Comparative analysis of Thyroid disease Classification Models

Ashutosh Shirsat

2023-11-15

Table of Content

List of Figures

List of Tables

1 Introduction

This data science project focuses on analyzing thyroid data from the UCI Machine Learning Repository and developing a 2-class classification prediction model for thyroid disease for early detection and improved patient outcomes in the diagnosis and management of thyroid diseases.

In this project will compare and evaluate three different models, namely Generalized Linear Model (GLM), Support Vector Machines (SVM), and Random Forest (RF), to determine the most suitable model for accurate predictions on this dataset.

About Thyroid

- Thyroid gland's job is to produces thyroid hormones that regulates the body's metabolism.
- There are 2 types of thyroid abnormalities Hyperthyroidism and Hypothyroidism.

  Hyperthyroidism is caused by the release of too much thyroid hormones

  Hypothyroidism is caused by release of too little thyroid hormones.

- Thyroid functional test include Thyroid blood test which check hormones i.e. TSH, T3, T4/Free T4 Index(FTI)

Assumptions : As there is no information regarding unit of parameters TSH, T3, TT4, T4U and FTI. As per understanding of data value and normal range information from online following units are assumed.

TSH: mlU/L, T3: nmol/L, TT4: nmol/L, T4U: no unit , FTI: nmol/L

New binary outcome variable 'thyroid':

The original outcome variable, 'target,' has 32 classes, has been transformed into a binary outcome variable, 'thyroid,' consisting of two classes. This conversion was undertaken to simplify and reduce the complexity of the problem, address imbalanced class issues, and establish a baseline model for potential future research.

A value of 1 indicates the presence of the disease if 'target' had a disease code from 'A' to 'H'; otherwise, the value is 0, indicating the absence of the disease.

```
# Function to create new Outcome variable "thyroid"
create_thyroid_variable = function(rawThyroidData) {
  disease_code = c("A", "B", "C", "D", "E", "F", "G", "H")
  modifiedData = rawThyroidData |>
                 mutate(thyroid = ifelse(grepl(paste(disease_code,
                                                     collapse = "|"),
                                               target), 1, 0))
```

```
  return(modifiedData)
}
# Create new binary Outcome variable i.e. 'thyroid'.
rawThyroidData = create_thyroid_variable(rawThyroidData) |>
                 dplyr::select(thyroid, everything())
```

Table 1: Variable Description

Nr	Variable	Description	Scale	Details
1	patient_id	Unique id of the patient	Integer	-
2	age	Age of the patient	Integer	In years
3	sex	Gender of patient	Binary	F: female, M: male
4	on_thyroxine	Patient is on Thyroxine	Binary	t: True, f: False
5	query_on_thyroxine	Query weather patient is on thyroxine	Binary	t: True, f: False
6	on_antithyroid_meds	Patient is on antithyroid meds	Binary	t: True, f: False
7	sick	Patient is sick	Binary	t: True, f: False
8	pregnant	Patient is pregnant	Binary	t: True, f: False
9	thyroid_surgery	Patient has undergone thyroid surgery	Binary	t: True, f: False
10	I131_treatment	Patient has undergone I131 treatment	Binary	t: True, f: False
11	query_hypothyroid	Patient believes they have hypothyroid	Binary	t: True, f: False
12	query_hyperthyroid	Patient believes they have hyperthyroid	Binary	t: True, f: False
13	lithium	Whether patient Lithium	Binary	t: True, f: False
14	goitre	Patient has goitre	Binary	t: True, f: False
15	tumor	Patient has tumor	Binary	t: True, f: False
16	hypopituitary	Patient has hypopituitary	Binary	t: True, f: False
17	psych	Patient has psych	Binary	t: True, f: False
18	TSH_measured	Whether TSH measured in blood	Binary	t: True, f: False
19	TSH	TSH level in blood	Flot	In mIU/L
20	T3_measured	Whether T3 measured in blood	Binary	t: True, f: False
21	T3	T3 level in blood	Flot	In ng/dL or nmol/L
22	TT4_measured	Whether TT4 measured in blood	Binary	t: True, f: False
23	TT4	Total T4 level in blood	Flot	In nmol/L
24	T4U_measured	Whether T4U measured in blood	Binary	t: True, f: False
25	T4U	T4 Uptake level in blood	Flot	In nmol/L or in %
26	FTI_measured	Whether FTI measured in blood	Binary	t: True, f: False
27	FTI	FTI level in blood	Flot	In ng/dL or nmol/L
28	TBG_measured	Whether TBG measured in blood	Binary	t: True, f: False
29	TBG	TBG level in blood	Flot	NA
30	referral_source	Source of Patient referral	Categorical	NA
31	target	Thyroid Diagnose Status	Categorical	Negative Diagnosis: - ; Hyperthyroid: A, B, C, D ; Hypothyroid: E, F, G, H ; Binding protein: I, J ; Non-thyroidal illness: K ; Replacement Therapy: L, M, N ; Antithyroid treatment: O, P, Q ; Miscellaneous: R, S, T ;
32	THYROID	Thyroid Diagnose Status. When value of 'target' variable has any letter 'A' to 'H', Thyroid disease is 'Yes'. Else Thyroid disease is 'No'.	Binary	0: No, 1: Yes

2 Data Cleanup

Convert category variables to factor.

3 Data Pre-Processing

3.1 Missing values

Missing Value General Approach is exclude observation with missing values. As predictor 'TBG' has too many missing values i.e. 96%. Predictor 'TBG' is removed. Predictor 'TBG_measured' is also removed along with 'TBG' as this is a flag for measurement of TBG as per data context and interpretation.

Table 2: Missing values per variable

thyroid	0
patient_id	0
age	0
sex	307
referral_source	0
on_thyroxine	0
query_on_thyroxine	0
on_antithyroid_meds	0
sick	0
pregnant	0
thyroid_surgery	0
I131_treatment	0
query_hypothyroid	0
query_hyperthyroid	0
lithium	0
goitre	0
tumor	0
hypopituitary	0
psych	0
TSH_measured	0
TSH	842
T3_measured	0
T3	2604
TT4_measured	0
TT4	442
T4U_measured	0
T4U	809
FTI_measured	0
FTI	802
TBG_measured	0
TBG	8823
target	0

```
# Remove predictor TBG, TBG_measured and Remove observation with Missing values
tidyThyroidData = cleanThyroidData  |>
                  dplyr::select(!c(TBG, TBG_measured)) |>
                  na.omit()
```

3.2 Outcome variable

Due to the imbalanced nature of the outcome variable, the selection of a suitable performance metric is essential for evaluating the predictive model. Accuracy can be misleading when dealing with imbalanced datasets. Therefore, the ROC-AUC metric is chosen over Accuracy for feature selection and Model fitting, as it offers a more robust evaluation, is informative, not affected by the class distribution and provides visual interpretability for the model's performance. See Table 3

Table 3: Outcome Variable Summary

Thyroid disease	Number of patients
No	5176
Yes	613

3.3 Zero- and Near Zero-Variance Predictors

Identifying and removing zero variance predictors is a crucial step in the data pre-processing phase before building models. These uninformative features, characterized by having constant values across the entire dataset, can significantly impact the stability and consistency of many models excluding tree-based models. The concern here is that these predictors may become zero-variance predictors when the data are split into cross-validation sub-samples or that a few samples may have an undue influence on the model.

Predictors "TSH_measured", "T3_measured", "TT4_measured", T4U_measured" and "FTI_measured" are zero variance predictors. See Table 4 of Zero and Near Zero variance predictors of category variables and in conjunction with inspection of predictor's frequency table (See Table 16 and Table 16 in Appendix Section 9.1)

Table 4: Zero and Near Zero variance predictors

Features	Freq Ratio	Percent Unique	zeroVar	nzn
TSH_measured	0.00000	0.0172741	TRUE	TRUE
T3_measured	0.00000	0.0172741	TRUE	TRUE
TT4_measured	0.00000	0.0172741	TRUE	TRUE
T4U_measured	0.00000	0.0172741	TRUE	TRUE
FTI_measured	0.00000	0.0172741	TRUE	TRUE
sick	23.95259	0.0345483	FALSE	TRUE
tumor	43.53077	0.0345483	FALSE	TRUE
I131_treatment	50.23009	0.0345483	FALSE	TRUE
on_antithyroid_meds	60.58511	0.0345483	FALSE	TRUE
thyroid_surgery	66.31395	0.0345483	FALSE	TRUE
pregnant	78.30137	0.0345483	FALSE	TRUE
lithium	86.71212	0.0345483	FALSE	TRUE
query_on_thyroxine	95.48333	0.0345483	FALSE	TRUE
goitre	122.17021	0.0345483	FALSE	TRUE
hypopituitary	2893.50000	0.0345483	FALSE	TRUE

3.4 Exclude non-informative variables

Zero variance predictors : Predictors with zero variance are removed. See table Table 4 in section Section 3.3.

'patient_id' : It does not contribute to model from context point of view.

'target' : As new Outcome variable is drive from this variable. It does not contribute to model.

```
# Remove zero variance predictors from data set.
# Remove old outcome var 'target' and predictor 'patient_id'
tidyThyroidData = tidyThyroidData |> dplyr::select(!c(patient_id, target)) |>
                  dplyr::select(-(zero_var$features))
```

3.5 Outliers and Non-Normal Distributions

Age : From statistic summary of age, there are implausible values in dataset. Subject's age is above 110 are removed.

