Inferență statistică în ML

Cap 10. Alegerea modelului regresiei. Generalized linear model.

May 28, 2019

Regresia multivariabilă

- modelul regresiei liniare de mai multe variabile urmărește crearea unui model care să poată fi interpretabil
- modelul trebuie să fie cât mai simplu ca să explice datele observate, dar nu mai simplu[1]
- exemplu: dacă o variablilă explică într-o mică măsură o parte din variabilitate, dar impietează puternic asupra interpretabilității, acea variabilă va fi omisă din model
- modelele nu sunt nici bune nici proaste, ci ajută la explicarea datelor
- ne concentrăm asupra a ce variabile includem/excludem din model

[1] https://en.wikipedia.org/wiki/Occam%27s_razor

Reguli

- neincluderea unor variabile importante în model poate duce la valori polarizate (biased) a coeficienților regresorilor (în cazul în care există corelație între regresorii incluși și cei neincluși)
- includerea unor variabile care nu ar trebui incluse crește eroarea standard a variabilelor regresiei
- de fapt includerea oricăror variabile duce la creșterea erorii standard al altor regresori
- nu dorim să includem variabile în model fără discernământ
- în practică se fac teste randomizate: dacă avem două grupuri, grupul tratat vs. grupul de control, vom calcula diferența între medii pentru fiecare grup; apoi facem reetichetarea grupurilor, și recalculăm diferența între medii[2]
- în urma testelor randomizate, putem vedea dacă diferența dintre medii are sau nu o valoare extremă (calculăm p-value)

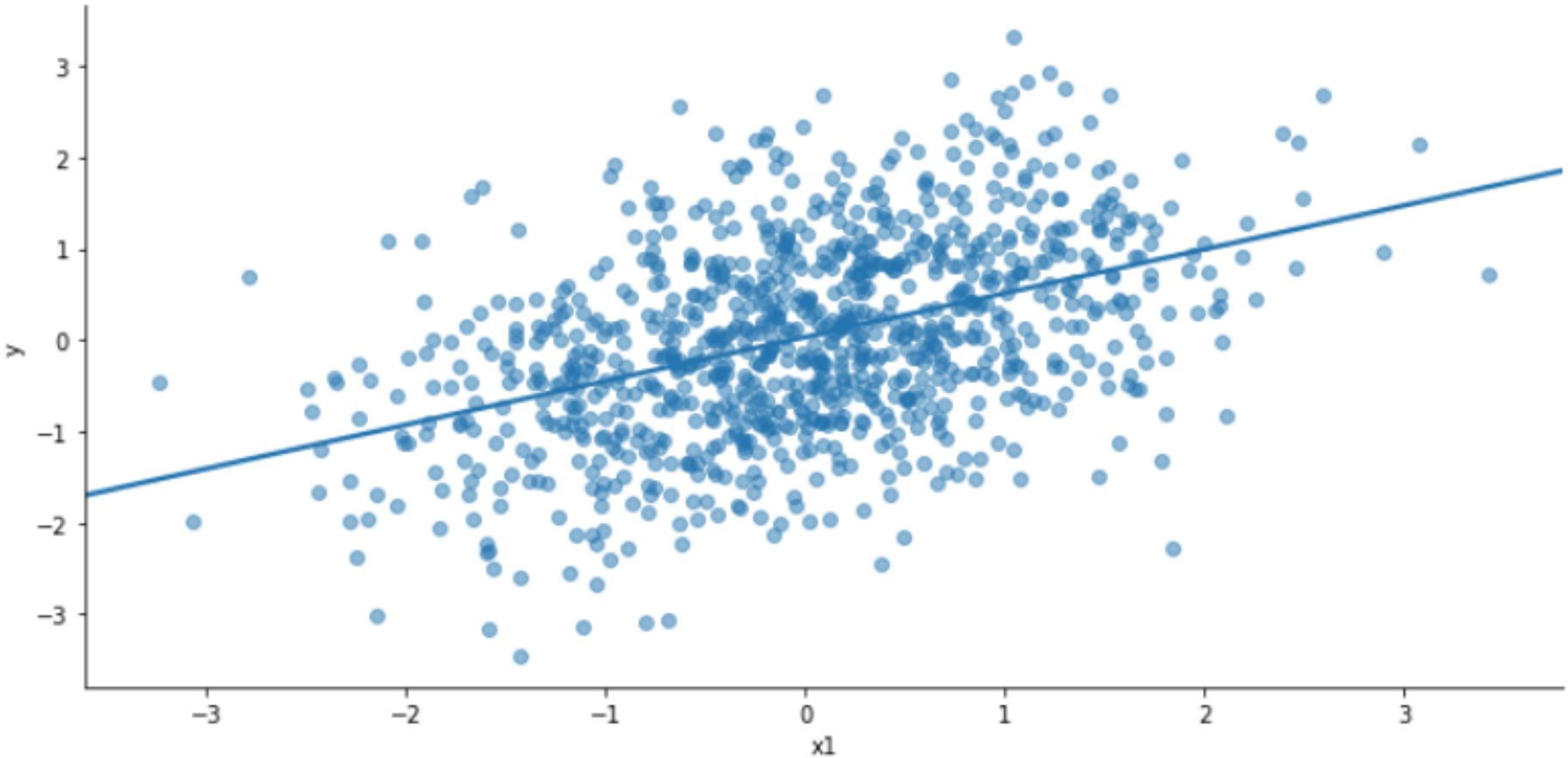
[2] https://www.uvm.edu/~dhowell/StatPages/Randomization%20Tests/RandomizationTestsOverview.html

