MISSING VALUE ANALYSIS

Mohammad Anas

12/3/2021

Introduction

Missing values are popular problem among research of all kinds specifically in industries like finance that contain sensitive information about people. It not only leads to loss of information and reduced statistical power but can also lead to researchers introducing selection bias, which in many cases ivalidates the study (Frisell, 1). In this project, we will look at the two major methods of imputing missing values; single imputation and multiple imputation. To compare the two approaches, we took a complete data set and created missing values within the data. These missing values are than imputed with fore mentioned methods. A classification model is created, fitted onto the imputed and the actual data sets and the resulting standard estimates are then compared. Model building process was carried out on one of imputed data sets rather than the complete data as I wanted to replicate a real life scenario.

Data and Missing Value Amputation

The data chosen for this analysis was customer churn data of a banking product. Given that this was a fictional data set there is not much background information available in this data. The data set contained information on customers' geography, demographics, bank account details and some personal information like salary. The codebook for the dataset has been provided in the appendix.

The missing values were created within the age and the salary variable as these variables are regarded as personal information and people are less likely to disclose this information. To ensure the missing at random (MAR) property is intact, we created missing values based on information from other variables. We used logistic regression to construct a binary variable indicating data entries in which we create missing values for the variables age and salary. We assume that females are more likely to hide their age as compared to males. Moreover, Germans and French people are less likely to not reveal their age. We also added an interaction between age and gender variables in our model to create the binary indicator of missingness for the age variable.

$$Logit(\frac{Pr(Age_{missing}=1)}{1-Pr(Age_{missing}=1)}) = -3.8 + 2.8Fe_i - 0.02Ger_i - 0.01Fren_i + 1.1Fe_i * Ger_i + 1.9Fe_i * Fren_i$$

To generate missing indicator for salary variable, we assumed that customers who are not an active member of the bank are less likely to report their salary. We also assumed that people who have had a bank account for more years in the bank are more likely to trust the bank with their salary. The assumptions made are totally arbitrary. The equation of the model used to generate missing indicator for salary variable is as follows.

$$Logit(\frac{Pr(Age_{missing}=1)}{1-Pr(Age_{missing}=1)}) = 4.4 - 1.3Tenure_i - 1.2ActiveMember_i$$

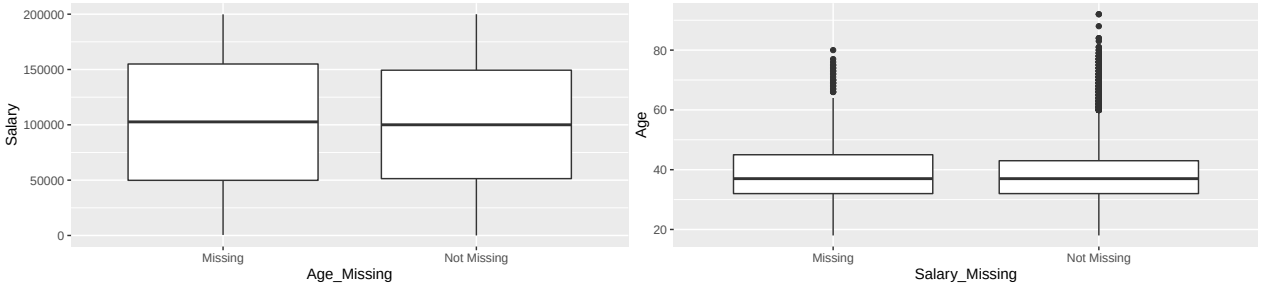
In total 4,868 missing values were introduced into the data, out of which 2,423 were created within the salary column and 2,445 in the age column. 1,314 data entries had missing values for both age and salary.

Single Imputation

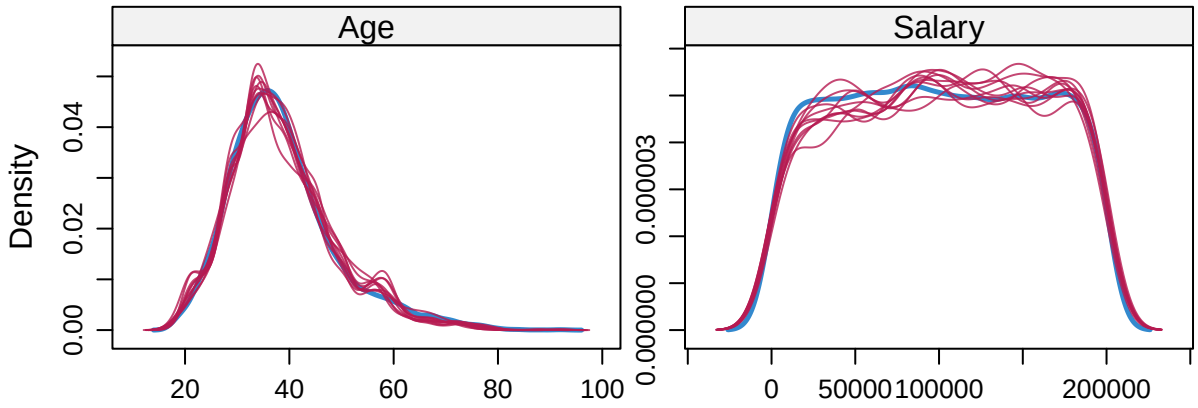
We chose nearest neighbor imputation as the single imputation method. For each missing value, we look at the two most similar complete data entries based on the other variables provided in the data. Given that our variables which have missing data are continuous, we replace the missing value with the average value of the 2 identified nearest neighbors for that particular column. Euclidean distance was used as a metric to determine similarity between data entries. Given that over fitting is preferred when dealing with missing values 2 was chosen as the hyper parameter (K).

Multiple Imputation

For multiple imputation method, we built a series of 10 sequential regression models to predict the missing values of age and salary. We do this separately for age and salary variable. The below plot shows that the variables age and salary have similar distributions irrespective of the fact whether the other variable is missing or not. Hence we decided to not include salary and age as a predictor variable for each other while doing imputation.



We chose 'Predictive Mean Matching' here as it resulted in similar distribution for the imputed data when compared to the observed data for our age and salary columns.



We fit a chosen model to all these data sets and than pool the estimates from each data together by taking the average. We also calculated variance of our estimates by taking a weighted average of the average variance of estimates from all imputed data sets and variance of the estimate across all the data sets. The formula for these calculations is given in the appendix.

