Data Mining 2 Final Project

Mo Sharieff

Predictive Questions

1) Which categorical or numerical variable is the best at predicting the outcome of the Gift Amount donated?

2) How do the different types of predictive models differ in terms of accurately predicting the Gift Amount?

The only dependent variable that will be used in this analysis is the Gift Amount received by the organization.

The unit of analysis in this report is the Gift Amount, most specifically the classified Gift Amount shown later in this presentation (Low Donation/High Donation).

Initial Pre-Processing Method

1) Use Python to eliminate rows and columns with unnecessary characters

2) Create a new .csv file written from Python and import those .csv files into Microsoft SQL Server

3) Import the Bike Events and Bike Donations datasets from SQL into R

4) Join the two datasets by “EventID”

5) Convert appropriate string values to numeric, and eliminate rows with “NA”

6) Use the fastDummies libraries to convert categorical variables into binary

7) Take out the categorical variables using the select(-category) method, and leave the binary variables in the dataset

8) Use the (x – min(x))/(max(x) – min(x)) to normalize the numeric data

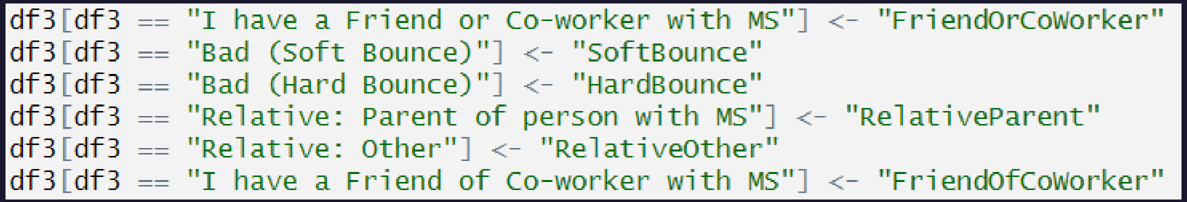
9) Once you are ready with the steps above, split the normalized set and the original set (which will be used to convert normalized predicted values back into regular numeric form)

10) The number of records imported from the BikeDonations dataset are the top 50,000 rows (had major issues w/ random sampling)

11) The number of records imported from the BikeEvents dataset are 87 rows.

Pre-Train Pre-Processing Method

Before I prepared the data to be passed into the predictive model, I had to make some adjustments to some of the categorical variables because they had special characters that did not parse properly.



```
df3[df3 == "I have a Friend or Co-worker with MS"] <- "FriendOrCoWorker"
df3[df3 == "Bad (Soft Bounce)"] <- "SoftBounce"
df3[df3 == "Bad (Hard Bounce)"] <- "HardBounce"
df3[df3 == "Relative: Parent of person with MS"] <- "RelativeParent"
df3[df3 == "Relative: Other"] <- "RelativeOther"
df3[df3 == "I have a Friend of Co-worker with MS"] <- "FriendOfCoWorker"
```

I made sure to store the pre-normalized data into a separate data frame as I will use it to return normalized values back to their original state.

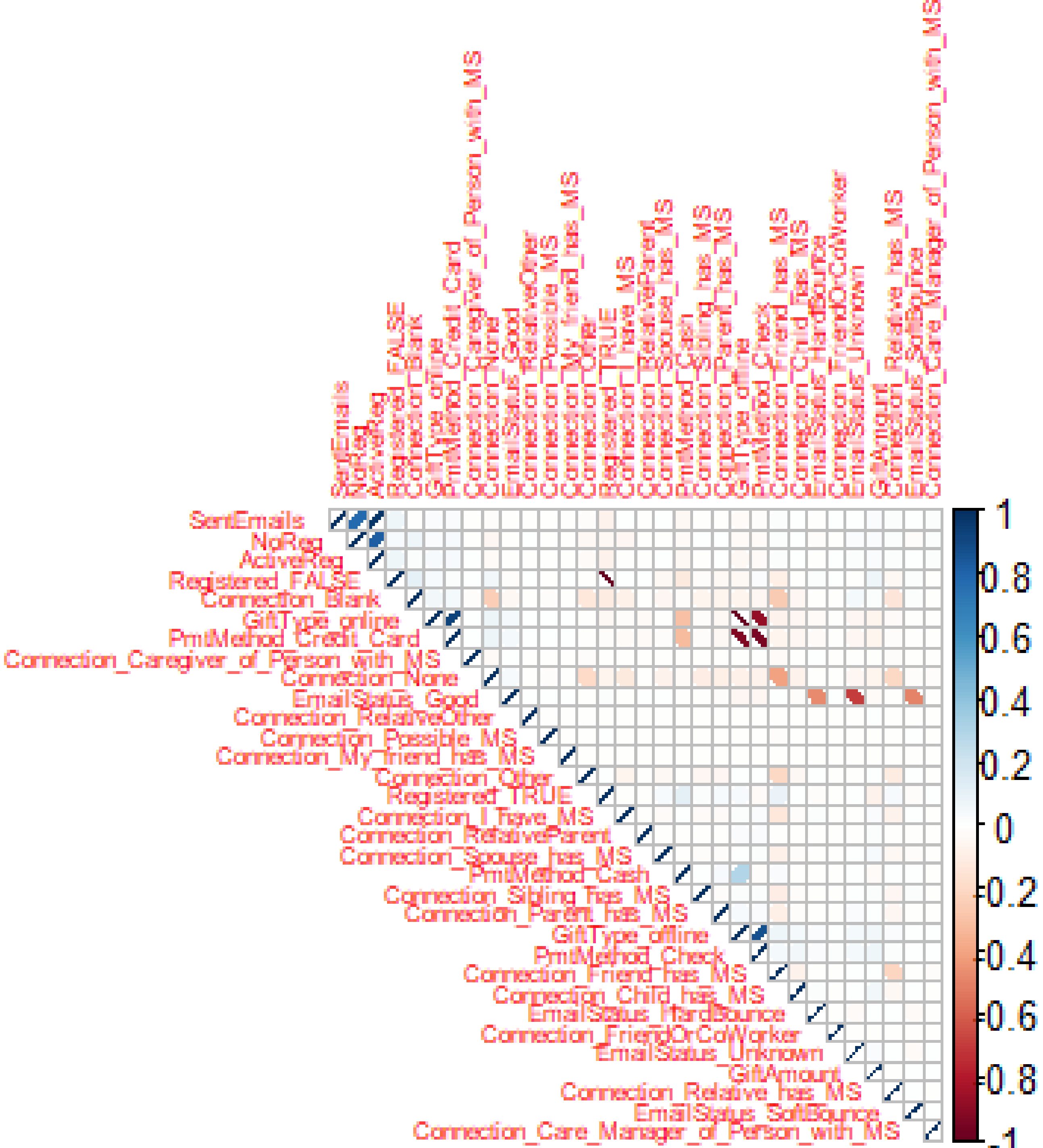
Initially I split the regular non-normalized data into a training set and a testing set. I then take the initial data frame and normalize it, and then split that into a normalized training and testing set.

Correlation of Variables

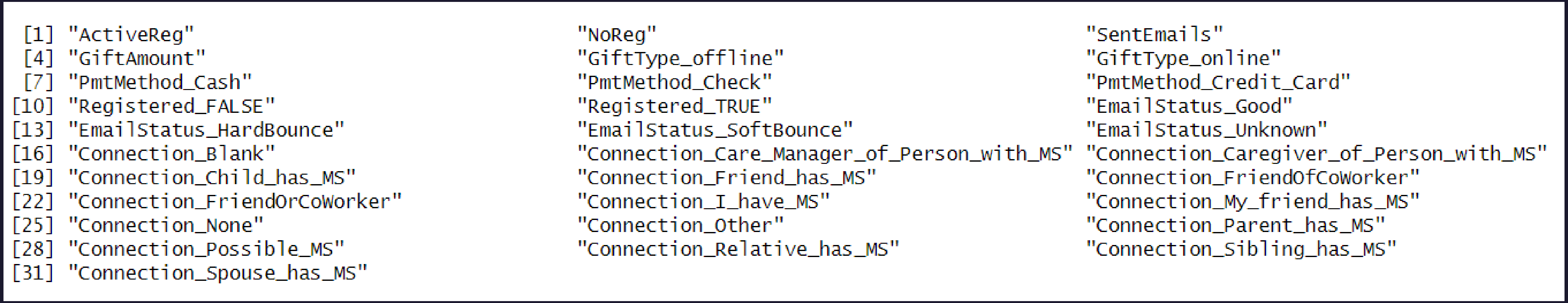
As shown in the correlation matrix, the highest correlated variables are Active Registration and Sent Emails.

Additionally, the variables GiftType_offline and PmtMethod_check are also correlated, which makes sense seeing as a check is an offline payment method.

The variable EmailStatus_Good and EmailStatus_Unknown are negatively correlated.



