Data Analytics in Business

Zhaohu (Jonathan) Fan

Information Technology Management

Scheller College of Business

Georgia Institute of Technology

September 12, 2024

Week 4: Upcoming deadlines and updates

- **Week 4 (Module 4)** is now available on Canvas.
- **(Graded) Self-Assessment 3** has been released and is due by this Sunday, September 15, at 11:59 PM EST.
- **(Graded) Homework #1:** Released last Monday, due by September 22, at 11:59 PM EST. This assignment includes:
  - **Homework #1, Part 1 (Theoretical): One attempt allowed.**
  - **Homework #1, Part 2 (Computation): One attempt allowed.**
  - You can work on both parts as much as you want within the due period, but remember to click "submit" only when you're completely ready.
- **Piazza Forum:** Always open for questions! It's the perfect place to interact with our teaching team and your classmates.
  - Simply click on "Piazza" in the left panel of our Canvas course page.

Main topics

- **Analytics & Modeling (weeks 1-5)**
    - Week 4 (Module 4): Logistic Regression

Logistic regression

Introduction

- **What are Odds**?

$$\text{Odds} \ = \frac{\text{Probability of Event}}{1 - \ \text{Probability of Event}}$$

- For example, if **the probability of an event is 0.8**, then the odds of the event happening would be

$$0.8/(1 - 0.8) = 0.8/0.2 = 4.$$

`

From odds to probability

- If you have the odds and you want to convert it back to probability, you can use the following formula:

$$\text{Probability } = \frac{\text{Odds}}{1 + \text{ Odds}}$$

- For example, if **the odds of an event happening is 4**, then the probability of the event would be

$$4/(1+4) = 4/5 = 0.8$$

Logistic regression

- In logistic regression, the binary outcome variable Y is modeled using predictor variables X.
- Here's how Y fits into the logistic regression equations.
- In terms of Y, the log-odds are modeled as:

$$\log\left(\frac{p(Y=1)}{1-p(Y=1)}\right)=\beta_0+\beta_1X_1+\beta_2X_2+\ldots+\beta_pX_p$$

Here, $p(Y=1)$ is the probability of the event $Y=1$ occurring, and $1-p(Y=1)$ is the probability of the event $Y=0$ occurring.

R code

```
# Fit logistic regression model
model <- glm(response ~ predictor1 + predictor2, data = data, family = binomial())
# Get summary of the model
summary(model)
```

Calculating probability

To calculate the probability `(p(Y=1))` from the log-odds, we can transform the equation:

$$p(Y=1) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p}}$$

R code I

```r
# Predict probabilities
probabilities <- predict(model, type = "response")
```

R code II

```r
# Predict probabilities
probabilities <- predict(model, newdata = newData, type = "response")
```

Revisit the logistic regression model

- Assume we have a logistic regression model that outputs a probability p for each observation.

- The model's task is to classify each observation into one of two classes: positive (1) or negative (0).

- The cutoff value θ is the **threshold** at which you decide whether the predicted probability is high enough to classify an observation as positive. Here's how the classification decision is made based on the cutoff:

