ISEN 613 – Fall 2018

COURSE PROJECT

Final Report

Submitted By:

Preksha Beohar (UIN- 927001577)
Vikrant Raj Khatri (UIN- 526005488)
Nitesh Kumar (UIN - 327003765)
Gaurav Rai (UIN- 727004317)
Rajat Rawal (UIN- 526003006)

CONTENTS

1. EXECUTIVE SUMMARY

The insurance industry is a very competitive industry. The risk involved in the insurance industry is prominent and that makes it one of the less predictable business spheres. Further, the industry focuses on three main objectives viz. providing a superior customer experience, to achieve operational efficiency, and manage risks effectively.

Advances in big data analytics seems to bring out a revolution in the insurance industry and help them make their business model more effective. Using these techniques, the insurance companies can move away from the old technique of “understand and protect” towards “predict and prevent”.

The insurance industry gains enormous benefits from the big data analysis. Big data technologies are used nowadays to make the insurance industries more effective. Using these techniques, we predict risks and claims and further analyze and monitor them to make an effective strategy for the customer retention and attraction.

Data analytics can be used to see the trends that help us determine how much risk each applicant represents. Risk of the car insurance depends on various factors such as environment of the insured person, the make of the car and the insured person. The data analyst should have a very good understanding of these factors and develop the strategy for the company based on the important factors.

The intended models will predict the risk of an insurance claim by distinguishing whether insured car will file a claim or not. Various factors when studied in combination equip us with odds of a person filing a claim. We are tracking information for various households. A household may have several vehicles insured and need to consider all those vehicles. The information regarding in which year the vehicle was insured, model year, make, and model of vehicle would substantiate the model developed for prediction. All these parameters when keyed in to train the model will provide 99% overall accuracy with 2-5% accuracy in predicting whether the person will file a claim, whereas a trade-off of 80% overall accuracy will be able to identify 25% of those people who file a claim. The model could be designed industry specific to cater to the demands, since the trade-off to figure out the people who would file a claim leads to a surge in false prediction about the people who did not claim the insurance. Various costs such as premium, claim, people monitoring, etc. are to be kept in mind while deciding the trade-off.

Once we have trained our models and determined the best one based on the factors provided to us, we will be able to align the premiums and risks to reduce the overall cost of the insurance. This will help the insurance company to further increase its performance and operational efficiency.

Note: All the team members worked in a collaborative environment and contributed equally towards the completion of this project for the partial fulfillment of the subject grade.

2. MODEL 1 – LOGISTIC REGRESSION

DATA CLEANING / PREPROCESSING

Before applying any model to the training data, we first had to transform the dataset into a usable form. Following were the steps taken to process the data:

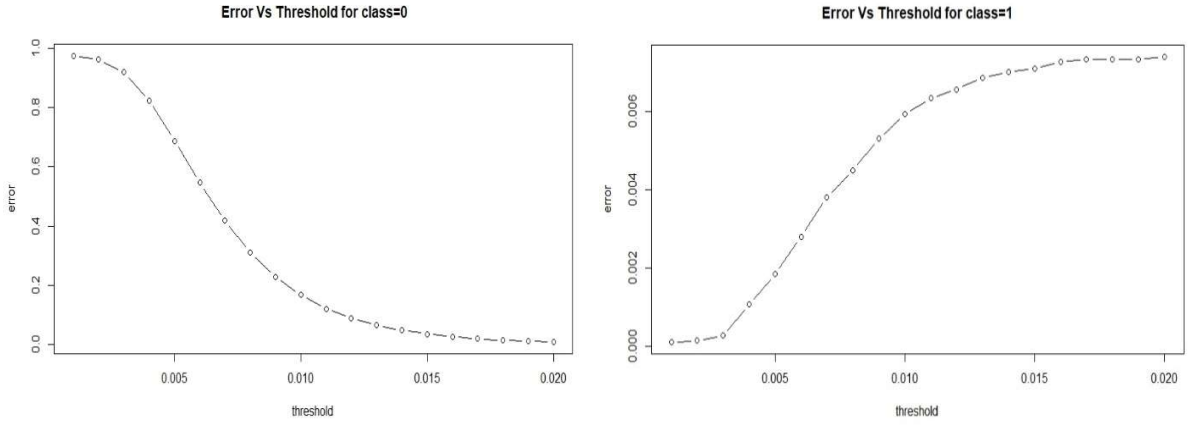
1. Added a new Categorical response variable as 'C_Claim' to the dataset with values as either '0' (if Claim_Amount' = 0) and '1' (if 'Claim_Amount' > 0) after which Claim_Amount column was omitted.
2. Imputed the empty cells present in multiple columns like Cat1, Cat2, Cat3, Cat7, OrdCat etc. by the mode value for that particular missing class (Claim 0 or Claim 1).
3. Replaced the remaining blank cells with their rows leaving us with 99908 observations out of 100,000.
4. Transformed some predictors such as Calendar_Year, Model_Year, Ord_Cat into categorical type.
5. We removed IDs and Household IDs from the predictors since their order and scale are meaningless for the prediction part.