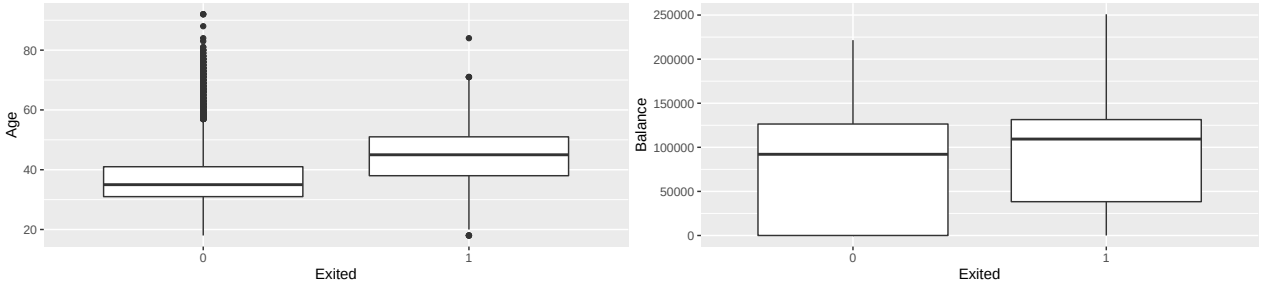
Model Building

We randomly choose one of the data sets that we got from multiple imputation and perform model building process on it, including the EDA, model selection and model assessment. Once we have our model,we will

ensure that our model's assumptions are not violated when fitted on another randomly chosen imputed data set. Given that our response variable is binary one, we will perform logistic regression.

Exploratory Data Analysis

We start our EDA by testing the independence of our response variable with the categorical variables in our data. We see that apart from 'HasCrCard' (issued credit card by the bank), the p-values for all the catgorical variables was extremely low, indicating that the variables were not independent. To explore the relationship between our response variable and continuous variables, we used box plots. Out of these age and bank balance were found to be the most interesting ones.



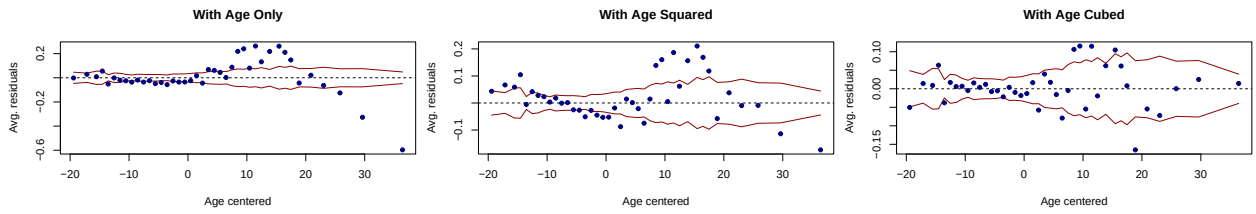
We also tested the interactions that made intuitive sense like credit score and credit card, age and gender, age and geography and age and balance. The interaction between Geography and Salary and Geography and Balance were found interesting. The boxplots for interactions have been added to the appendix.

Model Selection

During model selection, we use AIC as a criteria and to ensure that we do not miss out on any significant variable, we test all of them regardless they were found interesting in the EDA using step wise selection. The variables age, IsActiveMember, Geography, Gender, Balance and Credit Score were kept by AIC. The Salary variable was excluded by the AIC and was also not found interesting in the EDA. Despite this, we decided to keep it in our model as we have imputed missing values for it and hence,it is a variable of interest. All the interactions were removed by AIC.

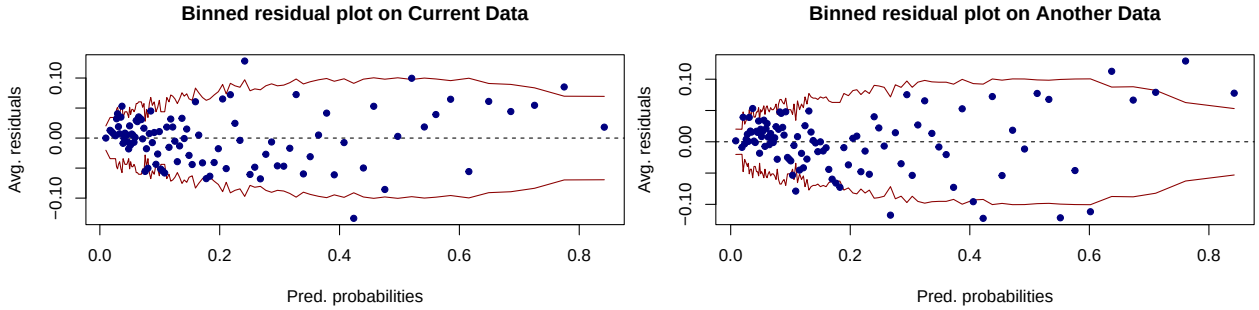
Model Assessment

Once we have our features finalized, we fit our model onto one of the imputed data sets. To see if we have captured the trend of our continuous variables in our model we make binned plots of our residuals against these variables.However, we see that there is a quadratic relationship between our response and age variable that has not been captured. We decided to include quadratic terms of age in our model. When we add a squared term of age in our model, we note that there is still a quadratic relationship in age that has not been captured as the residuals vs age plot now looks something like a sinusoidal graph. After including the cubic term for age we do see a significant improvement and the binned plot now looked random. The observed binned plots of age against our residuals can be seen below.



Now that we have ensured that the trends within continuous variables have been captured we check our model for the independence assumption. Below shown are the plots of predicated probabilities against

binned residuals. On the left side, we see binned residual plot when our model is fitted onto the data we have been using uptil now. The right binned residual plot is the one we observe when the same model was fitted onto another imputed data set. We do see randomness in the plots and it seems that we are doing fine in regards to the independence assumption.



The VIF of all the variables in our model was low and hence, multicollinearity was not as issue.The VIF table has been included in the appendix. We also check the binned plot of residuals against the continuous variables when the model is fitted onto the new data set and randomness was observed in those plots as well. The binned residual plot for age variable has been added to the appendix. Our final model equation is presented below.

$$Logit(\frac{Pr(Exit_i = 1)}{1 - Pr(Exit_i = 1)}) = \beta_0 + \beta_1 Age_i + \beta_2 Age_i^2 + \beta_3 Age_i^3 + \beta_4 IsActiveMember_i + \beta_5 Geography_i + \beta_6 Salary_i + \beta_7 Gender_i + \beta_8 Balance_i + \beta_9 CreditScore_i$$

Sensitivity Analysis

Now that we have our model prepared, we fit this model onto the actual complete data set and the data set that we got from Single Imputation. Lastly, we fit the same model to 10 imputed data sets that we got from multiple imputation method and pool the results together. The results of the models are shown below.

