horizontal line

**Moises Marin Martinez**

Student ID A20349918  
BAN 5753 Online

mmarinm@okstate.edu

1. **Use a Data Partition node (70 for training, 30 for validation and random seed of 12345). Attach an Impute node (use default imputation is SAS but create unique indicator for capturing missing value replacement and change its role to Input). Now, build a logistic regression model to Predict Target(Enroll). Use Stepwise (not Forward) selection and use defaults for selection. Report results from this model and comment on how well does this model work? (1 point)**

This is the diagram created.

Diagram

Description automatically generated

The model’s fit statistics are:

Graphical user interface, application, table, Excel

Description automatically generated

The model is overfitted, it has a misclassification rate of 0.07.

1. **Add a Decision Tree node after the data partition node and build the best decision tree model you can. Explain this tree model. Compare your model to the one in #1. Which is better? (1 point)**

This is the diagram after adding the decision tree

Diagram, text

Description automatically generated

These are the fit statistics comparing the decision tree and the logistic regression model.

Table

Description automatically generated

This is the tree created, it uses satscore at the root level and has 5 levels deep.

Diagram

Description automatically generated

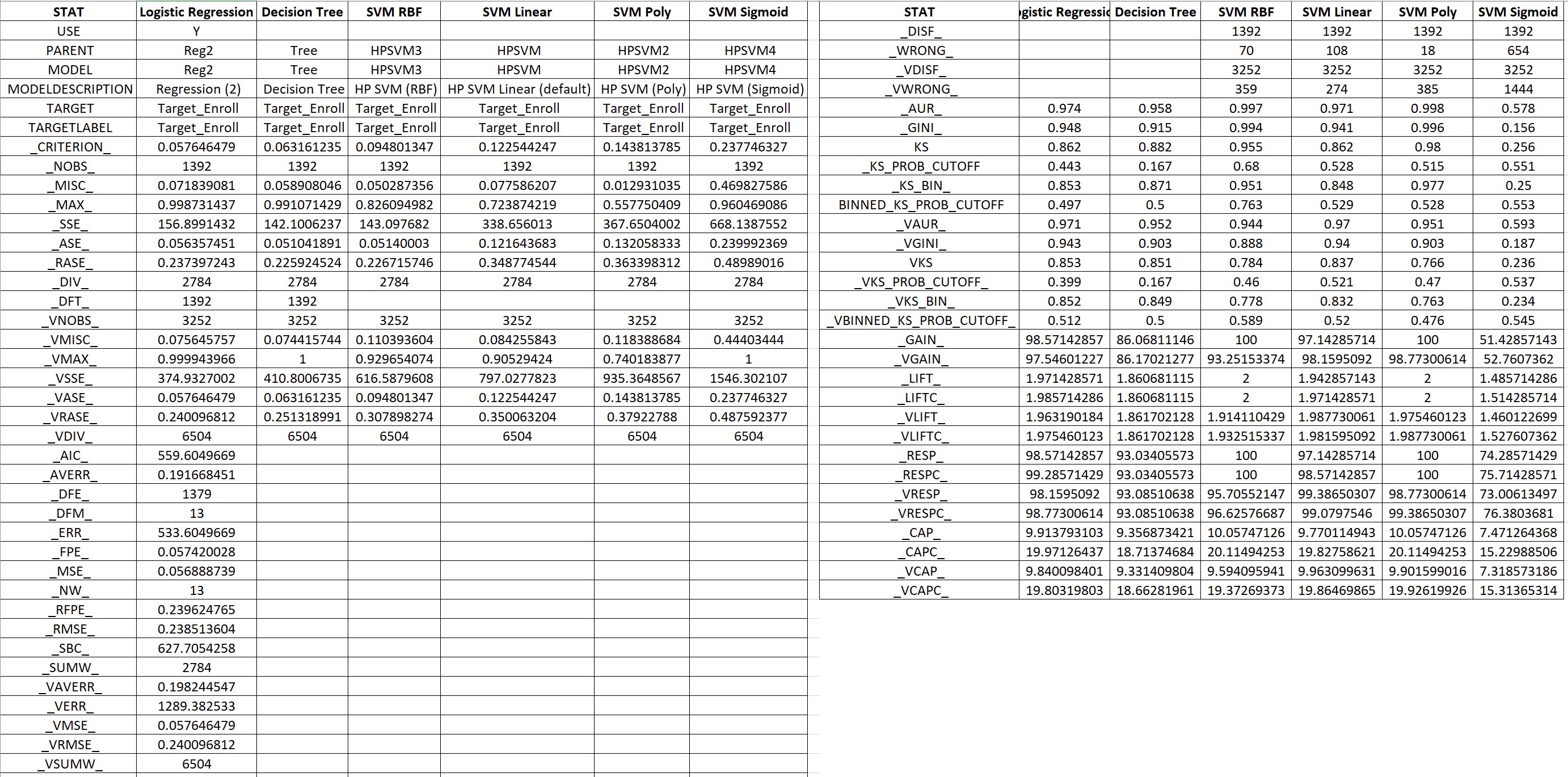
The decision tree has a slightly lower misclassification rate, it is the model selected by EM. Both models are overfitted.