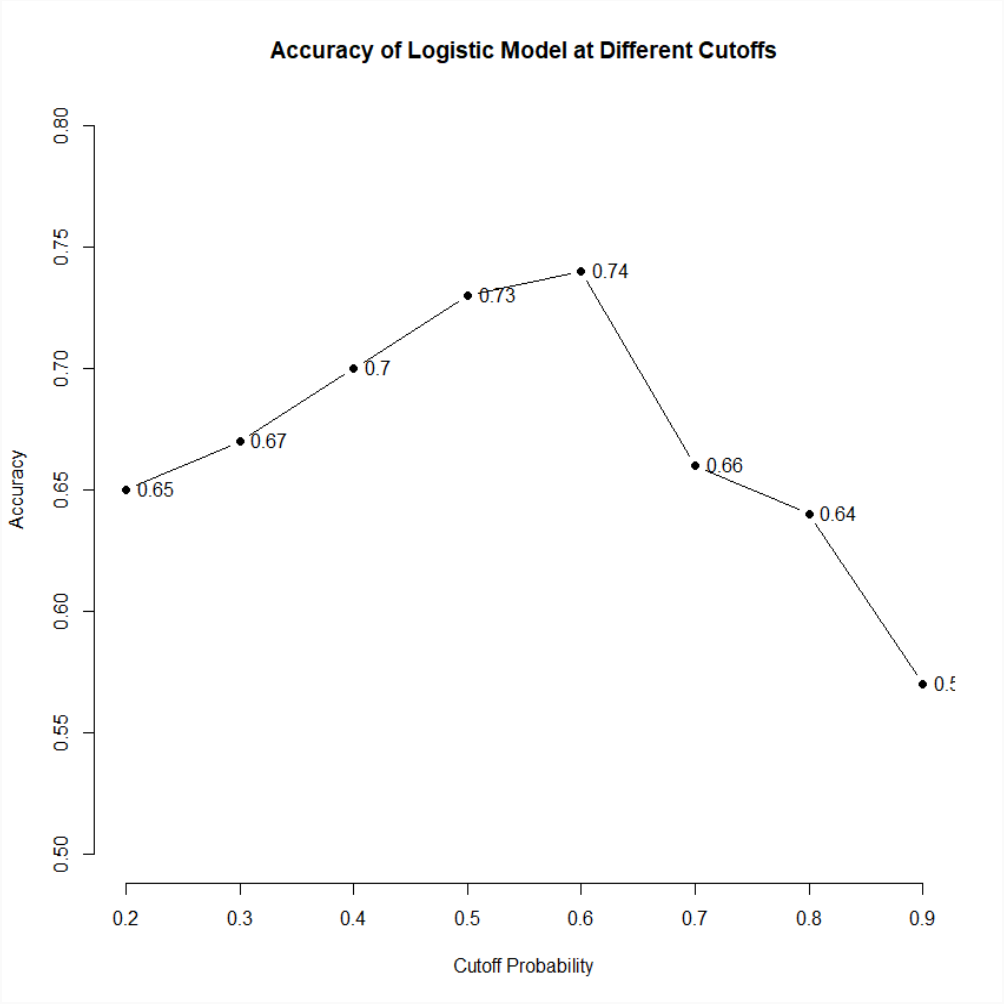
    - $p >= \theta$, classify as positive (1)

    - $p < \theta$, classify as negative (0)

R code

```
# Predict outcomes directly using a threshold
predicted_outcomes <- ifelse(predict(model, type = "response") > 0.5, 1, 0)
```

Plotting the accuracies for each cutoff values



For this R example, please refer to the file named `Use_different_cutoffs_calculate_accuracy.R`.

Statistical concepts

- The **accuracy** is defined as the proportion of true results (both **true positives** and **true negatives**) among the total number of cases examined. Mathematically, **accuracy** is calculated as:
    - **Accuracy**: It's the proportion of true results (both true positives and true negatives) among the total number of cases examined.

