Assignment DM finished

Saturday, 8 March 2025 18:41

2.1

7. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

Obs.	X_1	X_2	X_3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	−1	0	1	Green
6	1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

(a) Compute the Euclidean distance between each observation and the test point, $X_1 = X_2 = X_3 = 0$.

(b) What is our prediction with $K = 1$? Why?

(c) What is our prediction with $K = 3$? Why?

a) Euclidean distance

$$d(X_i, X_j) = \sqrt{\sum_{i,j=1}^{K} (x_i - x_j)^2}$$

$d_{obs1} = \sqrt{(0-0)^2 + (3-0)^2 + (0-0)^2} = 3$

$d_{obs2} = \sqrt{4} = 2$

$d_{obs3} = \sqrt{1+9} \simeq 3.16$

$d_{obs4} = \sqrt{5} \simeq 224$

$d_{obs5} = \sqrt{2} \simeq 1.41$

$d_{obs6} = \sqrt{3} = 1.73$

OBS #	d	Y
5	1,41	GREEN } k=1
6	1,73	RED
2	2	RED } k=3
4	2,24	GREEN
1	3	RED
3	3,16	RED

b) With k=1 we consider the smallest observations, in this case obs_5 because the distance ≃ 1 the predictionwill be classified as 'Green'

c) For k=3, the smallest distance can be found in obs_1 and also obs_3. So the prediction will be 'Red'

2.2

3. Suppose we have a data set with five predictors, $X_1 =$ GPA, $X_2 =$ IQ, $X_3 =$ Level (1 for College and 0 for High School), $X_4 =$ Interaction between GPA and IQ, and $X_5 =$ Interaction between GPA and Level. The response is starting salary after graduation (in thousands of dollars). Suppose we use least squares to fit the model, and get $\hat\beta_0 = 50, \hat\beta_1 = 20, \hat\beta_2 = 0.07, \hat\beta_3 = 35, \hat\beta_4 = 0.01, \hat\beta_5 = -10$.

(a) Which answer is correct, and why?

i. For a fixed value of IQ and GPA, high school graduates earn more, on average, than college graduates.

ii. For a fixed value of IQ and GPA, college graduates earn more, on average, than high school graduates.

iii. For a fixed value of IQ and GPA, high school graduates earn more, on average, than college graduates provided that the GPA is high enough.

iv. For a fixed value of IQ and GPA, college graduates earn more, on average, than high school graduates provided that the GPA is high enough.

(b) Predict the salary of a college graduate with IQ of 110 and a GPA of 4.0.

(c) True or false: Since the coefficient for the GPA/IQ interaction term is very small, there is very little evidence of an interaction effect. Justify your answer.

X_1 = GPA ; X_2 = IQ, X_3 = Level $\left(\begin{smallmatrix}1 = \text{College}\\ 0 = \text{HS}\end{smallmatrix}\right)$; $X_4 = \overset{X_1 \cdot X_2}{\text{GPA} \times \text{IQ}}$; $X_5 = \overset{X_1 \cdot X_3}{\text{GPA} \times \text{LEVEL}}$

Y = starting salary after graduation

$y = B_0 + B_1X_1 + B_2X_2 + B_3X_3 + B_4X_4 + B_5X_5$
$= 50 + 20X_1 + 0.07X_2 + 35X_3 + 0.01X_4 - 10X_5$

a) As GPA is also present in X_5 inside an interaction term we could tell that if GPA is not high enough this coefficient term could penalize the whole expression.
So, (IV)

$Y_{Col} - Y_{HS} = (50 + 20X_1 + 0.07X_2 + 35 \cdot 1 + 0.01X_4 - 10X_4) - (50 + 20X_1 + 0.07X_2 + 35 \cdot 0 + 0.01X_4 - 10X_4)$ [$X_1 \cdot 1$; $= 0$]
$= (85 + 20X_1 + 0.07X_2 + 0.01X_4 - 10X_1) - (50 + 20X_1 + 0.07X_2 + 0.01X_4)$
$= (85 - 10X_1) - (50 + 20X_1)$
$= 35 - 10X_1$
$X_1 = 35/10 \simeq 3.5$ threshold

b) $y = 50 + (20 \cdot 4) + (0.07 \cdot 110) + (35 \cdot 1) + 0.01(4 \cdot 110) - 10 \cdot (4 \cdot 1)$
$\simeq 137.1$

c) False. To understand the interaction effect one needs to calculate if it is significative, it happens when p-value < 0.05

2.3

6. Suppose we collect data for a group of students in a statistics class with variables $X_1 =$ hours studied, $X_2 =$ undergrad GPA, and $Y =$ receive an A. We fit a logistic regression and produce estimated coefficient, $\hat\beta_0 = -6, \hat\beta_1 = 0.05, \hat\beta_2 = 1$.

