STA 336: Statistical Machine Learning

Homework 3

Rento Saijo

February 20, 2026

Table of Contents

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

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

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:

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

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 $\boxed{0.3775}$.

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 $\boxed{\text{50 hours}}$ 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 $\boxed{\dfrac{37}{137}}$, 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 $\boxed{\dfrac{4}{21}}$ (equivalently, 0.19).

Problem 3

Problem 3

In this problem, you will 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

```
# Create mpg01 variable.
auto_df <- ISLR2::Auto %>%
  dplyr::mutate(mpg01 = dplyr::if_else(mpg > median(mpg), 1L, 0L)) %>%
  dplyr::relocate(mpg01, .before = mpg)

# Compute core results.
median_mpg    <- median(auto_df$mpg)
class_balance <- as.integer(table(auto_df$mpg01))

# Build display table.
part3a_tbl <- tibble::tibble(
  metric = c('Median mpg', 'Count for mpg01 = 0', 'Count for mpg01 = 1'),
  value  = c(median_mpg, class_balance[1], class_balance[2])
)
part3a_tbl %>%
  table_latex(
    col_names = c('Result', 'Value'),
    caption   = 'Constructed Binary Response mpg01'
  )
```

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

```r
# Set feature order for boxplots.
features <- c('cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', 'year')

# Reshape data for faceted boxplots.
auto_long <- auto_df %>%
  dplyr::select(mpg01, tidyselect::all_of(features)) %>%
  dplyr::mutate(mpg01 = factor(mpg01, levels = c(0, 1), labels = c('Below Med. MPG', 'Above Med. MPG'))) %>%
  tidyr::pivot_longer(cols = -mpg01, names_to = 'feature', values_to = 'value') %>%
  dplyr::mutate(feature = factor(feature, levels = features))

# Boxplots
ggplot2::ggplot(auto_long, ggplot2::aes(x = mpg01, y = value, fill = mpg01)) +
  ggplot2::geom_boxplot(alpha = 0.8, outlier.alpha = 0.3) +
  ggplot2::facet_wrap(~ feature, scales = 'free_y', ncol = 3) +
  ggplot2::labs(
    x    = NULL,
    y    = 'Feature Value',
    fill = 'MPG Class'
  ) +
  ggplot2::theme_minimal(base_family = 'cmuserif', base_size = 8) +
  ggplot2::theme(
    strip.text      = ggplot2::element_text(size = 7),
    axis.text       = ggplot2::element_text(size = 6),
    axis.title      = ggplot2::element_text(size = 7),
    legend.text     = ggplot2::element_text(size = 6),
    legend.title    = ggplot2::element_text(size = 7),
    legend.position = 'bottom'
  )
```