Table 1: Comparison of Estimates

Variables	Actual_Estimates	Pooled_Estimates	SI_Estimates
Intercept	-1.12e+00	-1.16e+00	-1.09e+00
Age	1.49e-01	1.55e-01	1.66e-01
Age_squared	1.92e-03	2.04e-03	1.34e-03
Age_cube	-2.15e-04	-2.27e-04	-2.27e-04
IsActiveMember	-9.95e-01	-9.73e-01	-1.02e+00
German	7.53e-01	7.84e-01	7.53e-01
Spanish	5.82e-03	2.25e-02	9.04e-03
Male	-5.32e-01	-5.51e-01	-5.55e-01
Salary	4.41e-07	7.09e-07	1.20e-06
Balance	2.76e-06	2.77e-06	2.56e-06
Credit Score	-5.17e-04	-7.06e-04	-6.41e-04

We see that the for all the variables except Geography and Credit Score, the pooled estimates that we got from fitting the model on multiple imputed data sets were closer to the actual estimates as compared to the estimates we got from fitting the same model on KNN imputed data set. These differences were not restricted to the value of the standard estimates but also seen in standard errors of these estimates. The variable Salary was declared insignificant by the actual and the pooled model but was found to significant when the model

was fitted onto the KNN imputed data set (assuming a 10% significance level). The significance of all the other variables was consistent across all three models. The p- values we got from all models can be seen below.

Table 2: Comparison of P-values

Variables	P_values_Actual	P_values_MI	P_values_SI
Intercept	6.54e-80	0.00e+00	1.37e-76
Age	1.14e-223	0.00e+00	3.35e-236
Age_squared	5.33e-09	1.36e-06	1.04e-03
Age_cube	1.12e-47	0.00e+00	7.29e-35
IsActiveMember	2.00e-63	0.00e+00	2.71e-65
German	4.74e-27	0.00e+00	1.51e-26
Spanish	9.36e-01	7.65e-01	9.02e-01
Male	3.87e-21	0.00e+00	2.47e-22
Salary	3.68e-01	2.08e-01	2.44e-02
Balance	6.59e-08	2.36e-07	6.33e-07
Credit Score	7.53e-02	1.84e-02	2.85e-02

Model Validation

We now move on to assessing how things look like if our main aim here was prediction rather than understanding associations between variables. To do this, we fit our models to the actual data set and compare the predicted value of each model against the true values. Although there is not much difference observed in the evaluation metrics, we do note that we are classifying more true negatives by using the model we fitted onto the KNN imputed data. The ROC curves that we get from all the three models are given in the appendix. The comparison of the evaluation metrics is seen below.

Table 3: Evaluation Metrics Comparison

Evaluation_metric	Actual_Model	Pooled_Model	KNN_Imputed_Model
Accuracy	0.734	0.731	0.75
Sensitivity	0.716	0.718	0.717
Specificity	0.739	0.733	0.759
AUC	0.797	0.797	0.796

Conclusion and Limitations

Multiple Imputation methods clearly outperform the single imputation method that is most commonly used in the practical world. If our aim is understanding association between variables, we see estimates that we got from multiple imputation are way more closer to the ones that we get when the model is fitted onto the real data set. Moreover, the standard errors are also close to the actual standard errors. Therefore, multiple imputation is more reliable when understanding whether a particular variable has a significant impact on the response variable. When predicting values, we see that there was much difference in the evaluation metrics. However, multiple imputation performs better in that scenario as well. Before we end, I would like to state that there is one limitation to this analysis. Whenever we do multiple imputation we assume that the values are missing at random for each variable, that means that each column's missingness is dependent on the value of some other column/s. Imputation only work if this assumption is satisfied, which is not always the case. Unfortunately, we do not have any way to check of this is assumption is violated or not.

Apendix

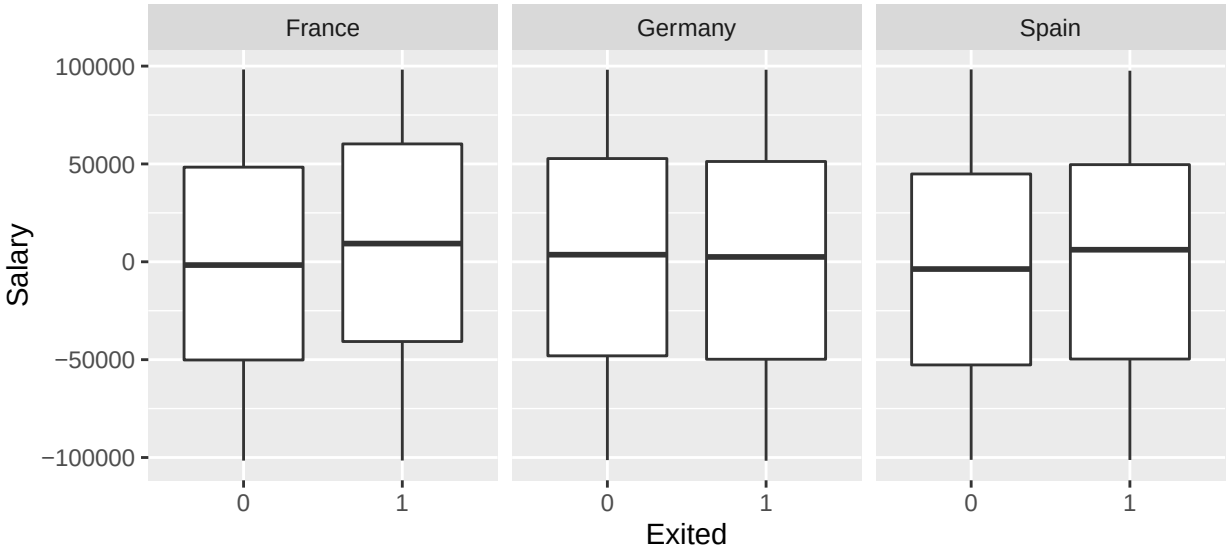
The data set used in this analysis can be found on my github profile. Click here for that.

The code book for the data set used is provided below.

