



## CSC380: Principles of Data Science

### Linear Models 4

Credit:

- Jason Pacheco,
- Kwang-Sung Jun,
- Chicheng Zhang
- Xincheng yu

1

## Outline

2

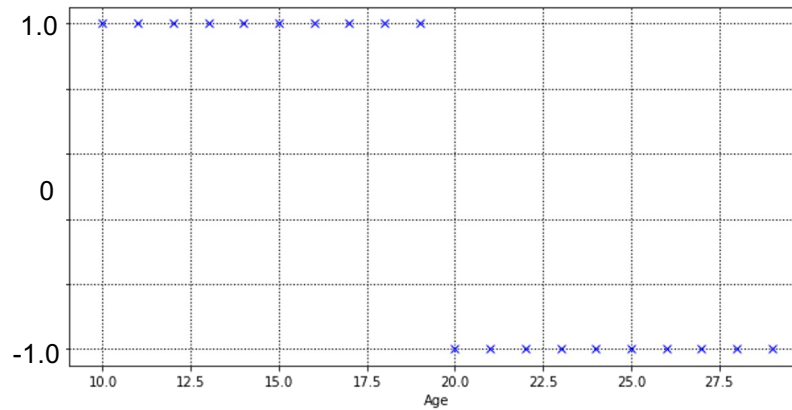
- Linear Regression
- Least Squares Estimation
- Regularized Least Squares
- Logistic Regression

2

## Classification as Regression

3

Suppose our response variables are binary  $y=\{-1,1\}$ . How can we use linear regression ideas to solve this classification problem?

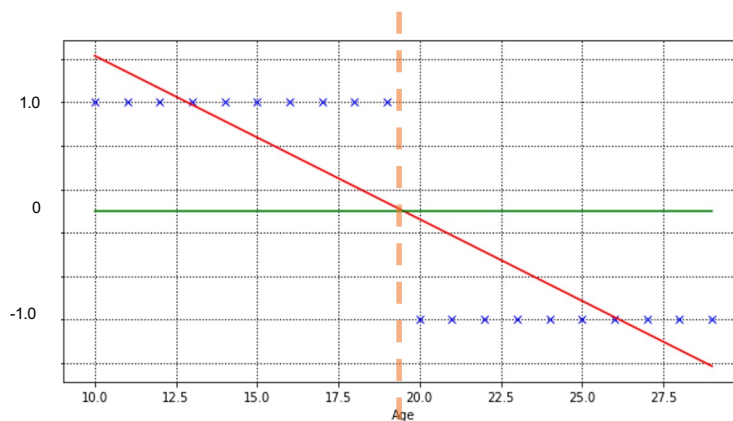


3

## Classification as Regression

4

**Idea** Fit a regression function (red) to the data. Classify points based on whether they are *above* or *below* the (green).



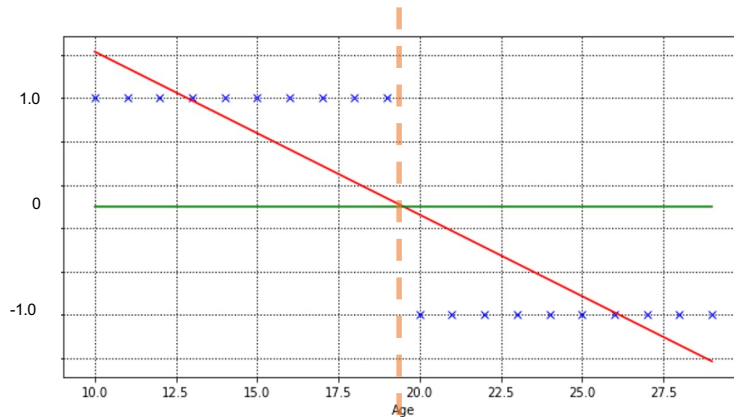
predict 1 if  $w^T x \geq 0$   
0 if  $w^T x < 0$

4

## Classification as Regression

5

**Idea** Fit a regression function (red) to the data. Classify points based on whether they are *above* or *below* the (green).