NOTE: We applied the same data processing for the two other models as well.

FITTING LOGISTIC MODEL

Since we have a binary response we tried logistical regression model.

- Divided the dataset into training and validation set of (70:30) ratio which is considered ideal for the validation set approach.
- Fitting logistic regression model, using C_Claim as a response variable and Calendar_Year, Model_Year, Vehicle, OrdCat, Var1, Var2, Var3, Var4, Var5, Var6, Var7, Var8, Cat1, Cat2, Cat3, Cat4, Cat5, Cat6, Cat7, Cat8, Cat9, Cat10, Cat11, Cat12, NVVar1, NVVar2, NVVar3, NVVar4, NVCat as the predictors. Other predictors such as Blind_Make, Blind Model, Blind_Submodel are not included as they have factor levels of 64, 967 and 1934 respectively within and makes our model computationally challenging also we could not establish any correlation between the response and these variables.
- Since our aim is prediction and not drawing inference and we have large number of data points to keep our variance in check therefore we did not apply any of the regularization techniques.
- Calculating optimum value of threshold for minimum mis-classification:



By default for logistic regression the set threshold as 0.5 for classification purpose (if prob_1 > 0.5 then classify it as 1) but in this case, since dataset is so heavily skewed that only 720 out of 100,000 observations are in the claimed class, the probability for classifying to class_1 is much lesser than 0.5. Hence, we have to set the threshold lower than usual division of 0.5. In doing so, we would mis-classify some points actually belonging to 0's but will classify more 1's correctly. We plotted the mis-classification error and threshold on the validation set. Figure 1 shows the relation between the Error Rate (mis-classification) and the Threshold which helps us in selecting an optimum threshold value for the model.

LOGISTIC REGRESSION							
THRESHOLD	TN	FP	TP	FN	TPR	TNR	Accuracy
0.001	516	29233	221	3	0.986607	0.017345	0.024589
0.002	900	28849	220	4	0.982143	0.030253	0.037367
0.003	2196	27553	216	8	0.964286	0.073818	0.080472
0.004	5049	24700	192	32	0.857143	0.16972	0.174857
0.005	9149	20600	169	55	0.754464	0.30754	0.31088
0.006	13371	16378	140	84	0.625	0.44946	0.450772
0.007	17233	12516	110	114	0.491071	0.57928	0.578621
0.008	20425	9324	89	135	0.397321	0.686578	0.684416
0.009	22881	6868	65	159	0.290179	0.769135	0.765556
0.01	24735	5014	46	178	0.205357	0.831457	0.826777
0.011	26141	3608	34	190	0.151786	0.878719	0.873286
0.012	27062	2687	27	197	0.120536	0.909678	0.90378
0.013	27789	1960	18	206	0.080357	0.934115	0.927735
0.014	28304	1441	14	210	0.0625	0.951555	0.944784
0.015	28688	1061	11	213	0.049107	0.964335	0.957495
0.016	28972	777	6	218	0.026786	0.973881	0.966803

The above table shows us the confusion matrix generated at each threshold value and the accuracy of the model associated at each threshold. Since it is inappropriate to choose a threshold for the model giving high TPR but unacceptable TNR because we have to simultaneously predict the true positive cases along with the true negatives, thus we have to select a model that strikes a balance between the TPR and TNR so as to not sacrifice the accuracy for a better prediction of claims. We selected a threshold of 0.01 as the optimal, we can accurately predict more than 20% of the claims with an overall Accuracy of **80%** (our assumed optimum baseline accuracy level)