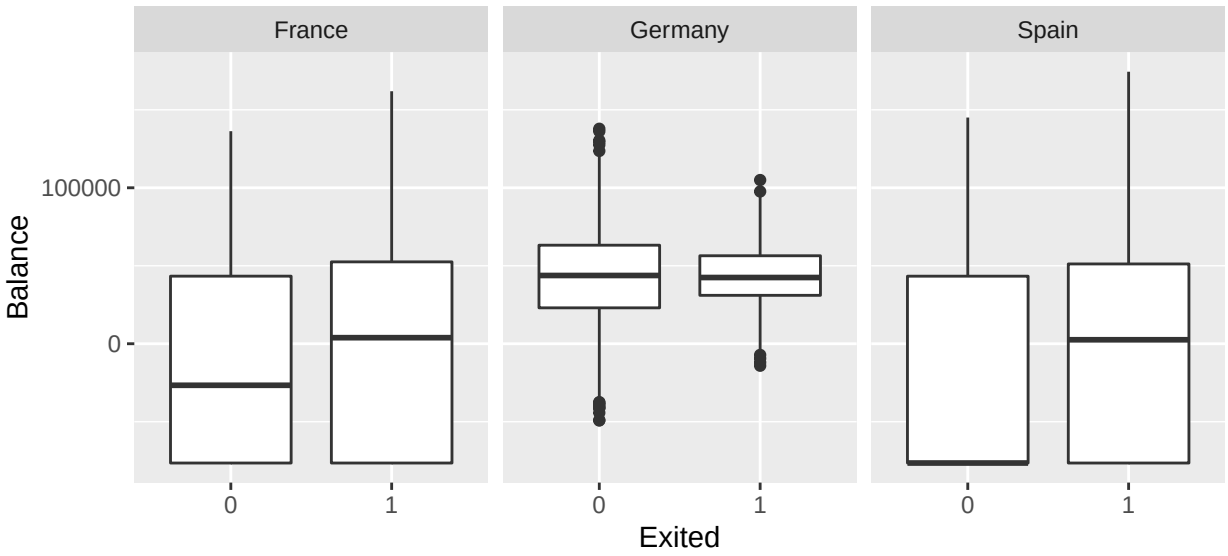
Table 4: Codebook for the Dataset used

Variable	Data Type	Description
Customer Id	Numeric	Unique Identifier
Surname	String	Last Name
Credit Score	Numeric	Credit Score
Geography	Factor	Country - Spain, France, Germany
Gender	Factor	Male or Female
Age	Numeric	Age
Tenure	Numeric	The number of years customer has stayed with bank
Balance	Numeric	Bank Balance
NumOfProducts	Factor	The number of products offered by bank that customer utilizes
HasCrCard	Factor	Whether the person has a credit card of that bank or not
IsActiveMember	Factor	Whether a person is an active member or not
Estimated Salary	Numeric	The customers' salary estimated by the bank
Exited	Factor	Whether the customer churned or not

The boxplot for interaction between Geography and Salary that was found interesting in the EDA is provided below.



The boxplot for interaction between Geography and Salary that was found interesting in the EDA is provided below.



The formula used for calculating pooled estimates is given below.

$$PooledEstimate = \sum_{i=1}^{i=n} \frac{Estimate_i}{n} \quad \textbf{where}$$

n = number of imputed data sets

i = corresponds to each imputed data set

The variance across data sets of our standard estimates was calculated using this formula.

$$Variance_{across} = \sum_{i=1}^{i=m} \frac{q_i - \overline{q}}{m-1} \quad \textbf{where}$$

m = The number of imputed data sets

$\overline{q}$ = Average of standard estimates across all imputed data sets

q_i = The value of standard estimate for the i data set

i = corresponds to each imputed set

The weighted average for calculating final variance was:

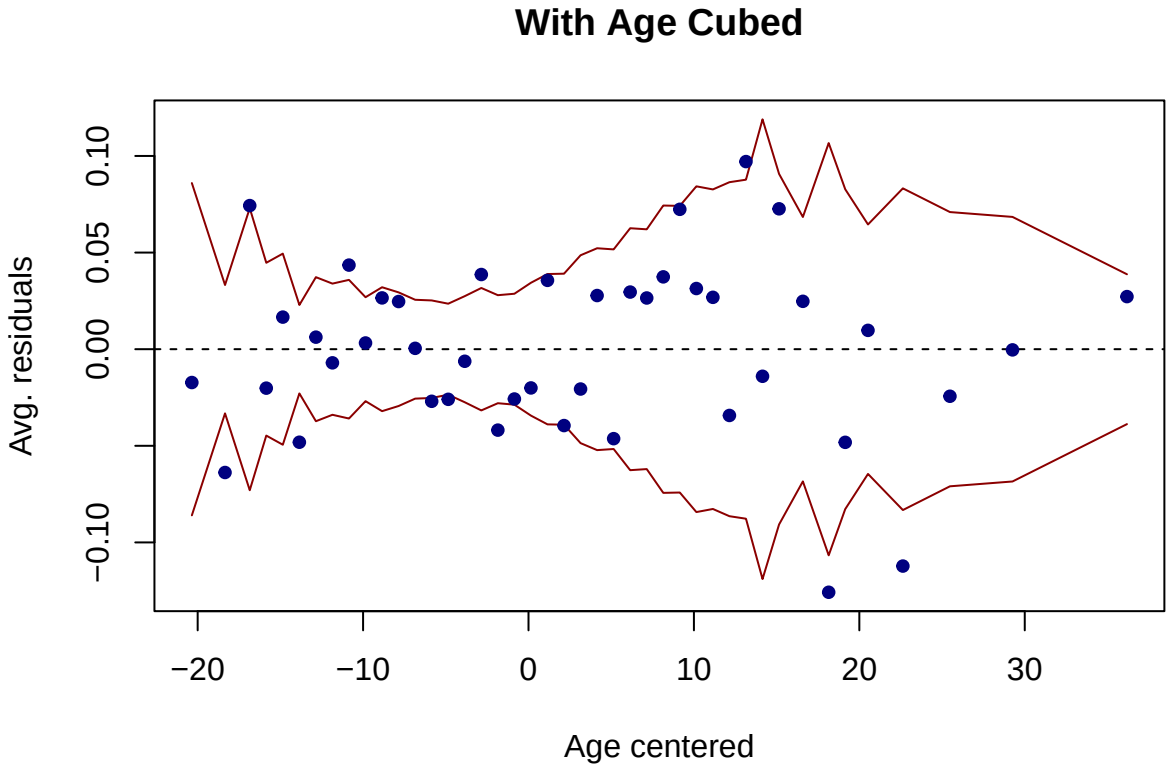
$$T_m = (1 + \frac{1}{m})b_m + u_m \quad \textbf{where}$$