predict 1 if  $w^T x \geq 0$   
0 if  $w^T x < 0$

Recall:

$$w^{L2} = \arg \min_w \sum_{i=1}^m (y^{(i)} - w^T x^{(i)})^2$$

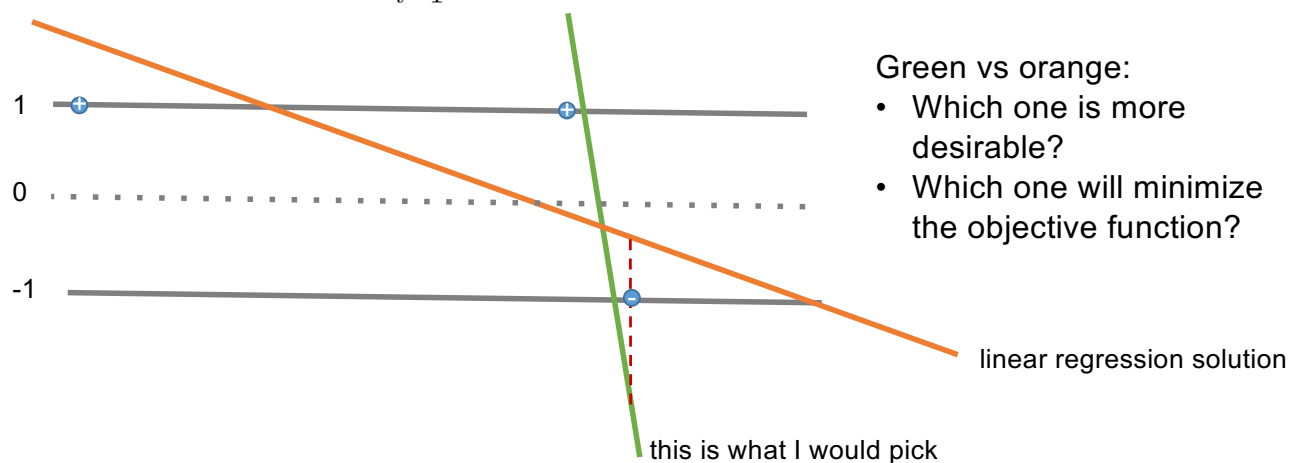
Turns out, this is not a desirable approach. Any guess?

5

## Classification as Regression is Not Desirable

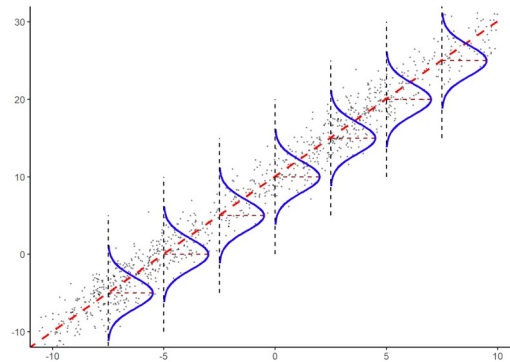
6

Recall:  $w^{L2} = \arg \min_w \sum_{i=1}^m (y^{(i)} - w^T x^{(i)})^2$



6

## Probability Assumptions



Recall the probabilistic motivation for linear regression:

Assume  $x \sim \mathcal{D}_x$  from some distribution. We then assume that

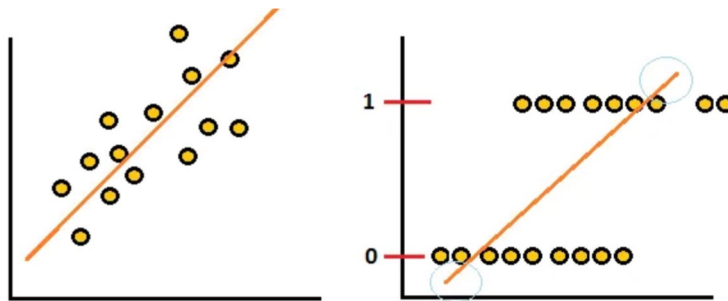
$$y = w^T x + \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Equivalently,

