# Credit Scoring Coursework 2

*Alexander Pinches*

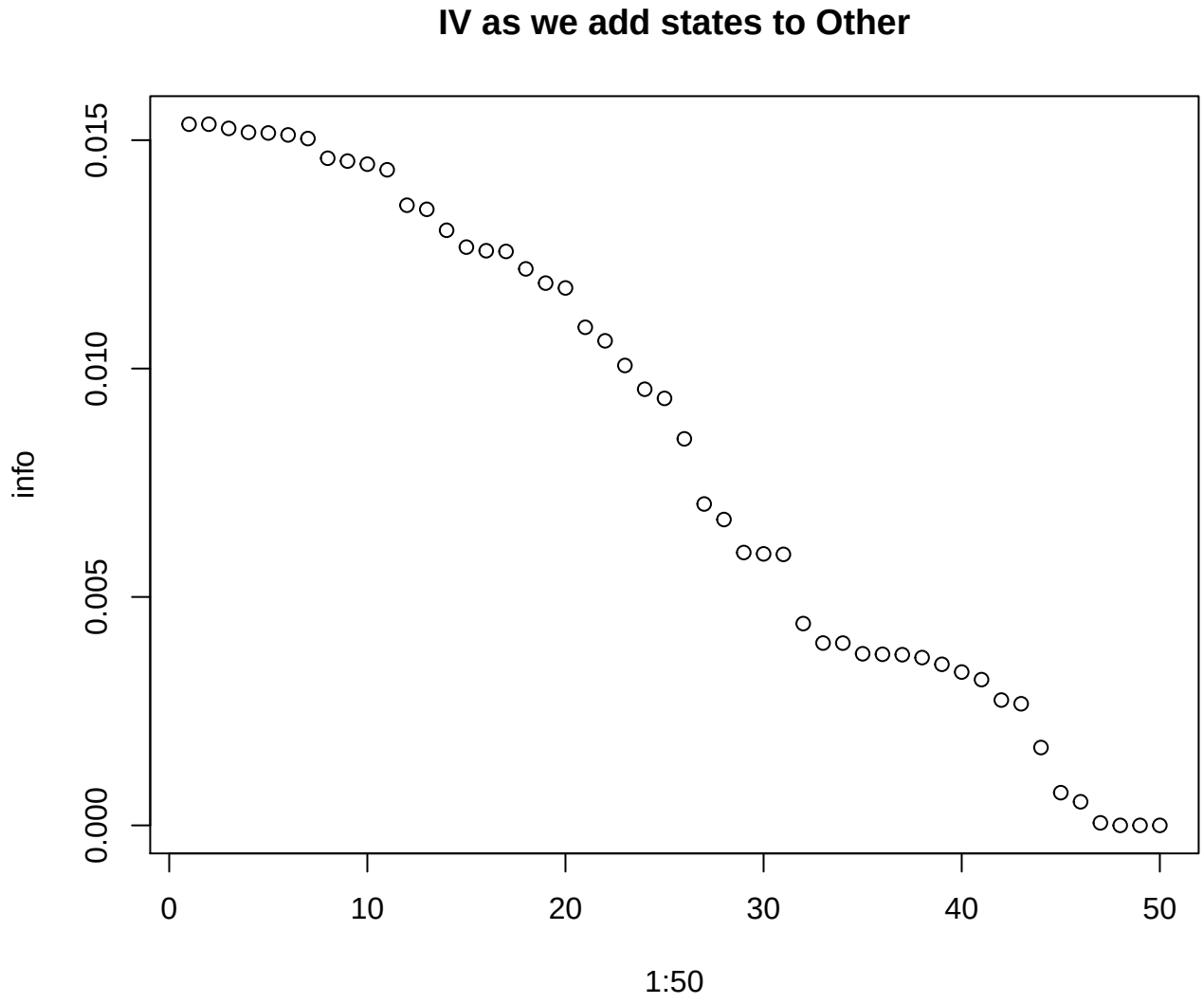
*2 January 2019*

## Question 1

### Data preperation and validation

We can summarise the data frame with `summary()`in R to see the types of each predictor in the data frame the values they take and any NA's. We also use `str()` to check the types of each variable and plot histograms of each variable to show their distributions so we can transform them if needed.

