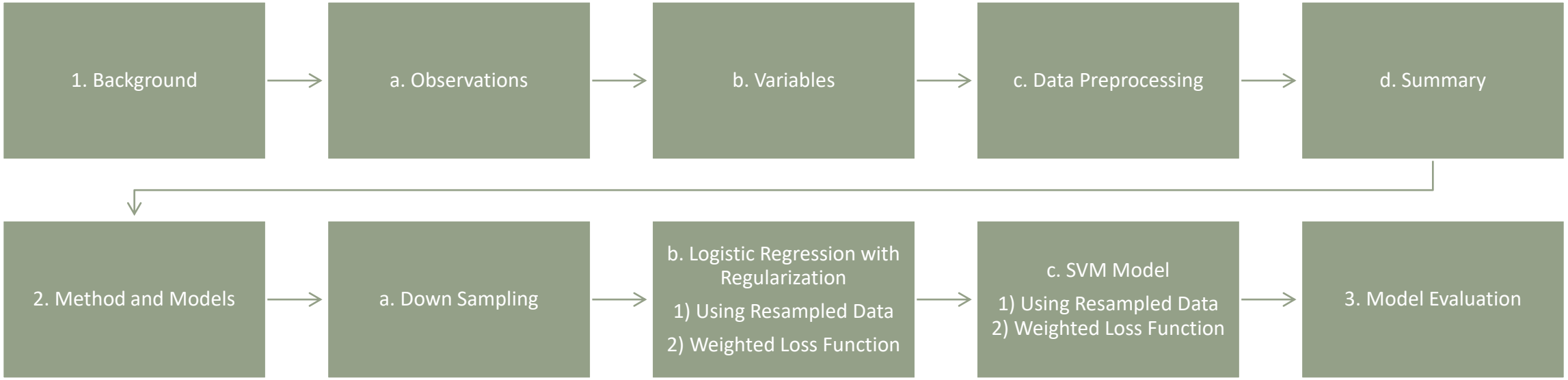
Insurance Cross Sell Prediction

PROFESSOR: YUPING ZHANG
TEAM MEMBER: RUOFAN CHEN


1. Background
a. Observations
b. Variables
c. Data Preprocessing
d. Summary
2. Method and Models
a. Down Sampling
b. Logistic Regression with Regularization
1) Using Resampled Data
2) Weighted Loss Function
c. SVM Model
1) Using Resampled Data
2) Weighted Loss Function
3. Model Evaluation

Contents

Dataset Resource: Health Insurance Cross Sell Prediction-JantaHack

Data link: https://www.kaggle.com/shivan118/crosssell-prediction

Background: An insurance company provides both health insurance and vehicle insurance. This company wants to find out if the customer who already purchased health insurance would be interested in its vehicle insurance.

Background

```
> head(train)
  id Gender Age Driving_License Region_Code Previously_Insured Vehicle_Age
1  1   Male  44               1          28                   0   > 2 Years
2  2   Male  76               1           3                   0    1-2 Year
3  3   Male  47               1          28                   0   > 2 Years
4  4   Male  21               1          11                   1    < 1 Year
5  5 Female  29               1          41                   1    < 1 Year
6  6 Female  24               1          33                   0    < 1 Year
  Vehicle_Damage Annual_Premium Policy_Sales_Channel Vintage Response
1            Yes          40454                  26     217        1
2             No          33536                  26     183        0
3            Yes          38294                  26      27        1
4             No          28619                 152     203        0
5             No          27496                 152      39        0
6            Yes           2630                 160     176        0
```

Observations

381,109 observations, 12 columns, no N/A.
Response variable: Response
First 6 rows of original dataset

