German Credit Data Exploration\_3

Dr. Prashant Mishra

3/26/2018

# 1. Import the Clean data

We already have a clean file “german\_credit\_full.csv” to import.

credit\_dataset <- read.csv("german\_credit\_full.csv",stringsAsFactors = TRUE)  
str(credit\_dataset)

## 'data.frame': 1000 obs. of 21 variables:  
## $ Class : Factor w/ 2 levels "Bad","Good": 2 1 2 2 1 2 2 2 2 1 ...  
## $ CheckingAccountStatus : Factor w/ 4 levels "0.to.200","gt.200",..: 3 1 4 3 3 4 4 1 4 1 ...  
## $ Duration : int 6 48 12 42 24 36 24 36 12 30 ...  
## $ CreditHistory : Factor w/ 5 levels "Critical","Delay",..: 1 4 1 4 2 4 4 4 4 1 ...  
## $ Purpose : Factor w/ 10 levels "Business","DomesticAppliance",..: 7 7 3 4 5 3 4 10 7 5 ...  
## $ Amount : int 1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...  
## $ SavingsAccountBonds : Factor w/ 5 levels "100.to.500","500.to.1000",..: 5 4 4 4 4 5 2 4 3 4 ...  
## $ EmploymentDuration : Factor w/ 5 levels "0.to.1","1.to.4",..: 4 2 3 3 2 2 4 2 3 5 ...  
## $ InstallmentRatePercentage: int 4 2 2 2 3 2 3 2 2 4 ...  
## $ Personal : Factor w/ 4 levels "Female.NotSingle",..: 4 1 4 4 4 4 4 4 2 3 ...  
## $ OtherDebtorsGuarantors : Factor w/ 3 levels "CoApplicant",..: 3 3 3 2 3 3 3 3 3 3 ...  
## $ ResidenceDuration : int 4 2 3 4 4 4 4 2 4 2 ...  
## $ Property : Factor w/ 4 levels "CarOther","Insurance",..: 3 3 3 2 4 4 2 1 3 1 ...  
## $ Age : int 67 22 49 45 53 35 53 35 61 28 ...  
## $ OtherInstallmentPlans : Factor w/ 3 levels "Bank","None",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ Housing : Factor w/ 3 levels "ForFree","Own",..: 2 2 2 1 1 1 2 3 2 2 ...  
## $ NumberExistingCredits : int 2 1 1 1 2 1 1 1 1 2 ...  
## $ Job : Factor w/ 4 levels "Management.SelfEmp.HighlyQualified",..: 2 2 4 2 2 4 2 1 4 1 ...  
## $ NumberPeopleMaintenance : int 1 1 2 2 2 2 1 1 1 1 ...  
## $ Telephone : int 1 0 0 0 0 1 0 1 0 0 ...  
## $ ForeignWorker : int 1 1 1 1 1 1 1 1 1 1 ...

credit\_dataset$Duration <- cut(credit\_dataset$Duration, c(0,6,12,18,24,30,36,42,48,54,60,66,72,78),labels = c("0.to.6","6.to.12","12.to.18","18.to.24","24.to.30","30.to.36","36.to.42","42.to.48","48.to.54","54.to.60","60.to.66","66.to.72","72.to.78"))  
credit\_dataset$Amount <- cut(credit\_dataset$Amount, c(0,25,30,35,40,50,60,70,80),labels = c("0.to.25","25.to.30","30.to.35","35.to.40","40.to.50","50.to.60","60.to.70","70.to.80"))  
  
credit\_dataset$Age <- cut(credit\_dataset$Age, c(0,6,12,18,24,30,36,42,48,54,60,66,72,78),labels = c("0.to.6","6.to.12","12.to.18","18.to.24","24.to.30","30.to.36","36.to.42","42.to.48","48.to.54","54.to.60","60.to.66","66.to.72","72.to.78"))  
  
credit\_dataset$InstallmentRatePercentage <- as.factor(credit\_dataset$InstallmentRatePercentage)  
credit\_dataset$ResidenceDuration <- as.factor(credit\_dataset$ResidenceDuration)  
credit\_dataset$NumberExistingCredits <- as.factor(credit\_dataset$NumberExistingCredits)  
credit\_dataset$InstallmentRatePercentage <- as.factor(credit\_dataset$InstallmentRatePercentage)

