Exercise Collection – ML Basics

Contents

Lecture exercises

Exercise 1: ML Tasks

Identify which type of machine learning (supervised or unsupervised, type of of task, learning to predict or to explain) could be used in these cases:

a) When crossing the alps using the Brenner Autobahn, there is the option to pay electronically in advance. When approaching the toll station, the barrier automatically opens when the number plate was recognised. The recognition happens automatically by a digital camera system.

b) Diagnose whether a patient suffers from cancer or not.

c) The owner of an internet site wants to protect her system against various violations of the terms of service (bot programs, manipulation of timestamps, etc.)

d) An online shopping portal wants to determine products that are automatically offered to registered customers upon login.

e) We want to sort the contents in our news feed into different groups.

f) We want to sort our e-mails into spam / non-spam.

g) In a supermarket, products that are often bought together shall be placed side by side to increase the sales.

h) We want to determine our top customers (e.g., w.r.t. highest sales, logistics etc.).

i) A call center estimates the amount of customer traffic to facilitate staff planning.

Solution 1:

a) supervised learning / multi-class classification (plate digits) / learning to predict

b) supervised / binary classification / learning to predict, perhaps also learning to explain

c) (un)supervised / outlier detection / learning to predict

d) unsupervised / frequent pattern mining

e) (un)supervised / classification or clustering / learning to predict

f) supervised / binary classification / learning to predict

g) unsupervised / clustering or association rules

h) not a machine learning task

i) supervised / regression / learning to predict, perhaps also learning to explain

Exercise 2: Simple Regression Problem I

Suppose we observe 6 data pairs and want to describe the underlying relationship between target y and feature $\mathbf{x}$.

x	0.56	0.22	1.7	0.63	0.36	1.2
y	160	150	175	185	165	170

a) Assume a standard linear relationship

$$y^{(i)} = \beta_0 + \beta_1 \mathbf{x}^{(i)} + \epsilon^{(i)}$$

with iid errors $\epsilon^{(i)}$ and calculate the least squares estimator $\hat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} = (\beta_0, \beta_1)^\top$ manually (+ calculator).

b) Assume a non-linear relationship (polynomial degree 2)

$$y^{(i)} = \beta_0 + \beta_1 \mathbf{x}^{(i)} + \beta_2 (\mathbf{x}^{(i)})^2 + \epsilon^{(i)}$$

with iid errors $\epsilon^{(i)}$ and calculate the least squares estimator $\hat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)^\top$ with R.

Solution 2:

a) We use the least squares-estimator introduced in the lecture: $\hat{\beta} = (X^T X)^{-1} X^T y$ with

$$X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,m} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,m} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \dots & x_{n,m} \end{bmatrix}$$

$$x = \begin{bmatrix} 0.56 \\ 0.22 \\ 1.7 \\ 0.63 \\ 0.36 \\ 1.2 \end{bmatrix}, X = \begin{bmatrix} 1 & 0.56 \\ 1 & 0.22 \\ 1 & 1.7 \\ 1 & 0.63 \\ 1 & 0.36 \\ 1 & 1.2 \end{bmatrix} \text{ and } y = \begin{bmatrix} 160 \\ 150 \\ 175 \\ 185 \\ 165 \\ 170 \end{bmatrix}$$

Then

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

$$= \left(\begin{bmatrix} 1 & 1 & 1 & ... & 1 \\ x_{1,1} & x_{2,1} & x_{3,1} & ... & x_{n,1} \\ \vdots & \vdots & \vdots & ... & \vdots \\ x_{1,m} & x_{2,m} & x_{3,m} & ... & x_{n,m} \end{bmatrix} \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & ... & x_{1,m} \\ 1 & x_{2,1} & x_{2,2} & ... & x_{2,m} \\ \vdots & \vdots & \vdots & ... & \vdots \\ 1 & x_{n,1} & x_{n,2} & ... & x_{n,m} \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & 1 & ...1 & \\ x_{1,1} & x_{2,1} & x_{3,1} & ... & x_{n,1} \\ \vdots & \vdots & \vdots & ... & \vdots \\ x_{1,m} & x_{2,m} & x_{3,m} & ... & x_{n,m} \end{bmatrix} \begin{bmatrix} 160 \\ 150 \\ 175 \\ 185 \\ 165 \\ 170 \end{bmatrix}$$

