IDS 575 - STATISTICAL MODELS AND METHODS

PROJECT REPORT

Palak Gupta
Tanveer Singh Rainu
Aparna Pai

ABSTRACT

The problem statement of this project is to predict the Loss associated with the accident claims of AllState customers. Allstate insurance claims was captured in a humongous dataset and hosted in Kaggle. The sheer volume of data as well as higher proportion of categorical variables made the dataset interesting and challenging to our team. For this regression problem, we performed extensive data treatment without excluding any independent variables. The baseline OLS model that we developed gave us insights regarding further treatment necessary for the categorical variables. The advanced models like Gradient Boost and Extreme Gradient Boost were developed on AWS EC2 Compute instance to achieve parallel processing and surging the efficiency. This not only improved our skills on Machine Learning, but also the nuances of using cloud computing technology to exponentially increase the performance.

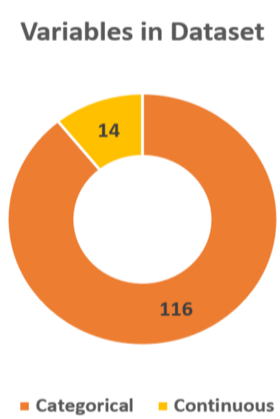
INTRODUCTION

Allstate is a leading Insurance provider who is trying to ease the process of insurance claims for their clients. When one of their clients is devastated by a serious car accident, Allstate wants to ensure that the client has to go through as minimal paperwork as possible to get the claim settled. Settling an insurance claim using an automated system of Loss and Severity prediction is what Allstate intends to materialize. The aim of this project is to develop advanced regression models which are trained and tested on the datasets hosted by Allstate in Kaggle. These regression model would predict the dependent variable that is Loss, associated with each accident with respect to the 100+ explanatory parameters using 100k+ rows of claims training data.

The Dataset

The training and test datasets have 130 variables excluding the Row ID and Target variable, out of which 116 are categorical

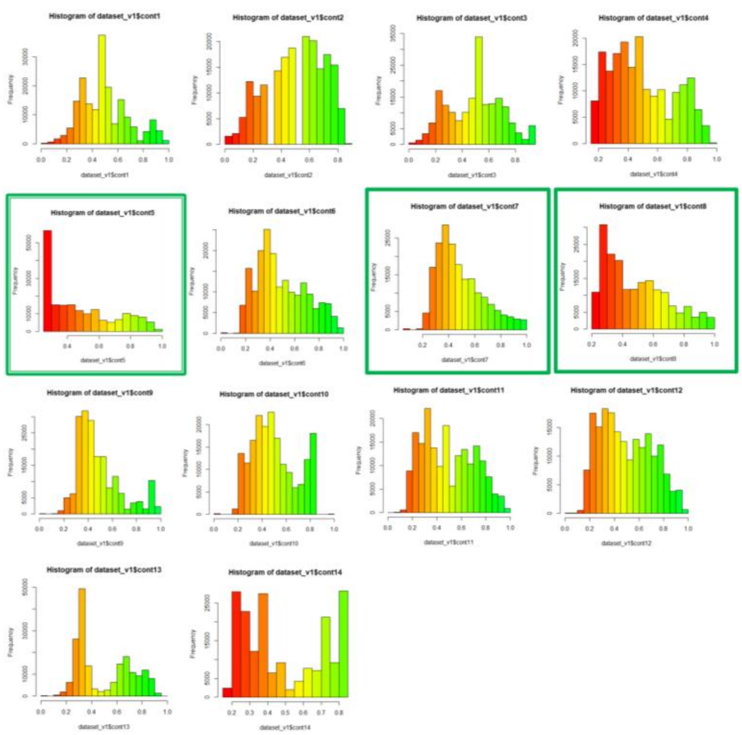
- Training data size - 188,318 rows
- Test data size - 125,546 rows
- Independent variables Categorical - 116
- Independent variables Continuous - 14
- Target variable – ***Loss*** (continuous)
- Missing data - none

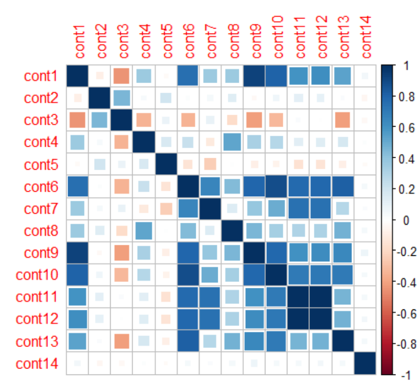


MODELS AND METHODS

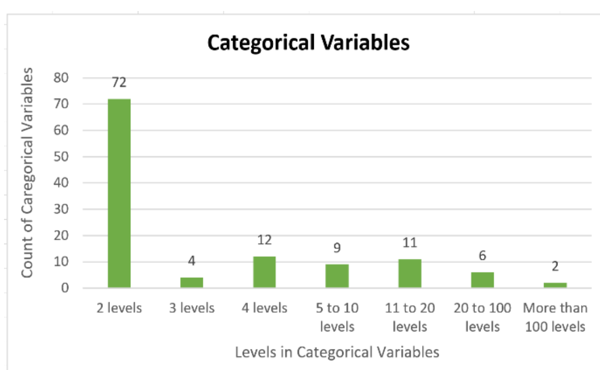
Univariate & Multivariate Analysis for Continuous Independent Variables

To begin with the 14 continuous variables were normalized and plotted for detecting skewness and outliers. Given below are the histograms of these variables. To build an OLS model, the variables are required to be normally distributed. So, the proposed treatment on the highlighted skewed variables would be a suitable transformation like log, polynomial or sqrt. Correlation between the continuous variables was analyzed next. The correlation plot is as sown in the image below.





