Logistic regression in tidymodels

Jaime Davila

3/7/2022

Introduction

Let's start by loading an old friend of ours, the `Default` dataset (remember Homework 3?)

We are interested in predicting whether a person would go on *default* (that is, would not pay back a loan) given the following information:

- `balance` (How much does the person owe?)
- `income` (How much does the person earn?)
- `student` (Is the person a student?)

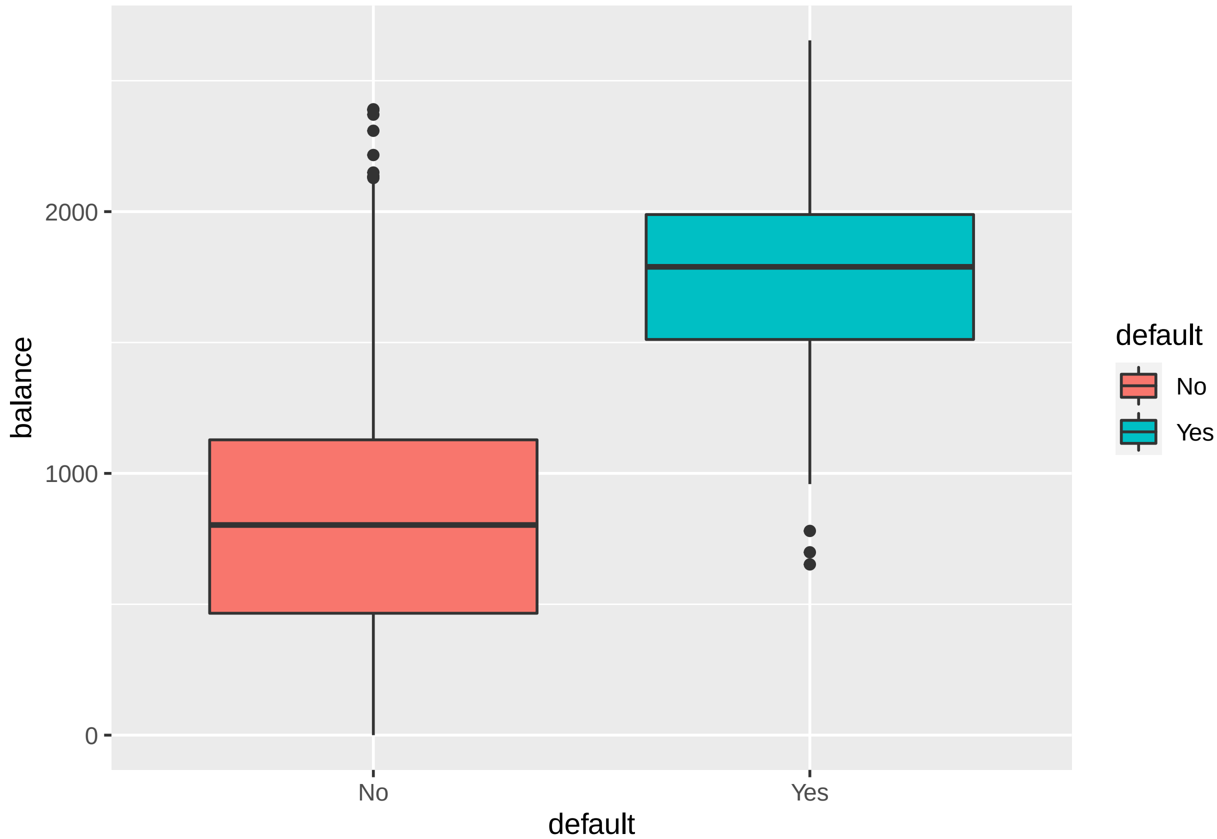
0. How many observations does this dataset have? How many defaults and non-defaults? How many defaults by student status?

```
## [1] 10000     4
```