m = number of imputed data sets

b_m = The variance across all data sets

u_m = The average variance of variance obtained from each imputed data

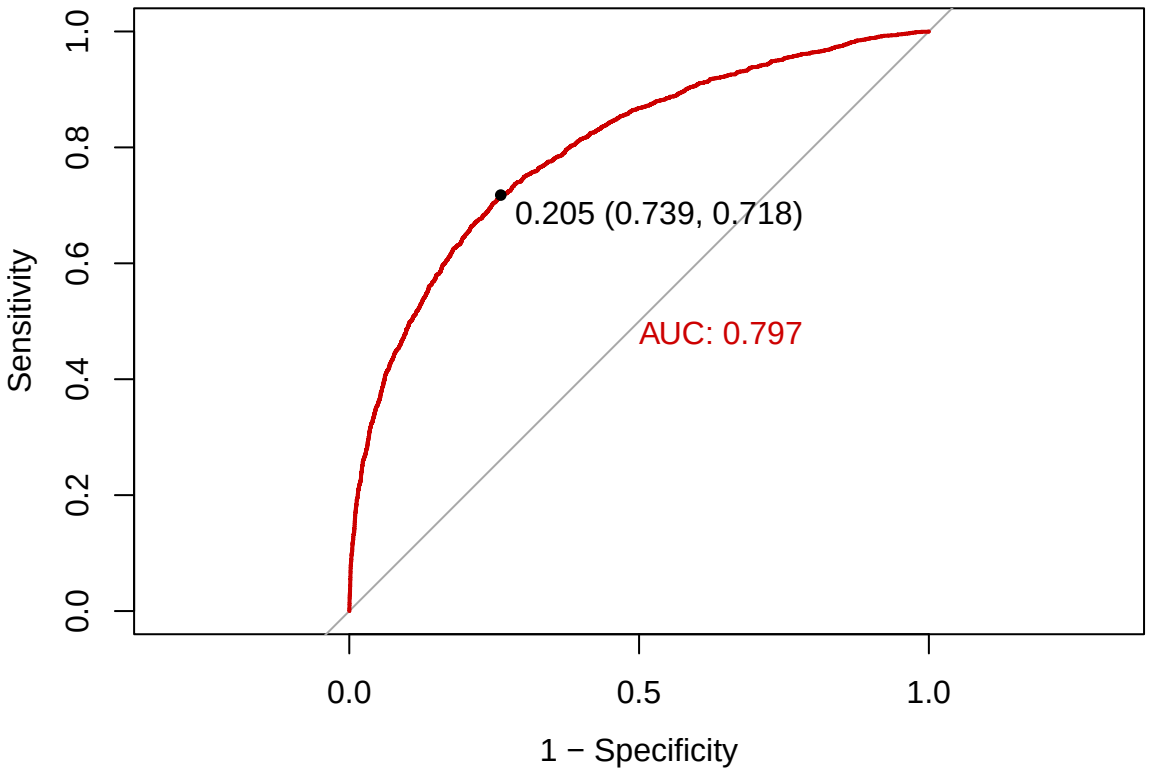
During model assessment, the finalized model was fitted onto the another data set that we got from multiple imputation. The binned plot of residuals vs age for that model is given below.



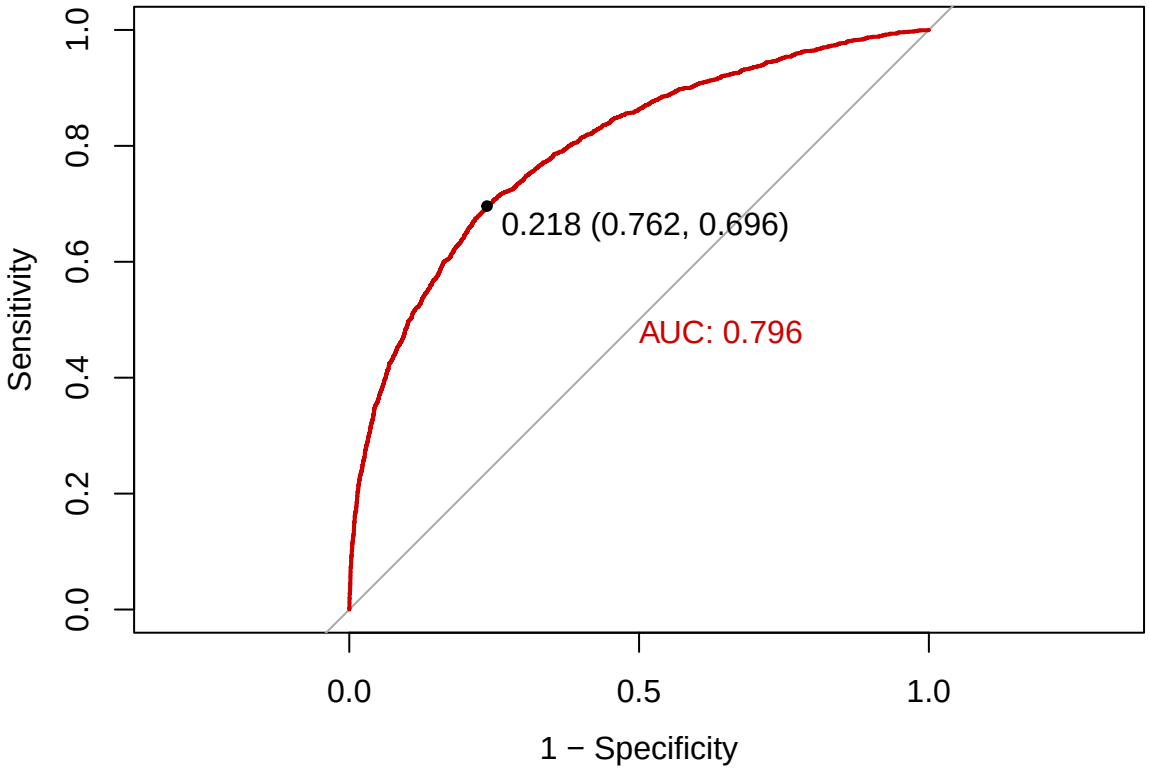
The VIF table for checking multicollinearity is provided below:

	x
Age	2.611937
Age_sq	4.122038
Age_cb	6.220245
IsActiveMember1	1.065197
GeographyGermany	1.370837
GeographySpain	1.169316
GenderMale	1.009899
Salary	1.003230
Balance	1.204095
CreditScore	1.001834