Save everything to a csv file, so that it is reusable

save(credit\_dataset, file = 'credit\_dataset')  
write.csv(credit\_dataset, 'credit\_dataset.csv',  
 row.names = FALSE)

str(credit\_dataset)

## 'data.frame': 1000 obs. of 21 variables:  
## $ Class : Factor w/ 2 levels "Bad","Good": 2 1 2 2 1 2 2 2 2 1 ...  
## $ CheckingAccountStatus : Factor w/ 4 levels "0.to.200","gt.200",..: 3 1 4 3 3 4 4 1 4 1 ...  
## $ Duration : Factor w/ 13 levels "0.to.6","6.to.12",..: 1 8 2 7 4 6 4 6 2 5 ...  
## $ CreditHistory : Factor w/ 5 levels "Critical","Delay",..: 1 4 1 4 2 4 4 4 4 1 ...  
## $ Purpose : Factor w/ 10 levels "Business","DomesticAppliance",..: 7 7 3 4 5 3 4 10 7 5 ...  
## $ Amount : Factor w/ 8 levels "0.to.25","25.to.30",..: NA NA NA NA NA NA NA NA NA NA ...  
## $ SavingsAccountBonds : Factor w/ 5 levels "100.to.500","500.to.1000",..: 5 4 4 4 4 5 2 4 3 4 ...  
## $ EmploymentDuration : Factor w/ 5 levels "0.to.1","1.to.4",..: 4 2 3 3 2 2 4 2 3 5 ...  
## $ InstallmentRatePercentage: Factor w/ 4 levels "1","2","3","4": 4 2 2 2 3 2 3 2 2 4 ...  
## $ Personal : Factor w/ 4 levels "Female.NotSingle",..: 4 1 4 4 4 4 4 4 2 3 ...  
## $ OtherDebtorsGuarantors : Factor w/ 3 levels "CoApplicant",..: 3 3 3 2 3 3 3 3 3 3 ...  
## $ ResidenceDuration : Factor w/ 4 levels "1","2","3","4": 4 2 3 4 4 4 4 2 4 2 ...  
## $ Property : Factor w/ 4 levels "CarOther","Insurance",..: 3 3 3 2 4 4 2 1 3 1 ...  
## $ Age : Factor w/ 13 levels "0.to.6","6.to.12",..: 12 4 9 8 9 6 9 6 11 5 ...  
## $ OtherInstallmentPlans : Factor w/ 3 levels "Bank","None",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ Housing : Factor w/ 3 levels "ForFree","Own",..: 2 2 2 1 1 1 2 3 2 2 ...  
## $ NumberExistingCredits : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 2 1 1 1 1 2 ...  
## $ Job : Factor w/ 4 levels "Management.SelfEmp.HighlyQualified",..: 2 2 4 2 2 4 2 1 4 1 ...  
## $ NumberPeopleMaintenance : int 1 1 2 2 2 2 1 1 1 1 ...  
## $ Telephone : int 1 0 0 0 0 1 0 1 0 0 ...  
## $ ForeignWorker : int 1 1 1 1 1 1 1 1 1 1 ...

# Let’s prepare the data for modelling

* Make it all numerical data
* Pivot all categorical data such that each category reprsents a column

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(dummies)

## dummies-1.5.6 provided by Decision Patterns

credit\_dataset\_withoutclass <- credit\_dataset %>% select(c(-Class))  
ml\_credit\_dataset <- dummy.data.frame(credit\_dataset\_withoutclass, sep = ".")  
ml\_credit\_dataset$Class <- credit\_dataset$Class  
str(ml\_credit\_dataset)

