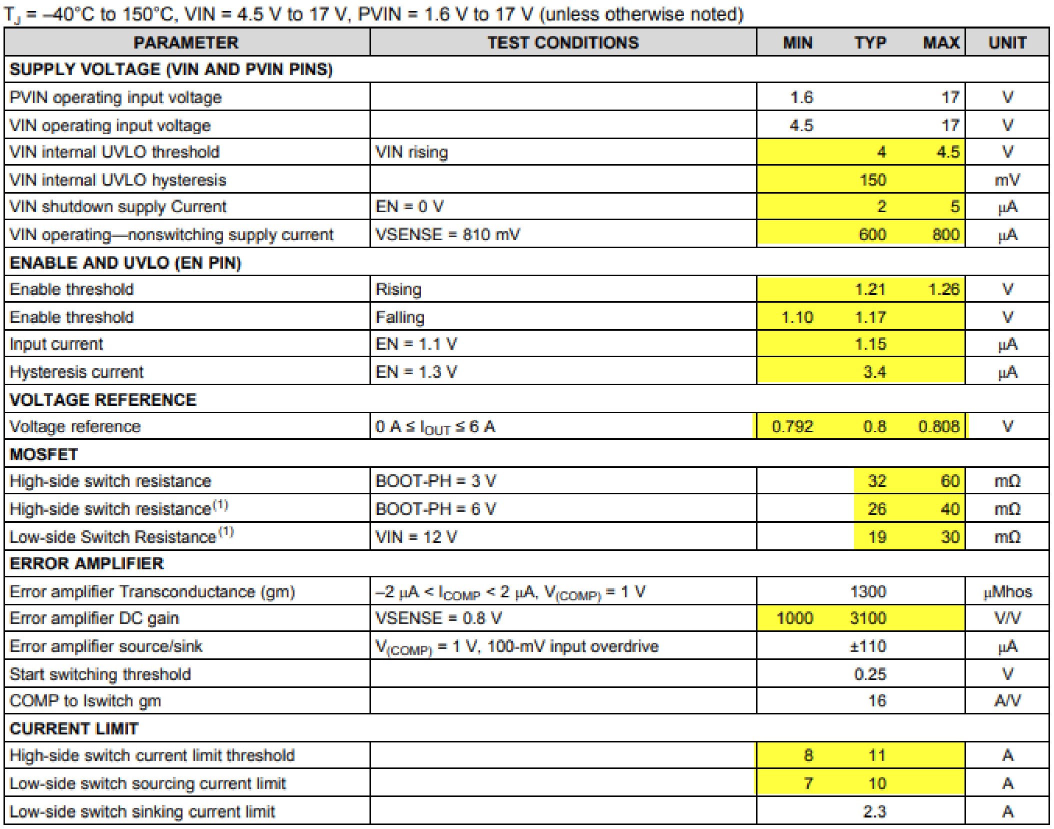
Predicting Performance of Integrated Circuits using Regression Analysis

Joanna Duran, Nikhil Gupta, Max Moro

INTRODUCTION

Semiconductor manufacturing is a variable process and outcomes depend on several factors. In order to meet target specifications, some parameters are controlled by engineers. However, some parameters are beyond human control (e.g. process variation). The output variables are typically measured after the manufacturing process is complete and all research and development has been performed. There is a set list of specified values that are acceptable, usually falling between a min and max range. Variation in the process leads to issues if the outputs are outside target specifications.

Below is an example specification sheet (Fig 1). Each row is a single output and their respective values but not all values are populated because they cannot be practically measured due to time, resource and cost constraints.



T_J = –40°C to 150°C, VIN = 4.5 V to 17 V, PVIN = 1.6 V to 17 V (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
SUPPLY VOLTAGE (VIN AND PVIN PINS)					
PVIN operating input voltage		1.6		17	V
VIN operating input voltage		4.5		17	V
VIN internal UVLO threshold	VIN rising		4	4.5	V
VIN internal UVLO hysteresis			150		mV
VIN shutdown supply Current	EN = 0 V		2	5	μA
VIN operating—nonswitching supply current	VSENSE = 810 mV		600	800	μA
ENABLE AND UVLO (EN PIN)					
Enable threshold	Rising		1.21	1.26	V
Enable threshold	Falling	1.10	1.17		V
Input current	EN = 1.1 V		1.15		μA
Hysteresis current	EN = 1.3 V		3.4		μA
VOLTAGE REFERENCE					
Voltage reference	0 A ≤ I_{OUT} ≤ 6 A	0.792	0.8	0.808	V
MOSFET					
High-side switch resistance	BOOT-PH = 3 V		32	60	mΩ
High-side switch resistance[1]	BOOT-PH = 6 V		26	40	mΩ
Low-side Switch Resistance[1]	VIN = 12 V		19	30	mΩ
ERROR AMPLIFIER					
Error amplifier Transconductance (gm)	–2 μA < I_{COMP} < 2 μA, $V_{(COMP)}$ = 1 V		1300		μMhos
Error amplifier DC gain	VSENSE = 0.8 V	1000	3100		V/V
Error amplifier source/sink	$V_{(COMP)}$ = 1 V, 100-mV input overdrive		±110		μA
Start switching threshold			0.25		V
COMP to Iswitch gm			16		A/V
CURRENT LIMIT					
High-side switch current limit threshold		8	11		A
Low-side switch sourcing current limit		7	10		A
Low-side switch sinking current limit			2.3		A

Fig 1: Sample output from an integrated circuit Reference

Like in most manufacturing environments, the question is “Can we predict the limits before the integrated circuits are manufactured to preemptively make changes when specs are expected to be out of range?” The answer is yes. Current practice is to use electrical simulation (plus running Monte Carlo simulations). However, this is very resource and time intensive as each electrical simulation can take several hours. Our objective is to build a model to predict the performance (min, typical, max) of an output variable.

DATA

Data for this project has been sponsored and approved for use by Texas Instruments Inc. (TI) who provided two files. Due to proprietary nature of the information, the variables have been anonymized.