Analysis for Categorical Independent Variables



There were multiple categorical variables in the dataset which had 100+ levels within them. The above graph shows the number of levels carried by the categorical variables. For the ease of modelling, the levels needed to be reduced.

S no	variable name	num Levels	A	B	% of B	Comment
70	cat70	2	188295	23	0.0%	Exclude from analysis since very less B
15	cat15	2	188284	34	0.0%	Exclude from analysis since very less B
22	cat22	2	188275	43	0.0%	Exclude from analysis since very less B
62	cat62	2	188273	45	0.0%	Exclude from analysis since very less B
64	cat64	2	188271	47	0.0%	Exclude from analysis since very less B
63	cat63	2	188239	79	0.0%	Exclude from analysis since very less B
68	cat68	2	188176	142	0.1%	Exclude from analysis since very less B
55	cat55	2	188173	145	0.1%	Exclude from analysis since very less B
56	cat56	2	188136	182	0.1%	Exclude from analysis since very less B
20	cat20	2	188114	204	0.1%	Exclude from analysis since very less B
35	cat35	2	188105	213	0.1%	Exclude from analysis since very less B
58	cat58	2	188079	239	0.1%	Exclude from analysis since very less B
48	cat48	2	188049	269	0.1%	Exclude from analysis since very less B
59	cat59	2	188018	300	0.2%	Exclude from analysis since very less B
69	cat69	2	188011	307	0.2%	Exclude from analysis since very less B
21	cat21	2	187905	413	0.2%	Exclude from analysis since very less B
60	cat60	2	187872	446	0.2%	Exclude from analysis since very less B
34	cat34	2	187734	584	0.3%	Exclude from analysis since very less B
67	cat67	2	187626	692	0.4%	Exclude from analysis since very less B
47	cat47	2	187617	701	0.4%	Exclude from analysis since very less B
61	cat61	2	187596	722	0.4%	Exclude from analysis since very less B
46	cat46	2	187436	882	0.5%	Exclude from analysis since very less B
33	cat33	2	187361	957	0.5%	Exclude from analysis since very less B
18	cat18	2	187331	987	0.5%	Exclude from analysis since very less B
32	cat32	2	187107	1211	0.6%	Exclude from analysis since very less B
51	cat51	2	187071	1247	0.7%	Exclude from analysis since very less B
17	cat17	2	187009	1309	0.7%	Exclude from analysis since very less B
42	cat42	2	186623	1695	0.9%	Exclude from analysis since very less B
19	cat19	2	186510	1808	1.0%	
65	cat65	2	186056	2262	1.2%	
14	cat14	2	186041	2277	1.2%	
57	cat57	2	185296	3022	1.6%	
30	cat30	2	184760	3558	1.9%	
29	cat29	2	184593	3725	2.0%	
43	cat43	2	184110	4208	2.2%	
45	cat45	2	183991	4327	2.3%	
54	cat54	2	183762	4556	2.4%	
7	cat7	2	183744	4574	2.4%	
39	cat39	2	183393	4925	2.6%	
31	cat31	2	182980	5338	2.8%	
24	cat24	2	181977	6341	3.4%	
16	cat16	2	181843	6475	3.4%	
41	cat41	2	181177	7141	3.8%	
28	cat28	2	180938	7380	3.9%	
40	cat40	2	180119	8199	4.4%	
66	cat66	2	179982	8336	4.4%	
52	cat52	2	179505	8813	4.7%	
49	cat49	2	179127	9191	4.9%	
71	cat71	2	178646	9672	5.1%	
3	cat3	2	177993	10325	5.5%	
8	cat8	2	177274	11044	5.9%	
26	cat26	2	177119	11199	5.9%	

Categorical Variable – 2 levels

For the 72 variables with 2 factors, we noticed that occurrence of level A was much higher compared to B. By sorting the percentage of contribution of B in each categorical variable and sorting the list from lowest to highest, below table was obtained. Those variables where B occurred less than 1% were excluded.

Categorical Variable – 3 levels

For the 4 variables with 3 factors, we again noticed that occurrence of level A was very scarce compared to A. However, B had reasonably good percentage of occurrence. Thus, we merged B and C to form a new level BC within these four variables.

S no	variable name	num Levels	A	B	C	% of B	% of c	Comment
73	cat73	3	154275	34017	26	18.06%	0.01%	Merge B and C as on Category BC
74	cat74	3	184731	3561	26	1.89%	0.01%	Merge B and C as on Category BC
75	cat75	3	154307	34010	1	18.06%	0.00%	Merge B and C as on Category BC
76	cat76	3	181347	6183	788	3.28%	0.42%	Merge B and C as on Category BC

Categorical Variable – 4 levels

Similarly, among the 12 variables with 4 levels, two were excluded as most of their % of occurrence was due to a single level among the four.