We see that `avg_cur_ball` has 5 `NA`'s as this is a tiny proportion of the data set it is best to remove these. We also see the types's of some predictors need to be changed such as `purpose_p` and we should scale the data to improve the quality of the model and the training speed. We change purpose from a string to factors, term to a factor and grade from a number to a factor as this is more appropriate given their descriptions. We scale by taking the logarithm and adding a constant greater than 1 before taking the loarithm so that zero values and values below 1 don't go asymtotically to infinity. This is done for all the variables other than `grade`, `home_ownership`, `verification_status`,`term`,`initial_list_status`, `addr_state` and `issue_d`. Some factor levels aren't used so we remove them to reduce unnecessary complexity in the dataframe. We also notice that the `addr_state` variable has 50 factors and some may have a low number of members using `dplyr` the outputted tibble shows this to be true. To combat this can first combine smaller factors into a single factor `Other` of other states we determine the number of the smallest states to combine by plotting the information value as we successively add states to `Other`.



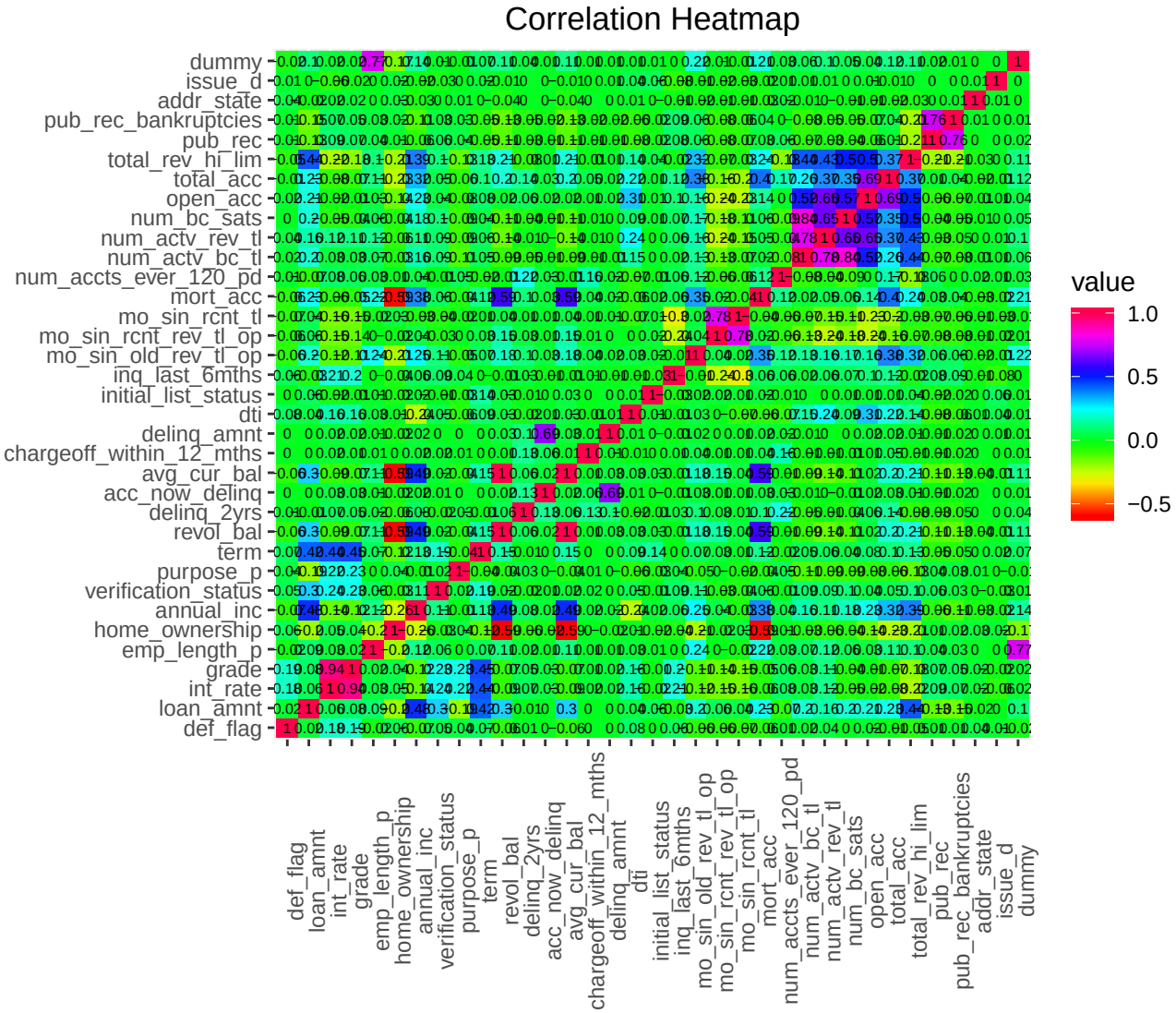

Looking at the plot we want the point before which IV drops from its initial value as this is the least information lost but reduces the effect of small factors in the training data creating false inferences. The point at which it drops below 0.015 is a 7. So we will combine the 7 smallest states by number of members into an other category. Then instead of leaving them as factor type we will replace them with their weights of evidence.We do this because otherwise the number of factors in this group will be high reducing the performance of our model. Replacing it with the weights of evidence means we can use a numerical scale which is already scaled as each WOE is the probability of that state.

After scaling the variables and changing their types we need to decide what to do with the `NA`'s in `emp_length_p` as they account for a reasonable amount of the data we can't discount them from the model. We could impute them with their mean or by regression the latter of which we will do with logistic regression training on the data which doesn't have `emp_length_p`=`NA` then predicting and replacing the `NA`'s with predicted values. We also calculate the correlation of other predictors with employment length and only include ones with a significant p-value at the 5% level. We then recombine the data and shuffle. We can also segment the model and train one model where employment length is known and another on if it's `NA` we will do this also for comparison. As if the `NA`'s are `NA` for a reason which is not clear from the provided information this may provide a better overall model than imputation will. Also as the 10+ category is a different size than the rest of the factors we create a dummy variable so that the model can learn this. Which simply says if it is or isn't in the 10+ category using a true/false binary.