Data Dictionary

The data consists of 10,000 rows capturing the performance of an integrated circuit under various conditions. There are 240 features consisting of:

- Engineer-controlled variables (x1 – x23). Values differ, some are between 1 to 100 while others are in Nano or Micro range.
- Process variation variables (stat1 – stat217). These parameters are beyond human control. They represent various statistical manufacturing parameters whose values represent the sigma variation around the mean. Range is from -3 (sigma) to 3 (sigma).
- Output Variables (y1 - y19) which represent various output variables.

Data Collection Process

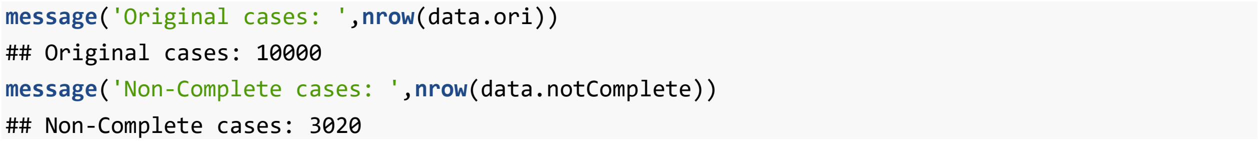
The engineer-controlled variables have a predefined range of values that the engineer can chose from. Since they can pick any value in this range, the values for these variables were uniformly sampled from the range of acceptable values while the data was being collected. Statistical features were also uniformly randomly sampled since the goal was to obtain good model accuracy throughout the statistical variation range and not just closer to the population means (which would have been the case if the data was sampled using a gaussian distribution since in that case, the training data would have had more points closer to the mean and very few points at the ± 3 sigma level).

GOAL

The goal was to pick one output variable and build a model to predict the mean value and the statistical variation of the output with respect to the process. Target accuracy of ±10% is desired, but ±15% would be acceptable. After discussion with subject matter experts, it was decided to focus on 'y3' as it is the most critical output of the process.

DATA PREPARATION

Like most data, this dataset also needed some cleaning. Basic descriptive statistics revealed that 3020 NA values were present (Fig 2). After consulting with the expert from TI, we found that the predictors (features) for these data points were not practical in combination with each other. Hence, these points are not valid and can be removed without impacting the predicting power of the model being developed. We cleaned up labels and removed NA values. Once the data was cleaned, the datasets were merged.



```
message('Original cases: ',nrow(data.ori))
## Original cases: 10000
message('Non-Complete cases: ',nrow(data.notComplete))
## Non-Complete cases: 3020
```

Fig 2: Original Data and number of rows containing NA values

Note that if these were indeed valid data points, we could not have simply removed them from the dataset since it would have violated the random sampling we performed initially, and this would have affected the generalization of the model to the entire design space (population).

EXPLORATORY DATA ANALYSIS

Output Variable

We begin exploratory data analysis on the target variable y3 and notice right skewness, therefore we perform a log transformation. Log transformation makes the data a little more normal, therefore we proceeded with the log transformed variable “y3.log” (Fig 3). It is important to note that base 10 is more common in this industry, which is why it was used instead of natural log.

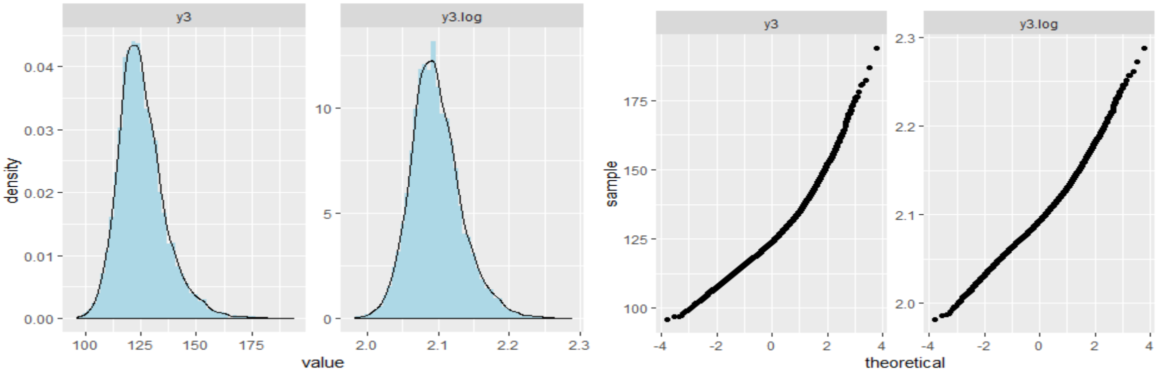


Fig 3: Histogram and QQ plot of y3 and log(y3)

Input Predictors (Features)

VIF and Correlations

To determine if there is a correlation within predictors (features), we begin by checking for multicollinearity. Since inputs were randomly selected, we did not expect there to be multicollinearity. After running the analysis, the correlation table (Table 1) and VIF values (Fig 4) confirmed that there is no multicollinearity. Note that the correlation table only shows a partial view of the first 10 variables, but none of the other variables showed any highly corelated terms either.