$$p(y|x; w) = \mathcal{N}(w^T x, \sigma^2)$$

7

## Probability Assumptions

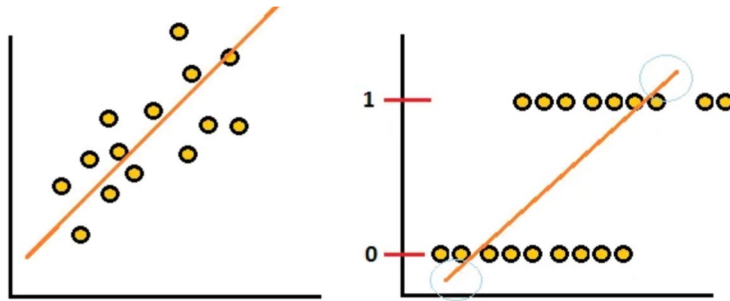


Q: What would be a reasonable alternative?

$$y \sim \text{Bernoulli}(p = w^T x)$$

8

## Probability Assumptions



Q: Once we compute the estimate  $\hat{w}$ , how do we make prediction for  $x^*$

$$y^* = \arg \max_{y' \in \{0,1\}} p(y = y' | x^*; \hat{w})$$

9

## Making Predictions

$$p = 0.4 \quad P(x = 1) = 0.4^1 \times 0.6^0 = 0.4$$

$$p^x \cdot (1 - p)^{1-x} \quad P(x = 0) = 0.4^0 \times 0.6^1 = 0.6 \quad \text{Prediction: 0}$$

Let's assume we have already learned the estimator:  $\hat{w} = 0.2$

$$y \sim \text{Bernoulli}(p = w^T x) \quad (\hat{w}x)^y \cdot (1 - \hat{w}x)^{1-y}$$

When  $x = 2$

$$y_{\text{predict}} = 0: (0.2 \times 2)^0 \times (1 - 0.2 \times 2)^1 = 0.6 \quad \text{Prediction: 0}$$

$$y_{\text{predict}} = 1: (0.2 \times 2)^1 \times (1 - 0.2 \times 2)^0 = 0.4$$

When  $x = 4$

$$y_{\text{predict}} = 0: (0.2 \times 4)^0 \times (1 - 0.2 \times 4)^1 = 0.2$$

$$y_{\text{predict}} = 1: (0.2 \times 4)^1 \times (1 - 0.2 \times 4)^0 = 0.8 \quad \text{Prediction: 1}$$



