

STA 336: STATISTICAL MACHINE LEARNING

Homework 3

Rento Saijo

February 20, 2026

Table of Contents

Disclosure	2
Problem 1	3
Problem 1 Part (a)	4
Problem 1 Part (b)	5
Problem 2	6
Problem 2 Part (a)	7
Problem 2 Part (b)	8
Problem 3	9
Problem 3 Part (a)	10
Problem 3 Part (b)	11
Problem 3 Part (c)	14
Problem 3 Part (f)	15

Disclosure

GPT-5.3-Codex was used to create the YAML portion and some `LaTeX` code to format the text/equations nicely. Page formatting code was also provided by Derin Gezgin. In the setup chunk, libraries were loaded and some helper functions were defined including but not limited to `table_latex()` and `with_family()`. See the original RMD file [here](#) for more details.

Problem 1

PROBLEM 1

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 coefficients

$$\hat{\beta}_0 = -6, \quad \hat{\beta}_1 = 0.05, \quad \hat{\beta}_2 = 1.$$

Problem 1 Part (a)**PROBLEM 1 PART (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.

Using the fitted logistic model,

$$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -6 + 0.05X_1 + X_2,$$

$$\frac{\hat{p}}{1-\hat{p}} = e^{-6+0.05X_1+X_2}.$$

Solving for \hat{p} , we get:

$$\begin{aligned}\hat{p} &= (1 - \hat{p})e^{-6+0.05X_1+X_2} \\ \hat{p} &= e^{-6+0.05X_1+X_2} - \hat{p}e^{-6+0.05X_1+X_2} \\ \hat{p}(1 + e^{-6+0.05X_1+X_2}) &= e^{-6+0.05X_1+X_2} \\ \hat{p} &= \frac{e^{-6+0.05X_1+X_2}}{1 + e^{-6+0.05X_1+X_2}}.\end{aligned}$$

Substituting $X_1 = 40$ and $X_2 = 3.5$ into the \hat{p} equation, we get:

$$\begin{aligned}\hat{p} &= \frac{e^{-6+0.05(40)+3.5}}{1 + e^{-6+0.05(40)+3.5}} \\ &= \frac{e^{-0.5}}{1 + e^{-0.5}} \\ &= \frac{1}{1 + e^{0.5}} \\ &= 0.3775.\end{aligned}$$

Therefore, the estimated probability that the student gets an A in the class is 0.3775.

Problem 1 Part (b)**PROBLEM 1 PART (B)**

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

A target probability of .50 (that is, 50%) means $\hat{p} = 0.5$. Substituting $\hat{p} = 0.5$ and $X_2 = 3.5$ into

$$\log\left(\frac{\hat{p}}{1 - \hat{p}}\right) = -6 + 0.05X_1 + X_2,$$

we get:

$$\begin{aligned}\log\left(\frac{0.5}{1 - 0.5}\right) &= -6 + 0.05X_1 + 3.5 \\ 0 &= -6 + 0.05X_1 + 3.5 \\ 0.05X_1 &= 2.5 \\ X_1 &= \frac{2.5}{0.05} \\ &= 50.\end{aligned}$$

Therefore, the student in part (a) would need to study 50 hours to have a 50% chance of getting an A in the class.

Problem 2

PROBLEM 2

This problem has to do with *odds*.

Problem 2 Part (a)**PROBLEM 2 PART (A)**

On average, what fraction of people with an odds of 0.37 of defaulting on their credit card payment will in fact default?

Let p be the probability that a person defaults. By definition of odds,

$$\frac{p}{1-p} = 0.37.$$

Solving for p , we get:

$$\begin{aligned} p &= 0.37(1-p) \\ p &= 0.37 - 0.37p \\ 1.37p &= 0.37 \\ p &= \frac{0.37}{1.37} \\ &= \frac{37}{137}. \end{aligned}$$

Therefore, the fraction of people who will default is $\boxed{\frac{37}{137}}$, which is 27.0% rounded to one decimal place.

Problem 2 Part (b)

PROBLEM 2 PART (B)

Suppose that an individual has a 16% chance of defaulting on her credit card payment.
What are the odds that she will default?

If the default probability is $p = 0.16$, then the odds of default are

$$\begin{aligned}\frac{p}{1-p} &= \frac{0.16}{1-0.16} \\ &= \frac{0.16}{0.84} \\ &= \frac{16}{84} \\ &= \frac{4}{21}.\end{aligned}$$

Therefore, the odds that she defaults are $\boxed{\frac{4}{21}}$ (equivalently, 4 : 21).