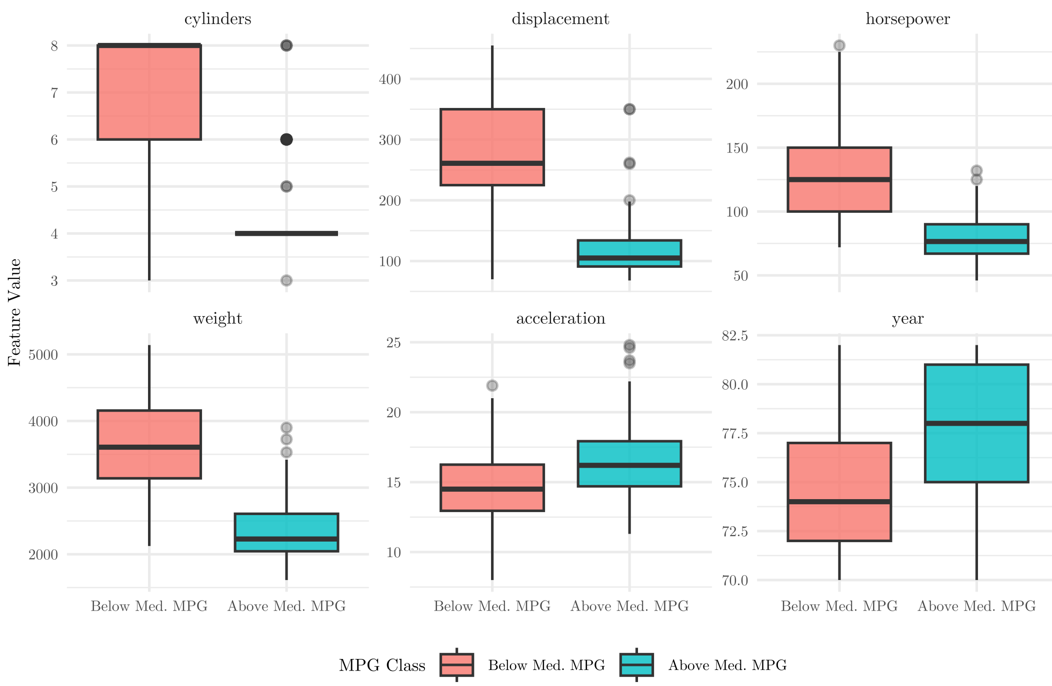

Figure 1: Distributions of Predictors by MPG Class.

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

Cell 3

```
# Prepare data for pair plot.
pairs_df <- auto_df %>%
  dplyr::transmute(
    mpg01 = factor(mpg01, levels = c(0, 1), labels = c('Below Med. MPG', 'Above
    Med. MPG')),
    cylinders,
    displacement,
    horsepower,
    weight,
    acceleration,
    year
  )

# Plot pair-wise.
GGally::ggpairs(
  data    = pairs_df,
  columns = 2:7,
  columnLabels = c('Cylinders', 'Displacement', 'Horsepower', 'Weight',
  'Acceleration', 'Year'),
  mapping = ggplot2::aes(color = mpg01),
  upper   = list(continuous = GGally::wrap('cor', size = 3.3, color = 'black')),
  lower   = list(continuous = GGally::wrap('points', size = 0.45, alpha = 0.60)),
```

```r
  diag    = list(continuous = GGally::wrap('densityDiag', alpha = 0.55))
) +
  ggplot2::labs(color = 'MPG Class') +
  ggplot2::theme_minimal(base_family = 'cmuserif', base_size = 8) +
  ggplot2::theme(
    strip.text       = ggplot2::element_text(size = 8, color = 'black'),
    axis.text        = ggplot2::element_text(size = 6, color = 'black'),
    axis.title       = ggplot2::element_text(color = 'black'),
    legend.text      = ggplot2::element_text(size = 7, color = 'black'),
    legend.title     = ggplot2::element_text(size = 8, color = 'black'),
    panel.grid.major = ggplot2::element_blank(),
    panel.grid.minor = ggplot2::element_blank(),
    legend.position  = 'bottom'
  )
```