- **Id**: Unique ID for the customer
- **Gender**: Gender of the customer
- **Age**: Age of the customer
- **Driving_License**: 0 (Customer does not have DL) 1 (Customer already has DL)
- **Region_Code**: Unique code for the region of the customer
- **Previously_Insured**:1 (Customer already has Vehicle Insurance) ,0 (Customer doesn't have Vehicle Insurance)
- **Vehicle_Age**: Age of the Vehicle
- **Vehicle_Damage**: 1 (Customer got his/her vehicle damaged in the past), 0 (Customer didn't get his/her vehicle damaged in the past)
- **Annual_Premium**: The amount customer needs to pay as premium in the year
- **Policy_Sales_Channel**: Anonymized Code for the channel of outreaching to the customer ie. Different Agents, Over Mail, Over Phone, In Person, etc.
- **Vintage**: Number of Days, Customer has been associated with the company
- **Response**: 1 (Customer is interested), 0 (Customer is not interested)

Variables

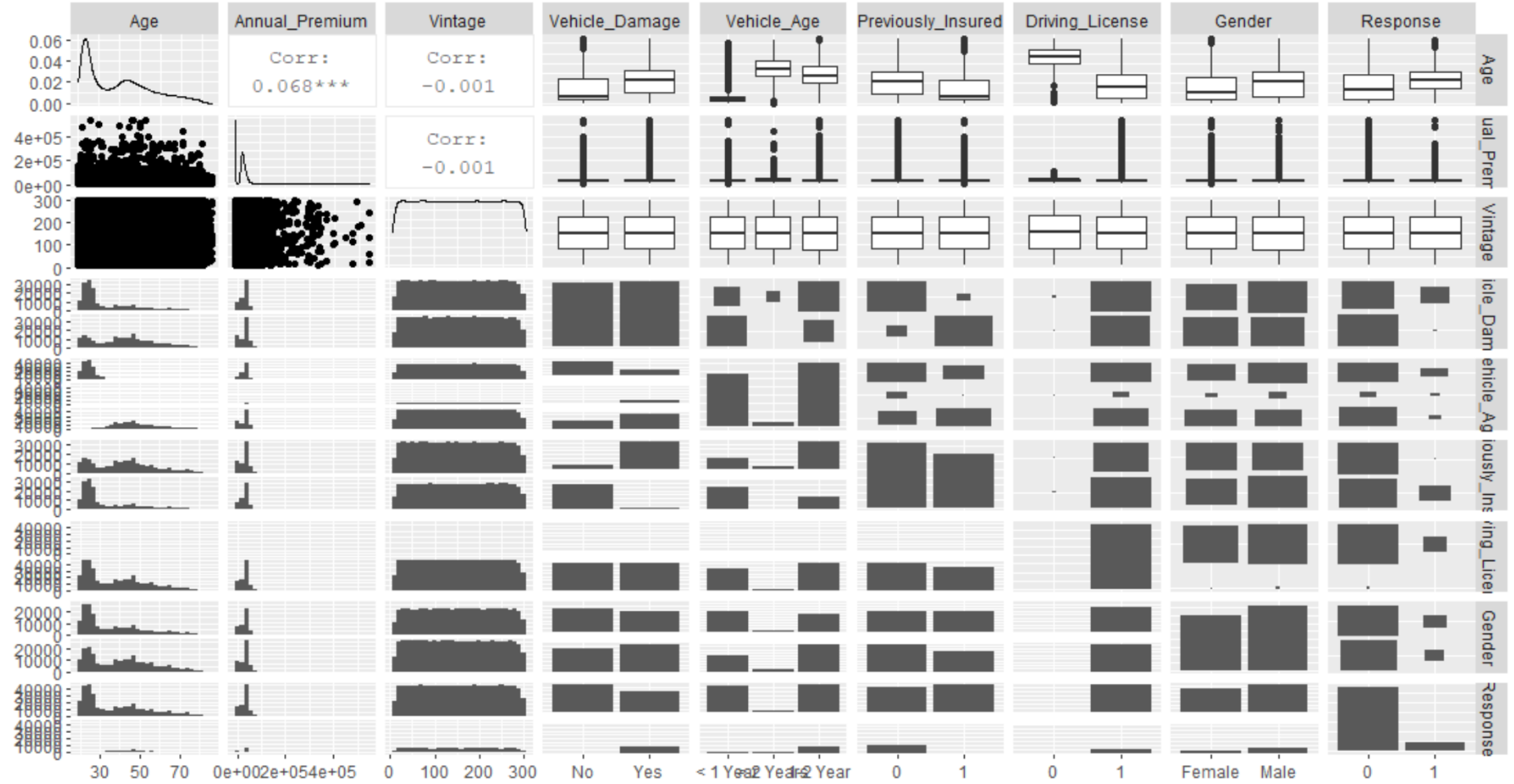
Set categorical variables **‘Gender’, ‘Diving_License’, ‘Region_Code’, ‘Previously_Insured’, ‘Vehicle_Damage’, ‘Policy_Sales_Channel’, ‘Response’** as factor.