TSH, T3, TT4, T4U and FTI : These are the functional diagnostic test for Thyroid. Due to limited availability of domain knowledge and the inability to define a plausible range for these variables, outliers in the dataset are retained for the following reasons:

(1) Outliers may contain valuable information and removing them without domain knowledge could lead to information loss and model inaccuracy; and

(2) Removing outliers could introduce bias and result in an imbalanced outcome variable.

Transformation to resolved non-normal distribution cause by outliers : Outliers might be a result of a skewed distribution. Some model are sensitive to outliers, e.g. GLM. Transformation resolves non-normal distributions caused by outliers by reduce skewness and stabilize variation. So sensitivity to outliers in some models, such as GLM, is addressed by transforming predictors. This enhances model robustness and improves overall performance.

After evaluating skewness of these variables, skew or Non-Normal distributed variables TSH, T3, T4U will be log-transformed. Skewness of FTI is increased after log transform. TT4 skewness factor is near 1.0. This is the reason FTI and TT4 are not log transformed. See skewness evaluation in Table 5.

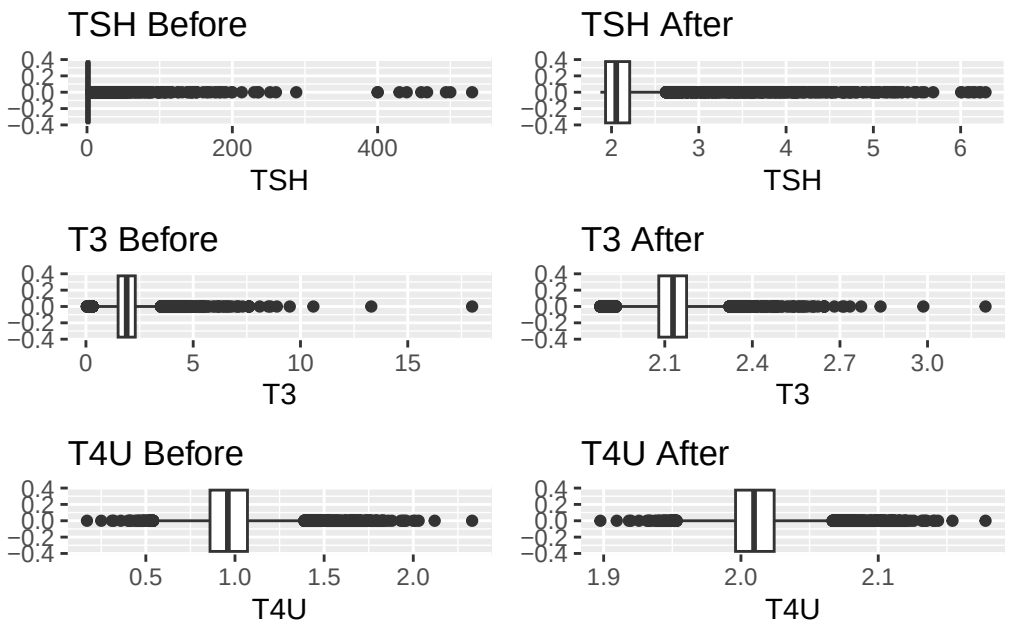


Figure 1: Log-Transformed Vaiables TSH T3 T4U

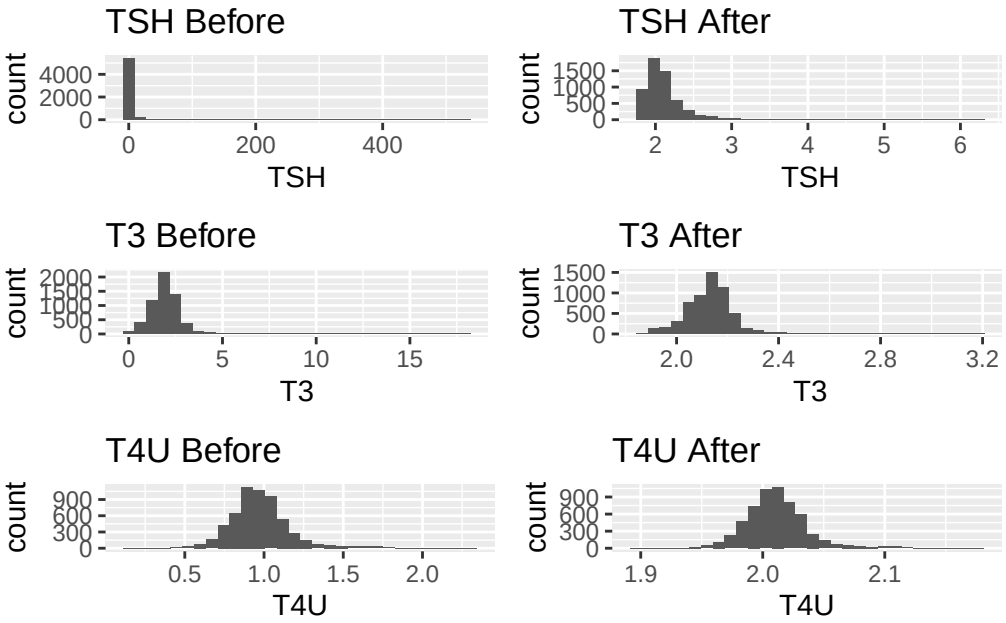


Figure 2: Log-Transformed Vaiables TSH T3 T4U

Table 5: Numerical Variables Skewness

Row	TSH	T3	TT4	T4U	FTI
Before	12.81	3.06	1.27	1.11	1.92
After	4	1.21	1.27	0.94	1.92

Table 6: Statistic summary of continuous variables after Outliers process

Variable	Min	1st qu.	Median	Mean	3rd qu.	Max
FTI	1.40	93.00	109.00	112.13	127.00	642.00
T3	1.88	2.08	2.13	2.13	2.17	3.20
T4U	1.90	2.00	2.01	2.01	2.02	2.18
TSH	1.87	1.93	2.05	2.18	2.21	6.29
TT4	2.00	87.00	104.00	107.87	125.00	450.00
age	1.00	38.00	56.00	53.61	69.00	97.00

Variable	Min	1st qu.	Median	Mean	3rd qu.	Max
FTI	1.40	93.00	109.00	112.13	127.00	642.00
T3	0.05	1.50	1.90	1.95	2.30	18.00
T4U	0.17	0.86	0.96	0.98	1.07	2.33
TSH	0.00	0.40	1.30	5.16	2.60	530.00
TT4	2.00	87.00	104.00	107.88	125.00	450.00
age	1.00	38.00	56.00	76.29	69.00	65512.00

4 Exploratory Data Analysis

Exploratory analysis and correlation index shows correlation between following predictors; see Figure 6

1. The strongest positive correlation is between TT4 and FTI (0.81), indicating a high association between these two variables.
2. There is a moderate positive correlation between T3 and TT4 (0.55) and between T3 and T4U (0.39).
3. TSH shows moderate negative correlations with TT4 (-0.41) and T3 (-0.27).
4. The rest of the correlations are relatively small (absolute values closer to zero), indicating weak or negligible correlations