(a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

(b) How many hours would the student in part (a) need to study to have a 50 % chance of getting an A in the class?

a) Logistic regression $\frac{e^{\beta}}{1 + e^{\beta}}$

$$P(Y=1) = \frac{e^{\hat\beta_0 + \hat\beta_1 X_1 + \hat\beta_2 X_2}}{1 + e^{\hat\beta_0 + \hat\beta_1 X_1 + \hat\beta_2 X_2}} = \frac{e^{-6 + 0.05 \cdot 40 + 1 \cdot 3.5}}{1 + e^{\cdots}} \simeq 37.7\,\%$$

b) X_1?

$P(Y) = 50\%$

$0.5 = \frac{1}{1 + e^{-z}}$

$0.5 \cdot (1 + e^{-z}) = 1$

$0.5 + 0.5e^{-z} = 1$

$0.5 - 0.5 + 0.5e^{-z} = 1 - 0.5$

$\frac{0.5e^{-z}}{0.5} = \frac{0.5}{0.5}$

$e^{-z} = 1$

$\ln(e^{-z}) = \ln(1)$

$-z = 0$

$z = 0$

So, since $z = \beta_0 + \beta_1 X_1 + \beta_2 X_2$

$0 = -6 + (0.05X_1) + (1 \cdot 3.5)$

$6 - 3.5 = 0.05X_1$

$X_1 = 2.5/0.05 = 50$ hours to study to get an A

2.4

2.4 A retailer wants for marketing purposes distinguish between costumers younger then 35 and customers older then 35. The following table summarizes the data set in the data base of the retailer in an abstract form. The relevant attributes, determined by domain knowledge, are for convenience denoted by *A* and *B*. The values for *A* are *a1*, *a2* and *a3*. The values for *B* are *b1* and *b2*. The retailer wants to use Data Mining

A	B	Number of Instances	
		Y	O
a1	b1	4	10
a2	b1	6	2
a3	b1	8	6
a1	b2	2	8
a2	b2	6	2

techniques to classify the costumers in the class "young", denoted by *Y*, and "old", denoted by *O*.

(a) Assume a new costumer enters the web-store and the retailer has no information at all about this customer. How will this new customer be classified based on the above data and explain why.
(b) Now assume a new customer comes in for which the retailer knows the values for attribute $A = a3$ and $B = b2$. Is it possible to apply the standard (non Naive) Bayes to classify this new customer? Explain what the problems is.
(c) Hence the retailer decides to use a naive Bayes classifier for the classification of this new customer.
- How will this customer now be classified based on the values of *A* and *B*? Explain your answer.

a) If a new customer enters the web-store and we don't have any information about him, we can use the prior probabilities to classify the customer:

Total number of istances = 54

$P(Y) = \frac{26}{54} \simeq 0.48$

$P(O) = \frac{28}{54} \simeq 0.52$

Since $P(O) > P(Y)$, the new customer will be classified as old "O".

b) $A = a_3$, $B = b_2$

The standard Bayes Theorem requires calculating the posterior probabilities of each class, given the known attributes.

- $P(Y \mid A=a_3, B=b_2) = \frac{P(A=a_3, B=b_2)\, P(Y)}{P(A=a_3, B=b_2 \mid Y)}$
- $P(O \mid A=a_3, B=b_2) = \frac{P(A=a_3, B=b_2)\, P(Y)}{P(A=a_3, B=b_2 \mid Y)}$

Considering the given dataset, there is no data for $(A=a_3, B=b_2)$, so we cannot apply standar Bayes Theorem because of zero-frequeny issue and no prior Knowleye

c) If we assume indipendence of A and B (Naive Bayes), we can estimate the probabilities separately.

- $P(Y \mid A=a_3, B=b_2) = \frac{P(A=a_3 \mid Y) \cdot P(B=b_2 \mid Y) \cdot P(Y)}{\sout{P(A=a_3, B=b_2)}} = \left(\frac{8}{26}\right) \cdot \left(\frac{2+6}{26}\right) \cdot \left(\frac{26}{54}\right) \simeq 0.31 \cdot 0.31 \cdot 0.48 \simeq 0.046$
- $P(O \mid A=a_3, B=b_2) = \frac{P(A=a_3 \mid O) \cdot P(B=b_2 \mid O) \cdot P(O)}{\sout{P(A=a_3, B=b_2)}} = \left(\frac{6}{28}\right) \cdot \left(\frac{8+2}{28}\right) \cdot \left(\frac{28}{54}\right) \simeq 0.21 \cdot 0.36 \cdot 0.52 \simeq 0.039$

!! Considering that the denominator is the same, we can just compare the numerators.

Since $P(Y \mid A=a_3, B=b_2) > P(O \mid A=a_3, B=b_2)$, the new customer will be classified as Young "Y".

2.5

2.5 Consider the same dataset as in the previous question. Now the retailer (data analyst) wants to use Decision Trees to classify new customers.

