Interest_Rate_Prediction.R

Mina Mahesh

2026-01-16

```
# Objective: Develop a predictive model for interest rates based on borrower profiles

# Loading the dataset
ld=read.csv("loans data.csv",stringsAsFactors = FALSE)
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

glimpse(ld)

## Rows: 2,500
## Columns: 15
## $ ID                              <int> 81174, 99592, 80059, 15825, 33182,
6240~
## $ Amount.Requested                <chr> "20000", "19200", "35000", "10000",
"12~
## $ Amount.Funded.By.Investors      <chr> "20000", "19200", "35000", "9975",
"120~
## $ Interest.Rate                   <chr> "8.90%", "12.12%", "21.98%", "9.99%
", "~
## $ Loan.Length                     <chr> "36 months", "36 months", "60 month
s", ~
## $ Loan.Purpose                    <chr> "debt_consolidation", "debt_consoli
dati~
## $ Debt.To.Income.Ratio            <chr> "14.90%", "28.36%", "23.81%", "14.3
0%",~
## $ State                           <chr> "SC", "TX", "CA", "KS", "NJ", "CT",
"MA~
## $ Home.Ownership                  <chr> "MORTGAGE", "MORTGAGE", "MORTGAGE",
"MO~
## $ Monthly.Income                  <dbl> 6541.67, 4583.33, 11500.00, 3833.33
, 31~
## $ FICO.Range                      <chr> "735-739", "715-719", "690-694", "6
```

```
95-6~
## $ Open.CREDIT.Lines              <chr> "14", "12", "14", "10", "11", "17",
"10~
## $ Revolving.CREDIT.Balance       <chr> "14272", "11140", "21977", "9346",
"144~
## $ Inquiries.in.the.Last.6.Months <int> 2, 1, 1, 0, 0, 2, 0, 0, 1, 0, 0, 0,
0, ~
## $ Employment.Length              <chr> "< 1 year", "2 years", "2 years", "
5 ye~

# Cleaning numeric fields by removing special characters and casting types
ld=ld %>%
  mutate(Interest.Rate=as.numeric(gsub("%","",Interest.Rate)) ,
         Debt.To.Income.Ratio=as.numeric(gsub("%","",Debt.To.Income.Ratio)) ,
         Open.CREDIT.Lines=as.numeric(Open.CREDIT.Lines) ,
         Amount.Requested=as.numeric(Amount.Requested) ,
         Amount.Funded.By.Investors=as.numeric(Amount.Funded.By.Investors),
         Revolving.CREDIT.Balance=as.numeric(Revolving.CREDIT.Balance)
  )

## Warning: There were 4 warnings in `mutate()`.
## The first warning was:
## i In argument: `Open.CREDIT.Lines = as.numeric(Open.CREDIT.Lines)`.
## Caused by warning:
## ! NAs introduced by coercion
## i Run `dplyr::last_dplyr_warnings()` to see the 3 remaining warnings.

glimpse(ld)