Reguli (2)

- randomizarea poate fi aplicată în alegerea variabilelor, dar dacă sunt prea multe variabile al căror efect se confundă, atunci nu ajută (ex. includerea ambelor tensiuni arteriale sistolică și diastolică ca predictori)
- includerea tuturor variabilelor neimportante elimină polarizarea (bias)
- însă includerea tuturor variabilelor duce la creșterea erorilor standard reale (nu estimate)
- R^2 crește odată cu introducerea unui număr mai mare de regresori
- SSE, suma pătratelor erorilor (reziduurilor) scade pe măsură ce adăugăm regresori

R-square crește (1)

```
n = 1000
x1, x2, x3 = np.random.randn(n), np.random.randn(n), np.random.randn(n)
y = x1/2 + np.random.randn(n)
df = pd.DataFrame({'x1': x1, 'y': y})
sns.lmplot(x='x1', y='y', data=df, aspect=2, ci=None,  # ci='95'
        scatter_kws={'lw': 1, 'alpha': 0.5}) ; plt.show()
```



R-square crește (2)

```
model = smf.ols(formula='y ~ x1', data=df).fit()
print(model.rsquared, np.sum(model.resid ** 2))
```

```
0.18839681815757192 951.6344495838188
```

```
model = smf.ols(formula='y ~ x1 + x2', data=df).fit()
print(model.rsquared, np.sum(model.resid ** 2))
```

```
0.18940568602457175 950.4515150675498
```

```
model = smf.ols(formula='y ~ x1 + x2 + x3', data=df).fit()
print(model.rsquared, np.sum(model.resid ** 2))
```

```
0.19354712410544028 945.5955272686529
```

Variance Inflation: predictori necorelați

```
n, nosim = 100, 1000
x1, x2, x3 = np.random.randn(n), np.random.randn(n), np.random.randn(n)
betas = np.zeros((nosim, 3))
for i in range(nosim):
    y = x1 + np.random.randn(n)*.3
    df = pd.DataFrame({'x1': x1, 'x2': x2, 'x3': x3, 'y': y})
    model = smf.ols(formula='y ~ x1', data=df).fit()
    betas[i, 0] = model.params[1]
    model = smf.ols(formula='y ~ x1 + x2', data=df).fit()
    betas[i, 1] = model.params[1]
    model = smf.ols(formula='y ~ x1 + x2 + x3', data=df).fit()
    betas[i, 2] = model.params[1]

np.std(betas, axis=0)
```

```
array([0.03074886, 0.03092766, 0.0311712 ])
```

Variance Inflation: predictori corelați

```
n, nosim = 100, 1000
x1 = np.random.randn(n)
x2 = x1/2 + np.random.randn(n)/2
x3 = x1 * 0.95 + np.random.randn(n) * (1 - 0.95)
betas = np.zeros((nosim, 3))
for i in range(nosim):
    y = x1 + np.random.randn(n)*.3
    df = pd.DataFrame({'x1': x1, 'x2': x2, 'x3': x3, 'y': y})
    model = smf.ols(formula='y ~ x1', data=df).fit()
    betas[i, 0] = model.params[1]
    model = smf.ols(formula='y ~ x1 + x2', data=df).fit()
    betas[i, 1] = model.params[1]
    model = smf.ols(formula='y ~ x1 + x2 + x3', data=df).fit()
    betas[i, 2] = model.params[1]

np.std(betas, axis=0)
```

```
array([0.02912786, 0.0433571 , 0.59134793])
```

Variance Inflation

- inflația dispersiei este mult mai accentuată când includem o variabilă puternic corelată cu predictorul existent
- nu cunoaștem deviația standard a coeficientului β, așa încât putem doar estima deviația standard a unui regresor
- dacă adăugăm variabile predictor, putem verifica dispersia pentru fiecare includere
- dacă regresorii adăugați sunt ortogonali celor existenți, nu avem variance inflation
- factorul de inflație VIF este creșterea dispersiei pentru includerea acelui regresor comparativ cu situația ideală în care acesta este ortogonal cu restul regresorilor