Predictor and outcome variable analysis

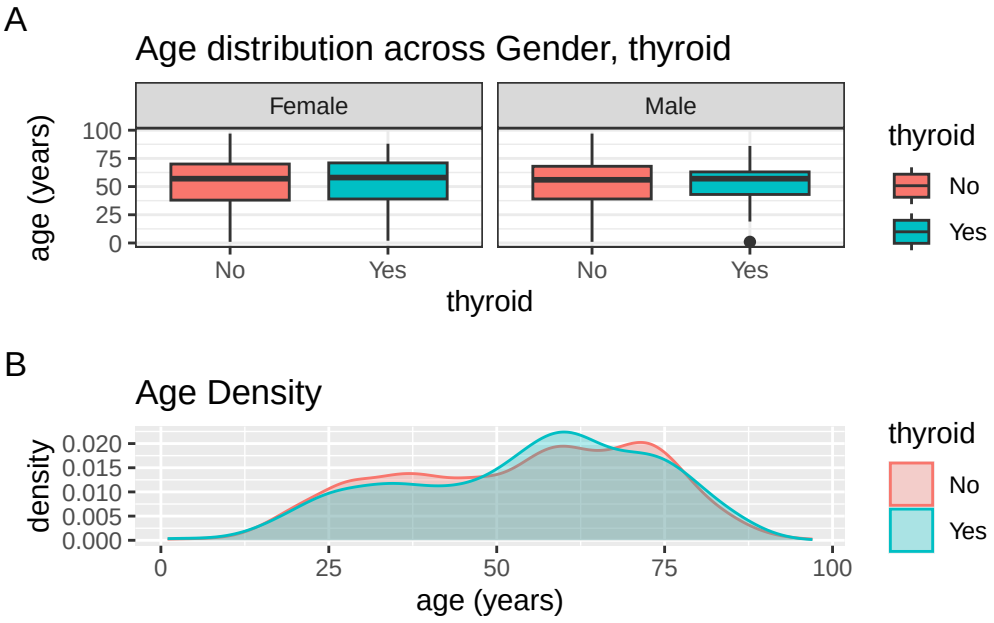


Figure 3: Age Gender distribution

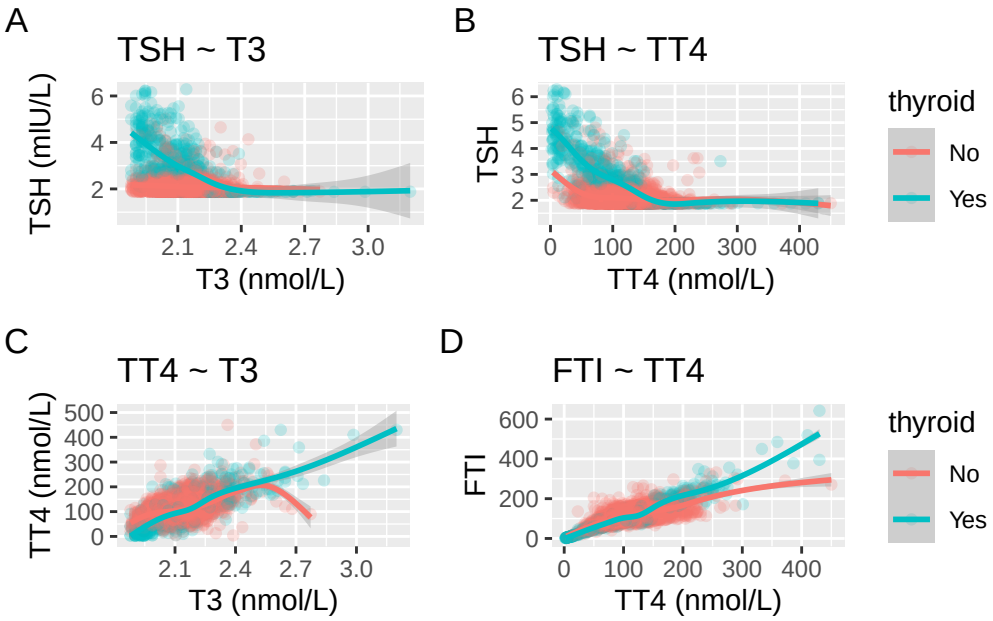


Figure 4: Scatter Plots distribution TSH,T3, TT4, FTI with Outcome

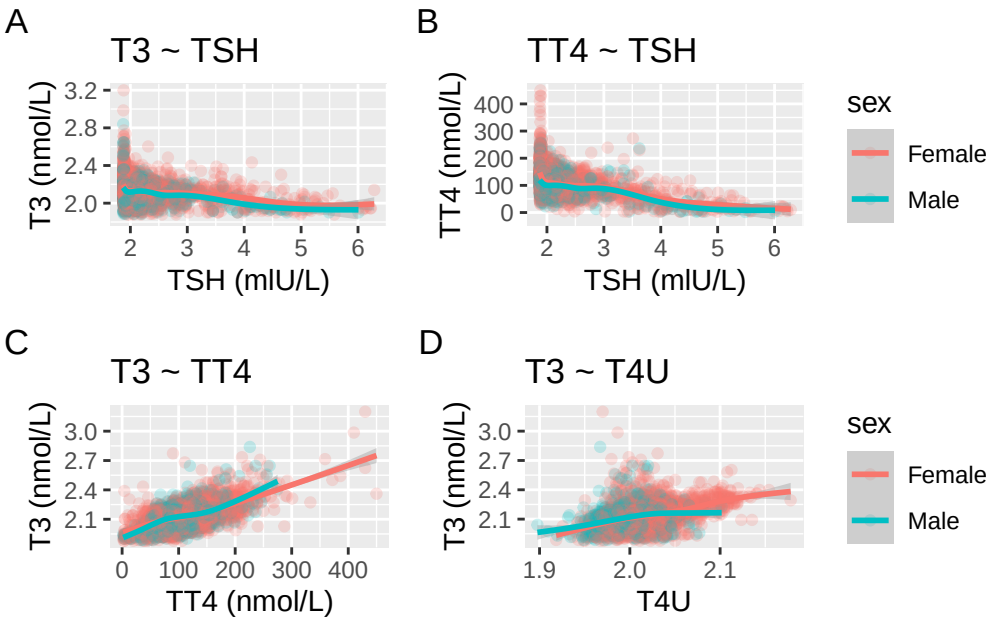


Figure 5: Scatter plot distribution T3, TSH, TT4, T4U with gender

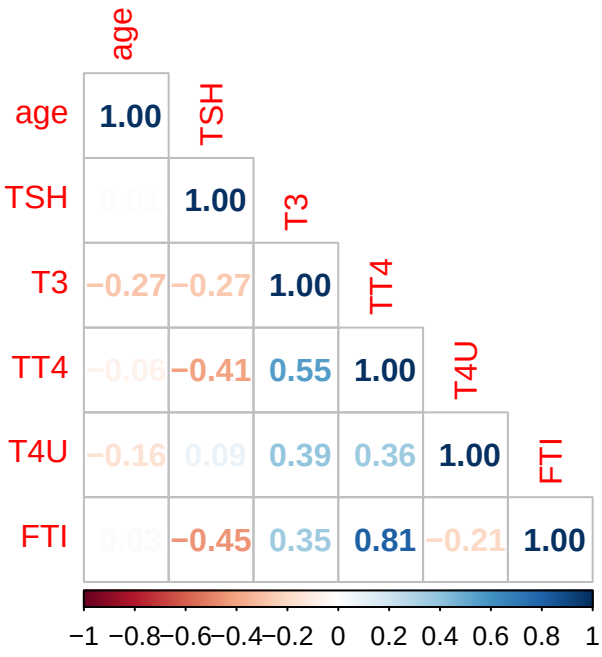


Figure 6: Correlation Matrix

5 Feature Selection for GLM and SVM Model

Random Forest has implicit feature selection. GLM and SVM model has no implicit feature selection. For GLM and SVM Model, there are 2 possible methods for feature selection are considered:

1. Step-wise Subset Selection
2. Regularization : a) Lasso, b) Elastic Net

5.1 Stepwise Subset Selection

AICstep() and RFE() functions were explored, but both encountered runtime errors with SVM models. AICstep() lacked ROC/AUC metric and built-in cross-validation support. Stepwise selection was not pursued due to these technical constrain and limitations by R functions.

AICStep() Runtime Error : See the code in Appendix Section 9.3.1 for reproducibility

"Error in UseMethod("extractAIC") : no applicable method for 'extractAIC' applied to an object of class "c('svm.formula', 'svm')"

RFE() Runtime Error : See the code in Appendix Section 9.3.2 for reproducibility.

"Error in { : task 1 failed -"dim(X) must have a positive length"

5.1.1 Resampling

5-fold cross validation 5 times resampling is used throughout this project due to the imbalanced nature of the outcome variable and a moderate sample size. Although this method helps address class imbalance and provides robust performance evaluation and assess the variance in model performance. However it may require increased computational time and resource utilization.

Stratified Randomization sampling based on Response variable is used. Preserve the response variable class proportion in train and test or cross validation folders dataset same as original dataset.

5.2 Regularization

To address limitations of stepwise selection (see Section 5.1), evaluated regularization approach using 'caret' and 'glmnet' packages. Observed following benefits over stepwise selections.

- Feature selection is independent of the model.
- Allows consistent features for both SVM and GLM models.

- It offers ROC as performance metric and support for cross validation is in-built.
- Streamlined and consistent syntax for model fitting.

Based on these finding, regularization method is selected for feature selection. As provides a robust and efficient way to perform feature selection.

Lasso and Elastic Net regularization are performed using package 'glmnet' and 'caret' in following steps;

1. Perform glmnet() with default automatic lambda generation.
2a. Set 'lambda' Grid from values generated in Step 1.
2b. Perform caret train() with cross-validation.

5.2.0.1 Lasso

```
# Step 1: glmnet lasso Model with default lambda generation. Alpha = 1.
set.seed(12356)
lasso_glmnet = glmnet(x = model.matrix(thyroid ~ . , thyroidData)[,-1],
                      y = as.numeric(thyroidData$thyroid),
                      nlambda = 100, alpha = 1,
                      family = "binomial")

# Step 2: Set lambda Grid :
# get Lambdas from glmnet() and set as grid. Alpha = 1
lambda_glmnet_lasso = lasso_glmnet$lambda
grid_lasso = expand.grid(lambda = lambda_glmnet_lasso, alpha = 1)

# Step 3: Perform caret Train. method = "glmnet", metric="ROC".
# Set caret trainControl with 5-fold cross validation 5 times
set.seed(12356)
lasso_glmnet_cv_caret = train(data = thyroidData, thyroid ~ . ,
                              method = "glmnet", metric = "ROC",
                              preProcess = c("center", "scale"),
                              savePredictions = TRUE,
                              tuneGrid = grid_lasso,
                              trControl = fitCtrl)
```

5.2.0.2 Elastic Net

```
# Step 1: glmnet elastic Net Model with default lambda generation. Alpha = 0.5.
set.seed(12356)
```

```r
elasticNet_glmnet = glmnet(x = model.matrix(thyroid ~ . , thyroidData)[,-1],
                           y = as.numeric(thyroidData$thyroid),
                           nlambda = 100, alpha = 0.5,
                           family = "binomial")

# Step 2: Set lambda Grid :
# get Lambdas from glmnet() and set as grid. Alpha = 0.5
lambda_glmnet_elasticNet = elasticNet_glmnet$lambda
grid_elasticNet = expand.grid(lambda = lambda_glmnet_elasticNet, alpha = 0.5)

# Step 3: Perform caret Train. method = "glmnet", metric="ROC".
# Set caret trainControl with 5-fold cross validation 5 times

set.seed(12356)
elasticNet_glmnet_cv_caret = train(data = thyroidData, thyroid ~ . ,
                                   method = "glmnet", metric = "ROC",
                                   preProcess = c("center", "scale"),
                                   savePredictions = TRUE,
                                   tuneGrid = grid_elasticNet,
                                   trControl = fitCtrl)
```