## 'data.frame': 1000 obs. of 87 variables:  
## $ CheckingAccountStatus.0.to.200 : int 0 1 0 0 0 0 0 1 0 1 ...  
## $ CheckingAccountStatus.gt.200 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ CheckingAccountStatus.lt.0 : int 1 0 0 1 1 0 0 0 0 0 ...  
## $ CheckingAccountStatus.none : int 0 0 1 0 0 1 1 0 1 0 ...  
## $ Duration.0.to.6 : int 1 0 0 0 0 0 0 0 0 0 ...  
## $ Duration.6.to.12 : int 0 0 1 0 0 0 0 0 1 0 ...  
## $ Duration.12.to.18 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Duration.18.to.24 : int 0 0 0 0 1 0 1 0 0 0 ...  
## $ Duration.24.to.30 : int 0 0 0 0 0 0 0 0 0 1 ...  
## $ Duration.30.to.36 : int 0 0 0 0 0 1 0 1 0 0 ...  
## $ Duration.36.to.42 : int 0 0 0 1 0 0 0 0 0 0 ...  
## $ Duration.42.to.48 : int 0 1 0 0 0 0 0 0 0 0 ...  
## $ Duration.48.to.54 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Duration.54.to.60 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Duration.66.to.72 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ CreditHistory.Critical : int 1 0 1 0 0 0 0 0 0 1 ...  
## $ CreditHistory.Delay : int 0 0 0 0 1 0 0 0 0 0 ...  
## $ CreditHistory.NoCredit.AllPaid : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ CreditHistory.PaidDuly : int 0 1 0 1 0 1 1 1 1 0 ...  
## $ CreditHistory.ThisBank.AllPaid : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Purpose.Business : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Purpose.DomesticAppliance : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Purpose.Education : int 0 0 1 0 0 1 0 0 0 0 ...  
## $ Purpose.Furniture.Equipment : int 0 0 0 1 0 0 1 0 0 0 ...  
## $ Purpose.NewCar : int 0 0 0 0 1 0 0 0 0 1 ...  
## $ Purpose.Others : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Purpose.Radio.Television : int 1 1 0 0 0 0 0 0 1 0 ...  
## $ Purpose.Repairs : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Purpose.Retraining : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Purpose.UsedCar : int 0 0 0 0 0 0 0 1 0 0 ...  
## $ SavingsAccountBonds.100.to.500 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ SavingsAccountBonds.500.to.1000 : int 0 0 0 0 0 0 1 0 0 0 ...  
## $ SavingsAccountBonds.gt.1000 : int 0 0 0 0 0 0 0 0 1 0 ...  
## $ SavingsAccountBonds.lt.100 : int 0 1 1 1 1 0 0 1 0 1 ...  
## $ SavingsAccountBonds.Unknown : int 1 0 0 0 0 1 0 0 0 0 ...  
## $ EmploymentDuration.0.to.1 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ EmploymentDuration.1.to.4 : int 0 1 0 0 1 1 0 1 0 0 ...  
## $ EmploymentDuration.4.to.7 : int 0 0 1 1 0 0 0 0 1 0 ...  
## $ EmploymentDuration.gt.7 : int 1 0 0 0 0 0 1 0 0 0 ...  
## $ EmploymentDuration.Unemployed : int 0 0 0 0 0 0 0 0 0 1 ...  
## $ InstallmentRatePercentage.1 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ InstallmentRatePercentage.2 : int 0 1 1 1 0 1 0 1 1 0 ...  
## $ InstallmentRatePercentage.3 : int 0 0 0 0 1 0 1 0 0 0 ...  
## $ InstallmentRatePercentage.4 : int 1 0 0 0 0 0 0 0 0 1 ...  
## $ Personal.Female.NotSingle : int 0 1 0 0 0 0 0 0 0 0 ...  
## $ Personal.Male.Divorced.Seperated : int 0 0 0 0 0 0 0 0 1 0 ...  
## $ Personal.Male.Married.Widowed : int 0 0 0 0 0 0 0 0 0 1 ...  
## $ Personal.Male.Single : int 1 0 1 1 1 1 1 1 0 0 ...  
