SMU®

Data Science Machine Learning DS7331.402

Amber Burnett
Lance Dacy
Shawn Jung
Jeremy Otsap

Mini-Lab
Logistic Regression and SVMs

February 16, 2020

Table of Contents

SVM and Logistic Regression Modeling

Classification problems revolve around being able to predict what category a feature falls into within the dataset. The common tools for classification problems are:

- Percent correction classification (PCC): measures overall accuracy. Every error has the same weight.
- Confusion matrix: also measures accuracy but distinguished between errors, i.e false positives, false negatives and correct predictions.
- Area Under the ROC Curve (AUC – ROC): is one of the most widely used metrics for evaluation. Popular because it ranks the positive predictions higher than the negative. Also, ROC curve is independent of the change in proportion of responders.

This project focused on the use of Sub-vector Machine (SVM) as well as Logistic Regression Modeling using PCC, Confusion Matrix, and AUC-ROC. The Data Science team has made their coding available. Simply click the links for Logistic Regression or SVM 1 / SVM 2 to follow along.

Executive Summary

Figure 1 outlines the summary that the Logistic Regression Model performed slightly better than the SVM in terms of accuracy and definitely better in speed of execution. However, the SVM performed slightly better than the Logistic Model in terms of Specificity.

While accuracy is an important metric to consider, sensitivity and specificity must be taken into account to ensure the model performs beyond just one class and is robust. The ultimate goal is to ensure the model refrains from performing poorly with the introduction of new data when it is deployed.

Figure 1: Model Summary

Metrics	Logistic Regression	Sub-vector-machine
Accuracy	82%	79%
Sensitivity	56%	45%
Specificity	90%	91%
ROC-AUC	81%	68%

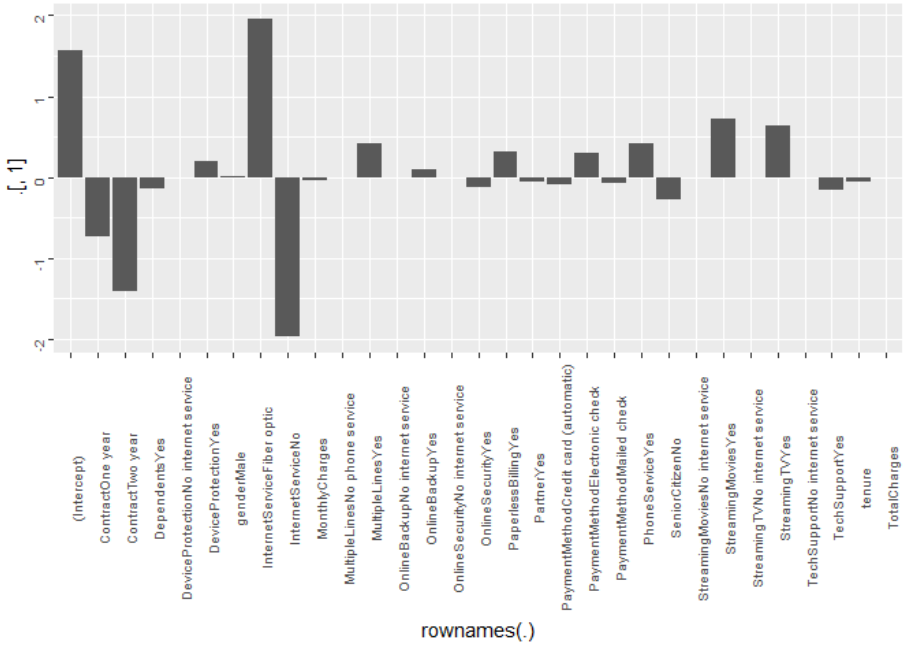
Interpreting the Models

While the details of the model performance and execution can be found in the Executive Summary, the various figures below will provide insight into the path followed by the Data Science team in exploring the data set and various models. Inference of the models will be discussed as well.