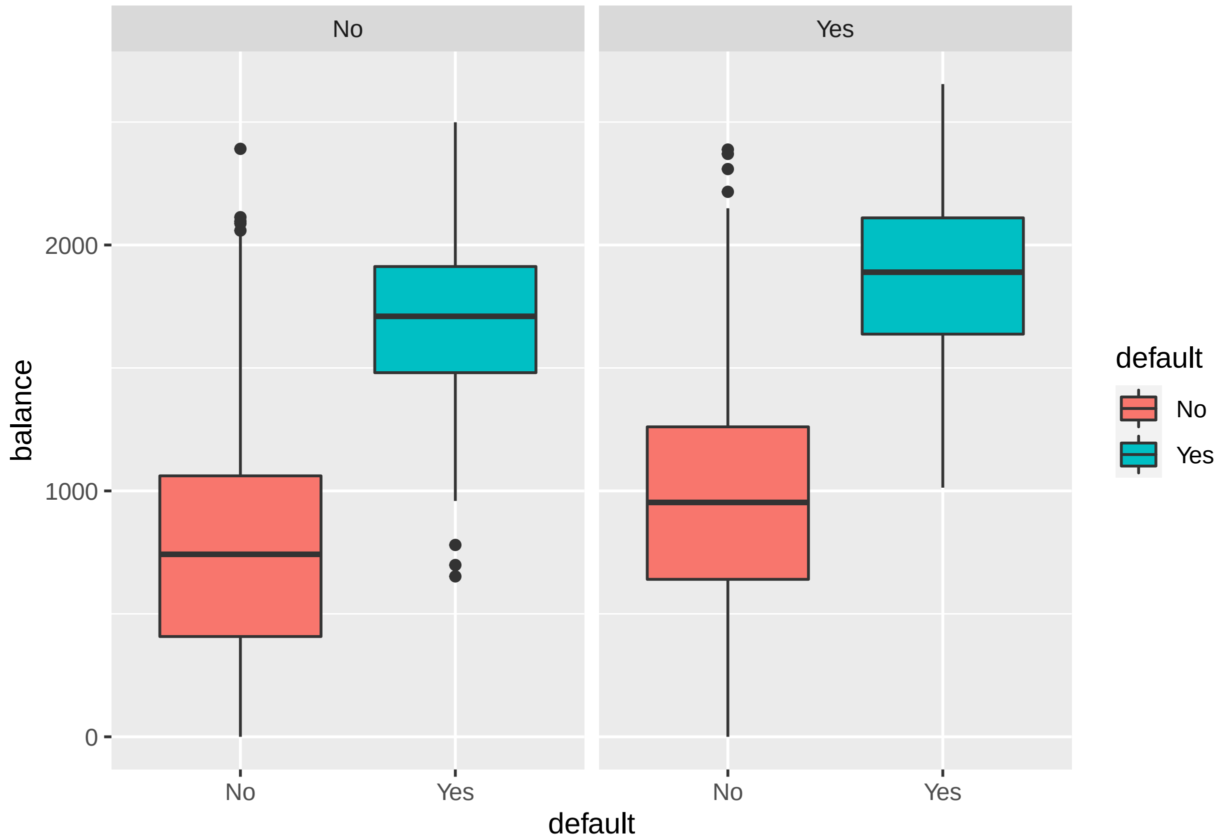
```
##
##   No  Yes
## 9667  333
```

```
##
##          No  Yes
##   No   6850 2817
##   Yes   206  127
```

