Chapter 1: Using Supervised Learning for Binary Prediction of Jitter Events

Author: Vikram Singh (Student ID: 3032166489)

Introduction:

Stanford Linear Accelerator (SLAC) is incentivized to maximize linear particle accelerator (linac) uptime for two vital and related reasons.

First, the time elapsed from application submission for use of the linac to actual use of the linac ranges between two and three years. When a research group is granted linac access, they are limited to less than 24 hours of linac time over the course of a single calendar week. If the linac experiences downtime during a research group's allotted period, additional time cannot be granted given commitments already made to upcoming research groups. Thus, even downtime limited to hours can cause substantial detriment to research.

Second, the consistent performance of the linac inherently application volume for research at SLAC versus competing institutions. Application volume from research groups is directly tied to the funding the Department of Energy (DOE) awards SLAC on an annual basis; thus, optimizing linac performance will lead to additional SLAC funding.

The linac is powered by a series of RF stations. When the linac experiences downtime, it is often because several RF stations fail in parallel. To maximize overall linac uptime, an intuitive upstream objective is to minimize downtime of each RF station by training algorithms that can predict downtime of an RF station before it occurs. During exploratory sessions with SLAC operations scientists during which domain knowledge was explored, extended periods of high-value phase jitter within an RF station was identified as highly associated with RF station downtime. Thus, the goal of this project track was to train an algorithm that can effectively predict upcoming phase jitter within an RF station.

Definitions:

- RF station - Responsible for emitting bursts of high voltage to power linac. Each station is comprised of many complex, high-powered, expensive components.

- Attribute data: A data type / measurement that describes the condition of one or many components of an RF station.
- Jitter Event: A period of time for which extended periods of high-value phase jitter are observed within an RF station.

Discovery:

To build a predictive model that can predict a jitter event within an RF station, raw data streams that to be used for response variable ('jitter event') and feature variable generation was accessed.

SLAC has built an advanced data architecture for collecting Attribute data streams from various hardware components and storing it within servers that are collectively named the 'EPICS Network'. Most Attribute data streams produce a new data record every one to five seconds. Assuming all attributes are recorded every five seconds, this amounts to ~6.31 million records per Attribute annually. There are ~65 Attributes for which data is collected and 81 RF stations; when summed, this suggests that over 33.2 billion data records are flowing into the EPICS Network annually. This data volume requires that old data be archived in a scalable database - SLAC accomplishes this by storing historical data records in its 'Archiver' system. The Archiver system includes a Python-based API for which data for specific Attributes, RF Stations, and time periods can be queried.

Accessing all Attribute data for all RF stations was not feasible due to the data volume nor was it necessary for two reasons. First, the domain experts believe that different RF stations generally behave in similar ways over the stations' lifetimes. This allowed us to select a subset of RF stations to use for algorithm development without compromising the applicability of a model across all RF stations. Ultimately, three RF stations were chosen. Second, domain experts were able to eliminate Attributes that will certainly not be relevant to a 'jitter event', leaving eight Attributes for which to focus. Data was limited to three fiscal years (2015-2017).

Following data extraction, data was uploaded to a cloud-based data warehouse instance in order to allow for computationally efficient access and analysis of the data. Querying data via the Archiver API and the setup of the data warehouse instance were led by teammates.

Wrangling / Profiling

In order to generate a data set appropriate for supervised learning, significant efforts were spent developing both the appropriate response variables and feature variables that could be used to predict the responses. To that end, this section will be divided into 'Response Variables' and 'Feature Variables' sections.

Response Variables

Exploratory Analysis: The Attribute that measures phase jitter is labeled PJTN. Exploratory analysis on PJTN was conducted to answer the following questions in this order:

- *What are the ranges of PJTN values?* After presented the results below, domain experts at SLAC determined that PJTN values greater than or equal to 0.15 should be considered 'jitter records'.

Min	1st quartile	Median	Mean	3rd quartile	Max
0	0.08	0.1	0.13	0.13	11.25

- *Does the distribution of 'jitter records' differ by hour and/or by day of week?* Total jitter records were aggregated over the course of 2015-2017 for all RF stations. Figure 1 suggests there are less jitter records on Wednesdays. The SLAC team confirmed that maintenance is often done on Wednesdays, potentially explaining the reduction in jitter. Figure 2 suggests that jitter records fluctuate throughout the day, trending up during the night hours. A chi-square goodness of fit test was applied to both day-of-week and hour-of-day distributions, with the null hypothesis being that jitter record distribution should be even throughout each day-of-week and hour-of-day. In both cases, the null hypothesis was rejected using a .01 statistical significance threshold, suggesting that these variables should be use as features during model development. Environmental factors (e.g. cool temperature during the night time) that are correlated with hour of day will also be captured in feature generation against Attributes.