Comparing the histograms from before the transformations to after we see that the distribution of these variables should allow for better models and faster training. We also use the histograms to decide the constant to add inside the log as for variables with more sparsley distributed points a larger constant will bring them closer together to allow for better models.

### Variable Selection

Below is a heatmap of the correlations of all the predictors . We can single out those relating to default and calculate their p-values. These are mostly likely the predictors we should use in our models and we shouldnt include the others when we test them at a significance level such as 0.05 as in this case.




Below the significant predictors from our p-values is printed. Most of the predictors are included but a few are removed as they likely have little predictive value. So should be discounted to make the model train faster and reduce the chance of overfitting.

```
##  [1] "loan_amnt"             "int_rate"
##  [3] "grade"                 "emp_length_p"
##  [5] "home_ownership"        "annual_inc"
##  [7] "verification_status"   "purpose_p"
```

```
##  [9] "term"                  "revol_bal"
## [11] "delinq_2yrs"           "avg_cur_bal"
## [13] "dti"                   "inq_last_6mths"
## [15] "mo_sin_old_rev_tl_op"  "mo_sin_rcnt_rev_tl_op"
## [17] "mo_sin_rcnt_tl"        "mort_acc"
## [19] "num_accts_ever_120_pd" "num_actv_bc_tl"
## [21] "num_actv_rev_tl"       "open_acc"
## [23] "total_acc"             "total_rev_hi_lim"
## [25] "pub_rec"               "pub_rec_bankruptcies"
## [27] "addr_state"            "issue_d"
## [29] "dummy"
```

To confirm this we can add and remove predictors from a model only containing those that are significant to improve the AIC of the model. This will give us an idea of what predictors to include in the models we train. It will penalise adding unnecessary variables by nature of the AIC. This is very computationally expensive as we train every model with each combination of adding or removing a variable and calculate its AIC. So we will start with our initial guess as the significant values and move in both directions adding predictors and pruning them at each step to improve AIC we should reach the optimum faster than if we started at a model with no predictors. We use `stepAIC()` from the `MASS` package you can also use `step()` from base R. We could also only perform this stepwise algorithm forwards or backwards however as we already have a guess as to the optimal model from our significance tests it will faster most likely to start from this point and go in both directions.