<u>For threshold of 0.01</u>

Training Error | TPR= 29% and Overall Error = 16%

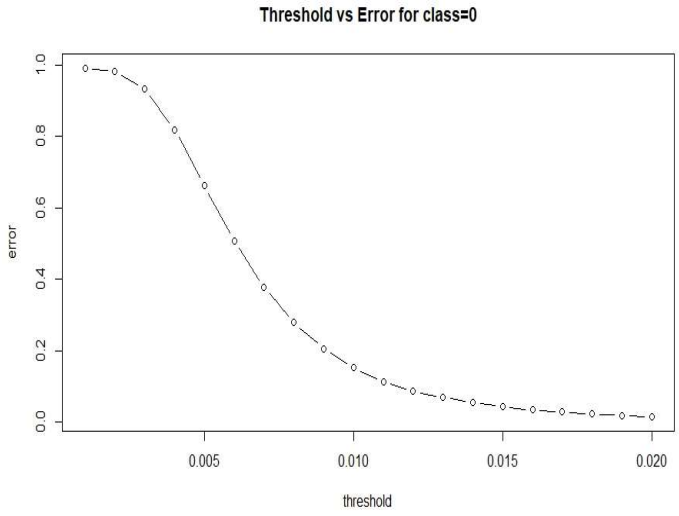
Test Error | TPR= 21% and Overall Error= 17.3%

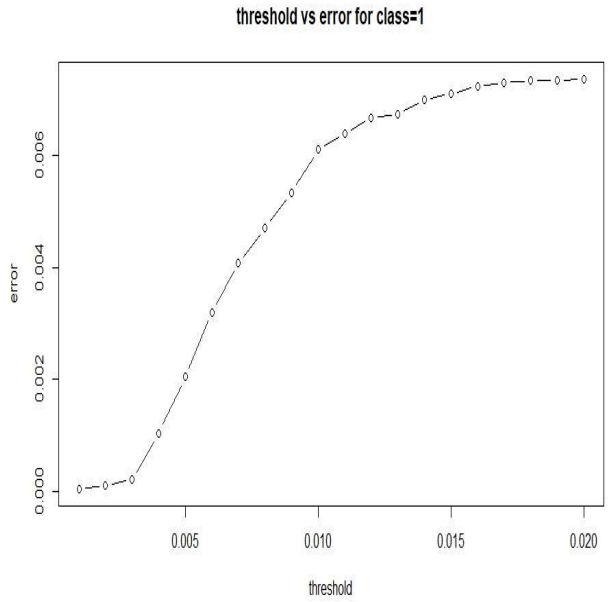
3. MODEL 2 – Linear Discriminant Analysis

FITTING LDA MODEL

Since we have binary output present in the dataset, Therefore LDA is a good candidate for our model.

- Divided the dataset into test data and training data (30:70) using the Validation Set approach.
- Firstly, the LDA model with C_Claim as our response variable and all the remaining variables as our predictors was build. After running the model, we determined the significant variables and then again trained our model just based on the significant variables.
- We fitted the LDA model using C_Claim as a response variable and Cat3, Cat6, NVVar2, NVVar3, Var1, Var2, Var3, Var4, Var5, Var6, Var7, Var8 as the predictors.
- Predictors such as Blind_Make, Blind Model, Blind_Submodel are also not included as they have factor levels of 64, 967 and 1934 respectively and do not contribute significantly to our predictions.
- Calculating optimum value of threshold for minimum mis-classification





- Evidently the graph above shows the True negative rate increasing with an increase in threshold value which implies Class 1 being classified to Class 0
- Inversely to the trend shown by Class 0, the true positive rate decreases with an increase in threshold which implies Class 1 being classified to Class 0
- Keeping a baseline accuracy is important to deal with this kind of situation to strike a perfect trade-off
- The model was run on different threshold values to pick one that meets our expectation