Data Preprocessing- Part 1

```
> summary(train1)
      Age          Annual_Premium      Vintage       Policy_Sales_Channel
 Min.   :20.00   Min.   :  2630   Min.   : 10.0   152    :134784
 1st Qu.:25.00   1st Qu.: 24405   1st Qu.: 82.0   26     : 79700
 Median :36.00   Median : 31669   Median :154.0   124    : 73995
 Mean   :38.82   Mean   : 30564   Mean   :154.3   160    : 21779
 3rd Qu.:49.00   3rd Qu.: 39400   3rd Qu.:227.0   156    : 10661
 Max.   :85.00   Max.   :540165   Max.   :299.0   122    :  9930
                                                  (Other): 50260
 Vehicle_Damage    Vehicle_Age     Previously_Insured  Region_Code
 No :188696     < 1 Year :164786   0:206481           28     :106415
 Yes:192413     > 2 Years: 16007   1:174628           8      : 33877
                1-2 Year :200316                      46     : 19749
                                                      41     : 18263
                                                      15     : 13308
                                                      30     : 12191
                                                      (Other):177306

 Driving_License    Gender        Response
 0:   812        Female:175020   0:334399
 1:380297        Male  :206089   1: 46710
```

Summary


Age
Annual_Premium
Vintage
Vehicle_Damage
Vehicle_Age
Previously_Insured
Driving_License
Gender
Response
Corr:
0.068***
Corr:
-0.001
Corr:
-0.001
No
Yes
< 1 Year
1-2 Year
Female
Male

Using ‘nlevels’ function to find how many levels a categorical variable have.

```
> nlevels(train1$Region_Code)
[1] 53
```

```
> nlevels(train1$Policy_Sales_Channel)
[1] 155
```

The ‘Region_Code’ has 53 levels,

‘Policy_Sale_Channels’ has 155 levels

Dropped levels with small number of observations and create a new level, named ‘200’.

Set variable ‘Response’ with No:0 Yes:1

Data Preprocessing-
Part 2

```
> summary(train1)
      Age          Annual_Premium       Vintage       Policy_Sales_Channel Vehicle_Damage
 Min.   :20.00   Min.   :  2630   Min.   : 10.0   26 : 79700           No :188696
 1st Qu.:25.00   1st Qu.: 24405   1st Qu.: 82.0   122:  9930           Yes:192413
 Median :36.00   Median : 31669   Median :154.0   124: 73995
 Mean   :38.82   Mean   : 30564   Mean   :154.3   152:134784
 3rd Qu.:49.00   3rd Qu.: 39400   3rd Qu.:227.0   156: 10661
 Max.   :85.00   Max.   :540165   Max.   :299.0   160: 21779
                                                  200: 50260
   Vehicle_Age     Previously_Insured Region_Code  Driving_License    Gender       Response
 < 1 Year :164786   0:206481           8  : 33877   0:   812        Female:175020   No :334399
 > 2 Years: 16007   1:174628           15 : 13308   1:380297        Male  :206089   Yes: 46710
 1-2 Year :200316                      28 :106415
                                       30 : 12191
                                       41 : 18263
                                       46 : 19749
                                       200:177306
```