S no	variable name	num Levels	A	B	C	D	% A	%B	% C	% D	Comment
77	cat77	4	49	358	408	187503	0.03%	0.19%	0.22%	99.57%	Exclude Variable
78	cat78	4	788	186526	645	359	0.42%	99.05%	0.34%	0.19%	Exclude Variable
79	cat79	4	7064	152929	1668	26657	3.75%	81.21%	0.89%	14.16%	No Treatment
80	cat80	4	783	46538	3492	137505	0.42%	24.71%	1.85%	73.02%	No Treatment
81	cat81	4	788	24132	9013	154385	0.42%	12.81%	4.79%	81.98%	No Treatment
82	cat82	4	19322	147536	2655	18805	10.26%	78.34%	1.41%	9.99%	No Treatment
83	cat83	4	26038	141534	4958	15788	13.83%	75.16%	2.63%	8.38%	No Treatment
84	cat84	4	29450	431	154939	3498	15.64%	0.23%	82.28%	1.86%	No Treatment
85	cat85	4	788	186005	1011	514	0.42%	98.77%	0.54%	0.27%	Merge A B and C as Category ACD
86	cat86	4	1589	103852	10290	72587	0.84%	55.15%	5.46%	38.54%	No Treatment
87	cat87	4	788	166992	8819	11719	0.42%	88.68%	4.68%	6.22%	No Treatment
88	cat88	4	168926	7	19302	83	89.70%	0.00%	10.25%	0.04%	Merge B C D as Category BCD

Categorical Variable – 5 levels

For the three variables with 5 levels, no treatments were necessary as the distribution was moderate.

S no	variable name	num Levels	A	B	C	D	E	% A	%B	% C	% D	% E	comment
93	cat93	5	432	1133	35788	150237	728	0.2%	0.6%	19.0%	79.8%	0.4%	No Treatment
95	cat95	5	3736	109	87531	79525	17417	2.0%	0.1%	46.5%	42.2%	9.2%	No Treatment
98	cat98	5	105492	542	21485	50557	10242	56.0%	0.3%	11.4%	26.8%	5.4%	No Treatment

Categorical Variable – 7 and 8 levels

For the variables with 7 to 8 levels, merging was inevitable due to the low percentage of occurrence of certain levels. However, no variables were excluded.

S no	variable name	num Levels	A	B	C	D	E	F	G	% A	%B	% C	% D	% E	% F	%G	comment
90	cat90	7	177993	9515	728	70	6	4	2	94.52%	5.05%	0.39%	0.04%	0.00%	0.00%	0.00%	Merge BCDEFG
92	cat92	7	124689	628	62	11	1	62901	26	66.21%	0.33%	0.03%	0.01%	0.00%	33.40%	0.01%	Merge BCDFI
94	cat94	7	738	51710	13623	121642	91	494	20	0.39%	27.46%	7.23%	64.59%	0.05%	0.26%	0.01%	Merge AEFG
97	cat97	7	41970	34	78127	3779	47450	213	16745	22.29%	0.02%	41.49%	2.01%	25.20%	0.11%	8.89%	Merge BDF

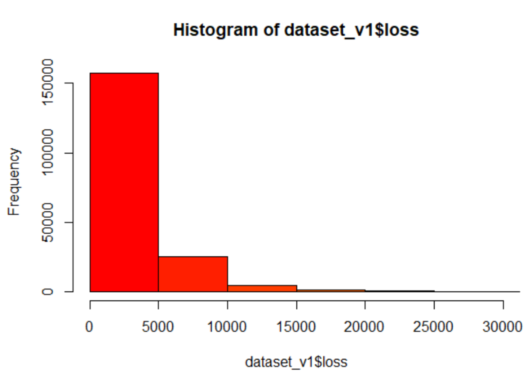
S no	variable name	num Levels	A	B	C	D	E	F	G	H	% A	%B	% C	% D	% E	% F	%G	%H	Comment
89	cat89	8	183744	4312	220	33	5	1	1	2	97.57%	2.29%	0.12%	0.02%	0.00%	0.00%	0.00%	0.00%	Merge ABCDEGHI
91	cat91	8	111028	42630	6400	1149	254	97	26734	26	58.96%	22.64%	3.40%	0.61%	0.13%	0.05%	14.20%	0.01%	Merge DEFH
96	cat96	8	35	2957	24	7922	174360	343	2665	12	0.02%	1.57%	0.01%	4.21%	92.59%	0.18%	1.42%	0.01%	Merge ACFI

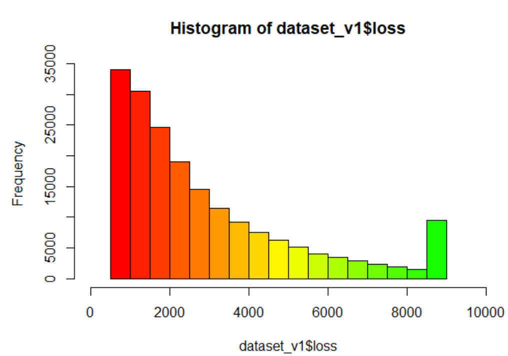
Categorical Variable – 8+ levels

There were around 20 variables which had more than 8 levels. Particularly, the variable ‘cat116’ had 326 levels and ‘cat110’ had 131 levels. Their distributions are given below. The percentage of occurrence of most of the levels were scarce hence those were clubbed together as a single level namely ‘Other’. This would make the process of interpreting the results easier.