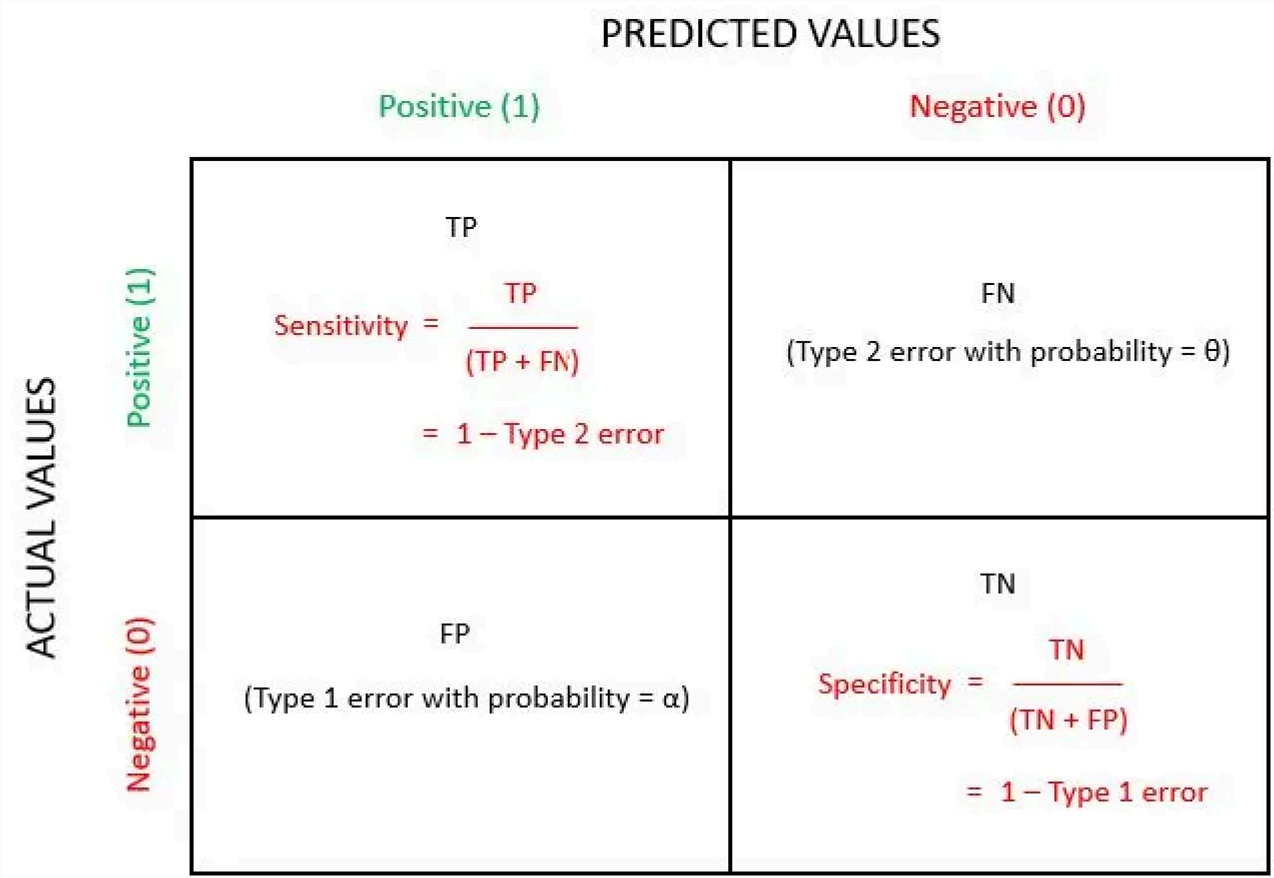
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **Changing the cutoff θ alters the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).**
    - For instance, in a medical testing scenario where missing a positive case (**false negative**) is much more critical than incorrectly identifying a negative case as positive (**false positive**), a lower threshold might be used to ensure a more sensitive test.

Revisit the logistic regression model

- **Confusion Matrix**: Primarily used to evaluate the performance of a classification model (e.g., logistic regression model), providing a clear visualization of the model's accuracy, including errors.
    - **Elements**:
        - **TP (True Positive)** is the number of positives correctly predicted as positive.
        - **TN (True Negative)** is the number of negatives correctly predicted as negative.
        - **FP (False Positive)** is the number of negatives incorrectly predicted as positive.
        - **FN (False Negative)** is the number of positives incorrectly predicted as negative.

Confusion matrix



Perform a diagonal reflection or rotation (transposing the confusion matrix/rotating the matrix)

Confusion matrix (cont'd)

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Fruit basket example

- **Imagine you have a basket of fruit, and your model's job is to identify all the apples among a mix of apples and oranges**.