Problem 3

PROBLEM 3

In this problem, we develop a model to predict whether a given car gets high or low gas mileage based on the `Auto` data set.

Problem 3 Part (a)

PROBLEM 3 PART (A)

Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median.

```
Cell 1
1 # Create mpg01 variable.
2 auto_df <- ISLR2::Auto %>%
3   dplyr::mutate(mpg01 = dplyr::if_else(mpg > median(mpg), 1L, 0L)) %>%
4   dplyr::relocate(mpg01, .before = mpg)
5
6 # Compute core results.
7 median_mpg <- median(auto_df$mpg)
8 class_balance <- as.integer(table(auto_df$mpg01))
9
10 # Build display table.
11 part3a_tbl <- tibble::tibble(
12   metric = c('Median mpg', 'Count for mpg01 = 0', 'Count for mpg01 = 1'),
13   value = c(median_mpg, class_balance[1], class_balance[2])
14 )
15 part3a_tbl %>%
16   table_latex(
17     col_names = c('Result', 'Value'),
18     caption = 'Constructed Binary Response mpg01'
19   )
```

Table 1: Constructed Binary Response mpg01

Result	Value
Median mpg	22.75
Count for mpg01 = 0	196.00
Count for mpg01 = 1	196.00

Problem 3 Part (b)

PROBLEM 3 PART (B)

Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

Cell 2

```

20 # Set feature order for boxplots.
21 features <- c('cylinders', 'displacement', 'horsepower', 'weight',
22   'acceleration', 'year')
23
24 # Reshape data for faceted boxplots.
25 auto_long <- auto_df %>%
26   dplyr::select(mpg01, tidyselect::all_of(features)) %>%
27   dplyr::mutate(mpg01 = factor(mpg01, levels = c(0, 1), labels = c('Below Med.
28     MPG', 'Above Med. MPG'))) %>%
29   tidyr::pivot_longer(cols = -mpg01, names_to = 'feature', values_to = 'value')
30   %>%
31   dplyr::mutate(feature = factor(feature, levels = features))
32
33 # Boxplots
34 ggplot2::ggplot(auto_long, ggplot2::aes(x = mpg01, y = value, fill = mpg01)) +
35   ggplot2::geom_boxplot(alpha = 0.8, outlier.alpha = 0.3) +
36   ggplot2::facet_wrap(~ feature, scales = 'free_y', ncol = 3) +
37   ggplot2::labs(
38     x      = NULL,
39     y      = 'Feature Value',
40     fill   = 'MPG Class'
41   ) +
42   ggplot2::theme_minimal(base_family = 'cmuserif', base_size = 8) +
43   ggplot2::theme(
44     strip.text      = ggplot2::element_text(size = 7),
45     axis.text       = ggplot2::element_text(size = 6),
46     axis.title      = ggplot2::element_text(size = 7),
47     legend.text     = ggplot2::element_text(size = 6),
48     legend.title    = ggplot2::element_text(size = 7),
49     legend.position = 'bottom'
50   )