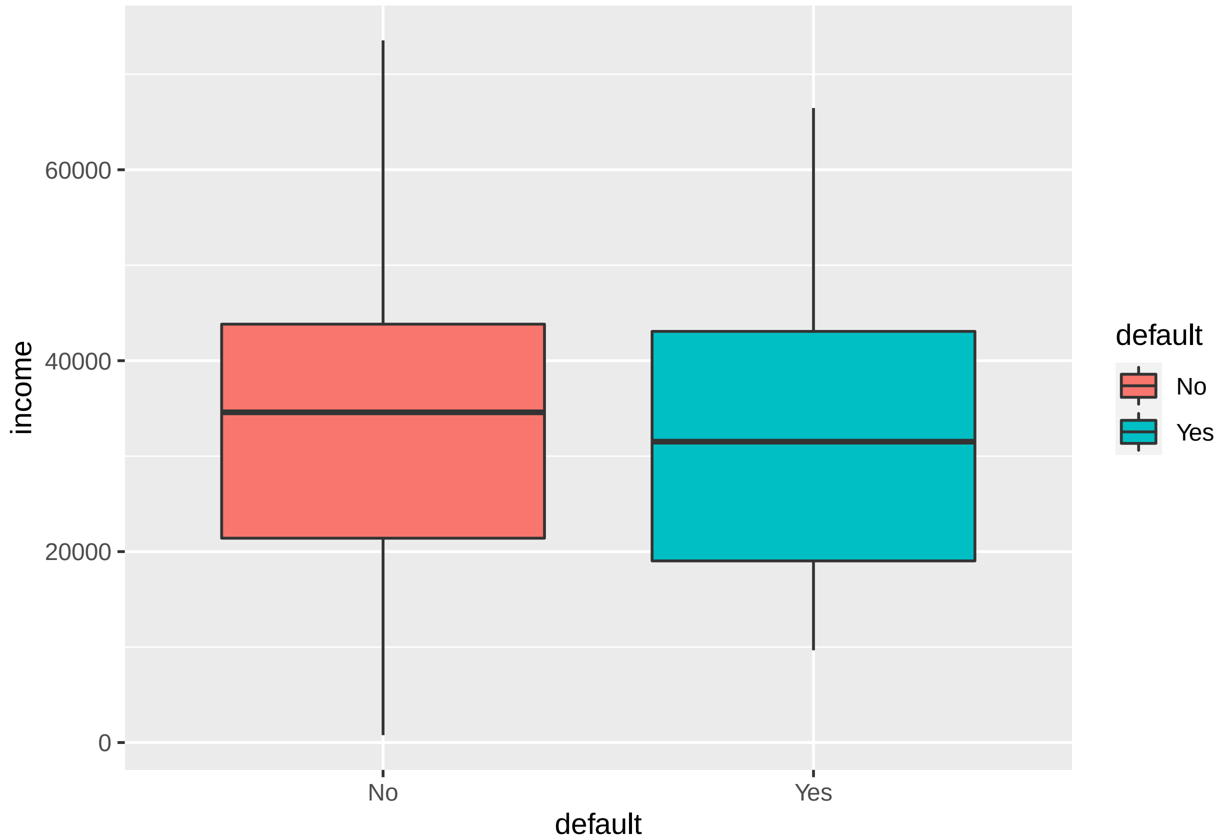
1. Create a boxplot of `balance` across people who default or not. What do you observe? What do you observe when you facet the boxplot by `student`?