Image courtesy of AI Image Generator, generated on August 25, 2024.

Fruit basket example (cont'd)

- **Now, imagine we plot every possible TPR against the FPR on a graph — this is our ROC curve. The better our model is at distinguishing apples from oranges, the more the curve will stretch towards the top left corner of the graph.**
  - True Positives (TP): **The apples that your model correctly identifies as apples.**
  - False Positives (FP): **The oranges that your model mistakenly identifies as apples.**
  - True Positive Rate (TPR): **The percentage of all the actual apples that your model correctly identifies.**

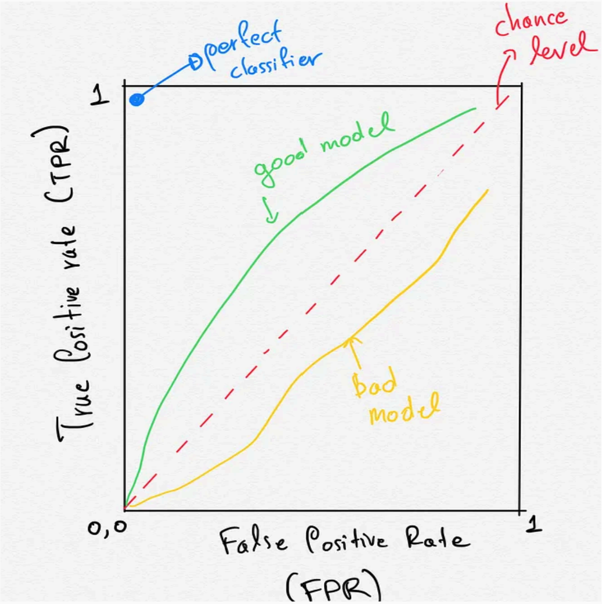
$$\text{True Positive Rate (TPR)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

  - False Positive Rate (FPR): **The percentage of all the actual oranges that your model incorrectly identifies as apples.**

$$\text{False Positive Rate (FPR)} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

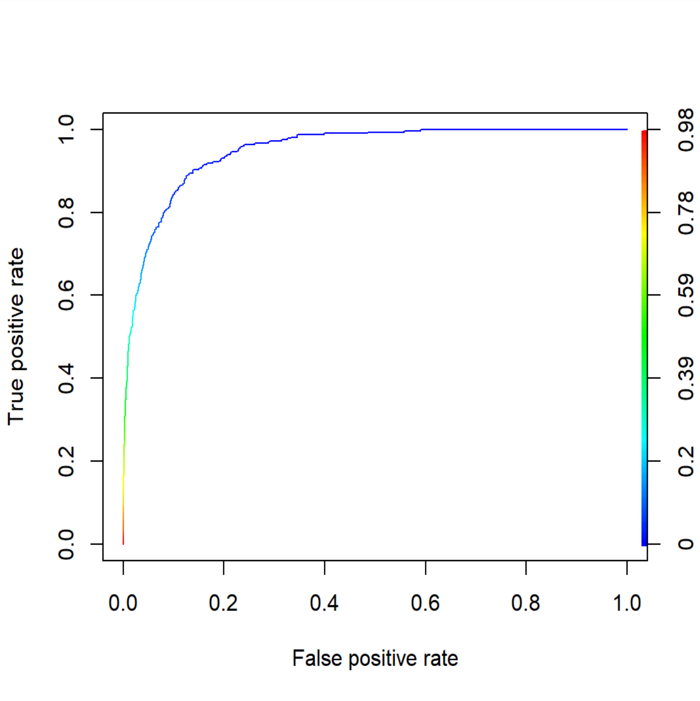
Fruit basket example (cont'd)

- **Now, imagine we plot every possible TPR against the FPR on a graph — this is our ROC curve. The better our model is at distinguishing apples from oranges, the more the curve will stretch towards the top left corner of the graph.**
- **The AUC is like taking a piece of string and measuring the space under this curve.**
    - If our model is no better than random guessing, the AUC will be 0.5 — like a diagonal line from the bottom left to the top right.
    - If our model is perfect, the AUC will be 1, and the ROC curve will hug the left and top borders of the graph.
- So, in our fruit basket example, a higher AUC means our model is really good at picking out just the apples without being fooled by the oranges!