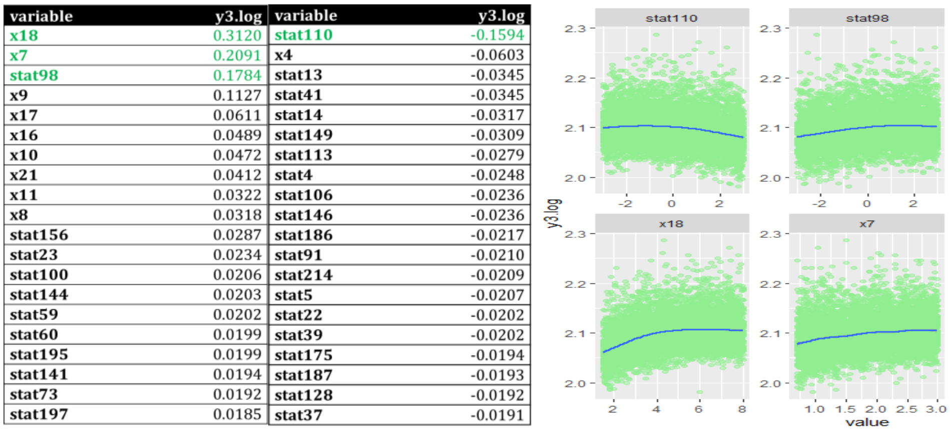
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	1.0000	0.0034	-0.0028	0.0085	0.0068	0.0159	0.0264	-0.0012	0.0142	0.0013
x2	0.0034	1.0000	-0.0057	0.0004	-0.0094	-0.0101	0.0089	0.0078	0.0049	-0.0214
x3	-0.0028	-0.0057	1.0000	0.0029	0.0046	0.0006	-0.0105	-0.0002	0.0167	-0.0137
x4	0.0085	0.0004	0.0029	1.0000	-0.0059	0.0104	0.0098	0.0053	0.0061	-0.0023
x5	0.0068	-0.0094	0.0046	-0.0059	1.0000	0.0016	-0.0027	0.0081	0.0259	-0.0081
x6	0.0159	-0.0101	0.0006	0.0104	0.0016	1.0000	0.0200	-0.0157	0.0117	-0.0072
x7	0.0264	0.0089	-0.0105	0.0098	-0.0027	0.0200	1.0000	-0.0018	-0.0069	-0.0221
x8	-0.0012	0.0078	-0.0002	0.0053	0.0081	-0.0157	-0.0018	1.0000	0.0142	-0.0004
x9	0.0142	0.0049	0.0167	0.0061	0.0259	0.0117	-0.0069	0.0142	1.0000	0.0149
x10	0.0013	-0.0214	-0.0137	-0.0023	-0.0081	-0.0072	-0.0221	-0.0004	0.0149	1.0000

Table 1: Correlation between inputs

```
##    Variables      VIF
## 1    stat202 1.063592
## 2    stat141 1.062435
## 3     stat52 1.062123
## 4    stat178 1.062030
## 5    stat164 1.059900
## 6    stat184 1.059400
## 7     stat70 1.058888
## 8    stat150 1.058825
## 9     stat14 1.058728
## 10    stat37 1.058385
```

Fig 4: Top 10 predictors by VIF

We also wanted to determine if there is a correlation of features to output “y3.log”. The correlation output in Fig 5 shows the most positively and the most negatively correlated features to the output. As we can see, the variables are not very highly correlated and only 4 out of 240 predictors have an absolute value of the correlation coefficient greater than 0.15. Hence, we may not be able to get the best predictive power from any model build from these features as is without significant feature engineering. The scatterplots of the variables with the most predictive power have also been shown in Fig 5 for reference.



variable	y3.log	variable	y3.log
x18	0.3120	stat110	-0.1594
x7	0.2091	x4	-0.0603
stat98	0.1784	stat13	-0.0345
x9	0.1127	stat41	-0.0345
x17	0.0611	stat14	-0.0317
x16	0.0489	stat149	-0.0309
x10	0.0472	stat113	-0.0279
x21	0.0412	stat4	-0.0248
x11	0.0322	stat106	-0.0236
x8	0.0318	stat146	-0.0236
stat156	0.0287	stat186	-0.0217
stat23	0.0234	stat91	-0.0210
stat100	0.0206	stat214	-0.0209
stat144	0.0203	stat5	-0.0207
stat59	0.0202	stat22	-0.0202
stat60	0.0199	stat39	-0.0202
stat195	0.0199	stat175	-0.0194
stat141	0.0194	stat187	-0.0193
stat73	0.0192	stat128	-0.0192
stat197	0.0185	stat37	-0.0191

Fig 5: Correlation coefficients and scatterplots of most correlated features to output

Feature Transformations

One of the highly correlated features, x18, shows a slight curvature in the scatter plot therefore a square root transformation was performed. After the transformation, the scatter plot showed slightly better linearity (Fig 6). Based on our observations above we proceed to modeling with the target variable y3 as log transformed and the predictor x18 transformed with square root.

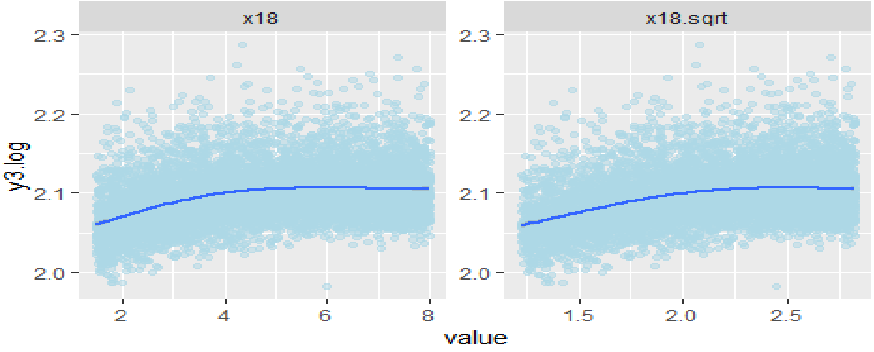


Fig 6: Scatterplot of x18 and sqrt(x18) vs. log(y3)

GOOD MODELING PRACTICES

When keeping in mind good modeling practices we want to make sure to avoid overfitting. Even though we have a lot of data points and many features, it is still possible to overfit the data. In order to avoid overfitting, we used an 80:20 ratio split on our dataset to develop the Train and Test sets. During the training process, a 10-fold cross validation technique was used for model selection. This allows for the use of the entire training dataset for model development.