Full Logistic Regression Model

The initial goal was to determine the most influential factors in the data which end up being the type of Internet Service and the type of Contract. In this case, the Full Logistic Model unfortunately is better at determining loyal customers (or the probability when a customer won't leave) which is a bit counter to the original question at play (the probability of when a customer will leave, attrition).

Figure 2: Visualization of the Coefficients



The tools used automatically processes the 1-bit encoding of factor variables. This model however had some issues due to the fact that there is a high amount of correlation. For example, if a customer has no Internet service, they will always be absent of the ancillary services like Online Security or Online Backup.

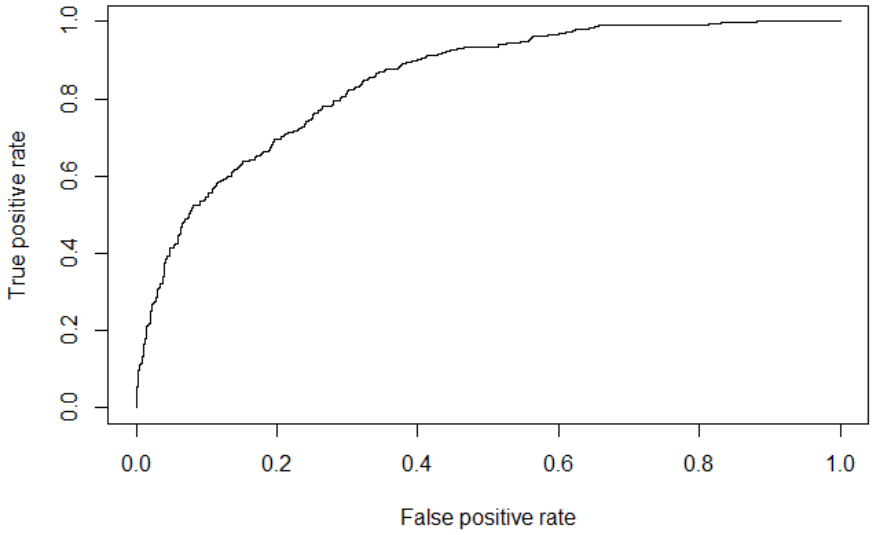
Despite the correlation issue, there are some parameters that emerge as more significant influencers in our model (all of these are deemed statistically significant):

- SeniorCitizen
- tenure
- MultipleLines
- InternetService
- Contract
- PaperlessBilling
- PaymentMethod
- TotalCharges

Total Charges, tenure, and contract variables are more significant than the other variables due to their economic impact. The longer the customers stay with the phone company, the more they pay the phone company. The total charges will naturally remain higher for the loyal customers as their tenure in their contract status is also longer than those who "Churn".

In comparing the two models, the ROC curve tends to provide great insight into the accuracy and robustness of the model. Figure 3 compares the number of times the model predicts a yes or no comparison to the actual values. Accuracy is thus estimated at 82%.

Figure 3: ROC Curve for Full Logistic Regression Model



Along with visualizing the ROC Chart, Figure 4 highlights a confusion matrix which will assist in determining the performance of the Full Model.

Figure 4: Confusion Matrix for the Full Logistic Model

Prediction	Yes	No
Yes	208	108
No	165	926

Due to the fact that the model is predominantly concerned with categorical predictors, it's not as obvious to identify potential correlation between them; as compared to continuous variables that allow the ability to calculate the Variance Inflation Factor (VIF).

Given the data model has over twenty (20) predictors, overfitting becomes a concern. This highlights the chance that the model performs well on the training data set (the one used to construct the model), but then performs poorly on new customers. Naturally, the goal is to accurately predict the customer's attrition probability in either case.