10

## Making Predictions

Let's assume we have already learned the estimator:  $\hat{w} = 0.2$

When  $x = 2$  **Prediction: 0**

$$y_{\text{predict}} = 0: (0.2 \times 2)^0 \times (1 - 0.2 \times 2)^1 = 0.6$$

$$y_{\text{predict}} = 1: (0.2 \times 2)^1 \times (1 - 0.2 \times 2)^0 = 0.4$$

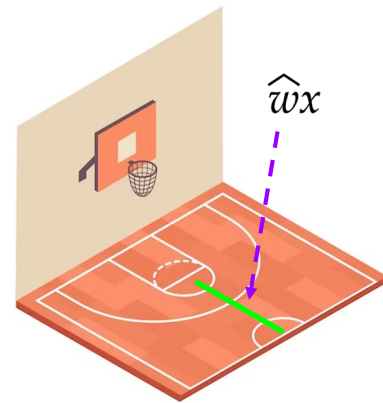
When  $x = 4$  **Prediction: 1**

$$y_{\text{predict}} = 0: (0.2 \times 4)^0 \times (1 - 0.2 \times 4)^1 = 0.2$$

$$y_{\text{predict}} = 1: (0.2 \times 4)^1 \times (1 - 0.2 \times 4)^0 = 0.8$$

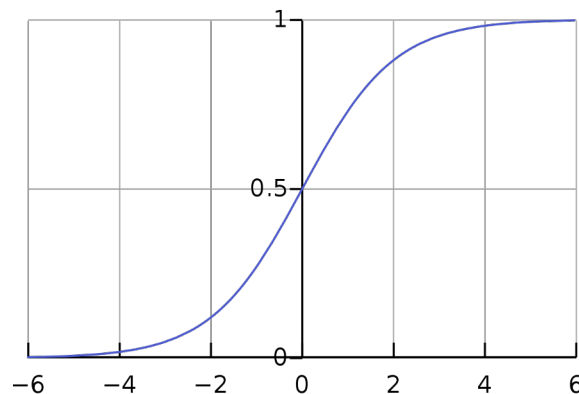
Q: what if  $x = 8$ ?

$$p = \hat{w}x = 0.2 \times 8 = 1.6$$



11

## Sigmoid Function



$$S(x) = \frac{1}{1 + e^{-x}}$$

12

## Logistic Regression

13

**Idea** Distort the prediction  $w^\top x$  in some way to map to  $[0,1]$  so that it is always a probability.

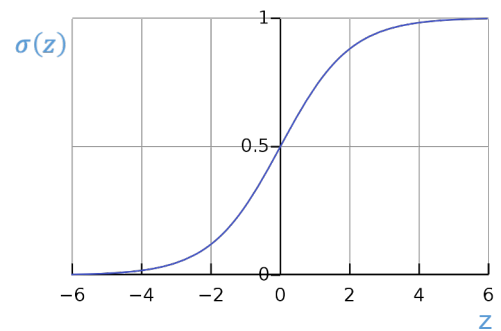
$$\sigma(w^\top x) \text{ instead of } w^\top x$$

where

$$\sigma(w^\top x) = \frac{\exp(w^\top x)}{1 + \exp(w^\top x)}$$

That is, assume

$$y \sim \text{Bernoulli}(p = \sigma(w^\top x))$$



- **Logistic function** is a type of *sigmoid function*, since it maps any value to the range  $[0,1]$
- Logistic also widely used in Neural Networks – for classification last layer is typically just a logistic regression

13

## Logistic Regression

14

**Model:**

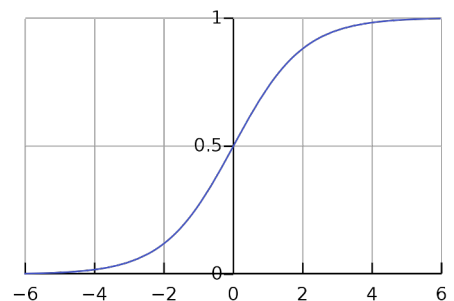
$$y \sim \text{Bernoulli}(p = \sigma(w^\top x))$$

**Train:** compute the MLE  $\hat{w}$

**Test:** Given test point  $x^*$  compute

$$y^* = \arg \max_{v \in \{-1,1\}} p(y = v \mid x^*; \hat{w})$$

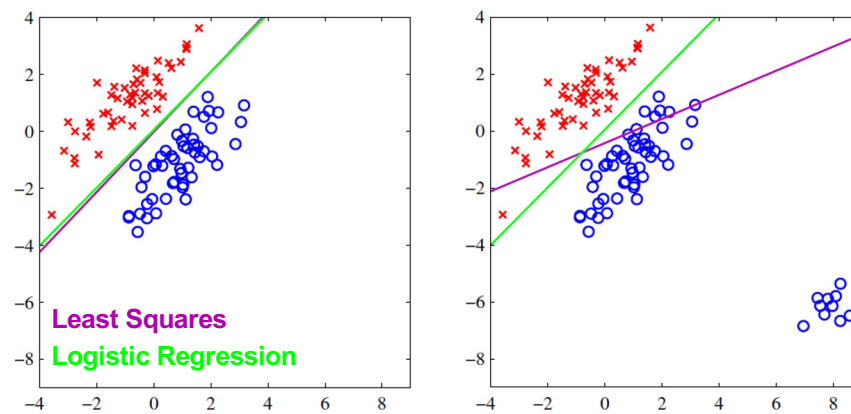
- Equivalent to  $y^* = \mathbf{I}\{\hat{w}^\top x^* \geq 0\}$