Summary

Use ‘downSample’ function in library ‘caret’ to resample the ‘train1’ dataset.

```
down_train <-
    downSample(train1[,1:10],train1$Response,yname='Response’)
```

Now, this data becomes balanced.

```
> table(down_train$Response)

   No   Yes
46710 46710
```

Resampling Technique: Down Sampling

$$p_i = \frac{e^{(\beta_0+\beta_1 x_i)}}{1+e^{(\beta_0+\beta_1 x_i)}} \qquad \text{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_i$$

$$\ln(L) = \sum_{i=1}^{N}\left[\ln(1-p_i) + y_i \ln\left(\frac{p_i}{1-p_i}\right)\right]$$

$$\mathrm{L}_{log} = -\ln(L) = -\sum_{i=1}^{N}\left[-\ln(1+e^{(\beta_0+\beta_1 x_i)}) + y_i\,(\beta_0+\beta_1 x_i)\right]$$

$$L_{log} + \lambda \sum_{j=1}^{p} {\beta_j}^2 \qquad L_{log} + \lambda \sum_{j=1}^{p} |\beta_j|$$

Logistic Regression with Regularization

```
> ctrl <- trainControl(method="cv", number=5,classProbs=TRUE,
+                      summaryFunction = twoClassSummary)
> set.seed(123)
> down_glm_model <- train(down_x, down_y, method = "glmnet",
+                                 trControl = ctrl,metric = "ROC",
+                           tuneGrid = expand.grid(alpha = 0:1,
+                lambda = seq(0,0.001,length=100)))
```

Logistic Regression with Down Sampling Majority Class

Using package ‘caret’, its function ‘train’ to find best penalized parameter on 5-fold cv: alpha=1, lambda=0.0003030303

Obtain model performance:

```
 alpha        lambda         ROC        Sens       Spec       ROCSD
SensSD       SpecSD

 131      1 0.0003030303 0.8434906 0.6326054 0.9508671 0.00171263
0.003566652 0.001457915
```

Logistic Regression with Down Sampling Majority Class

```
Create weights: weights for loss function
> model_weights <- ifelse(train$Response == 0,
+                         (1/table(train$Response)[1]) * 0.5,
+                         (1/table(train$Response)[2]) * 0.5)
```

Similarly find the best penalized parameter:

alpha = 1 and lambda = 0.0003232323

alpha	lambda	ROC	Sens	Spec	ROCSD	SensSD	SpecSD
1	0.0003232323	0.8447601	0.636826	0.9479769	0.0005215345	0.004055134	0.003206841