## Rows: 2,500
## Columns: 15
## $ ID                             <int> 81174, 99592, 80059, 15825, 33182,
6240~
## $ Amount.Requested               <dbl> 20000, 19200, 35000, 10000, 12000,
6000~
## $ Amount.Funded.By.Investors     <dbl> 20000.00, 19200.00, 35000.00, 9975.
00, ~
## $ Interest.Rate                  <dbl> 8.90, 12.12, 21.98, 9.99, 11.71, 15
.31,~
## $ Loan.Length                    <chr> "36 months", "36 months", "60 month
s", ~
## $ Loan.Purpose                   <chr> "debt_consolidation", "debt_consoli
dati~
## $ Debt.To.Income.Ratio           <dbl> 14.90, 28.36, 23.81, 14.30, 18.78,
20.0~
## $ State                          <chr> "SC", "TX", "CA", "KS", "NJ", "CT",
"MA~
## $ Home.Ownership                 <chr> "MORTGAGE", "MORTGAGE", "MORTGAGE",
"MO~
## $ Monthly.Income                 <dbl> 6541.67, 4583.33, 11500.00, 3833.33
, 31~
```

```
## $ FICO.Range                      <chr> "735-739", "715-719", "690-694", "6
95-6~
## $ Open.CREDIT.Lines               <dbl> 14, 12, 14, 10, 11, 17, 10, 12, 9,
8, 1~
## $ Revolving.CREDIT.Balance        <dbl> 14272, 11140, 21977, 9346, 14469, 1
0391~
## $ Inquiries.in.the.Last.6.Months <int> 2, 1, 1, 0, 0, 2, 0, 0, 1, 0, 0, 0,
0, ~
## $ Employment.Length               <chr> "< 1 year", "2 years", "2 years", "
5 ye~

# Excluding 'Amount.Funded.By.Investors' (not known at the time of applicatio
n)
ld = ld %>%
  select(-Amount.Funded.By.Investors)
glimpse(ld)

## Rows: 2,500
## Columns: 14
## $ ID                              <int> 81174, 99592, 80059, 15825, 33182,
6240~
## $ Amount.Requested                <dbl> 20000, 19200, 35000, 10000, 12000,
6000~
## $ Interest.Rate                   <dbl> 8.90, 12.12, 21.98, 9.99, 11.71, 15
.31,~
## $ Loan.Length                     <chr> "36 months", "36 months", "60 month
s", ~
## $ Loan.Purpose                    <chr> "debt_consolidation", "debt_consoli
dati~
## $ Debt.To.Income.Ratio            <dbl> 14.90, 28.36, 23.81, 14.30, 18.78,
20.0~
## $ State                           <chr> "SC", "TX", "CA", "KS", "NJ", "CT",
"MA~
## $ Home.Ownership                  <chr> "MORTGAGE", "MORTGAGE", "MORTGAGE",
"MO~
## $ Monthly.Income                  <dbl> 6541.67, 4583.33, 11500.00, 3833.33
, 31~
## $ FICO.Range                      <chr> "735-739", "715-719", "690-694", "6
95-6~
## $ Open.CREDIT.Lines               <dbl> 14, 12, 14, 10, 11, 17, 10, 12, 9,
8, 1~
## $ Revolving.CREDIT.Balance        <dbl> 14272, 11140, 21977, 9346, 14469, 1
0391~
## $ Inquiries.in.the.Last.6.Months <int> 2, 1, 1, 0, 0, 2, 0, 0, 1, 0, 0, 0,
0, ~
## $ Employment.Length               <chr> "< 1 year", "2 years", "2 years", "
5 ye~

# Tranforming 'fico': Derived as the midpoint of the provided FICO Range
ld= ld %>%
```

```
  mutate(f1=as.numeric(substr(FICO.Range,1,3)),
         f2=as.numeric(substr(FICO.Range,5,7)),
         fico=0.5*(f1+f2)
  ) %>%
  select(-FICO.Range,-f1,-f2)
glimpse(ld)
```

```
## Rows: 2,500
## Columns: 14
## $ ID                              <int> 81174, 99592, 80059, 15825, 33182,
6240~
## $ Amount.Requested                <dbl> 20000, 19200, 35000, 10000, 12000,
6000~
## $ Interest.Rate                   <dbl> 8.90, 12.12, 21.98, 9.99, 11.71, 15
.31,~
## $ Loan.Length                     <chr> "36 months", "36 months", "60 month
s", ~
## $ Loan.Purpose                    <chr> "debt_consolidation", "debt_consoli
dati~
## $ Debt.To.Income.Ratio            <dbl> 14.90, 28.36, 23.81, 14.30, 18.78,
20.0~
## $ State                           <chr> "SC", "TX", "CA", "KS", "NJ", "CT",
"MA~
## $ Home.Ownership                  <chr> "MORTGAGE", "MORTGAGE", "MORTGAGE",
"MO~
## $ Monthly.Income                  <dbl> 6541.67, 4583.33, 11500.00, 3833.33
, 31~
## $ Open.CREDIT.Lines               <dbl> 14, 12, 14, 10, 11, 17, 10, 12, 9,
8, 1~
## $ Revolving.CREDIT.Balance        <dbl> 14272, 11140, 21977, 9346, 14469, 1
0391~
## $ Inquiries.in.the.Last.6.Months <int> 2, 1, 1, 0, 0, 2, 0, 0, 1, 0, 0, 0,
0, ~
## $ Employment.Length               <chr> "< 1 year", "2 years", "2 years", "
5 ye~
## $ fico                            <dbl> 737, 717, 692, 697, 697, 672, 722,
707,~
```

```
table(ld$Employment.Length)
```

```
##
##         .  < 1 year    1 year 10+ years   2 years   3 years   4 years   5
years
##         2       249       177       653       243       235       191
202
##   6 years   7 years   8 years   9 years       n/a
##       163       127       108        72        77
```

```
# Transforming 'Employment.Length' into a continuous numeric scale for modeli
ng
ld=ld %>%
```

```
  mutate(el=ifelse(substr(Employment.Length,1,2)=="10",10,Employment.Length),
         el=ifelse(substr(Employment.Length,1,1)=="<",0,el),
         el=gsub("years","",el),
         el=gsub("year","",el),
         el=as.numeric(el)
  ) %>%
  select(-Employment.Length) %>%
  na.omit()
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `el = as.numeric(el)`.