1. **Now run another Logistic regression (add after Impute node) with backward elimination (with probability of entry as 1 and probability of stay as 0.05). Connect several SVM nodes with this backward elimination node and use the different functions as shown in class. Comment on which SVM seems to perform the best. Explain the best performing SVM model and compare its performance with the model in #1 and #2. (2 points)**

This is the diagram after adding several SVM nodes and backward elimination node.

Diagram

Description automatically generated

These are the fit statistics of all the models.

The model chosen by EM is the logistic regression, out of the SVM models the best one is the SVM with the polynomial function, it has the lowest misclassification rate at 0.012, the sigmoid function has the highest misclassification rate at 0.469.

These are the SVM polynomial classification chart and cumulative lift chart.

Bar chart

Description automatically generatedChart, line chart

Description automatically generated

The SVM polynomial fit statistics are shown in the next table, with the C value (Bias) highlighted.

Table

Description automatically generated

1. **Now add two versions of Random Forest models as demonstrated in the lecture using same settings as in the lecture after partition node. Which of the two Random Forest model did better?. Did this model perform better or worse compared to model in #1 - #3? Explain. (2 points)**

This is the diagram created with the two random forest nodes.

Diagram

Description automatically generated with low confidence

The random forest that did better was the larger Forest with a misclassification rate of 0.060, it has a higher misclassification rate than the decision tree and lower than the logistic regression model.

Table

Description automatically generated

1. **Use Gradient Boosting as a model – add after partition node. Play with the 3 options mentioned din the lecture to come up with the best model. How well did this model perform compared to models in #1-#4. Comment. (1 point)**

This is the diagram with the Gradient Boosting node.

Diagram

Description automatically generated

The Gradient Boosting model has a higher misclassification rate than the logistic regression, lower than the Forests and the decision tree.

Table

Description automatically generated

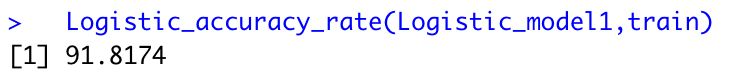
1. Repeat #1-#5 using your preferred open source codes. You may use *Python* in answering this question. *Please note that you may need to impute before splitting data in Python to make it easier on you (but theoretically speaking, it is not the correct approach!). Also, if you are using sklearn package, use L1/L2 regularization instead of stepwise option*. Gradient Boosting and Random Forest in Python does not have same set of options as SAS EM. So, you need to figure out what the common options and use these options to build the models Python. Make sure you mention what options you used in your write up. Also, the compare and contrast you results from open-source with those from SAS? What are similar and what are different between these results? How do these two differ in terms of ease of use in analyzing and comparing multiple models? Comment.
   1. Turn-in written answers in a Word document for Q1-Q6. (**1 point**)
   2. But also include a self-explainable Jupyter notebook with comments (if using Python). We will run your codes with the data and if the codes don’t run, you will not get credits. You will also not get full credits, if codes do not have comments. (**2 points**)

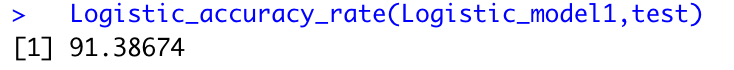
The Logistic Regression Model created with R code has these statistics.

Table

Description automatically generated

The accuracy rate of the model is:





The second model, the decision tree, has a slightly better accuracy rate:

Graphical user interface, text, application

Description automatically generated

This is the decision tree that was created:

Diagram

Description automatically generated

The SVM model with linear kernel has these accuracy results:

Graphical user interface, text

Description automatically generated with medium confidence

The second SVM model, the SVM with polynomial kernel has these accuracy results:

Graphical user interface, text

Description automatically generated

The SVM with radial kernel has these accuracy results:

Graphical user interface

Description automatically generated with medium confidence

The SVM with Sigmoid kernel has these accuracy results:

Graphical user interface

Description automatically generated with medium confidence

The SVM with the radial kernel had the best accuracy results, this is different from the results found in SAS where the polynomial kernel had the lowest misclassification rate. This can be because the model in R was created with the variables selected in the stepwise regression.

The random forest has this accuracy.

Text

Description automatically generated

The gradient boosting model has this accuracy and confusion matrix.

Table

Description automatically generated with low confidence

This table has the misclassification rates of all the models created, misclassification rate was calculated as 1- accuracy of train data.

|  |  |  |
| --- | --- | --- |
| # | Model | Misclassification rate |
| 1 | Logistic Regression (SAS) | 0.071 |
| 2 | Decision Tree (SAS) | 0.058 |
| 3 | SVM Polynomial (SAS) | 0.012 |
| 4 | SVM Linear (SAS) | 0.077 |
| 5 | SVM Radial (SAS) | 0.050 |
| 6 | SVM Sigmoid (SAS) | 0.469 |
| 7 | Random Forest (SAS) | 0.060 |
| 8 | Gradient Boosting (SAS) | 0.051 |
| 9 | Logistic Regression (R) | 0.087 |
| 10 | Decision Tree (R) | 0.08 |
| 11 | SVM Polynomial (R) | 0.248 |
| 12 | SVM Linear (R) | 0.263 |
| 13 | SVM Radial (R) | 0.209 |
| 14 | SVM Sigmoid (R) | 0.334 |
| 15 | Random Forest (R) | 0.064 |
| 16 | Gradient Boosting (R) | 0.066 |

For the R code, see the file submitted with this word document.