TREATMENT FOR DEPENDENT VARIABLE

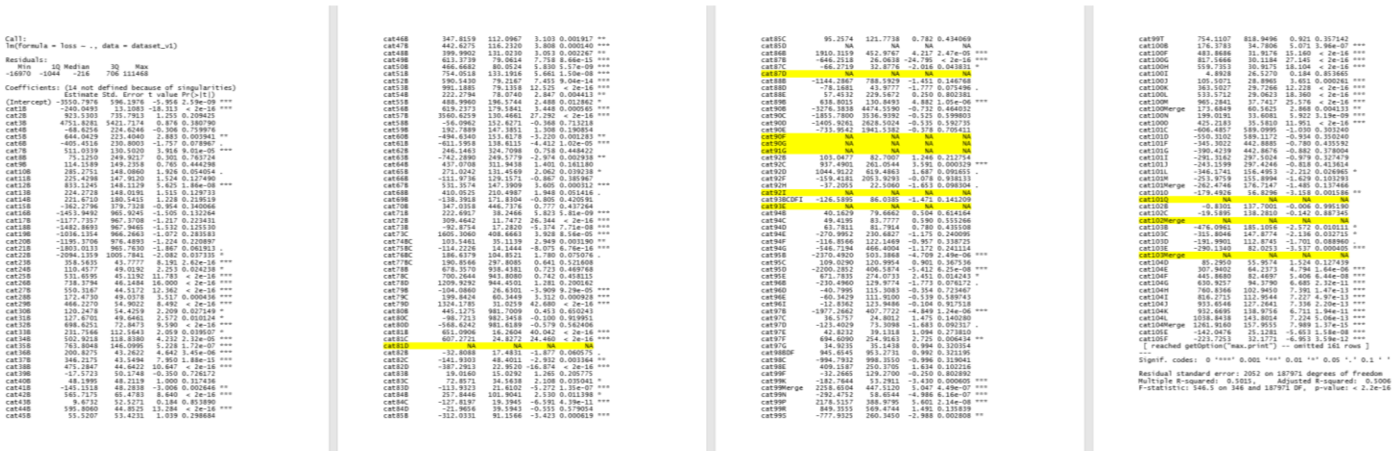
The dependent variable “Loss” was continuous with a highly right skewed distribution as seen in the left histogram. We binned the variable into quantiles and detected the outliers. We truncated the variable to obtain the distribution shown in the right.





LINEAR REGRESSION MODEL

The treated variables were combined to form the intermediate dataset. We developed a basic regression model on this dataset to obtain the beta estimates. However, we found some of the beta estimates to be "NA" which indicated multicollinearity. The list of such coefficients is highlighted in the OLS model summary shown below. These variables were further treated as mentioned before to obtain our final dataset.



EXPERIMENTAL RESULTS & DISCUSSION

The advanced models used were as given below

1. LASSO Regression
2. RIGDE Regression
3. Elastic Net Regression
4. Gradient Boosted Regression
5. Extreme Gradient Boosting Regression

Performance Measures used were conspicuous

- Mean Absolute Error (As suggested in the Problem Statement on Kaggle)
- Root Mean Squared Error

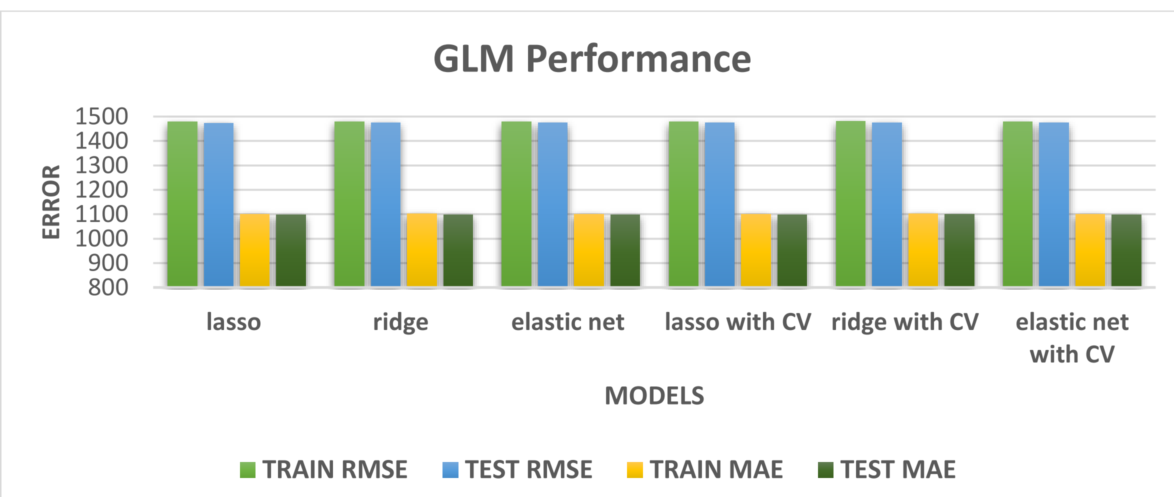
GLM – LASSO, RIDGE, ELASTIC NET

RIDGE - Ridge regression is very similar to least squares, except that the coefficients ridge is estimated by minimizing the loss function of least squares along with a regularization term L2 norm of the weights of the variables which is regularized by lambda. As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small. However, the second term, the regularization term, called a shrinkage penalty, is small when $\beta 1, \ldots, \beta p$ are close to zero, and so it has the effect of shrinking penalty the estimates of βj towards zero. The tuning parameter λ serves to control the relative impact of these two terms on the regression coefficient estimates. When $\lambda = 0$, the penalty term has no effect, and ridge regression will produce the least squares estimates. However, as $\lambda \rightarrow \infty$, the impact of the shrinkage penalty grows, and the ridge regression coefficient estimates will approach zero. Unlike least squares, which generates only one set of coefficient estimates, ridge regression will produce a different set of coefficient estimates, λ, for each value of λ.