## Caused by warning:
## ! NAs introduced by coercion
```

```
# One-Hot Encoding: Transforming 'Home.Ownership' into binary indicators
table(ld$Home.Ownership)
```

```
##
## MORTGAGE    OTHER      OWN     RENT
##     1098        5      186     1107
```

```
ld=ld %>%
  mutate(HW_RENT=as.numeric(Home.Ownership=="RENT"),
         HW_MORT=as.numeric(Home.Ownership=="MORTGAGE"),
         HW_OWN=as.numeric(Home.Ownership=="OWN")) %>%
  select(-Home.Ownership)
```

```
# Consolidating 'Loan.Purpose' into broader categories based on average inter
est rate impact
table(ld$Loan.Purpose)
```

```
##
##                car        credit_card debt_consolidation        educationa
l
##                 49                426               1261                1
4
##   home_improvement              house     major_purchase             medica
l
##                144                 20                 98                2
8
##             moving              other   renewable_energy     small_busines
s
##                 28                184                  4                8
2
##           vacation            wedding
##                 20                 38
```

```
round(tapply(ld$Interest.Rate,ld$Loan.Purpose,mean),0)
```

```
##                car        credit_card debt_consolidation        educationa
l
```

```
##                 11                 13                 14                  1
1
##   home_improvement              house     major_purchase             medica
l
##                 12                 13                 11                  1
2
##             moving              other   renewable_energy     small_busines
s
##                 14                 13                 10                  1
3
##           vacation            wedding
##                 12                 12
```

```
# 1 : car,educational, major_purchase - 11
# 2 : credit_card, house, small_business, other - 13
# 3 : home_improvement, medical, vacation, wedding - 12
# 4 : debt_consolidation, moving - 14
# 5 : renewable_energy - 10

ld=ld %>%
  mutate( LP_1=as.numeric(Loan.Purpose %in% c("car","educational", "major_pur
chase")),
          LP_2=as.numeric(Loan.Purpose %in% c("credit_card","house","small_bu
siness","other")),
          LP_3=as.numeric(Loan.Purpose %in% c("home_improvement"," medical",
"vacation", "wedding")),
          LP_4=as.numeric(Loan.Purpose %in% c("debt_consolidation","moving"))
  ) %>%
  select(-Loan.Purpose)

# Creating an indicator for 36-month loan terms (Baseline: 60 months)
table(ld$Loan.Length)
```

```
##
##          . 36 months 60 months
##          1      1861       534
```

```
ld=ld %>%
  mutate(LL_36=as.numeric(Loan.Length=="36 months")) %>%
  select(-Loan.Length)

# # Target Encoding: Grouping states into 'Tiers' based on observed mean inte
rest rates
# This reduces the dimensionality of 'State'
# preventing model overfitting and addressing the high cardinality of geograp
hic data
table(ld$State)
```

```
##
##  AK  AL  AR  AZ  CA  CO  CT  DC  DE  FL  GA  HI  IA  IL  IN  KS  KY  LA  M
```

```
A  MD
##  11  36  13  45 414  56  47  11   8 161  96  11   1  99   3  20  23  20  7
0  66
##  MI  MN  MO  MS  MT  NC  NH  NJ  NM  NV  NY  OH  OK  OR  PA  RI  SC  SD  T
X  UT
##  43  37  33   1   7  60  15  89  11  31 243  68  21  28  93  15  24   4 17
0  16
##  VA  VT  WA  WI  WV  WY
##  75   5  54  26  12   4

round(tapply(ld$Interest.Rate, ld$State, mean), 0)

## AK AL AR AZ CA CO CT DC DE FL GA HI IA IL IN KS KY LA MA MD MI MN MO MS MT
NC
## 17 13 13 13 13 13 14 14 12 13 13 16 14 13 13 14 12 15 12 13 15 13 13 16 11
13
## NH NJ NM NV NY OH OK OR PA RI SC SD TX UT VA VT WA WI WV WY
## 12 13 13 14 13 12 14 13 13 13 13 10 13 13 13 18 13 14 14 13

# 1.Low_int_rates (<=12) - "SD", "MT", "DE", "KY","MA","NH", "OH"
# 2.High_int_rates (>=16) - "HI", "AK", "MS", "VT"
# 3.Medium_int_rates (13-15) - Remaining States

ld <- ld %>%
  mutate(State_Tier = case_when(
    State %in% c("SD", "MT", "DE", "KY","MA","NH", "OH") ~ "Low_Rate_State",
    State %in% c("HI", "AK", "MS", "VT") ~ "High_Rate_State",
    TRUE ~ "Mid_Rate_State" # Baseline category representing the majority of
observations
  ))

ld <- ld %>%
  mutate(
    State_High_Risk = as.numeric(State_Tier == "High_Rate_State"),
    State_Low_Risk  = as.numeric(State_Tier == "Low_Rate_State")
  ) %>%
  select(-State, -State_Tier)

glimpse(ld)

## Rows: 2,396
## Columns: 20
## $ ID                              <int> 81174, 99592, 80059, 15825, 33182,
6240~