Run One: Independent Variables



```
 [1] "ActiveReg"                                "NoReg"                                     "SentEmails"
 [4] "GiftAmount"                               "GiftType_offline"                          "GiftType_online"
 [7] "PmtMethod_Cash"                           "PmtMethod_Check"                           "PmtMethod_Credit_Card"
[10] "Registered_FALSE"                         "Registered_TRUE"                           "EmailStatus_Good"
[13] "EmailStatus_HardBounce"                   "EmailStatus_SoftBounce"                    "EmailStatus_Unknown"
[16] "Connection_Blank"                         "Connection_Care_Manager_of_Person_with_MS" "Connection_Caregiver_of_Person_with_MS"
[19] "Connection_Child_has_MS"                  "Connection_Friend_has_MS"                  "Connection_FriendOfCoWorker"
[22] "Connection_FriendOrCoWorker"              "Connection_I_have_MS"                      "Connection_My_friend_has_MS"
[25] "Connection_None"                          "Connection_Other"                          "Connection_Parent_has_MS"
[28] "Connection_Possible_MS"                   "Connection_Relative_has_MS"                "Connection_Sibling_has_MS"
[31] "Connection_Spouse_has_MS"
```

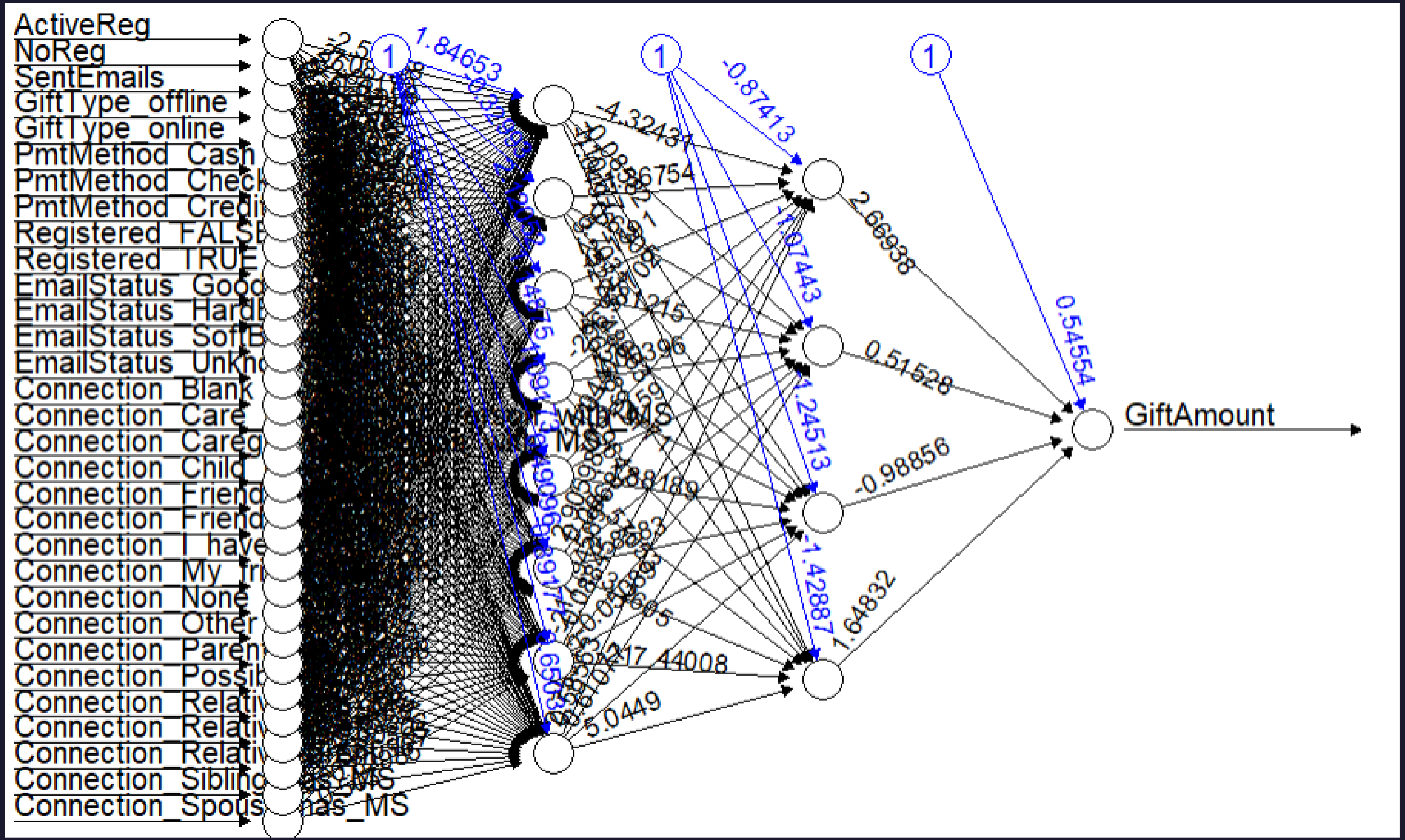
These are the initial independent variables which will be passed as inputs into our two predictive models. The following are numeric variables: ActiveReg (Active Registration), NoReg (No Registration), and SentEmails. The rest of the independent variables are binary categorical variables.

Run One | Predictive Model A1: Neural Network

This neural network is the first predictive model I created in order to predict GiftAmount from the independent variables on the previous slide.

The dimensions on this neural network are two hidden layers with the first layer having eight nodes and the second layer having four.

I chose to model it this way to avoid overfitting had I stuck to a single layer. But I was careful to not add too many layers due to the limitations of the training memory useage.



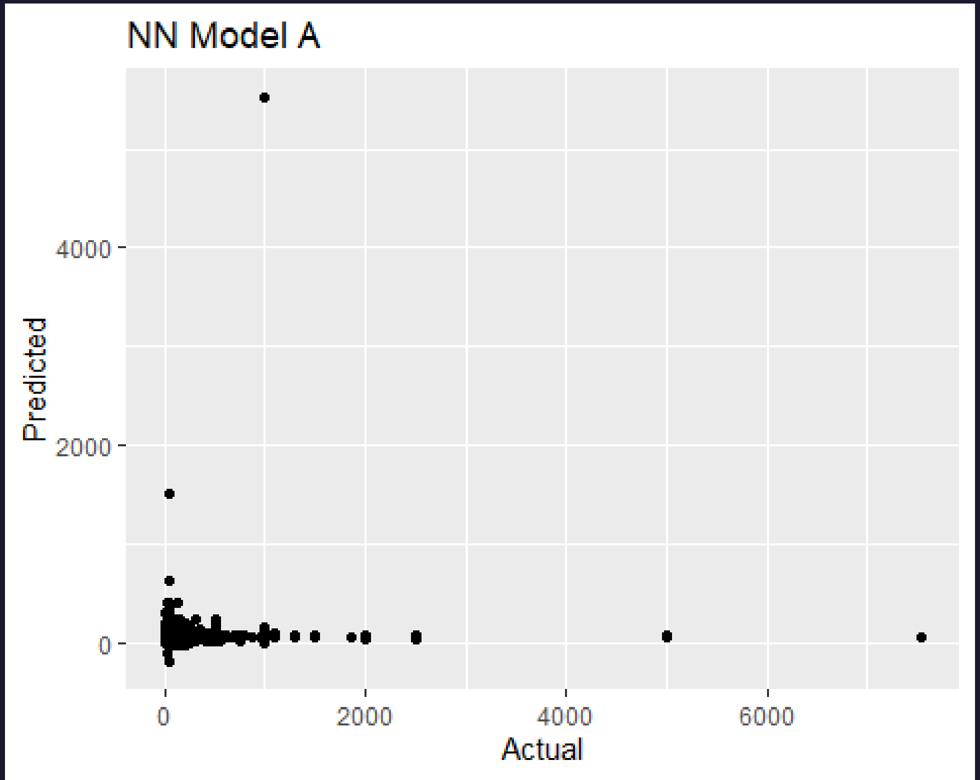
Run One | Predictive Model B1: Regression

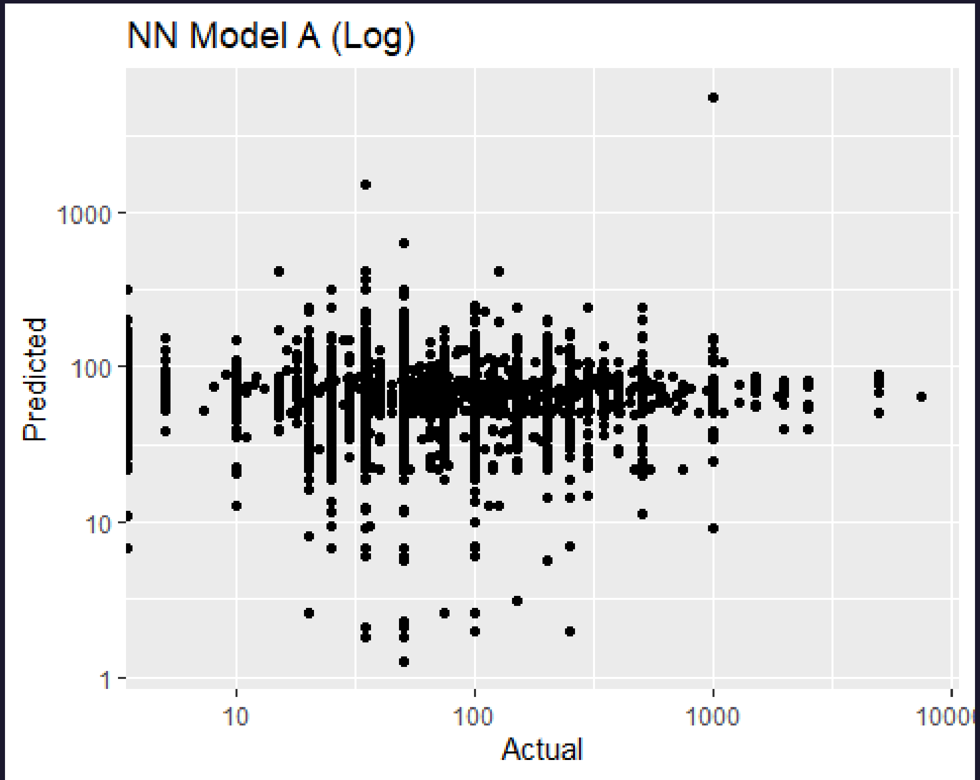
I am using the same inputs inputted into the neural network into the multi-variable regression model. The key difference is I do not have to normalize the numeric data in this model; I can run the original dataset alongside the dummy variables. Additionally, in the next few slides we will see which independent variables we can eliminate from the significance test.