## Models

### Logistic Regression

After perfroming this we see that we stay at the model consisting of the significant predictors only and to check this we can train a logistic model on all and just the significant predictors and compare their AUC's. To test their performance in a new dataset we will use bootstrapping where we select a training dataset randomly with replacement from the data and use the unselected data as the validation data set. We could use k-folds cross validation also as an alternative where we split the data frame into k portions (folds) and use one as a validation dataset and the rest as training data and change the validation set each loop until each fold has been the validation set once and only once. Usually bootstrapping is the better out of the two. Both though are usually significantly better than selecting a proportion of the data to train on and test on. We will use bootstrapping through out and compare the AUC's of each bootstrap and their mean to decide on the optimal model.

```
##     auc_all          auc_sig
##  Min.   :0.6871   Min.   :0.6871
##  1st Qu.:0.6906   1st Qu.:0.6906
##  Median :0.6908   Median :0.6908
##  Mean   :0.6912   Mean   :0.6912
##  3rd Qu.:0.6936   3rd Qu.:0.6936
##  Max.   :0.6938   Max.   :0.6938
```

Looking at the AUC's of the logistic regression model with all verses the significant predictors we get the same AUC each time so these predictors clearly aren't useful for predicting default based on this dataset and we are probably correct in not including them. We can also apply a logistic regression model on the two segments we created and calculate the AUC to compare their performance. Using the significant predictors for all the following models. We can now tran the logistic regression model for our segmented model training on the set with `NA`'s and without with the significant predictors but in the former case we omit the `emp_length_p` variable as it contains only one value.

```
##      auc_NA          auc_wo_NA
##  Min.   :0.6113   Min.   :0.6867
##  1st Qu.:0.6117   1st Qu.:0.6912
##  Median :0.6141   Median :0.6916
##  Mean   :0.6202   Mean   :0.6921
##  3rd Qu.:0.6265   3rd Qu.:0.6938
##  Max.   :0.6376   Max.   :0.6974
```

Above we can see the AUC of each segment of the segmented model. The AUC for without `NA`'s is higher slightly on average than the unsegmented model but the AUC of the other segment is significantly lower. This would make the segmented model worse than the imputed full model unless we can find a better model for this segment. ###CART We could try a tree based model CART on the models to see if we can acheive a higher AUC.

```
##   auc_unpruned      auc_pruned         auc_NA       auc_NA_pruned
##  Min.   :0.6565   Min.   :0.6565   Min.   :0.5028   Min.   :0.5028
##  1st Qu.:0.6611   1st Qu.:0.6611   1st Qu.:0.5035   1st Qu.:0.5035
##  Median :0.6619   Median :0.6619   Median :0.5088   Median :0.5088
##  Mean   :0.6610   Mean   :0.6610   Mean   :0.5181   Mean   :0.5181
##  3rd Qu.:0.6620   3rd Qu.:0.6620   3rd Qu.:0.5365   3rd Qu.:0.5365
##  Max.   :0.6637   Max.   :0.6637   Max.   :0.5387   Max.   :0.5387
##    auc_wo_NA      auc_wo_NA_pruned
##  Min.   :0.6615   Min.   :0.6615
##  1st Qu.:0.6638   1st Qu.:0.6638
##  Median :0.6662   Median :0.6662
##  Mean   :0.6662   Mean   :0.6662
##  3rd Qu.:0.6694   3rd Qu.:0.6694
##  Max.   :0.6701   Max.   :0.6701
```

Note that pruning the tree kept the tree kept the tree the same. The AUC of all these models is worse than the logistic regression model suggesting we should stick to logistic regression or a model which isn't CART.