LDA							
THRESHOLD	TN	FP	TP	FN	TPR	TNR	Accuracy
0.001	19	29730	223	1	0.995535	0.00063	0.008073933207
0.002	329	29420	221	3	0.986607	0.01105	0.0183498482
0.003	1781	27968	218	6	0.973214	0.05986	0.06669335735
0.004	5186	24563	193	31	0.861607	0.17432	0.1794615154
0.005	9906	19843	163	61	0.727678	0.33298	0.3359356754
0.006	14536	15213	128	96	0.571428	0.48862	0.4892403163
0.007	18446	11303	102	122	0.455357	0.62005	0.6188236079
0.008	21414	8335	83	141	0.370535	0.71982	0.7172121576
0.009	23590	6159	64	160	0.285714	0.79296	0.7891769259
0.01	25216	4533	41	183	0.183035	0.84762	0.8426583926
0.011	26358	3391	32	192	0.142857	0.88601	0.8804590798
0.012	27175	2574	24	200	0.107142	0.91347	0.9074500384
0.013	27707	2042	22	202	0.098214	0.93135	0.9251326194
0.014	28133	1616	14	210	0.0625	0.94567	0.939078504
0.015	28468	1281	11	213	0.049107	0.95693	0.9501551396
0.016	28748	1001	7	217	0.03125	0.96635	0.9593634271

The figure shows us the confusion matrix generated at each threshold value and the accuracy of the model associated at each threshold. Since it is inappropriate to generate a model with either TPR as 1 or TNR as 1, thus we have to select a model that has an equilibrium between the TPR and TNR and has a good accuracy. We selected a baseline overall accuracy of 80% at a threshold of 0.009 which accurately classifies 28.57% of Class 1.

For threshold of 0.009

Training Error | TPR= 35.6% | Overall Error= 22%

Test Error | TPR= 28.57% and Overall Error= 21%

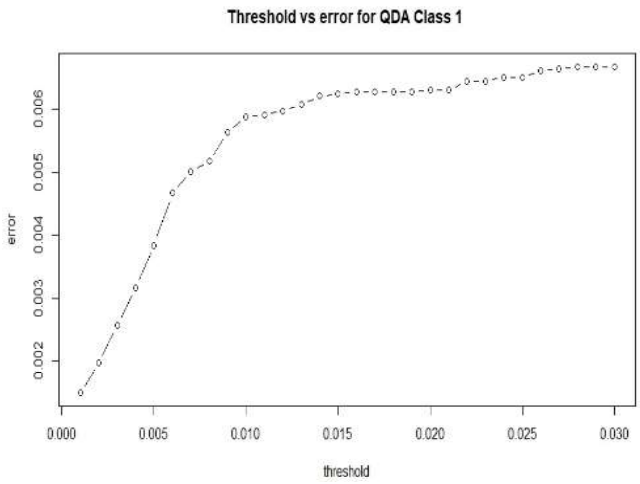
Technically due to the bias characteristics the training error should have been low but it boils down to the skewness of dataset and the number of data that could be misclassified Since training set is huge compared to test data, the change in threshold will lead to more of Class 0 being classified to Class 1.

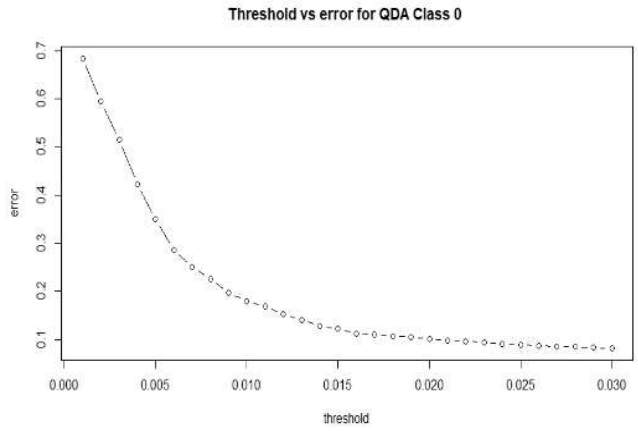
4. MODEL 3 – Quadratic Discriminant Analysis

FITTING QDA MODEL

Since we have binary output in the dataset, we tried fitting QDA model to the data.