Run One | Neural Network Model A I

Neural Network Model: A
Regular (Un-Logged)

Neural Network Model: A
Log10 (X, Y)

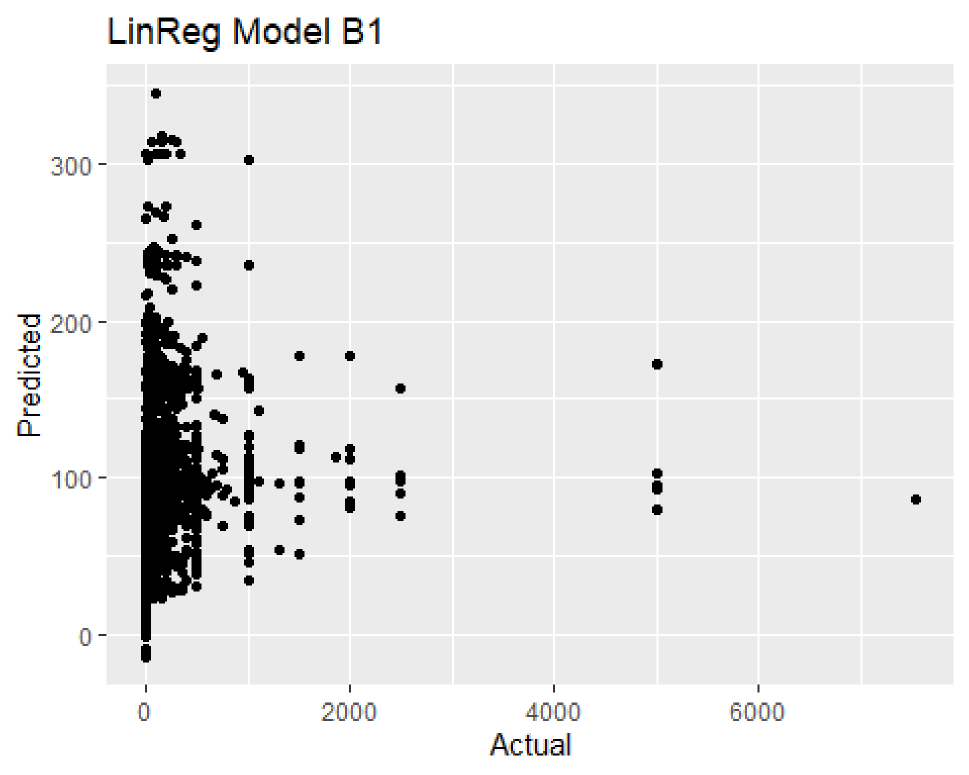




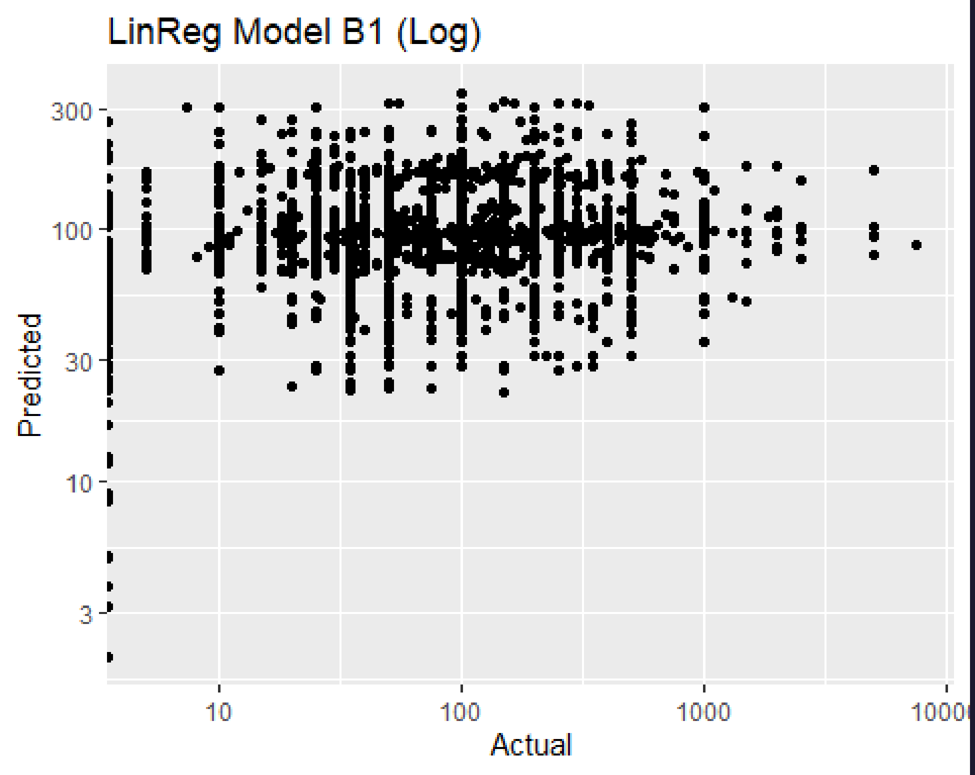
As we can see, there is not much of a linear trend between the actual and predicted as shown in both figures. However the neural network seems to be good at predicting outliers as we can see on the graph.

Run One | Regression Model B1

Multi-Variable Regression Model: A
Regular (Un-Logged)



Multi-Variable Regression Model: A
Log10 (X, Y)



These two charts have even a less of a trend than the first neural network’s charts did . Additionally, this model is not as accurate as the neural network in predicting outliers in the data as the Gift Amount caps off at 300.

