiziramo je log likelihood.

log(y\_actual)\*y\_pred + log(1-y\_actual)\*(1-y\_pred) – log likelihood

sum(y\_actual\*y\_pred + (1-y\_actual)\*(1-y\_pred)) – želimo da maksimiziramo ovaj izraz.

y\_actual imamo (ono je ili 0 ili 1), a y\_pred je logistička regresija čija vrijednost zavisi od dva parametra, intercept i slope.

calc\_neg\_log\_likelihood <- function(coeffs) {

*# Get the intercept coeff*

intercept <- coeffs[1]

*# Get the slope coeff*

slope <- coeffs[2]

*# Calculate the predicted y values*

y\_pred <- 1/(1+exp(-(intercept+slope\*data$MonthlyCharges)))

*# Calculate the log-likelihood for each term*

log\_likelihoods <- log(y\_pred)\*data$Churn + log(1-y\_pred)\*(1-data$Churn)

*# Calculate minus the sum of the log-likelihoods for each term*

-sum(log\_likelihoods)

}

optim(

*# Initially guess -1 intercept and 0.1 slope*

par = c(intercept = -1,slope = 0.1),

*# Use calc\_neg\_log\_likelihood as the optimization fn*

fn = calc\_neg\_log\_likelihood

)

Pretpostavka normalne distribucije i statističko testiranje koeficijenata linearne i logističke regresije.