Variance Inflation (2)

$$VIF_i = \frac{1}{1 - R_i^2}$$

- pentru un regresor X_i, VIF_i este măsura în care predictorul X_i este sau nu corelat cu ceilalți predictori
- R_i^2 este R-square calculat pentru regresia $X_i \sim X_1, X_2, \ldots X_{i-1}, X_{i+1}, \ldots X_n$
- dacă X_i nu este corelat cu ceilalți predictori, atunci R^2 va fi foarte mic, iar VIF_i tinde spre 1
- cu cât R^2 se apropie de 1, VIF_i tinde la infinit (observați că pentru $R^2 = 0.9$, $VIF_i = 10$)
- VIF exprimă raportul dintre dispersia în situația curentă vs. situația în care predictorul i ar fi necorelat cu ceilalți predictori

Variance Inflation: Swiss dataset

```
df = pd.read_csv('swiss.csv')
df = df.iloc[:, 2:]
df.head()
```

	Agriculture	Examination	Education	Catholic	Infant.Mortality
0	17.0	15	12	9.96	22.2
1	45.1	6	9	84.84	22.2
2	39.7	5	5	93.40	20.2
3	36.5	12	7	33.77	20.3
4	43.5	17	15	5.16	20.6

Variance Inflation (3)

```
vif = pd.DataFrame()
vif['VIF factor'] = [
    variance_inflation_factor(df.values, i) \
    for i in range(len(df.columns)) ]
vif['predictor'] = df.columns ; vif
```

	VIF factor	predictor
0	8.127512	Agriculture
1	15.858235	Examination
2	6.337873	Education
3	3.850196	Catholic
4	19.570671	Infant.Mortality

Variance Inflation (4)

	VIF factor	predictor
0	5.319546	Agriculture
1	9.174906	Examination
2	6.271528	Education
3	3.176365	Catholic

- euristic, un factor VIF_i mai mare ca 10 indică o corelație puternică între predictorul i și alt (alți) predictori ai modelului

Estimarea variabilității reziduale

- dacă omitem variabile predictor, dispersia estimată este biased (pentru că nu includem anumite contribuții pe care acele variabile le aduc modelului)
- dacă includem doar variabilele necesare sau toate (overfit), dispersia estimată este unbiased
- totuși, dacă includem mai mulți predictori decât este necesar, dispersia dispesiei estimate este 'inflated'
- o soluție pentru alegerea predictorilor poate fi PCA (analiza componentelor principale), care proiectează predictorii într-un alt spațiu, în care componentele sunt ortogonale
- prin PCA se pierde însă interpretabilitatea predictorilor (fiecare predictor din acel) spațiu devine o combinație liniară a predictorilor inițiali)

Analiza variabiliății (ANOVA)

- dorim să comparăm variabilitatea reziduală pentru două sau mai multe regresii
- pentru aceasta se calculează statistica F-test[3]:

$$F = \frac{\text{variabilitatea între grupuri}}{\text{variabilitatea în grupuri}} = \frac{\sum_{i=1}^{K} n_i(\bar{Y}_i - \bar{Y})^2/(K-1)}{\sum_{i=1}^{K}\sum_{j=1}^{n_i}(Y_{ij} - \bar{Y}_i)^2/(N-K)}$$

- $\bar{Y}_i$ este sample mean al grupului i, iar $\bar{Y}$ media pentru toate datele
- n_i numărul de observații din grupul i
- K este numărul de grupuri iar N numărul total de sample-uri
- Y_{ij} este observația j din grupul i
- statistica are o distribuție de tip F cu $d_1 = K - 1$ și $d_2 = N - K$ grade de libertate sub H_0

[3] https://en.wikipedia.org/wiki/F-test

Analiza variabiliății (2)

```
df = pd.read_csv('swiss.csv')
df.columns = np.r_[df.columns.values[:-1], ['InfantMortality']]
fit1 = smf.ols(formula='Fertility ~ Agriculture', data=df).fit()
fit2 = smf.ols(formula='Fertility ~ Agriculture + Examination + Education' \
    , data=df).fit()
fit3 = smf.ols(formula='Fertility ~ Agriculture + Examination + Education + \
    Catholic + InfantMortality', data=df).fit()

anova_lm(fit1, fit2, fit3)
```

Analiza variabiliății (3)

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	45.0	6283.115793	0.0	NaN	NaN	NaN
1	43.0	3180.924879	2.0	3102.190915	30.210744	6.389104e-09
2	41.0	2105.042930	2.0	1075.881948	10.477497	2.111080e-04

- df_resid: degrees of freedom, (număr de puncte - nr. parametri)
- SSR: Sum of Squared Residuals
- df_diff: degrees of freedom în exces față de modelul anterior
- F-statistic are o p-value asociată care arată dacă prin includerea predictorilor variabilitatea reziduurilor se schimbă fundamental (H_a) sau nu se schimbă (H_0)