FULL MODEL ANALYSIS

Fit

The full model analysis was performed by analyzing the full model consisting of all 240 features with the transformations (red) described above (Table 2). The fit statistics show that only a handful of predictors were statistically significant which was expected based on the correlation of features to output analyzed in the exploratory analysis (Table 2). The adjusted $R^2$is 0.2275 (Table 2) which is low and expected based on low correlation of features to outputs. We wondered if two-way interactions would improve predictions but considering that this would increase our predictors by 28680 (from 240 to 28920), it is not practical given that we only have 6980 data points. In the future, we may need to do intelligent (selective) feature engineering using domain expertise to improve prediction capability.

```
# y3.log ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 +
##     x12 + x13 + x14 + x15 + x16 + x17 + x19 + x20 + x21 + x22 +
##    x23 + stat1 + stat2 + stat3 + stat4 + stat5 + stat6 + stat7 +
##    stat8 + stat9 + stat10 + stat11 + stat12 + stat13 + stat14 +
##    stat15 + stat16 + stat17 + stat18 + stat19 + stat20 + stat21 +
##    stat22 + stat23 + stat24 + stat25 + stat26 + stat27 + stat28 +
##    stat29 + stat30 + stat31 + stat32 + stat33 + stat34 + stat35 +
##    stat36 + stat37 + stat38 + stat39 + stat40 + stat41 + stat42 +
##    stat43 + stat44 + stat45 + stat46 + stat47 + stat48 + stat49 +
##    stat50 + stat51 + stat52 + stat53 + stat54 + stat55 + stat56 +
##    stat57 + stat58 + stat59 + stat60 + stat61 + stat62 + stat63 +
##    stat64 + stat65 + stat66 + stat67 + stat68 + stat69 + stat70 +
##    stat71 + stat72 + stat73 + stat74 + stat75 + stat76 + stat77 +
##    stat78 + stat79 + stat80 + stat81 + stat82 + stat83 + stat84 +
##    stat85 + stat86 + stat87 + stat88 + stat89 + stat90 + stat91 +
##    stat92 + stat93 + stat94 + stat95 + stat96 + stat97 + stat98 +
##    stat99 + stat100 + stat101 + stat102 + stat103 + stat104 +
##    stat105 + stat106 + stat107 + stat108 + stat109 + stat110 +
##    stat111 + stat112 + stat113 + stat114 + stat115 + stat116 +
##    stat117 + stat118 + stat119 + stat120 + stat121 + stat122 +
##    stat123 + stat124 + stat125 + stat126 + stat127 + stat128 +
##    stat129 + stat130 + stat131 + stat132 + stat133 + stat134 +
##    stat135 + stat136 + stat137 + stat138 + stat139 + stat140 +
##    stat141 + stat142 + stat143 + stat144 + stat145 + stat146 +
##    stat147 + stat148 + stat149 + stat150 + stat151 + stat152 +
##    stat153 + stat154 + stat155 + stat156 + stat157 + stat158 +
##    stat159 + stat160 + stat161 + stat162 + stat163 + stat164 +
##    stat165 + stat166 + stat167 + stat168 + stat169 + stat170 +
##    stat171 + stat172 + stat173 + stat174 + stat175 + stat176 +
##    stat177 + stat178 + stat179 + stat180 + stat181 + stat182 +
##    stat183 + stat184 + stat185 + stat186 + stat187 + stat188 +
##    stat189 + stat190 + stat191 + stat192 + stat193 + stat194 +
##    stat195 + stat196 + stat197 + stat198 + stat199 + stat200 +
##    stat201 + stat202 + stat203 + stat204 + stat205 + stat206 +
##    stat207 + stat208 + stat209 + stat210 + stat211 + stat212 +
##    stat213 + stat214 + stat215 + stat216 + stat217 + x18.sqrt
```

```
##                Estimate Std. Error t value Pr(>|t|)
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.970e+00  9.597e-03 205.243  < 2e-16 ***
## x4          -5.252e-05  9.046e-06  -5.806 6.76e-09 ***
## x7           1.136e-02  6.408e-04  17.728  < 2e-16 ***
## x8           3.673e-04  1.488e-04   2.468 0.013599 *
## x9           3.299e-03  3.309e-04   9.970  < 2e-16 ***
## x10          1.172e-03  3.082e-04   3.804 0.000144 ***
## x11          1.889e+05  7.379e+04   2.560 0.010499 *
## x16          8.416e-04  2.150e-04   3.915 9.16e-05 ***
## x17          1.519e-03  3.254e-04   4.670 3.09e-06 ***
## x21          1.298e-04  4.216e-05   3.078 0.002093 **
## x22         -5.997e-04  3.457e-04  -1.735 0.082801 .
## stat4       -5.704e-04  2.485e-04  -2.295 0.021752 *
## stat13      -6.822e-04  2.480e-04  -2.751 0.005964 **
## stat14      -7.839e-04  2.476e-04  -3.167 0.001551 **
## stat18      -4.600e-04  2.477e-04  -1.857 0.063317 .
## stat22      -4.362e-04  2.499e-04  -1.745 0.080965 .
## stat23       6.593e-04  2.487e-04   2.651 0.008048 **
## stat24      -4.303e-04  2.484e-04  -1.733 0.083206 .
## stat35      -4.613e-04  2.496e-04  -1.848 0.064630 .
## stat41      -4.256e-04  2.476e-04  -1.719 0.085760 .
## stat45      -4.338e-04  2.479e-04  -1.750 0.080178 .
## stat51       4.864e-04  2.469e-04   1.970 0.048836 *
## stat59       4.114e-04  2.484e-04   1.656 0.097693 .
## stat91      -4.426e-04  2.461e-04  -1.798 0.072172 .
## stat98       3.599e-03  2.455e-04  14.659  < 2e-16 ***
## stat100      5.062e-04  2.498e-04   2.027 0.042727 *
## stat103     -4.194e-04  2.506e-04  -1.673 0.094353 .
## stat110     -3.282e-03  2.473e-04 -13.273  < 2e-16 ***
## stat113     -4.284e-04  2.510e-04  -1.707 0.087947 .
## stat146     -4.353e-04  2.499e-04  -1.742 0.081524 .
## stat149     -4.619e-04  2.493e-04  -1.853 0.063961 .
## stat175     -5.229e-04  2.478e-04  -2.110 0.034862 *
## stat195      4.201e-04  2.480e-04   1.694 0.090350 .
## stat204     -5.073e-04  2.462e-04  -2.060 0.039421 *
## stat207      5.098e-04  2.482e-04   2.054 0.040009 *
## x18.sqrt     2.657e-02  9.510e-04  27.943  < 2e-16 ***
## (Note: only showing the significant predictors above)
```

```
## Residual standard error: 0.03144 on 5343 degrees of freedom
## Multiple R-squared:  0.2711, Adjusted R-squared:  0.2384
## F-statistic: 8.28  on 240 and 5343 DF,  p-value: < 2.2e-16
```