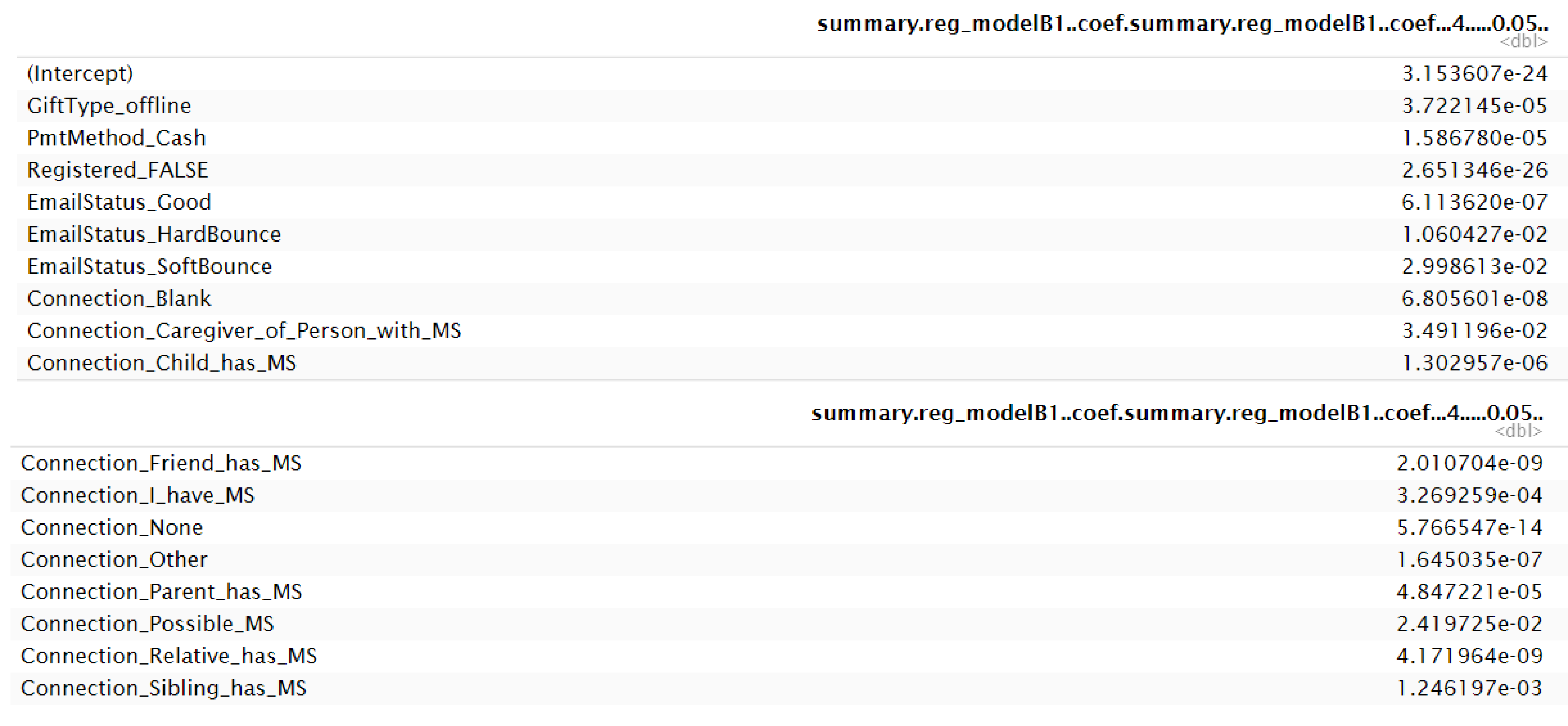
Run One: Model Evaluation Results

The result RMSE for the first Neural Network model is **209.1126**

The result RMSE for the first Multi-Regression model is **200.553**

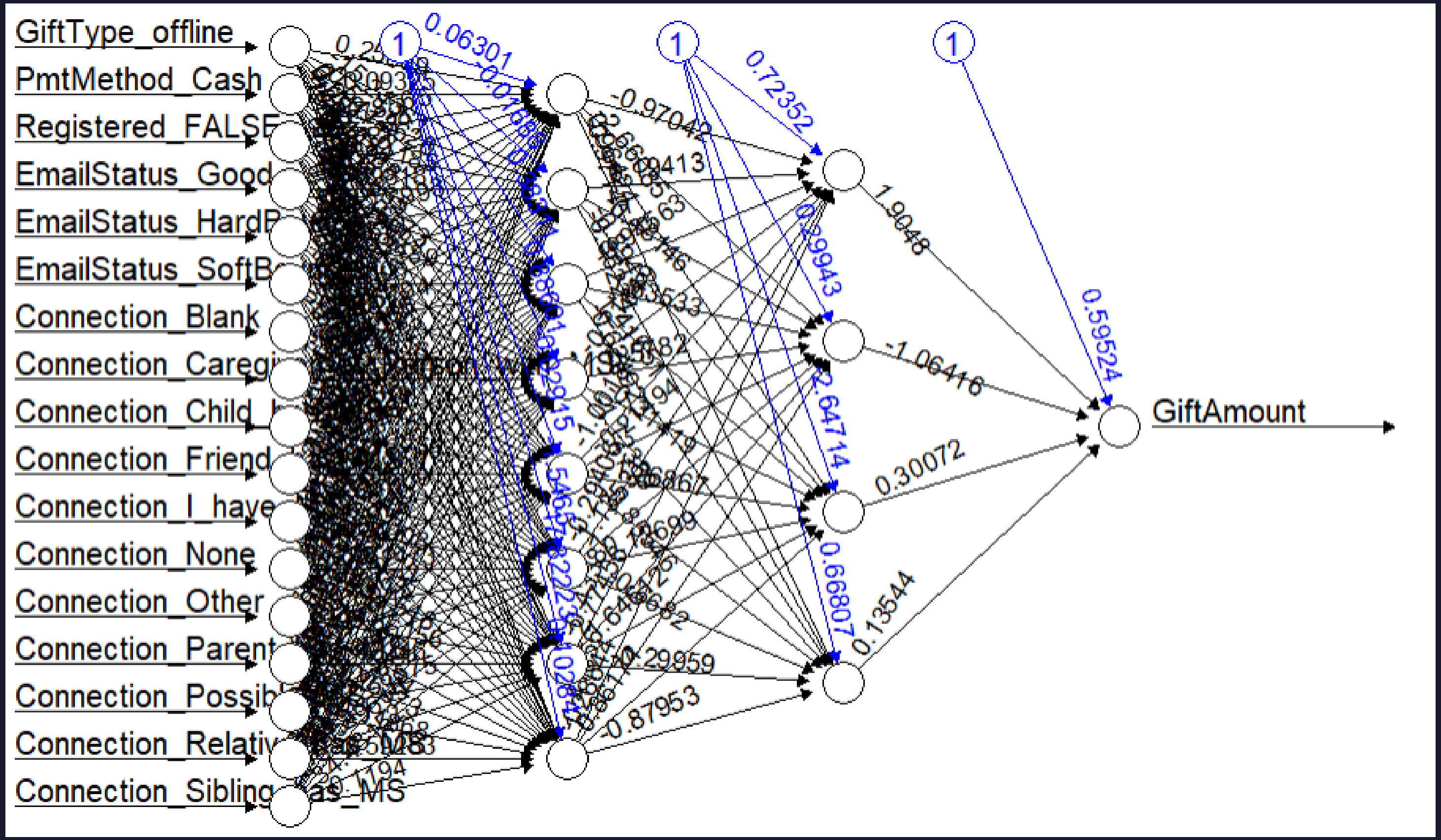
So far, the Multi-Regression has shown it performs better in terms of prediction. However, we can see in the previous slides that the neural network is better at predicting outliers. These numbers are only about 9 apart so it is decently close. We will see which model performs better in the later slides.

Extracted Significant Variables



	summary.reg_modelB1..coef.summary.reg_modelB1..coef...4.....0.05.. <dbl>
(Intercept)	3.153607e-24
GiftType_offline	3.722145e-05
PmtMethod_Cash	1.586780e-05
Registered_FALSE	2.651346e-26
EmailStatus_Good	6.113620e-07
EmailStatus_HardBounce	1.060427e-02
EmailStatus_SoftBounce	2.998613e-02
Connection_Blank	6.805601e-08
Connection_Caregiver_of_Person_with_MS	3.491196e-02
Connection_Child_has_MS	1.302957e-06
Connection_Friend_has_MS	2.010704e-09
Connection_I_have_MS	3.269259e-04
Connection_None	5.766547e-14
Connection_Other	1.645035e-07
Connection_Parent_has_MS	4.847221e-05
Connection_Possible_MS	2.419725e-02
Connection_Relative_has_MS	4.171964e-09
Connection_Sibling_has_MS	1.246197e-03

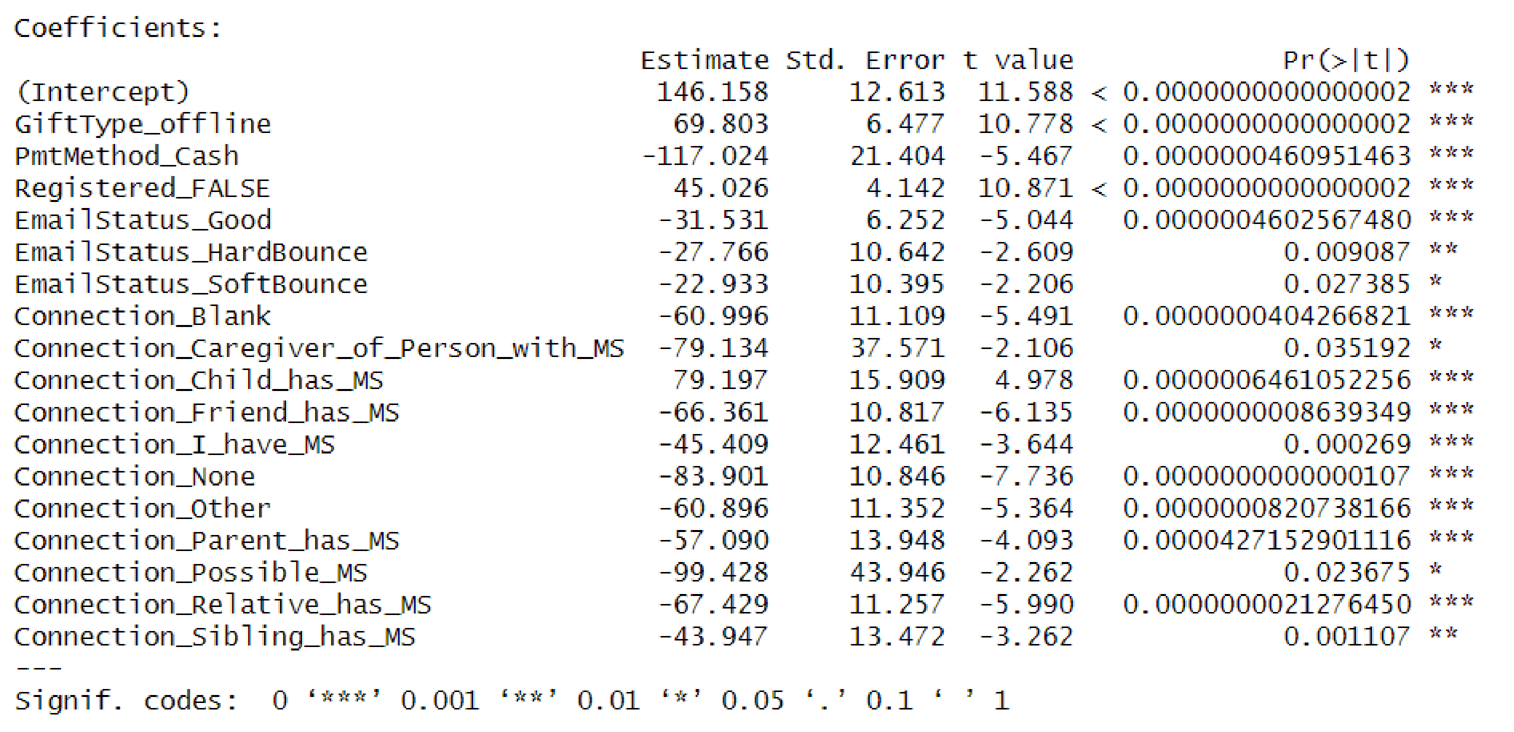
Run Two | Predictive Model A2: Neural Network



In this run, I chose to keep the same neural network structure with the new input variables which were found significant from the regression model. As I showed in the previous run that the RMSE measures had a little space of 9 between them, these new models may have similar RMSE scores as well.

Run Two | Predictive Model B2: Regression

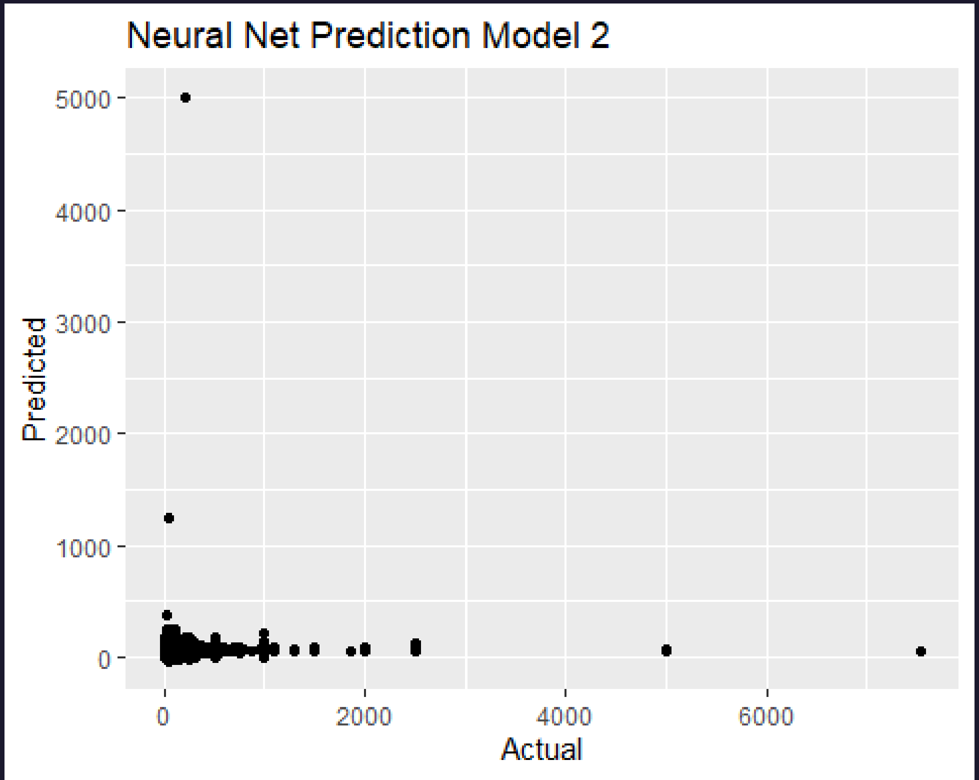
After dropping the variables which were not significant, I inputted all of the significant variables into the Regression. I used the same data frame as I had in the first but dropped all of the insignificant variables using the select() feature. After running the new model, I got the following results. All of the new variables were found to be significant in the second run of the Regression, which means our results will be more valid for both models.