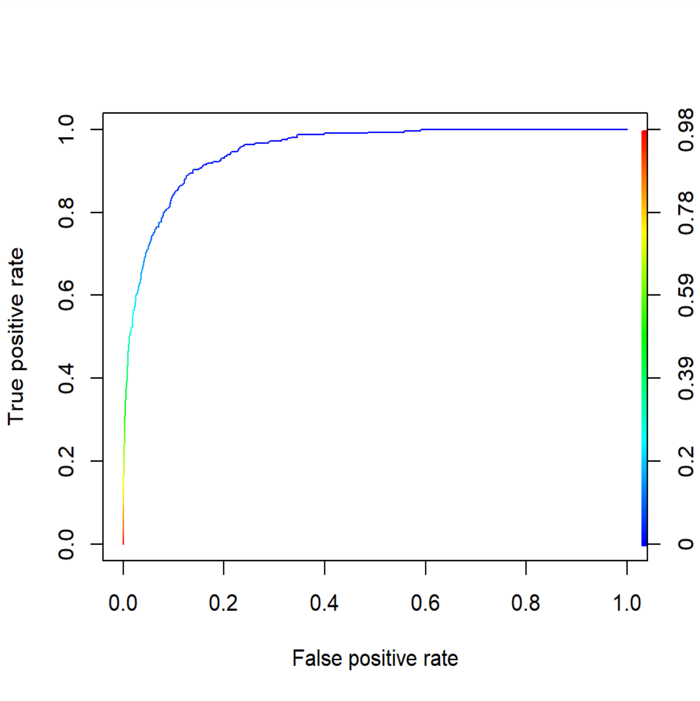
To link the TPR and FPR to the AUC

- **The AUC, short for "Area Under the Curve," is a way to measure how well a model can tell different groups apart, like if an email is spam or not.** The curve it refers to is the ROC curve, which shows the trade-off between catching true positives (like correctly identifying spam) and avoiding false positives (like marking good emails as spam).
- **Calculating TP and FP**: You calculate these rates based on different thresholds from your model's predicted probabilities.
- **Plotting the ROC Curve**: Using **these rates (TPR and FPR)**, you plot the ROC curve.



To link the TPR and FPR to the AUC

- **Computing the AUC**: The AUC is then calculated as the area under the ROC curve. It provides a single measure of overall accuracy that is not dependent on the choice of threshold.
  - **In R, you can use the `pred()`, `performance()`, and `plot()` functions from `ROCR` package to compute the TPR and FPR and then use these to plot the ROC curve and calculate the AUC.**
- For more R examples, please refer to `logistic-regression-I.R` and `logistic-regression-II.R`.



Statistical concepts

- **Accuracy:** It's the proportion of true results (both **true positives** and **true negatives**) among the total number of cases examined.

$$\text{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

- **Sensitivity** (also known as **True Positive Rate** or **Recall**): It's the proportion of actual positives that are correctly identified as such.

$$\text{Sensitivity} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

- **Specificity** (also known as **True Negative Rate**): It's the proportion of actual negatives that are correctly identified.

$$\text{Specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

Example I

Business context

- **A bank wants to predict whether or not a loan will default (Default: 0 = No, 1 = Yes)**.
  - The categorical predictor in this case could be `Account_Type`, with **three levels: "Basic", "Premium", and "Student"**.