Table 2: Full Model (Formula, Statistically Significant Variables and Fit Statistics)

Model Assumptions

We also analyzed the plots for issues with model assumption. For independence, the features were selected randomly therefore independence assumption has been met. For normality, histogram and QQ plot of residuals (Fig 7) shows that they are not normally distributed. The residual plot (Fig 8) also point to non-normality since there are several values beyond the 2-sigma line. However, given that we are working with such a large sample size, the concern is minimized. For equal variance, the residual plot (Fig 8) shows that the points do not exhibit equal variance at all predicted levels. The points at the lower predicted levels have much smaller variance compared to points at the higher predicted levels.

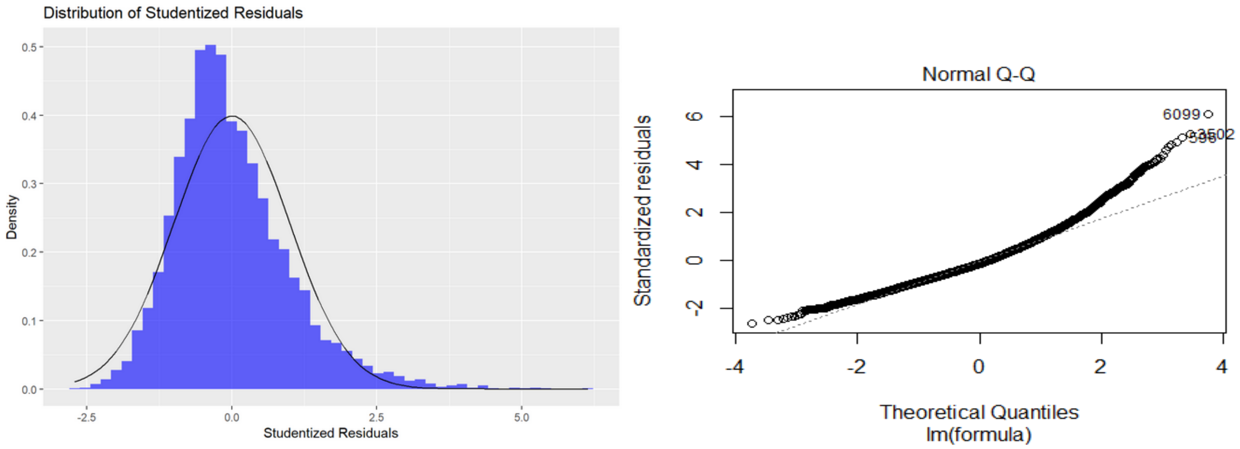


Fig 7: Histogram and QQ Plot of Residuals (Full Model)

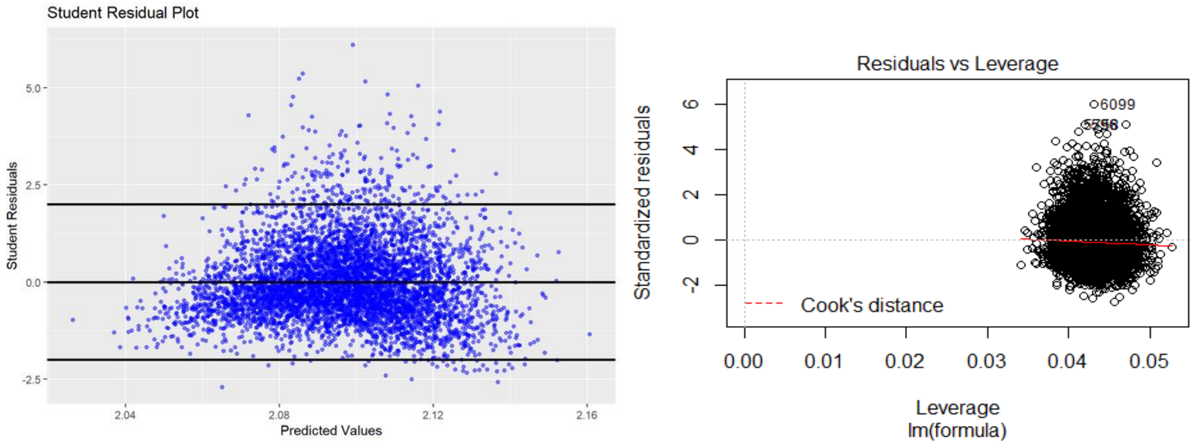


Fig 8: Studentized and Standardized Residuals (Full Model)

The influence statistics are also analyzed and from the Cook's D plot we notice that there are 288 points beyond the 4/n line (Fig 9). This is a large amount of points and it is impractical to analyze all of them manually. These points are influential and can impact the model coefficients adversely. It is tempting to remove these points because it would improve the model fit drastically (Fig 10). But after consulting with domain experts, we found that these are valid points and there is no justification to remove them. We will proceed with these points included, noting that the assumptions for linear regression have not been satisfied entirely.

