Computer Science

CSC380: Principles of Data Science

Linear Models 4

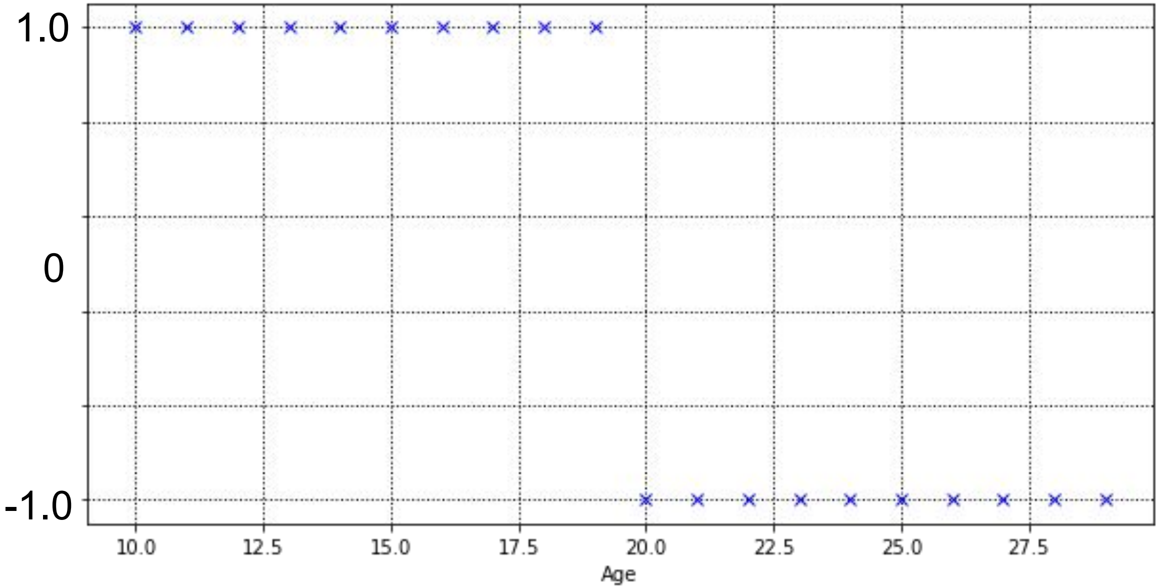
Xinchen Yu

Outline

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

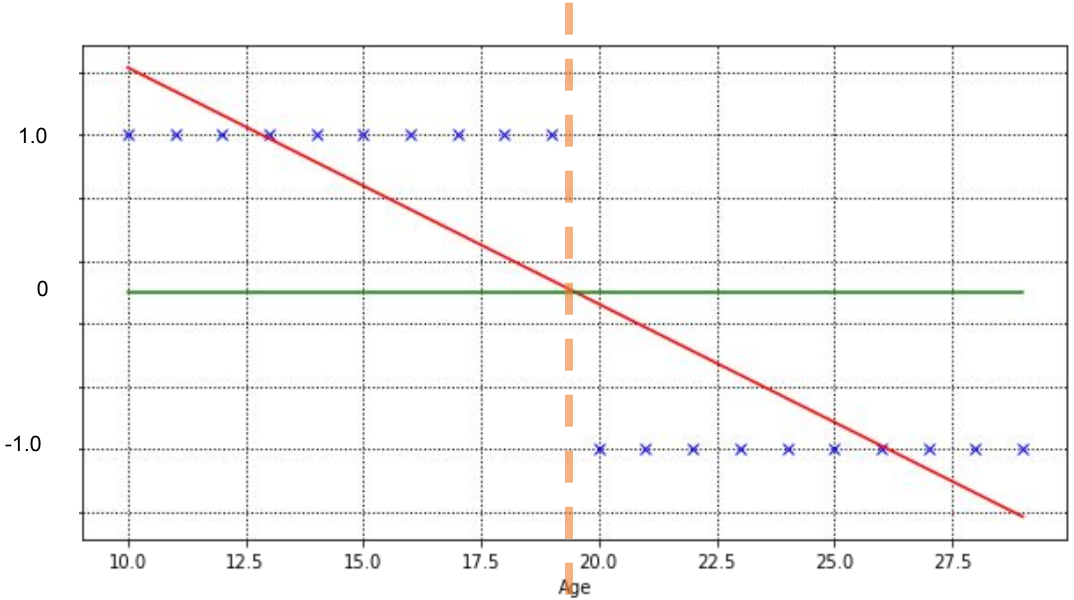
Classification as Regression