- First, we tried running the QDA model with C_Claim as our response variable and all the remaining variables as our predictors. After running the model, we determined the significant variables and then again trained our model just based on the significant variables.
- We fitted the QDA model using C_Claim as a response variable and Cat3, Cat6, NVVar2, NVVar3, Var1, Var2, Var3, Var4, Var5, Var6, Var7, Var8 as the predictors
- Other predictors such as Blind_Make, Blind Model, Blind_Submodel are not included as they have factor levels of 64, 967 and 1934 respectively within and do not contribute much to our predictions.
- Calculating optimum value of threshold for minimum mis-classification





- Evidently the graph above shows the True negative rate increasing with an increase in threshold value which implies Class 1 being classified to Class 0 and inverse for True positive rate
- Inversely to the trend shown by Class 0 , the true positive rate decreases with an increase in threshold which implies Class 1 being classified to Class 0
- Keeping a baseline accuracy is important to deal with this kind of situation to strike a perfect trade-off
- The model was run on different threshold values to pick one that meets our expectation

QDA							
THRESHOLD	**TN**	**FP**	**TP**	**FN**	**TPR**	**TNR**	**Accuracy**
0.001	5074	24675	196	28	0.875	0.170560355	0.1758249091
0.002	7323	22426	183	41	0.8169642857	0.2461595348	0.2504253828
0.003	9380	20369	173	51	0.7723214286	0.3153047161	0.3187201815
0.004	10650	19099	164	60	0.7321428571	0.3579952267	0.3607913789
0.005	11942	17807	158	66	0.7053571429	0.401425258	0.4036966603
0.006	13879	15870	145	79	0.6473214286	0.4665366903	0.4678877657
0.007	16350	13399	129	95	0.5758928571	0.5495983058	0.5497948153
0.008	18440	11309	108	116	0.4821428571	0.6198527682	0.6188236079
0.009	19799	9950	97	127	0.4330357143	0.665534976	0.6637974177
0.01	21220	8529	79	145	0.3526785714	0.7133012874	0.7106062123
0.011	22439	7310	70	144	0.3271028037	0.7542774547	0.7509758783
0.012	23629	6120	69	155	0.3080357143	0.7942787993	0.7906449138
0.013	24455	5294	53	171	0.2366071429	0.8220444385	0.8176692356
0.014	24971	4778	50	174	0.2232142857	0.8393895593	0.8347846395
0.015	25277	4472	47	177	0.2098214286	0.8496756193	0.8448937377
0.016	25524	4225	47	177	0.2098214286	0.8579784194	0.8531344877

The table shows us the confusion matrix generated at each threshold value and the accuracy of the model associated at each threshold. Since it is inappropriate to generate a model with either TPR as 1 or TNR as 1, thus we have to select a model that has an equilibrium between the TPR and TNR and has a good accuracy. We selected a baseline overall accuracy of 80% at a threshold of 0.012 which accurately classifies 30.8% of Class 1.

For threshold of 0.012:

Training Error | TPR = 39.5% | Overall Error = 25%

Test Error | TPR= 30.8% and Overall Error= 21%

Technically due to the bias characteristics the training error should have been low but it boils down to the skewness of dataset and the number of data that could be misclassified Since training set is huge compared to test data, the change in threshold will lead to more of Class 0 being classified to Class 1.

Note: Code for this model has been submitted in a separate .R file.

5. Comparison of Models

This project deals with accurately predicting whether a vehicle will claim insurance or not, based on 33 predictor variables. Our aim is to predict a categorical variable C_Claim which takes a value of 0 (if no insurance is claimed) and 1 (if insurance is claimed) using these 33 predictors. Here the training dataset is very unbalanced in the sense that the elements in class_0(unclaimed insurance) are far much more than the elements in class_1(cases where insurance was claimed).

The insurance company would want to have a good True Positive Rate(TPR) which means to predict points accurately to class 1 because they want to predict which customers are in a high-risk group but at the same time overall accuracy of this model shouldn't be compromised much. which means we have to keep misclassifications in either of the 0 or 1 class is check because if we classify too many points in the class 1 we will definitely identify more points in high risk group but we will also misclassify many points which actually belong to class 0 into class 1.

So, decreasing the True Negative Rate(TNR) which is also not advisable in this scenario. *Our criteria while choosing the right model for classification is to strike a balance between the* ***True Positive Rate*** *and* ***True Negative Rate.***

In our initial exploratory data analysis, we could not establish any pattern or correlation in-between the predictors and the response variable mostly because of the fact that the dataset had a lot of categorical predictors variables and a categorical response too which normally create a problem in establishing any significant pattern between the variables. We tried out different models such as Decision Trees, Random Forests, SVMs with linear and radial kernels and finally went ahead with the simpler **Logistic, LDA and QDA** models as they were the most promising of all the models we tried.

