Linear Regression. Logistic Regression. Overfitting and Regularization.

COMP5318 Machine Learning and Data Mining

semester 1, 2021, week 3

Irena Koprinska

Reference: Witten ch.4: 128-131, Müller & Guido: ch.2: 28-31, 47-63,
Geron : ch.4 112-117, 134-140, 128-131

Outline

- Linear regression
- Logistic regression
- Overfitting and regularization
- Ridge and Lasso regression

Introduction

- Linear regression is a prediction method used for *regression* tasks
  - Regression tasks – the predicted variables is numeric
  - Examples: predict the exchange rate of AU$ based on economic indicators, predict the sales of a company based on the amount spent for advertising
- Logistic regression is an extension of linear regression for *classification* tasks
  - Classification tasks – the predicted variable is nominal
- Both linear regression and logistic regression are very popular in statistics

THE UNIVERSITY OF SYDNEY

Linear Regression

Simple (Bivariate) Regression

- Given: a dataset with 2 continuous variables:
  - feature x (also called independent variable)
  - predicted variable y (also called target variable or dependent variable)

- Goal: Approximate the relationship between these variables with a straight line for the given dataset
  - Prediction (typical task in DM): Given a new value of independent variable, use the line to predict the value of the dependent variable
  - Descriptive analysis (typical task in psychology, health and social sciences): assess the strength of the relationship between x and y

Example – *cereals* dataset

- Contains nutritional information for 77 breakfast cereals
- 14 features
  - cereal manufacturer, type (hot or cold), calories, protein [g], fat [g], sodium [mg], fiber [g], carbohydrates [g], sugar [g], potassium [mg], %recommended daily vitamins, weight of 1 serving, number of cups per serving, shelf location (bottom, middle or top)
- Class variable (numeric): nutritional rating

- Task: Predict the nutritional rating of a cereal based on its sugar content
  1. Use this data to build the model
  2. Given the sugar content of a new cereal, use the model to predict is nutritional rating
     - New cereal = cereal not used for building of the model

Task

Task: Predict the nutritional rating of a cereal based on its sugar content

1. Use this data to build the model
2. Given the sugar content of a new cereal, use the model to predict is nutritional rating

Cereal Name	Manuf.	Sugars	Calories	Protein	Fat	Sodium	Rating
100% Bran	N	6	70	4	1	130	68.4030
100% Natural Bran	Q	8	120	3	5	15	33.9837
All-Bran	K	5	70	4	1	260	59.4255
All-Bran Extra Fiber	K	0	50	4	0	140	93.7049
Almond Delight	R	8	110	2	2	200	34.3848
Apple Cinnamon Cheerios	G	10	110	2	2	180	29.5095
Apple Jacks	K	14	110	2	0	125	33.1741
Basic 4	G	8	130	3	2	210	37.0386
Bran Chex	R	6	90	2	1	200	49.1203
Bran Flakes	P	5	90	3	0	210	53.3138
Cap'n crunch	Q	12	120	1	2	220	18.0429
Cheerios	G	1	110	6	2	290	50.7650
Cinnamon Toast Crunch	G	9	120	1	3	210	19.8236
Clusters	G	7	110	3	2	140	40.4002
Cocoa Puffs	G	13	110	1	1	180	22.7364

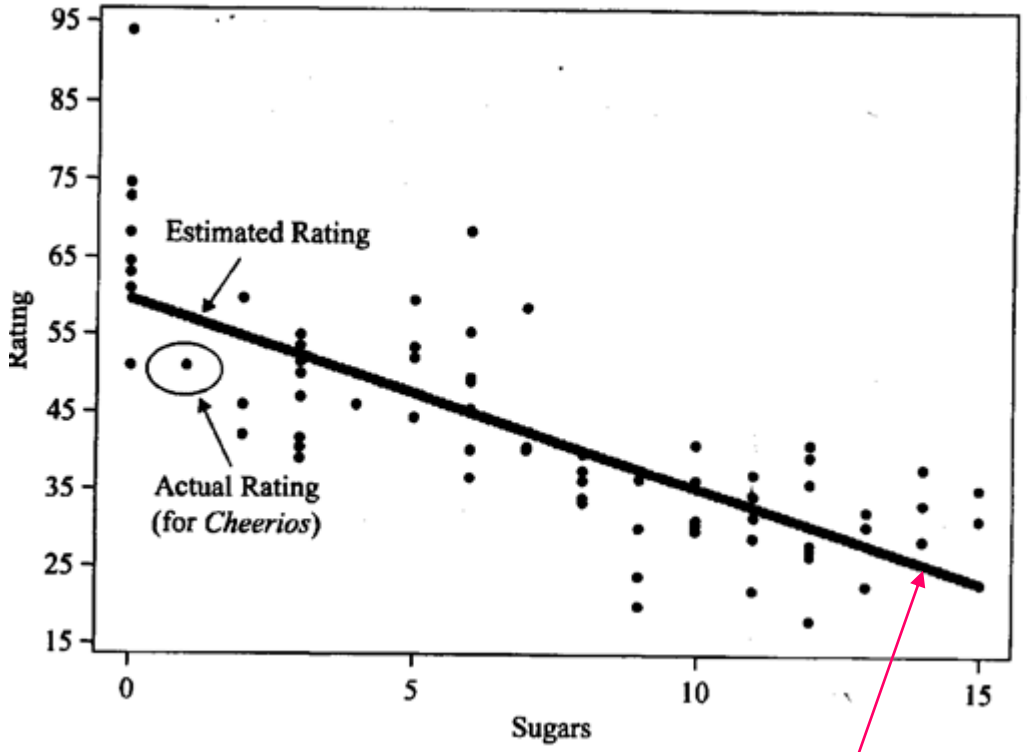
Dependent variable?

Independent variable?

Regression model

- The relationship between sugars and rating is modeled by a line
- The line is used to make predictions

Cereal Name	Manuf.	Sugars	Calories	Protein	Fat	Sodium	Rating
100% Bran	N	6	70	4	1	130	68.4030
100% Natural Bran	Q	8	120	3	5	15	33.9837
All-Bran	K	5	70	4	1	260	59.4255
All-Bran Extra Fiber	K	0	50	4	0	140	93.7049
Almond Delight	R	8	110	2	2	200	34.3848
Apple Cinnamon Cheerios	G	10	110	2	2	180	29.5095
Apple Jacks	K	14	110	2	0	125	33.1741
Basic 4	G	8	130	3	2	210	37.0386
Bran Chex	R	6	90	2	1	200	49.1203
Bran Flakes	P	5	90	3	0	210	53.3138
Cap'n crunch	Q	12	120	1	2	220	18.0429
Cheerios	G	1	110	6	2	290	50.7650
Cinnamon Toast Crunch	G	9	120	1	3	210	19.8236
Clusters	G	7	110	3	2	140	40.4002
Cocoa Puffs	G	13	110	1	1	180	22.7364