## $ Amount.Requested                <dbl> 20000, 19200, 35000, 10000, 12000,
6000~
## $ Interest.Rate                   <dbl> 8.90, 12.12, 21.98, 9.99, 11.71, 15
.31,~
## $ Debt.To.Income.Ratio            <dbl> 14.90, 28.36, 23.81, 14.30, 18.78,
20.0~
```

```
## $ Monthly.Income                 <dbl> 6541.67, 4583.33, 11500.00, 3833.33
, 31~
## $ Open.CREDIT.Lines              <dbl> 14, 12, 14, 10, 11, 17, 10, 12, 9,
10, ~
## $ Revolving.CREDIT.Balance       <dbl> 14272, 11140, 21977, 9346, 14469, 1
0391~
## $ Inquiries.in.the.Last.6.Months <int> 2, 1, 1, 0, 0, 2, 0, 0, 1, 0, 0, 0,
0, ~
## $ fico                           <dbl> 737, 717, 692, 697, 697, 672, 722,
707,~
## $ el                             <dbl> 0, 2, 2, 5, 9, 3, 10, 10, 8, 6, 0,
1, 1~
## $ HW_RENT                        <dbl> 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0,
1, ~
## $ HW_MORT                        <dbl> 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1,
0, ~
## $ HW_OWN                         <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, ~
## $ LP_1                           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, ~
## $ LP_2                           <dbl> 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0,
0, ~
## $ LP_3                           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, ~
## $ LP_4                           <dbl> 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1,
1, ~
## $ LL_36                          <dbl> 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0,
1, ~
## $ State_High_Risk                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, ~
## $ State_Low_Risk                 <dbl> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0, ~
```

```
# Checking columns for number of NA
apply(ld,2,function(x) sum(is.na(x)))
```

```
##                             ID              Amount.Requested
##                              0                             0
##                  Interest.Rate          Debt.To.Income.Ratio
##                              0                             0
##                 Monthly.Income             Open.CREDIT.Lines
##                              0                             0
##       Revolving.CREDIT.Balance Inquiries.in.the.Last.6.Months
##                              0                             0
##                           fico                            el
##                              0                             0
##                       HW_RENT                       HW_MORT
##                              0                             0
##                        HW_OWN                          LP_1
##                              0                             0
```

```
##                            LP_2                            LP_3
##                               0                               0
##                            LP_4                           LL_36
##                               0                               0
##                State_High_Risk                State_Low_Risk
##                               0                               0
```

```
# 70/30 Train-Test Split to evaluate model performance
set.seed(2)
s=sample(1:nrow(ld),0.70*nrow(ld))
ld_train=ld[s,]
ld_test=ld[-s,]

glimpse(ld_train)
```

```
## Rows: 1,677
## Columns: 20
## $ ID                                <int> 34262, 97076, 56401, 16158, 83786,
9736~
## $ Amount.Requested                  <dbl> 12500, 30000, 10000, 10000, 6000, 2
0800~
## $ Interest.Rate                     <dbl> 15.96, 15.31, 14.65, 6.91, 15.80, 6
.62,~
## $ Debt.To.Income.Ratio              <dbl> 13.87, 15.42, 16.68, 8.83, 7.94, 15
.64,~
## $ Monthly.Income                    <dbl> 5300.00, 8333.33, 8333.33, 4166.67,
500~
## $ Open.CREDIT.Lines                 <dbl> 7, 21, 18, 5, 7, 11, 9, 9, 7, 7, 11
, 3,~
## $ Revolving.CREDIT.Balance          <dbl> 11439, 29060, 39461, 6381, 14784, 1
5192~
## $ Inquiries.in.the.Last.6.Months <int> 3, 1, 2, 0, 1, 0, 1, 0, 0, 1, 1, 0,
1, ~
## $ fico                              <dbl> 697, 682, 682, 732, 672, 712, 682,
697,~
## $ el                                <dbl> 5, 8, 4, 1, 10, 10, 3, 10, 9, 7, 0,
0, ~
## $ HW_RENT                           <dbl> 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
1, ~