5.3 Lasso vs ElasticNet train Evaluation

The performance of Lasso and ElasticNet is evaluated based on the following metrics:

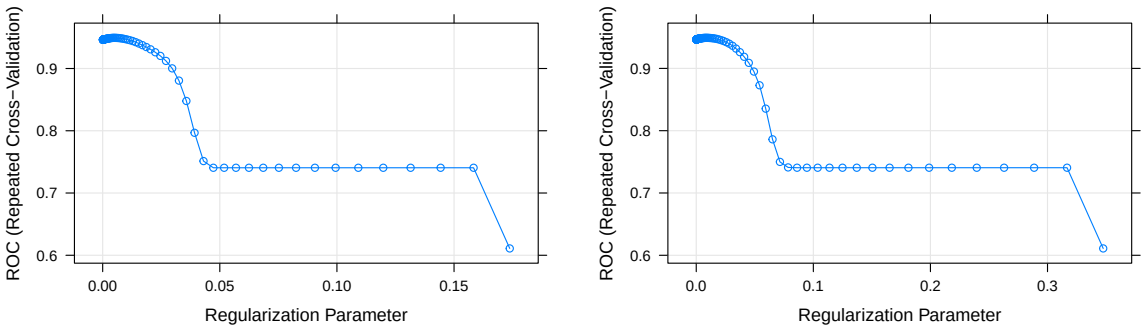
1. Resampling AUC during cross validation
2. Important variables index

Evaluation Findings:

- Lasso has a slightly better AUCs distribution across resampling. (See Figure 8)
- Lasso removes more features (See Table 9) , including correlated ones. (See correlation Matrix Figure 6)

When interpretability and a simpler model are of high importance, Lasso could be preferred. However, if multicollinearity is a major concern and preserving correlated features is essential, ElasticNet might be a better choice. Since Lasso increases interpretability and simplifies the model, Lasso is chosen.

5.3.1 Resampling AUCs



(a) Lasso (b) Elastic Net

Figure 7: Hyperparameteres tunning

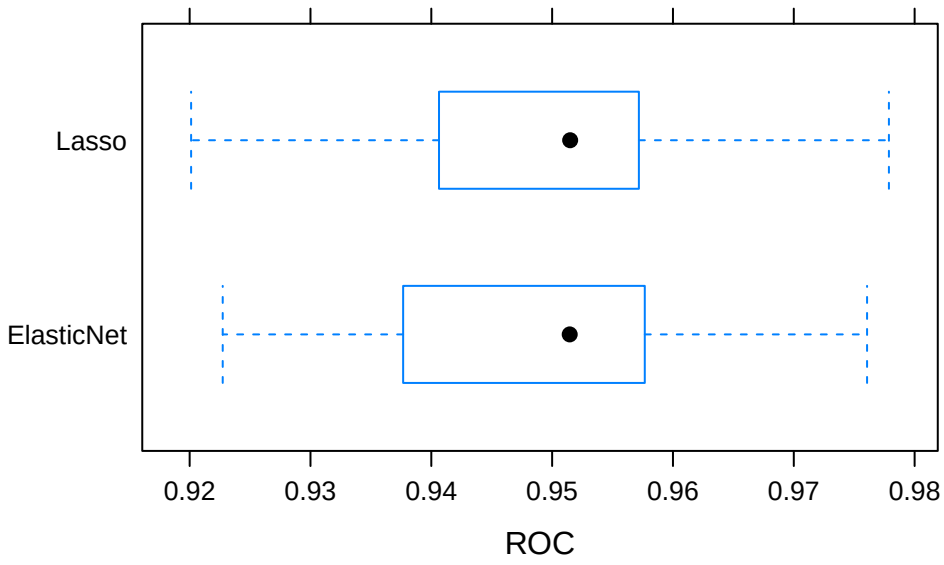


Figure 8: Lasso and ElasticNet Train resampled AUCs

5.3.2 Important Variables

Table 9: Lasso and ElasticNet VarIMP Features Index Table

Features	Lasso_Overall	ElasticNet_Overall
TSH	100.00	100.00
on_thyroxineTrue	47.99	40.43
FTI	36.67	33.87
thyroid_surgeryTrue	19.05	15.93
T3	14.68	15.65
referral_sourceSVHC	8.71	10.14
tumorTrue	8.31	8.82
query_hyperthyroidTrue	6.85	8.32
sexMale	3.17	4.36
referral_sourceSVI	2.50	3.02
I131_treatmentTrue	0.68	1.27
age	0.19	1.14
TT4	0.00	1.67
T4U	0.00	0.66
psychTrue	0.00	0.25
goitreTrue	0.00	0.00
hypopituitaryTrue	0.00	0.00
lithiumTrue	0.00	0.00
on_antithyroid_medsTrue	0.00	0.00
pregnantTrue	0.00	0.00
query_hypothyroidTrue	0.00	0.00
query_on_thyroxineTrue	0.00	0.00
referral_sourceSTMW	0.00	0.00
referral_sourceSVHD	0.00	0.00
referral_sourceWEST	0.00	0.00
sickTrue	0.00	0.00

6 Model Fitting and Tuning

For GLM, SVM and RF model fitting and also tuning hyper-parameters, the R caret package and 5-fold cross-validation 5 times resampling is used. See section Section 5.1.1. Performance metric 'ROC' is chosen. See section Section 3.2. Predictors selected for GLM and SVM Model fitting are based on LASSO feature selection. See Section 5.3 and Table 9.

6.1 GLM

Interpretation :

- TSH : For every unit increase in TSH, the log-odds ofhaving thyroid disease increase by 6.12.
- T3 : For every unit increase in T3, the log-odds of havingthyroid disease increase by 4.72.
- Tumor : Having Tumor increase the log-odds of havingThyroid disease by 1.81
- undergone thyroid surgery reduces the log-odds of havingthyroid disease by 11.18.
- High TSH, T3, FTI, and the presence of a tumor increase the risk of Thyroid disease.
- Medical interventions (Thyroid Surgery and ThyroxinUsage) reduce the risk Thyroid disease.

6.1.1 Full Model - Model Fitting

```
Call:
NULL

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.4691  -0.2796  -0.1660  -0.0696   4.2854

Coefficients: (1 not defined because of singularities)
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              1.589e+01  1.544e+01   1.029 0.303364
age                      6.686e-03  3.612e-03   1.851 0.064156 .
sexMale                 -3.567e-01  1.525e-01  -2.339 0.019351 *
referral_sourceSTMW      2.441e-01  4.022e-01   0.607 0.543941
referral_sourceSVHC     -1.057e+00  2.920e-01  -3.620 0.000295 ***
referral_sourceSVHD     -4.912e-02  6.093e-01  -0.081 0.935744
referral_sourceSVI      -5.323e-01  1.537e-01  -3.463 0.000534 ***
referral_sourceWEST             NA         NA      NA       NA
```

```
on_thyroxineTrue        -6.109e+00  5.216e-01 -11.712  < 2e-16 ***
query_on_thyroxineTrue  -7.058e-02  6.440e-01  -0.110 0.912723
on_antithyroid_medsTrue -6.457e-01  5.846e-01  -1.104 0.269393
sickTrue                -1.342e-01  3.392e-01  -0.395 0.692495
pregnantTrue             7.563e-01  5.735e-01   1.319 0.187229
thyroid_surgeryTrue     -1.066e+01  1.468e+00  -7.261 3.85e-13 ***
I131_treatmentTrue      -1.212e+00  5.880e-01  -2.062 0.039195 *
query_hypothyroidTrue    1.233e-01  2.457e-01   0.502 0.615808
query_hyperthyroidTrue   6.787e-01  1.979e-01   3.430 0.000603 ***
lithiumTrue              5.432e-01  6.203e-01   0.876 0.381182
goitreTrue              -1.305e+01  3.035e+02  -0.043 0.965707
tumorTrue                1.772e+00  2.942e-01   6.025 1.69e-09 ***
hypopituitaryTrue       -7.952e+00  1.470e+03  -0.005 0.995683
psychTrue               -3.281e-01  4.181e-01  -0.785 0.432687
TSH                      6.224e+00  2.477e-01  25.132  < 2e-16 ***
T3                       5.315e+00  7.947e-01   6.688 2.27e-11 ***
TT4                      1.968e-02  8.136e-03   2.418 0.015585 *
T4U                     -2.332e+01  7.829e+00  -2.978 0.002898 **
FTI                      9.901e-03  7.674e-03   1.290 0.196960
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3910.8  on 5785  degrees of freedom
Residual deviance: 1810.9  on 5760  degrees of freedom
AIC: 1862.9

Number of Fisher Scoring iterations: 15
```