**Interaction terms with logistic regression**

We can also include interaction terms to improve the AUC of our model. We could add them step wise but as testing all the interaction terms increases the total number of variables from 34 to 595 (as the variables are commutative) if you include all the predictors. This is therefore too computationally expensive as `step()`and `stepAIC()` both require you to train the full model and starting model first and we run into memory constraints training a 595 variable model. Instead we can add interaction terms that we think would likely increase our predictive ability using our knowledge and educated guesses. In this case the two I try are interest rate with annual income and intrest rate with revolving balance as the interaction of these terms would I think tell us more about their chance of defaulting. As high income means you can afford higher interest rates and a large balance also means you are less likely to not have money to make payments if the interest rate is high.

```
##      auc_int          auc_int_NA      auc_int_wo_NA
##  Min.   :0.6871   Min.   :0.6117   Min.   :0.6899
##  1st Qu.:0.6889   1st Qu.:0.6145   1st Qu.:0.6923
##  Median :0.6895   Median :0.6177   Median :0.6926
##  Mean   :0.6894   Mean   :0.6279   Mean   :0.6945
##  3rd Qu.:0.6896   3rd Qu.:0.6413   3rd Qu.:0.6980
##  Max.   :0.6920   Max.   :0.6545   Max.   :0.6995
```

Above are the summaries of their calculated AUC's over the 5 bootstraps. We see slight improvements in AUC on average however this could be due to test to test variance as the difference is small. We would need to increase the number of bootstraps to confirm if this difference is significant but this will also greatly increase computation time for testing. #Question 2 If we split the data by term and predict on the signficant predictors again but without term as a predictor with a logistic regression model. We can compare it with the previous models to determine if its a good idea and which model performs best on our data.

```
##      auc_36           auc_60
##  Min.   :0.6875   Min.   :0.6582
##  1st Qu.:0.6880   1st Qu.:0.6598
##  Median :0.6905   Median :0.6617
##  Mean   :0.6917   Mean   :0.6622
##  3rd Qu.:0.6926   3rd Qu.:0.6652
##  Max.   :0.6997   Max.   :0.6663
```

We see looking at the summary of the AUC's over the bootstraps that we get similar performance to the original logistic regression model for `term`= 36 but a much lower AUC for `term`= 60. Therefore as the 60 level is a large proportion of the dataset $\backsim \frac{1}{3}$ I would recommend that the lender doesn't segment the model. This is likely because knowing the term of the load is important information to determine default. This is firstly shown by it being a signficant predictor in our p-value test and we can also check its infomation value shown below.