There are tools used to help mitigate these concerns. The model should remain as simple as possible; leaving only the most influential predictors. In order to help refine the Full Logistic Model, the tools explored were:

- Feature Selection
- Regularized or Shrinkage (to zero)
- Manually Adjusted

Stepwise Regression

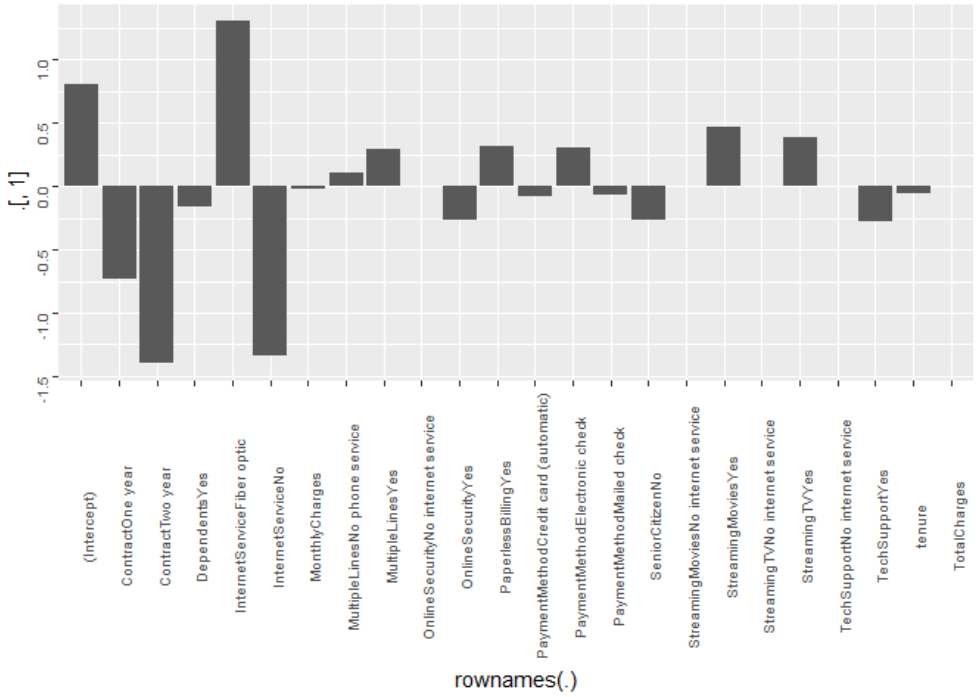
Stepwise regression is a method of fitting regression models in which the choice of predictive variables is carried out by an automated procedure.

In each step, a variable is considered for addition to or subtraction from the set of explanatory variables based on some pre-specified criterion.

Figure 5 will outline the AIC based Stepwise Model recommendation of the following significant factors:

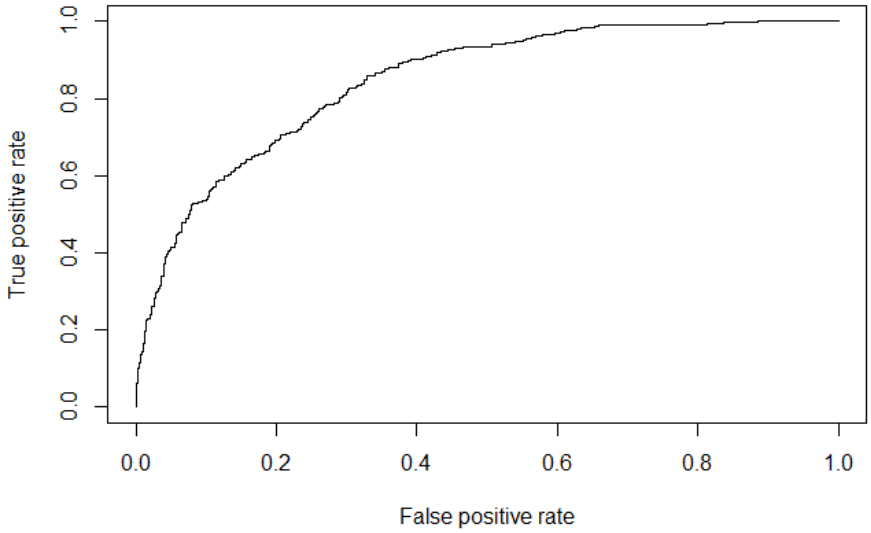
- SeniorCitizen
- tenure
- MultipleLines
- InternetService
- OnlineSecurity
- TechSupport
- StreamingTV
- StreamingMovies
- Contract
- PaperlessBilling
- PaymentMethod
- TotalCharges

Figure 5: Stepwise Model Coefficients



Using the Stepwise technique, the ROC curve can be calculated (displayed in Figure 6). The accuracy is still hovering around 82%.