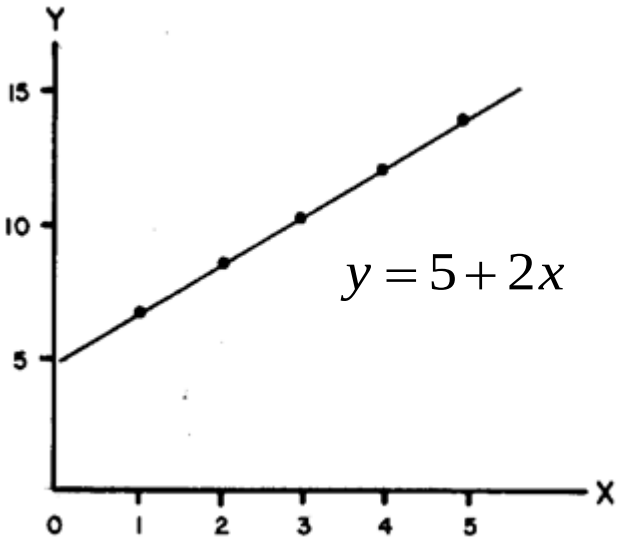


model (regression line)

Equation of a line

$$y = b_0 + b_1 x$$

intercept slope

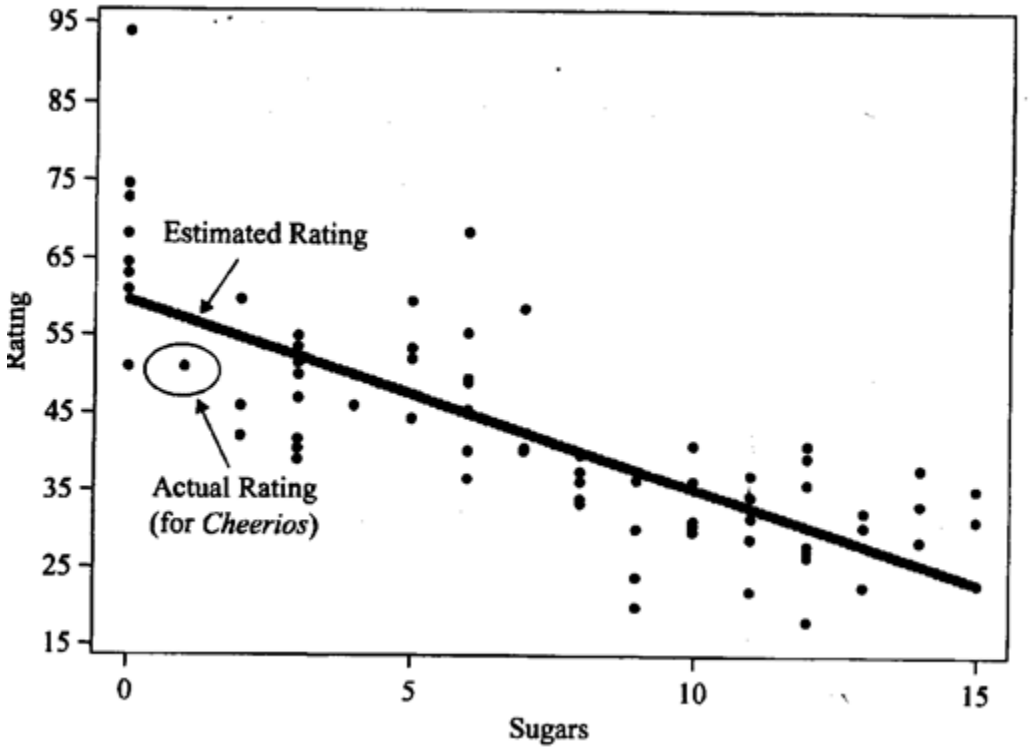


Equation of a regression line

$$\hat{y} = b_0 + b_1 x$$

$\hat{y}$ Estimated (predicted) value of y from the regression line

b_0 and b_1 Regression coefficients



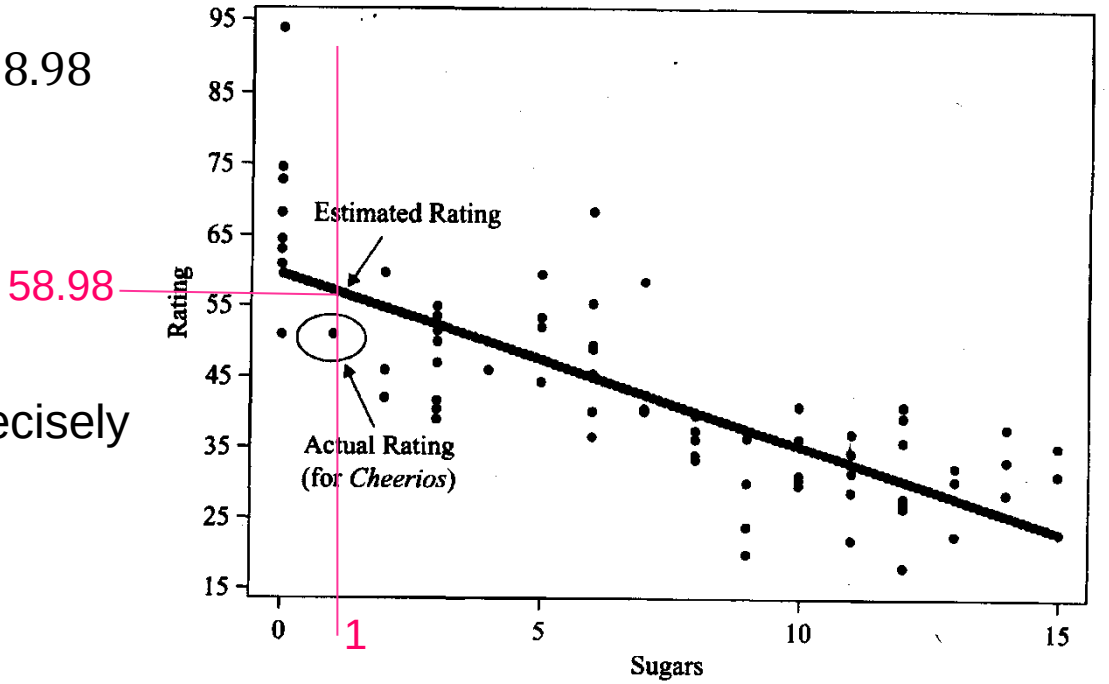
How to make predictions

- In our case the computed regression line (model) is

  $\hat{y} = 59.4 - 2.42x$

- It can be used to make predictions
  - e.g. predict the nutritional rating of a new cereal type (not in the original data) that contains x=1g sugar