$$= \left(\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.56 & 0.22 & 1.7 & 0.63 & 0.36 & 1.2 \end{bmatrix} \begin{bmatrix} 1 & 0.56 \\ 1 & 0.22 \\ 1 & 1.7 \\ 1 & 0.63 \\ 1 & 0.36 \\ 1 & 1.2 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.56 & 0.22 & 1.7 & 0.63 & 0.36 & 1.2 \end{bmatrix} \begin{bmatrix} 160 \\ 150 \\ 175 \\ 185 \\ 165 \\ 170 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 4.67 \\ 4.67 & 5.2185 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.56 & 0.22 & 1.7 & 0.63 & 0.36 & 1.2 \end{bmatrix} \begin{bmatrix} 160 \\ 150 \\ 175 \\ 185 \\ 165 \\ 170 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5491944 & -0.4914703 \\ -0.4914703 & 0.6314394 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.56 & 0.22 & 1.7 & 0.63 & 0.36 & 1.2 \end{bmatrix} \begin{bmatrix} 160 \\ 150 \\ 175 \\ 185 \\ 165 \\ 170 \end{bmatrix}$$

$$= \begin{bmatrix} 0.2739710 & 0.4410709 & -0.2863051 & 0.23956809 & 0.3722651 & -0.04056998 \\ -0.1378643 & -0.3525536 & 0.5819766 & -0.09366351 & -0.2641521 & 0.26625693 \end{bmatrix} \begin{bmatrix} 160 \\ 150 \\ 175 \\ 185 \\ 165 \\ 170 \end{bmatrix}$$

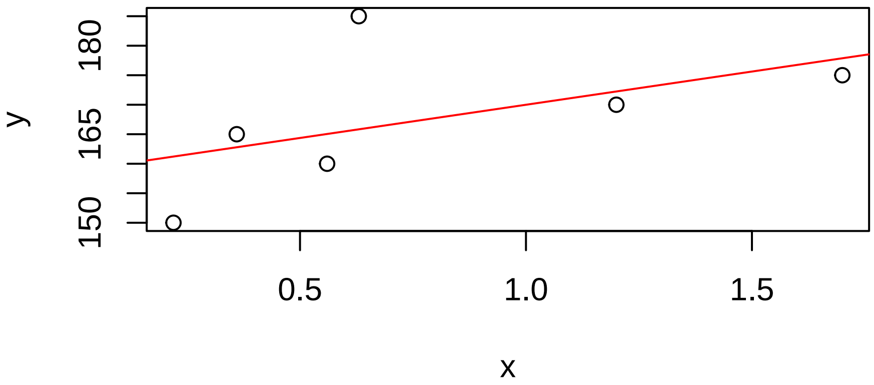
$$= \begin{bmatrix} 158.73954 \\ 11.25541 \end{bmatrix}$$

Hence the linear model $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x = 158.73954 + 11.25541x$

```
x = c(0.56, 0.22, 1.7, 0.63, 0.36,1.2)
y = c(160,150,175,185,165,170)

X <- sapply(0:1, function(k) x^k)
solve(t(X) %*% X) %*% t(X) %*% y

##           [,1]
## [1,] 158.73954
## [2,]  11.25541
```