(a) What is the classification error rate for attribute *A*?
(b) What is the classification error rate for attribute *B*?
(c) What will be the splitting attribute in the top (root) of the Decision Tree if one uses the classification error rate?
(d) What is the Gini index for attribute *A*?
(e) What is the Gini index for attribute *B*?
(f) What will be the splitting attribute in the top (root) of the Decision Tree if one uses the Gini index?
(g) Construct the full Decision Tree, using the error rate as heuristic, and what is the overall classification error rate on the above dataset?
(h) Is this classification error rate an optimistic or pessimistic estimate of the error rate on unseen new data? Explain your answer.

a) • ClassErrorRate$(A) = \frac{6 + 4 + 6}{54} \simeq 0.30$

A	Y	O	label	# errors	TOT
a_1	6	18	O	6	24
a_2	12	4	Y	4	16
a_3	8	6	Y	6	14

b) • ClassErrorRate$(B) = \frac{8 + 18}{54} \simeq 0.481$

A	Y	O	label	# errors	TOT
b_1	18	18	/	18	36
b_2	8	10	O	8	18

c) If one uses classification error rate as a measure, the splitting attribute in the top (root) of the DT will be "A" because we seek for a smaller class. error value to minimize entropy.

d) $\text{GINI} = \sum_{k=1}^{K} \hat p_{mk}(1 - \hat p_{mk}) = 1 - \sum_{i=1}^{C} p_i^2$ where p_i = proportion of istances ∈ class i, C = total # of classes

$= \sum \frac{\text{category total}}{\text{total samples}} \cdot G(\text{category})$

$G(A) = \frac{24}{54} \cdot G_{a_1} + \frac{16}{54} \cdot G_{a_2} + \frac{14}{54} \cdot G_{a_3}$

$G_{a_1} = 1 - \left[\left(\frac{6}{24}\right)^2 + \left(\frac{18}{24}\right)\right] = \frac{3}{8} = 0.375$

$G_{a_2} = 1 - \left[\left(\frac{12}{16}\right)^2 + \left(\frac{4}{16}\right)\right] = \frac{3}{8} = 0.375$

$G_{a_3} = 1 - \left[\left(\frac{8}{14}\right)^2 + \left(\frac{6}{14}\right)\right] = \frac{24}{49} \simeq 0.490$

$G(A) = \frac{24}{54} \cdot 0.375 + \frac{16}{54} \cdot 0.375 + \frac{14}{54} \cdot 0.490 \simeq 0.405$

e) $G(B) = \frac{36}{54} \cdot G_{b_1} + \frac{18}{54} \cdot G_{b_2}$

$G_{b_1} = 1 - \left[\left(\frac{18}{36}\right)^2 + \left(\frac{18}{36}\right)\right] = \frac{1}{2} = 0.5$

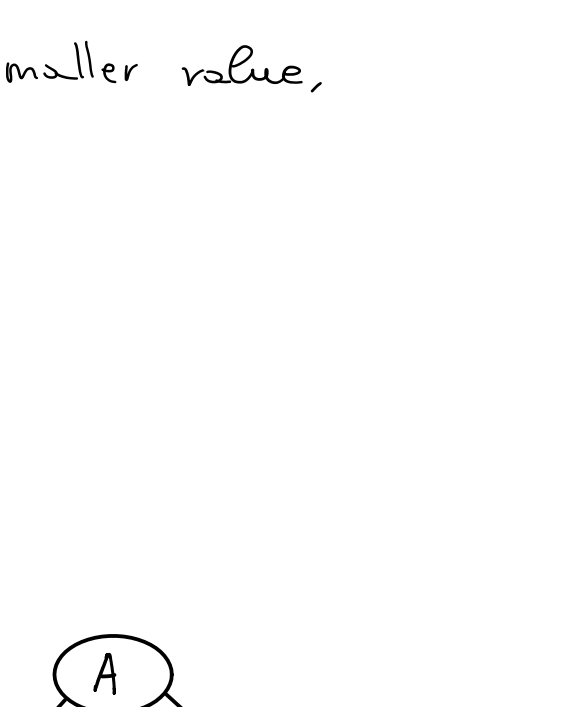
$G_{b_2} = 1 - \left[\left(\frac{8}{18}\right)^2 + \left(\frac{10}{18}\right)\right] = \frac{40}{81} = 0.49$

$G(B) = \frac{36}{54} \cdot 05 + \frac{18}{54} \cdot 0.49 = 0.498$

f) Same as classifiction error rate, we look for the smaller value, the root of the DT will be 'A'.

g) From the previous answers, we know that the root is A so we split A into his 3 branches
Then, we calculate the error rates:

$\text{ErrorRate} = \frac{\text{category errors}}{\text{category total}} = \frac{\text{misclassified istances}}{\text{tot. istances}}$



ErrorRate$(a_1) = \frac{6}{24} \simeq 0.25$

ErrorRate$(a_2) = \frac{4}{16} \simeq 0.25$

ErrorRate$(a_3) = \frac{6}{14} \simeq 0.43$ Since this is the higher, further splits are considered

Since $a_3 > a_1, a_2$, we decide that $a_1 \to O$, $a_2 \to Y$ and a_3 will be split into b_1 and b_2 but we only have b_1.