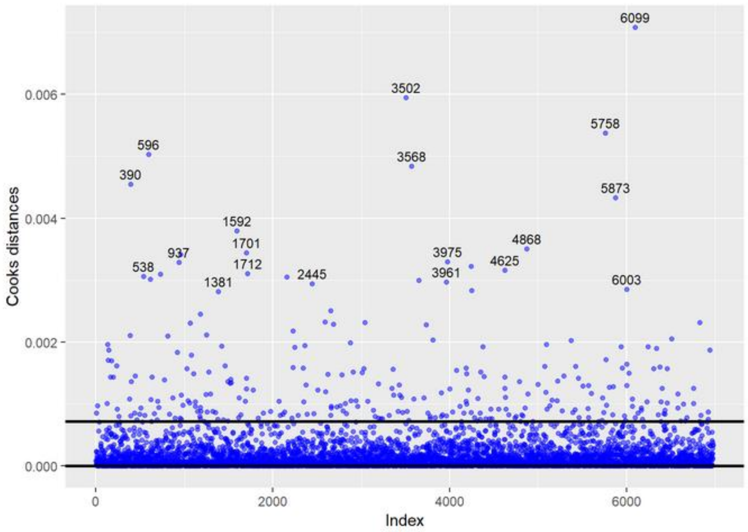


Fig 9: Cook's D for Full Model

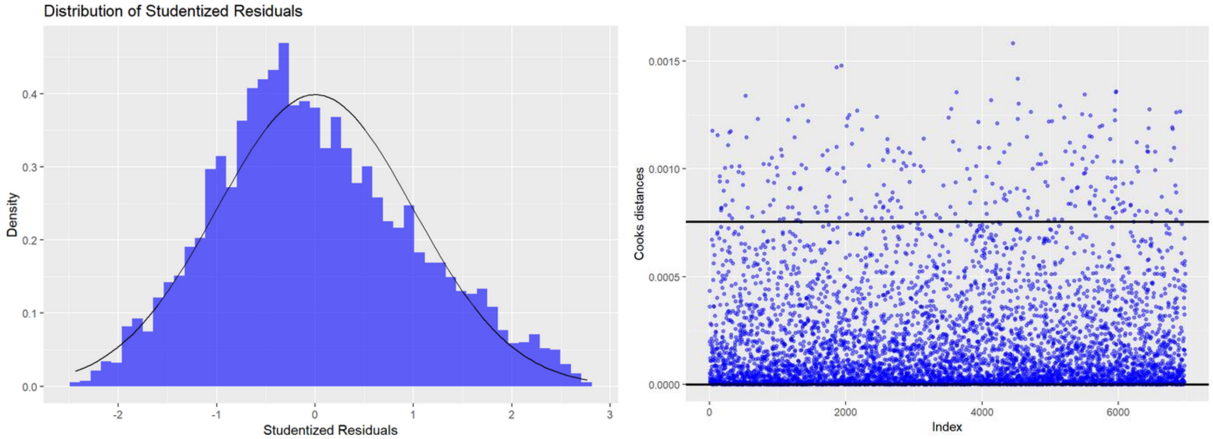


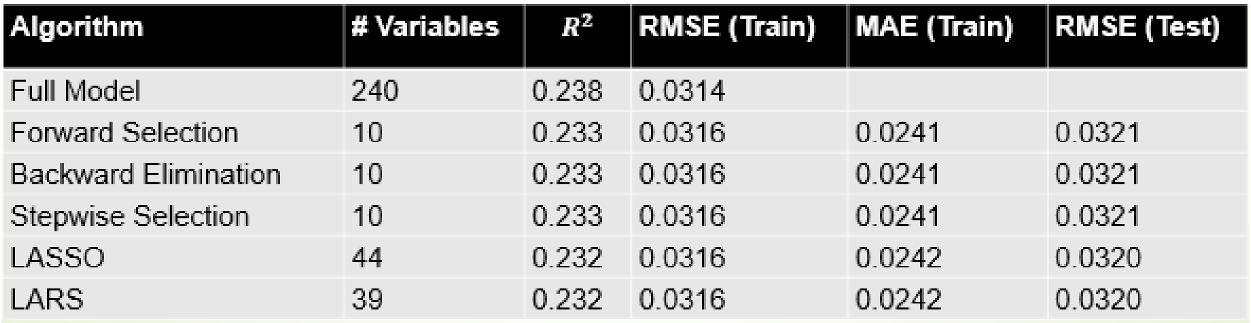
Fig 10: Histogram and Cook's D after removal of influential points

VARIABLE SELECTION PROCESS

It was determined that variable selection may be appropriate at this step to remove unnecessary predictor variables. In selecting variables, we know there are many techniques that provide many different results. The variable selection techniques that were used were:

A. Forward Selection
B. Backward Elimination
C. Stepwise Selection
D. LASSO
E. LARS

We found that all techniques give essentially the same fit statistics. Forward Selection, Backward Elimination and Stepwise Selection have only 13 predictors while LASSO and LARS have 37 predictors in the final model (Table 3).



Algorithm	# Variables	R^2	RMSE (Train)	MAE (Train)	RMSE (Test)
Full Model	240	0.238	0.0314		
Forward Selection	10	0.233	0.0316	0.0241	0.0321
Backward Elimination	10	0.233	0.0316	0.0241	0.0321
Stepwise Selection	10	0.233	0.0316	0.0241	0.0321
LASSO	44	0.232	0.0316	0.0242	0.0320
LARS	39	0.232	0.0316	0.0242	0.0320

Table 3: Comparison of various variable selection techniques

The best model is Backward Elimination. It was chosen with one of the reasons being that it has the least number of predictors. RMSE was used to choose the final model during backward selection. The lowest value of RMSE obtained for the model with 13 variables. We also note that the number of predictors selected would have been the same even if we would have chosen metric MAE or R^2 instead as RMSE for final model selection (Fig 11).