LASSO – The lasso is a relatively recent alternative to ridge regression that over comes disadvantage of ridge that all p variables feature in the final coefficient estimates. The lasso coefficients, minimize the quantity least square and L1 norm of the weights. As with ridge regression, the lasso shrinks the coefficient estimates towards zero. However, in the case of the lasso, the L1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large. Hence, much like best subset selection, the lasso performs variable selection. As a result, models generated from the lasso are generally much easier to interpret than those produced by ridge regression. Before finalizing the model following steps were performed

- Models were made without cross validation first and then with cross validation and the results were compared
- Models with cross validation performed better
- We then experimented with different values of cross validation, the best results came at CVfolds = 20, after that the net improvement in the model accuracy was very minimal as compared to the computational expense
- Different values of lambda were used while predicting the results
- Graph of cross validation error and lambda values was plotted
- The best results were given by the lowest lambda values

For elastic net we tried various alpha values to get the best results.



GRADIENT BOOSTED REGRESSION

Gradient Boosted Regression – The gradient boosted regression is an additive model that is developed sequentially. It as combination of weak learners to build a strong learner. At each step a new learner is developed to address the short comings of the current model. In GB Regression, at each step, we fit the residuals and try to minimize the loss. Parameters of Gradient Boosting

1. Loss function – For continuous, outcomes the choices are gaussian (for minimizing squared error), Laplace (for minimizing absolute error), and quantile regression (for estimating percentile of the conditional distribution of the outcome). In our analysis, we have used the gaussian loss function

2. No. of iterations – It is the number of rounds of boosting that needs to be performed. Increasing the no. of iteration to a very high value might lead to overfitting.

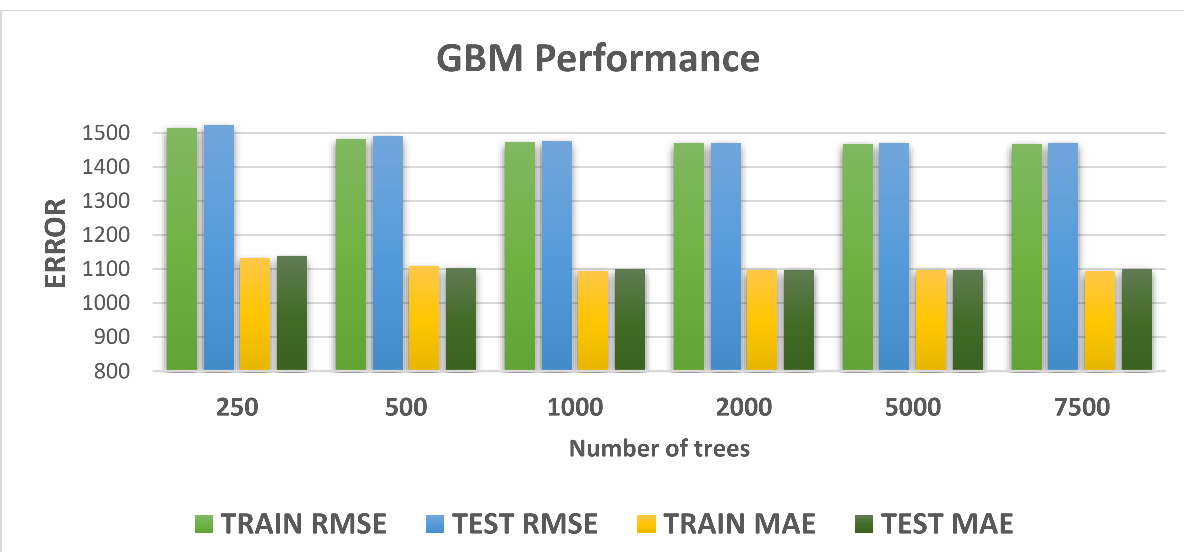
3. Shrinkage (or learning) parameter lambda – This is regularization term and affects the learning rate like as in gradient descent algorithm. Larger value of shrinkage will lead to larger update in each iteration but not lead to convergence however smaller value of shrinkage will require more of the iterations to converge i.e. will learn slowly but is more likely to give better results. For our analysis we have varied Shrinkage between 0.1 to 0.01