ErrorRate$(a_3, b_1) = \frac{6}{14} = 0.43$ ⟶ changed to "Y" based on the majority class

Overall Classification Error Rate $= \frac{6 + 4}{54} \simeq 0.203$

h) Since the accuracy is $1 - 0.203 \simeq 0.80$ we can say it is an optimistic estimate

Assignment Data Mining

2.6 DATASET: VOOBEELD7

(a) ScatterPlot

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels as stm
import pyreadstat

chol1, meta = pyreadstat.read_sav("./voorbeeld7_1-1.sav")
print(meta.column_names)
chol1.head()
```

```
['id', 'chol', 'leeftijd', 'bmi', 'actief', 'roken', 'sekse',
'alcohol']
```

```
    id  chol  leeftijd    bmi  actief  roken  sekse  alcohol
0  1.0   7.9      66.0  27.41     1.0    0.0    1.0      0.0
1  2.0   7.5      83.0  26.85     1.0    0.0    1.0      2.0
2  3.0   7.6      84.0  27.24     1.0    0.0    1.0      2.0
3  4.0   6.0      65.0  27.50     1.0    0.0    1.0      2.0
4  5.0   6.2      56.0  27.78     1.0    0.0    1.0      0.0
```

```
print(chol1.describe())
# sekse=sex is a factorial variable
chol1.alcohol.unique() #also alcohol so i will convert them to factors
##chol1 = pd.get_dummies(chol1, columns=["sekse", "alcohol"],
drop_first=True)
```

```
               id        chol    leeftijd         bmi      actief
roken  \
count  200.000000  200.000000  200.000000  200.000000  200.000000
200.000000
mean   100.500000    6.283400   61.895000   28.241310    1.955000
0.350000
std     57.879185    0.802654    6.310554    2.961234    0.784892
0.478167
min      1.000000    4.400000   49.000000   20.797000    1.000000
0.000000
25%     50.750000    5.800000   57.750000   26.271250    1.000000
0.000000
50%    100.500000    6.200000   62.000000   28.125000    2.000000
0.000000
75%    150.250000    6.732500   65.000000   29.732500    3.000000
```

```
1.000000
max    200.000000    8.200000   84.000000   37.565000    3.000000
1.000000
```

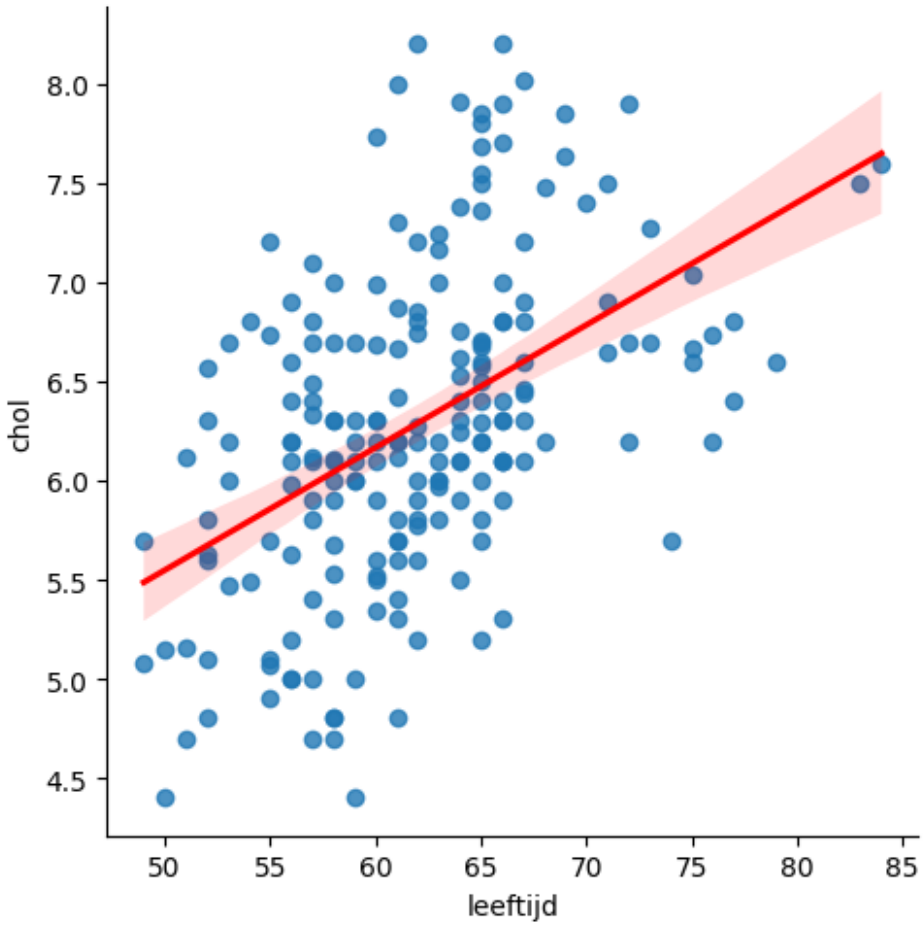
```
            sekse     alcohol
count  200.000000  200.000000
mean     0.410000    0.765000
std      0.493068    0.722763
min      0.000000    0.000000
25%      0.000000    0.000000
50%      0.000000    1.000000
75%      1.000000    1.000000
max      1.000000    2.000000
```