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



Classification as Regression

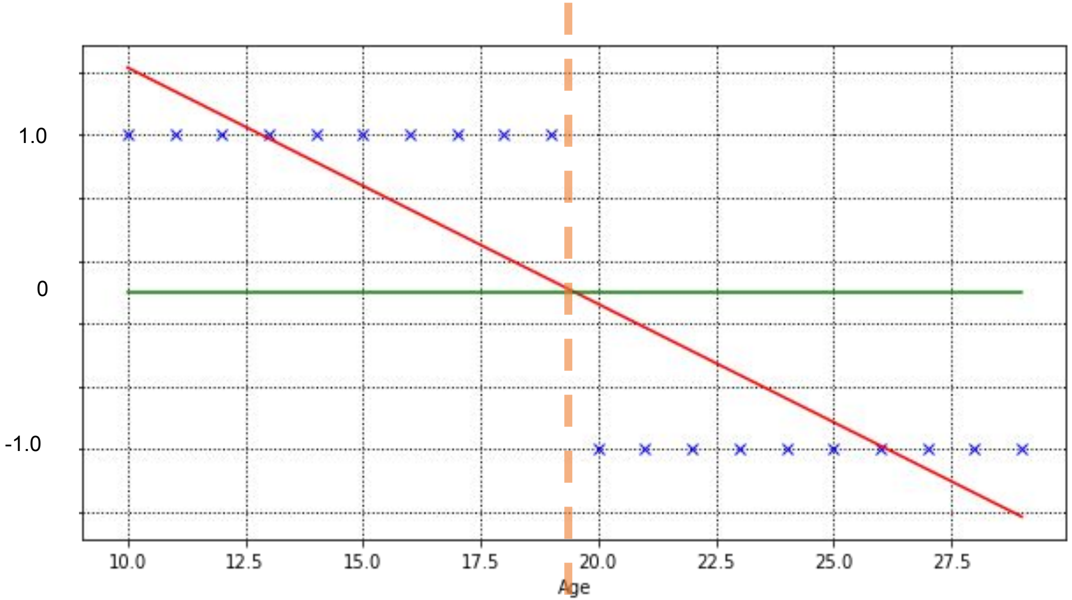
Idea Fit a regression function (red) to the data. Classify points based on whether they are *above* or *below* the (green).