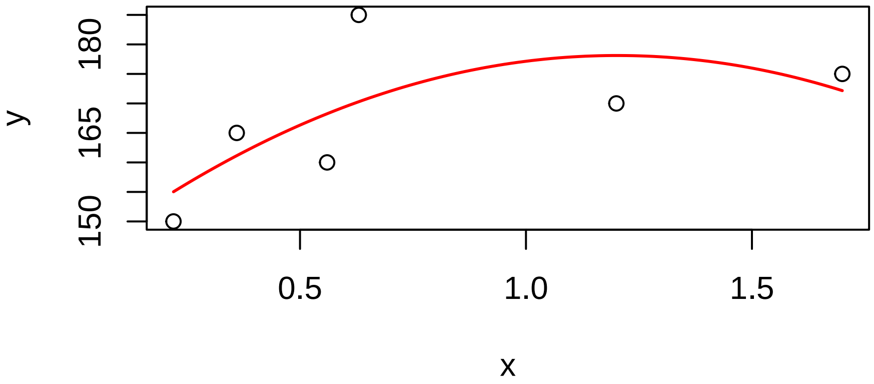


b) Here $X = \begin{bmatrix} 1 & 0.56 & 0.3136 \\ 1 & 0.22 & 0.0484 \\ 1 & 1.7 & 2.89 \\ 1 & 0.63 & 0.3969 \\ 1 & 0.36 & 0.1296 \\ 1 & 1.2 & 1.44 \end{bmatrix}$ and $\hat{\beta} = \begin{bmatrix} 143.51682 \\ 57.59155 \\ -23.96347 \end{bmatrix}$

```
x = c(0.56, 0.22, 1.7, 0.63, 0.36,1.2)
y = c(160,150,175,185,165,170)

X <- sapply(0:2, function(k) x^k)
solve(t(X) %*% X) %*% t(X) %*% y

##            [,1]
## [1,] 143.51681
## [2,]  57.59155
## [3,] -23.96347
```



Exercise 3: Simple Regression Problem II

Suppose we observe 6 data pairs and want to describe the underlying relationship between target y and feature $\mathbf{x}$.

x	0.56	0.22	1.7	0.63	0.36	1.2
y	160	150	175	185	165	170

a) For the linear model

$$f\left(\mathbf{x}^{(i)}\right) = \theta_0 + \theta_1 \mathbf{x}^{(i)}$$

with L2 loss, starting from $\boldsymbol{\theta}^{[0]} = (0, 0)$ calculate one step of gradient descent with a stepsize of $\alpha = 0.1$.

b) Implement a function `grad_desc(x, y, iterations, alpha = 0.1)` that computes `iterations` steps of gradient descent with a learning rate of `alpha` for a linear regression with L2 loss. You can initialize all model parameters at 0.

c) How do the parameters estimated by gradient descent differ from the parameters estimated by the analytical solution (see ex. sheet 1) for different values of `iterations`.

Solution 3:

a) From the lecture we can retrieve the empirical risk function:

$$R_{emp}(\theta) = \frac{1}{n}(X\theta - Y)^2$$

A gradient descend step is given by:

$$\theta^{[t+1]} = \theta^{[t]} - \alpha \frac{\partial}{\partial \theta} R_{emp}(\theta^{[t]})$$

Therefore we need the derivative of R_{emp}:

$$\frac{\partial}{\partial \theta} R_{emp}(\theta^{[t]}) = \frac{2}{n} X^T [X\theta^{[t]} - y]$$

$$\frac{\partial}{\partial \theta} R_{emp}(\theta^{[0]}) = \frac{2}{6} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.56 & 0.22 & 1.7 & 0.63 & 0.36 & 1.2 \end{bmatrix} \left(\begin{bmatrix} 1 & 0.56 \\ 1 & 0.22 \\ 1 & 1.7 \\ 1 & 0.63 \\ 1 & 0.36 \\ 1 & 1.2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 160 \\ 150 \\ 175 \\ 185 \\ 165 \\ 170 \end{bmatrix} \right)$$

$$= -\frac{1}{3} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.56 & 0.22 & 1.7 & 0.63 & 0.36 & 1.2 \end{bmatrix} \begin{bmatrix} 160 \\ 150 \\ 175 \\ 185 \\ 165 \\ 170 \end{bmatrix}$$

$$= \begin{bmatrix} -335 \\ -266.683 \end{bmatrix}$$

$$\theta^{[1]} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - 0.1 \begin{bmatrix} -335 \\ -266.683 \end{bmatrix} = \begin{bmatrix} 33.5 \\ 26.6683 \end{bmatrix}$$