6.1.2 Lasso Model - Model Fitting

```
Call:
NULL

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.4083  -0.2812  -0.1715  -0.0770   4.2235

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)           -29.585170   1.808224 -16.361  < 2e-16 ***
```

```
TSH                       6.120957   0.240771  25.422  < 2e-16 ***
on_thyroxineTrue         -6.028628   0.524572 -11.492  < 2e-16 ***
FTI                       0.029169   0.001888  15.449  < 2e-16 ***
thyroid_surgeryTrue     -11.176343   1.514889  -7.378 1.61e-13 ***
T3                        4.720837   0.704194   6.704 2.03e-11 ***
referral_sourceSVHC      -1.096121   0.245811  -4.459 8.23e-06 ***
tumorTrue                 1.808145   0.290990   6.214 5.17e-10 ***
query_hyperthyroidTrue    0.713296   0.195199   3.654 0.000258 ***
sexMale                  -0.297126   0.147964  -2.008 0.044633 *
referral_sourceSVI       -0.514290   0.152219  -3.379 0.000728 ***
I131_treatmentTrue       -1.243277   0.587390  -2.117 0.034293 *
age                       0.006831   0.003525   1.938 0.052598 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3910.8  on 5785  degrees of freedom
Residual deviance: 1828.6  on 5773  degrees of freedom
AIC: 1854.6

Number of Fisher Scoring iterations: 8
```

6.2 Model Evaluation - Full vs Lasso Model of GLM

Based on the performance metrics summary, ROC plots and Resampling AUCs summarize the evaluation of Full and lasso models as follows:

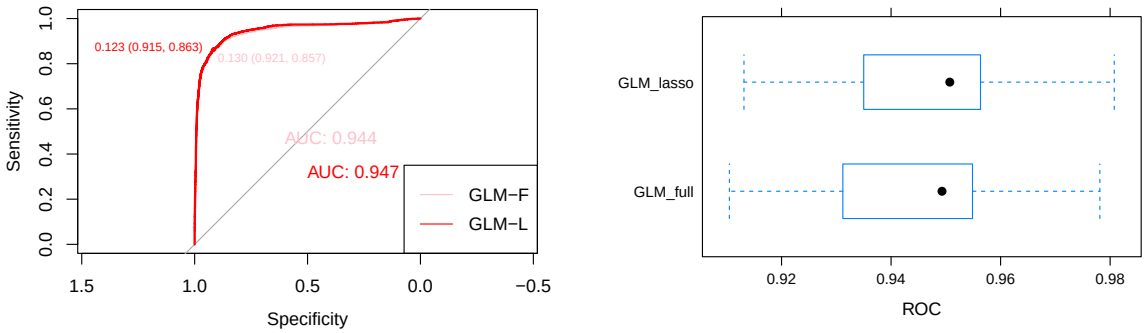
- **Metric :** Both Full and Lasso models metrics are almost same.
- **ROC Plot:** Lasso Model has highest AUC compare to full model. Indicate better probability of correctly distinguishing instances in different classes. Lasso Model has more likely less False Positive case as best threshold compare to Full model.
- **Resampling AUCs:** Lasso Model consistently performed well across different validation folds. This indicates that the Lasso Model model generalizes better and is less sensitive to different data splits.

Conclusion :

GLM Lasso model perform better than Full model. GLM Lasso model is selected for further evaluation.

Table 10: Model Performance Metric GLM Full vs lasso

	TN	FP	FN	TP	Precision	Recall_Score	F1_Score	Accuracy	Specificity	Sensitivity	AUC
GLM_full	25584	281	1250	1815	0.87	0.59	0.7	0.95	0.99	0.59	0.944
GLM_lasso	25586	279	1256	1809	0.87	0.59	0.7	0.95	0.99	0.59	0.947



(a) Plot ROC GLM

(b) Plot Resampling AUCs GLM

Figure 9: GLM Model Evaluation Plots between Full and Lasso

6.3 SVM

The SVM model is a C-Support Vector Classification (C-svc) model. It uses the Gaussian Radial Basis kernel function.

- Best Tune Hyperparameters :

Grid = [C(0.25, 1, 1.5, 2, 5), Sigma(0, 0.5, 1, 1.5)]

Cost = 0.25, Sigma = 0.5

- Cost : Cost is smaller, indicate more tolerant of violation to margin, relatively wider margin.
- Sigma: Sigma is small, implies complex and less smoothness in decision boundary (hyperplane).
- Number of Support Vector : 1305 data point that define decision boundary (hyperplane). Higher number indicate complex decision boundary.

6.3.1 Full Model - Hyper-parameter Tuning

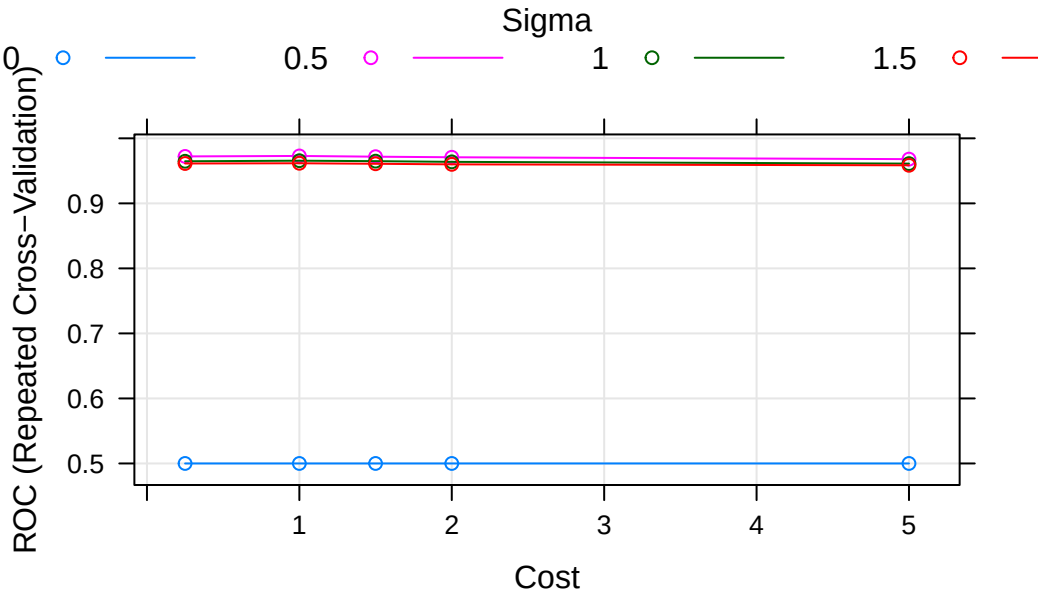


Figure 10: SVM Full model - Hyperparameteres tunning

6.3.2 Full Model - Model Fitting

```
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.5

Number of Support Vectors : 2057

Objective Function Value : -578.8236
Training error : 0.01037
Probability model included.
```

6.3.3 Lasso Model - Hyper-parameter Tuning

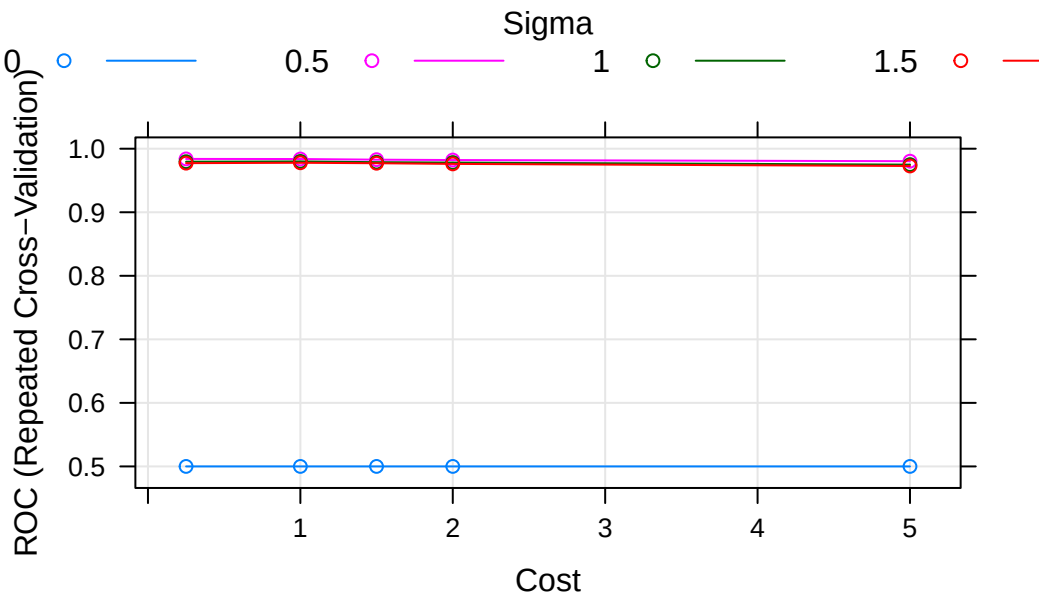


Figure 11: SVM Lasso model - Hyperparameteres tunning

6.3.4 Lasso Model - Model Fitting

```
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 0.25

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.5

Number of Support Vectors : 1305

Objective Function Value : -204.2054
Training error : 0.050121
Probability model included.
```

6.4 Model Evaluation - Full vs Lasso Model of SVM

Based on the performance metrics summary, ROC plots and Resampling AUCs summarize the evaluation of Full and lasso models as follows:

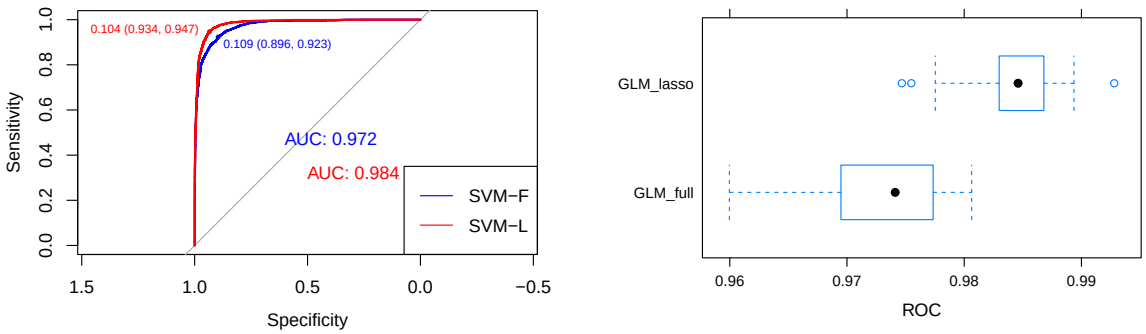
- **Metric :** Both Full and Lasso models metrics are almost same.
- **ROC Plot:** Lasso Model has highest AUC compare to full model. Indicate better probability of correctly distinguishing instances in different classes. Lasso Model has more likely less False Positive case as best threshold compare to Full model.
- **Resampling AUCs:** Lasso Model consistently performed well across different validation folds. This indicates that the Lasso Model model generalizes better and is less sensitive to different data splits.