Logistic Regression with Weighted Loss Function

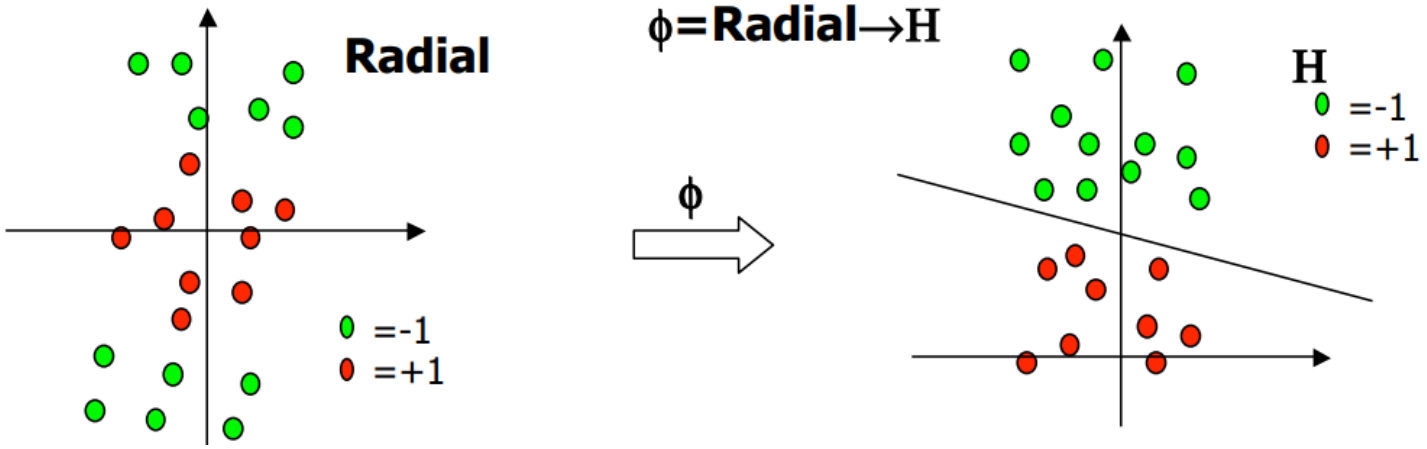
So, the function we end up optimizing is:

$$L_d = \sum a_i - \tfrac{1}{2}\sum a_i a_j y_i y_j K(x_i \bullet x_j), \quad \text{s.t.} \sum_{i=1}^{l} a_i y_i = 0$$

$$0 \le a_i \le C$$

The Kernel trick

Imagine a function ϕ that maps the data into another space:



$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y}+1)^p$$

$$K(\mathbf{x},\mathbf{y}) = \exp\left\{-\|\mathbf{x}-\mathbf{y}\|^2 / 2\sigma^2\right\}$$

$$K(\mathbf{x},\mathbf{y}) = \tanh(\kappa\mathbf{x}\cdot\mathbf{y}-\delta)$$

1st is polynomial (includes x•x as special case)
2nd is radial basis function (gaussians)
3rd is sigmoid (neural net activation function)