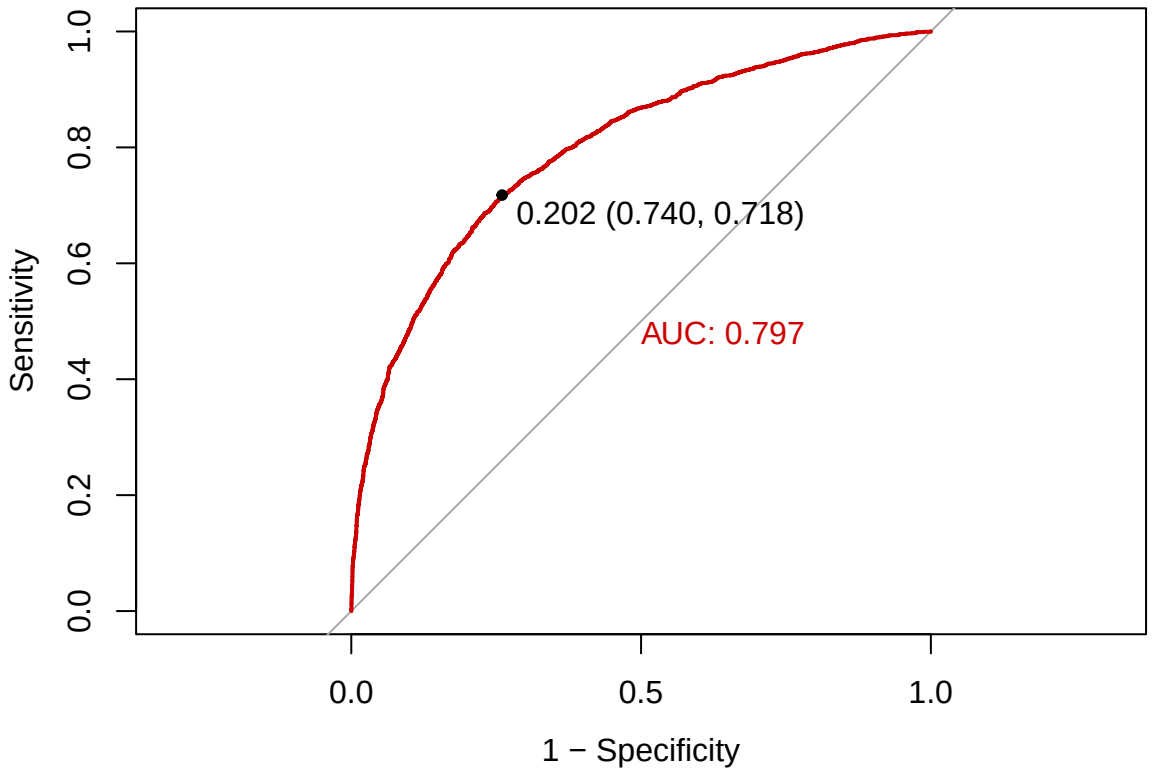
The ROC Curve while doing model validation when the actual model is fitted onto the actual data set is given below.



The ROC Curve while doing model validation when the Single Imputation model is fitted on the actual data set is given below.



The ROC Curve while doing model validation when the Multiple Imputation pooled model is fitted on the actual data set is given below.


Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
0.202 (0.740, 0.718)
AUC: 0.797
0.0
0.5
1.0
1 − Specificity