## $ HW_MORT                           <dbl> 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1,
0, ~
## $ HW_OWN                            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, ~
## $ LP_1                              <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
1, ~
## $ LP_2                              <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
0, ~
## $ LP_3                              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, ~
## $ LP_4                              <dbl> 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0,
```

```
0, ~
## $ LL_36                          <dbl> 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
0, ~
## $ State_High_Risk                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, ~
## $ State_Low_Risk                 <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, ~
```

```
# Computing correlation between different variables
View(cor(ld_train))

## 1. LINEAR REGRESSION
# Initial Fit: Including all features except ID
fit= lm(Interest.Rate ~ .-ID ,data=ld_train)
summary(fit)
```

```
##
## Call:
## lm(formula = Interest.Rate ~ . - ID, data = ld_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3087 -1.3517 -0.2165  1.2315 10.0301
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.541e+01  1.493e+00  50.512  < 2e-16 ***
## Amount.Requested              1.509e-04  8.269e-06  18.249  < 2e-16 ***
## Debt.To.Income.Ratio          4.294e-03  7.783e-03   0.552  0.58117
## Monthly.Income               -4.304e-06  1.502e-05  -0.286  0.77453
## Open.CREDIT.Lines            -3.335e-02  1.272e-02  -2.621  0.00884 **
## Revolving.CREDIT.Balance     -5.794e-06  3.084e-06  -1.879  0.06046 .
## Inquiries.in.the.Last.6.Months  3.761e-01  4.013e-02   9.372  < 2e-16 ***
## fico                         -8.663e-02  1.565e-03 -55.353  < 2e-16 ***
## el                            1.052e-02  1.466e-02   0.718  0.47308
## HW_RENT                      -2.152e-01  9.313e-01  -0.231  0.81727
## HW_MORT                      -5.204e-01  9.315e-01  -0.559  0.57643
## HW_OWN                       -4.650e-03  9.470e-01  -0.005  0.99608
## LP_1                          2.802e-01  5.032e-01   0.557  0.57774
## LP_2                          9.096e-03  4.706e-01   0.019  0.98458
## LP_3                         -1.143e-02  4.954e-01  -0.023  0.98160
## LP_4                         -1.372e-01  4.676e-01  -0.293  0.76922
## LL_36                        -3.277e+00  1.363e-01 -24.044  < 2e-16 ***
## State_High_Risk               1.353e+00  4.890e-01   2.767  0.00572 **
## State_Low_Risk                3.514e-02  1.861e-01   0.189  0.85026
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.046 on 1658 degrees of freedom
```

```
## Multiple R-squared:  0.7543, Adjusted R-squared:  0.7517
## F-statistic: 282.8 on 18 and 1658 DF,  p-value: < 2.2e-16

# Multicollinearity Check: Using Variance Inflation Factor (VIF) to detect re
dundant features
library(car)

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

t=vif(fit)
sort(t,decreasing = TRUE)

##                        HW_MORT                        HW_RENT
##                      86.474039                      86.194218
##                         HW_OWN                           LP_4
##                      24.779806                      21.646244
##                           LP_2                           LP_3
##                      18.302728                       7.325876
##                           LP_1                 Amount.Requested
##                       5.952540                       1.671723
##                  Monthly.Income            Debt.To.Income.Ratio
##                       1.486951                       1.348001
##       Revolving.CREDIT.Balance               Open.CREDIT.Lines
##                       1.343930                       1.337674
##                          LL_36                           fico
##                       1.279111                       1.147585
##                             el Inquiries.in.the.Last.6.Months
##                       1.107790                       1.029891
##                  State_Low_Risk                 State_High_Risk
##                       1.019819                       1.017042

# Refining model by iteratively removing high-VIF variables
fit=lm(Interest.Rate~. -ID - HW_MORT,data=ld_train)
t=vif(fit)
sort(t,decreasing = TRUE)