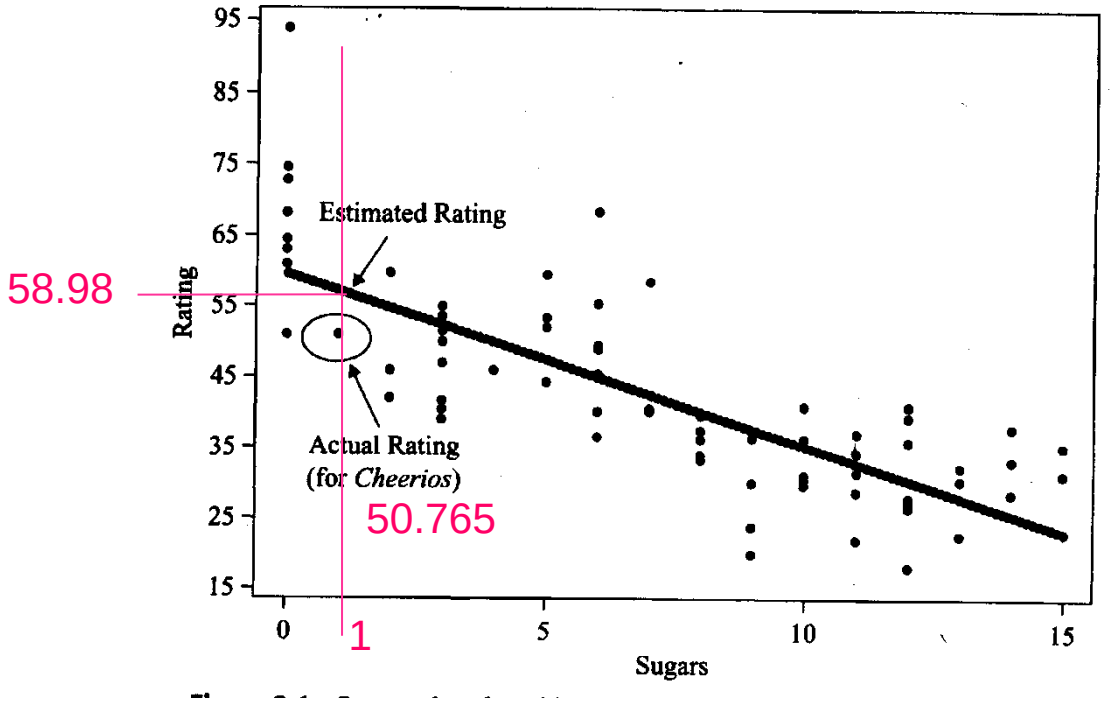
$\hat{y} = 59.4 - 2.42 * 1 = 58.98$



- The predicted value lies precisely on the regression line

How to make predictions (2)

- We have a cereal type in our dataset with sugar =1g: Cheerios
- Its nutritional rating is: 50.765 (actual value) not 58.98 (predicted)
- The difference is called prediction error or residual

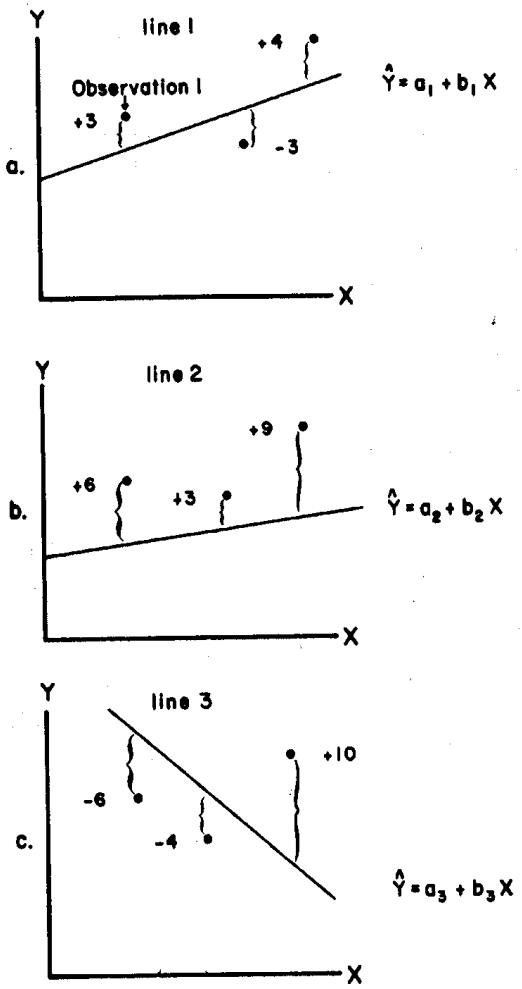


Fitting a line

- There are many lines that can be fitted to the given dataset. Which one is the best one?
  - The one “closest” to the data
  - Mathematically:
    - Prediction error (residual) = observed-predicted value =

$$= \varepsilon = y_i - \hat{y}_i$$

- Performance index: sum of squared prediction errors (SSE): $SSE = \sum_i (y_i - \hat{y}_i)^2$
- Our goal: select the line which minimizes SSE
- Can be solved using the *method of the least squares*



Solution using the least squares method

$$b_1 = \frac{\sum x_i y_i - [(\sum x_i)(\sum y_i)]/n}{\sum x_i^2 - (\sum x_i)^2/n}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

$\bar{x}$ – mean value of *x*

$\bar{y}$ – mean value of *y*

n – number of training examples (data points, observations)

- This solution is obtained by minimizing SSE using differential calculus

Coefficient of determination R^2

- The least squares method finds the best fit to the data but doesn't tell us how good this fit it
  - E.g. SSE=12; is this large or small?

- R^2 measures the *goodness of fit* of the regression line found by the least squares method:

$$R^2 = \frac{SSR}{SST}$$

- Values between 0 and 1; the higher the better
  - =1: the regression line fits perfectly the training data
  - close to 0: poor fit

- What are SSR and SST?

Three types of errors

- 1. SSE - Sum of squared *prediction* errors

$$SSE = \sum_{i}^{n} (y_i - \hat{y}_i)^2$$ = actual value – predicted value

- 2. SST - Sum of squared *total* errors

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$ = actual value – mean value

  - Hence, SST measures the prediction error when the predicted value is the mean value
  - SST is a function of the variance of y (variance = standard deviation^2) => SST is a measure of the variability of y, without considering x