$$\hat{\alpha}_1, \ldots \hat{\alpha}_n$$

$$y(x) = sign\left(\sum_{i=1}^{n} \hat{\alpha}_i y_i K(x_i, x)\right)$$

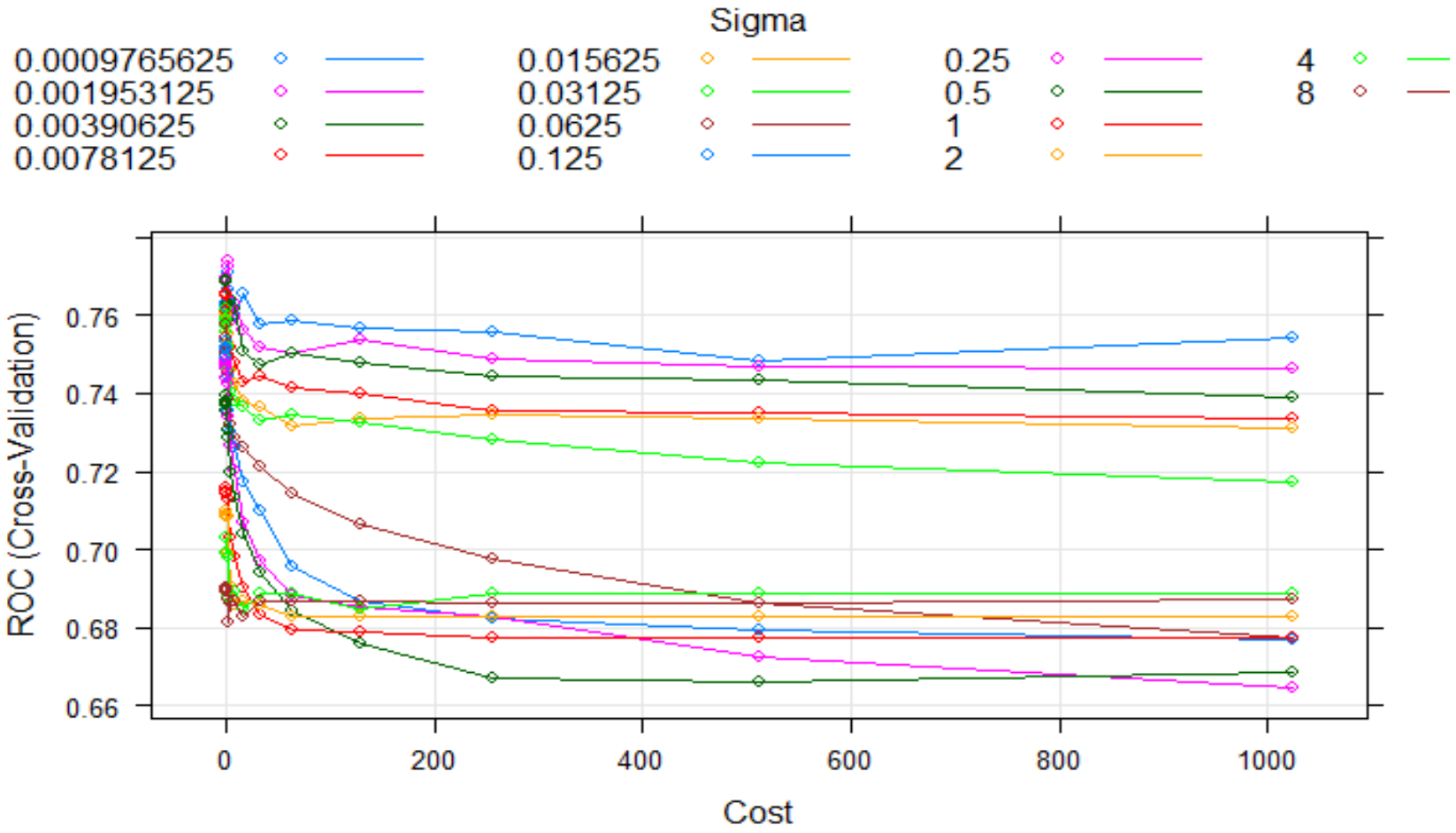
Support Vector Machine

Randomly select 934 observations (1%) from ‘down_train’, using 5-fold cross validation to find penalized parameter, then use these parameters to build a model on whole down sampled data set.

```
myGrid <- expand.grid(C = 2^(-5:10),sigma = 2^(-10:3))
```

```
> down_sample_svm_model_radial$bestTune
          sigma C
   0.001953125 1
```