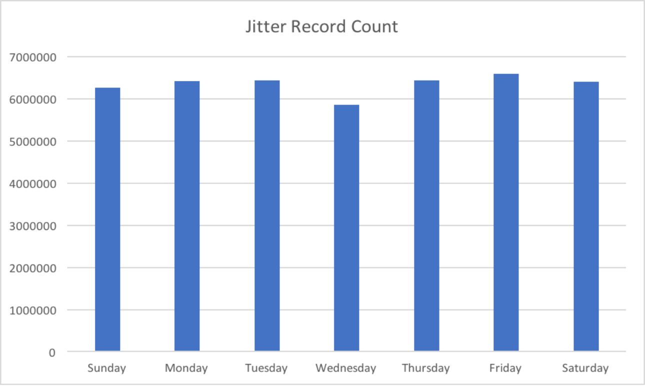


Figure 1 - Jitter records count by day of week

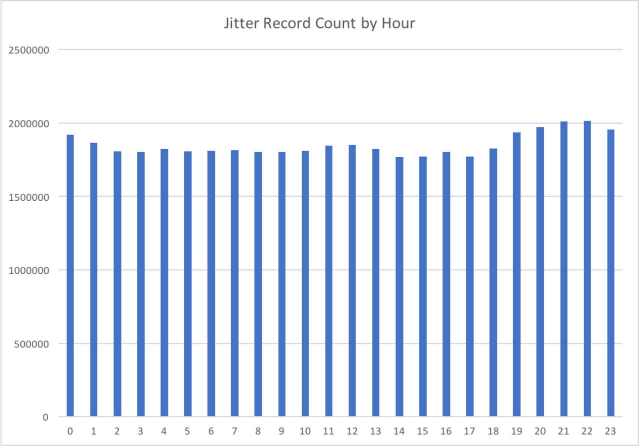


Figure 2 - Jitter records count by hour of day

- *Does the frequency of 'jitter records' change over time? Does the frequency of 'jitter records' differ across RF stations or is it interdependent?* Figure 3 charts the jitter record frequency per month across the three considered RF stations. Per RF station, jitter record frequency can vary dramatically month over month. This suggests that using data from feature variables that aggregate previous week's or month's of Attribute data from an RF station will likely be ineffective in prediction. In addition, jitter record frequency across RF stations appear to have no interdependence. This suggests that the algorithm, when in production, will need to make predictions using data specific to the RF station versus using a collection of data from related RF stations.

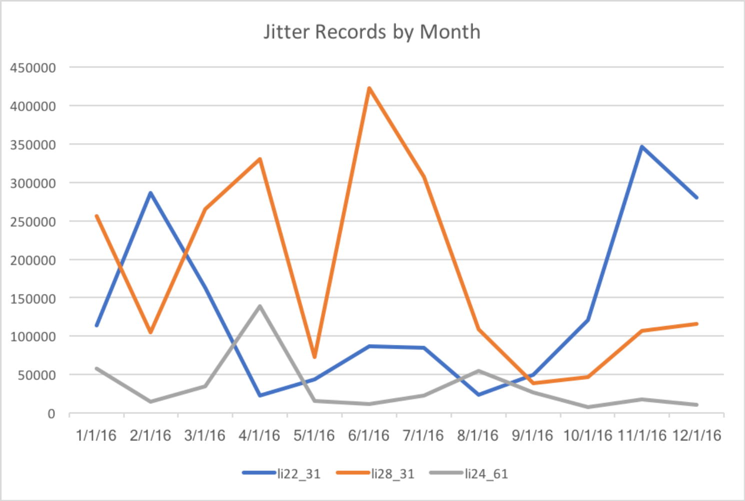


Figure 3 - Jitter records count by month per RF station

- *What percentage of PJTN records are jitter records?* Across the RF stations, the average percentage of PJTN records that qualify as jitter is 8% and the median is 5%.