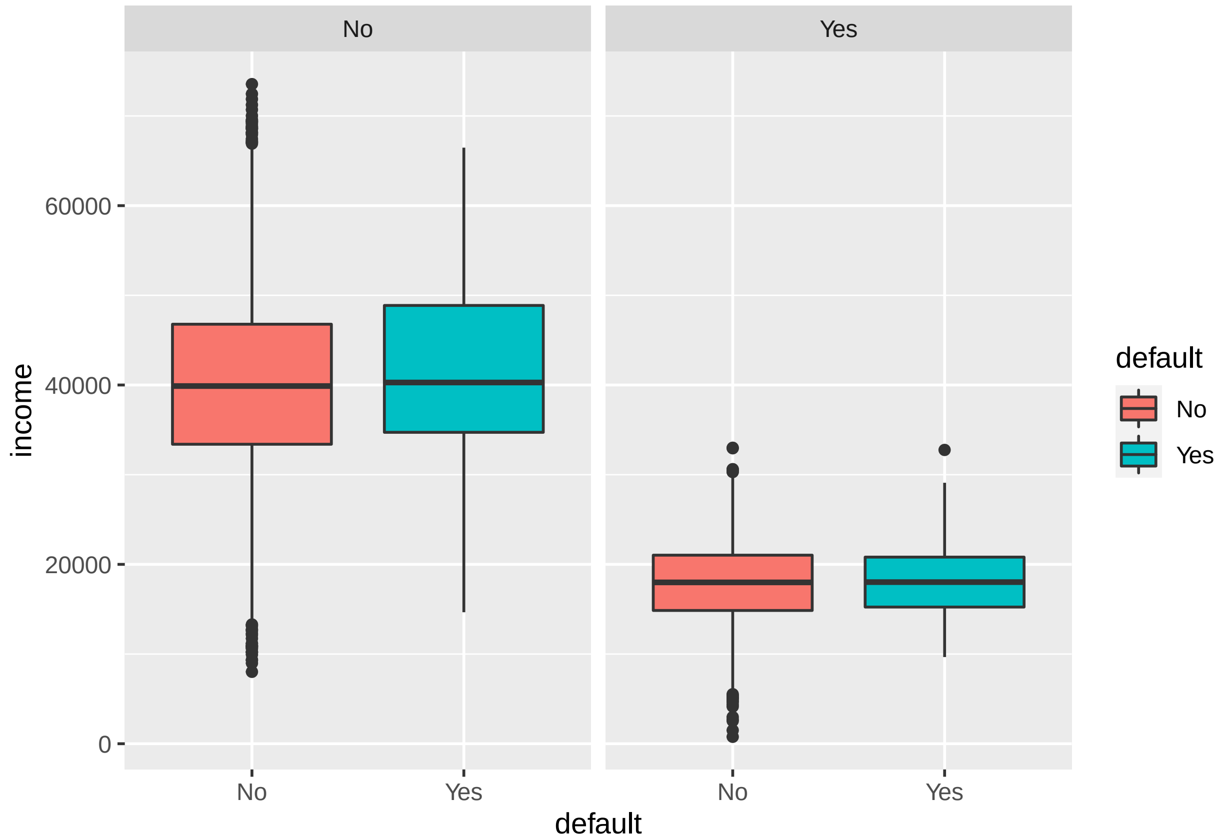


2000
1000
0
balance
No
Yes
default
default
No
Yes



2. How about the effect of `income` on defaulting? Does it change according to `student` status?


income
60000
40000
20000
0
No
Yes
default
default
No
Yes



3. Load `tidymodels` and create a training dataset using 8000 observations and a testing dataset using 2000 observations. Make sure to call your training dataset `default.train.tbl` and your testing dataset `default.test.tbl`

```
set.seed(12345)
```

Modeling probabilities with linear models

Let's start by trying to predict the default using a linear model whose input variable is `balance`, that is

$$y = \alpha_0 + \alpha_1 \times balance$$

In homework 3 we learned that using linear models in this setting creates a number of issues for classification, among them the fact that we are not guaranteed that the prediction will correspond to a probability (a number between `0` and `1`)