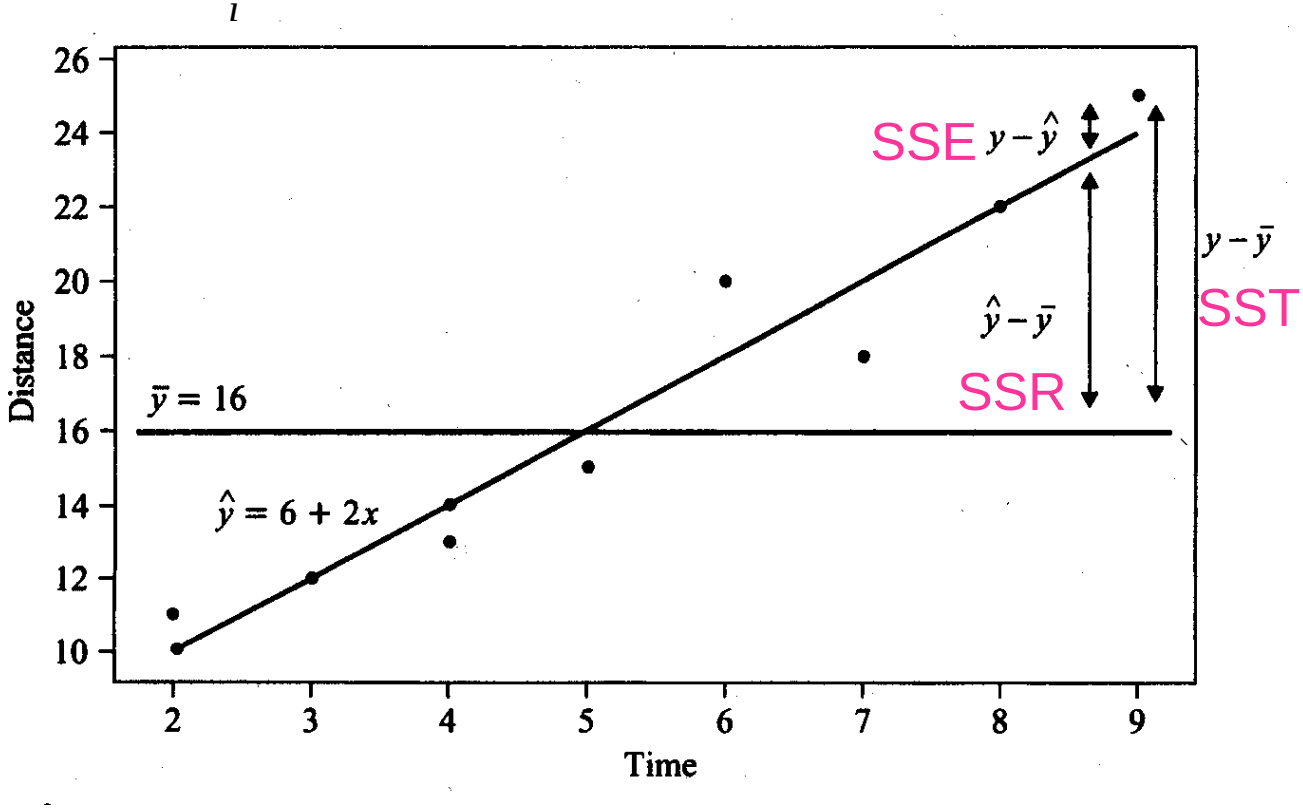
$$SST = \sum_{i}^{n} (y_i - \bar{y})^2 = (n-1)\,\mathrm{var}(y)$$

Can be used as a baseline - predicting y without knowing x

Three types of errors (2)

- 3. SSR - Sum of squared *regression* errors = predicted value – mean value

$$SSR = \sum_{i}^{n} (\hat{y}_i - \bar{y})^2$$



Ex.: Distance travelled for a number of hours

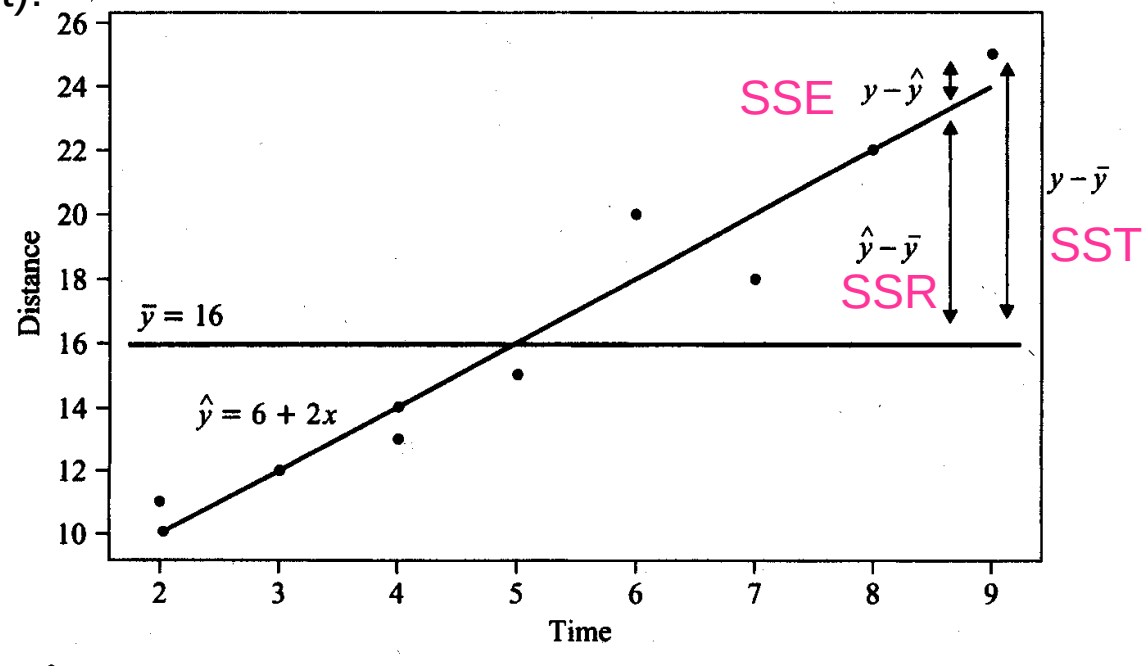
Subject	Time, x (hours)	Distance, y (km)	S
1	2	10	
2	2	11	
3	3	12	
4	4	13	
5	4	14	
6	5	15	
7	6	20	
8	7	18	
9	8	22	
10	9	25	

Relation between SST, SSR and SSE

- From the graph: $y_i - \bar{y}_i = (\hat{y}_i - \bar{y}_i) + (y_i - \hat{y}_i)$
- It can be shown that SST=SSR+SSE

Square each side (the cross product cancels out):

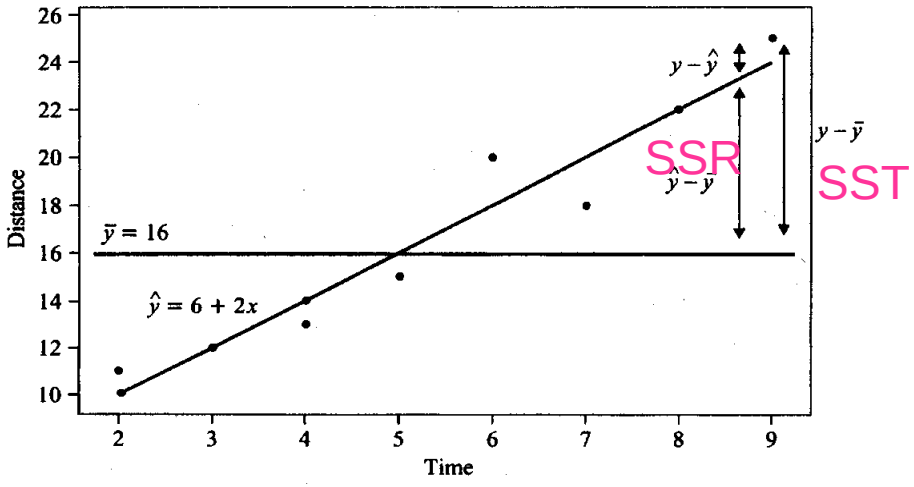
$$\sum_{i=1}^{n}(y_i - \bar{y}_i)^2 = \sum_{i=1}^{n}(\hat{y}_i - \bar{y}_i)^2 + \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$



