Midterm2

Langchen Liu

2024-04-02

Load Data and pre-processing

```
load('data/labeled_points.Rdata')

labeled = labeled %>%
  select(ID, landcover)

d = labeled_train %>%
  left_join(labeled, by = 'ID')

d = d %>%
  mutate(veg = ifelse(landcover %in% c('natforest', 'orchard', 'cropland'),
                      1, 0),
         NDVI100 = NDVI * 100)
head(d,2)
```

```
## # A tibble: 2 x 22
##      B1    B2    B3    B4    B5 B6_VCID_1 B6_VCID_2    B7    B8   lat   lon
##   <dbl> <dbl> <dbl> <dbl> <dbl>     <dbl>     <dbl> <dbl> <dbl> <dbl> <dbl>
## 1    66    60    73    68    93       146       176    76    69  11.2  3.41
## 2    64    59    69    72    89       147       178    72    69  11.2  3.41
## # i 11 more variables: year <int>, month <int>, day <int>, date <date>,
## #   ID <int>, NDVI <dbl>, NDBI <dbl>, EVI <dbl>, landcover <chr>, veg <dbl>,
## #   NDVI100 <dbl>
```

```
dm = d %>%
  group_by(ID) %>%
  summarise(
    mean.NDVI100 = mean(NDVI100, na.rm = T),
    landcover = unique(landcover),
    veg = unique(veg)) %>%
  as.data.frame()
head(dm,2)
```

```
##     ID mean.NDVI100 landcover veg
## 1 2043     15.06720   orchard   1
## 2 2069     13.42962   orchard   1
```

Fit a logistic model

```
m1 = glm(veg ~ mean.NDVI100,
        data = dm,
        family = binomial(link = logit))
summary(m1)
```

```
##
## Call:
## glm(formula = veg ~ mean.NDVI100, family = binomial(link = logit),
##     data = dm)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.1318     0.2599  -0.507    0.612
## mean.NDVI100   0.7953     0.1029   7.729 1.09e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 449.87  on 399  degrees of freedom
## Residual deviance: 105.95  on 398  degrees of freedom
## AIC: 109.95
##
## Number of Fisher Scoring iterations: 8
```

a. Data exploration

Find another predictor involving NDVI (other than mean(NDVI100)) that might be useful in predicting vegetation.