```
array([0., 2., 1.])
```

chol1

```
        id  chol  leeftijd     bmi  actief  roken  sekse  alcohol
0      1.0  7.90      66.0  27.410     1.0    0.0    1.0      0.0
1      2.0  7.50      83.0  26.850     1.0    0.0    1.0      2.0
2      3.0  7.60      84.0  27.240     1.0    0.0    1.0      2.0
3      4.0  6.00      65.0  27.500     1.0    0.0    1.0      2.0
4      5.0  6.20      56.0  27.780     1.0    0.0    1.0      0.0
..     ...   ...       ...     ...     ...    ...    ...      ...
195  196.0  6.53      64.0  30.417     3.0    1.0    0.0      1.0
196  197.0  6.50      65.0  30.300     3.0    1.0    0.0      1.0
197  198.0  6.40      66.0  29.910     3.0    1.0    0.0      1.0
198  199.0  5.50      64.0  27.400     3.0    1.0    0.0      1.0
199  200.0  6.90      71.0  33.860     3.0    1.0    0.0      1.0

[200 rows x 8 columns]
```

```
sns.lmplot(data=chol1, y='chol', x='leeftijd',fit_reg=True,
line_kws={"color": "red"})
plt.show()
```



INTERPRETATION: As age increases, cholesterol levels tend to rise. However, the spread of points around the line indicates that other factors may also influence cholesterol levels. This plot suggests that age is a significant but not exclusive predictor of cholesterol levels.

(b) Fit a Linear Model

Fit a linear model for chol with leeftijd using the function ols (using the statsmodels library). The formula for the model is chol~leeftijd. Save the fitobject under the name fit1. View the result with fit1.summary().

```
import statsmodels.api as sm
import statsmodels.formula.api as smf

fit1 = smf.ols("chol ~ leeftijd", data=chol1).fit()
print(fit1.summary())
```

```
                            OLS Regression Results

==============================================================================
Dep. Variable:                   chol   R-squared:
0.236
```

```
Model:                            OLS   Adj. R-squared:
0.232
Method:                 Least Squares   F-statistic:
61.19
Date:                Thu, 20 Feb 2025   Prob (F-statistic):
3.04e-13
Time:                        15:44:13   Log-Likelihood:
-212.39
No. Observations:                 200   AIC:
428.8
Df Residuals:                     198   BIC:
435.4
Df Model:                           1

Covariance Type:            nonrobust

==============================================================================
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
Intercept      2.4584      0.492      5.002      0.000       1.489
3.428
leeftijd       0.0618      0.008      7.822      0.000       0.046
0.077
==============================================================================
Omnibus:                        3.098   Durbin-Watson:
1.036
Prob(Omnibus):                  0.212   Jarque-Bera (JB):
3.162
Skew:                           0.286   Prob(JB):
0.206
Kurtosis:                       2.770   Cond. No.
615.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

INTERPRETATION: The model indicates a significant positive relationship between age and cholesterol levels. However, with an R-squared of only 0.236, age alone does not explain the majority of the variation in cholesterol levels. Adding other relevant variables (such as smoking status, gender, education level, etc.) may improve the model's performance.

(c) Fit a Multiple Linear Model

Fit a model fit2 for chol with leeftijd, bmi, sekse and alcohol. Which factors are statistically significant?

```
fit2 = smf.ols("chol ~ leeftijd + bmi + sekse + alcohol",
data=chol1).fit()
print(fit2.summary())
```

```
                            OLS Regression Results

==============================================================================
========
Dep. Variable:                    chol   R-squared:
0.396
Model:                             OLS   Adj. R-squared:
0.383
Method:                  Least Squares   F-statistic:
31.93
Date:                 Thu, 20 Feb 2025   Prob (F-statistic):
1.83e-20
Time:                         15:44:13   Log-Likelihood:
-188.94
No. Observations:                 200   AIC:
387.9
Df Residuals:                     195   BIC:
404.4
Df Model:                           4

Covariance Type:            nonrobust

==============================================================================
========
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--------
Intercept     -0.1829      0.609     -0.301      0.764      -1.383
1.017
leeftijd       0.0329      0.008      3.921      0.000       0.016
0.049
bmi            0.1330      0.019      6.937      0.000       0.095
0.171
sekse          1.0012      0.189      5.293      0.000       0.628
1.374
alcohol        0.3492      0.109      3.212      0.002       0.135
0.564
==============================================================================
========
Omnibus:                        4.439   Durbin-Watson:
```

```
1.015
Prob(Omnibus):                  0.109   Jarque-Bera (JB):
4.221
Skew:                           0.258   Prob(JB):
0.121
Kurtosis:                       3.489   Cond. No.
940.
==============================================================================
========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

INTERPRETATION: The model demonstrates that age, BMI, sex, and alcohol consumption are all significant predictors of cholesterol levels. Including these additional variables improved the model's explanatory power from 23.6% to 39.6%.

COMMENT

Considering fit2, we see that the all the factors are statistically significant beside the Intercept. As their p-values are less than 0.05.