Exploratory analysis informed next steps in response variable development. Given the sheer amount of jitter records on a given day (can reach tens of thousands), predicting which PJTN records will be jitter records in infeasible and also not fruitful for SLAC. Developing an intraday prediction algorithm that indicates whether there will be jitter records in a given hour is also frivolous because an RF station experiences modest jitter every hour (Figure 2). In addition, given jitter records make up ~5% of all records, class imbalance is intrinsic to the dataset. To mitigate these issues, logic was developed for identifying 'jitter events', defined as an extended period of time during which an RF station experiences jitter.

Developing 'Jitter Events': A generic function was developed that required the maximum allowable duration of an event to be defined. For example, assume the maximum duration is set to 60 minutes. For each calendar day, the first jitter record within the day will be considered the start of the first event. Any jitter record that occurred within 60 minutes of the first jitter record will be considered apart of the existing event. The first record that occurs more than 60 minutes after the initial jitter record will be considered the start of the next event. Each event contains all PJTN records occurring between the event start and end. Several descriptive attributes of the event are then generated, including average PTJN value and duration of the event. If the average PJTN value of the event is greater than or equal to 0.15, then it is considered a 'jitter event'. In order to cast a wide 'response variable' net, jitter events spanning 15, 60, 120, and 180 minutes were generated against all historical PJTN data.

Developing 'Non-Jitter Events': To complement the jitter event, which can serve as the positive response within a supervised learning algorithm, a negative response variable is needed. In this case, logic to define a 'non-jitter event' was developed. For each hour within a calendar day, if the average PJTN value was less than 0.15 then the hour was considered a 'non-jitter event', allowing a maximum of 24 non-jitter events on a single calendar day. The frequency of non-jitter events has an inverse correlation with the frequency of jitter events identified, suggesting that complementary logic is being used for identify jitter and non-jitter events.

Reduced Positive Response: Analysis of longitudinal jitter event distribution per RF station revealed that once jitter events occur with frequency for two to three consecutive days, there is a high likelihood that jitter event frequency will increase until jitter is essentially constant. Constant jitter experienced within an RF station suggests that certain component(s) of the RF station have failed and the RF station must be maintained. Once maintained, jitter event frequency approaches zero.

Persistent jitter within an RF station due to failed component(s) could prove problematic for algorithm development because a single failure, if not addressed quickly, will manifest as several, redundant positive response records within the data set used for machine learning. This could cause the algorithm to over fit for the Attribute(s) associated with the component(s) that have failed within the RF station.

To mitigate this issue, recursive logic was introduced to reduce the positive response rate by removing redundant positive responses likely caused by a persistent RF station issue. Within the data set, for the first calendar day for which a high frequency of jitter events occurred, a maximum of four consecutive calendar days worth of jitter events were allowed within the data set. If jitter events continued to occur beyond the four consecutive day, the assumption is the RF station has failed, and further jitter events are excluded. Once a day with a low frequency of jitter events occurs, the logic repeats until all calendar days between 2015-2017 have been processed.

Figure 4 demonstrates the jitter event count without normalization (Normal) and with reduced positive response (RPP). For months where the jitter event count is below ~200, RPP is essentially equivalent to Normal; however, when there is excessive jitter RPP continues to stay lower than ~200.

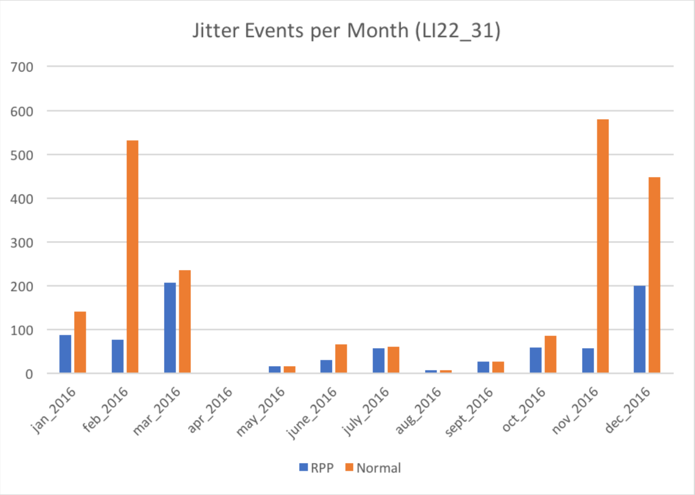


Figure 4 - Jitter events by month, normalized and Reduced Positive Response (RPP)

The response variable engineering led to a 4:1 ratio between negative and positive response within the data, a substantial improvement over the class imbalance displayed in the raw PJTN data (20:1 ratio). The ratio could be further improved through under sampling techniques, such as a random selection of records, on the negative response data[1].