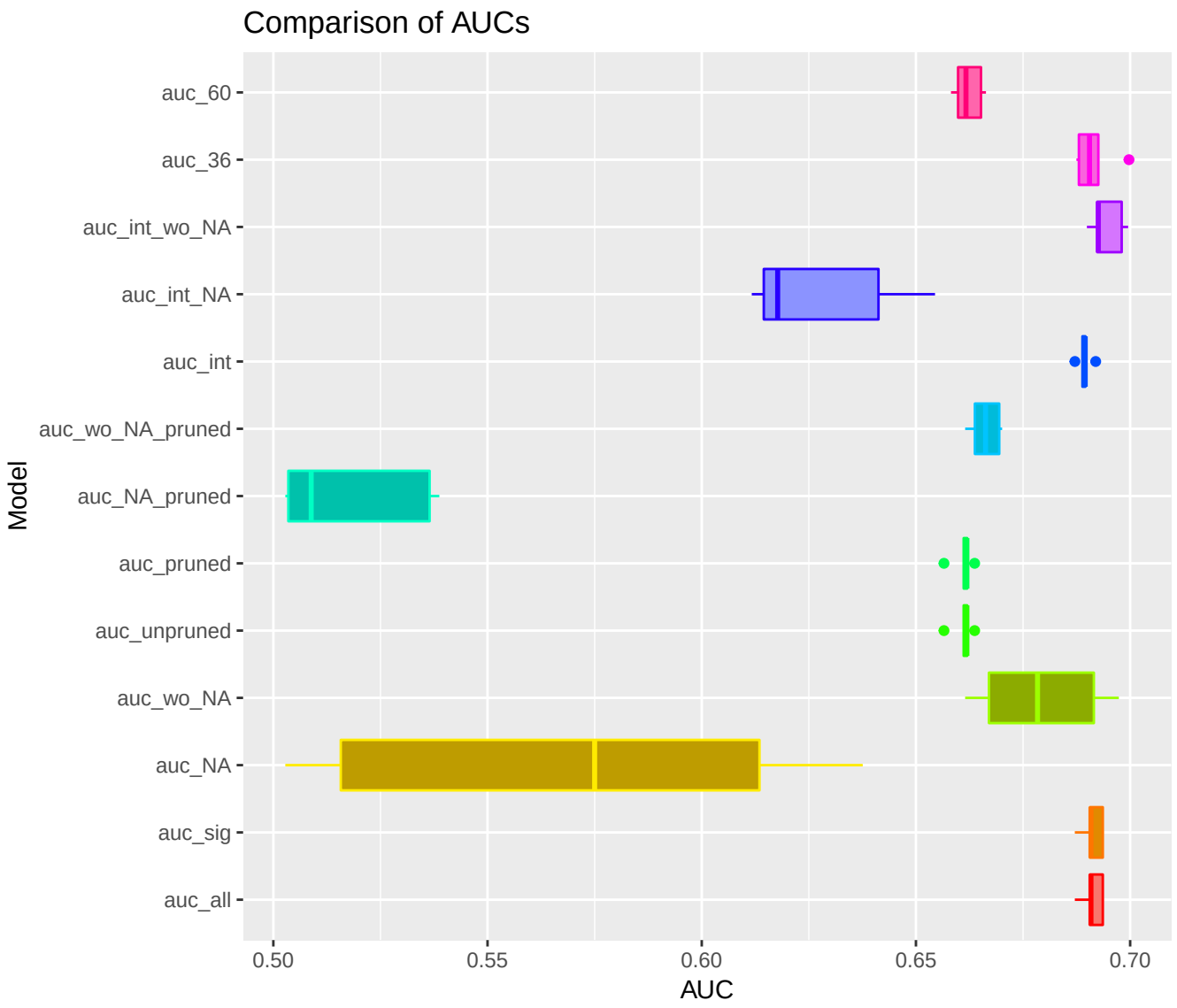
```
## [1] 0.04246723
## attr(,"howgood")
## [1] "Somewhat Predictive"
```

This shows it is useful for prediction and when we segment the model we reduce its predictive ability thus shown be the reduced AUC of the logistic regression models. The improved preditive ability of the model will likely outweigh the improvement in marketing for segmenting the model. As I doubt the improvement in marketing will be that substantual but the difference in AUC is.

## Conclusion

We can create a boxplot of the AUC's of all the models to allow us to compare them and determine which is the best model for predicting default based their average and range of AUC's over the bootstraps. As we want a high average AUC but a high range may suggest the model can overfit easily on some datasets so is not a good model.

We see in the plot that the best model is the logistic regression model without interaction terms and that isn't segmented. The AUC is higher than that of the first model 0.6730 verses 0.6912 on average it also has a small range. However the improvement although significant is small and suggests the model still isn't a very good predictive model. Also this model uses a lot more predictors and thus takes longer to train than the previous model. Also we used WOE for `addr_state` and combined the smaller factors unlike the first model which kept them all as seperate factors. It is unlikely this improvement is due to overfitting either as we used statistical tests (p-values) to confirm relationships and stepwise addition and removal of variables using the AIC so as to only include variables that are relevant and confirmed the performance of the model by using bootstrapping. We didn't perform bootstrapping on the first model so testing of the first model doesnt give us as good of an idea about how well the model will perform on a new dataset. Another model which may offer improved performance is a penalised regression technique such as LASSO as some variables have extreme values which may skew the data. Also seeing if other model segmentations may be better using domain knowledge to decide on segmentations or calculating the information gain for the segments. If




we had access to a HPC cluster we could test using more and other interaction terms as we are unable to test all of them here easily due to memory constraints and my domain knowledge is arguably not sufficient to be able to guess the best interaction terms to include.