We can easily verify the computation in R:

```r
r_emp_derivative = function(theta_t, X, y) {
  2 / nrow(X) * (t(X) %*% (X %*% theta_t - y))
}
```

```r
grad_desc_step = function(theta_t, X, y, alpha) {
  theta_t - alpha * r_emp_derivative(theta_t, X, y)
}
```

Note that X is a matrix with $X = [1 \; x]$, as

$$f(X) = \theta X = \theta_0 1_n + \theta_1 x$$

```r
small_x = c(0.56, 0.22, 1.7, 0.63, 0.36, 1.2)
X = cbind(1, small_x)
y = c(160, 150, 175, 185, 165, 170)

grad_desc_step(c(0, 0), X, y, 0.1)

##              [,1]
##          33.50000
## small_x 26.66833
```

b)

```r
grad_desc = function(small_x, y, iterations, alpha = 0.1) {
  X = cbind(1, small_x)
  theta_t = c(0, 0)
  # Repeat for n steps
  for (i in 1:iterations) {
    theta_t = grad_desc_step(theta_t, X, y, alpha)
  }
  return(theta_t)
}
```

c)

```r
iterations = c(1, 25, 50, 100, 250, 500, 750, 1000)
theta_hat = sapply(iterations, function(z) grad_desc(small_x, y, z))
predictions = cbind(iterations, t(theta_hat))
colnames(predictions) = c("iterations", "theta_0", "theta_1")

predictions

##      iterations  theta_0  theta_1
## [1,]          1  33.5000 26.66833
## [2,]         25 127.9945 44.67538
## [3,]         50 144.6637 26.55818
## [4,]        100 155.7887 14.46344
## [5,]        250 158.7124 11.28496
## [6,]        500 158.7395 11.25542
## [7,]        750 158.7395 11.25541
## [8,]       1000 158.7395 11.25541
```

Looking at exercise 2 of the previous sheet, one can see, that $\hat{\theta_0} = 158.73954$ and $\hat{\theta_1} = 11.25541$. We know, that this is a local minimum. The values from the gradient descent only slowly approaches the local minimum. As we can see, the predicted $\hat{\theta}_{\text{grad\_desc}}$ of the `grad_desc` function only coincides with $\hat{\theta}_{\text{analytical}}$ for a large number of iterations. Since the starting value $\theta^{[0]} = (0, 0)$ is not close to the minimum, it takes some steps to get a good approximation.

Exercise 4: Car Price Prediction

Imagine you work at a second-hand car dealer and are tasked with finding for-sale vehicles your company can acquire at a reasonable price. You decide to address this challenge in a data-driven manner and develop a model that predicts adequate market prices (in EUR) from vehicles' properties.

a) Characterize the task at hand: supervised or unsupervised? Regression or classification? Learning to explain or learning to predict? Justify your answers. **[only for lecture group B]**

b) How would you set up your data? Name potential features along with their respective data type and state the target variable.

c) Assume now that you have data on vehicles' age (days), mileage (km), and price (EUR). Explicitly define the feature space $\mathcal{X}$ and target space $\mathcal{Y}$.

d) You choose to use a linear model (LM) for this task. The LM models the target as a linear function of the features with Gaussian error term.

State the hypothesis space for the corresponding model class. For this, assume the parameter vector $\boldsymbol{\theta}$ to include the intercept coefficient.

e) Which parameters need to be learned? Define the corresponding parameter space Θ.

f) State the loss function for the i-th observation using $L2$ loss.

g) Now you need to optimize this risk to find the best parameters, and hence the best model, via empirical risk minimization. State the optimization problem formally and list the necessary steps to solve it.