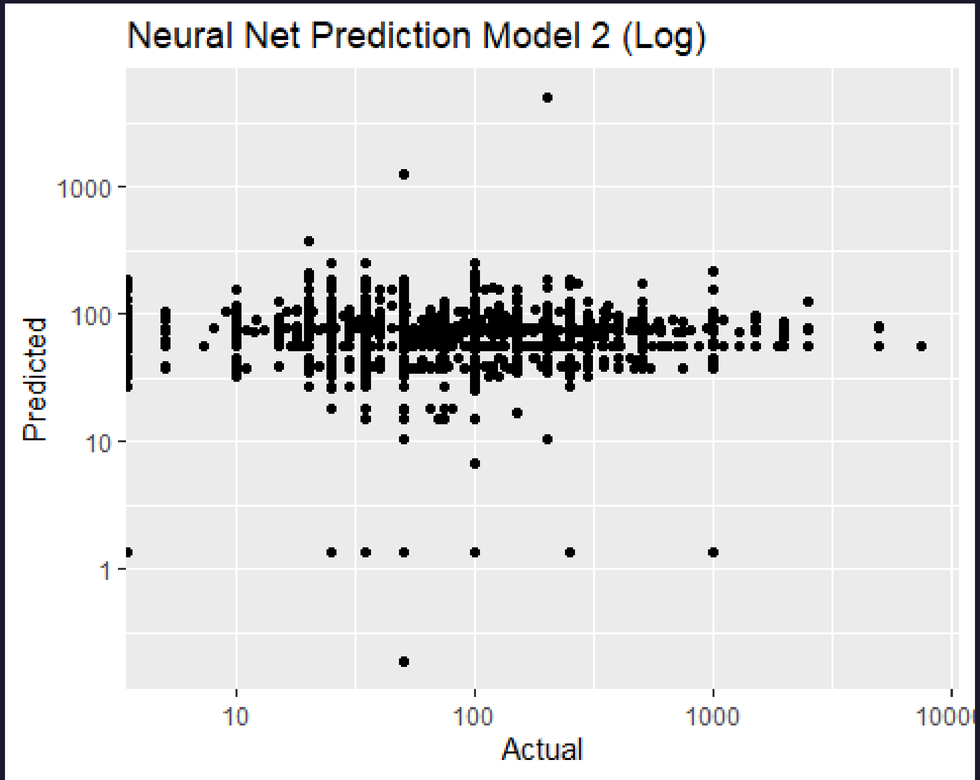
```
Coefficients:
                                       Estimate Std. Error t value             Pr(>|t|)    
(Intercept)                             146.158     12.613  11.588 < 0.0000000000000002 ***
GiftType_offline                         69.803      6.477  10.778 < 0.0000000000000002 ***
PmtMethod_Cash                         -117.024     21.404  -5.467   0.0000000460951463 ***
Registered_FALSE                         45.026      4.142  10.871 < 0.0000000000000002 ***
EmailStatus_Good                        -31.531      6.252  -5.044   0.0000004602567480 ***
EmailStatus_HardBounce                  -27.766     10.642  -2.609             0.009087 ** 
EmailStatus_SoftBounce                  -22.933     10.395  -2.206             0.027385 *  
Connection_Blank                        -60.996     11.109  -5.491   0.0000000404266821 ***
Connection_Caregiver_of_Person_with_MS  -79.134     37.571  -2.106             0.035192 *  
Connection_Child_has_MS                  79.197     15.909   4.978   0.0000006461052256 ***
Connection_Friend_has_MS                -66.361     10.817  -6.135   0.0000000008639349 ***
Connection_I_have_MS                    -45.409     12.461  -3.644             0.000269 ***
Connection_None                         -83.901     10.846  -7.736   0.0000000000000107 ***
Connection_Other                        -60.896     11.352  -5.364   0.0000000820738166 ***
Connection_Parent_has_MS                -57.090     13.948  -4.093   0.0000427152901116 ***