Figure 6: Stepwise ROC Curve



The specificity (essentially predicting negative outcomes) is still around 90%. And the sensitivity, predicting positive outcomes, is slightly lower now at 59%

Figure 7: Confusion Matrix for the Stepwise Regression

Prediction	Yes	No
Yes	210	110
No	163	924

Lasso and Ridge Penalized Regression

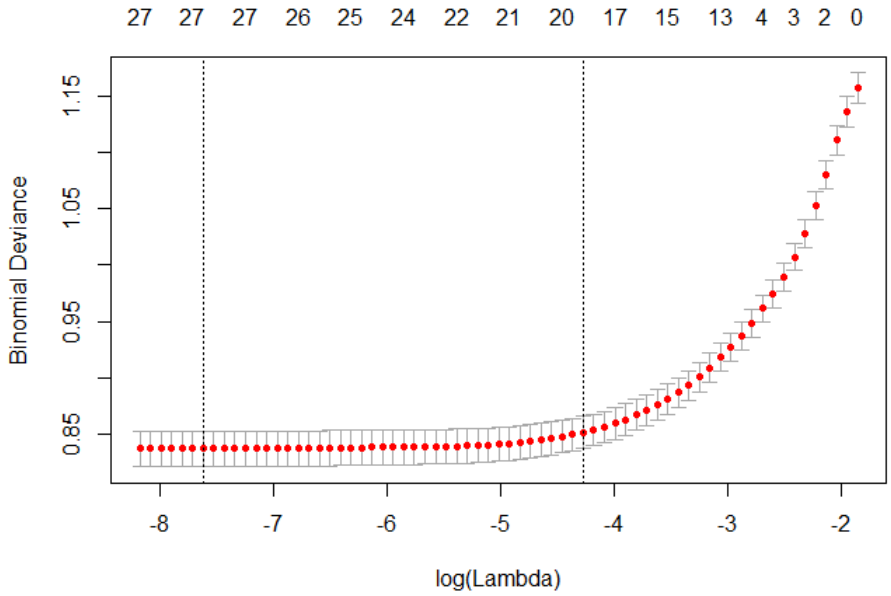
Rather than removing predictors as indicated in the Stepwise model, the goal of Lasso/Ridge is to introduce a constraint in the training process that shrinks the estimated coefficients.

Ridge Regression introduces bias in order to reduce the model's variance. It then tries to minimize the sum of the Residual Sum-of-Squares (RSS) by shrinking the variables to approximately zero.

Lasso Regression is similar to Ridge, however it involves minimizing the sum of the absolute values of the coefficients, pushing the least significant ones closer to zero. Unlike Ridge, the Lasso Penalty will actually push them all the way to zero.

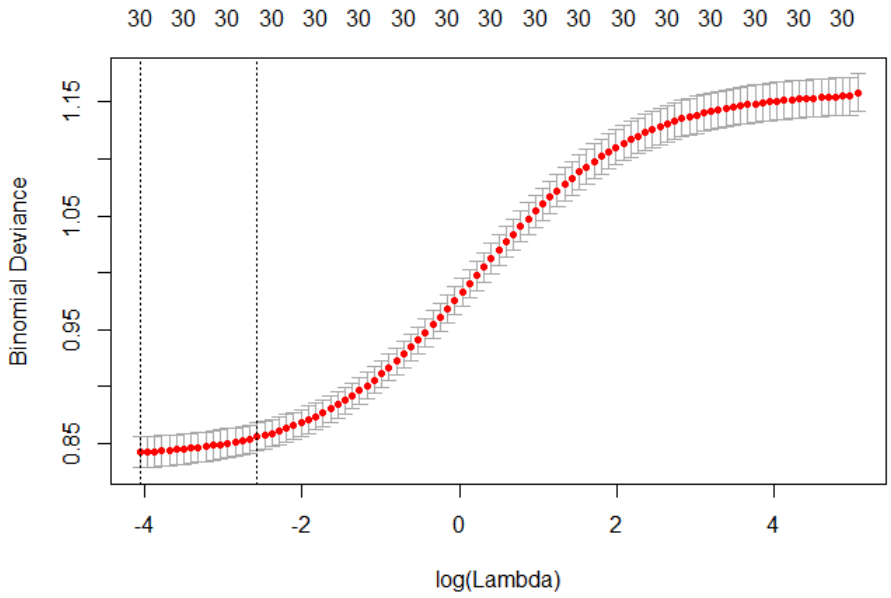
Both of these rely on a constant lambda which acts as a tuning parameter; affecting the size of the penalty. By affecting the size of the penalty, the coefficients's move closer to zero. A more detailed view can be seen in Figure 8 below.