Code

```
library(mice)
library(VIM)
library(arm)
library(dplyr)
library(pROC)
library(caret)
library(VIM)
library(ggplot2)
library(rms)

#load csv
churn <-read.table('/Users/mohammadanas/Desktop/Duke MIDS/
                   Fall 2021/MODELLING AND REPRESENTATION OF DATA/
                   Final Project/Churn_Modelling.csv',
                   sep =",", header = TRUE, quote ="\"")


# load again to preserve actual values
churn_complete <-read.table('/Users/mohammadanas/
                            Desktop/Duke MIDS/Fall 2021/
                            MODELLING AND REPRESENTATION OF DATA/
                            Final Project/Churn_Modelling.csv',
                            sep =",", header = TRUE, quote ="\"")

churn_complete <- rename(churn_complete, Salary = EstimatedSalary)

# Rename Salary column
churn <- rename(churn, Salary = EstimatedSalary)
# create gender
columns = dim(churn)[1]
churn$gender_b = 0
churn$gender_b[churn$Gender == 'Female'] = 1


churn$Spain = 0
churn$Spain[churn$Geography == 'Spain'] = 1

churn$Germany = 0
churn$Germany[churn$Geography == 'Germany'] = 1

churn$France = 0
churn$France[churn$Geography == 'France'] = 1

# Age dependent variables
age <- churn[,c('gender_b','Germany','France')]
age$French_w <- age$gender_b*age$France
age$Germany_w <- age$gender_b*age$Germany


estimators <- as.matrix(c(-3.8,2.8,-0.02,-0.01,1.1,1.9)) #MAR
data <- as.matrix(cbind(rep(1,columns),age[,c(1,2,3,4,5)]))
```

```
logit_pi_R_age <- data %*% estimators

pi_R_age <- exp(logit_pi_R_age)/(1+exp(logit_pi_R_age))
set.seed(1031)
R_age <- rbinom(columns,1,pi_R_age)

# salary dependent variables
Salary <- churn[,c('Tenure','IsActiveMember')]

estimators <- as.matrix(c(4.4,-1.3,-1.2)) #MAR
data <- as.matrix(cbind(rep(1,columns),Salary[,c(1,2)]))

logit_pi_R_salary <- data %*% estimators

pi_R_salary <- exp(logit_pi_R_salary)/(1+exp(logit_pi_R_salary))
set.seed(1021)
R_salary <- rbinom(columns,1,pi_R_age)

# based on that impute missing values
churn <- cbind(churn, R_age, R_salary)
churn$Age[churn$R_age == 1] <- NA
churn$Salary[churn$R_salary == 1] <- NA

churn_missing <- churn[,4:14]
churn_only_missing_columns <- churn_missing[,c(4,10)]
md.pattern(churn_only_missing_columns)

# create box plots to check distribution

churn2 <- churn_missing

churn2$Age_Missing <- 'Not Missing'
churn2$Salary_Missing <- 'Not Missing'
churn2$Age_Missing[is.na(churn2$Age)] <- 'Missing'
churn2$Salary_Missing[is.na(churn2$Salary)] <-'Missing'

ggplot(churn2, aes(x=Age_Missing, y = Salary)) + geom_boxplot()
ggplot(churn2, aes(x=Salary_Missing, y = Age)) + geom_boxplot()


# change data type and create missing values
churn_missing$Geography <- factor(churn_missing$Geography)
churn_missing$Gender <- factor(churn_missing$Gender)
churn_missing$NumOfProducts <- factor(churn_missing$NumOfProducts)
churn_missing$HasCrCard <- factor(churn_missing$HasCrCard)
churn_missing$IsActiveMember <- factor(churn_missing$IsActiveMember)
churn_missing$Exited <- factor(churn_missing$Exited)

set.seed(20)
imputed_Ds_ppm <- mice(churn_missing, m=10, defaultMethod = c("pmm", "rf","rf"
                                                              ,"ppm","ppm","ppm","rf","rf"
                                                              ,"rf","pmm","rf"), print=F)
```

```r
predmatrix <- imputed_Ds_ppm$predictorMatrix
predmatrix["Age","Salary"] <- 0
predmatrix["Salary","Age"] <- 0

set.seed(26)
imputed_Ds_ppm <- mice(churn_missing, m=10, defaultMethod = c("pmm", "rf","rf"
                                                              ,"ppm","ppm","ppm","rf","rf"
                                                              ,"rf","pmm","rf"), print=F),
                                                                predictormatrix = predmatrix)
options(scipen=10)
densityplot(imputed_Ds_ppm)


# Create several datasets
d3 <- complete(imputed_Ds_ppm, 3)
d7 <- complete(imputed_Ds_ppm, 7)
d6 <- complete(imputed_Ds_ppm, 6)
d8 <- complete(imputed_Ds_ppm, 8)
d9 <- complete(imputed_Ds_ppm, 9)
d10 <- complete(imputed_Ds_ppm, 10)

#EDA
# low p-value geography is dependent include in the model
chisq.test(table(d6[,c("Geography","Exited")]))
# include in the model low p-value include in the model
chisq.test(table(d6[,c("Gender","Exited")]))
# include in the model low p-value  include in the model
chisq.test(table(d6[,c("NumOfProducts","Exited")]))
# surprisingly independent so do not include
chisq.test(table(d6[,c("HasCrCard","Exited")]))
# surprisingly independent so do not include
chisq.test(table(d6[,c("HasCrCard","Exited")]))
# dependent so include
chisq.test(table(d6[,c("IsActiveMember","Exited")]))

# continuous variables

# all most the same makes sense the as having a credit card was not also independent
# Probably the product was not revolving credit card
ggplot(d6,aes(x=Exited, y=CreditScore)) + geom_boxplot()
# include in the model as older people more likely to exit
ggplot(d6,aes(x=Exited, y=Age)) + geom_boxplot()
# tenure does not matter much
ggplot(d6,aes(x=Exited, y=Tenure)) + geom_boxplot()
# tenure does not matter much
ggplot(d6,aes(x=Exited, y=Tenure)) + geom_boxplot()
# there is a difference but might just include
ggplot(d6,aes(x=Exited, y=Balance)) + geom_boxplot()
# there is not much affect of salary but given that we have imputed it, lets just include
ggplot(d6,aes(x=Exited, y=EstimatedSalary)) + geom_boxplot()


# check for interactions for salary (Except Geography)
```

```
# Gender
ggplot(d6,aes(x=Exited, y=Salary)) + geom_boxplot() + facet_wrap(~Gender)

# Member
ggplot(d6,aes(x=Exited, y=Salary)) + geom_boxplot() + facet_wrap(~IsActiveMember)

# Credit card
ggplot(d6,aes(x=Exited , y=Salary)) + geom_boxplot() + facet_wrap(~HasCrCard)

# Num of products
ggplot(d6,aes(x=Exited, y=Salary)) + geom_boxplot() + facet_wrap(~NumOfProducts)

# Geography
ggplot(d6,aes(x=Exited, y=Salary)) + geom_boxplot() + facet_wrap(~Geography)



# check for interactions for balance (Except Geography)



# Gender
ggplot(d6,aes(x=Exited, y=Balance)) + geom_boxplot() + facet_wrap(~Gender)

# Member
ggplot(d6,aes(x=Exited, y=Balance)) + geom_boxplot() + facet_wrap(~IsActiveMember)

# Credit card
ggplot(d6,aes(x=Exited , y=Balance)) + geom_boxplot() + facet_wrap(~HasCrCard)

# Num of products
ggplot(d6,aes(x=Exited, y=Balance)) + geom_boxplot() + facet_wrap(~NumOfProducts)

# Geography
ggplot(d6,aes(x=Exited, y=Balance)) + geom_boxplot() + facet_wrap(~Geography)


# check for interactions for CreditScore (none needed)



# Gender
ggplot(d6,aes(x=Exited, y=CreditScore)) + geom_boxplot() + facet_wrap(~Gender)

# Member
ggplot(d6,aes(x=Exited, y=CreditScore)) + geom_boxplot() + facet_wrap(~IsActiveMember)

# Credit card
ggplot(d6,aes(x=Exited , y=CreditScore)) + geom_boxplot() + facet_wrap(~HasCrCard)
```

```r
# Num of products
ggplot(d6,aes(x=Exited, y=CreditScore)) + geom_boxplot() + facet_wrap(~NumOfProducts)

# Geography
ggplot(d6,aes(x=Exited, y=CreditScore)) + geom_boxplot() + facet_wrap(~Geography)


# check for interactions for Age (none needed)



# Gender
ggplot(d6,aes(x=Exited, y=Age)) + geom_boxplot() + facet_wrap(~Gender)

# Member
ggplot(d6,aes(x=Exited, y=Age)) + geom_boxplot() + facet_wrap(~IsActiveMember)

# Credit card
ggplot(d6,aes(x=Exited , y=Age)) + geom_boxplot() + facet_wrap(~HasCrCard)

# Num of products
ggplot(d6,aes(x=Exited, y=Age)) + geom_boxplot() + facet_wrap(~NumOfProducts)

# Geography
ggplot(d6,aes(x=Exited, y=Age)) + geom_boxplot() + facet_wrap(~Geography)

# preprocessing on d6
d6$CreditScore <- d6$CreditScore - mean(d6$CreditScore)
d6$Age <- d6$Age - mean(d6$Age)
d6$Salary <- d6$Salary - mean(d6$Salary)
d6$Balance <- d6$Balance - mean(d6$Balance)
d6$Age_sq <- (d6$Age)^2
d6$Age_cb <- (d6$Age)^3