Features Variables

Eight Attributes were chosen for developing feature variables to be used for response variable prediction. Attribute data was aggregated for the calendar day and features were developed against the calendar day, simulating a prediction job being run at midnight by using Attribute data collected over the last 24 hours. Feature variable development was an iterative process driven by a teammate.

The correlation amongst the eight Attributes used for feature generation, in addition to each Attribute's correlation to jitter, was explored. Within Figure 5, larger blue circles represent stronger positive correlation between Attributes and larger red circles represent stronger negative correlation. The most statistically powerful (but not significant) relationship ($p = 0.78$) exists between Sigma and Jitter, where Sigma is a measure of amplitude jitter from the modulator within the RF station. Domain experts suggested that high value Sigma is problematic and indicates an RF station that may need maintenance.

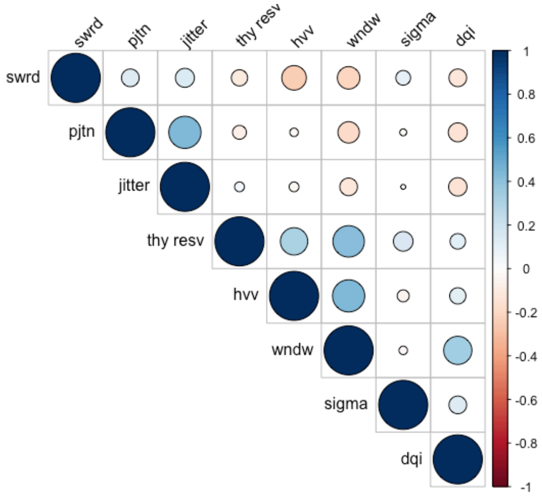


Figure 5 - Correlation amongst features and response variable (jitter). Larger blue circles represent higher positive correlations, larger red circles represent larger negative correlations.

Modeling

The goal was to predict whether an RF station would experience a jitter event or not on any given calendar day, a binary classification problem. The jitter event was defined as a 60-minute period during which the RF station experienced jitter. The reduced positive response data set was used. Each data set was split into training and test data sets at a 70/30 ratio and the training data set was used for algorithm development. Details on training steps by algorithm below:

Logistic Regression (LR): Several issues were encountered in the initial runs of LR. Issues were addressed in the following order and the model re-run after each issue was resolved.

1. *Coefficients not defined because of singularities.* This occurred due to features related to the Sigma Attribute being perfectly collinear. These were removed.
2. *When fitting model, numerically 0 or 1 occurred.* This indicates that there are certain feature(s), that when split at a threshold, perfectly predict the response. Studying the models revealed select WNDW Attribute feature coefficients defined as infinite, an indicator of perfect separation. After investigation, a combination of outlier WNDW values and feature design flaws was found to be unfairly influencing the model. Outliers were removed and the features subsequently recalculated.
3. *Algorithm did not converge.* This again indicates there are feature variables contributing to perfect separation. Analysis of coefficients indicated that PJTN-feature variables may be problematic, suggesting that using fifteen-minute jitter events to predict subsequent extended jitter events is

essentially equivalent to predicting the response variable with the response variable. To resolve this, PJTN feature variables were removed.

After the model was run, a collinearity check using the Variance Inflation Factor (VIF) metric was run and the 'rule of ten' was used to evaluate features. A VIF equaling ten indicates that the feature's coefficient is ten times greater than it would have been if it didn't display collinearity with another feature[2]. Five features being removed - all were Attribute averages that demonstrated collinearity with median metrics for the same Attributes.

CART: In order to optimize for model interpretability, k-fold cross validation was used to tune the complexity parameter (cp), a parameter that facilitates tree pruning by removing splits that do not sufficiently improve the model. A cp value of 0.015 was chosen. PJTN-related features were included in model training because the CART algorithm handled them well (unlike LR) and those features improved model performance.

Random Forest (RF): To optimize model performance, k-fold cross validation was used to tune the number of variables considered within each split of the tree (mtry). A mtry range of 1-16 was considered and an mtry value of 4 yielded the highest accuracy on the training set.