Congratulations, you just designed your first machine learning project!

Solution 4:

a) We face a **supervised regression** task: we definitely need labeled training data to infer a relationship between cars' attributes and their prices, and price in EUR is a continuous target (or quasi-continuous, to be exact – as with all other quantities, we can only measure it with finite precision, but the scale is sufficiently fine-grained to assume continuity). **Prediction** is definitely the goal here, however, it might also be interesting to examine the explanatory contribution of each feature.

b) Target variable and potential features:

Variable	Role	Data type
Price in EUR	Target	Numeric
Age in days	Feature	Numeric
Mileage in km	Feature	Numeric
Brand	Feature	Categorical
Accident-free y/n	Feature	Binary
...	...	...

c) Let x_1 and x_2 measure age and mileage, respectively. Both features and target are numeric and (quasi-) continuous. It is also reasonable to assume non-negativity for the features, such that we obtain $\mathcal{X} = (\mathbb{R}_0^+)^2$, with $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)})^\top \in \mathcal{X}$ for $i = 1, 2, \dots, n$ observations. As the standard LM does not impose any restrictions on the target, we have $\mathcal{Y} = \mathbb{R}$, though we would probably discard negative predictions in practice.

d) We can write the hypothesis space as:

$$\mathcal{H} = \{f(\mathbf{x} \mid \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x} \mid \boldsymbol{\theta} \in \mathbb{R}^3\} = \{f(\mathbf{x} \mid \boldsymbol{\theta}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \mid (\theta_0, \theta_1, \theta_2) \in \mathbb{R}^3\}.$$

Note the **slight abuse of notation** here: in the lecture, we first define $\boldsymbol{\theta}$ to only consist of the feature coefficients, with $\mathbf{x}$ likewise being the plain feature vector. For the sake of simplicity, however, it is more convenient to append the intercept coefficient to the vector of feature coefficients. This does not change our model formulation, but we have to keep in mind that it implicitly entails adding an element 1 at the first position of each feature vector.

e) The parameter space is included in the definition of the hypothesis space and in this case given by $\Theta = \mathbb{R}^3$.

f) Loss function for the i-th observation: $L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) = \left(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x}^{(i)}\right)^2$.

g) In order to find the optimal $\hat{\boldsymbol{\theta}}$, we need to solve the following minimization problem:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}\in\Theta} \left(\sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x}^{(i)}\right)^2\right)$$

This is achieved in the usual manner of setting the derivative w.r.t. $\boldsymbol{\theta}$ to 0 and solving for $\boldsymbol{\theta}$, yielding the familiar least-squares estimator.

Exercise 5: Credit Scoring Project

Imagine you work at a bank and have the job to develop a credit scoring model. This means, your model should predict whether a customer applying for a credit will be able to pay it back in the end.

a) Is this a supervised or unsupervised learning problem? Justify your answer.

b) How would you set up your data? Which is the target variable, what feature variables could you think of? Do you need labeled or unlabeled data? Justify all answers.

c) Is this a regression or classification task? Justify your answer.

d) Is this "learning to predict" or "learning to explain"? Justify your answer.

e) In classical statistics, you could use e.g. the logit model for this task. This means we assume that the targets are conditionally independent given the features, so $y^{(i)}|\mathbf{x}^{(i)}\mathcal{I}y^{(j)}|\mathbf{x}^{(j)}$ for all $i, j = 1, \ldots, n, i \neq j$, where n is the sample size. We further assume that $y^{(i)}|\mathbf{x}^{(i)} \sim Bin(\pi^{(i)})$, where $\pi^{(i)} = \frac{\exp(\boldsymbol{\theta}^\top \mathbf{x}^{(i)})}{1+\exp(\boldsymbol{\theta}^\top \mathbf{x}^{(i)})}$. Looking at this from a Machine Learning perspective, write down the hypothesis space for this model. State explicitly which parameters have to be learned.