##                           LP_4                           LP_2
##                      21.433890                      18.139616
##                           LP_3                           LP_1
##                       7.256322                       5.901296
##                 Amount.Requested                  Monthly.Income
##                       1.670676                       1.486478
##            Debt.To.Income.Ratio       Revolving.CREDIT.Balance
##                       1.346352                       1.342036
##                Open.CREDIT.Lines                          LL_36
```

```
##                        1.335368                        1.279058
##                         HW_RENT                            fico
##                        1.277164                        1.144577
##                              el                          HW_OWN
##                        1.107435                        1.104927
## Inquiries.in.the.Last.6.Months                 State_Low_Risk
##                        1.029816                        1.019727
##                State_High_Risk
##                        1.006857
```

```
fit=lm(Interest.Rate~. -ID - HW_MORT-LP_4,data=ld_train)
t=vif(fit)
sort(t,decreasing = TRUE)
```

```
##               Amount.Requested                  Monthly.Income
##                        1.664167                        1.486476
##           Debt.To.Income.Ratio       Revolving.CREDIT.Balance
##                        1.344660                        1.341985
##              Open.CREDIT.Lines                           LL_36
##                        1.335326                        1.279037
##                         HW_RENT                            fico
##                        1.276318                        1.142812
##                            LP_1                            LP_3
##                        1.140824                        1.114940
##                              el                          HW_OWN
##                        1.107253                        1.104576
##                            LP_2 Inquiries.in.the.Last.6.Months
##                        1.102839                        1.029545
##                 State_Low_Risk                State_High_Risk
##                        1.019427                        1.006703
```

```
summary(fit)
```

```
##
## Call:
## lm(formula = Interest.Rate ~ . - ID - HW_MORT - LP_4, data = ld_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3129 -1.3541 -0.2141  1.2317 10.0265
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.478e+01  1.147e+00  65.183  < 2e-16 ***
## Amount.Requested              1.506e-04  8.247e-06  18.263  < 2e-16 ***
## Debt.To.Income.Ratio          4.046e-03  7.769e-03   0.521  0.60263
## Monthly.Income               -4.460e-06  1.501e-05  -0.297  0.76646
## Open.CREDIT.Lines            -3.308e-02  1.271e-02  -2.604  0.00930 **
## Revolving.CREDIT.Balance     -5.723e-06  3.080e-06  -1.858  0.06338 .
## Inquiries.in.the.Last.6.Months  3.761e-01  4.011e-02   9.377  < 2e-16 ***
## fico                         -8.666e-02  1.561e-03 -55.509  < 2e-16 ***
```

```
## el                                1.044e-02  1.465e-02   0.713  0.47617
## HW_RENT                           3.022e-01  1.133e-01   2.668  0.00769 **
## HW_OWN                            5.113e-01  1.998e-01   2.558  0.01061 *
## LP_1                              4.118e-01  2.202e-01   1.870  0.06161 .
## LP_2                              1.434e-01  1.155e-01   1.242  0.21437
## LP_3                              1.206e-01  1.932e-01   0.624  0.53254
## LL_36                            -3.277e+00  1.362e-01 -24.058  < 2e-16 ***
## State_High_Risk                   1.378e+00  4.863e-01   2.834  0.00465 **
## State_Low_Risk                    3.303e-02  1.860e-01   0.178  0.85904
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.045 on 1660 degrees of freedom
## Multiple R-squared:  0.7543, Adjusted R-squared:  0.7519
## F-statistic: 318.5 on 16 and 1660 DF,  p-value: < 2.2e-16

# Stepwise Regression by removing non-significant variables (p > 0.05)
fit=lm(Interest.Rate~. -ID - HW_MORT-LP_4-State_Low_Risk,data=ld_train)
summary(fit)

##
## Call:
## lm(formula = Interest.Rate ~ . - ID - HW_MORT - LP_4 - State_Low_Risk,
##     data = ld_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3152 -1.3555 -0.2167  1.2290 10.0218
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.476e+01  1.144e+00  65.359  < 2e-16 ***