Generalized Linear Models

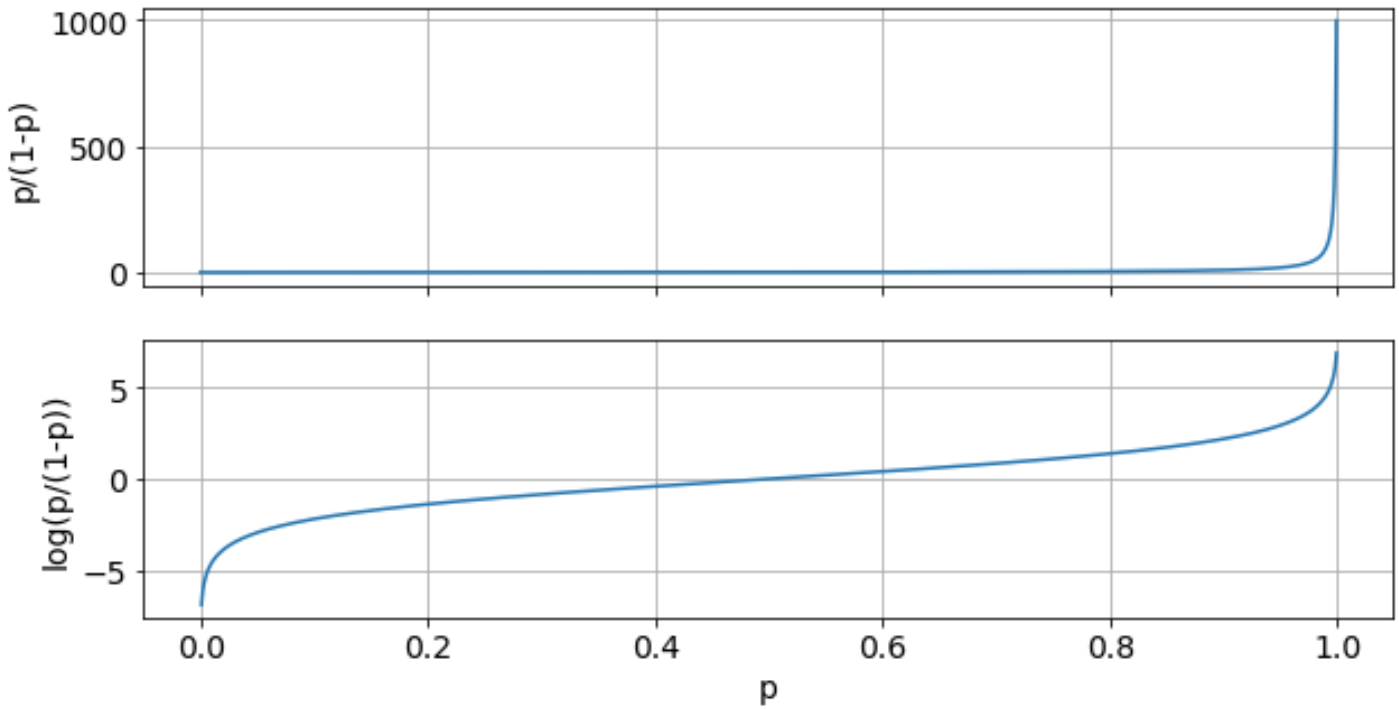
- compus din:
  - un model din familia exponențială[4] pentru răspuns (distribuțiile normală, binomială, Poisson, .. sunt distribuții exponențiale)
  - o componentă sistematică: predictorul liniar (componenta stochastică erau reziduurile)
  - o funcție de legătură între media modelului exponențial și predictorul liniar

- exemplu: modelul liniar
  - presupunem $Y_i \sim N(\mu_i, \sigma^2)$: distribuția Gaussiană face parte din familia exponențială
  - predictorul liniar este $\eta_i = \sum_{k=1}^{p} X_{ik}\beta_k$
  - funcția de legătură este g astfel încât $g(\mu) = \eta$
  - pentru modelul liniar $g(\mu) = \mu$, astfel încât $\mu_i = \eta_i$

[4] https://en.wikipedia.org/wiki/Exponential_family

Exemplu: regresia logistică

- presupunem că $Y_i \sim Bernoulli(\mu_i)$, astfel că $E[Y_i] = \mu_i$, unde $0 \leq \mu_i \leq 1$ (aruncarea monezii)
- predictorul liniar este $\eta_i = \sum_{k=1}^{p} X_{ik}\beta_k$
- funcția de legătură este $g(\mu) = \eta = log(\frac{\mu}{1-\mu})$
- g este referită sub denumirea de **logit** (log odds)
- se transformă media distribuției și nu Y_i direct



Exemplu: regresia Poisson

- presupunem că $Y_i \sim Poisson(\mu_i)$, astfel că $E[Y_i] = \mu_i$, unde $0 \leq \mu_i$ (modelarea unor variabile ce numără evenimente)
- predictorul liniar este $\eta_i = \sum_{k=1}^{p} X_{ik}\beta_k$ (la fel)
- funcția de legătură este $g(\mu) = \eta = log(\mu)$