f) In classical statistics, you would estimate the parameters via Maximum Likelihood estimation. (The log-Likelihood of the Logit-Model is: $\sum\limits_{i=1}^{n} y^{(i)} \log(\pi^{(i)}) + (1 - y^{(i)})(\log(1 - \pi^{(i)})))$. How could you use the model assumptions to define a reasonable loss function? Write it down explicitly.

g) Now you have to optimize this risk function to find the best parameters and hence the best model. Describe with a few sentences, how you could do this.

Congratulations, you just designed your first Machine Learning project!

Solution 5:

a) Supervised learning problem - the model will be learned from historical credit data for which payment history has been observed (knowing the ground truth is vital here since we need to evaluate our model's accuracy)

b) Target variable: classes (default y/n), continuous credit scores, or class probabilities). Potential features: monthly income, current level of indebtedness, past credit behavior, profession, residential environment, age, number of kids etc. Labels: yes, since we have a supervised learning problem.

c) This is a classification problem - we want to assign our customers to classes *default* and *non-default*.

d) (Primarily) learning to predict - we want to score future borrowers.

e) $\mathcal{H} = \{\pi : \mathcal{X} \mapsto [0,1] \mid \pi(\mathbf{x} \mid \boldsymbol{\theta}) = s(\boldsymbol{\theta}^T\mathbf{x}), \boldsymbol{\theta} \in \mathbb{R}^d\}$, where $s(z) = 1/(1+exp(-z))$ is the sigmoid function. Parameters to be learned: $\boldsymbol{\theta}$.

f) We know that, in the optimum, (log-)likelihood is maximal. We can directly translate this into risk minimization by using the *negative* log-likelihood as our empirical risk. We will just use the pointwise negative log-likelihood as our loss function:

$$L\left(y^{(i)}, \pi\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) = -\left(y^{(i)} \log\left(\pi\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) + \left(1 - y^{(i)}\right)\left(\log\left(1 - \pi\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right)\right)\right)$$

(the so-called *Bernoulli loss*). The empirical risk is then the sum of point-wise losses:

$$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = -\sum_{i=1}^{n} y^{(i)} \log\left(\pi\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) + \left(1 - y^{(i)}\right)\left(\log\left(1 - \pi\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right)\right)$$

g) We can now solve this optimization problem via empirical risk minimization, which, in this case, is perfectly equivalent to ML estimation. Therefore, we set the first derivative of $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ wrt $\boldsymbol{\theta}$ to 0 and solve for $\boldsymbol{\theta}$. However – unlike linear regression – this has no closed-form solution, so a numerical optimization procedure such as gradient descent is required.

Exercise 6: Own Use Case

Think of a practical use case of your daily work where you would like to apply machine learning methods.

a) Describe your use case in a way that allows non-experts of your field to understand the main idea.

b) Is this a supervised or unsupervised learning problem? Justify your answer. (Try to think of a supervised problem to answer the next questions.)

c) How would you set up your data? Which is the target variable, what feature variables could you think of? Do you need labeled or unlabeled data? Justify all answers.

d) Is this a regression or classification task? Justify your answer.

e) Is this 'learning to predict' or 'learning to explain'? Justify your answer.