Results

A summary of results are below (OOS = Out of Sample/Test Set Prediction):

Logistic Regression	K-Fold (n=50) Accuracy	OOS Accuracy (p > .5) *	OOS TPR (p > .5)	OOS FPR (p > .5)	ROC AUC**	OOS TPR (p > .2)	OOS FPR (p > .2)
All Variables	.777	.757	.057	.030	0.66	.595	.470
Stepwise Variables	.788	.766	.066	.027	0.66	.586	.442

** Accuracy of model on test set*

***Area under curve for Receiver Operating Characteristic (ROC) curve*

Tree-based Models	K-Fold (n=50) Accuracy*	OOS Accuracy	OOS TPR	OOS FPR
CART	.842	.845	.471	.043
CART with Loss*	.830	.827	.619	.111
Random Forest	.857	.851	.553	.061

** Out of sample (OOS) model accuracy estimated via k-fold cross validation*

Out of sample baseline accuracy was 0.77 (where baseline model is predicting that an RF station will not experience a jitter event on a given day). Tree-based models slightly outperformed this baseline accuracy. The high baseline accuracy is due to class imbalance within the data (4:1 ratio of negative to positive responses) and shouldn't be used as the driving metric for model evaluation.

The SLAC operations team requires a model that does not compromise on a low False Positive Rate (FPR). If FPR is too high, the maintenance team will spend resources proactively maintaining RF stations when maintenance is not required, which may end up being less operationally effective than the current paradigm of responsive maintenance in which an RF station is serviced after it fails.

The ROC curves (Figure 5) for the Logistic Regression (LR), in which PJTN-features did not cooperate with model, suggest that the LR models should not be used given SLAC's goal.

The tree-based models performed substantially better than the LR models. The CART model demonstrated a .471 TPR and a .043 FPR, meaning RF stations that will experience jitter are identified nearly 50% of the time with seldom false positives. The additional benefit of the CART model is high interpretability. Random Forest (RF) outperformed the CART model with a TPR of .55 and a similar FPR, which is the expected outcome when comparing RF to CART because RF is the hybrid of a collection of trees versus a singular, best tree that CART produces[3].

A Loss function punishing False Negatives to False Positives at a ratio of 2:1 was introduced. This improve the TPR of the CART model to beyond .6 while limiting the FPR at ~.1, a ratio that could allow for meaningful improvements in efficiency for SLAC.

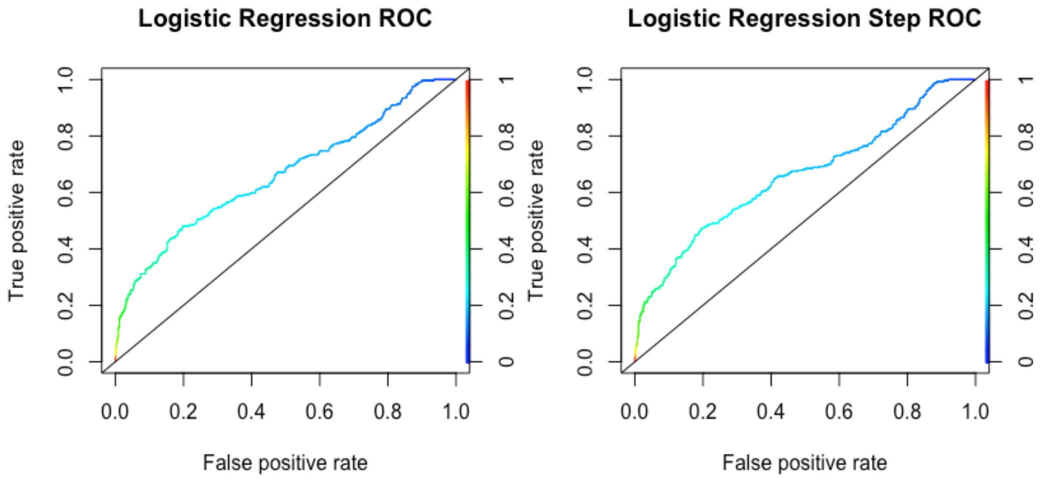


Figure 6 - ROC curves for Logistic Regression with/without Step

Reporting:

Exploratory analysis suggests that an RF station experiences jitter on an exponential curve. Initial jitter is an indicator that the RF station's health is deteriorating but there appears to be a window of time where maintenance can be done without the RF station experiencing actual downtime. With this being the case, a rule the SLAC operations team could adopt is simply to maintain an RF station if jitter is observed in the RF station in the prior one to two days. Simulating this heuristic shows a TPR of .22 and an overall accuracy of .66. The CART model (Figure 7) largely aligns with this logic - both features used within the CART model relate to jitter (PJTN), but is more granular and effective. The CART model reads as follows: If yesterday's average value of Phase Jitter records is greater than or equal to 13, and there is greater than or equal to twelve, fifteen-minute Jitter events yesterday, then the SLAC expect there to be at least one sixty-minute Jitter event within the RF station today.

