# Getting Started

Final Project - STAT 362 - Sp20

## The Trivial Model

Let's take a look at the trivial model, which assigns any predicted probabilities the proportion of the defaulting cases in the training set. So the predicted probabilities will always be:

```
train %>% group_by(default) %>% tally() %>% mutate(proportion = n/sum(n)) %>% .[2,c(1,3)]
```

```
## # A tibble: 1 x 2
##   default proportion
##   <fct>        <dbl>
## 1 1            0.208
```

To evaluate the performance of the trivial model, we can see that it will predict every case as the non-defaulting one. Hence, the misclassification for the trivial model is the proportion of defaulting cases in the test set. I will use the competition set. You can estimate your model's performance through your own testing sets.

```
compete %>% group_by(default) %>% tally() %>% mutate(proportion = n/sum(n)) %>% .[2,c(1,3)]
```

```
## # A tibble: 1 x 2
##   default proportion
##   <fct>        <dbl>
## 1 1            0.206
```

So we see the trivial model has a misclassification rate of .206.

## The Log Loss Metric

We can also evaluate the trivial model by the log loss metric, which can be calculated as

```
pred_trivial <- train %>% group_by(default) %>%
  tally() %>% mutate(proportion = n/sum(n)) %>%
  .[2,c(1,3)] %>% pull()
#to convert the factor default into a numeric to make it easier on the next line
compete <- compete %>% mutate(default_num = as.numeric(as.character(default)))
compete %>%
  summarise(log_loss = -mean(default_num*log(pred_trivial) + (1-default_num)*log(1-pred_trivial)))
```

```
## # A tibble: 1 x 1
##   log_loss
##      <dbl>
## 1    0.509
```