One way to overcome this is to use the log odds on our response variable. The log odds y of an event with probability p is defined as

$$\tilde{y} := \log\left(\frac{p}{1-p}\right)$$

Hence, if $\tilde{y}$ represents the log odds, we can invert this expression to get the probability.

$$p = \frac{e^{\tilde{y}}}{1+e^{\tilde{y}}} = \frac{1}{1+e^{-\tilde{y}}}$$

Logistic regression using `tidymodels`

Fortunately for us, logistic models are readily available in R. We can create such a model on the training dataset using the following code:

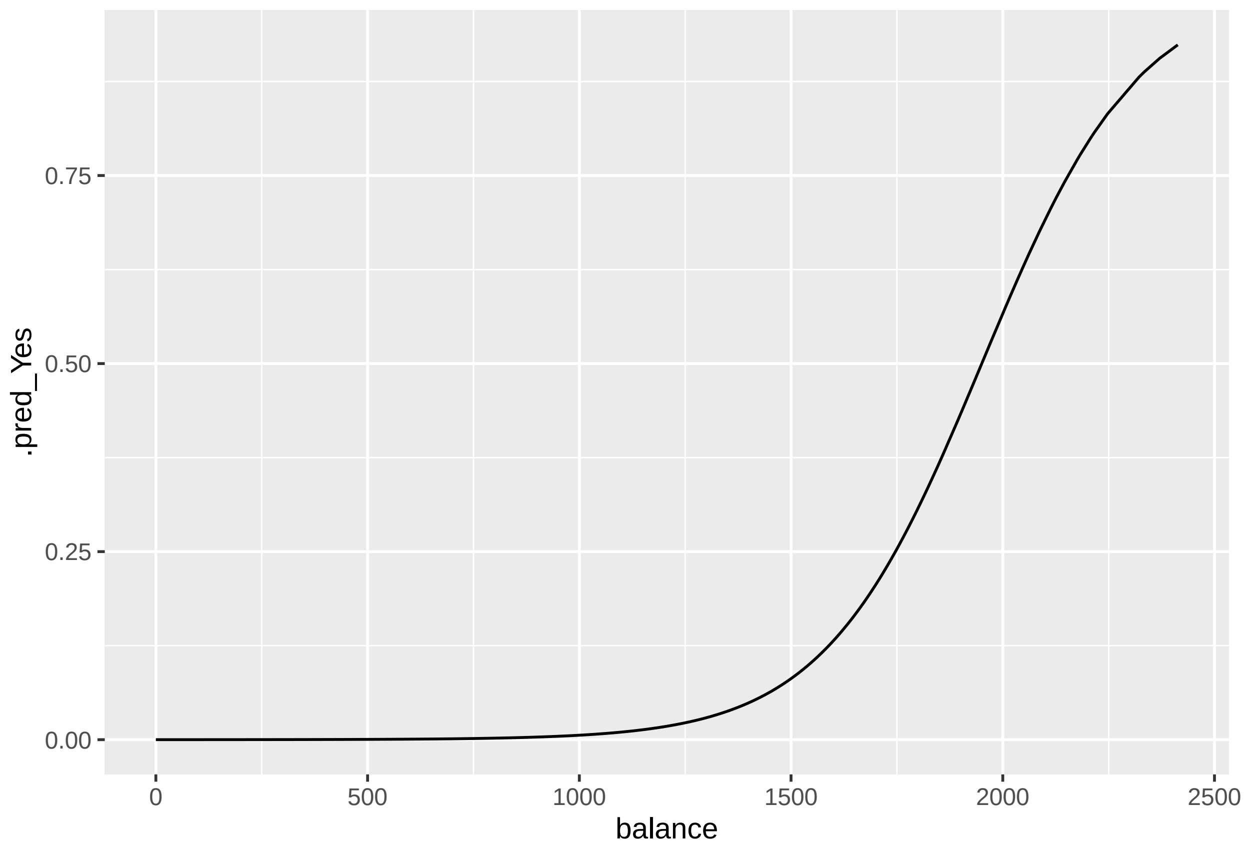
```
logit.model <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

default.recipe <-
  recipe(default ~ balance, data=default.train.tbl)

logit.wflow <- workflow() %>%
  add_recipe(default.recipe) %>%
  add_model(logit.model)

logit.fit <- fit(logit.wflow, default.train.tbl)
```