Figure 8: Plot of Lasso Model Lambda



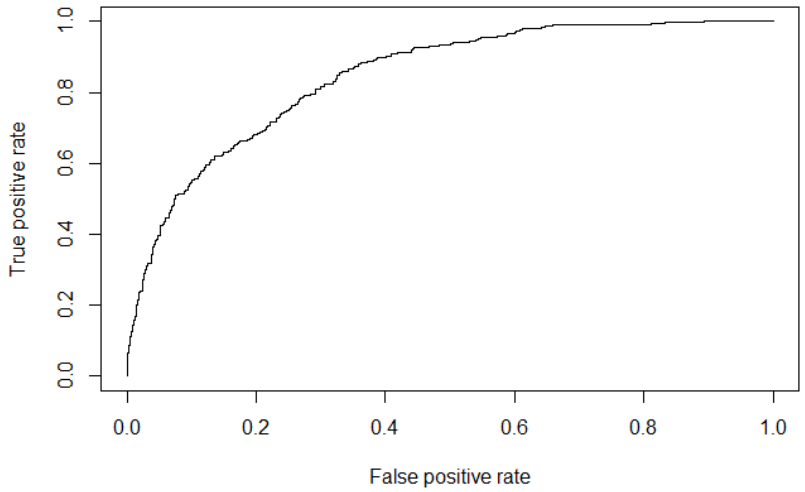
While Lambda is typically optimized by the computer, it does show how much the coefficients can be contained while maximizing on accuracy. The lambda model tends to evaluate accuracy alone rather than focusing on specificity and/or sensitivity.

Figure 9: Plot of Ridge Model Lambda



In Figure 8 and 9, the first dotted line represents lambda with the smallest MSE and the second represents with an MSE within 1 standard-error of the minimum MSE. Balancing these comparisons with an ROC curve, Figure 10 shows that the model is still very closely aligned with the Full Regression Model.

Figure 10: ROC Curve for Lasso Model with Optimized Lambda



Comparing the number of times the model predicts an accurate result, either "Yes" or "No" in comparison to the actual values, we can see we have a slightly lower accuracy of approximately 82%.

Figure 11: Confusion Matrix for the Lasso Model

Prediction	Yes	No
Yes	207	108
No	166	926

Support-vector-machine Model

The Support-vector-machine model is designed for the supervised learning environment as in the Logistic Regression Model. The SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

The model will employ an automated hyper parameter list generation to make C value candidates. C values are a regularization parameter that controls the trade off between achieving a low training error and low testing error. It will then compare accuracy among different SVM kernel methods:

- Linear
- Polynomial
- Gaussian (RBF)
- Sigmoid

Sigmoid Kernel

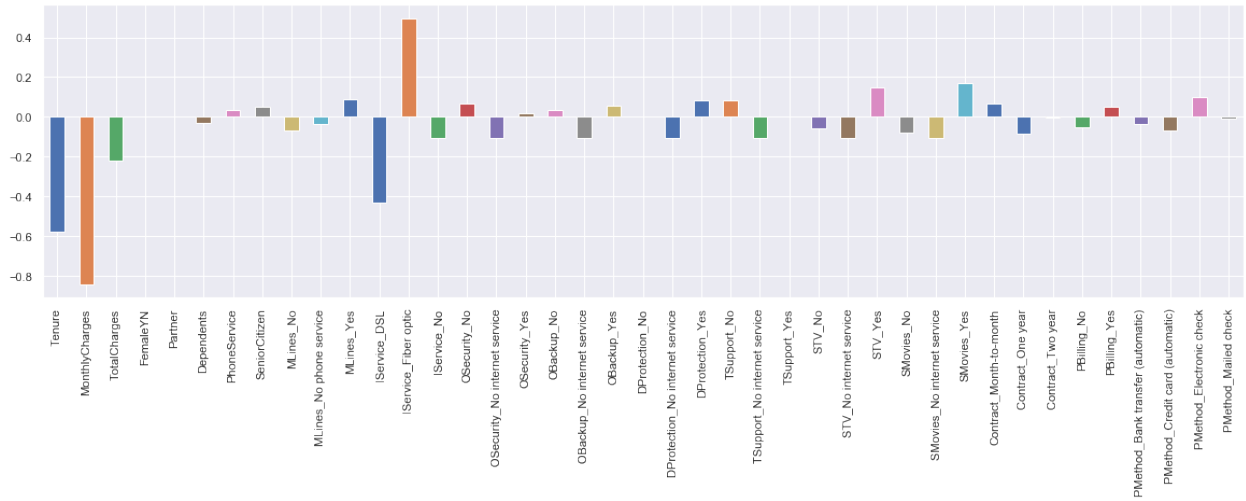
The automated model generated the sigmoid kernel with a suggested accuracy score and weights listed below:

- Accuracy Score: 0.802484
- Using: "C" with grid result of 0.04642 with Sigmoid Kernel

Linear Kernel

Forcing a Linear Kernel result in the parameter list generation for weighted values in Figure 12 below.

Figure 12: Weighted Values from SVM-Linear Kernel Training



Gaussian (RBF) Kernel

In order to avoid data snooping by the data science team, the train data was chosen to be scaled to keep the test data in-tact. The automated model generated the RBF kernel with a suggested accuracy score and weights:

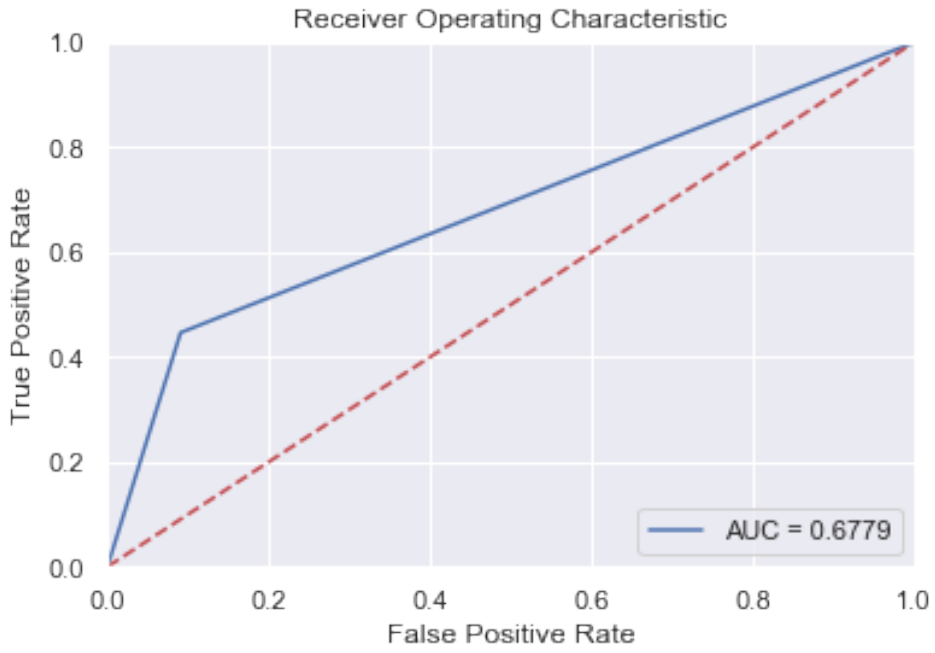
- Accuracy Score: 0.804969
- Using: "C" with grid result of 0.59948 with RBF Kernel