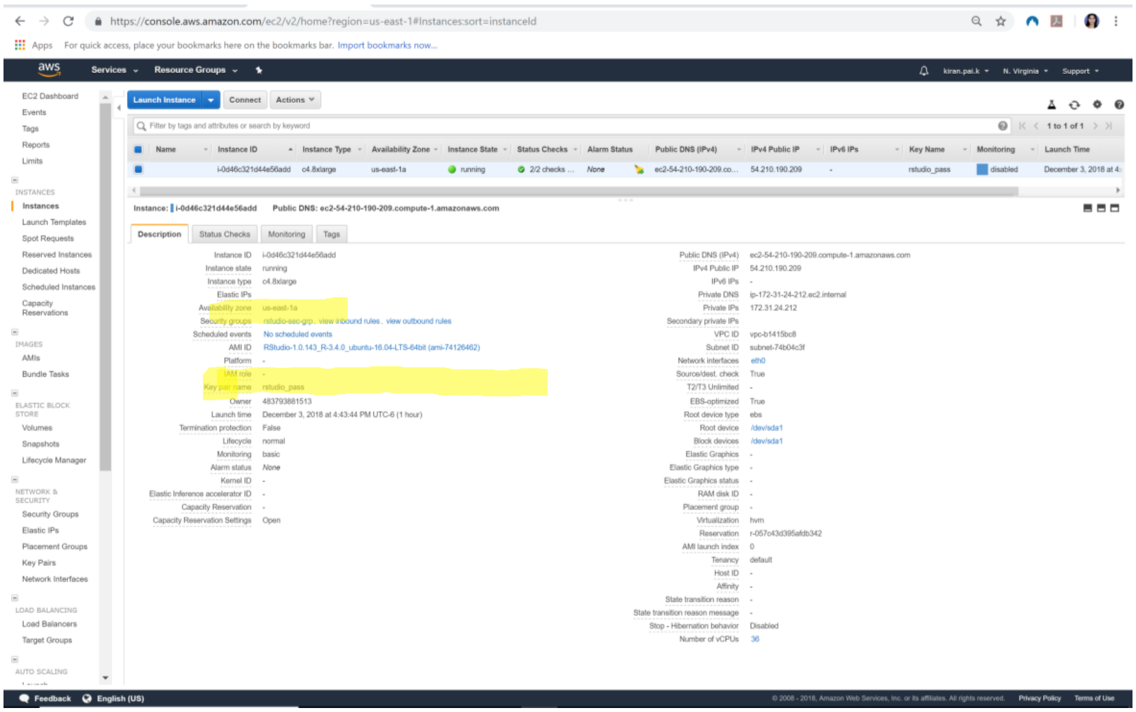
4. The subsampling rate – For our analysis we have used cv.folds = 10 to perform a 10 fold cross validation.

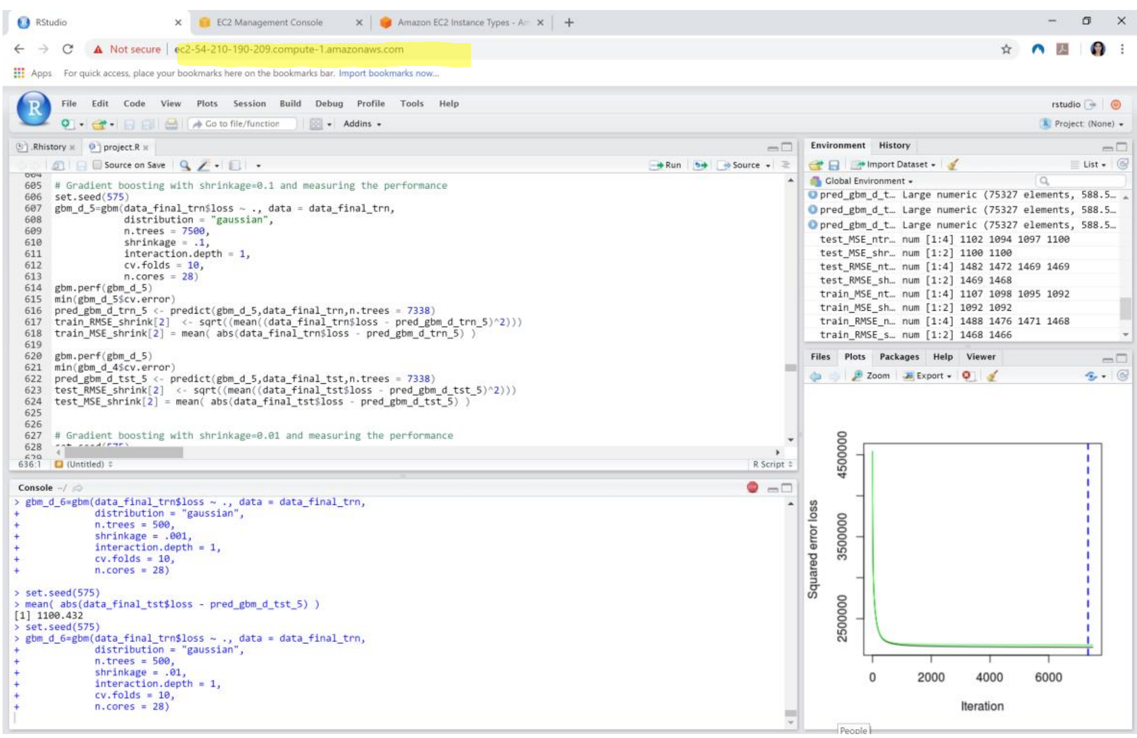
SETTING UP AWS ELASTIC COMPUTE CLOUD INSTANCE