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

Classification as Regression

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



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

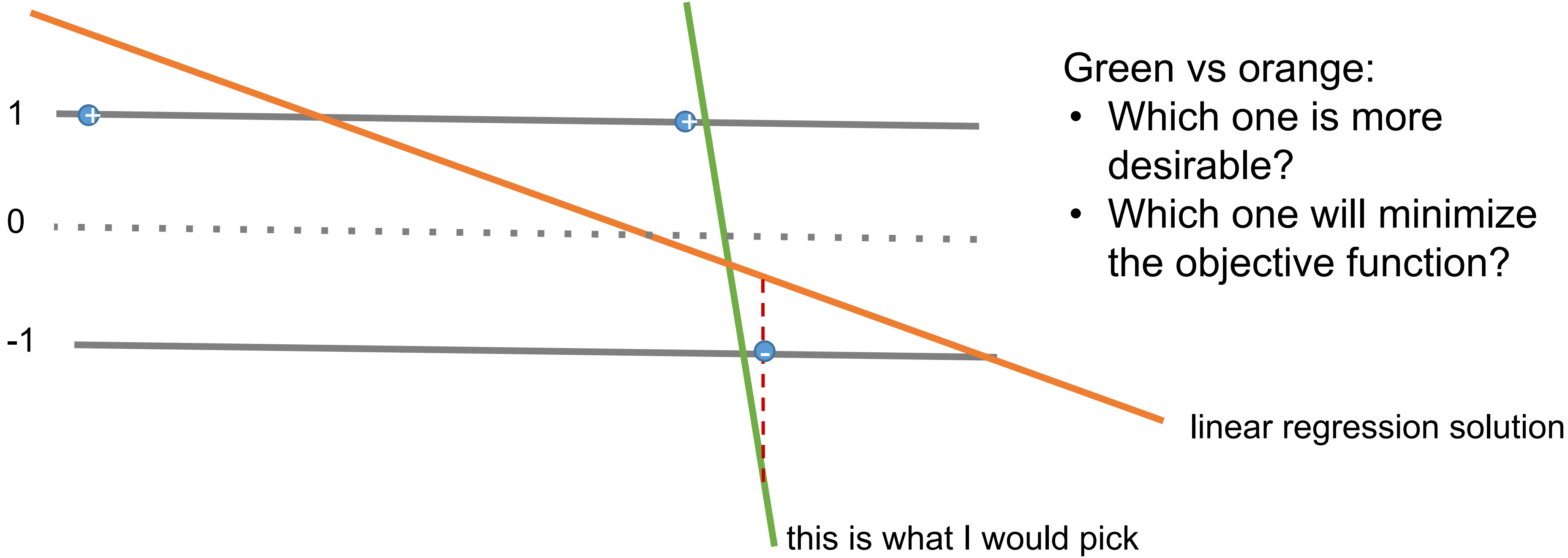
Recall:

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

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

Classification as Regression is Not Desirable

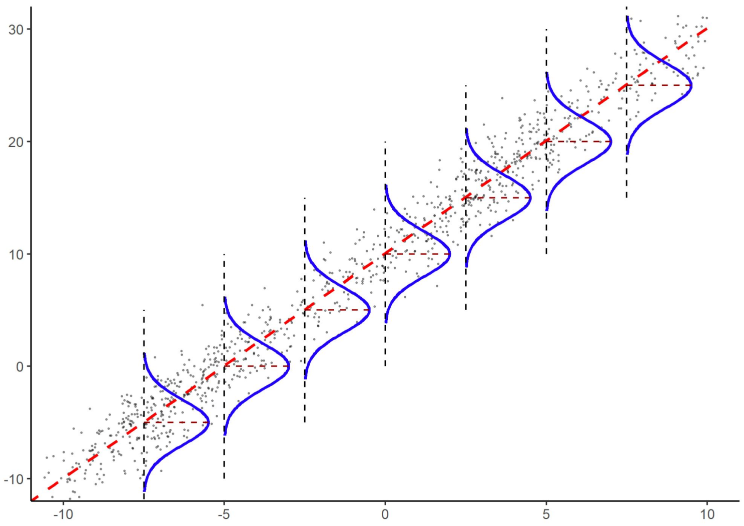
Recall: $$w^{\mathrm{L2}} = \arg\min_w \sum_{i=1}^{m} (y^{(i)} - w^T x^{(i)})^2$$



Green vs orange:

- Which one is more desirable?
- Which one will minimize the objective function?

Probability Assumptions



Recall the probabilistic motivation for linear regression:

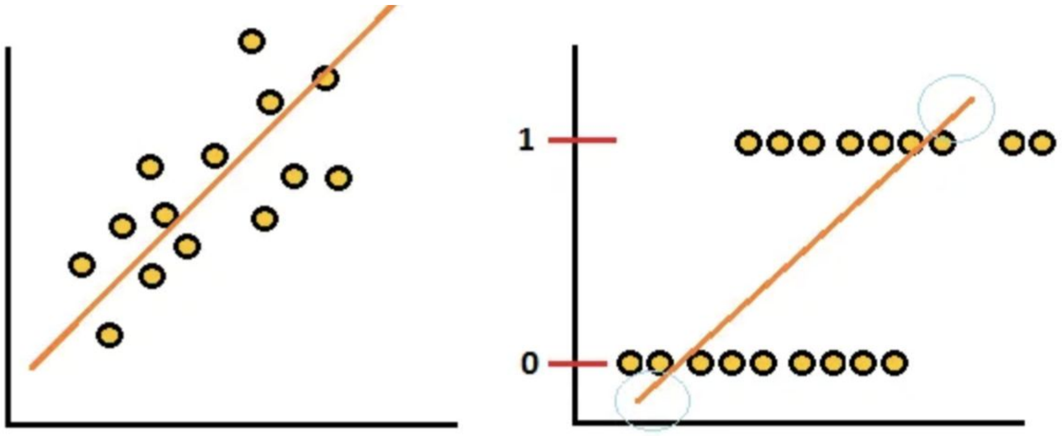
Assume $x \sim \mathcal{D}_X$ from some distribution. We then assume that

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

Equivalently,

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

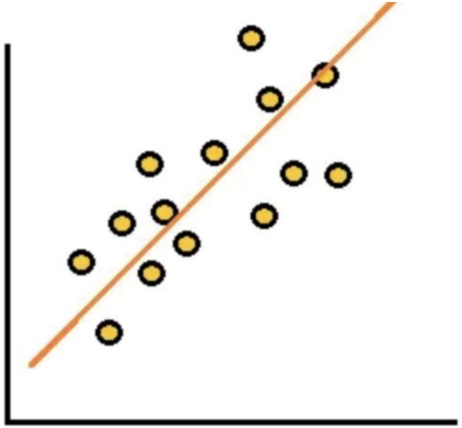
Probability Assumptions

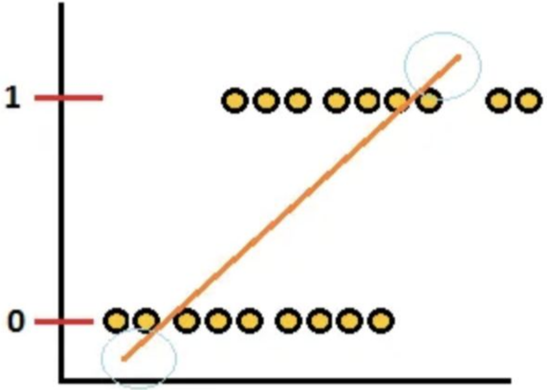


Q: What would be a reasonable alternative?