14

## Least Squares vs. Logistic Regression

15



- Both models learn a linear decision boundary
- 😊 Least squares can be solved in closed-form (convex objective)
- 😞 Least squares is sensitive to outliers

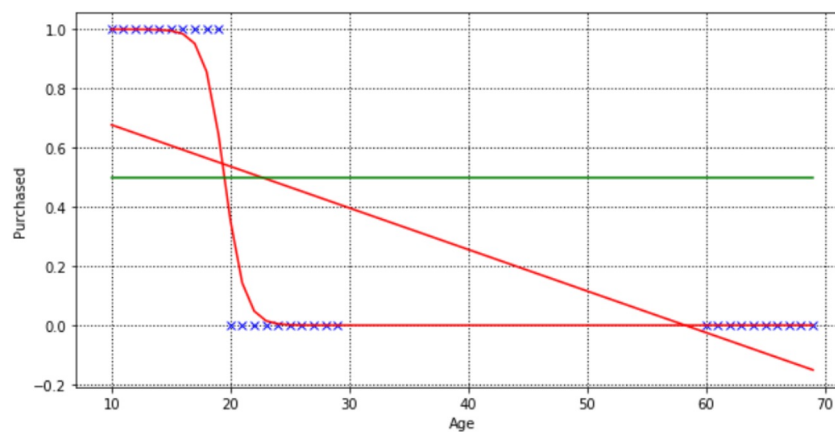
[Source: Bishop "PRML"]

15

## Least Squares vs. Logistic Regression

16

## Similar results in 1-dimension



16



## Fitting Logistic Regression

17

Fit by maximizing likelihood—start with the *binary* case

Posterior probability of class assignment is Bernoulli,

$$p(y \mid x; w) = p(y = 1 \mid x; w)^y (1 - p(y = 1 \mid x; w))^{(1-y)}$$

Given N iid training data pairs the log-likelihood function is,

$$\begin{aligned} \mathcal{L}_m(w) &= \sum_{i=1}^m \log p(y_i \mid x_i; w) \\ &= \sum_i \{y_i \log p(y_i = 1 \mid x_i; w) + (1 - y_i) \log p(y_i = 0 \mid x_i; w)\} \\ \text{(algebra)} \quad &= \sum_i \left\{ y_i w^T x_i - \log \left( 1 + e^{w^T x_i} \right) \right\} \end{aligned}$$

17

## Fitting Logistic Regression

18

$$w^{\text{MLE}} = \arg \max_w \sum_i \left\{ y^{(i)} w^T x^{(i)} - \log \left( 1 + e^{w^T x^{(i)}} \right) \right\}$$

Computing the derivatives with respect to each element  $w_d$ ,

$$\frac{\partial \mathcal{L}}{\partial w_d} = \sum_i x_d^{(i)} \left( y^{(i)} - \frac{e^{w^T x^{(i)}}}{1 + e^{w^T x^{(i)}}} \right) = 0$$

- Does not give a closed-form solution.
- Need to use iterative methods to solve it
- The objective function is concave => global solution can be found!

18

## Regularization

19

Regularization also works:

$$\begin{aligned} w^{\text{L2}} &= \arg \max_w \sum_i \left\{ y^{(i)} w^T x^{(i)} - \log \left( 1 + e^{w^T x^{(i)}} \right) \right\} - \lambda \|w\|^2 \\ &= \arg \min_w \sum_i \left\{ -y^{(i)} w^T x^{(i)} + \log \left( 1 + e^{w^T x^{(i)}} \right) \right\} + \lambda \|w\|^2 \end{aligned}$$

L1 regularization also possible

- Shares the same 'feature selection' property!

$$w^{\text{L1}} = \arg \min_w \sum_i \left\{ -y^{(i)} w^T x^{(i)} + \log \left( 1 + e^{w^T x^{(i)}} \right) \right\} + \lambda \|w\|_1$$