Conclusion :

SVM Lasso model perform better than Full model. GLM Lasso model is selected for further evaluation.

Table 11: Model Performance Metric SVM Full vs lasso

	TN	FP	FN	TP	Precision	Recall_Score	F1_Score	Accuracy	Specificity	Sensitivity	AUC
SVM_full	25551	314	999	2066	0.87	0.67	0.76	0.95	0.99	0.67	0.972
SVM_lasso	25516	349	763	2302	0.87	0.75	0.81	0.96	0.99	0.75	0.984



(a) Plot ROC SVM

(b) Plot Resampling AUCs SVM

Figure 12: SVM Model Evaluation Plots between Full and Lasso

6.5 Random Forest

Here are the key points of final model based on best tune 'mtry' value;

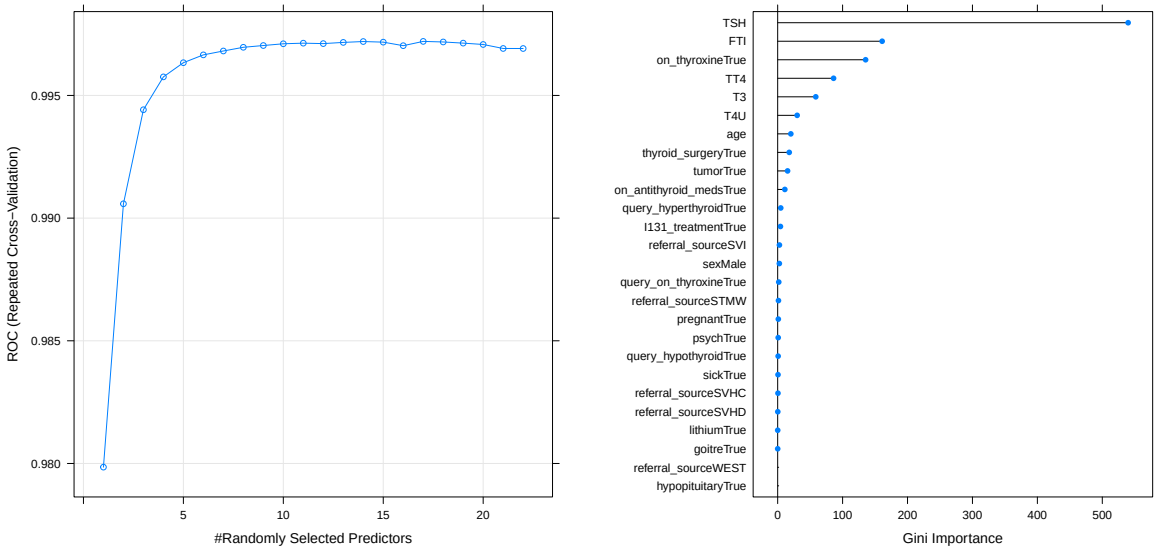
- Number of Trees: The model consists of 600 decision trees.
- mtry: This parameter specifies the number of variables tried at each split. In this case, 17 variables are considered at each split during tree building.
- OOB (Out-of-Bag) Error Rate: The OOB estimate of the error rate is a cross-validation technique specific to Random Forests. It estimates the model's performance on unseen data (samples not used during the tree construction). In this case, the OOB error rate is approximately 1.35%, which is very low and indicates a well-performing model.

As per Gini Index Figure 14b, The feature "TSH" has the highest importance value, followed by "FTI", "on_thyroxineTrue", "TT4" and so on. These features are the most important in making predictions. The feature "referral_sourceWEST" and "hypopituitaryTrue" have importance values of 0, which suggests that they are not be relevant for making predictions.

The corresponding Area Under the Curve (AUC) value was 0.99 indicating its ability to discriminate between the two classes. A precision of 0.94 indicates that when the model predicts a positive class (Yes), it is correct 94% of the time. See Table 12. in Section 7.

```
Call:
 randomForest(x = x, y = y, ntree = 600, mtry = param$mtry)
               Type of random forest: classification
                     Number of trees: 600
No. of variables tried at each split: 17

        OOB estimate of  error rate: 1.35%
Confusion matrix:
      No Yes class.error
No  5135  38 0.007345834
Yes   40 573 0.065252855
```



(a) RF Tuning parameter and Resampling ROC

(b) RF Gini Index

Figure 13: RF Plots

7 Model Evaluation - GLM vs SVM vs RF

Based on the performance metrics summary, ROC plots of 3 models and Resampling AUCs summarize the evaluation as follows: (See models evaluation Table 12 and Figure 14a)

- **Metric :** Random Forest performed best in terms of all performance metrics.
- **Precision:** Random Forest correctly predicts with 94% accuracy whether a person has thyroid, minimizing false positives.
  - Likelihood false Thyroid diagnosis is less than other models.
- **Recall :** The Random Forest model correctly predicted True Positive cases with an accuracy of 92%.
  - Likelihood failure to diagnose is less than other models.
- **ROC Plot:** Random Forest has highest AUC. Indicate 99.7% probability of correctly distinguishing instances in different classes. RF has more likely less False Positive case as best threshold of RF Model is higher than GLM and SVM.
- **Resampling AUCs:** Random Forest consistently performed well across different validation folds. (See Figure 14b). This indicates that the Random Forest model generalizes better and is less sensitive to different data splits.

Considering feature selection, TSH is most important feature in both Random Forest and Lasso model.

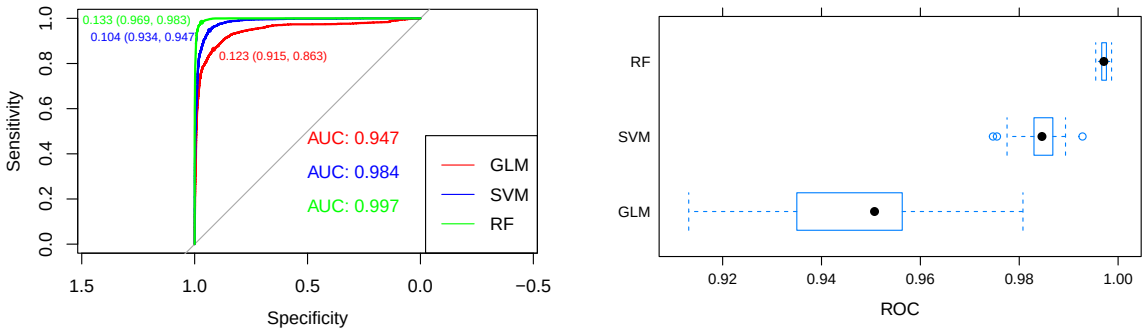
Predictors TT4 and T4U are eliminated by Lasso for GLM and SVM. However the random forest model showed high Gini Index values for predictor TT4 and T4U, indicating their substantial importance in predictive modeling. (See Lasso varIMP Table 9 and RF Gini Index Figure 13b)

Table 12: Model Performance Metric RF vs GLM and SVM Lasso Model

	TN	FP	FN	TP	Precision	Recall_Score	F1_Score	Accuracy	Specificity	Sensitivity	AUC
GLM_Lasso	25586	279	1256	1809	0.87	0.59	0.7	0.95	0.99	0.59	0.947
SVM_Lasso	25516	349	763	2302	0.87	0.75	0.81	0.96	0.99	0.75	0.984
RF	25670	195	241	2824	0.94	0.92	0.93	0.98	0.99	0.92	0.997

Table 13: Model Performance Metric RF vs GLM and SVM Full Model

	TN	FP	FN	TP	Precision	Recall_Score	F1_Score	Accuracy	Specificity	Sensitivity	AUC
GLM_Full	25584	281	1250	1815	0.87	0.59	0.7	0.95	0.99	0.59	0.944
SVM_Full	25551	314	999	2066	0.87	0.67	0.76	0.95	0.99	0.67	0.972
RF	25670	195	241	2824	0.94	0.92	0.93	0.98	0.99	0.92	0.997



(a) Plot ROC

(b) Plot Resampling AUCs

Figure 14: Model Evaluation Plots

8 Conclusion

- Random Forest model is the most favorable choice for predictive modeling in this data context.
- Model Interpretation is more complex compared to GLM.
- Blood test parameter 'TSH' is the most important feature for predicting Thyroid disease in context of this data.

Data Modeling is often an iterative process. This prediction model comparison can be used as baseline for further model enhancements for this data. For example, alternative predictive models or when additional data is collected.

