Prerequisite

All code and figures are contained in *https://github.com/feedlord18/CSDS313.git*

Problem 1

The initial determining variables has 4 of them selected, which are 33, 768, 902, and 261, those are highlighted in bold. The top index and respective coefficients are:

Index	p-Value
33	0.3877549
768	0.3599364
625	0.3137587
902	0.3109434
610	0.2937186
379	0.2933485
261	0.2863394
965	0.2788449
62	0.2751364
43	0.2704340

Table 1: Top 10 Variables Ranked by Pearson Correlation

From the results are can see there are 6 variables which the multiple linear regression believes is significant. 4 of the determining variables are selected, which again matches our conclusion from above: V1 (33), V2 (768). V4(902). V7(261).

```
Residuals:
    Min      1Q  Median      3Q     Max
-5.4918 -1.2588 -0.0111  1.3621  3.7290

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.2040     0.2091  -0.976 0.331781
V1            1.0435     0.1888   5.528 3.20e-07 ***
V2            0.9632     0.2172   4.434 2.63e-05 ***
V3            0.6695     0.2466   2.715 0.007955 **
V4            0.5563     0.2029   2.741 0.007398 **
V5            0.3439     0.1918   1.793 0.076387 .
V6            0.5368     0.1990   2.698 0.008342 **
V7            0.7078     0.1892   3.740 0.000325 ***
V8            0.3241     0.2277   1.424 0.158023
V9            0.3280     0.2081   1.576 0.118570
V10          -0.3906     0.2108  -1.853 0.067176 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.943 on 89 degrees of freedom
Multiple R-squared:  0.6279,    Adjusted R-squared:  0.5861
F-statistic: 15.02 on 10 and 89 DF,  p-value: 2.601e-15
```

We perform lasso regression with $\alpha = 1$ and $\lambda = 0.1, 1, 10$. All variables when $\lambda = 10$ except V1 (33), and V2 (768) are dropped. When $\lambda = 0.1, 1$ all variables are determined to be significant which contributes to the outcome. However we see the initial determining variables yield the largest coefficients.

```
> coef(model_0.1)
11 x 1 sparse Matrix of class "dgCMatrix"
                    s0
(Intercept) -0.2066421
V1           1.0009602
V2           0.9263280
V3           0.6516628
V4           0.5427971
V5           0.3412275
V6           0.5184410
V7           0.6778745
V8           0.3200639
V9           0.3253496
V10         -0.3818272
```

```
> coef(model_1)
11 x 1 sparse Matrix of class "dgCMatrix"
                    s0
(Intercept) -0.2281232
V1           0.7197942
V2           0.6792222
V3           0.5118164
V4           0.4309196
V5           0.2971260
V6           0.3900093
V7           0.4787636
V8           0.2766610
V9           0.2805805
V10         -0.3047672
```

```
> coef(model_10)
11 x 1 sparse Matrix of class "dgCMatrix"
                    s0
(Intercept) -0.30311299
V1           0.03814372
V2           0.02266347
V3           .
V4           .
V5           .
V6           .
V7           .
V8           .
V9           .
V10          .
```

We apply discretization on all feature values and below are top 10 variable indexes ranked according to their concordance scores. We see out of the 10 features, all 5 initial variables are within the list. The determining values are highlighted in bold.

Index	concordance-Scores
768	0.70
425	0.65
33	0.64
102	0.64
902	0.64
334	0.63
716	0.63
906	0.63
261	0.62
308	0.62

Table 2: Top 10 Variables Ranked by Concordance Scores

Naive Bayes

Below, we create the naive Bayesian conditional probability table. We only need to record for one binary variable because 0/1 are complementary. Therefore with a binary feature, we only need one variable to record the CPT.

	Y = 0	Y = 1
$x_{768} = 1$	18	36
$x_{425} = 1$	14	327
$x_{33} = 1$	19	31
$x_{102} = 1$	18	30
$x_{902} = 1$	17	29
$x_{334} = 1$	21	32
$x_{716} = 1$	19	30
$x_{906} = 1$	20	31
$x_{261} = 1$	17	27
$x_{308} = 1$	16	26

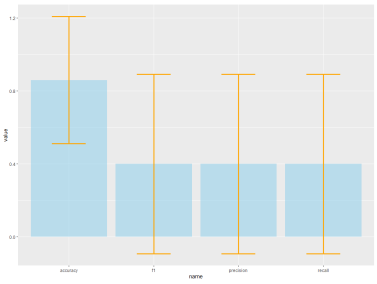
Table 3: NB CPT

Using LOOCV (Leave One Out Cross Validation) we can train our naive Bayesian model on each 99 samples and validate our results on the dropout sample. Therefore we can achieve the following metrics (This is results calculated after all 100 validation steps are finished, I couldn't understand how we would obtain variance because precision and recall and f1 would always be equal in a single validation test):

metrics	value
accuracy	0.86
precision	0.87
recall	0.83
f1	0.85

Table 4: Metrics of 100 Validation Prediction Results

This is the results from average single validation metrics, the results are vastly different from the previous result for precision, recall, and f1:



metrics	value	variance
accuracy	0.86	0.122
precision	0.40	0.242
recall	0.40	0.242
f1	0.40	0.242

We generate 50 test samples from the same mechanism as the training data. Then generate the CPT from all 100 samples of training data and evaluate the model with the test data. From below we can see the ROC curve and Precision-Recall curve which visualizes the performance of the model. The ROC curve demonstrates an AUC larger than 0.5, which suggests it performs better than a random classifier.