LOGISTIC REGRESSION			
THRESHOLD	TPR	TNR	Accuracy
0.001	0.986607	0.017345	0.024589
0.002	0.982143	0.030253	0.037367
0.003	0.964286	0.073818	0.080472
0.004	0.857143	0.16972	0.174857
0.005	0.754464	0.30754	0.31088
0.006	0.625	0.44946	0.450772
0.007	0.491071	0.57928	0.578621
0.008	0.397321	0.686578	0.684416
0.009	0.290179	0.769135	0.765556
0.01	0.205357	0.831457	0.826777
0.011	0.151786	0.878719	0.873286
0.012	0.120536	0.909678	0.90378
0.013	0.080357	0.934115	0.927735
0.014	0.0625	0.951555	0.944784
0.015	0.049107	0.964335	0.957495
0.016	0.026786	0.973881	0.966803

LDA			
THRESHOLD	TPR	TNR	Accuracy
0.001	0.995536	0.000639	0.008074
0.002	0.986607	0.011059	0.01835
0.003	0.973214	0.059868	0.066693
0.004	0.861607	0.174325	0.179462
0.005	0.727679	0.332986	0.335936
0.006	0.571429	0.488621	0.48924
0.007	0.455357	0.620054	0.618824
0.008	0.370536	0.719823	0.717212
0.009	0.285714	0.792968	0.789177
0.01	0.183036	0.847625	0.842658
0.011	0.142857	0.886013	0.880459
0.012	0.107143	0.913476	0.90745
0.013	0.098214	0.931359	0.925133
0.014	0.0625	0.945679	0.939079
0.015	0.049107	0.95694	0.950155
0.016	0.03125	0.966352	0.959363

QDA			
THRESHOLD	TPR	TNR	Accuracy
0.001	0.875	0.17056	0.175825
0.002	0.816964	0.24616	0.250425
0.003	0.772321	0.315305	0.31872
0.004	0.732143	0.357995	0.360791
0.005	0.705357	0.401425	0.403697
0.006	0.647321	0.466537	0.467888
0.007	0.575893	0.549598	0.549795
0.008	0.482143	0.619853	0.618824
0.009	0.433036	0.665535	0.663797
0.01	0.352679	0.713301	0.710606
0.011	0.327103	0.754277	0.750976
0.012	0.308036	0.794279	0.790645
0.013	0.236607	0.822044	0.817669
0.014	0.223214	0.83939	0.834785
0.015	0.209821	0.849676	0.844894
0.016	0.209821	0.857978	0.853134

So, to choose from the models we first decided a **baseline of overall accuracy 80% on a Validation set**. which will prevent us from choosing a model with too high a TPR so as to reduce the True Negative rate(TNR), so effectively we chose the model with the highest True Positive Rate (TPR) within a minimum baseline accuracy of 80%. We got a max TPR of 30.08% with QDA followed by LDA with 28.5% followed by logistic regression with 20.5%.

Therefore, our final model to be considered for test data classification will be our QDA model with a threshold of 0.012

6. EVALUATING TEST RESULTS

For accuracy as the sole decision criteria we put threshold of 0.5 in our QDA model

- We got an overall Accuracy rate of 98.5%.
- **Test Error |** 1 – Accuracy = 1.5%

Actual values	Predicted_0	Predicted_1
0	49250	383
1	363	4

6. SCOPE OF IMPROVEMENT

Earlier the QDA model was designed to strike a balance in the trade-off. Since the objective of the competition solely lies on the test accuracy. The threshold has been changed to cater the needs.

The only change that has been made in the entire code is the scaling up of threshold value to 0.5 from 0.012.

Earlier model

Our accuracy on that model at threshold 0.012 was is 84.1% and true positive rate (TPR) is 23.7%

Actual values	Predicted_0	Predicted_1
0	41924	7709
1	280	87

Final Model

For accuracy as the sole decision criteria we put threshold of 0.5 and get an overall accuracy rate of 98.5%.

Actual values	Predicted_0	Predicted_1
0	49250	383
1	363	4