Treat Time Complexity

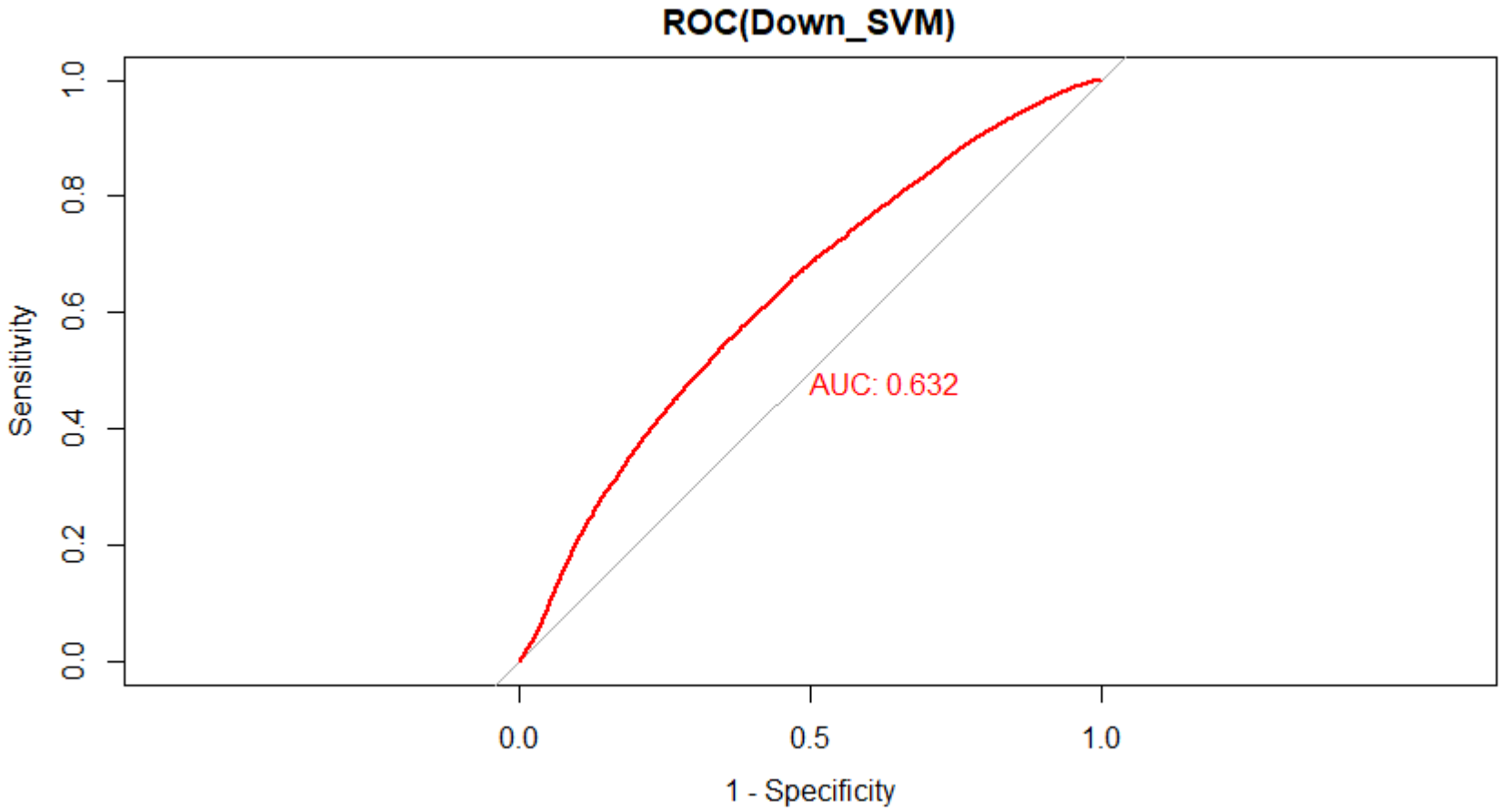


Treat Time Complexity

Split the ‘down_train’ dataset into train set and test set, with proportion of 0.7 and 0.3, respectively.

```
> dim(for_training)
[1] 65393    11
> dim(for_testing)
[1] 28027    11
> down_svm_model <- ksvm(Response ~ .,data = for_training,kernel =
"rbfdot",kpar = list(sigma=0.001953125),C = 1,prob.model = TRUE,
scaled = FALSE)
> down_svm_model
```