```
## this if we want to use one-hot encoding
### convert to factors
chol2 = chol1.copy()
chol2["sekse"] = chol2["sekse"].astype(str)
chol2["alcohol"] = chol2["alcohol"].astype(str)

chol2 = pd.get_dummies(chol2, columns=["sekse", "alcohol"],
drop_first=True)

chol2 = chol2.rename(columns={
    "sekse_1.0": "sekse_1",
    "alcohol_1.0": "alcohol_Low",
    "alcohol_2.0": "alcohol_High"
})

print(chol2.columns
      )
fit2_factors = smf.ols("chol ~ leeftijd + bmi + actief + roken +
sekse_1 + alcohol_Low + alcohol_High", data=chol2).fit()
print(fit2_factors.summary())

Index(['id', 'chol', 'leeftijd', 'bmi', 'actief', 'roken', 'sekse_1',
       'alcohol_Low', 'alcohol_High'],
      dtype='object')
                            OLS Regression Results
```

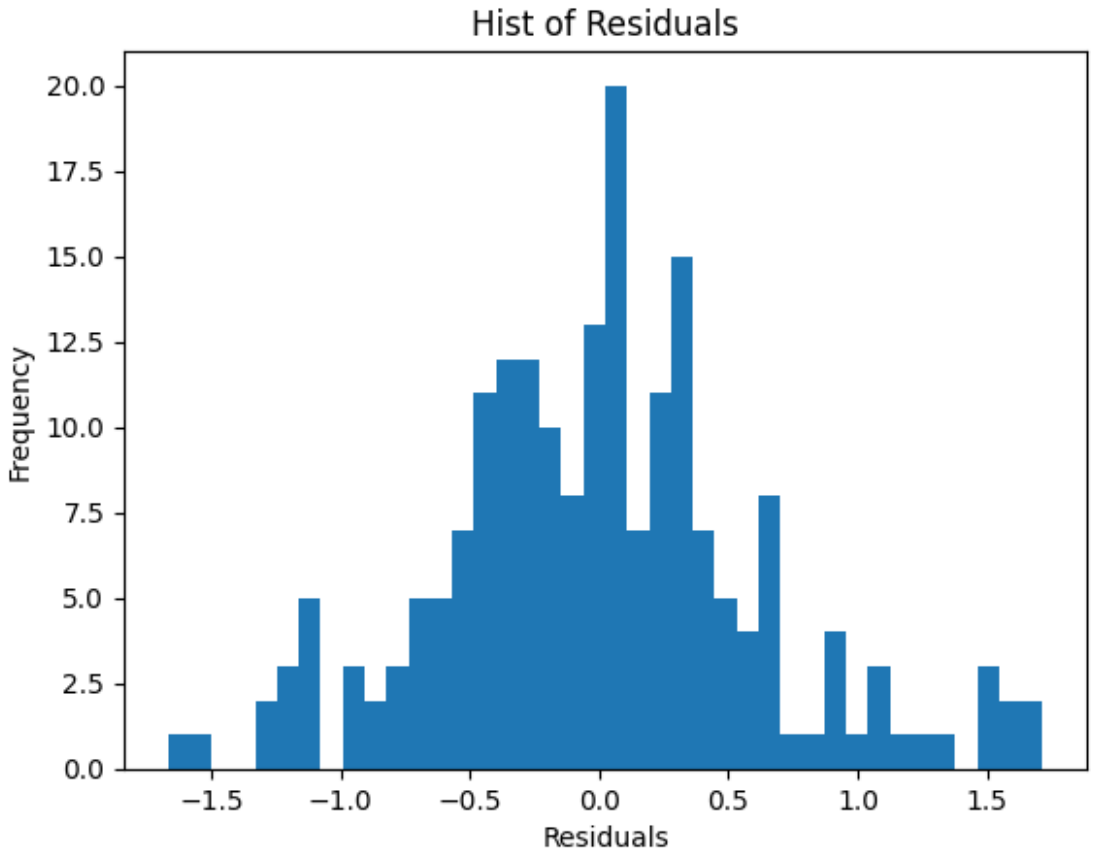
```
==============================================================================
Dep. Variable:                   chol   R-squared:                       0.489
Model:                            OLS   Adj. R-squared:                  0.471
Method:                 Least Squares   F-statistic:                     26.29
Date:                Thu, 20 Feb 2025   Prob (F-statistic):           4.64e-25
Time:                        15:44:13   Log-Likelihood:                -172.11
No. Observations:                 200   AIC:                             360.2
Df Residuals:                     192   BIC:                             386.6
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept                 -0.7894      0.575     -1.372      0.172      -1.924       0.345
sekse_1[T.True]            0.5087      0.264      1.927      0.055      -0.012       1.029
alcohol_Low[T.True]       -0.4757      0.269     -1.770      0.078      -1.006       0.055
alcohol_High[T.True]       0.1643      0.249      0.661      0.510      -0.326       0.655
leeftijd                   0.0354      0.008      4.220      0.000       0.019       0.052
bmi                        0.1672      0.020      8.297      0.000       0.127       0.207
actief                    -0.0044      0.063     -0.070      0.945      -0.128       0.120
roken                      0.3861      0.100      3.870      0.000       0.189       0.583
==============================================================================
Omnibus:                        6.605   Durbin-Watson:                   1.136
Prob(Omnibus):                  0.037   Jarque-Bera (JB):                6.780
Skew:                           0.448   Prob(JB):
```