The results of this analysis suggest that SLAC should do the following:

1. Run a nightly, batch data processing job on the PJTN data from each RF station to identify 'jitter events' using similar logic developed for this analysis.
2. If the SLAC operations team has a vested interest in using ML, adopt the CART model with a 2:1 Loss function. This should result in ~60% accuracy in identifying RF stations that are experiencing jitter events (TPR) with a relatively small FPR (~10%).

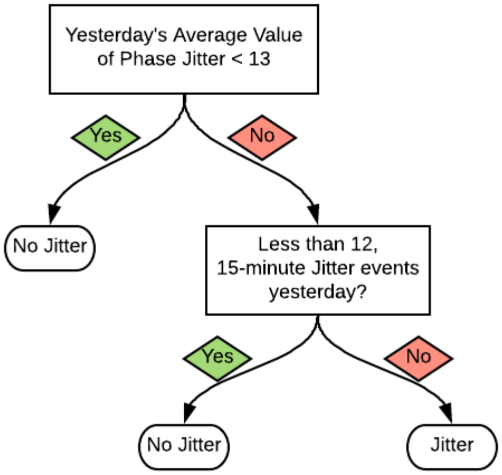

Yesterday's Average Value of Phase Jitter < 13
Yes
No
No Jitter
Less than 12, 15-minute Jitter events yesterday?
Yes
No
No Jitter
Jitter

Figure 7 - Decision Tree produced by CART model. If yesterday's average value of Phase Jitter records is greater than or equal to 13, and there is greater than or equal to twelve, fifteen-minute Jitter events yesterday, then expect there to be at least one sixty-minute Jitter event today.

References:

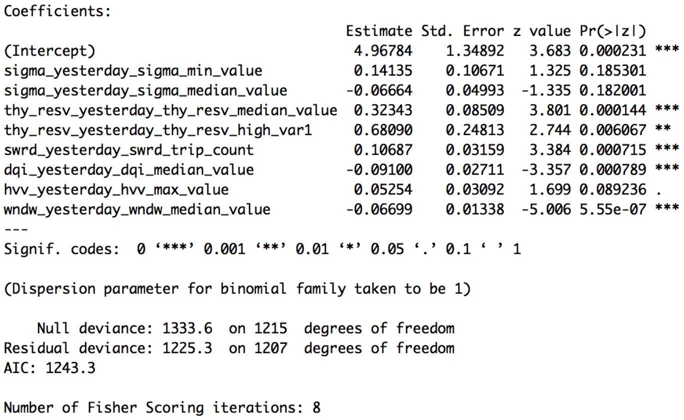
1. R. Longadge, S. Dongre, et. al., (2013). Class Imbalance Problem in Data Mining: Review. Retrieved from https://arxiv.org/pdf/1305.1707.pdf

2. R. O'Brien (2007). A Caution Regarding Rules of Thumb for Variance Inflation Factors. Retrieved from https://pdfs.semanticscholar.org/ed1f/4466a0982f3e8de202de01ecceb473d11893.pdf

3. J. Ali, R. Khan et. al., (2012). Random Forests and Decision Trees. Retrieved from https://pdfs.semanticscholar.org/959a/8e906ee26b940374b719253c8e188ed78fd3.pdf

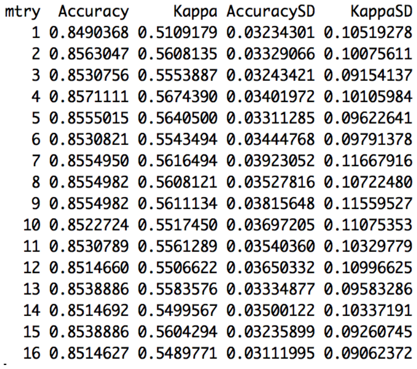
Appendix:

Logistic Regression



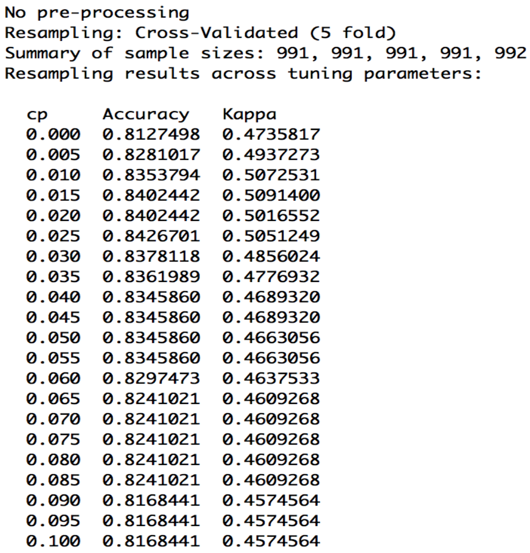
```
Coefficients:
                                         Estimate Std. Error z value Pr(>|z|)    
(Intercept)                               4.96784    1.34892   3.683 0.000231 ***
sigma_yesterday_sigma_min_value           0.14135    0.10671   1.325 0.185301    
sigma_yesterday_sigma_median_value       -0.06664    0.04993  -1.335 0.182001    
thy_resv_yesterday_thy_resv_median_value  0.32343    0.08509   3.801 0.000144 ***
thy_resv_yesterday_thy_resv_high_var1     0.68090    0.24813   2.744 0.006067 ** 
swrd_yesterday_swrd_trip_count            0.10687    0.03159   3.384 0.000715 ***
dqi_yesterday_dqi_median_value           -0.09100    0.02711  -3.357 0.000789 ***
hvv_yesterday_hvv_max_value               0.05254    0.03092   1.699 0.089236 .  
wndw_yesterday_wndw_median_value         -0.06699    0.01338  -5.006 5.55e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1333.6  on 1215  degrees of freedom
Residual deviance: 1225.3  on 1207  degrees of freedom
AIC: 1243.3

Number of Fisher Scoring iterations: 8
```

Random Forest



```
mtry  Accuracy     Kappa AccuracySD    KappaSD
   1 0.8490368 0.5109179 0.03234301 0.10519278
   2 0.8563047 0.5608135 0.03329066 0.10075611
   3 0.8530756 0.5553887 0.03243421 0.09154137
   4 0.8571111 0.5674390 0.03401972 0.10105984
   5 0.8555015 0.5640500 0.03311285 0.09622641
   6 0.8530821 0.5543494 0.03444768 0.09791378
   7 0.8554950 0.5616494 0.03923052 0.11667916
   8 0.8554982 0.5608121 0.03527816 0.10722480
   9 0.8554982 0.5611134 0.03815648 0.11559527
  10 0.8522724 0.5517450 0.03697205 0.11075353
  11 0.8530789 0.5561289 0.03540360 0.10329779
  12 0.8514660 0.5506622 0.03650332 0.10996625
  13 0.8538886 0.5583576 0.03334877 0.09583286
  14 0.8514692 0.5499567 0.03500122 0.10337191
  15 0.8538886 0.5604294 0.03235899 0.09260745
  16 0.8514627 0.5489771 0.03111995 0.09062372
```

CART



```
No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 991, 991, 991, 991, 992
Resampling results across tuning parameters:

  cp     Accuracy   Kappa
  0.000  0.8127498  0.4735817
  0.005  0.8281017  0.4937273
  0.010  0.8353794  0.5072531
  0.015  0.8402442  0.5091400
  0.020  0.8402442  0.5016552
  0.025  0.8426701  0.5051249
  0.030  0.8378118  0.4856024
  0.035  0.8361989  0.4776932
  0.040  0.8345860  0.4689320
  0.045  0.8345860  0.4689320
  0.050  0.8345860  0.4663056
  0.055  0.8345860  0.4663056
  0.060  0.8297473  0.4637533
  0.065  0.8241021  0.4609268
  0.070  0.8241021  0.4609268
  0.075  0.8241021  0.4609268
  0.080  0.8241021  0.4609268
  0.085  0.8241021  0.4609268
  0.090  0.8168441  0.4574564
  0.095  0.8168441  0.4574564
  0.100  0.8168441  0.4574564
```