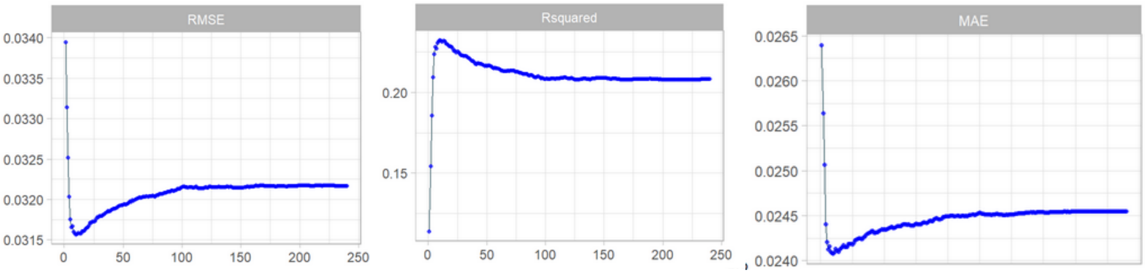
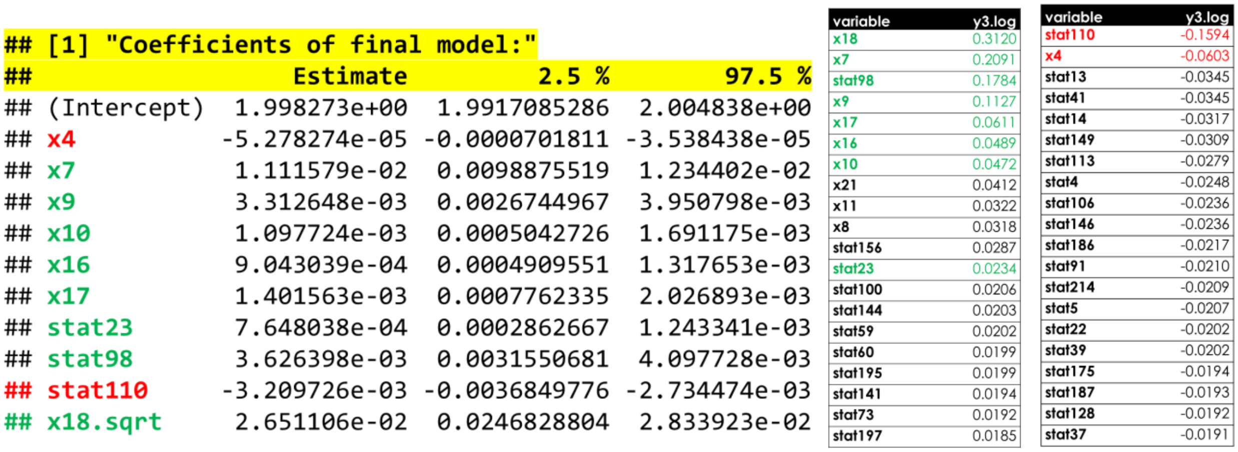


Fig 11: Fit statistics vs. step in backward selection process

The final model includes some of the same variables that we identified to be highly correlated to output in the exploratory analysis (Fig 12).



```
## [1] "Coefficients of final model:"
##                   Estimate          2.5 %         97.5 %
## (Intercept)   1.998273e+00   1.9917085286   2.004838e+00
## x4           -5.278274e-05  -0.0000701811  -3.538438e-05
## x7            1.111579e-02   0.0098875519   1.234402e-02
## x9            3.312648e-03   0.0026744967   3.950798e-03
## x10           1.097724e-03   0.0005042726   1.691175e-03
## x16           9.043039e-04   0.0004909551   1.317653e-03
## x17           1.401563e-03   0.0007762335   2.026893e-03
## stat23        7.648038e-04   0.0002862667   1.243341e-03
## stat98        3.626398e-03   0.0031550681   4.097728e-03
## stat110      -3.209726e-03  -0.0036849776  -2.734474e-03
## x18.sqrt      2.651106e-02   0.0246828804   2.833923e-02
```

variable	y3.log
x18	0.3120
x7	0.2091
stat98	0.1784
x9	0.1127
x17	0.0611
x16	0.0489
x10	0.0472
x21	0.0412
x11	0.0322
x8	0.0318
stat156	0.0287
stat23	0.0234
stat100	0.0206
stat144	0.0203
stat59	0.0202
stat60	0.0199
stat195	0.0199
stat141	0.0194
stat73	0.0192
stat197	0.0185

variable	y3.log
stat110	-0.1594
x4	-0.0603
stat13	-0.0345
stat41	-0.0345
stat14	-0.0317
stat149	-0.0309
stat113	-0.0279
stat4	-0.0248
stat106	-0.0236
stat146	-0.0236
stat186	-0.0217
stat91	-0.0210
stat214	-0.0209
stat5	-0.0207
stat22	-0.0202
stat39	-0.0202
stat175	-0.0194
stat187	-0.0193
stat128	-0.0192
stat37	-0.0191

Fig 12: Final Model Parameter Values, Confidence Intervals and comparison to highly correlated features

In analyzing the residuals, we noticed the histogram of the residuals (Fig 13) shows similar right skewed distribution as was seen in the full model. In the residual scatterplot we noticed the equality of variance at lower values of prediction are questionable as before (Fig 13). We conclude that the final model suffers from same assumption violations as the full model.

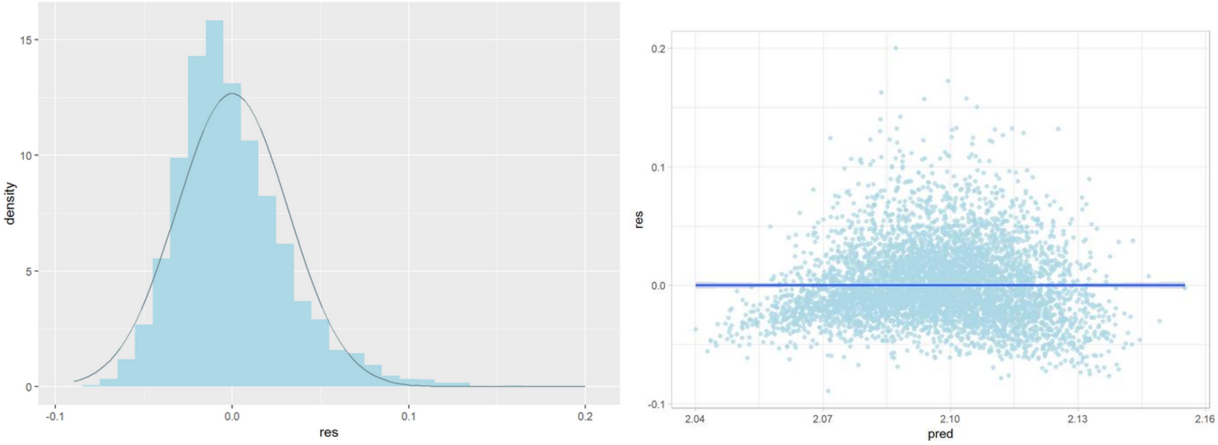


Fig 13: Histogram of residuals and residual plot (final model)