```

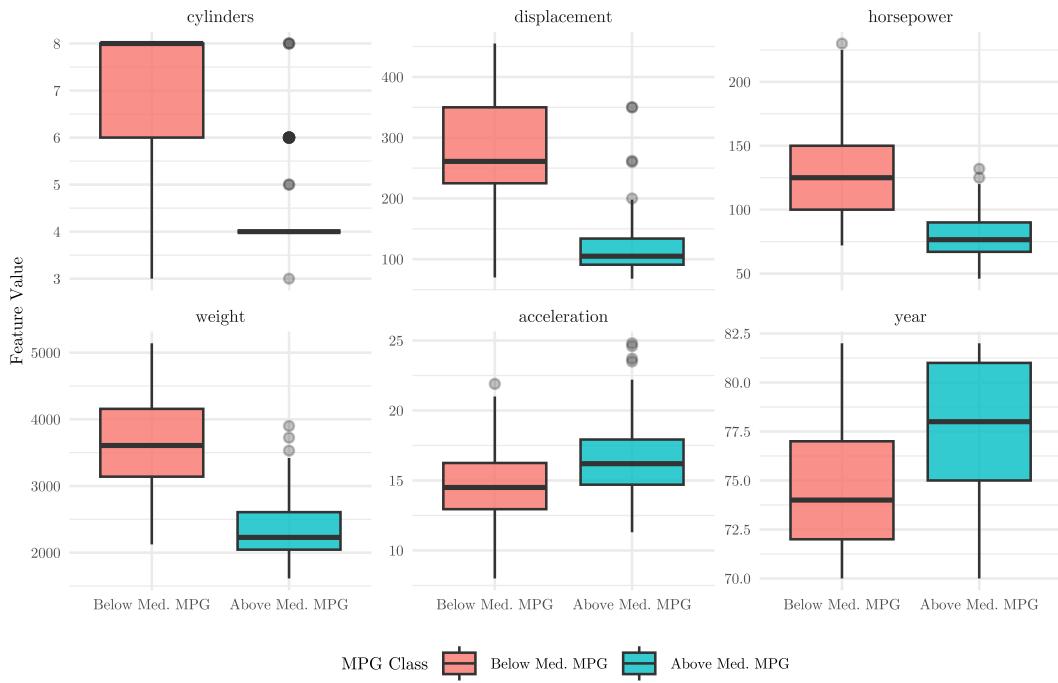


Figure 1: Distributions of Predictors by MPG Class.

These boxplots show strongest separation for `cylinders`, `displacement`, `horsepower`, and `weight` while `year` and `acceleration` overlaps more.

Cell 3

```

48 # Prepare data for pair plot.
49 pairs_df <- auto_df %>%
50   dplyr::transmute(
51     mpg01 = factor(mpg01, levels = c(0, 1), labels = c('Below Med. MPG', 'Above
52       Med. MPG')),
53     cylinders,
54     displacement,
55     horsepower,
56     weight,
57     acceleration,
58     year
59   )
60
61 # Plot pair-wise.
62 GGally::ggpairs(
63   data      = pairs_df,
64   columns   = 2:7,
65   columnLabels = c('Cylinders', 'Displacement', 'Horsepower', 'Weight',
66   'Acceleration', 'Year'),
67   mapping   = ggplot2::aes(color = mpg01),
68   upper     = list(continuous = GGally::wrap('cor', size = 3.3, color = 'black')),
69   lower     = list(continuous = GGally::wrap('points', size = 0.45, alpha = 0.60)),
70 
```

```

68 diag      = list(continuous = GGally::wrap('densityDiag', alpha = 0.55))
69 )
70 +
71 ggplot2::labs(color = 'MPG Class') +
72 ggplot2::theme_minimal(base_family = 'cmuserif', base_size = 8) +
73 ggplot2::theme(
74   strip.text      = ggplot2::element_text(size = 8, color = 'black'),
75   axis.text       = ggplot2::element_text(size = 6, color = 'black'),
76   axis.title      = ggplot2::element_text(color = 'black'),
77   legend.text     = ggplot2::element_text(size = 7, color = 'black'),
78   legend.title    = ggplot2::element_text(size = 8, color = 'black'),
79   panel.grid.major = ggplot2::element_blank(),
80   panel.grid.minor = ggplot2::element_blank(),
81   legend.position = 'bottom'
)