Table 14: Random Forest and Lasso VarIMP Features Index Table

Features	RF_Overall	Lasso_Overall
TSH	100.00	100.00
FTI	29.81	36.67
on_thyroxineTrue	25.09	47.99
TT4	15.95	0.00
T3	10.87	14.68
T4U	5.56	0.00
age	3.75	0.19
thyroid_surgeryTrue	3.30	19.05
tumorTrue	2.83	8.31
on_antithyroid_medsTrue	2.04	0.00
query_hyperthyroidTrue	0.88	6.85
I131_treatmentTrue	0.79	0.68
referral_sourceSVI	0.50	2.50
sexMale	0.47	3.17
query_on_thyroxineTrue	0.31	0.00
referral_sourceSTMW	0.22	0.00
pregnantTrue	0.18	0.00
psychTrue	0.15	0.00
query_hypothyroidTrue	0.13	0.00
sickTrue	0.10	0.00
referral_sourceSVHC	0.08	8.71
referral_sourceSVHD	0.03	0.00
lithiumTrue	0.01	0.00
goitreTrue	0.01	0.00
hypopituitaryTrue	0.00	0.00
referral_sourceWEST	0.00	0.00

9 Appendix

9.1 Statistical Summary after Data Cleanup

Table 15: Statstical Summary of Numerical Variables

			Thyroid	
Variable	**N**	**Overall**, N = 9,172	**No**, N = 8,264	**Yes**, N = 908
patient_id	9,172			
Mean (SD)		852,947,347 (7,581,969)	852,958,017 (7,601,591)	852,850,235 (7,404,384)
Median		851,004,027	851,004,027	851,004,032
IQR		850,409,012, 860,711,023	850,404,013, 860,711,086	850,421,024, 860,707,049
Range		840,801,013, 870,119,035	840,801,013, 870,119,035	840,815,067, 870,116,038
Missing		0	0	0
age	9,172			
Mean (SD)		74 (1,184)	76 (1,247)	54 (19)
Median		55	54	57
IQR		37, 68	37, 67	39, 68
Range		1, 65,526	1, 65,526	1, 91
Missing		0	0	0
TSH	8,330			
Mean (SD)		5.2 (24.2)	2.1 (5.6)	32.0 (67.2)
Median		1.4	1.3	9.7
IQR		0.5, 2.7	0.5, 2.3	6.2, 27.0
Range		0.0, 530.0	0.0, 177.0	0.0, 530.0
Missing		842	807	35
T3	6,568			
Mean (SD)		1.97 (0.89)	1.95 (0.72)	2.15 (1.74)
Median		1.90	1.90	1.80
IQR		1.50, 2.30	1.50, 2.30	1.10, 2.60
Range		0.05, 18.00	0.05, 9.50	0.05, 18.00
Missing		2,604	2,404	200
TT4	8,730			
Mean (SD)		109 (38)	109 (32)	102 (68)
Median		104	105	89
IQR		87, 126	89, 125	60, 137
Range		2, 600	4, 600	2, 430
Missing		442	433	9
T4U	8,363			
Mean (SD)		0.98 (0.20)	0.97 (0.20)	0.99 (0.19)
Median		0.96	0.96	0.97

(continued)

Variable	N	Overall, N = 9,172	Thyroid	
			No, N = 8,264	**Yes**, N = 908
IQR		0.86, 1.07	0.86, 1.06	0.87, 1.08
Range		0.17, 2.33	0.17, 2.33	0.28, 1.83
Missing		809	754	55
FTI	8,370			
Mean (SD)		114 (42)	114 (33)	111 (86)
Median		109	110	94
IQR		93, 128	95, 128	60, 136
Range		1, 881	4, 881	1, 839
Missing		802	748	54
TBG	349			
Mean (SD)		30 (21)	30 (21)	25 (4)
Median		26	26	25
IQR		21, 31	21, 31	22, 28
Range		0, 200	0, 200	18, 30
Missing		8,823	7,923	900

Table 16: Statstical Summary of Catagorical Variables

Variable	N	Overall, N = 9,172	Thyroid	
			No, N = 8,264	**Yes**, N = 908
sex	8,865			
Female		6,073 / 8,865 (69%)	5,398 / 7,999 (67%)	675 / 866 (78%)
Male		2,792 / 8,865 (31%)	2,601 / 7,999 (33%)	191 / 866 (22%)
Missing		307	265	42
referral_source	9,172			
other		5,493 / 9,172 (60%)	4,882 / 8,264 (59%)	611 / 908 (67%)
STMW		255 / 9,172 (2.8%)	227 / 8,264 (2.7%)	28 / 908 (3.1%)
SVHC		956 / 9,172 (10%)	926 / 8,264 (11%)	30 / 908 (3.3%)
SVHD		71 / 9,172 (0.8%)	63 / 8,264 (0.8%)	8 / 908 (0.9%)
SVI		2,394 / 9,172 (26%)	2,163 / 8,264 (26%)	231 / 908 (25%)
WEST		3 / 9,172 (<0.1%)	3 / 8,264 (<0.1%)	0 / 908 (0%)
Missing		0	0	0
on_thyroxine	9,172			
False		7,932 / 9,172 (86%)	7,056 / 8,264 (85%)	876 / 908 (96%)
True		1,240 / 9,172 (14%)	1,208 / 8,264 (15%)	32 / 908 (3.5%)
Missing		0	0	0
query_on_thyroxine	9,172			

(continued)

Variable	N	**Overall**, N = 9,172	**Thyroid**	
			No, N = 8,264	**Yes**, N = 908
False		9,019 / 9,172 (98%)	8,121 / 8,264 (98%)	898 / 908 (99%)
True		153 / 9,172 (1.7%)	143 / 8,264 (1.7%)	10 / 908 (1.1%)
Missing		0	0	0
on_antithyroid_meds	9,172			
False		9,056 / 9,172 (99%)	8,159 / 8,264 (99%)	897 / 908 (99%)
True		116 / 9,172 (1.3%)	105 / 8,264 (1.3%)	11 / 908 (1.2%)
Missing		0	0	0
sick	9,172			
False		8,828 / 9,172 (96%)	7,945 / 8,264 (96%)	883 / 908 (97%)
True		344 / 9,172 (3.8%)	319 / 8,264 (3.9%)	25 / 908 (2.8%)
Missing		0	0	0
pregnant	9,172			
False		9,065 / 9,172 (99%)	8,167 / 8,264 (99%)	898 / 908 (99%)
True		107 / 9,172 (1.2%)	97 / 8,264 (1.2%)	10 / 908 (1.1%)
Missing		0	0	0
thyroid_surgery	9,172			
False		9,038 / 9,172 (99%)	8,134 / 8,264 (98%)	904 / 908 (100%)
True		134 / 9,172 (1.5%)	130 / 8,264 (1.6%)	4 / 908 (0.4%)
Missing		0	0	0
I131_treatment	9,172			
False		9,003 / 9,172 (98%)	8,114 / 8,264 (98%)	889 / 908 (98%)
True		169 / 9,172 (1.8%)	150 / 8,264 (1.8%)	19 / 908 (2.1%)
Missing		0	0	0
query_hypothyroid	9,172			
False		8,542 / 9,172 (93%)	7,745 / 8,264 (94%)	797 / 908 (88%)
True		630 / 9,172 (6.9%)	519 / 8,264 (6.3%)	111 / 908 (12%)
Missing		0	0	0
query_hyperthyroid	9,172			
False		8,521 / 9,172 (93%)	7,733 / 8,264 (94%)	788 / 908 (87%)
True		651 / 9,172 (7.1%)	531 / 8,264 (6.4%)	120 / 908 (13%)
Missing		0	0	0
lithium	9,172			
False		9,079 / 9,172 (99%)	8,177 / 8,264 (99%)	902 / 908 (99%)
True		93 / 9,172 (1.0%)	87 / 8,264 (1.1%)	6 / 908 (0.7%)
Missing		0	0	0
goitre	9,172			
False		9,088 / 9,172 (99%)	8,180 / 8,264 (99%)	908 / 908 (100%)
True		84 / 9,172 (0.9%)	84 / 8,264 (1.0%)	0 / 908 (0%)
Missing		0	0	0

(continued)