## Amount.Requested              1.506e-04  8.244e-06  18.272  < 2e-16 ***
## Debt.To.Income.Ratio          4.072e-03  7.766e-03   0.524  0.60009
## Monthly.Income               -4.561e-06  1.500e-05  -0.304  0.76110
## Open.CREDIT.Lines            -3.311e-02  1.270e-02  -2.607  0.00921 **
## Revolving.CREDIT.Balance     -5.730e-06  3.079e-06  -1.861  0.06291 .
## Inquiries.in.the.Last.6.Months  3.762e-01  4.009e-02   9.384  < 2e-16 ***
## fico                         -8.663e-02  1.554e-03 -55.735  < 2e-16 ***
## el                            1.052e-02  1.464e-02   0.719  0.47244
## HW_RENT                       3.012e-01  1.131e-01   2.664  0.00781 **
## HW_OWN                        5.098e-01  1.996e-01   2.554  0.01074 *
## LP_1                          4.102e-01  2.199e-01   1.865  0.06235 .
## LP_2                          1.430e-01  1.154e-01   1.239  0.21555
## LP_3                          1.205e-01  1.931e-01   0.624  0.53280
## LL_36                        -3.276e+00  1.361e-01 -24.066  < 2e-16 ***
## State_High_Risk               1.376e+00  4.860e-01   2.831  0.00469 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.045 on 1661 degrees of freedom
## Multiple R-squared:  0.7543, Adjusted R-squared:  0.7521
## F-statistic: 339.9 on 15 and 1661 DF,  p-value: < 2.2e-16

fit=lm(Interest.Rate~. -ID - HW_MORT-LP_4-State_Low_Risk-Monthly.Income,data=
ld_train)
summary(fit)

##
## Call:
## lm(formula = Interest.Rate ~ . - ID - HW_MORT - LP_4 - State_Low_Risk -
##     Monthly.Income, data = ld_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3210 -1.3596 -0.2188  1.2342 10.0234
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.476e+01  1.144e+00  65.378  < 2e-16 ***
## Amount.Requested              1.498e-04  7.791e-06  19.229  < 2e-16 ***
## Debt.To.Income.Ratio          4.710e-03  7.475e-03   0.630  0.52872
## Open.CREDIT.Lines            -3.357e-02  1.261e-02  -2.663  0.00782 **
## Revolving.CREDIT.Balance     -6.005e-06  2.943e-06  -2.040  0.04147 *
## Inquiries.in.the.Last.6.Months  3.759e-01  4.007e-02   9.382  < 2e-16 ***
## fico                         -8.665e-02  1.553e-03 -55.785  < 2e-16 ***
## el                            1.041e-02  1.463e-02   0.712  0.47677
## HW_RENT                       3.042e-01  1.126e-01   2.702  0.00697 **
## HW_OWN                        5.148e-01  1.989e-01   2.588  0.00973 **
## LP_1                          4.073e-01  2.197e-01   1.854  0.06389 .
## LP_2                          1.436e-01  1.154e-01   1.245  0.21329
## LP_3                          1.177e-01  1.929e-01   0.611  0.54161
## LL_36                        -3.281e+00  1.353e-01 -24.257  < 2e-16 ***
## State_High_Risk               1.381e+00  4.855e-01   2.844  0.00450 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.044 on 1662 degrees of freedom
## Multiple R-squared:  0.7543, Adjusted R-squared:  0.7522
## F-statistic: 364.4 on 14 and 1662 DF,  p-value: < 2.2e-16

fit=lm(Interest.Rate~. -ID - HW_MORT-LP_4-State_Low_Risk-Monthly.Income-LP_3,
data=ld_train)
summary(fit)

##
## Call:
## lm(formula = Interest.Rate ~ . - ID - HW_MORT - LP_4 - State_Low_Risk -
##     Monthly.Income - LP_3, data = ld_train)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -9.2308 -1.3702 -0.2171  1.2364 10.0149
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.472e+01  1.142e+00  65.449  < 2e-16 ***