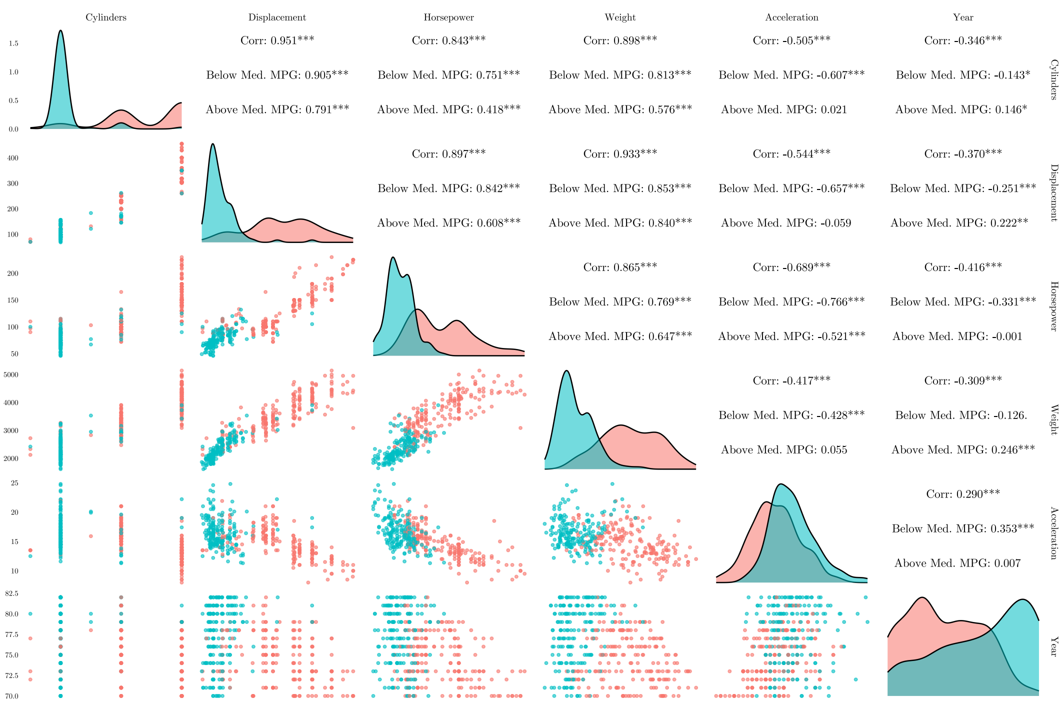

Figure 2: Pairwise Relationships and Correlations by MPG Class where Red denotes Below Median MPG and Blue denotes Above Median MPG

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. Since `cylinders`, `displacement`, `horsepower`, and `weight` are so correlated with one another (visually and by looking at the correlation coefficients), I will just use `displacement`, and `year`.

Problem 3 Part (c)

Problem 3 Part (c)

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

Cell 4

```r
# Create train/test split.
set.seed(20060527)
n <- nrow(auto_df)
train_idx  <- sample(seq_len(n), n / 2)
auto_train <- auto_df[train_idx, ]
auto_test  <- auto_df[-train_idx, ]

# Build split table.
part3c_tbl <- tibble::tibble(
  data_set     = c('Training', 'Test'),
  observations = c(nrow(auto_train), nrow(auto_test))
)
part3c_tbl %>%
  table_latex(
    col_names = c('Data set', 'Observations'),
    caption   = 'Train/Test Split Sizes.'
  )
```

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

$$\texttt{mpg01} \sim \texttt{displacement} + \texttt{year}.$$

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

```r
# Fit logistic model.
logit_fit <- stats::glm(
  mpg01 ~ displacement + year,
  data   = auto_train,
  family = stats::binomial
)

# Extract coefficient summary.
coef_tbl <- as.data.frame(stats::coef(summary(logit_fit)))
coef_tbl <- tibble::rownames_to_column(coef_tbl, var = 'term')

# Rename and round columns.
coef_tbl <- coef_tbl %>%
  dplyr::rename(
    estimate    = Estimate,
    std_error   = `Std. Error`,
    z_value     = `z value`,
    p_value     = `Pr(>|z|)`
  ) %>%
  dplyr::mutate(dplyr::across(-term, ~ round(.x, 4)))

# Print coefficient table.
coef_tbl %>%
  table_latex(
    col_names = c('Term', 'Estimate', 'Std. Error', 'z value', 'Pr(>|z|)'),
    caption   = 'Logistic Regression Coefficient Summary'
  )
```

Table 3: Logistic Regression Coefficient Summary

Term	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-19.1994	6.2028	-3.0953	0.0020
displacement	-0.0424	0.0074	-5.7619	0.0000
year	0.3424	0.0868	3.9437	0.0001

The fitted logistic model is

$$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -19.19940749 - 0.04242666\,\texttt{displacement} + 0.34239667\,\texttt{year},$$

where $\hat{p}$ is the estimated probability that a car is in the `Above Median MPG` class.

Cell 6

```r
# Create test predictions.
test_prob <- stats::predict(logit_fit, newdata = auto_test, type = 'response')
test_pred <- dplyr::if_else(test_prob > 0.5, 1L, 0L)

# Compute test metrics.
test_error    <- mean(test_pred != auto_test$mpg01)
test_accuracy <- 1 - test_error

# Print performance table.
part3f_tbl <- tibble::tibble(
  metric = c('Test error', 'Test accuracy'),
  value  = c(round(test_error, 4), round(test_accuracy, 4))
)
part3f_tbl %>%
  table_latex(
    col_names = c('Metric', 'Value'),
    caption   = 'Logistic Regression Test Performance'
  )
```

Table 4: Logistic Regression Test Performance

Metric	Value
Test error	0.102
Test accuracy	0.898

The model's test error is $\boxed{0.1020}$, i.e., about 10.20%.