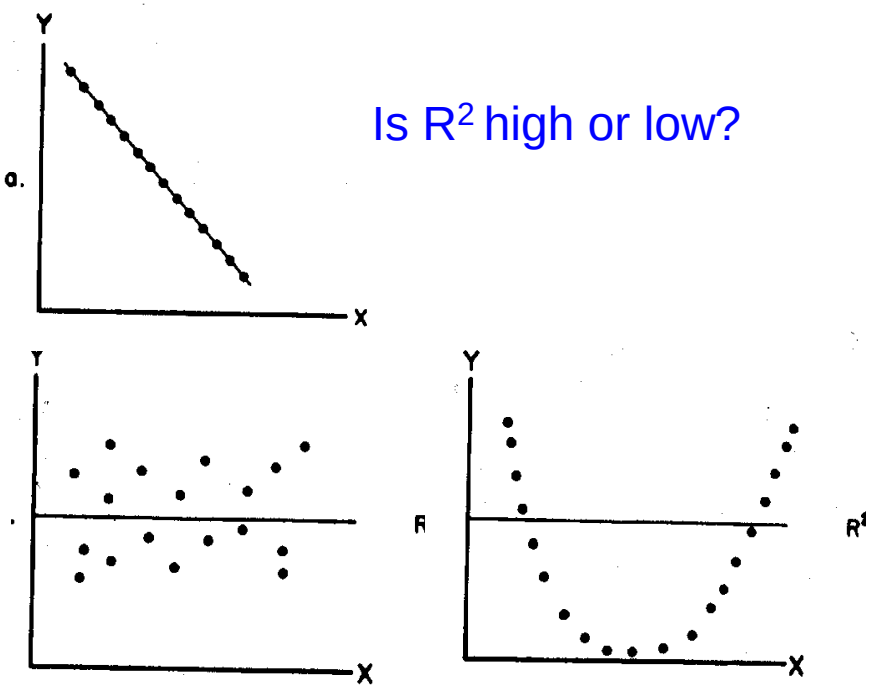
Coefficient of determination R^2 - again

$$R^2 = \frac{SSR}{SST}$$

- Measures the goodness of fit of the regression line to the training data
- Values between 0 and 1; the higher the better
  - 1: perfect fit, SSE=0 ; Why is it 1 when SSE=0?
  - 0: x is not helpful for predicting y, SSR=0



Is R^2 high or low?



Relation R^2 and r

- r - correlation coefficient; measures linear relationship between 2 vectors x and y (see slides for week 1b):

$$r = \text{corr}(\mathbf{x}, \mathbf{y}) = \frac{\text{covar}(\mathbf{x}, \mathbf{y})}{\text{std}(\mathbf{x})\,\text{std}(\mathbf{y})} = \frac{\text{covar}(\mathbf{x}, \mathbf{y})}{\sqrt{\text{var}(\mathbf{x})\,\text{var}(\mathbf{y})}}$$

- R^2 – coefficient of determination; measures how well the regression line represents the data: $R^2 = \frac{SSR}{SST}$

- It can be shown that $r = \sqrt{R^2}$

Except for the sign of r, which depends on the direction of the relationship, positive or negative, so: $r = \pm\sqrt{R^2}$

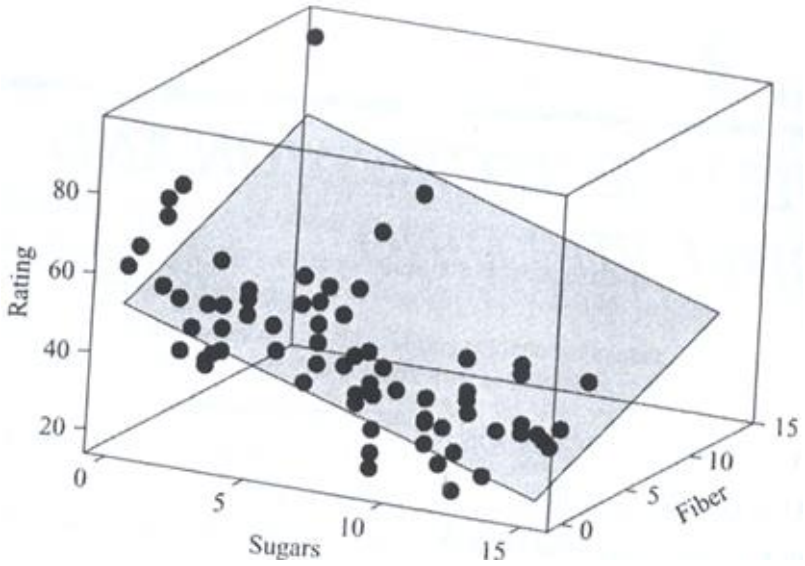
MAE, MSE and RMSE

- MAE, MSE and RMSE are other performance measures for evaluating:
  - how good the model is (performance on training data) and
  - how well it works on new data (performance on test data)
- They are widely used in ML and DM
- Mean Absolute Error (MAE): $$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$
- Mean Squared Error (MSE): $$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$
- Root Mean Squared Error (RMSE):

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

Multiple regression

- Simple regression: 1 feature
- Multiple regression: more than 1 feature



- The line becomes a plane in 2-dim. space and a hyperplane in >2-dim. space
- R^2 is similarly defined, called multiple coefficient of determination

Question time

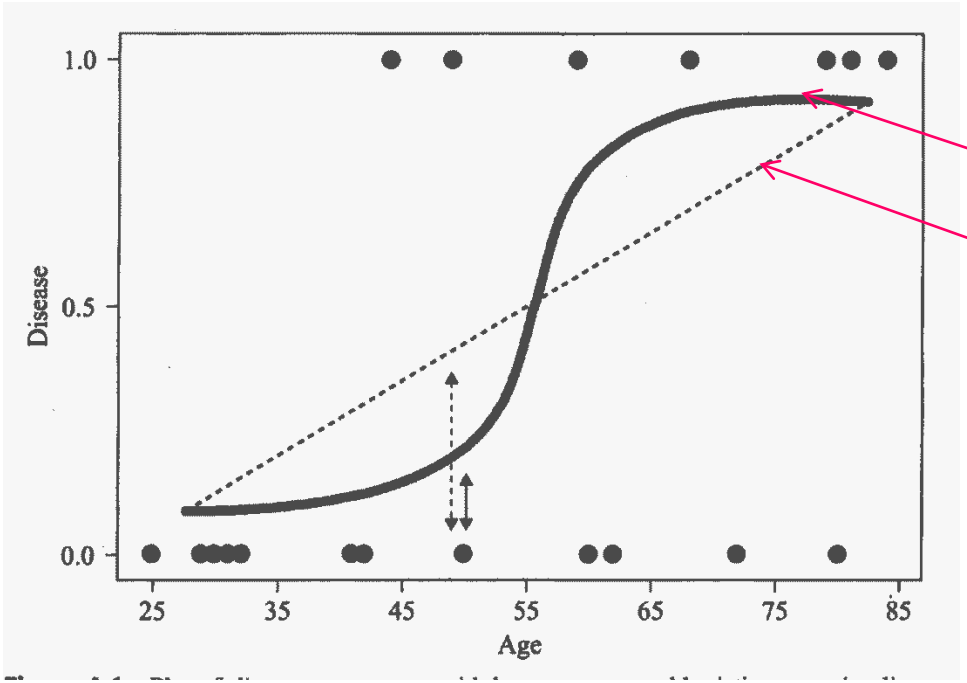
- True or False?
- 1) The regression line minimizes the sum of the residuals
- 2) If all residuals are 0, SST=SSR
- 3) If the value of the correlation coefficient is negative, this indicates that the 2 variables are negatively correlated
- 4) The value of the correlation coefficient can be calculated given the value of R^2
- 5) SSR represents an overall measure of the prediction error on the training set by using the regression line