Connection_Possible_MS                  -99.428     43.946  -2.262             0.023675 *  
Connection_Relative_has_MS              -67.429     11.257  -5.990   0.0000000021276450 ***
Connection_Sibling_has_MS               -43.947     13.472  -3.262             0.001107 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Run Two | Neural Network Model A2

Neural Network Model: A2
Regular (Un-Logged)



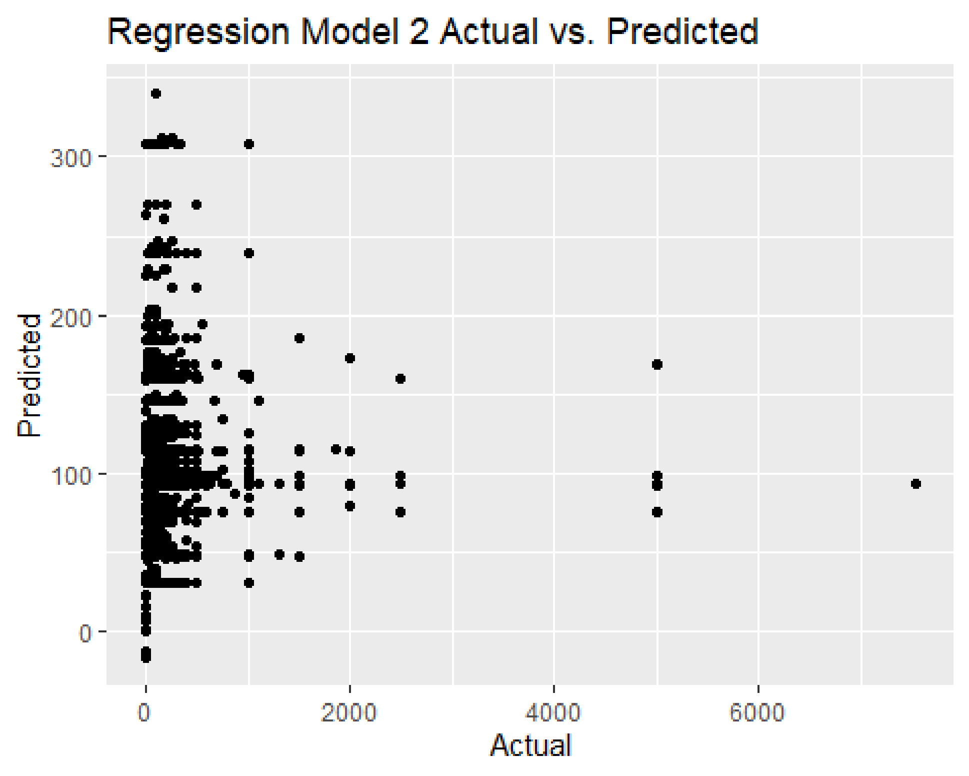
Neural Network Model: A2
Log10 (X, Y)



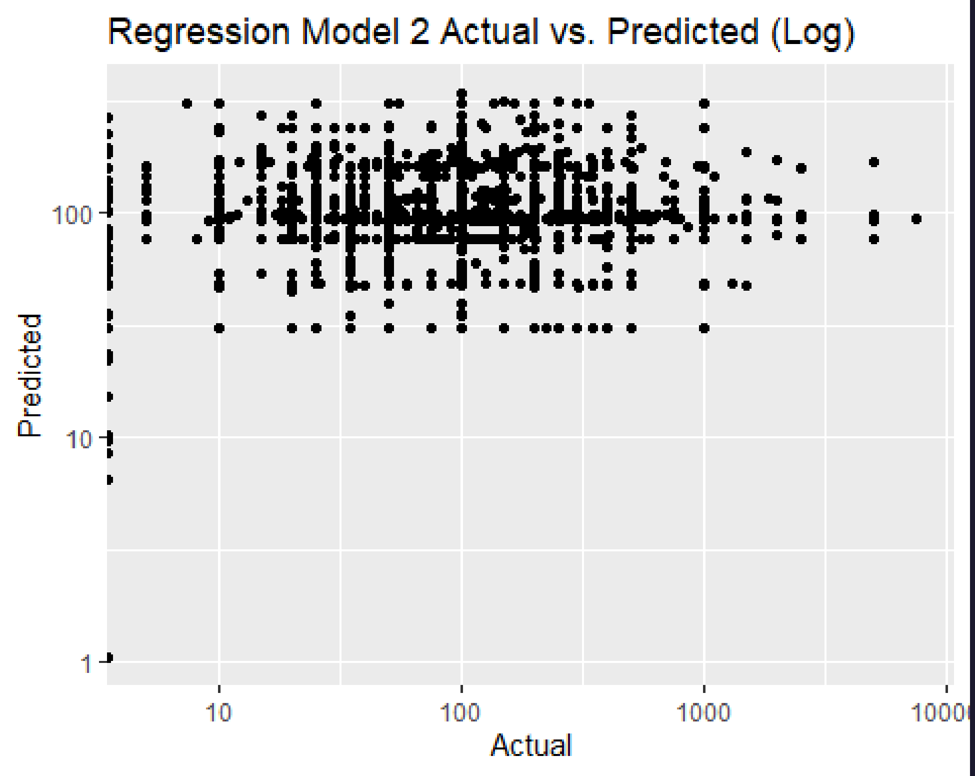
After the second run, the neural network still manages to predict the outlier Gift Amounts. However, there is still no linear trend between the Actual vs. Predicted which means our model still misses out on predicting some of its given inputs.

Run Two | Regression Model B2

Multi-Variable Regression Model: B2
Regular (Un-Logged)



Multi-Variable Regression Model: B2
Log10 (X, Y)



The regression model still is not able to properly predict the outliers, along with not even being able to predict Gift Amounts that are between 300-400 dollars. Additionally, there is no linear trend in the Actual vs. Predicted in the graphs shown above.

Run Two: Model Evaluation Results

The result RMSE for the second Neural Network model is **209.2063**

The result RMSE for the second Multi-Regression model is **200.5829**

