Solution 1:

a) multiclass classification (plate digits) (supervised learning)

b) binary classification (supervised)

c) outlier detection ((un)supervised)

d) frequent pattern mining (unsupervised)

e) classification (supervised) / clustering (unsupervised)

f) classification (supervised)

g) clustering / assocation rules (unsupervised)

h) not a machine learning task

i) not a machine learning task

Solution 2:

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

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

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

Then

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

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

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

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

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

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

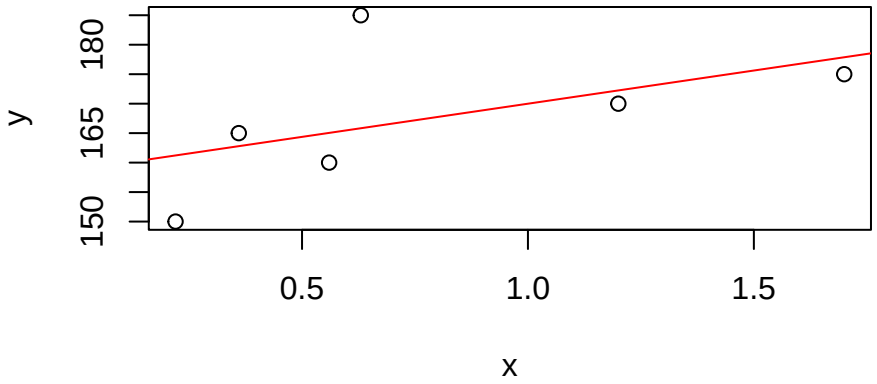
$$= \begin{bmatrix} 158.73954 \\ 11.25541 \end{bmatrix}$$

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
```

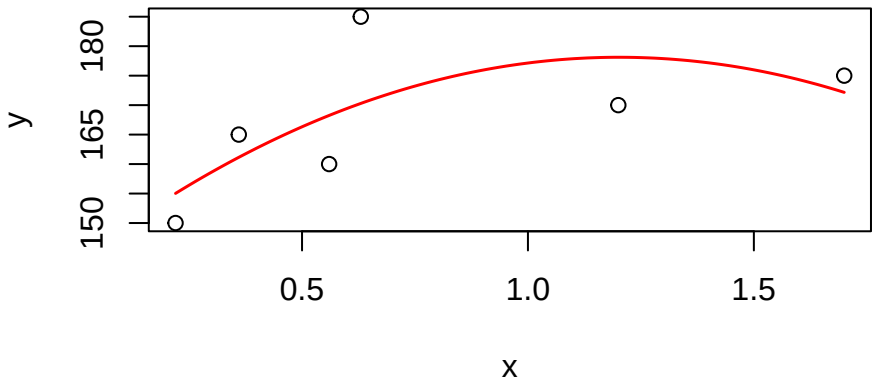


b) Here $X = \begin{bmatrix} 1 & 0.56 & 0.3136 \\ 1 & 0.22 & 0.0484 \\ 1 & 1.7 & 2.89 \\ 1 & 0.63 & 0.3969 \\ 1 & 0.36 & 0.1296 \\ 1 & 1.2 & 1.44 \end{bmatrix}$ and $\hat{\beta} = \begin{bmatrix} 143.51682 \\ 57.59155 \\ -23.96347 \end{bmatrix}$

```
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
```



Solution 3:

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.