Answers

Logistic Regression

Logistic regression

- Used for classification tasks
- Two classes: 0 and 1 (there are extensions for more than 2 classes)
- Fits the data to a logistic (sigmoidal) curve instead of fitting it to a straight line
  - => assumes that the relationship between the feature and class variable is *nonlinear*



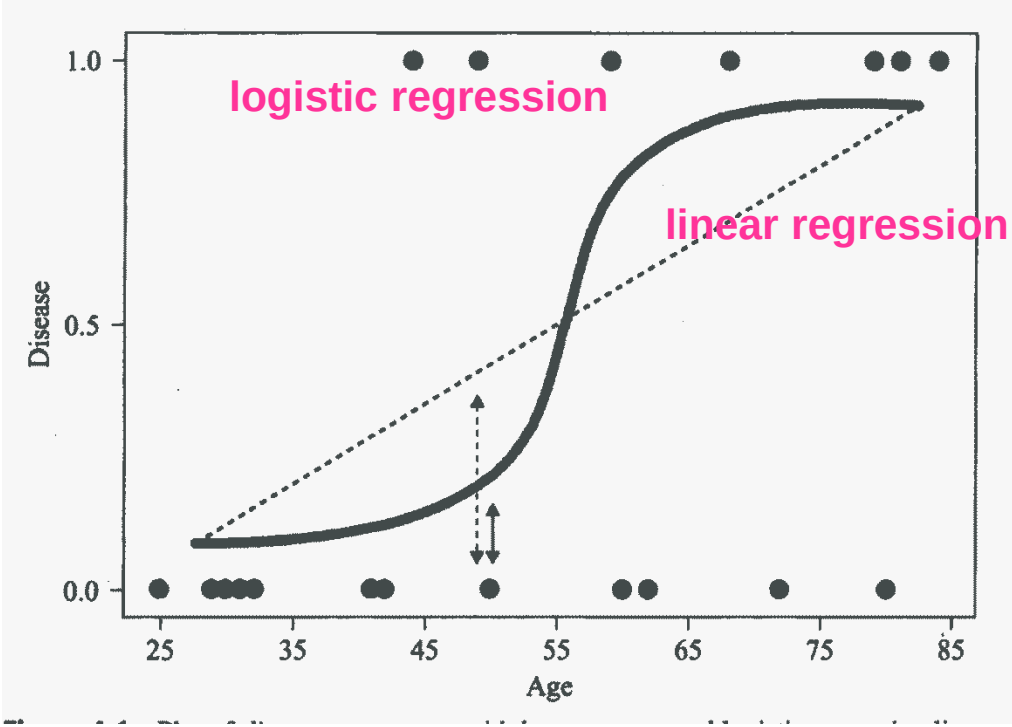
logistic regression

linear regression

Simple (bivariate) logistic regression

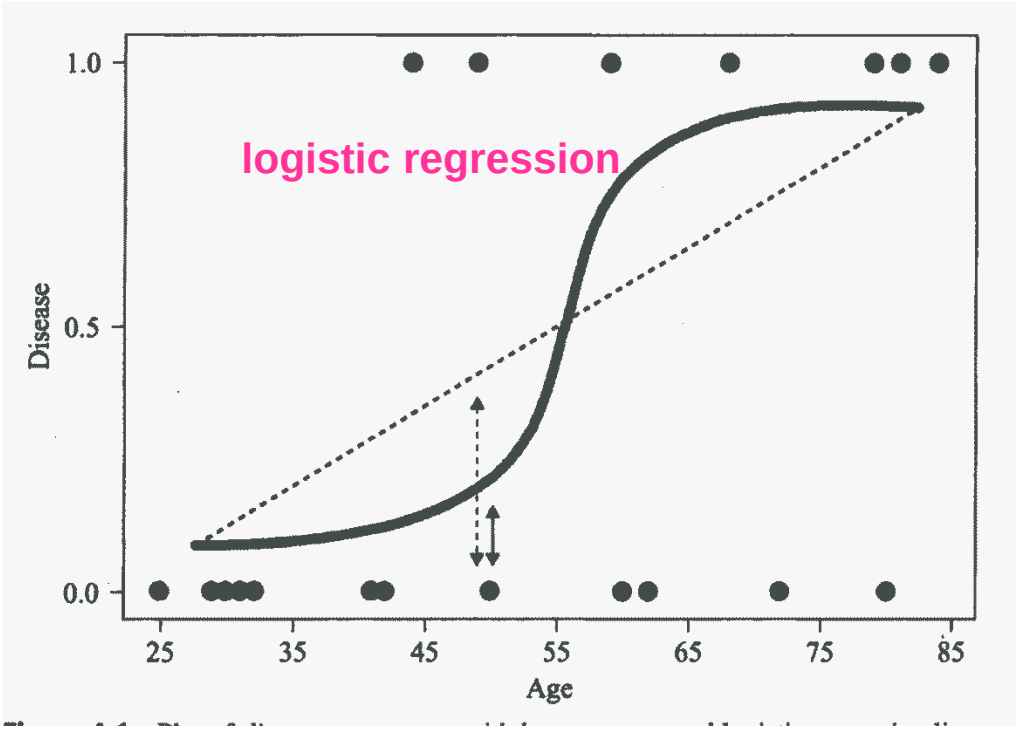
- Example: Predicting the presence (class=1) or absence (class=0) of a particular disease, given the patient's age

ID	age	disease	ID	age	disease
1	25	0	11	50	0
2	29	0	12	59	1
3	30	0	13	60	0
4	31	0	14	62	0
5	32	0	15	68	1
6	41	0	16	72	0
7	41	0	17	79	1
8	42	0	18	80	0
9	44	1	19	81	1
10	49	1	20	84	1



Logistic regression – example

- What will be the prediction of Logistic Regression for patient 11 from the training data (age=50, disease=0)?