In this case, both of the RMSE's increased slightly from Run One. This may be due to how I sampled the data, even though it was causing prediction errors, the RMSE may have performed better. Still this could mean that even though the variables selected for the second model were significant, they may have not been perfect predictor variables. Overall, the Multi-Variable Regression performed better than the Neural Network model in this scenario.

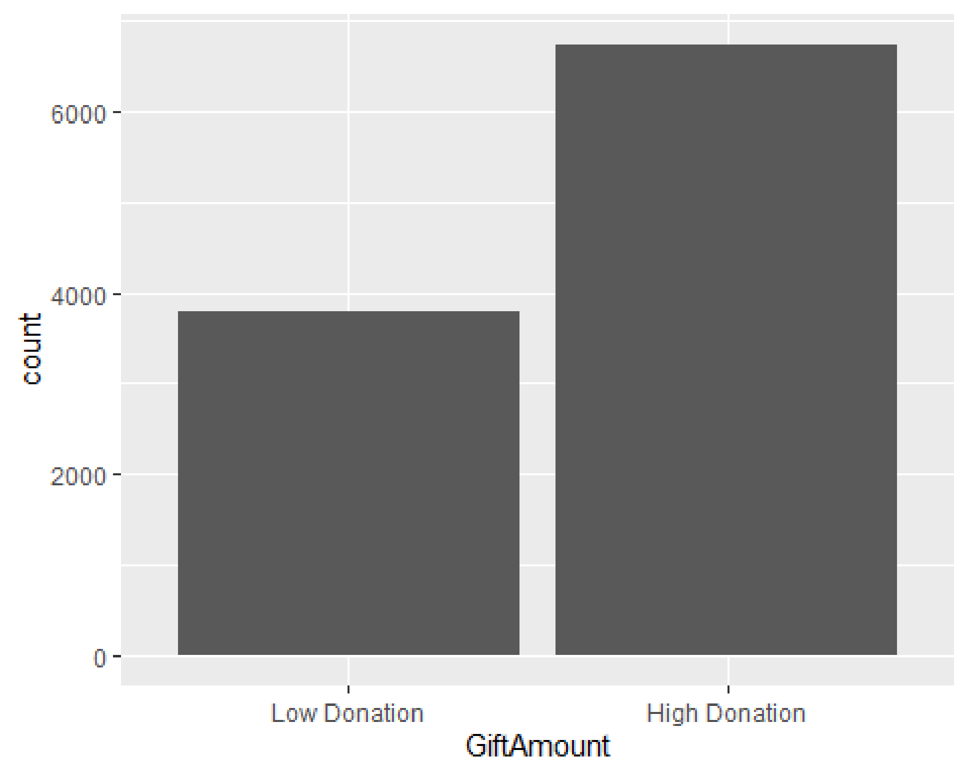
Ensemble Method Summary

In this method, I am using a neural network to predict the Gift Amount numerically. Once the neural network is finished with its predictions, I then classify all of the Gift Amounts into “Low Donations” and “High Donations”. The variables that I am using in this method are

- Active Registrations
- No Registrations
- Sent Emails
- Email Status (Good)
- Connection Friend has MS

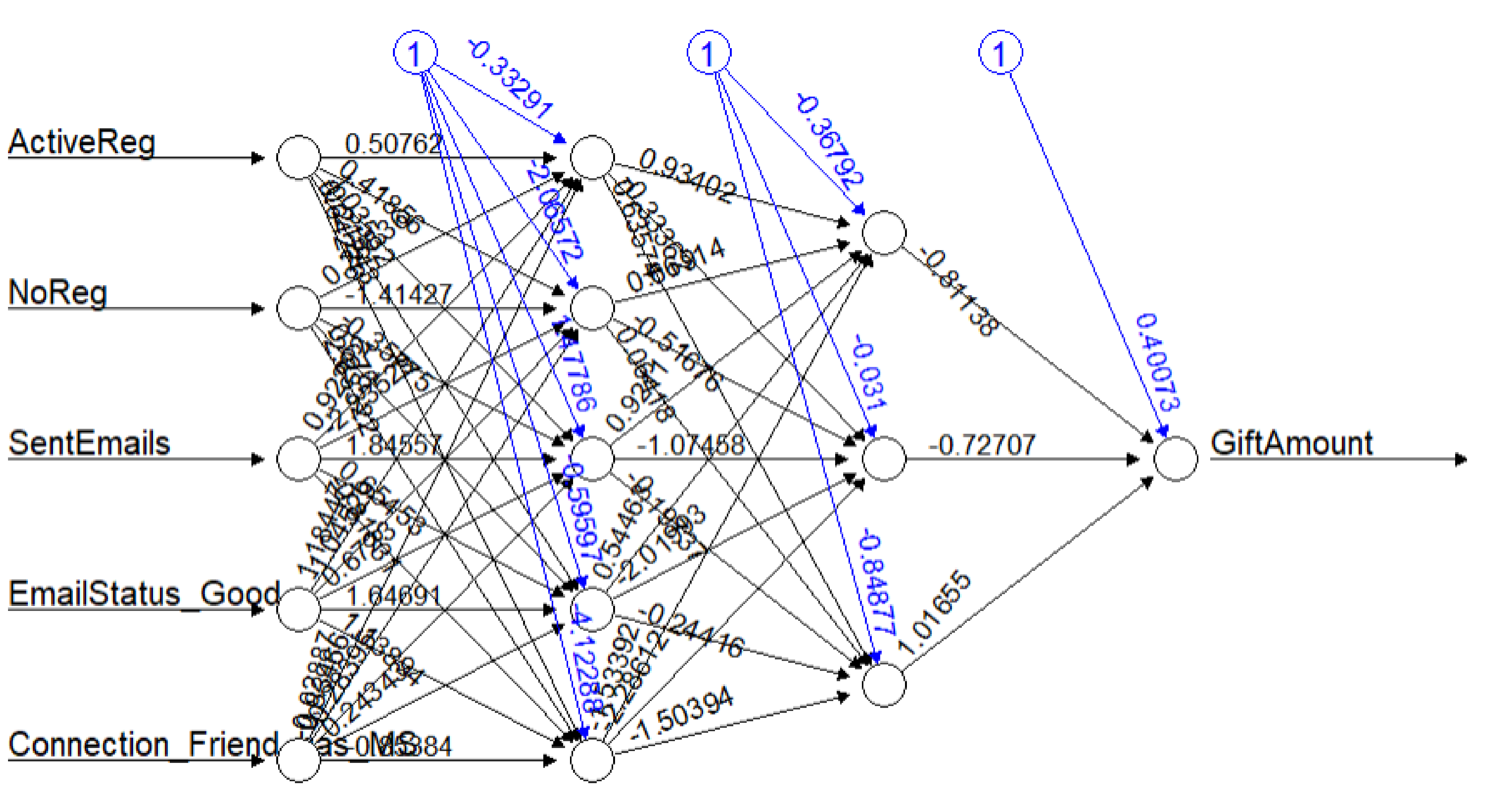
I chose these variables at random based on my personal opinion about the greatest predictor and classifier of the Gift Amount variables.

Ensemble Method | Gift Class Histogram

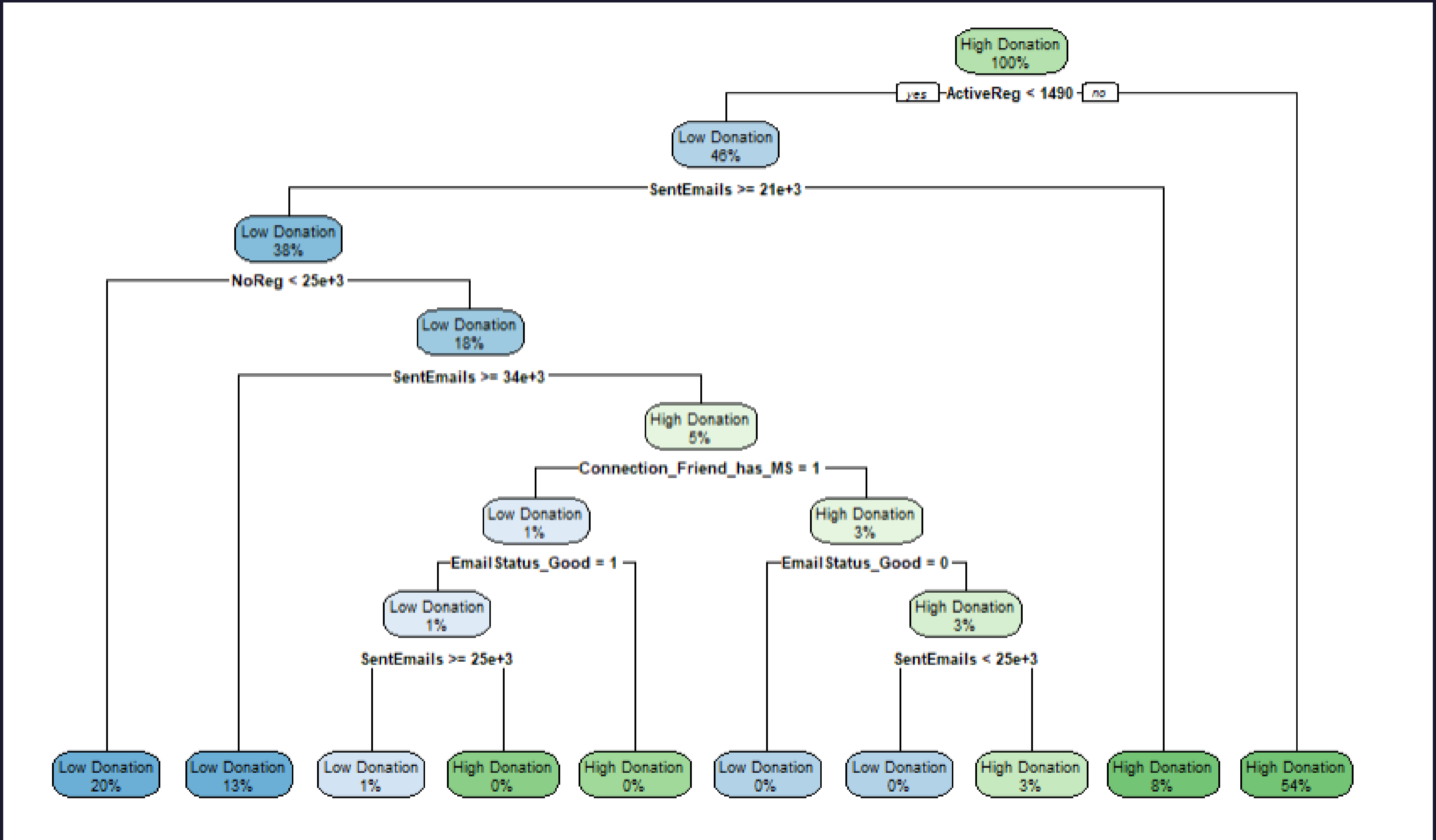


I chose to visualize the low and high donations that will be inputted into the decision tree using the cutoffs (0, 90) for the low donation and (90,150) for the high donation. The amount of Low Donations was 3789 and the amount of High Donations was 6741

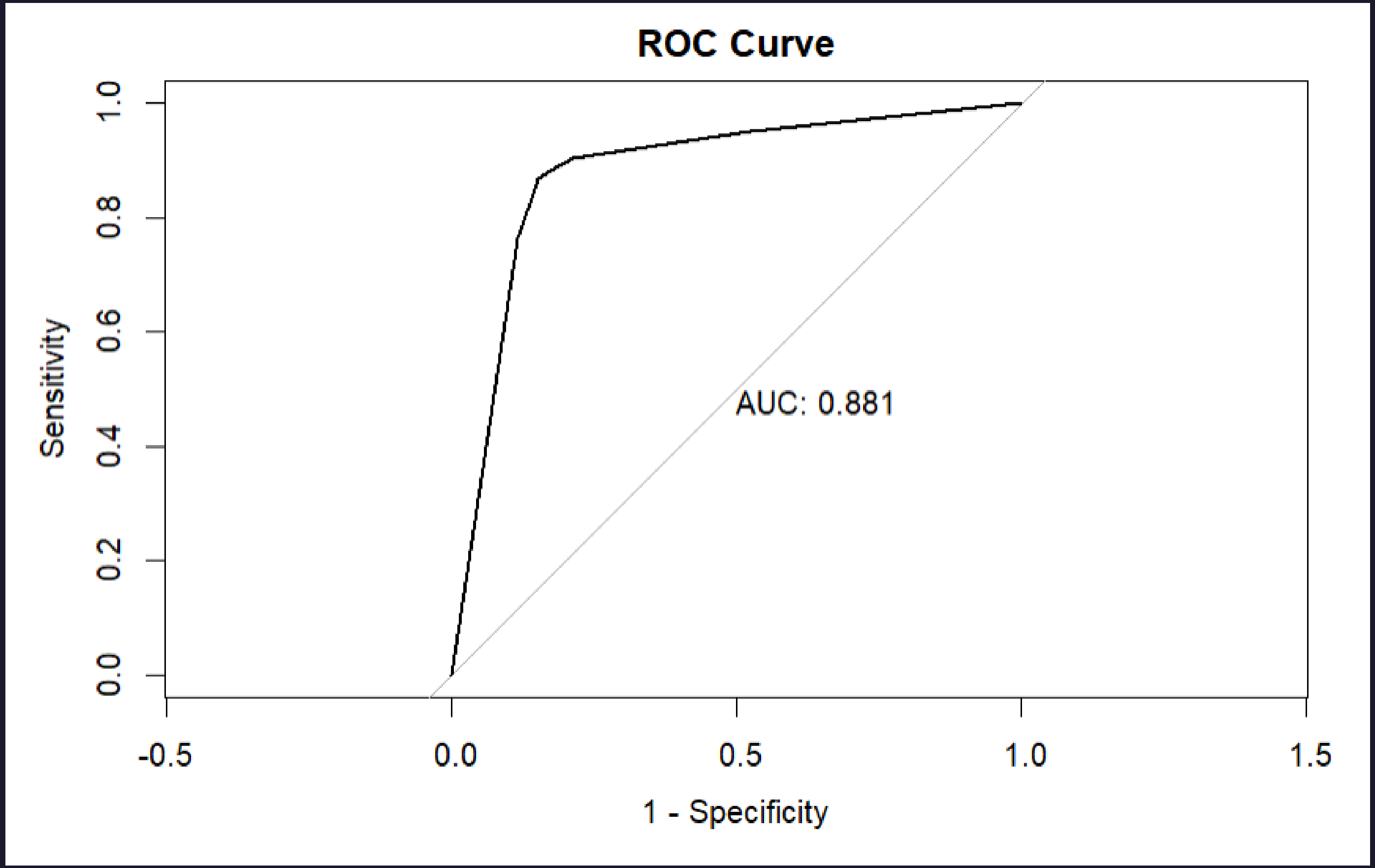
Ensemble Method | Pipeline | Neural Network

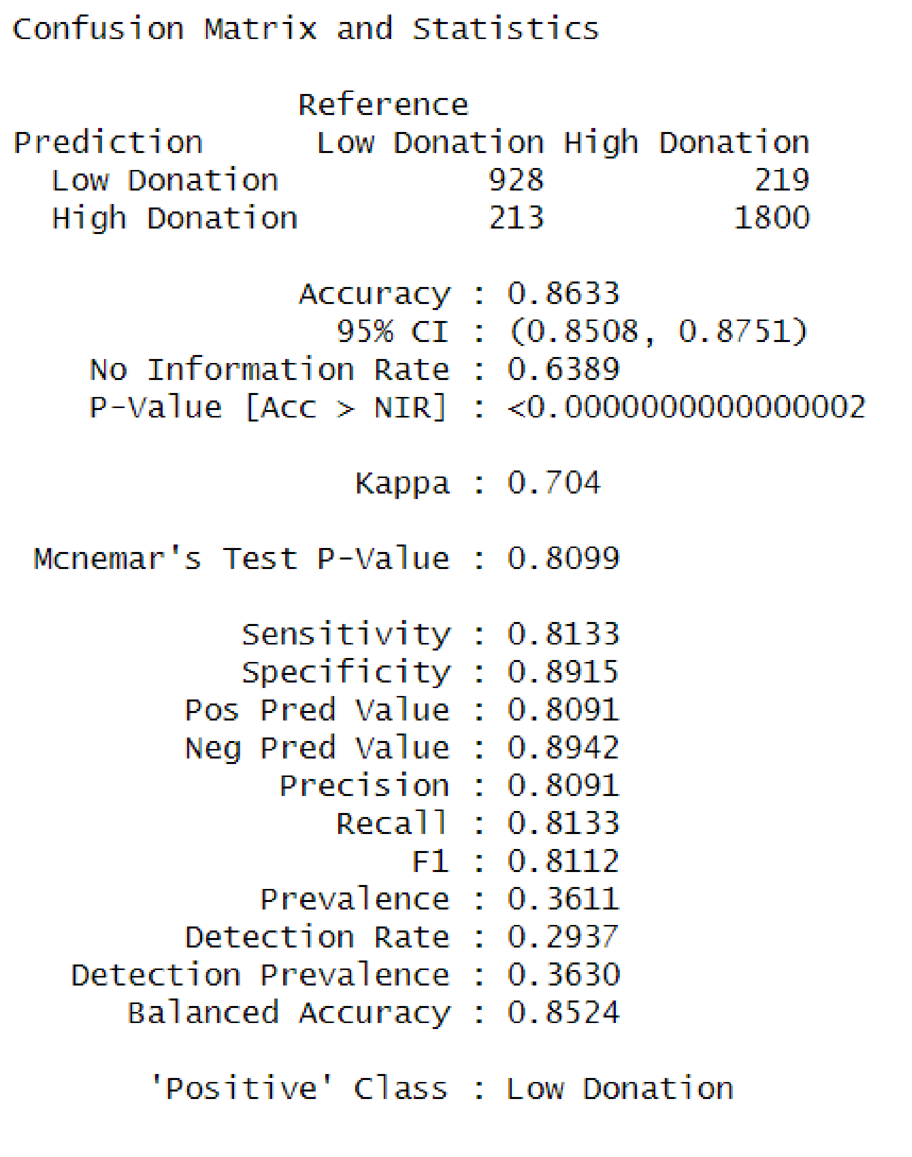


Ensemble Method | Pipeline | Decision Tree



Ensemble Results