## $ OtherDebtorsGuarantors.CoApplicant : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ OtherDebtorsGuarantors.Guarantor : int 0 0 0 1 0 0 0 0 0 0 ...  
## $ OtherDebtorsGuarantors.None : int 1 1 1 0 1 1 1 1 1 1 ...  
## $ ResidenceDuration.1 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ ResidenceDuration.2 : int 0 1 0 0 0 0 0 1 0 1 ...  
## $ ResidenceDuration.3 : int 0 0 1 0 0 0 0 0 0 0 ...  
## $ ResidenceDuration.4 : int 1 0 0 1 1 1 1 0 1 0 ...  
## $ Property.CarOther : int 0 0 0 0 0 0 0 1 0 1 ...  
## $ Property.Insurance : int 0 0 0 1 0 0 1 0 0 0 ...  
## $ Property.RealEstate : int 1 1 1 0 0 0 0 0 1 0 ...  
## $ Property.Unknown : int 0 0 0 0 1 1 0 0 0 0 ...  
## $ Age.18.to.24 : int 0 1 0 0 0 0 0 0 0 0 ...  
## $ Age.24.to.30 : int 0 0 0 0 0 0 0 0 0 1 ...  
## $ Age.30.to.36 : int 0 0 0 0 0 1 0 1 0 0 ...  
## $ Age.36.to.42 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Age.42.to.48 : int 0 0 0 1 0 0 0 0 0 0 ...  
## $ Age.48.to.54 : int 0 0 1 0 1 0 1 0 0 0 ...  
## $ Age.54.to.60 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Age.60.to.66 : int 0 0 0 0 0 0 0 0 1 0 ...  
## $ Age.66.to.72 : int 1 0 0 0 0 0 0 0 0 0 ...  
## $ Age.72.to.78 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ OtherInstallmentPlans.Bank : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ OtherInstallmentPlans.None : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ OtherInstallmentPlans.Stores : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Housing.ForFree : int 0 0 0 1 1 1 0 0 0 0 ...  
## $ Housing.Own : int 1 1 1 0 0 0 1 0 1 1 ...  
## $ Housing.Rent : int 0 0 0 0 0 0 0 1 0 0 ...  
## $ NumberExistingCredits.1 : int 0 1 1 1 0 1 1 1 1 0 ...  
## $ NumberExistingCredits.2 : int 1 0 0 0 1 0 0 0 0 1 ...  
## $ NumberExistingCredits.3 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ NumberExistingCredits.4 : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Job.Management.SelfEmp.HighlyQualified: int 0 0 0 0 0 0 0 1 0 1 ...  
## $ Job.SkilledEmployee : int 1 1 0 1 1 0 1 0 0 0 ...  
## $ Job.UnemployedUnskilled : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Job.UnskilledResident : int 0 0 1 0 0 1 0 0 1 0 ...  
## $ NumberPeopleMaintenance : int 1 1 2 2 2 2 1 1 1 1 ...  
## $ Telephone : int 1 0 0 0 0 1 0 1 0 0 ...  
## $ ForeignWorker : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Class : Factor w/ 2 levels "Bad","Good": 2 1 2 2 1 2 2 2 2 1 ...  
## - attr(\*, "dummies")=List of 16  
## ..$ CheckingAccountStatus : int 1 2 3 4  
## ..$ Duration : int 5 6 7 8 9 10 11 12 13 14 ...  
## ..$ CreditHistory : int 16 17 18 19 20  
## ..$ Purpose : int 21 22 23 24 25 26 27 28 29 30  
## ..$ SavingsAccountBonds : int 31 32 33 34 35  
## ..$ EmploymentDuration : int 36 37 38 39 40  
## ..$ InstallmentRatePercentage: int 41 42 43 44  
## ..$ Personal : int 45 46 47 48  
## ..$ OtherDebtorsGuarantors : int 49 50 51  
## ..$ ResidenceDuration : int 52 53 54 55  
## ..$ Property : int 56 57 58 59  
## ..$ Age : int 60 61 62 63 64 65 66 67 68 69  
## ..$ OtherInstallmentPlans : int 70 71 72  
## ..$ Housing : int 73 74 75  
## ..$ NumberExistingCredits : int 76 77 78 79  
## ..$ Job : int 80 81 82 83

save(ml\_credit\_dataset, file = 'ml\_credit\_dataset')  
write.csv(ml\_credit\_dataset, 'ml\_credit\_dataset.csv',  
 row.names = FALSE)