Regresia logistică

- răspunsul Y_i este o variabilă liniară: success/failure, win/loss etc.
- realizările variabilei aleatoare sunt binare: 0/1, sau Bernoulli
- exemplu: Diabetes dataset

```
df = pd.read_csv('diabetes.csv')
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Aplicarea (eronată) a regresiei liniare

$$Out_i = b_0 + b_1 Gluc_i + \epsilon_i$$

- Out_i: 1 dacă persoana are diabet, 0 dacă nu
- $Gluc_i$: nivelul glucozei din sânge
- b_0: probabilitatea de a avea diabet dacă nivelul de glucoză e 0
- b_1: creșterea în probabilitatea de a avea diabet dacă nivelul glucozei crește cu 1 punct
- ϵ_i: valoarea reziduală rămasă neexplicată de regresie

```
model = smf.ols('Outcome ~ Glucose', data=df).fit()
model.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.4925	0.060	-8.262	0.000	-0.610	-0.375
Glucose	0.0070	0.000	14.600	0.000	0.006	0.008

Modelarea folosind rata (odds)

- rezultat binar 0/1

$$Out_i$$

- probabilitatea (0, 1)

$$Pr(Out_i|Gluc_i, b_0, b_1)$$

- rata (odds) $(0, \infty)$

$$\frac{Pr(Out_i|Gluc_i, b_0, b_1)}{1 - Pr(Out_i|Gluc_i, b_0, b_1)}$$

- log odds (logit) $(-\infty, \infty)$

$$log\left(\frac{Pr(Out_i|Gluc_i, b_0, b_1)}{1 - Pr(Out_i|Gluc_i, b_0, b_1)}\right)$$

Regresia liniară vs. regresia logistică

- regresia liniară

$$Out_i = b_0 + b_1 Gluc_i + \epsilon_i$$

- sau

$$E[Out_i | Gluc_i, b_0, b_1] = b_0 + b_1 Gluc_i$$

- regresia logistică

$$Pr(Out_i | Gluc_i, b_0, b_1) = \frac{exp(b_0 + b_1 Gluc_i)}{1 + exp(b_0 + b_1 Gluc_i)}$$

- sau

$$log\left(\frac{Pr(Out_i | Gluc_i, b_0, b_1)}{1 - Pr(Out_i | Gluc_i, b_0, b_1)}\right) = b_0 + b_1 Gluc_i$$

Interpretarea regresiei logistice

$$log\left(\frac{Pr(Out_i|Gluc_i, b_0, b_1)}{1 - Pr(Out_i|Gluc_i, b_0, b_1)}\right) = b_0 + \dots$$

$$Pr(Out_i|Gluc_i, b_0, b_1) = \frac{exp(b_0)}{1 + exp(b_0)}$$

- b_0 - log din rata (odds) ca să aibă diabet dacă nivelul de glucoză e 0
- trecem de la răspunsul regresiei, prin funcția odds și apoi în funcția de probabilitate
- b_1 - log odds (log rata) de a avea boala, pentru fiecare punct în plus la glicemie
- $exp(b_1)$ - odds (rata) de a avea boala, pentru fiecare punct în plus la glicemie
- $\frac{exp(b_1)}{1+exp(b_1)}$ - probabilitatea de a avea boala, pentru fiecare punct în plus la glicemie

Odds

- aruncarea monezii: rata de succes (heads) este probabilitatea de succes p
- dacă iese heads, câștigăm X; dacă iese tail, pierdem suma Y
- cum ar trebui alese X și Y pentru ca jocul să fie echilibrat?
- pe medie, câștigul să fie zero:

$$E[\text{câștiguri}] = Xp - Y(1-p) = 0$$

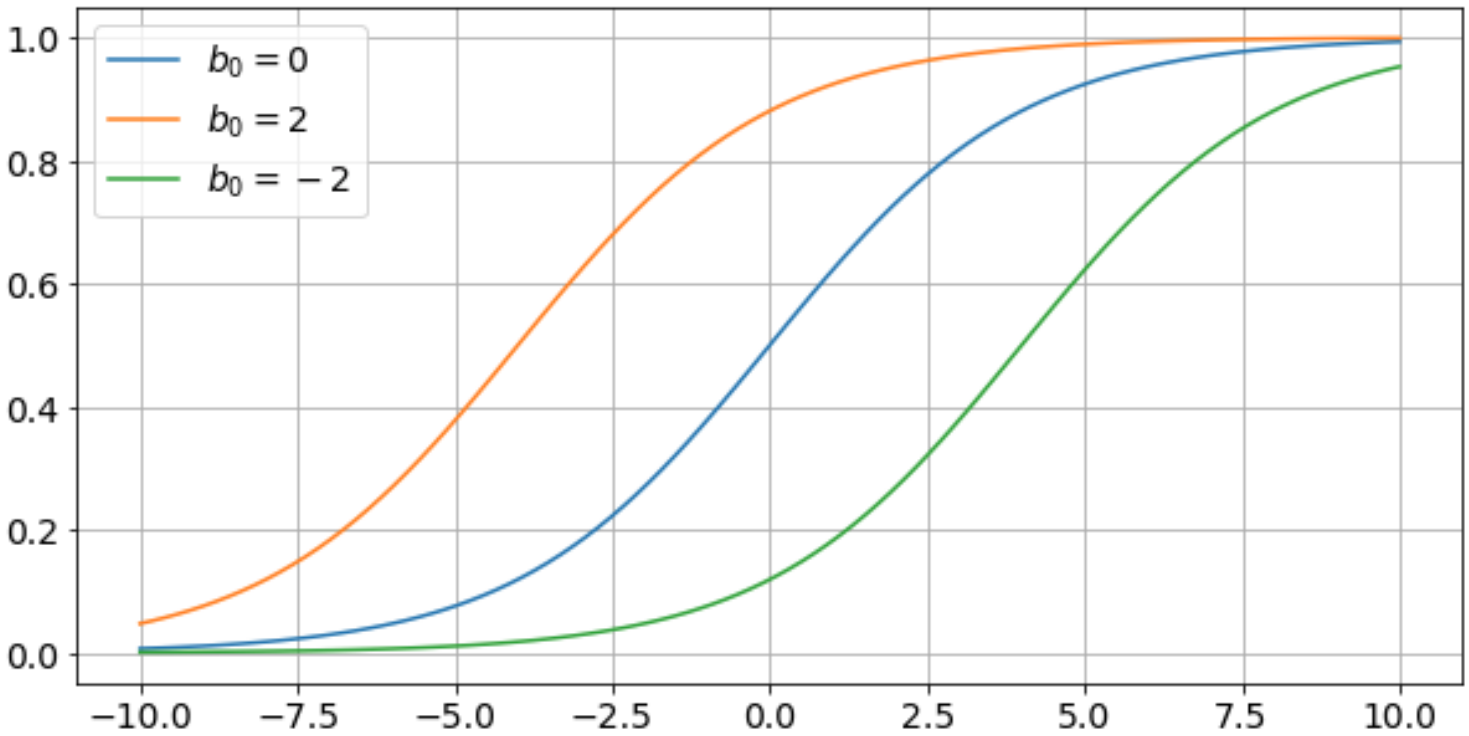
- adică

$$\frac{Y}{X} = \frac{p}{1-p} = odds$$

- odds poate fi exprimat astfel: 'cât de mult suntem dispuși să plătim pentru probabilitatea p de a câștiga un dolar?'
- exemplu: odds de 50/1 ca un cal să câștige. Dacă câștigă, casa ne plătește 50; dacă pierde, noi plătim 1. Probabilitatea de pierdere inerentă este 50/(50+1).

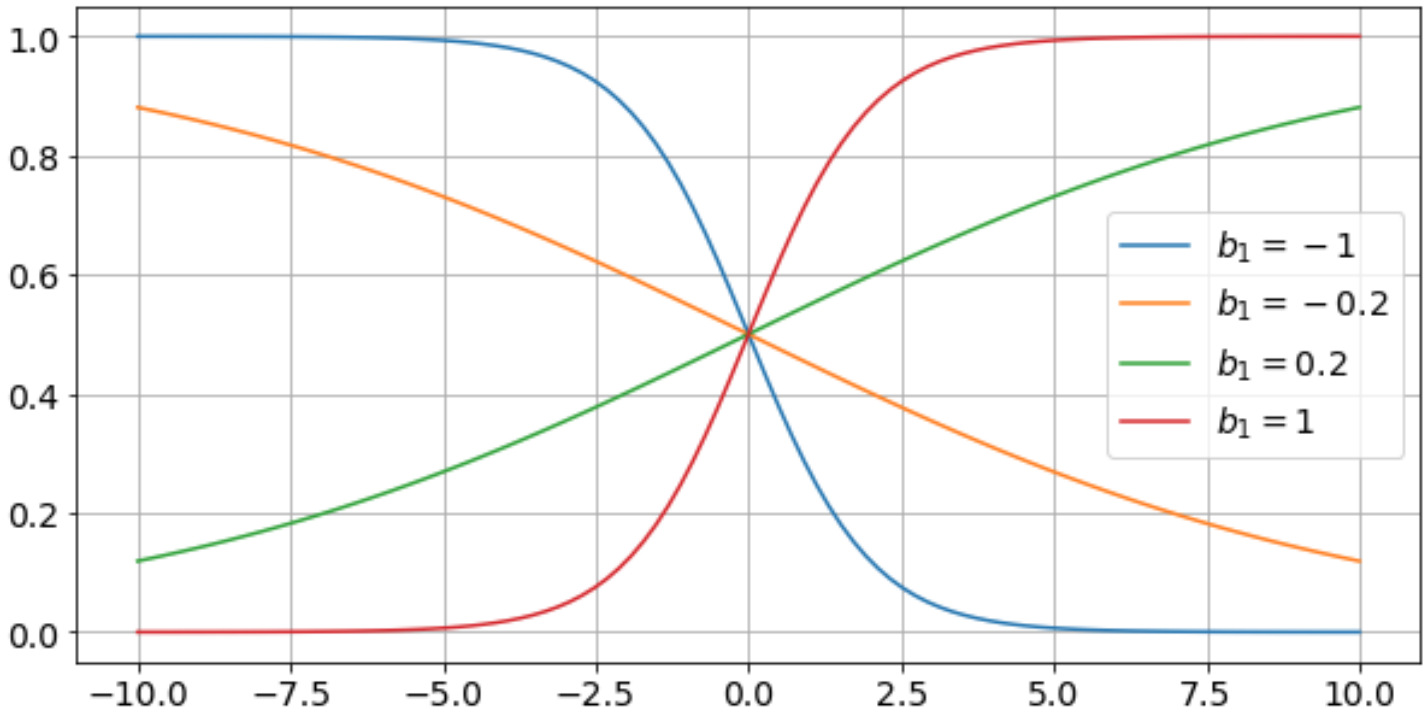
Interpretare (1)

```
x = np.linspace(-10, 10, 100)
def logit(x, b_0, b_1):
    o = b_0 + x * b_1
    return np.exp(o)/(1+np.exp(o))
fig, ax = plt.subplots(1, 1, figsize=(10, 5))
ax.plot(x, logit(x, 0, 0.5))
ax.plot(x, logit(x, 2, 0.5))
ax.plot(x, logit(x, -2, 0.5))
ax.grid() ; ax.legend(['$b_0=0$', '$b_0=2$', '$b_0=-2$']) ; plt.show()
```