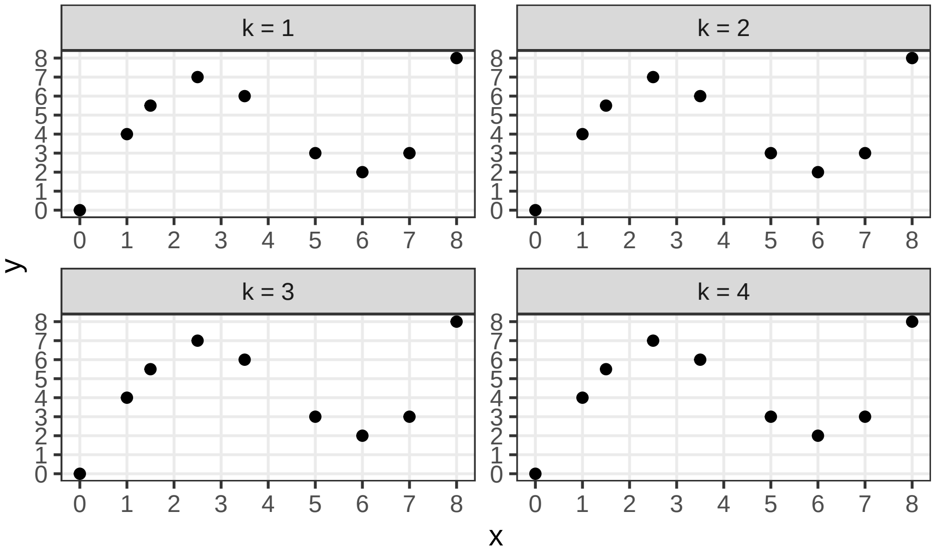
f) Which important properties should the ML algorithm fulfill for your use case? (E.g., accurate predictions, fast predictions, interpretability, scalability to large amounts of data, fast re-training with new data, ..)

Solution 6:
No model solution

Further exercises

Exercise 7: WS2020/21, second, question 1

ID	$\mathbf{x}$	y
1	0.0	0.0
2	1.0	4.0
3	1.5	5.5
4	2.5	7.0
5	3.5	6.0
6	5.0	3.0
7	6.0	2.0
8	7.0	3.0
9	8.0	8.0



Now we want to train a cubic polynomial, i.e., a polynomial regression model with degree $d = 3$ on the data used in a).

(i) Define the hypothesis space of this model and state explicitly how many parameters have to be estimated for training the model.

(ii) Define the minimization problem that we have to optimize in order to train the polynomial regression model. Use L2 loss and be as explicit as possible - without plugging in the data.

(iii) In order to estimate the parameters of the model, it is convenient to describe the model as a linear model. Compute the respective design matrix using the concrete values of $\mathbf{x}$ given above. Additionally, state a formula for estimating the parameters using this design matrix. (You do not have to derive this formula.)

Solution 7:

(i)

$$\mathcal{H} = \{f : f(\mathbf{x}) = \theta_0 + \theta_1\mathbf{x} + \theta_2\mathbf{x}^2 + \theta_3\mathbf{x}^3 \mid \theta_0, \theta_1, \theta_2, \theta_3 \in \mathbb{R}\}$$

The four parameters $\theta_0, \theta_1, \theta_2, \theta_3$ have to be estimated

(ii)

$$\hat{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$$

This means we have to optimize the following minimization problem wrt $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3)$:

$$\min_{\boldsymbol{\theta}\in\Theta} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}\in\Theta} \sum_{i=1}^{9} (y^{(i)} - (\theta_0 + \theta_1\mathbf{x}^{(i)} + \theta_2(\mathbf{x}^{(i)})^2 + \theta_3(\mathbf{x}^{(i)})^3))^2$$

(iii)

$$\hat{\theta} = (X^\top X)^{-1} X^\top y$$

X1	x	x.2	x.3
1	0.0	0.00	0.000
1	1.0	1.00	1.000
1	1.5	2.25	3.375
1	2.5	6.25	15.625
1	3.5	12.25	42.875
1	5.0	25.00	125.000
1	6.0	36.00	216.000
1	7.0	49.00	343.000
1	8.0	64.00	512.000

Exercise 8: WS2020/21, second, question 6

Describe a real-life application in which classification might be useful and where we want to “learn to explain”. Describe the response, as well as the predictors. Explain your answer thoroughly.

Solution 8:

No model solution

Ideas & exercises from other sources