Variable	N	**Overall**, N = 9,172	**Thyroid**	
			No, N = 8,264	**Yes**, N = 908
tumor	9,172			
False		8,931 / 9,172 (97%)	8,064 / 8,264 (98%)	867 / 908 (95%)
True		241 / 9,172 (2.6%)	200 / 8,264 (2.4%)	41 / 908 (4.5%)
Missing		0	0	0
hypopituitary	9,172			
False		9,170 / 9,172 (100%)	8,262 / 8,264 (100%)	908 / 908 (100%)
True		2 / 9,172 (<0.1%)	2 / 8,264 (<0.1%)	0 / 908 (0%)
Missing		0	0	0
psych	9,172			
False		8,754 / 9,172 (95%)	7,858 / 8,264 (95%)	896 / 908 (99%)
True		418 / 9,172 (4.6%)	406 / 8,264 (4.9%)	12 / 908 (1.3%)
Missing		0	0	0
TSH_measured	9,172			
False		842 / 9,172 (9.2%)	807 / 8,264 (9.8%)	35 / 908 (3.9%)
True		8,330 / 9,172 (91%)	7,457 / 8,264 (90%)	873 / 908 (96%)
Missing		0	0	0
T3_measured	9,172			
False		2,604 / 9,172 (28%)	2,404 / 8,264 (29%)	200 / 908 (22%)
True		6,568 / 9,172 (72%)	5,860 / 8,264 (71%)	708 / 908 (78%)
Missing		0	0	0
TT4_measured	9,172			
False		442 / 9,172 (4.8%)	433 / 8,264 (5.2%)	9 / 908 (1.0%)
True		8,730 / 9,172 (95%)	7,831 / 8,264 (95%)	899 / 908 (99%)
Missing		0	0	0
T4U_measured	9,172			
False		809 / 9,172 (8.8%)	754 / 8,264 (9.1%)	55 / 908 (6.1%)
True		8,363 / 9,172 (91%)	7,510 / 8,264 (91%)	853 / 908 (94%)
Missing		0	0	0
FTI_measured	9,172			
False		802 / 9,172 (8.7%)	748 / 8,264 (9.1%)	54 / 908 (5.9%)
True		8,370 / 9,172 (91%)	7,516 / 8,264 (91%)	854 / 908 (94%)
Missing		0	0	0
TBG_measured	9,172			
False		8,823 / 9,172 (96%)	7,923 / 8,264 (96%)	900 / 908 (99%)
True		349 / 9,172 (3.8%)	341 / 8,264 (4.1%)	8 / 908 (0.9%)
Missing		0	0	0
target	9,172			
-		6,771 / 9,172 (74%)	6,771 / 8,264 (82%)	0 / 908 (0%)
A		147 / 9,172 (1.6%)	0 / 8,264 (0%)	147 / 908 (16%)

(continued)

Variable	N	**Overall**, N = 9,172	**Thyroid**	
			No, N = 8,264	**Yes**, N = 908
AK		46 / 9,172 (0.5%)	0 / 8,264 (0%)	46 / 908 (5.1%)
B		21 / 9,172 (0.2%)	0 / 8,264 (0%)	21 / 908 (2.3%)
C		6 / 9,172 (<0.1%)	0 / 8,264 (0%)	6 / 908 (0.7%)
C|I		12 / 9,172 (0.1%)	0 / 8,264 (0%)	12 / 908 (1.3%)
D		8 / 9,172 (<0.1%)	0 / 8,264 (0%)	8 / 908 (0.9%)
D|R		1 / 9,172 (<0.1%)	0 / 8,264 (0%)	1 / 908 (0.1%)
E		1 / 9,172 (<0.1%)	0 / 8,264 (0%)	1 / 908 (0.1%)
F		233 / 9,172 (2.5%)	0 / 8,264 (0%)	233 / 908 (26%)
FK		6 / 9,172 (<0.1%)	0 / 8,264 (0%)	6 / 908 (0.7%)
G		359 / 9,172 (3.9%)	0 / 8,264 (0%)	359 / 908 (40%)
GI		10 / 9,172 (0.1%)	0 / 8,264 (0%)	10 / 908 (1.1%)
GK		49 / 9,172 (0.5%)	0 / 8,264 (0%)	49 / 908 (5.4%)
GKJ		1 / 9,172 (<0.1%)	0 / 8,264 (0%)	1 / 908 (0.1%)
H|K		8 / 9,172 (<0.1%)	0 / 8,264 (0%)	8 / 908 (0.9%)
I		346 / 9,172 (3.8%)	346 / 8,264 (4.2%)	0 / 908 (0%)
J		30 / 9,172 (0.3%)	30 / 8,264 (0.4%)	0 / 908 (0%)
K		436 / 9,172 (4.8%)	436 / 8,264 (5.3%)	0 / 908 (0%)
KJ		11 / 9,172 (0.1%)	11 / 8,264 (0.1%)	0 / 908 (0%)
L		115 / 9,172 (1.3%)	115 / 8,264 (1.4%)	0 / 908 (0%)
LJ		1 / 9,172 (<0.1%)	1 / 8,264 (<0.1%)	0 / 908 (0%)
M		111 / 9,172 (1.2%)	111 / 8,264 (1.3%)	0 / 908 (0%)
MI		2 / 9,172 (<0.1%)	2 / 8,264 (<0.1%)	0 / 908 (0%)
MK		16 / 9,172 (0.2%)	16 / 8,264 (0.2%)	0 / 908 (0%)
N		110 / 9,172 (1.2%)	110 / 8,264 (1.3%)	0 / 908 (0%)
O		14 / 9,172 (0.2%)	14 / 8,264 (0.2%)	0 / 908 (0%)
OI		1 / 9,172 (<0.1%)	1 / 8,264 (<0.1%)	0 / 908 (0%)
P		5 / 9,172 (<0.1%)	5 / 8,264 (<0.1%)	0 / 908 (0%)
Q		14 / 9,172 (0.2%)	14 / 8,264 (0.2%)	0 / 908 (0%)
R		196 / 9,172 (2.1%)	196 / 8,264 (2.4%)	0 / 908 (0%)
S		85 / 9,172 (0.9%)	85 / 8,264 (1.0%)	0 / 908 (0%)
Missing		0	0	0

[1] n / N (%)

9.2 Unit Test

9.2.1 function create_thyroid_variable()

```
# Test 1: Verify range of newly created variable.
testthat::test_that("Test levels of new Outcome variable", {
expect_equal(range(rawThyroidData$thyroid), c(0,1))
})
```

```
Test passed
```

```
# Test 2: Verify thyroid values for target values with disease code.
test_that("Test correctly assigns thyroid values", {
  raw_data = data.frame(target = c("B", "D", "G", "I", "A", "H", "C"))
  expected_data = data.frame(target = c("B", "D", "G", "I", "A", "H", "C"),
                             thyroid = c(1, 1, 1, 0, 1, 1, 1))

  result = create_thyroid_variable(raw_data)

  expect_identical(result, expected_data)
})
```

```
Test passed
```

```
# Test3:Verify thyroid values for target values with combinations of disease code.
test_that("Test values with combinations of letters", {
  raw_data = data.frame(target = c("-", "E", "F", "GKJ", "C|I", "OI"),
                        stringsAsFactors = FALSE)
  expected_data = data.frame(target = c("-","E", "F", "GKJ", "C|I", "OI"),
                             thyroid = c(0, 1, 1, 1, 1, 0),
                            stringsAsFactors = FALSE)

  result = create_thyroid_variable(raw_data)

  expect_identical(result, expected_data)
})
```

```
Test passed
```

9.2.2 Factor conversion levels()

```
# Unit Test: Verify factor level
testthat::test_that("correct levels of a factor", {
expect_equal(levels(cleanThyroidData$sick), c("False", "True"))
expect_equal(levels(cleanThyroidData$sex), c("Female", "Male"))
expect_equal(levels(cleanThyroidData$thyroid), c("No", "Yes"))
})
```

```
Test passed
```

9.3 RunTime Error - Stepwise Selection

9.3.1 SVM - AICstep()

```
# # Fund best tune hyperparamtere for Full SVM model.
#
# # Step 1: Split data into training and testing sets
# set.seed(12356)
# training.samples = thyroidData$thyroid |>
#   createDataPartition(p = 0.7, list = FALSE)
# train_data  = thyroidData[training.samples, ]
# test_data = thyroidData[-training.samples, ]
#
# # Step 2: Define parameter grid
# param_grid = expand.grid(
#   C = c(0.1, 1, 10),
#   kernel = c("linear", "polynomial", "radial")
# )
#
# # Step 3: Initialize variables
# best_auc = 0
# best_params = NULL
#
# # Step 4-6: Grid search
# for (i in 1:nrow(param_grid)) {
#   # Step 4: Fit SVM model
#   model = svm(thyroid ~ ., data = train_data,
#               kernel = param_grid$kernel[i],
#               cost = param_grid$C[i])
#
#   # Step 5: Evaluate model on testing data
#   predictions = predict(model, newdata = test_data, type = "response")
#
#   # Calculate ROC
#   roc = roc(as.numeric(test_data$thyroid), as.numeric(predictions))
#   auc = auc(roc)
#
#   # Step 6: Update best parameters if necessary
#   if (auc > best_auc) {
#     best_auc = auc
#     best_params = param_grid[i, ]
```

```
#   }
# }
#
# # Step 7: Train final model with best parameters
# svmFullModel = svm(thyroid ~ ., data = train_data,
#                    kernel = best_params$kernel,
#                    cost = best_params$C)
#
# # Fit the backward stepwise model
# stepModelSVM = svmFullModel |> stepAIC(direction = "backward")
```

9.3.2 SVM - REF()

```
# ################
# # RFE  # Error in { : task 1 failed - "dim(X) must have a positive length"
# ################
# #Predictor variables

# # ## Use function as 'caretFuncs' for RFE to get ROC as summary metrics.
# # # and assign twoClassSummary function summary function of caretFuncs.
#    caretFuncs$summary = twoClassSummary
#
# # # # RFE Train with Metric: ROC and CV : 5
# svm_RfeCtrl = rfeControl(functions = caretFuncs,
#                      rerank = TRUE,
#                      method="cv",
#                      number = 5,
#                      #repeats = 5,
#                      saveDetails = TRUE,
#                      returnResamp = "all")
#
#
# set.seed(12356)
# svm_RfeTrain = rfe(
#             #x = x,
#             #y = y,
#             thyroid ~ . ,
#             data = thyroidData,
#             sizes = c(1:12),
#             preProcess = c("center", "scale"),
#             method = "svmRadial",
#             metric = "ROC",
#             #trControl = fsCtrl,
#             rfeControl = svm_RfeCtrl)
```