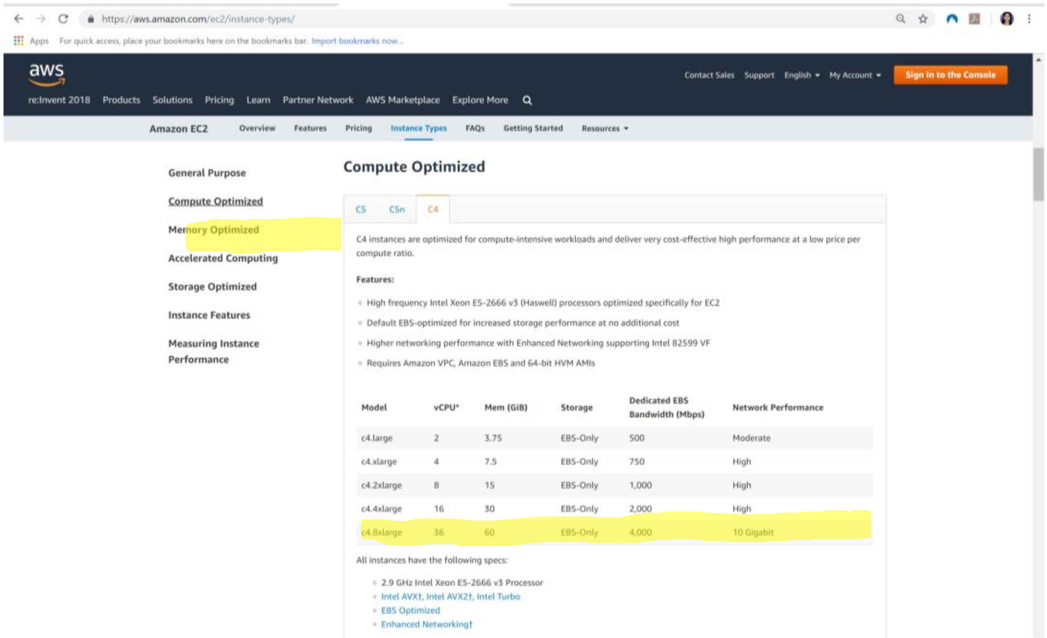
We used an EC2 compute instance from AWS and installed RStudio on it to perform parallel computing. The package used to pass the number of cores for GBM training was “parallel”. The screenshots of the same are shown below. The instance had following specification

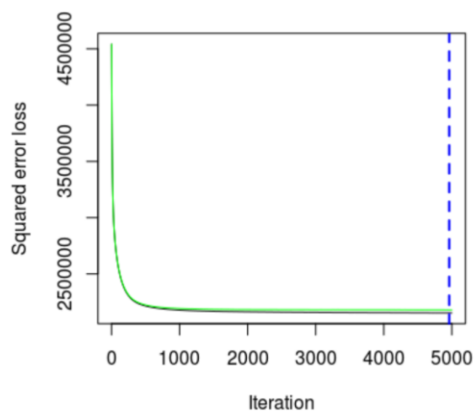
- Virtual CPU =36
- RAM = 60GB
- Dedicated EBS bandwidth = 4000 Mbps




https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceId
Launch Instance
Connect
Actions
Instance: i-0d46c321d44e56add Public DNS: ec2-54-210-190-209.compute-1.amazonaws.com
Description
Status Checks
Monitoring
Tags
Instance ID i-0d46c321d44e56add
Instance state running
Instance type c4.8xlarge
Availability zone us-east-1a
Security groups rstudio-sec-grp. view inbound rules. view outbound rules
AMI ID RStudio-1.0.143_R-3.4.0_ubuntu-16.04-LTS-64bit (ami-74126462)
Key pair name rstudio_pass
Launch time December 3, 2018 at 4:43:44 PM UTC-6 (1 hour)
Public DNS (IPv4) ec2-54-210-190-209.compute-1.amazonaws.com
IPv4 Public IP 54.210.190.209
Private DNS ip-172-31-24-212.ec2.internal
Private IPs 172.31.24.212
VPC ID vpc-b1415bc8
Subnet ID subnet-74b04c3f
Number of vCPUs 36


RStudio
EC2 Management Console
Amazon EC2 Instance Types
Not secure
ec2-54-210-190-209.compute-1.amazonaws.com
project.R
# Gradient boosting with shrinkage=0.1 and measuring the performance
set.seed(575)
gbm_d_5=gbm(data_final_trn$loss ~ ., data = data_final_trn,
distribution = "gaussian",
n.trees = 7500,
shrinkage = .1,
interaction.depth = 1,
cv.folds = 10,
n.cores = 28)
gbm.perf(gbm_d_5)
min(gbm_d_5$cv.error)
pred_gbm_d_trn_5 <- predict(gbm_d_5,data_final_trn,n.trees = 7338)
train_RMSE_shrink[2] <- sqrt((mean((data_final_trn$loss - pred_gbm_d_trn_5)^2)))
train_MSE_shrink[2] = mean(abs(data_final_trn$loss - pred_gbm_d_trn_5))
gbm.perf(gbm_d_5)
min(gbm_d_4$cv.error)
pred_gbm_d_tst_5 <- predict(gbm_d_5,data_final_tst,n.trees = 7338)
test_RMSE_shrink[2] <- sqrt((mean((data_final_tst$loss - pred_gbm_d_tst_5)^2)))
test_MSE_shrink[2] = mean(abs(data_final_tst$loss - pred_gbm_d_tst_5))
# Gradient boosting with shrinkage=0.01 and measuring the performance
Console
> gbm_d_6=gbm(data_final_trn$loss ~ ., data = data_final_trn,
+ distribution = "gaussian",
+ n.trees = 500,
+ shrinkage = .001,
+ interaction.depth = 1,
+ cv.folds = 10,
+ n.cores = 28)
> set.seed(575)
> mean(abs(data_final_tst$loss - pred_gbm_d_tst_5))
[1] 1100.432
> set.seed(575)
> gbm_d_6=gbm(data_final_trn$loss ~ ., data = data_final_trn,
+ distribution = "gaussian",
+ n.trees = 500,
+ shrinkage = .01,
+ interaction.depth = 1,
+ cv.folds = 10,
+ n.cores = 28)
Environment
History
test_MSE_ntr… num [1:4] 1102 1094 1097 1100
test_MSE_shr… num [1:2] 1100 1100
test_RMSE_nt… num [1:4] 1482 1472 1469 1469
test_RMSE_sh… num [1:2] 1469 1468
train_MSE_nt… num [1:4] 1107 1098 1095 1092
train_MSE_sh… num [1:2] 1092 1092
train_RMSE_n… num [1:4] 1488 1476 1471 1468
train_RMSE_s… num [1:2] 1468 1466
Squared error loss
2500000
3500000
4500000
0
2000
4000
6000
Iteration

