IE30301 - Data Mining Assignment 2 (70 Points)

Prof. Junghye Lee

March 14, 2022

Exercise 1

1.1

Summarize the following concepts in 3 ~ 4 sentences each (write it in your own words). [5 pts]

1. **Supervised Learning**
2. **Unsupervised Learning**
3. **Regression**
4. **Classification**
5. **Clustering**

1.2

Summarize the characteristics and examples of the following variable types. [5 pts]

1. **Nominal**
2. **Binary**
3. **Continuous**
4. **Numeric**
5. **Ordinal**

1.3

What is the goal of learning in data mining? Please explain the following three keywords: **parameter, model, generalization**. The answer does not need to contain the exact terminology, but it should explain the key concepts. [3 pts]

Exercise 2

Given a set of independent and identically distributed data points $\mathbf{x} = \{x_1, \dots, x_N\}$ follows a distribution of $\mathcal{N}(\mu, \sigma^2)$. Derive the **Maximum Likelihood Estimation(MLE)** process that results in parameter estimate in Equation (2.1) and (2.2). Assume that both μ and σ are known. Please show the whole process of your derivation. [15 pts]

$$\hat{\mu}_{ML} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.1}$$

$$\hat{\sigma}^2_{ML} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu}_{ML})^2 \tag{2.2}$$

Exercise 3

3.1

In simple linear regression, When you derive $SST = SSR + SSE$, you will need to use the Equation (3.1). Please prove that Equation (3.1) holds. [5 pts]

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) = 0 \tag{3.1}$$

3.2

Derive the following equations, which are related to simple linear regression. [10 pts]

$$\hat{\beta}_0 \sim \mathcal{N}\left(\beta_0, \sigma^2 \frac{\sum x_i^2}{nS_{xx}}\right) \tag{3.2}$$

$$\hat{\beta}_1 \sim \mathcal{N}\left(\beta_1, \frac{\sigma^2}{S_{xx}}\right) \tag{3.3}$$

3.3

Equation (3.2) and (3.3) both follow a normal distribution. However, why do we have to perform a t-test to test the significance of parameters? [3 pts]

Exercise 4

The F-test for multiple linear regression is to test whether there exists any independent variables that are significant. The F-test statistics F^* can be calculated using sum of squares regression(SSR) and sum of squares error(SSE). The formula for both SSR and SSE is given below.

$$SSE = \frac{S_{xy}^2}{S_{xx}} \quad SSR = S_{yy} - \frac{S_{xy}^2}{S_{xx}} \tag{4.1}$$

4.1

Calculate F^* using SSR and SSE. [4 pts]

4.2

Construct a null and alternative hypothesis for the F-test in (4.1). Explain the intuition behind this equation. (Hint: Think about the difference between fitting a linear regression on the data v.s. simply using the mean(average) of the data) [4 pts]

4.3

Let us say $\hat{\beta}_1$ is the slope of the linear regression. Represent $\hat{\beta}_1$ in terms of S_{xy} and S_{xx} as you've learned from class. [4 pts]

4.4

One can also perform a t-test to test whether there exists a linear relationship between dependent and independent variables in linear regression. By using $H_0 : \beta_1 = 0$ and test statistics $t^* = \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)}$, prove that $(F^*)^{\frac{1}{2}} = t^*$ holds true. [4 pts]

Exercise 5

Given the following data $\mathbf{X} \in \mathbb{R}^{4\times 2}$ and label $\mathbf{y} \in \mathbb{R}^{4\times 1}$, solve the below questions. You may use a matrix multiplication calculator. **Explicitly state** all the intermediate results of the calculation you used to find your answer. Otherwise, no points will be given.

$$\mathbf{X} = \begin{bmatrix} 2 & 1 \\ -2 & -2 \\ 1 & 0 \\ 3 & 2 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

5.1

Fit a multiple linear regression with intercept. Find the parameters $\beta_0, \beta_1, \beta_2$. Write the answer in **fractional form** (Hint. You may need to manipulate **X** to find the intercept) [4 pts]

5.2

Compare your results using *Python*. Use the skeleton code given below. Paste your codes and screen capture the results of intercepts and coefficients for question **5.1**. [4 pts]

```
# Import necessary libraries. Only the following will be needed
import numpy as np
from sklearn import linear_model

X = np.array([[2,1],[-2,-2],[1,0],[3,2]])
y = np.array([0,1,2,3])

```