Summary of SVM Performance

The training loop took 14 minutes (828 seconds). Given that the Logistic Regression model only took 38 seconds a conclusion can be made that SVM is definitely not as fast as the traditional Logistic Regression Model if automated. Naturally, the Logistic Regression Model is a bit more manual on the front-end, but could be faster on deployment.

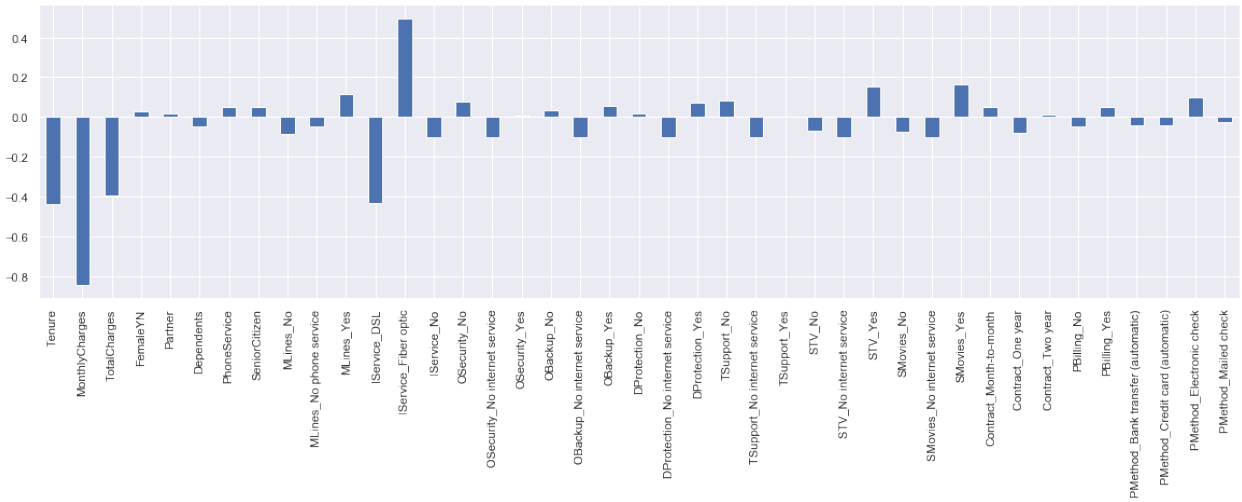
Final fitting of the model provided AUC of .06779 and can be evaluated with Figure 13 ROC.

Figure 13: ROC with Final Fitting



The final feature weights can be viewed in Figure 14.

Figure 14: Weighted Features



Comparison of the Models

The data science team experimented with the two (2) different approaches of SVM and Logistic Regression. Each one of them have their advantages and disadvantages. Myriad variables must be taken into account when selecting the final model for deployment.

Logistic Regression

Logistic regression is designed to predict the relationship between one binary variable (in this case "Churn"). The predicted or dependent variable is described by the sum of all features selected or independent variables multiplied by some weighted value that tends to be unique per variable.

The final bias is then added to the total. This last step typically separates regression from logistic regression by adding a log-odds (logit) function to the output in order to scale the results to the range of 0 and 1 for the true/false binary variable.

Given the manual aspect of choosing variables based on their effect on the model, Logistic Regression tends to be a lot of work in the beginning to finalize the model. This aspect allows the data scientist to have some ability and discernment when interpreting the coefficients. A human is much more capable at determining the "why" for the model than the machine. A human can then evaluate each coefficient to see how strongly it influences the model and visualize the probability for each observation; not simply looking at a 0 or 1 (yes or no).