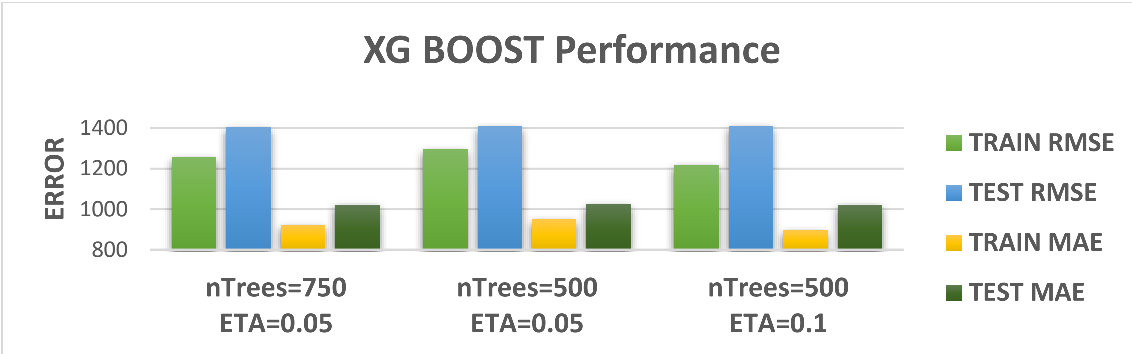




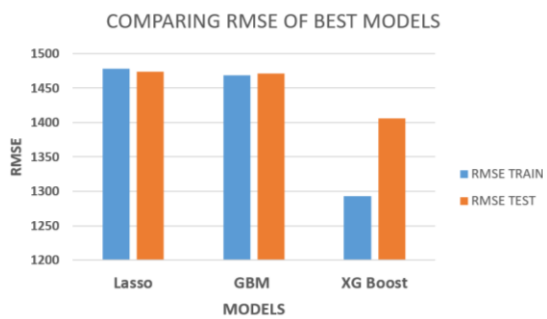
The best model among the various GBM models based on the RMSE and MAE values had the following parameters – umber of trees =5000, shrinkage = 0.1, 10 fold cross validation

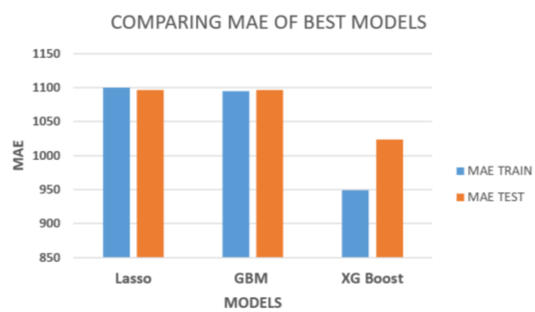
EXTREME GRADIENT BOOSTING REGRESSION

The XG boost package in R was used to develop three models as shown in the above graph. We noticed that as the number of trees increased, the model was overfitting. However, for an optimal ETA or Learning rate as well as number of trees, it gave us robust results.



CONCLUSION





Performance analysis of Gradient Boosted Trees and Lasso Regression

Since Gradient Boosted Trees is a combination of weak learners to obtain a strong learner, as we increase the number of iterations and reduced the shrinkage parameters, we obtained better results

Lasso penalizes the model estimates by shrinking coefficients to zero. Since in this case, relatively small number of predictors have substantial coefficients, the Lasso outperforms Ridge and Elastic Net

However, XG boost package provided the best results as it performs highly efficient gradient boosting. It uses multiple threads and parallelly performs most optimized regression in a swift and efficient fashion. For our problem statement, extracting a take away in the business point of view would be difficult as the variable names given in the data set were masked and not intuitive. However, when it comes to advanced models and methods which can be used for machine learning, our take away list is as follows.

1. Excluding categorical variables is not an option, instead using models like Lasso, Ridge and Elastic net provides satisfactory results by setting insignificant variable coefficients to zero.
2. The latest technologies like AWS can be used to exponentially decrease the computation times specially when running GBM with 1000+ trees and can help to obtain extremely good results.
3. XG Boost package gives surprisingly good results in no time, when used carefully. However, they are highly prone to overfitting.

REFERENCES

1. An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie, Robert Toshigami
2. https://www.saedsayad.com/docs/gbm2.pdf
3. https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a