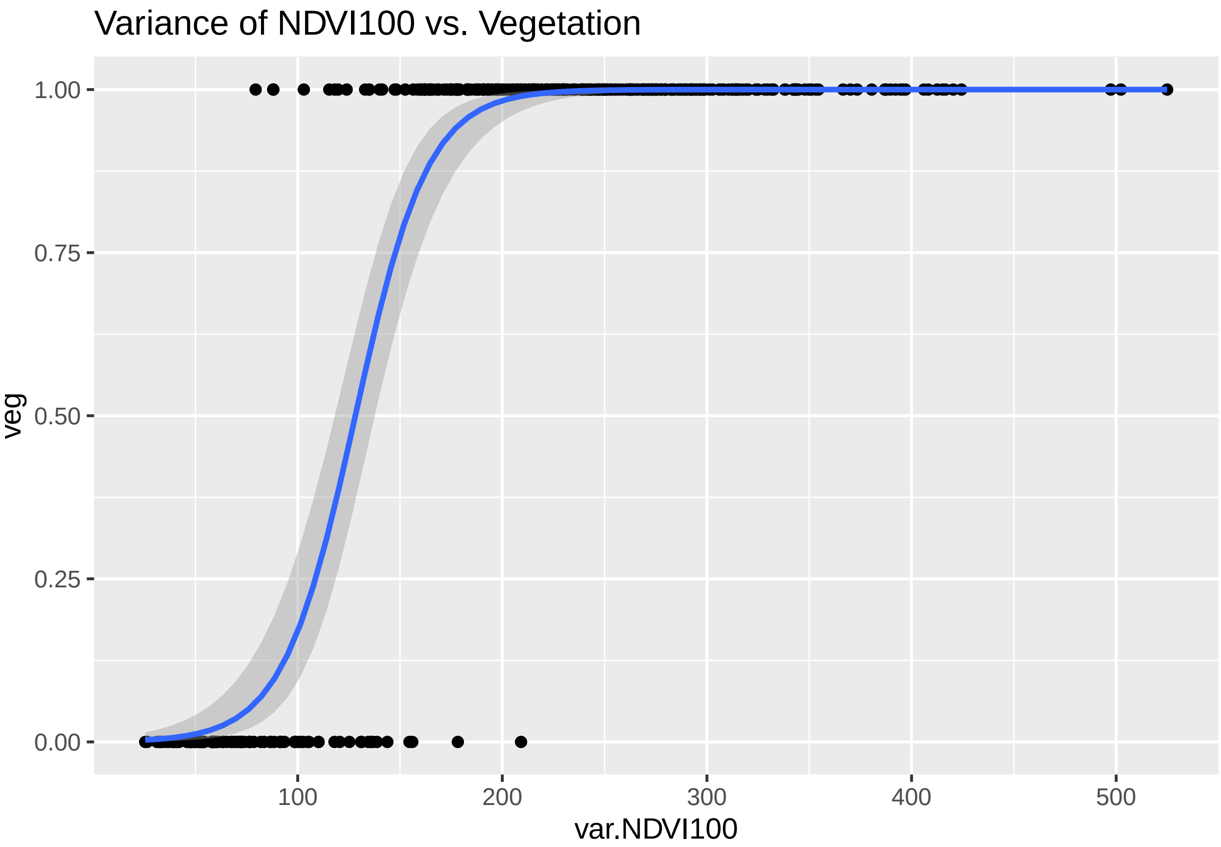
I might begin with variance(NDVI100)

```
dm = d %>%
  group_by(ID) %>%
  summarise(
    mean.NDVI100 = mean(NDVI100, na.rm = T),
    var.NDVI100 = var(NDVI100, na.rm = T),
    landcover = unique(landcover),
    veg = unique(veg)) %>%
  as.data.frame()

### plot the variance and make a smooth curve

ggplot(dm, aes(x = var.NDVI100, y = veg)) +
  geom_point() +
  geom_smooth(method = 'glm',
              method.args = list(family = 'binomial')) +
  labs(title = 'Variance of NDVI100 vs. Vegetation')
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



From the plot, we can see that the variance of NDVI100 is a good predictor of vegetation. The reason is veg = 0 is most likely to have a smaller variance in NDVI100, while the veg = 1 points have a significantly greater variance in NDVI100.

b. Fit a logistic model

```
m2 = glm(veg ~ var.NDVI100 + mean.NDVI100,
        data = dm,
        family = binomial(link = logit))

summary(m2)
```

```
##
## Call:
## glm(formula = veg ~ var.NDVI100 + mean.NDVI100, family = binomial(link = logit),
##     data = dm)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.453135   1.140102  -3.906 9.39e-05 ***
## var.NDVI100   0.033005   0.008608   3.834 0.000126 ***
## mean.NDVI100  0.404484   0.132986   3.042 0.002354 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 449.868  on 399  degrees of freedom
## Residual deviance:  84.101  on 397  degrees of freedom
## AIC: 90.101
##
## Number of Fisher Scoring iterations: 8
```

```r
## predict and compare the accuracy

dm$predm2 = predict(m2, type = 'response')
dm$predm2 = ifelse(dm$predm2 > 0.5, 1, 0)


dm$predm1 = predict(m1, type = 'response')
dm$predm1 = ifelse(dm$predm1 > 0.5, 1, 0)