SVM with Down Sampling Majority Class



SVM with Down Sampling Majority Class

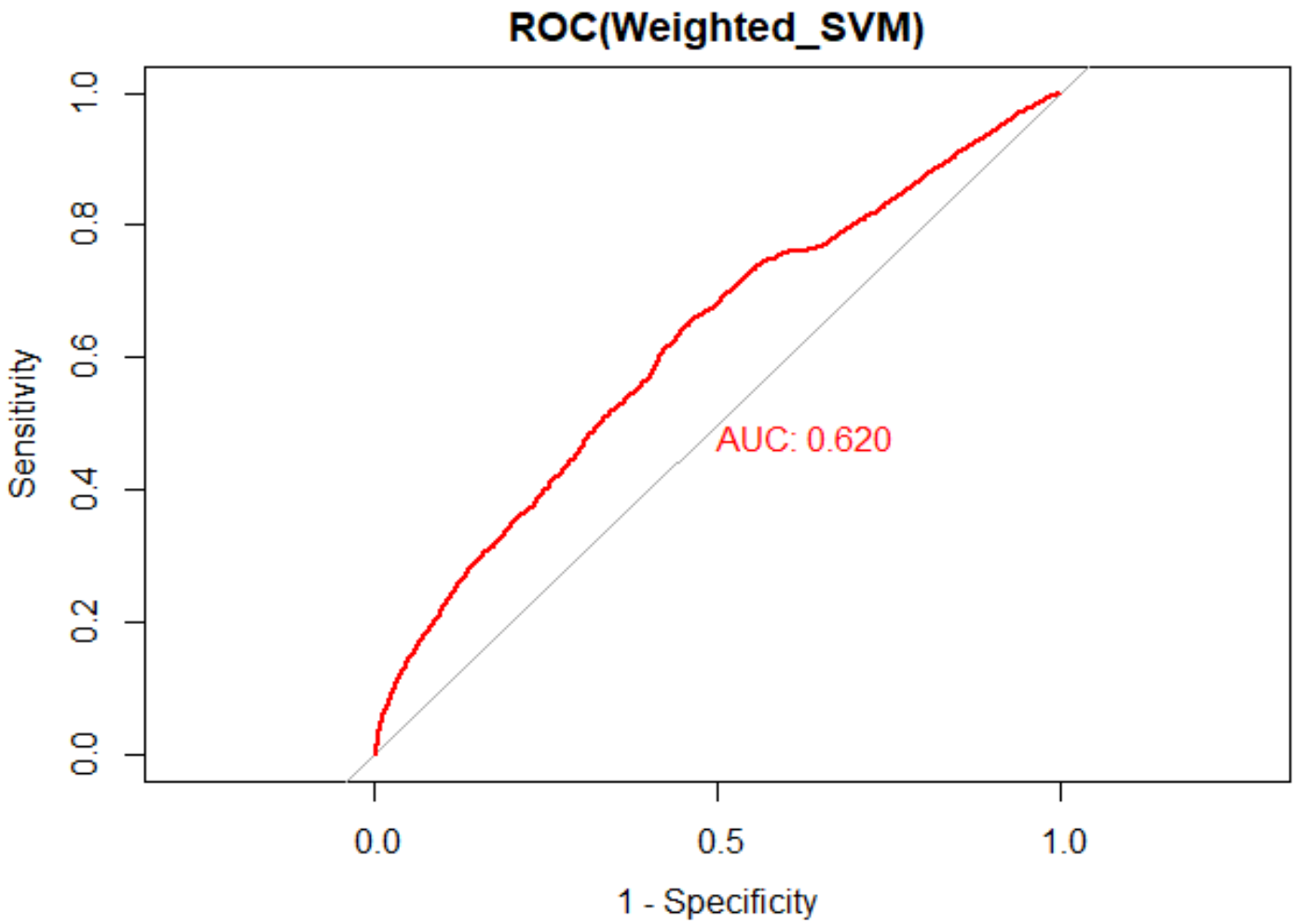
Randomly select 20% of observation in ‘train1’ dataset, and then split it into train and test set.

```
> dim(for_training1)
[1] 53354    11
> dim(for_testing1)
[1] 22867    11
```

➢`weighted_svm_model <- ksvm(Response ~ .,data = for_training1,kernel = "rbfdot",kpar = list(sigma=0.001953125),C = 1,weights=model_weights,prob.model = TRUE,scaled = FALSE)`

```
> weighted_svm_model
```

SVM with Weighted Loss Function



SVM with Weighted Loss Function

Thank you