Interpretare (2)

```
x = np.linspace(-10, 10, 100)
def logit(x, b_0, b_1):
    o = b_0 + x * b_1
    return np.exp(o)/(1+np.exp(o))
fig, ax = plt.subplots(1, 1, figsize=(10, 5))
ax.plot(x, logit(x, 0, -1))
ax.plot(x, logit(x, 0, -0.2))
ax.plot(x, logit(x, 0, 0.2))
ax.plot(x, logit(x, 0, 1))
ax.grid() ; ax.legend(['$b_1=-1$', '$b_1=-0.2$', '$b_1=0.2$', '$b_1=1$']) ; plt.show()
```



Regresia logistică (1)

```
model = smf.glm('Outcome ~ Glucose', data=df, family=sm.families.Binomial()).fit()
model.summary()
```

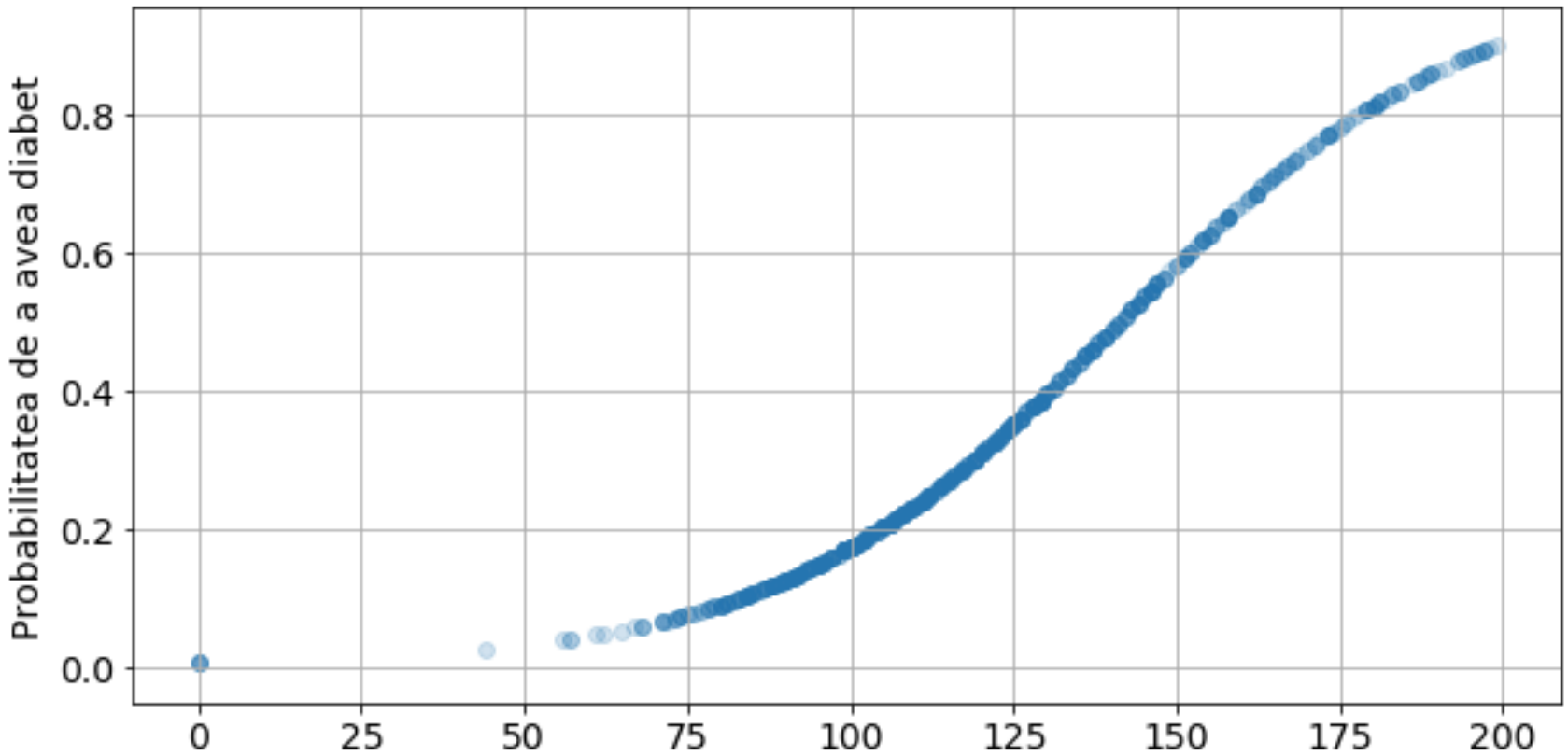
Generalized Linear Model Regression Results