19

## Extension: Multiclass

20

- What if we have more than 2 classes?
- For  $C$  classes,

$$y | x \sim \text{Categorical}(\pi) \quad \text{with} \quad \pi_j = \frac{\exp(w^{(j)T} x)}{\sum_{c=1}^C \exp(w^{(c)T} x)} \quad \text{CD}$$

- Alternatively, one can use

$$\pi_j = \frac{\exp(w^{(j)T} x)}{1 + \sum_{c=1}^{C-1} \exp(w^{(c)T} x)} \quad \text{for } j = 1, \dots, C-1, \text{ and then define } \pi_C = \frac{1}{1 + \sum_{c=1}^{C-1} \exp(w^{(c)T} x)} \quad \text{(C-1)D}$$

Q: Number of parameters for the top one and the bottom one (say  $D$  features)?

20

21

## sklearn.linear\_model.LogisticRegression

```
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1,
class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False,
n_jobs=None, l1_ratio=None) 1
```

[source]

**penalty** : {'l1', 'l2', 'elasticnet', 'none'}, default='l2'

Specify the norm of the penalty:

- 'none': no penalty is added;
- 'l2': add a L2 penalty term and it is the default choice;
- 'l1': add a L1 penalty term;
- 'elasticnet': both L1 and L2 penalty terms are added.

**tol** : float, default=1e-4

Tolerance for stopping criteria.

**C** : float, default=1.0

$$C = 1/\lambda$$

Inverse of regularization strength; must be a positive float. Like in support vector machines, smaller values specify stronger regularization.

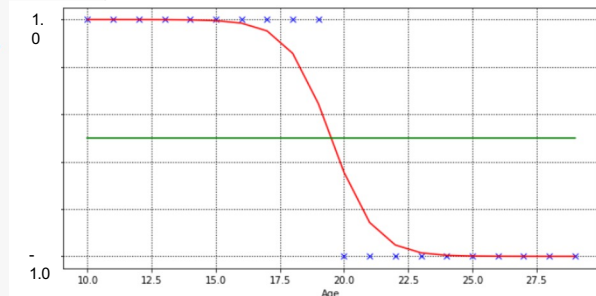
21

## Scikit-Learn Logistic Regression

22

```
log_regression = sklearn.linear_model.LogisticRegression()
_ = log_regression.fit(pd.DataFrame(x), y)
y_pred = log_regression.predict_proba(pd.DataFrame(x))
log_y_pred_1 = [item[1] for item in y_pred]

fig = plt.figure(figsize=(10,5))
xlabel = 'Age'
ylabel = 'Purchased'
plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.grid(color='k', linestyle=':', linewidth=1)
plt.plot(x, y, 'xb')
plt.plot(x, log_y_pred_1, '-r')
_ = plt.plot(x, line_point_5, '-g')
```



Function `predict_proba(X)` returns prediction of class assignment probabilities for each class. It returns n by C matrix if n data points were provided as argument.

<https://towardsdatascience.com/why-linear-regression-is-not-suitable-for-binary-classification-c64457be8e28>

22

## Using Logistic Regression

23

**The role of Logistic Regression differs in ML and Data Science,**

- In *Machine Learning*: use Logistic Regression for building **predictive** classification models
- In *Data Science*: use it for **understanding** how features relate to data classes / categories

**Example** South African Heart Disease (Hastie et al. 2001)

Data result from Coronary Risk-Factor Study in 3 rural areas of South Africa. Data are from white men 15-64yrs. Response is presence/absence of *myocardial infraction (MI)*.

Q: How predictive is each of the features?