```
0.0337
Kurtosis:                       2.899   Cond. No.
965.
==============================================================================
========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

INTERPRETATION: The model shows that Age, BMI, and Smoking are strong predictors of cholesterol levels. However, some variables like Physical Activity, Alcohol Consumption, and Gender are not clearly significant. The R-squared of 0.489 is an improvement from previous models, but there may still be unexplained variance.

(d) Residuals and Histogram

```
chol1["residuals"] = fit2.resid
plt.hist(chol1["residuals"], bins=40)
plt.xlabel("Residuals")
plt.ylabel("Frequency")
plt.title("Hist of Residuals")
plt.show()
```



INTERPRETATION: The residuals are approximately normally distributed, which supports the validity of the regression model. Minor deviations from normality are not likely to affect the model's performance significantly.

2.7 DATASET: Births

```
births = pd.read_csv('births-1.csv')
births['home'] = births['child_birth'].apply(lambda x: 'at_home' if
'first line' in x and 'at home' in x else 'not_at_home')

births.head(10)
```

```
   provmin        urban                                    child_birth
\
0       68       strong                 first line child birth, at home

1       12     moderate              first line child birth, outpatient

2       99          not              first line child birth, outpatient

3       68     moderate          during pregnacy referred to specialist

4       68       strong     originally at home, referred to specialist

5       11          not                 first line child birth, at home

6       69       strong                 first line child birth, at home

7       99          not  originally outpatient, referred to specialist

8       12          not     originally at home, referred to specialist

9       68  very strong                 first line child birth, at home


      age_cat  age       etnicity  parity         home
0  25-29 year   26          Dutch       2      at_home
1  25-29 year   29          Dutch       1  not_at_home
2  25-29 year   25  Mediterranean       2  not_at_home
3  30-34 year   30          Dutch       3  not_at_home
4  30-34 year   33          Dutch       1  not_at_home
5   > 35 year   39          Dutch       3      at_home
6  30-34 year   32          Dutch       4      at_home
7  25-29 year   25          Dutch       3  not_at_home
8  25-29 year   29          Dutch       1  not_at_home
9  25-29 year   29          Dutch       2      at_home
```

(b) Recode the variable parity in a new variable pari where pari has level primi if it concerns a first

```
births['pari'] = births['parity'].apply(lambda x: 'primi' if x<2 else
'multi')
```

(c) Recode the variable etnicity into a new variable etni where etni has level Dutch if the woman

```
births['etnicity'].unique()

array(['Dutch', 'Mediterranean', 'Hindu', 'other European', 'Creole',
       'Asian', 'other'], dtype=object)

births['etni'] = births['etnicity'].apply(lambda x: 'Dutch' if
x=='Dutch' else 'Not Dutch')
```

(d) Logistic Regression Model

- Using the sklearn library, create a logistic regression model with the function `LogisticRegression` to predict the probability of childbirth at home.
- Use the variables:
  - `pari`, `age_cat` (age categorized), `etni`, and `urban` (urbanization degree).
- View the model's outcomes using:

```
classification_report()
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split

# One-Hot Encoding categorical features
categorical_features = ['age_cat', 'etni', 'urban']
encoder = OneHotEncoder(drop='first', sparse=False)
encoded_features = encoder.fit_transform(births[categorical_features])
encoded_df = pd.DataFrame(encoded_features,
columns=encoder.get_feature_names_out(categorical_features))

# Merge with numerical data
X = pd.concat([births[['parity']], encoded_df], axis=1)
y = births['home']

# Convert target variable to binary (if needed)
y = y.map({'at_home': 1, 'not_at_home': 0})  # Adjust based on actual
labels

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=7)
```

```
clf = LogisticRegression(random_state=0)
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

report = classification_report(y_test, y_pred)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.65      0.83      0.73      5866
           1       0.59      0.35      0.44      4075

    accuracy                           0.63      9941
   macro avg       0.62      0.59      0.58      9941
weighted avg       0.63      0.63      0.61      9941


c:\Users\Mardeen\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\preprocessing\_encoders.py:972: FutureWarning:
`sparse` was renamed to `sparse_output` in version 1.2 and will be
removed in 1.4. `sparse_output` is ignored unless you leave `sparse`
to its default value.
  warnings.warn(
```