INFERENCE

Based on all the analysis of the best model we believe there is room for improvement. For population inference, given the fact that the features were sampled randomly, any predictions drawn from this model can be applied to the entire design and manufacturing space of the integrated circuit. Causality is not a concern here since the goal is mainly prediction of performance using the model. However, given the fact that some of the model assumptions for multiple linear regression have not been met, we should be careful in using this model to predict new values.

CONCLUSION

We did not meet our target accuracy (Fig 14 - green line represents ±10% and red line represents ±15% accuracy) but gained a lot of insights. We have included all variables possible in the model (domain of manually changeable variables as well as all the statistical variables), hence model accuracy cannot be improved by adding any new variables. We cannot do 2-way interactions as discussed above due to lack of data points.

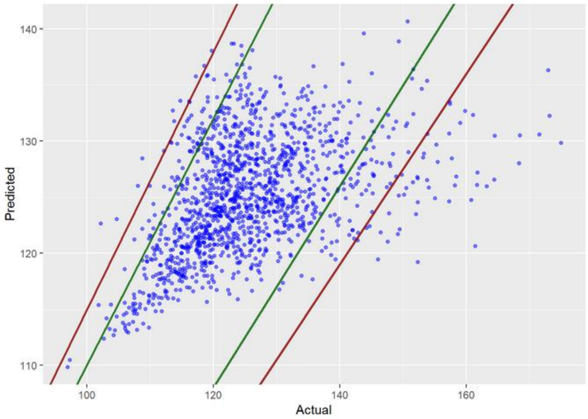
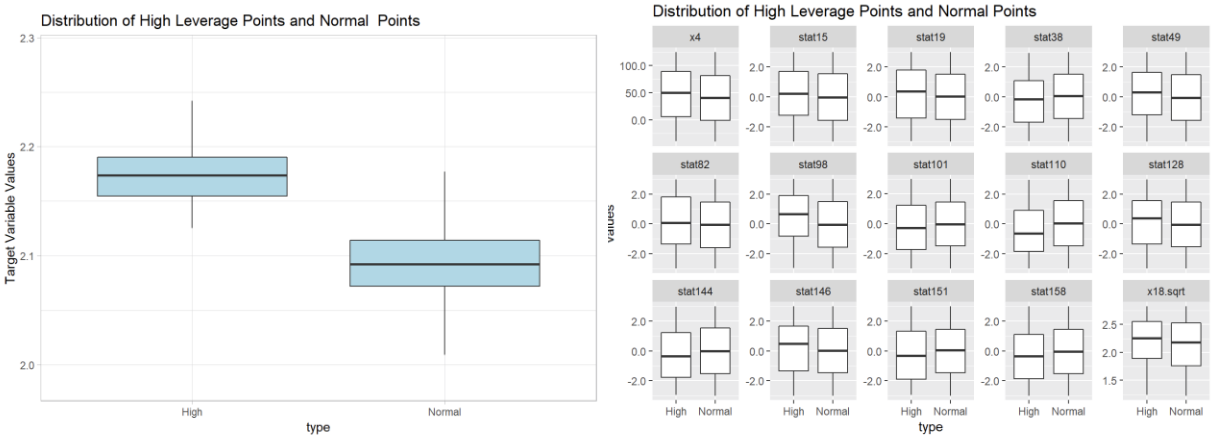


Fig 14: Predicted vs. Actual Values (Test Data)

However, to improve the model predictions, we recommend three follow-up steps:

1. Intelligent (selective) feature engineering can be used to improve prediction performance. This would include not just taking multiplication interaction terms, but rather taking more complex interaction terms (such as exponentials, division of features with one another etc.). The feature engineering in this case would be derived from integrated circuit design theory and would need an expert to derive.
2. We can also use non-parametric models that do not need to satisfy normality and equal variance assumptions. This would include more advanced modeling techniques, e.g. tree-based techniques such as random forest, XGBoost, etc.
3. Divide data into clusters and model each cluster separately. When analyzing the high Cook's D points in the full model, we realized that the distribution of some of the features were statistically different (Fig 15) between the 2 groups (those including the high Cook's D points and those including the low Cook's D points). Hence, it may be worthwhile to cluster the dataset into 2 (or more) groups and model each group using a separate model.



```r
comp.test = lapply(dplyr::select(plotData, one_of(feature.names))
                   , function(x) t.test(x ~ plotData$type, var.equal = TRUE))

sig.comp = list.filter(comp.test, p.value < 0.05)
sapply(sig.comp, function(x) x[['p.value']])
```

```
##           x4       stat15       stat19       stat38       stat49       stat82       stat98      stat101      stat110 
## 2.772613e-02 4.232345e-02 3.891725e-02 2.182942e-02 2.114386e-02 2.135716e-02 3.618666e-06 4.293858e-02 1.151664e-04 
##      stat128      stat144      stat146      stat151      stat158     x18.sqrt 
## 2.243626e-02 4.705614e-02 3.530774e-02 4.747317e-02 7.249900e-03 5.033029e-03
```

Fig 15: Statistical comparison of points with high Cook's D and low Cook's D values

APPENDIX

Refer to this link for detailed and reproducible analysis and this link for the final HTML file of the analysis.