23

## Example: African Heart Disease

	sbp	tobacco	ldl	famhist	obesity	alcohol	age	chd
0	160	12.00	5.73	1	25.30	97.20	52	1
1	144	0.01	4.41	0	28.87	2.06	63	1
2	118	0.08	3.48	1	29.14	3.81	46	0
3	170	7.50	6.41	1	31.99	24.26	58	1
4	134	13.60	3.50	1	25.99	57.34	49	1

### Features

- Systolic blood pressure
- Tobacco use
- Low density lipoprotein (ldl)
- Family history (discrete)
- Obesity
- Alcohol use
- Age

24

25



**Looking at Data**  
Each scatterplot shows pair  
of risk factors.  
Cases **with** MI (**red**) and  
**without** (**cyan**)

#### Features

- Systolic blood pressure
- Tobacco use
- Low density lipoprotein (ldl)
- Family history (discrete)
- Obesity
- Alcohol use
- Age

[Source: Hastie et al. (2001)]

25

## Example: African Heart Disease

26

	Coefficient	Std. Error	Z Score
(Intercept)	-4.130	0.964	-4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	-1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

**Goal:** hypothesis testing on  
whether the coefficient is 0 or not  
(hope to reject the hypothesis that  
the coefficient is 0)

Fit logistic regression to the data  
using MLE estimate

Standard error: estimated standard deviation of the learned coefficients

Z-score of coefficients is a random variable from standard Normal,

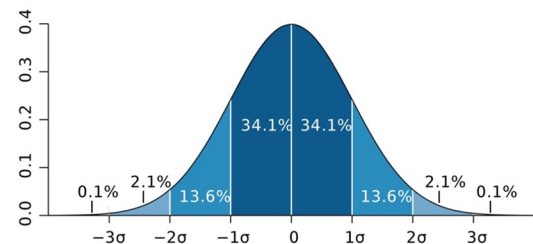
$$w_d \div \text{SE}(w_d) \sim \mathcal{N}(0, 1)$$

26

## Example: African Heart Disease

27

	Coefficient	Std. Error	Z Score
(Intercept)	-4.130	0.964	-4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	-1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184



Z-score of coefficients is a random variable from standard Normal,

$$w_d \div \text{SE}(w_d) \sim \mathcal{N}(0, 1)$$

Thus, anything with Z-score  $> 2$  is significant with 95% confidence.

27

## Example: African Heart Disease

28

	Coefficient	Std. Error	Z Score
(Intercept)	-4.130	0.964	-4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	-1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

**Finding** Systolic blood pressure (sbp) and alcohol are **not significant predictors**

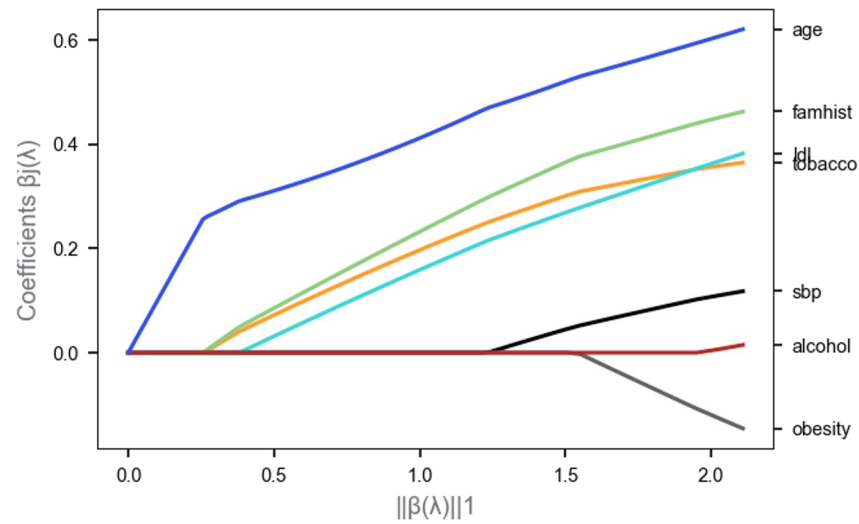
**Obesity** is not significant and negatively correlated with heart disease in the model

**Remember** All correlations / significance of features are based on presence of *other features*. We must always consider that features are strongly correlated.

**DO NOT INTERPRET IT AS CAUSALITY!**

28

## L1 regularized logistic regression coefficients



<https://github.com/empathv87/The-Elements-of-Statistical-Learning-Python-Notebooks/blob/master/examples/South%20African%20Heart%20Disease.ipynb>