cat('Accuracy of m1:', mean(dm$predm1 == dm$veg), '\n')
```

```
## Accuracy of m1: 0.945
```

```r
cat('Accuracy of m2:', mean(dm$predm2 == dm$veg), '\n')
```

```
## Accuracy of m2: 0.96
```

```r
## ROC curve

roc1 = roc(dm$veg, dm$predm1)
```

```
## Setting levels: control = 0, case = 1
```

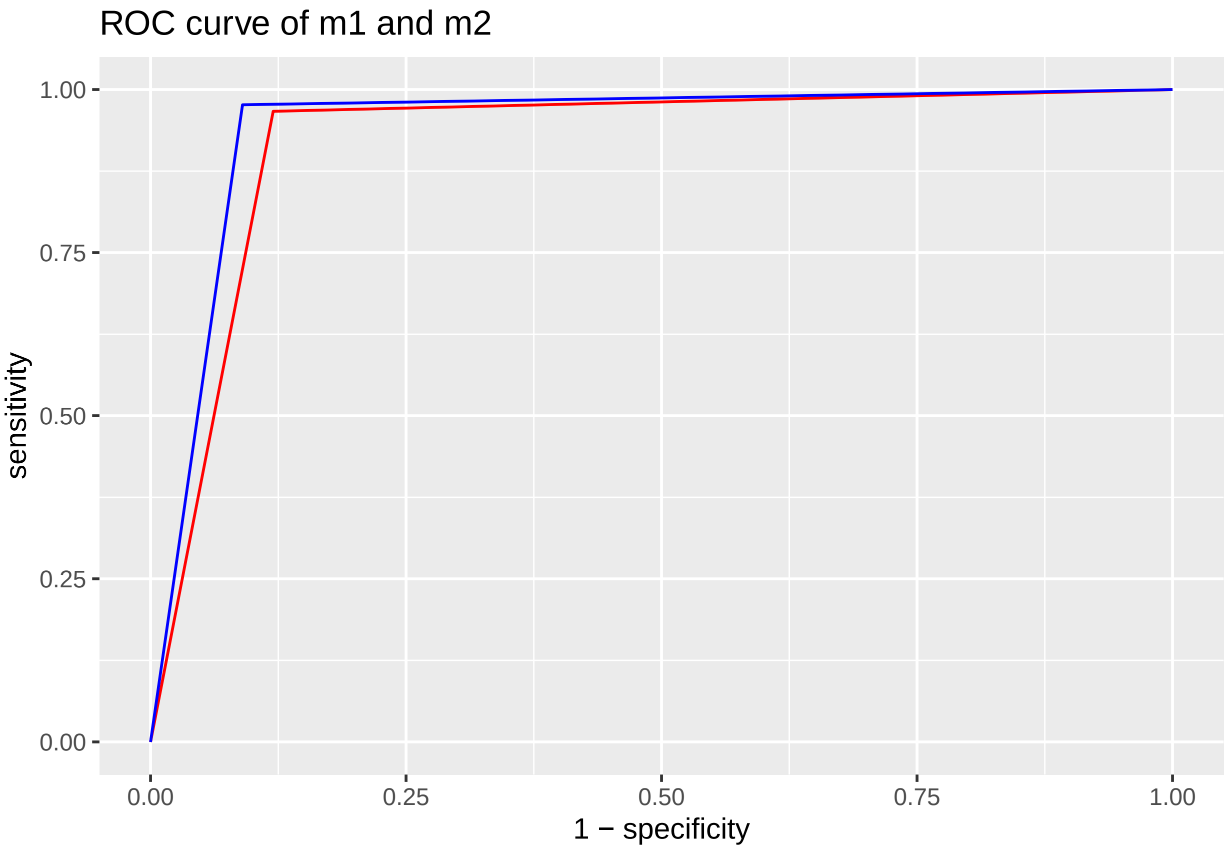
```
## Setting direction: controls < cases
```

```r
roc2 = roc(dm$veg, dm$predm2)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
roc1 = roc1 %>% coords()
roc2 = roc2 %>% coords()

ggplot() +
  geom_line(data = roc1, aes(x = 1 - specificity, y = sensitivity), color = 'red') +
  geom_line(data = roc2, aes(x = 1 - specificity, y = sensitivity), color = 'blue') +
  labs(title = 'ROC curve of m1 and m2')
```



We can see that the new predictor (var.NDVI100) significantly improves the previous model.

First, the intercept and the coefficients of the new model has a good significant level, while the intercept of the previous model is not as significant.

Second, both model share the same null deviance but the new model has a much smaller residual deviance, this means that the new model has a greater difference in residual and null deviance, which means the new model explains the statistical relationship of the predictors and the response variables better.

Third, the AIC of the new model is smaller than the previous model, which means the new model has a better fit.

Fourth, the accuracy of the new model (0.96) is higher than the previous model (0.945).

Fifth, the ROC curve of the new model is closer to the top-left corner, which means the new model has a better performance.

As a result, the new model is better than the previous model. I will recommend adding this new predictor (var.NDVI100) to the model.