### variables needed for model from EDA
# Geography
# active member
# Products
# Gender
# Age
# Balance
# Salary
# Salary and Geography
# Balance and Geography

# model building AIC and BIC
modeld6 <- glm(formula = Exited ~ Age + Geography + IsActiveMember +
                 CreditScore + HasCrCard +
                 Gender + Balance + EstimatedSalary + IsActiveMember +
                 EstimatedSalary:Geography +
                 Balance:Geography, family = binomial, data = d6)
```

```
null_model <- glm(Exited~1, data=d6, family=binomial)

step(null_model,scope=formula(modeld6),direction="both",
     trace=0)







# build model with age, age squared and cubed
modeld6 <-glm(formula = Exited ~ Age  + IsActiveMember + Geography +
                Gender + Salary + Balance +
                CreditScore, family = binomial, data = d6)
rawresid1 <- residuals(modeld6,"resp")

binnedplot(x=d6$Age,y=rawresid1,xlab="Age centered",
           col.int="red4",ylab="Avg. residuals",main="With Age Only",col.pts="navy")

modeld6_2 <-glm(formula = Exited ~ Age + Age_sq + IsActiveMember +
                  Geography + Gender + Salary + Balance +
                  CreditScore, family = binomial, data = d6)
rawresid1_2 <- residuals(modeld6_2,"resp")

binnedplot(x=d6$Age,y=rawresid1_2,xlab="Age centered",
           col.int="red4",ylab="Avg. residuals",main="With Age Squared",col.pts="navy")

modeld6_3 <-glm(formula = Exited ~ Age + Age_sq + Age_cb + IsActiveMember +
                  Geography + Gender + Salary + Balance +
                  CreditScore, family = binomial, data = d6)
rawresid1_3 <- residuals(modeld6_3,"resp")

binnedplot(x=d6$Age,y=rawresid1_3,xlab="Age centered",
           col.int="red4",ylab="Avg. residuals",main="With Age Cubed",col.pts="navy")

binnedplot(x=fitted(modeld6_3),y=rawresid1_3,xlab="Pred. probabilities",
           col.int="red4",ylab="Avg. residuals",
           main="Binned residual plot on Current Data",col.pts="navy")

vif(modeld6_3) # check multicollinearity

## test the assumptions on another imputed dataset
d3$CreditScore <- d3$CreditScore - mean(d3$CreditScore)
d3$Age <- d3$Age - mean(d3$Age)
d3$Salary <- d3$Salary - mean(d3$Salary)
d3$Balance <- d3$Balance - mean(d3$Balance)
d3$Age_sq <- (d3$Age)^2
d3$Age_cb <- (d3$Age)^3

modeld3 <-glm(formula = Exited ~ Age + Age_sq + Age_cb +
                IsActiveMember + Geography + Gender + Salary +
                Balance + CreditScore, family = binomial, data = d3)
```

```
rawresid3 <- residuals(modeld3,"resp")
binnedplot(x=fitted(modeld3),y=rawresid3,xlab="Pred. probabilities",
           col.int="red4",ylab="Avg. residuals",
           main="Binned residual plot on Another Data",col.pts="navy")

binnedplot(x=d3$Age,y=rawresid3,xlab="Age centered",
           col.int="red4",ylab="Avg. residuals",
           main="With Age Cubed",col.pts="navy")

## do the same thing on actual model

churn_complete$Geography <- factor(churn_complete$Geography)
churn_complete$Gender <- factor(churn_complete$Gender)
churn_complete$NumOfProducts <- factor(churn_complete$NumOfProducts)
churn_complete$HasCrCard <- factor(churn_complete$HasCrCard)
churn_complete$IsActiveMember <- factor(churn_complete$IsActiveMember)
churn_complete$Exited <- factor(churn_complete$Exited)
churn_complete$CreditScore <- churn_complete$CreditScore - mean(churn_complete$CreditScore)
churn_complete$Age <- churn_complete$Age - mean(churn_complete$Age)
churn_complete$alary <- churn_complete$Salary - mean(churn_complete$Salary)
churn_complete$Balance <- churn_complete$Balance - mean(churn_complete$Balance)
churn_complete$Age_sq <- (churn_complete$Age)^2
churn_complete$Age_cb <- (churn_complete$Age)^3


modelactual <-glm(formula = Exited ~ Age + Age_sq + Age_cb +
                    IsActiveMember + Geography + Gender + EstimatedSalary +
                    Balance + CreditScore, family = binomial, data = churn_complete)



### Define functions to create pooled model


centering <- function(s){
  a = s - mean(s)
  return(a)
}

centering_power <- function(s,power){
  k = s - mean(s)
  l <- k^power
  return(l)
}

model_all <- with(data=imputed_Ds_ppm, glm(Exited ~  centering(Age) +
                                             centering_power(Age,2) +
                                             centering_power(Age,3) +
                                             IsActiveMember +
                                             Geography +
                                             Gender +
                                             centering(Salary) +
                                             centering(Balance) +
```

```
                                                centering(CreditScore), family = binomial))
pooled_model <- pool(model_all)


### single imputation method


churn_knn <- kNN(churn_missing, variable = c('Age','Salary'), k=2)

churn_knn$Geography <- factor(churn_knn$Geography)
churn_knn$Gender <- factor(churn_knn$Gender)
churn_knn$NumOfProducts <- factor(churn_knn$NumOfProducts)
churn_knn$HasCrCard <- factor(churn_knn$HasCrCard)
churn_knn$IsActiveMember <- factor(churn_knn$IsActiveMember)
churn_knn$Exited <- factor(churn_knn$Exited)
churn_knn$CreditScore <- churn_knn$CreditScore - mean(churn_knn$CreditScore)
churn_knn$Age <- churn_knn$Age - mean(churn_knn$Age)
churn_knn$Salary <- churn_knn$Salary - mean(churn_knn$Salary)
churn_knn$Balance <- churn_knn$Balance - mean(churn_knn$Balance)
churn_knn$Age_sq <- (churn_knn$Age)^2
churn_knn$Age_cb <- (churn_knn$Age)^3

modelknn <-glm(formula = Exited ~ Age + Age_sq + Age_cb +
                 IsActiveMember + Geography + Gender + Salary +
                 Balance + CreditScore, family = binomial, data = churn_knn)


summary(modelknn)
summary(modelactual)
summary(pooled_model)



# ROC for Knn Model
predictknn <- predict(modelknn, churn_complete)
predict_by_knn <- exp(predictknn)/(1+exp(predictknn))
roc(churn_complete$Exited,predict_by_knn,plot=T,print.thres="best",legacy.axes=T,
    print.auc =T,col="red3")

# ROC for the actual real model
predictactual <- predict(modelactual, churn_complete)
predict_by_actual <- exp(predictactual)/(1+exp(predictactual))
roc(churn_complete$Exited,predict_by_actual,plot=T,print.thres="best",legacy.axes=T,
    print.auc =T,col="red3")


# ROC for the Pooled Model
model_pool_predict <- modelactual
pool_estimates <- summary(pooled_model)$estimate
names(pool_estimates) <- names(model_pool_predict$coefficients)
model_pool_predict$coefficients <- pool_estimates
predict_MI <- predict(model_pool_predict, churn_complete)
predict_by_MI <- exp(predict_MI)/(1+exp(predict_MI))
```

```
roc(churn_complete$Exited,predict_by_MI,plot=T,print.thres="best",legacy.axes=T,
    print.auc =T,col="red3")


# Confusion Matrices


# model actual
Conf_mat_actual <- confusionMatrix(as.factor(ifelse(fitted(modelactual) >= 0.205,
                                                    "1","0")),
                                   as.factor(churn_complete$Exited),positive = "1")


Conf_mat_actual$overall["Accuracy"];
Conf_mat_actual$byClass[c("Sensitivity","Specificity")]

#model MI pooled
Conf_mat_MI <- confusionMatrix(as.factor(ifelse(fitted(model_pool_predict) >= 0.202,
                                                "1","0")),
                               as.factor(churn_complete$Exited),positive = "1")

Conf_mat_MI$overall["Accuracy"];
Conf_mat_MI$byClass[c("Sensitivity","Specificity")]

# model knn
Conf_mat_KNN <- confusionMatrix(as.factor(ifelse(fitted(modelknn) >= 0.218,
                                                 "1","0")),
                                as.factor(churn_complete$Exited),positive = "1")

Conf_mat_KNN$overall["Accuracy"];
Conf_mat_KNN$byClass[c("Sensitivity","Specificity")]
```

Citations

Frisell, T. “SP0187 Why Missing Data Is a Problem, and What You Shouldn’t Do to Solve It.” Annals of the Rheumatic Diseases, vol. 75, no. Suppl 2, 2016, https://doi.org/10.1136/annrheumdis-2016-eular.6249.