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

Probability Assumptions





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

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

Making Predictions

$$p = 0.4$$

$$p^x \cdot (1-p)^{1-x}$$

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

$$P(x=0) = 0.4^0 \times 0.6^1 = 0.6$$ Prediction: 0

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

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

When $x = 2$

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

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

When $x = 4$

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

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

Making Predictions

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

When $x = 2$ Prediction: 0

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

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

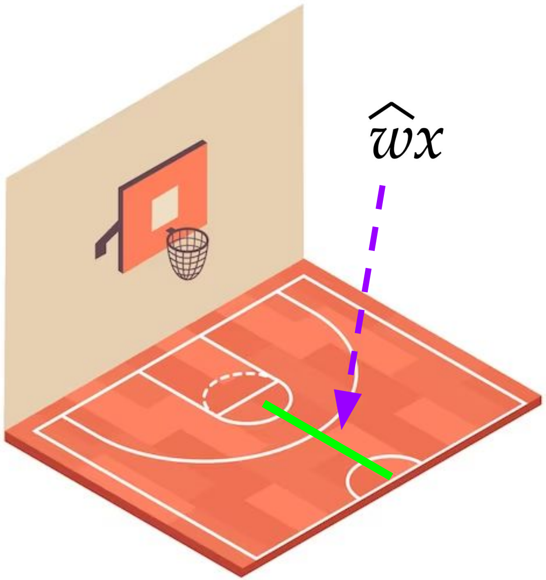
When $x = 4$ Prediction: 1

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

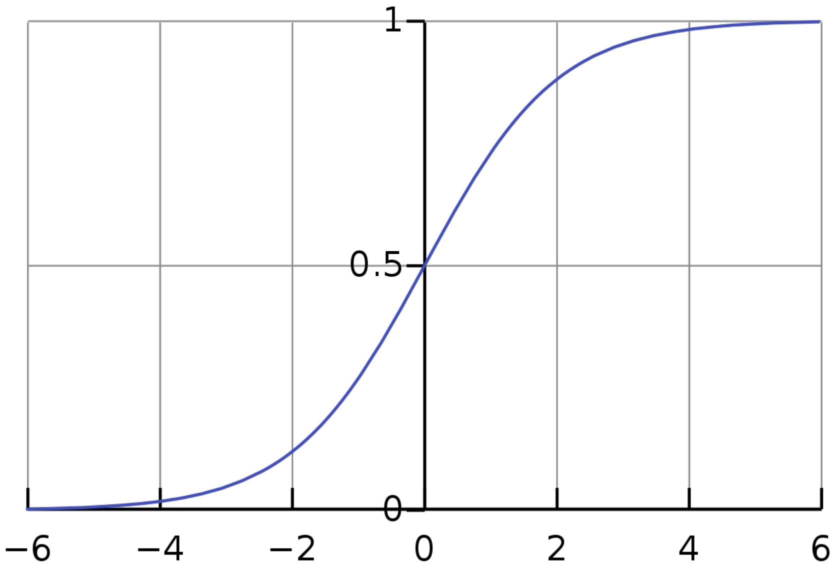
$y_{predict} = 1: \quad (0.2 \times 4)^1 \times (1 - 0.2 \times 4)^0 = 0.8$

Q: what if x = 8?

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



Sigmoid Function



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

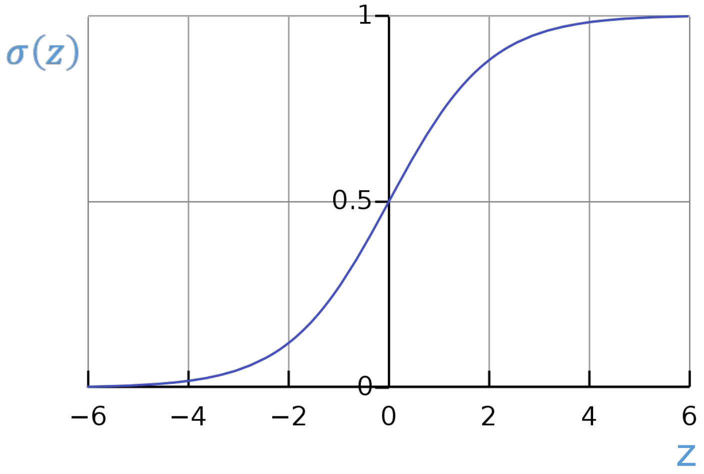
Logistic Regression

<u>Idea</u> 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

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



That is, assume

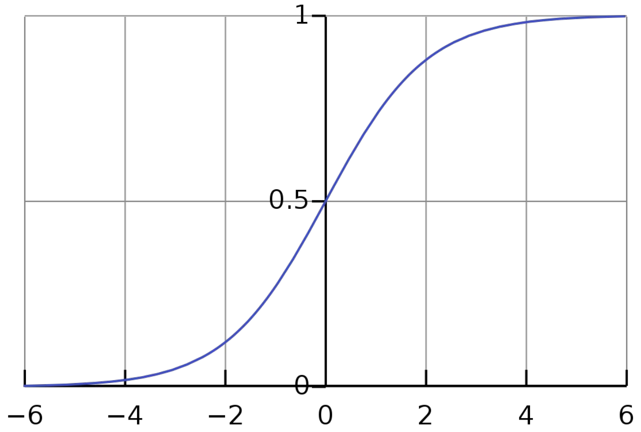
$$y \sim \text{Bernoulli}\big(p = \sigma(w^\top x)\big)$$

- **<u>Logistic function</u>** 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

Logistic Regression

Model:

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



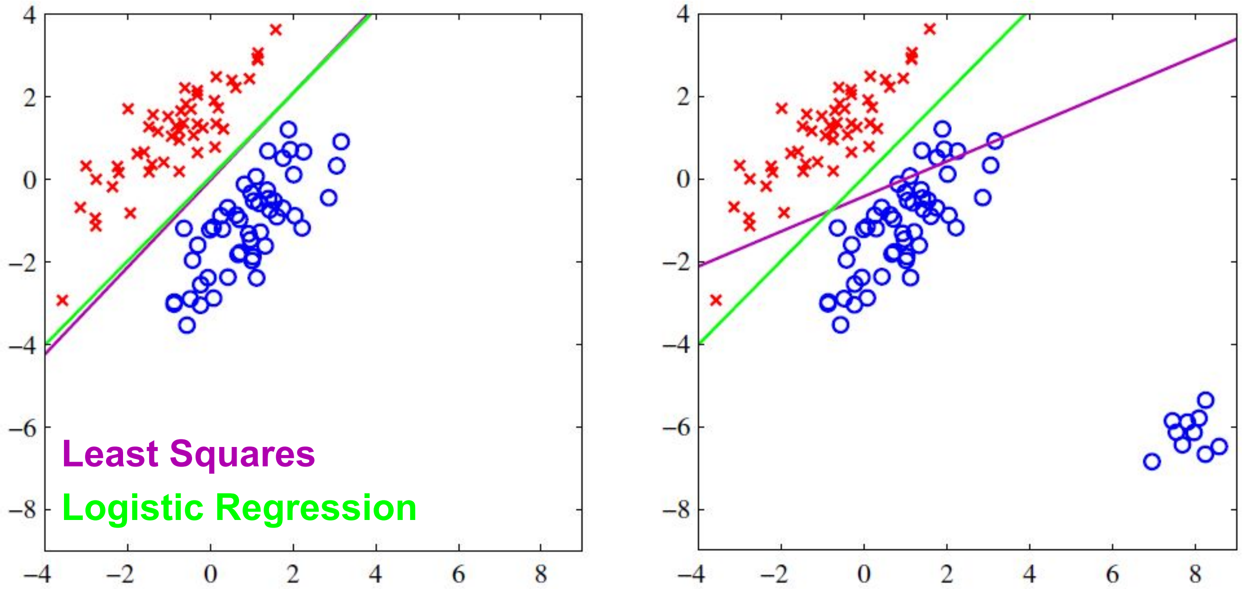
Train: compute the MLE $\hat{w}$

Test: Given test point x^* compute

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

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

Least Squares vs. Logistic Regression

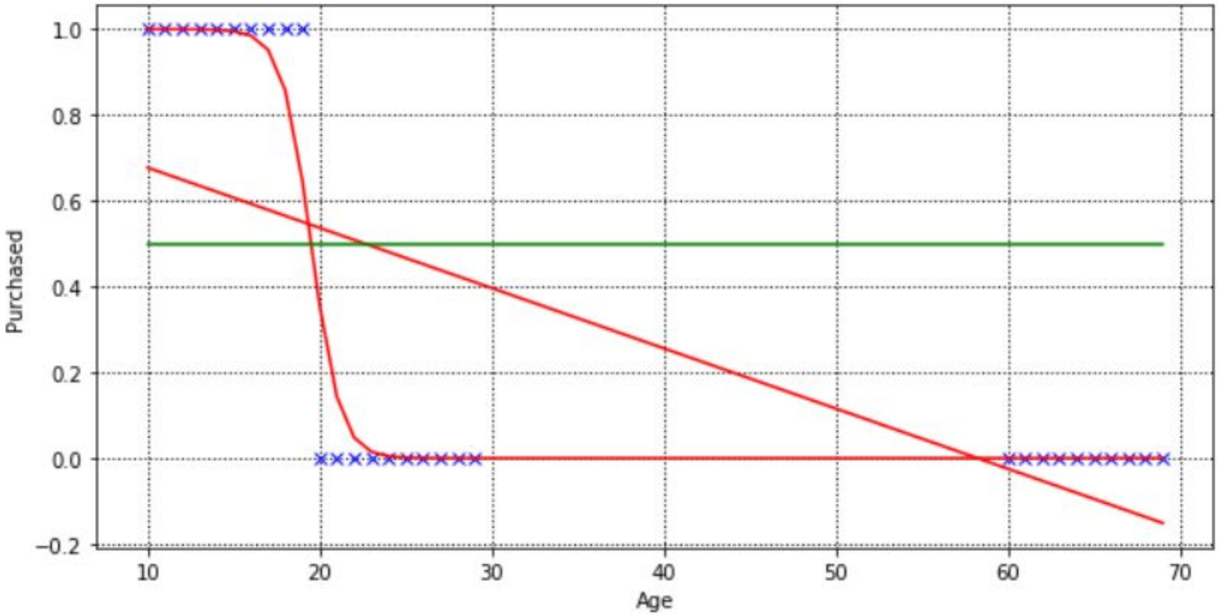


- Both models learn a linear decision boundary
- 😄 Least squares can be solved in closed-form (convex objective)
- ☹️ Least squares is sensitive to outliers

[Source: Bishop "PRML"]

Least Squares vs. Logistic Regression

Similar results in 1-dimension



Fitting Logistic Regression

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

Posterior probability of class assignment is Bernoulli,

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

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

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

Fitting Logistic Regression

$$w^{\mathrm{MLE}} = \arg\max_{w} \sum_{i} \left\{ y^{(i)} w^T x^{(i)} - \log\left(1 + e^{w^T x^{(i)}}\right) \right\}$$

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!

Regularization