```

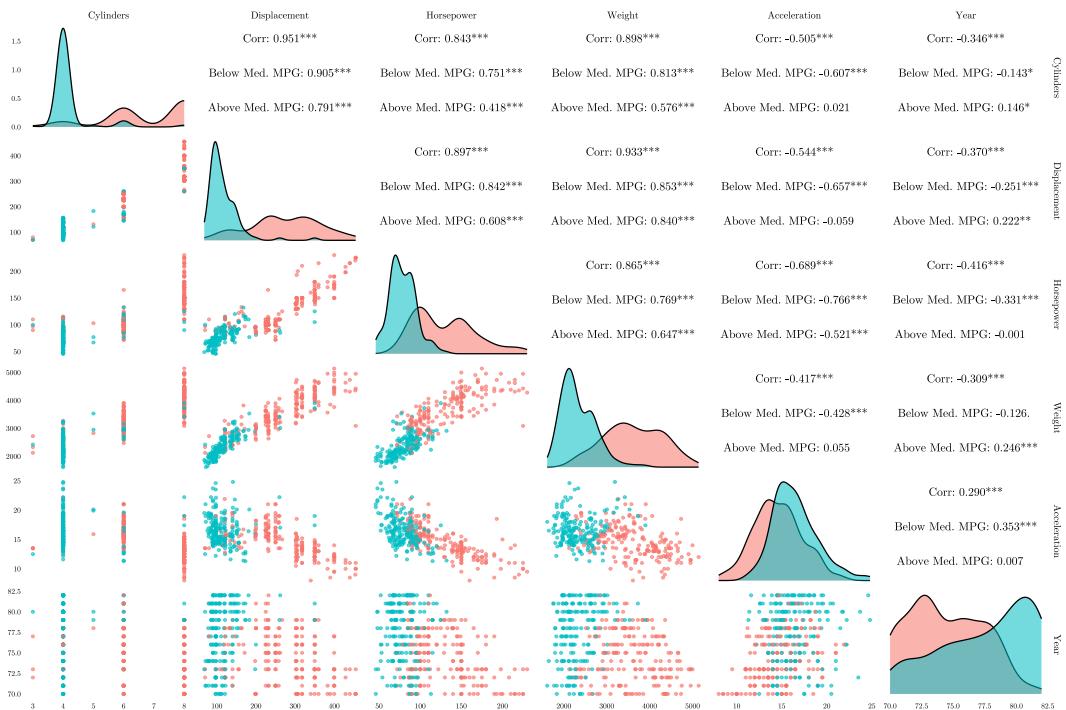


Figure 2: Pairwise Relationships and Correlations by MPG Class

The `ggpairs` matrix confirms that Below Median MPG cars cluster at higher `weight`, `horsepower`, and `displacement`, which supports these as key predictors. Overall, both graphics indicate that `cylinders`, `displacement`, `horsepower`, and `weight` provide the strongest class separation between Below Median MPG and Above Median MPG; `year` adds useful signal, while `acceleration` appears less informative. Therefore, the best predictor set for classification is `cylinders`, `displacement`, `horsepower`, `weight`, and `year`.

Problem 3 Part (c)

PROBLEM 3 PART (C)

Split the data into a training set and a test set.

Cell 4

```
82 # Create train/test split.  
83 set.seed(20060527)  
84 n <- nrow(auto_df)  
85 train_idx <- sample(seq_len(n), n / 2)  
86 auto_train <- auto_df[train_idx, ]  
87 auto_test <- auto_df[-train_idx, ]  
88  
89 # Build split table.  
90 part3c_tbl <- tibble::tibble(  
91   data_set      = c('Training', 'Test'),  
92   observations = c(nrow(auto_train), nrow(auto_test))  
93 )  
94 part3c_tbl %>%  
95   table_latex(  
96   col_names = c('Data set', 'Observations'),  
97   caption   = 'Train/Test Split Sizes.'  
98 )
```

Table 2: Train/Test Split Sizes.

Data set	Observations
Training	196
Test	196

Problem 3 Part (f)

PROBLEM 3 PART (F)

Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

Using the predictors selected in part (b), fit

$$\text{mpg01} \sim \text{cylinders} + \text{horsepower} + \text{weight} + \text{year}.$$

Then use the rule $\hat{y} = 1$ if $\hat{p} > 0.5$, else $\hat{y} = 0$, and compute

$$\text{test error} = \text{Ave}(I(y_0 \neq \hat{y}_0)).$$

Cell 5

```

99 # Fit logistic model.
100 logit_fit <- stats::glm(
101   mpg01 ~ cylinders + horsepower + weight + year,
102   data   = auto_train,
103   family = stats::binomial
104 )
105
106 # Create test predictions.
107 test_prob <- stats::predict(logit_fit, newdata = auto_test, type = 'response')
108 test_pred <- dplyr::if_else(test_prob > 0.5, 1L, 0L)
109
110 # Compute test metrics.
111 test_error    <- mean(test_pred != auto_test$mpg01)
112 test_accuracy <- 1 - test_error
113
114 # Print performance table.
115 part3f_tbl <- tibble::tibble(
116   metric = c('Test error', 'Test accuracy'),
117   value   = c(round(test_error, 4), round(test_accuracy, 4))
118 )
119 part3f_tbl %>%
120   table_latex(
121     col_names = c('Metric', 'Value'),
122     caption  = 'Logistic Regression Test Performance'
123   )

```

Table 3: Logistic Regression Test Performance

Metric	Value
Test error	0.102
Test accuracy	0.898

Therefore, the logistic model's test error is [0.1020], i.e., about 10.20%.