Dep. Variable:	Outcome	**No. Observations:**	768
Model:	GLM	**Df Residuals:**	766
Model Family:	Binomial	**Df Model:**	1
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-404.36
Date:	Tue, 28 May 2019	**Deviance:**	808.72
Time:	10:30:31	**Pearson chi2:**	1.14e+03
No. Iterations:	5	**Covariance Type:**	nonrobust

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-5.3501	0.421	-12.713	0.000	-6.175	-4.525
Glucose	0.0379	0.003	11.647	0.000	0.031	0.044

Regresia logistică (2)

```
x = df.Glucose.values
y = model.fittedvalues
fig, ax = plt.subplots(1, 1, figsize=(10, 5))
ax.scatter(x, y, alpha=0.2)
ax.set_xlabel('Nivel de glucoză')
ax.set_ylabel('Probabilitatea de a avea diabet')
ax.grid() ; plt.show()
```



Regresia logistică (3)

```
np.exp(model.params)
```

```
Intercept    0.004748
Glucose      1.038599
dtype: float64
```

```
np.exp(model.conf_int())
```

	0	1
Intercept	0.002081	0.010832
Glucose	1.032001	1.045240

```
model.pvalues
```

```
Intercept    4.998114e-37
Glucose      2.380722e-31
dtype: float64
```

Bonus: intenția de participare la vot cu OLS

	România		Polonia		NSM, fără Romania și Polonia		Vechile state membre UE	
imagine pozitivă UE	.426	***	.756	***	.613	***	.683	***
apartenența la UE a adus beneficii țării*	.482	**	.412		.822	***	.752	***
orientare politică de dreapta*	1.054	***	.984	***	1.227	***	.776	***
orientare politică de stânga	.743	***	.983	***	.626	***	.769	***
satisfacție cu viața	.023		.197		.266	**	.309	***
optimism*	.854	***	-.250		.356	**	-.035	
are dificultăți în plata facturilor*	-.332	*	-.021		-.151		-.316	***
satisfacție cu lupta anticorupție în UE	.191	*	.194	**	.095		.015	
vârsta	.010	*	.020	***	.033	***	.024	***
bărbat*	-.009		.298	*	-.107		.082	
educație universitară*	.569	**	.727	***	.386	**	.531	***
maxim educație gimnazială*	-.508	*	-.620	*	-.221		-.835	***
locuiește în oraș mare*	.860	***	-.816	***	-.122		.149	**
locuiește la sat* (referință oraș mic)	.643	**	-.369		-.155		.023	
(Constant)	3.557		1.167		.721		2.190	
R2	.209		.144		.169		.182	
N	860		1659		2139		14744	

Sursa de date: EB90.1, septembrie 2018. Regresie OLS. Variabila dependentă - probabilitatea de a fi prezent la vot pe o scală de la 1 (nu, sigur) la 10 (da, sigur). Date ponderate cu w23 din fisierul Eurobarometru 90.1. * la predictori - variabila dihotomică unde 1 da, 0 nu. Asteriscurile de la coeficienți indică nivelul de semnificație * 0.05, ** 0.01, *** 0.001. Variabilele referitoare la

Bonus: intenția de participare la vot cu OLS (2)

`http://www.contributors.ro/analize/`

intenții-de-vot- ale-romanilor-la-europarlamentarele-din-2019-comparații-europene-și-sondaje-contradictorii/