Regularization also works:

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

L1 regularization also possible

- Shares the same ‘feature selection’ property!

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

Extension: Multiclass

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

$$y \mid x \sim \text{Categorical}(\pi) \quad \text{with} \quad \pi_j = \frac{\exp\left({w^{(j)}}^\top x\right)}{\sum_{c=1}^{C} \exp\left({w^{(c)}}^\top x\right)}$$

CD

- Alternatively, one can use

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

(C-1)D

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

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*) ¶ [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.

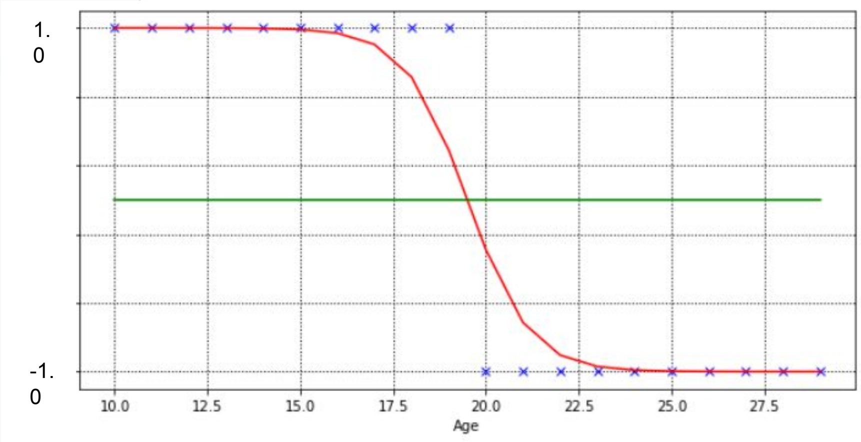
Scikit-Learn Logistic Regression

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

Using Logistic Regression

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

- In *Machine Learning* we use Logistic Regression for building **predictive** classification models
- In *Data Science* we often 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).* How predictive are each of the features?