Logistic curve

- The logistic curve gives a value between 0 and 1 that is interpreted as the probability for class membership:

$$p = \frac{e^{b_0 + b_1 x}}{1 + e^{b_0 + b_1 x}}$$

- p is the probability for class 1 and 1-p is the probability for class 0

- It uses the *maximum likelihood method* to find the parameters b_0 and b_1 - the curve that best fits the data

How to make predictions

- The logistic regression produced $b_0 = -4.372$, $b_1 = 0.06696$
- => the probability for a patient aged 50 (training example 11) to have the disease:

$$p = \frac{e^{b_0 + b_1 x}}{1 + e^{b_0 + b_1 x}} = \frac{e^{-4.372 + 0.06696 \ .age}}{1 + e^{-4.372 + 0.06696 \ .age}} = 0.26$$

- => 26% to have the disease and 74% not to have the disease
- We can use the probability directly or convert it into 0/1 answer required for classification tasks: 0 if p<0.5 and 1 if p>=0.5
- => We predict class 0 for this patient
- The class for new examples can be predicted similarly – e.g. make a prediction for a patient aged 45

Logistic regression equation

$$p = \frac{e^{b_0 + b_1 x}}{1 + e^{b_0 + b_1 x}}$$

- It also follows that:

$$b_0 + b_1 x = \ln \frac{p}{1 - p}$$

$$\ln \frac{p}{1 - p} = b_0 + b_1 x$$

linear calculation, as in linear regression

called odds ratio for the default class (class 1)

$$\ln(odds) = b_0 + b_1 x \quad \Rightarrow \quad odds = e^{(b_0 + b_1 x_1)}$$

Compare:

- Logistic regression: $\ln(odds) = b_0 + b_1 x$
- Linear regression: $\hat{y} = b_0 - b_1 x$

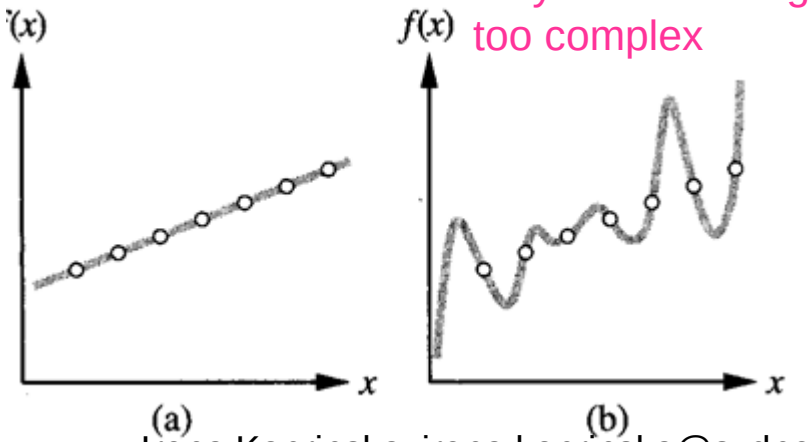
The model is still a linear combination of the input features, but this combination determines the log odds of the class not directly the predicted value

Overfitting and Regularization

Overfitting

- Overfitting:
  - Small error on the training set but high error on test set (new examples)
  - The classifier has memorized the training examples but has not learned to generalize to new examples!
- It occurs when
  - we fit a model too closely to the particularities of the training set – the resulting model is too specific, works well on the training data but doesn't work well on new data

Ex.1



may be overfitting – too complex

Ex.2

Rule1: may be overfitting – too specific
If age>45, income>100K, has_children=3, divorced=no -> buy_boat=yes

Rule2:
If age>45, income>100K -> buy_boat=yes

Overfitting (2)

- Various reasons, e.g.
  - Issues with the data
    - Noise in the training data
    - Too small training set – does not contain enough representative examples
  - How the algorithm operates
    - Some algorithms are more susceptible to overfitting than others
    - Different algorithms have different strategies to deal with overfitting, e.g.
      - Decision tree – prune the tree
      - Neural networks – early stopping of the training
      - …

Underfitting

- The model is too simple and doesn't capture all important aspects of the data
  - It performs badly on both training and test data

Rule1: may be overfitting – too specific
If age>45, income>100K, has_children=3,
divorced=no -> buy_boat=yes

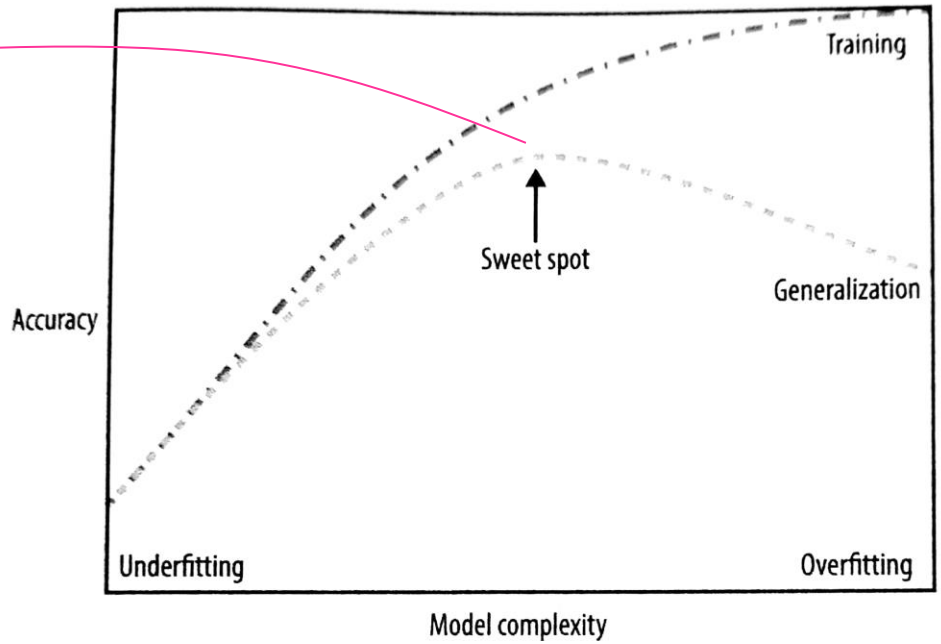
Rule2:
If age>45, income>100K -> buy_boat=yes

Rule3: may be underfitting – too general
If owns_hourse=yes -> buy_boat=yes

Trade-off between model complexity and generalization performance

- generalization performance = accuracy on test set
- Usually, the more complex we allow the model to be, the better it will predict on the training data
- However, if it becomes to complex, it will start focusing too much on each individual data point, and will not generalize well on new data

Image from A. Mueller and S. Guido, Introduction to ML with Python



- There is point in between, which will yield the best test accuracy
- This is the model we want to find

Regularization

- Regularization means explicitly restricting a model to avoid overfitting
- It is used in some regression models (e.g. Ridge and Lasso regression) and in some neural networks

Ridge and Lasso Regression

Ridge regression

- A regularized version of the standard Linear Regression (LR)
- Also called Tikhonov regularization
- Uses the same equation as LR to make predictions
- However, the coefficients w are chosen so that they not only fit well the training data (as in LR) but also satisfy an additional constraint:
  - the magnitude of the coefficients is as small as possible, i.e. close to 0
- Small values of the coefficients means
  - each feature will have little effect on the outcome
  - small slope of the regression line
- Rationale: a more restricted model (less complex) is less likely to overfit
- Ridge regression uses the so called L2 regularization (L2 norm of the weight vector)

Ridge regression (2)

- Minimizes the following cost function:

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \alpha \sum_{i=1}^{n} {w_i}^2$$

MSE

regularization term

Goal: high accuracy on training data (low MSE)

low complexity model – w close to 0

- Parameter α controls the trade-off between the performance on training set and model complexity