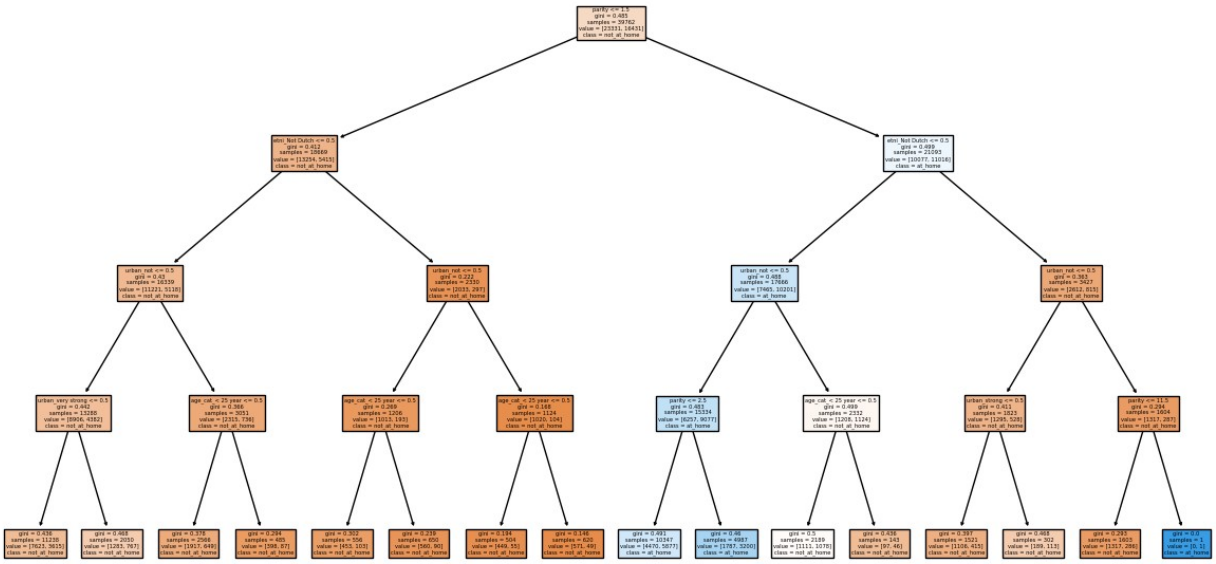
INTERPRETATION: The model performs reasonably well for predicting class 0 but struggles with class 1, particularly due to low recall. This suggests the model has a high false negative rate for class 1, which could be problematic depending on the application. The imbalance between precision and recall for class 1 indicates that the model may need further optimization or rebalancing techniques.

(e) Decision Tree Model

```
from sklearn.tree import DecisionTreeClassifier, plot_tree
dt_model = DecisionTreeClassifier(random_state=0, max_depth=4)
dt_model.fit(X_train, y_train)

y_pred = dt_model.predict(X_test)

report = classification_report(y_test, y_pred)
print("Decision Tree Classification Report:\n", report)

# Plot Decision Tree
plt.figure(figsize=(15, 8))
plot_tree(dt_model, feature_names=X.columns.tolist(),
class_names=['not_at_home', 'at_home'], filled=True)
plt.show()
```

```
Decision Tree Classification Report:
               precision    recall  f1-score   support
```

```
              0       0.69      0.72      0.71      5866
              1       0.58      0.54      0.56      4075

    accuracy                           0.65      9941
   macro avg       0.64      0.63      0.63      9941
weighted avg       0.65      0.65      0.65      9941
```



INTERPRETATION: The Decision Tree model performs better for class 0 than for class 1, especially in terms of recall. While the overall accuracy is acceptable, the lower recall and precision for class 1 suggest that the model struggles with identifying positive instances correctly.

(f) Model Comparison with Cross-Validation

```
from sklearn.model_selection import cross_val_score

log_reg = LogisticRegression(random_state=0)
decision_tree = DecisionTreeClassifier(random_state=0, max_depth=4)

# Perform Cross-Validation (5-Fold)
log_reg_scores = cross_val_score(log_reg, X_train, y_train, cv=5,
scoring='accuracy')
dt_scores = cross_val_score(decision_tree, X_train, y_train, cv=5,
scoring='accuracy')

print(f"Logistic Regression Accuracy (Cross-Validation Mean):
{log_reg_scores.mean():.4f}")
print(f"Decision Tree Accuracy (Cross-Validation Mean):
{dt_scores.mean():.4f}")
```

```
better_model = "Logistic Regression" if log_reg_scores.mean() >
dt_scores.mean() else "Decision Tree"
print(f"\nBetter Model based on Cross-Validation: {better_model}")
```

```
Logistic Regression Accuracy (Cross-Validation Mean): 0.6424
Decision Tree Accuracy (Cross-Validation Mean): 0.6571

Better Model based on Cross-Validation: Decision Tree
```

INTERPRETATION: Based on Cross-Validation Accuracy, the Decision Tree model is the better-performing model. However, it is important to consider other performance metrics like precision, recall, F1-score, and ROC-AUC, especially if the dataset is imbalanced or if there are concerns about false positives or false negatives.