- **R code: Simulated data (True)**

```
##    Default Account_Type
## 1        1      Student
## 2        0      Student
## 3        1      Student
## 4        0      Premium
## 5        0      Student
## 6        0      Premium
## 7        0      Premium
## 8        0      Premium
## 9        1      Student
## 10       0        Basic
```

Simulated data

- Fit a logistic regression model to predict `Default` based on `Account_Type`.

```r
# Set seed for reproducibility
set.seed(123)
# Step 1:Load the necessary package
library(dplyr)
# Step 2: Generate example data
n <- 200 # Number of samples
Account_Type <- sample(c("Basic", "Premium", "Student"), size = n, replace = TRUE)
# Step 3: Simulate the Default outcome based on Account_Type
Default <- ifelse(Account_Type == "Basic", sample(c(0, 1), size = n, replace = TRUE, prob = c(0.7, 0.3)),
                  ifelse(Account_Type == "Premium", sample(c(0, 1), size = n, replace = TRUE, prob = c(0.85, 0.15)
                         sample(c(0, 1), size = n, replace = TRUE, prob = c(0.5, 0.5))))
# Step 4: Create a data frame
df <- data.frame(Default, Account_Type)
# Step 5 Convert Account_Type to factor
df$Account_Type <- as.factor(df$Account_Type)
# Step 6: Fit logistic regression model
model <- glm(Default ~ Account_Type, data = df, family = binomial)
# Step 7: Summarize the model
summary(model)
```

Output only the first ten observations

R code

```r
# Set seed for reproducibility
set.seed(123)
# Step 1:Load the necessary package
library(dplyr)
# Step 2: Generate example data
n <- 200 # Number of samples
Account_Type <- sample(c("Basic", "Premium", "Student"), size = n, replace = TRUE)
# Step 3: Simulate the Default outcome based on Account_Type
Default <- ifelse(Account_Type == "Basic", sample(c(0, 1), size = n, replace = TRUE, prob = c(0.7, 0.3)),
                  ifelse(Account_Type == "Premium", sample(c(0, 1), size = n, replace = TRUE, prob = c(0.85, 0.15)
                         sample(c(0, 1), size = n, replace = TRUE, prob = c(0.5, 0.5))))
# Step 4: Create a data frame
df <- data.frame(Default, Account_Type)
head(df,10)
```

Logistic regression

- The intercept estimate of **-0.8475** represents the log-odds of a loan defaulting when the account type is `"Basic"`.
- The estimate for `Account_TypePremium` of **-1.2264** represents the change in the log-odds of the outcome (in this case, loan defaulting) when an account is of type "Premium" as compared to when it is of type "Basic" (the reference level). This **negative** value indicates that being a `Account_TypePremium` is associated with a **decrease** in the log odds of a loan defaulting compared to being a `"Basic"`.
- The estimate for `Account_TypeStudent` of **0.6218** represents the change in the log-odds of the outcome (in this case, loan defaulting) when the account is of the type "Student" compared to when it is of the type "Basic" (which is the reference level).

```
Call:
glm(formula = Default ~ Account_Type, family = binomial, data = df)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.2264  -0.8475  -0.6218   1.3412   2.3945  

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)    
(Intercept)          -0.8475     0.1885  -4.497 6.92e-06 ***
Account_TypePremium -1.2264     0.3123  -3.926 8.64e-05 ***
Account_TypeStudent  0.6218     0.2846   2.186   0.0288 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Logistic regression

- The intercept estimate of **-0.8475** represents the log-odds of a loan defaulting when the account type is `"Basic"`.
    - Odds: The odds can be calculated by exponentiating the log-odds. **Odds=exp(−0.8475)≈0.4285**
    - This means that the odds of a loan defaulting for a "Basic" account type are approximately **0.43 to 1**.
    - Probability: You can also convert the odds to probability using the formula:**Probability=Odds/(1+Odds)=0.4285/(1+0.4285)≈0.3**
    - This indicates that there's approximately a **30%** chance of a loan defaulting when the account type is "Basic".