The analogy can be viewed as a music studio mixing engineer. Visualize a mixing console as in Figure 15. Logistic Regression is comparable to this type of gear in that the engineer has complete control over all the variables to fine-tune sounds most people might not even hear. While SVM has less of this control (but more automated) as in Figure 16. Logistic Regression tends to be faster with better tuning options for multiple classification variables.

Figure 15: Logistic Regression Control (example)

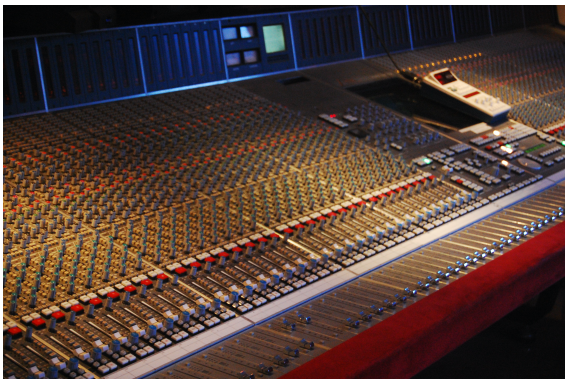


Figure 16: SVM Control (example)

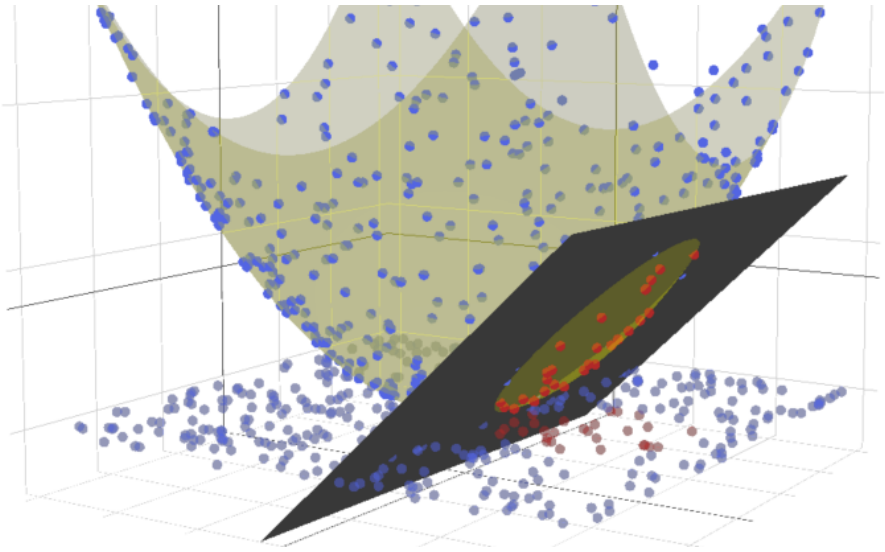
Support-vector Machine

In SVM, the goal is to separate two groups of the data points with boundary and the maximum distance between the two data sets and that boundary. The boundary is typically referred to as a hyperplane. If the number of input features is 2, then the hyperplane is just a line. If it is 3 or more then the hyperplane becomes a two-dimensional plane. Anything beyond 3 is very difficult to imagine (graphically).

While SVM is drastically more automated, there tends to be less options for tuning (thus less control from a scientist stand-point). The machine "robotically" tunes the model and usually gets very close, but the time to separate the data sets can be drastically more time than experienced in Logistic Regression. In addition, SVM is absent of probabilistic comparisons that allow for adjustment in a higher sensitive and specificity range. In Logistic Regression, the option is there to fine-tune .6 to .4 (and others).

In Figure 17 below, "Everybody's Favorite Data Blog" has shown an example of what multiple features can look like in a hyperplane. SVM appears to degrade in performance when more than a few features are being analyzed for learning.

Figure 17: SVM Hyperplane (example)



http://efavdb.com/svm-classification/

An advantage however, that is consistently heard around the community is that

> *-"SVMs are among the best (and many believe are indeed the best) 'off-the-shelf' supervised learning algorithms."*

As with most "off-the-shelf" products, customization is lost; which might be fine. It largely depends on the current needs and humans are more equipped to make that judgment.