Finally we can predict the probability of defaulting on the testing dataset or simply predict whether someone would go on default or not

```
## # A tibble: 2,000 x 2
##    .pred_No .pred_Yes
##       <dbl>     <dbl>
##  1    0.996 0.00386
##  2    1.00  0.0000273
##  3    0.981 0.0192
##  4    1.00  0.0000980
##  5    1.00  0.000128
##  6    1.00  0.0000273
##  7    1.00  0.000469
##  8    0.990 0.00988
##  9    0.999 0.000873
## 10    1.00  0.000393
## # ... with 1,990 more rows
```

```
## # A tibble: 2,000 x 1
##    .pred_class
##    <fct>
##  1 No
##  2 No
##  3 No
##  4 No
##  5 No
##  6 No
##  7 No
##  8 No
##  9 No
## 10 No
## # ... with 1,990 more rows
```

4. Plot the predicted probability of defaulting (using the logit model) as a function of `balance`.



5. How many observations are predicted to default in your testing dataset? How many of your predictions are wrong?

```
## [1] 0.026
```

5. `yardstick` allows you to evaluate the performance of your classification model in many different ways. Consult https://www.tmwr.org/performance.html#binary-classification-metrics and do the following:

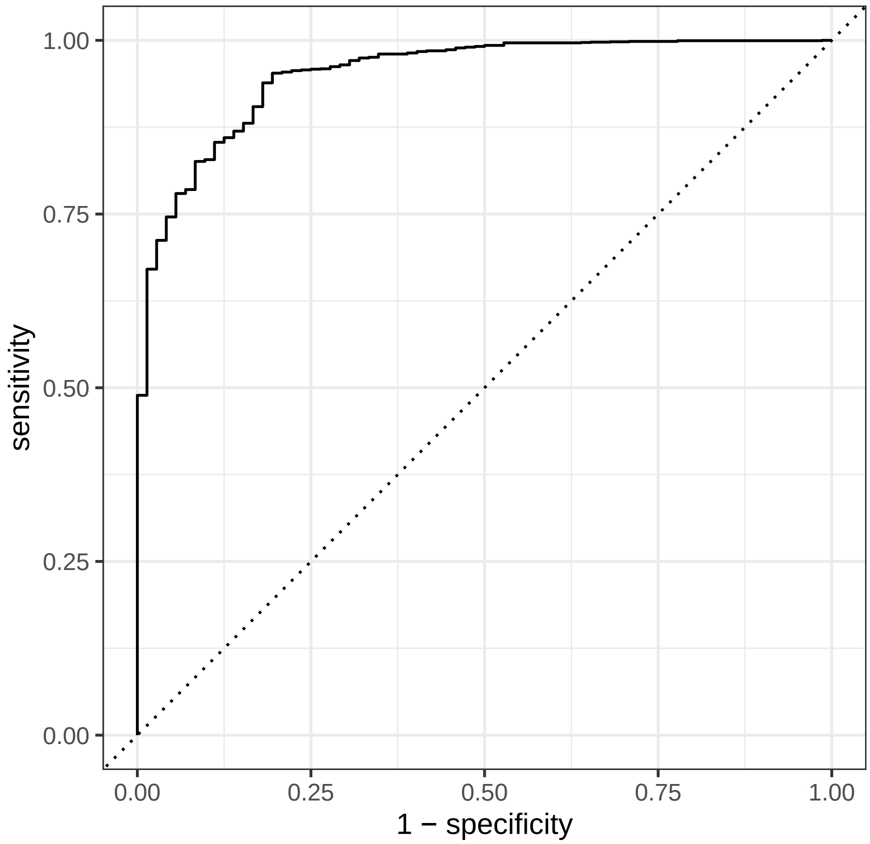
a. Calculate the confusion matrix of your model. Is your model making more errors on people that go on default or not?

```
##           Truth
## Prediction   No  Yes
##        No  1923   47
##        Yes    5   25
```

b. Define how `accuracy` is defined and calculate it for your model.

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.974
```

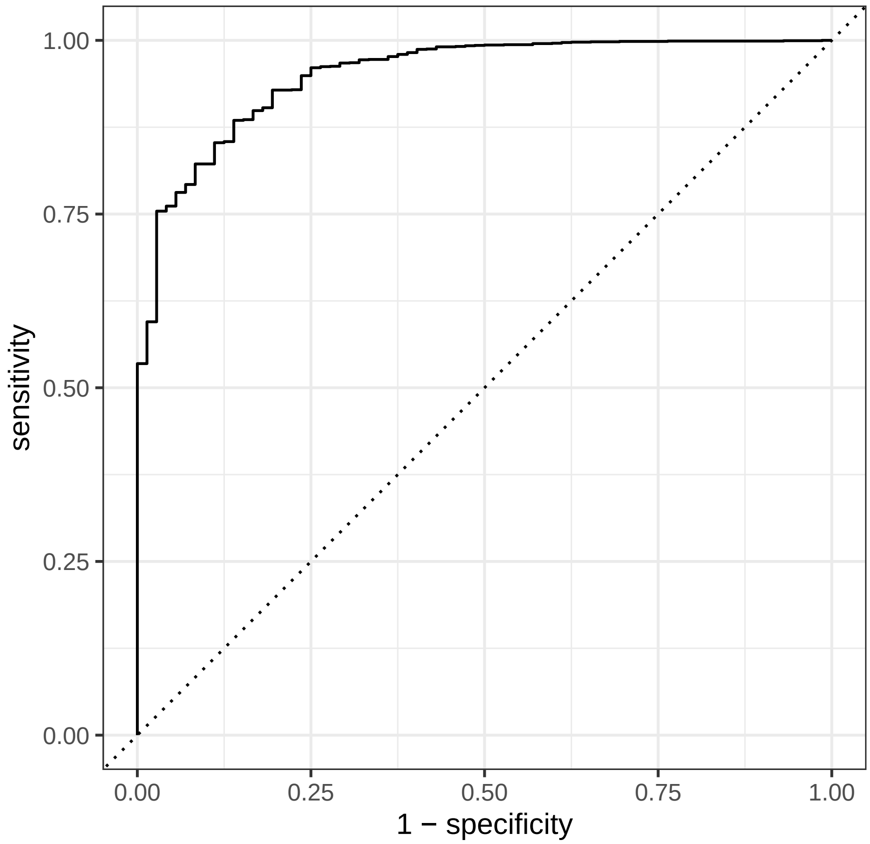
c. Define `specificity`, `sensitivity`, `ROC`, and `AUC`. Plot the ROC curve and calculate the AUC of your model




```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.951
```

7. (*Optional*) Calculate the AUC and plot the ROC for the logistic regression model that takes into account `balance`, `income` and `student`.

```
##           Truth
## Prediction   No  Yes
##        No  1924   47
##        Yes    4   25
```




```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.950
```