```
Confusion Matrix and Statistics

               Reference
Prediction      Low Donation High Donation
  Low Donation           928           219
  High Donation          213          1800

               Accuracy : 0.8633
                 95% CI : (0.8508, 0.8751)
    No Information Rate : 0.6389
    P-Value [Acc > NIR] : <0.0000000000000002

                  Kappa : 0.704

 Mcnemar's Test P-Value : 0.8099

            Sensitivity : 0.8133
            Specificity : 0.8915
         Pos Pred Value : 0.8091
         Neg Pred Value : 0.8942
              Precision : 0.8091
                 Recall : 0.8133
                     F1 : 0.8112
             Prevalence : 0.3611
         Detection Rate : 0.2937
   Detection Prevalence : 0.3630
      Balanced Accuracy : 0.8524

       'Positive' Class : Low Donation
```

Ensemble Interpretations

The confusion matrix for the decision tree can be interpreted as follows (Low Donations = TP & High Donations = TN):

- Precision: Out of all of the variables classified as a Low Donation, only 80.91% are actual Low Donations

- Sensitivity: Out of all of the variables that are Low Donations, 81.33% are predicted as Low Donations

- Specificity: Out of all of the variables that are High Donations, 89.75% are correctly classified as High Donations

Overall, the decision tree was a very useful model in this report because of its AUC = 0.881, which is a very high number of correct classifications.

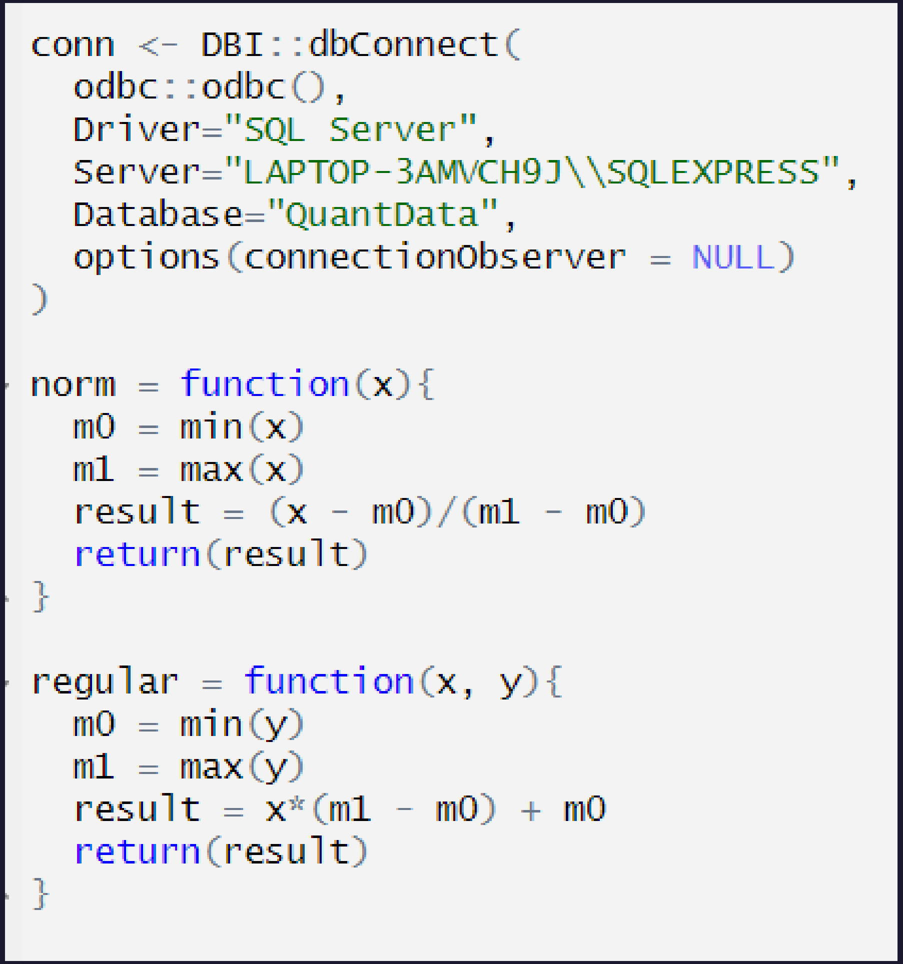
Model Summary & Conclusion

Overall, none of the models used in this project were fully effective. Each have their pros and cons. The neural network was able to predict outliers, The regression was useful for extracting significant variables, and the ensemble model was great for classifying predicted values.

To answer question one at the beginning slide, based on the significance test, the variable which is greatest at predicting the Gift Amount is Connection_None, the donors who had no connection to MS.

To answer question two at the beginning slide, it is simply that Neural Nets are good for outliers, Regression is good for filtering insignificant variables, and the Ensemble model is good for classifying categories. The model I recommend to use in order to answer this question is the Neural Network Model.

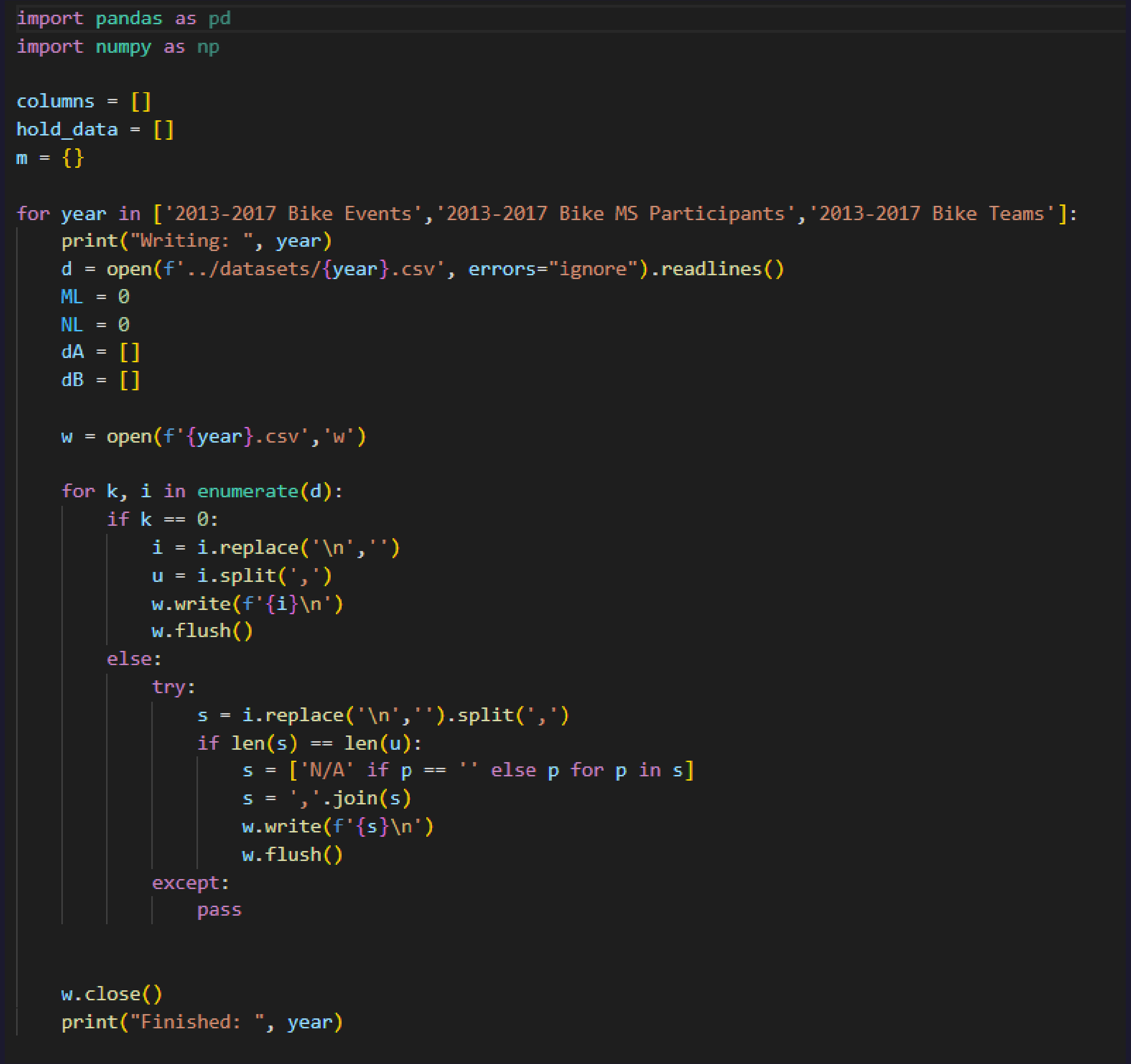
Appendix



```r
conn <- DBI::dbConnect(
  odbc::odbc(),
  Driver="SQL Server",
  Server="LAPTOP-3AMVCH9J\\SQLEXPRESS",
  Database="QuantData",
  options(connectionObserver = NULL)
)

norm = function(x){
  m0 = min(x)
  m1 = max(x)
  result = (x - m0)/(m1 - m0)
  return(result)
}

regular = function(x, y){
  m0 = min(y)
  m1 = max(y)
  result = x*(m1 - m0) + m0
  return(result)
}
```



```python
import pandas as pd
import numpy as np

columns = []
hold_data = []
m = {}

for year in ['2013-2017 Bike Events','2013-2017 Bike MS Participants','2013-2017 Bike Teams']:
    print("Writing: ", year)
    d = open(f'../datasets/{year}.csv', errors="ignore").readlines()
    ML = 0
    NL = 0
    dA = []
    dB = []

    w = open(f'{year}.csv','w')

    for k, i in enumerate(d):
        if k == 0:
            i = i.replace('\n','')
            u = i.split(',')
            w.write(f'{i}\n')
            w.flush()
        else:
            try:
                s = i.replace('\n','').split(',')
                if len(s) == len(u):
                    s = ['N/A' if p == '' else p for p in s]
                    s = ','.join(s)
                    w.write(f'{s}\n')
                    w.flush()
            except:
                pass


    w.close()
    print("Finished: ", year)
```

Suggestions

I would suggest paying attention to how you merge your datasets and how many variables you select for your predictive models. My presentation had a data error with the numeric variables and my regression was not able to predict outliers.

Additionally, if you have not used SQL in your project I would highly recommend you get familiar with it as it is very popular in data science.