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

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)]

Example: African Heart Disease

	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

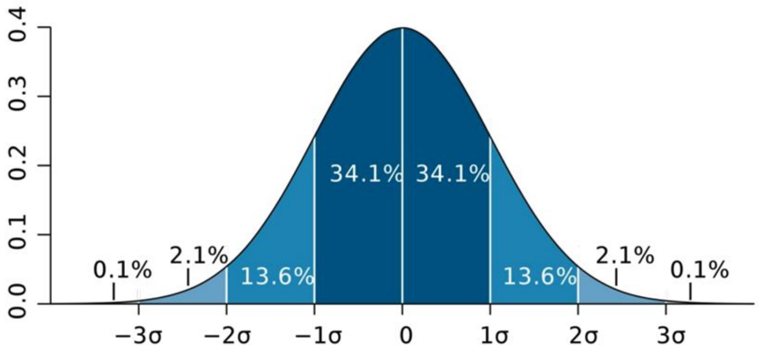
Standard error: estimated standard deviation of the learned coefficients

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

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

Example: African Heart Disease

	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,

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

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

Example: African Heart Disease

	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

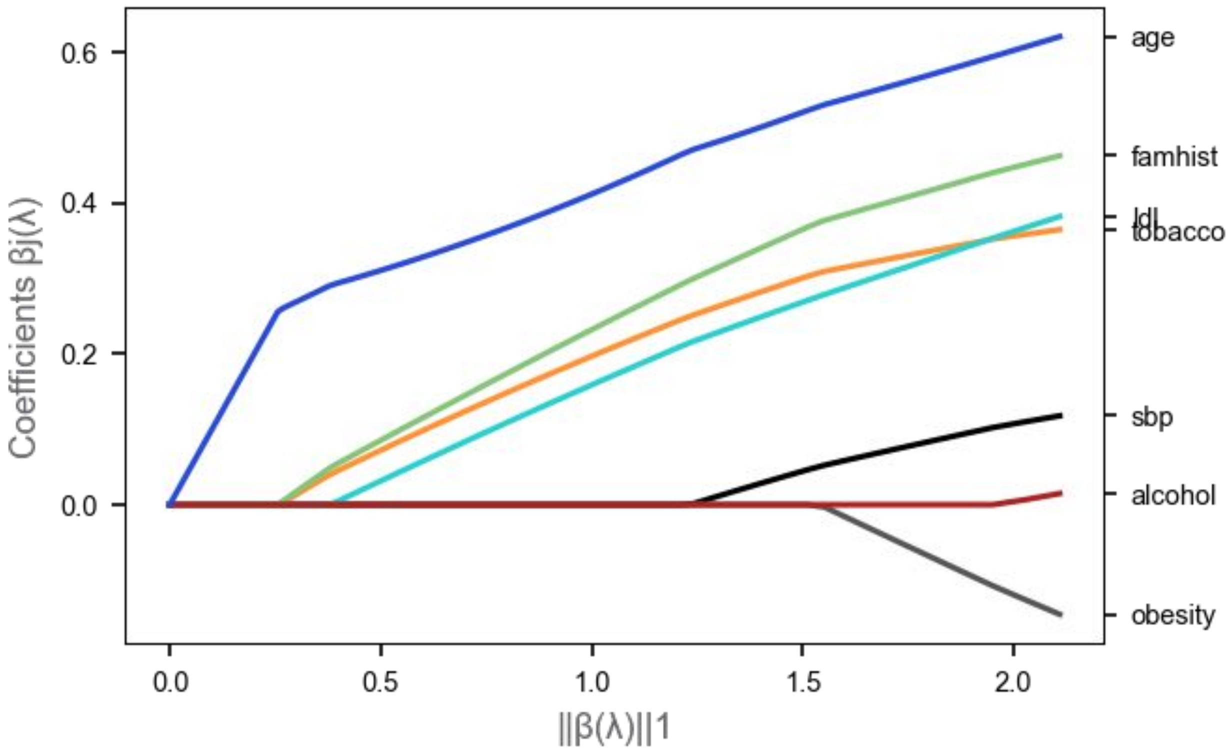
Finding Systolic blood pressure (sbp) and alcohol are **not significant predictors**

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!

L1 regularized logistic regression coefficients



https://github.com/empathy87/The-Elements-of-Statistical-Learning-Python-Notebooks/blob/master/examples/South%20African%20Heart%20Disease.ipynb