Ridge regression (3)

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \alpha \sum_{i=1}^{n} w_i^2$$

MSE

regularization term (L2 norm)

- α controls the trade-off between the performance on the training set and model complexity
  - Increasing α makes the coefficients smaller (close to 0); this typically decreases the performance on the training set but may improve the performance on the test set
  - Decreasing α means less restricted coefficients. For very small α , Ridge Regression will behave similarly to LR

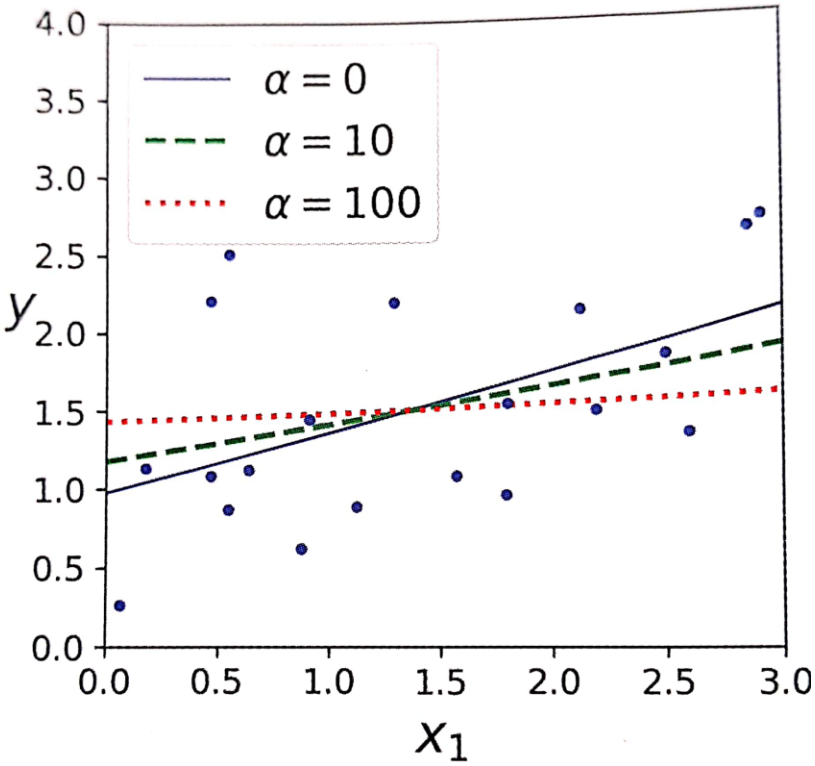


Image from A. Geron, Hands-on ML with Scikit-learn, Keras & TensorFlow

Lasso regression

- Another regularized version of the standard Linear Regression (LR)
- LASSO = Least Absolute Shrinkage and Selection Operator Regression
- As Ridge Regression, it adds a regularization term to the cost function but it uses the L1 norm of the weight vector w

$$\underbrace{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}_{\text{MSE}} + \underbrace{\alpha \sum_{i=1}^{n} ||w_i||}_{\text{regularization term (L1 norm)}}$$

Goal: high accuracy on training data (low MSE)

low complexity model

- Consequence of using L1 – some w will become exactly 0
- => some features will be completely ignored by the model – a form of automatic feature selection
- Less features – simpler model, easier to interpret

Lasso regression (2)

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \alpha \sum_{i=1}^{n} ||w_i||$$

MSE

regularization term (L1 norm)

- As in Ridge Regression:
  - α controls the trade-off between the performance on the training set and model complexity
  - Increasing/decreasing α - similar reasoning as before

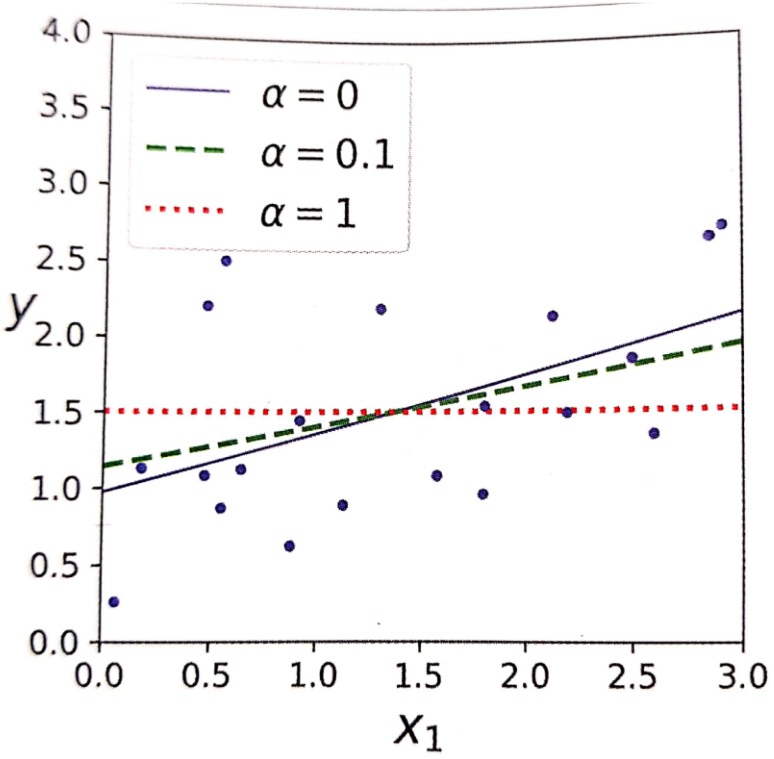


Image from A. Geron, Hands-on ML with Scikit-learn, Keras & TensorFlow

Summary

- Linear regression
  - Simple (bivariate) - a line is used to approximate the relationship between 2 continuous variables (feature x and class variable y)
  - Multiple – more than 1 feature; the line becomes a hyperplane
  - The least-square method is used to find the line (hyperplane) which best fit the given data (training data)
    - "Best fit": minimizes the sum of the squared errors (SSE) between the actual and predicted values of y, over all data points
  - R^2 = coefficient of determination=SSR/SST – how well the line fits the data [0,1]; the higher the better
  - MAE, MSE and RMSE –widely used accuracy measures in ML (can be measured on both training and test data)

Summary (2)

- Logistic regression
  - Simple (bivariate) - a sigmoidal curve is used to approximate the relationship between the feature x and class variable y
  - => assumes the relationship between the feature and class variable is nonlinear
  - Multiple – more than 1 feature; the sigmoidal curve becomes a sigmoidal hyperplane
  - Uses the maximum likelihood method to find the curve (hyperplane) which best fit the given data (training data)
- Overfitting and regularization
  - Overfitting - high accuracy on training data but low on test data (low generalization)
  - High model complexity -> low generalization
  - Regularization is a method to avoid overfitting – it makes the model more restrictive (less complex)
  - Ridge and Lasso regression are regularized linear regression models

Acknowledgements

- M. Lewis-Beck, Applied statistics, SAGE University Paper Series on Quantitative Analysis.
- D. Larose, Data Mining: Methods and Models, 2006, Wiley.