```
Call:
glm(formula = Default ~ Account_Type, family = binomial, data = df)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.2264  -0.8475  -0.6218   1.3412   2.3945  

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)       -0.8475     0.1885  -4.497 6.92e-06 ***
```

Logistic regression

- The estimate for `Account_TypePremium` of **-1.2264** represents the change in the log-odds of the outcome (in this case, loan defaulting) when an account is of type "Premium" as compared to when it is of type "Basic" (the reference level).
- Odds: To convert this to odds, you would exponentiate the coefficient:$\mathbf{exp(-1.2264) \approx 0.2936}$
  - This means that the odds of defaulting on a loan for a "Premium" account is **0.29** times that of a "Basic" account.
  - The odds of defaulting for a Premium" account are **71** $\%$ lower than those for a "Basic" account.
- If the odds for a "Basic" account to default are approximately **0.43** (as calculated from the intercept in the previous question), then the odds for a "Premium" account would be $\mathbf{0.43 \times 0.2936 \approx 0.1262}$.
- The probability can be calculated as: $\mathbf{Probability = 0.1262/(1+0.1262) \approx 0.112}$.
- The probability of a loan defaulting when the account is "Premium" is approximately **11.2** $\%$.

```
Call:
glm(formula = Default ~ Account_Type, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2264  -0.8475  -0.6218   1.3412   2.3945

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)

Account_TypePremium -1.2264     0.3123  -3.926 8.64e-05 ***
```

Example II

Logistic regression including both continuous and binary predictors

- First, let's create a toy example to demonstrate logistic regression with R, including both continuous and binary predictors.
  - **Outcome**: Whether or not an individual has a certain condition (1 = yes, 0 = no).
  - **Age**: A continuous predictor representing the individual's age.
  - **Smoker**: A binary predictor indicating whether the individual is a smoker (1) or not (0).
- Fit a logistic regression model
  - `model <- glm(Outcome ~ Age + Smoker, data = data, family = binomial(link = "logit"))`

For this R example, please refer to the file named `Logistic-regression-example.R`.

Outputs

- **Intercept**: it's around -2.20.
- **Age**: the estimated coefficient is around 0.04.
- **Smoker**: the estimated coefficient is around 0.84.

```
glm(formula = Outcome ~ Age + Smoker, family = binomial(link = "logit"),
    data = data)
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.1825917  0.0239110  -91.28   <2e-16 ***
Age          0.0398077  0.0005074   78.46   <2e-16 ***
Smoker       0.8395645  0.0134222   62.55   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 138377  on 99999  degrees of freedom
Residual deviance: 128072  on 99997  degrees of freedom
AIC: 128078

Number of Fisher Scoring iterations: 4
```

Interpretation

- **Intercept**: The log odds of having the condition for a non-smoker aged 0 (theoretically) is -2.20.
    - **Age (Continuous)**: For each additional year of age, the log odds of having the condition increases by 0.04, holding the smoking status constant. The odds ratio is exp(0.04)=1.04, indicating a 4 % increase in the odds of having the condition with each additional year of age.
    - **Smoker (Binary)**: Being a smoker increases the log odds of having the condition by 0.84 compared to non-smokers, holding age constant. The odds ratio is exp(0.84)=2.32, meaning smokers are over twice as likely to have the condition as non-smokers, everything else being equal.

```
glm(formula = Outcome ~ Age + Smoker, family = binomial(link = "logit"),
    data = data)
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.1825917  0.0239110  -91.28   <2e-16 ***
Age          0.0398077  0.0005074   78.46   <2e-16 ***
Smoker       0.8395645  0.0134222   62.55   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 138377  on 99999  degrees of freedom
Residual deviance: 128072  on 99997  degrees of freedom
AIC: 128078

Number of Fisher Scoring iterations: 4
```

Open for discussion