## Amount.Requested              1.495e-04  7.777e-06  19.228  < 2e-16 ***
## Debt.To.Income.Ratio          4.191e-03  7.425e-03   0.564  0.57254
## Open.CREDIT.Lines            -3.382e-02  1.260e-02  -2.685  0.00733 **
## Revolving.CREDIT.Balance     -5.964e-06  2.942e-06  -2.027  0.04279 *
## Inquiries.in.the.Last.6.Months  3.765e-01  4.005e-02   9.401  < 2e-16 ***
## fico                         -8.655e-02  1.544e-03 -56.039  < 2e-16 ***
## el                            1.030e-02  1.463e-02   0.704  0.48155
## HW_RENT                       2.971e-01  1.120e-01   2.653  0.00805 **
## HW_OWN                        5.185e-01  1.988e-01   2.609  0.00917 **
## LP_1                          3.864e-01  2.169e-01   1.781  0.07507 .
## LP_2                          1.289e-01  1.128e-01   1.143  0.25315
## LL_36                        -3.283e+00  1.352e-01 -24.289  < 2e-16 ***
## State_High_Risk               1.378e+00  4.854e-01   2.839  0.00458 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.044 on 1663 degrees of freedom
## Multiple R-squared:  0.7542, Adjusted R-squared:  0.7523
## F-statistic: 392.5 on 13 and 1663 DF,  p-value: < 2.2e-16
```

```
fit=lm(Interest.Rate~. -ID - HW_MORT-LP_4-State_Low_Risk-Monthly.Income-LP_3-
Debt.To.Income.Ratio,data=ld_train)
summary(fit)
```

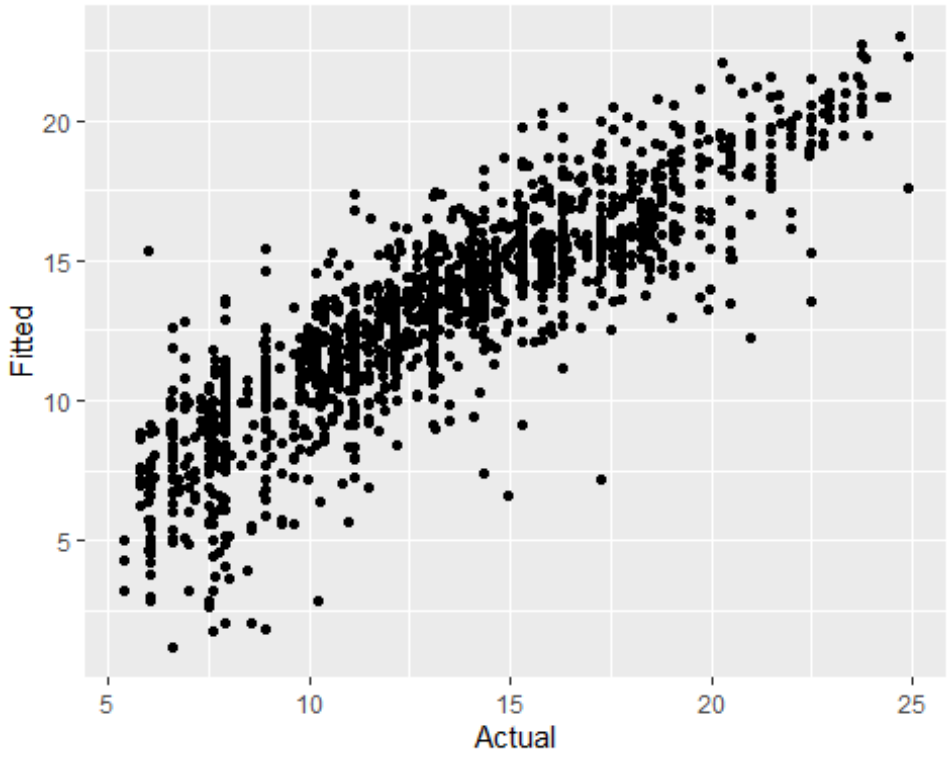
```
##
## Call:
## lm(formula = Interest.Rate ~ . - ID - HW_MORT - LP_4 - State_Low_Risk -
##     Monthly.Income - LP_3 - Debt.To.Income.Ratio, data = ld_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.2698 -1.3582 -0.2044  1.2461 10.0357
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.486e+01  1.114e+00  67.172  < 2e-16 ***
## Amount.Requested              1.495e-04  7.775e-06  19.227  < 2e-16 ***
## Open.CREDIT.Lines            -3.150e-02  1.191e-02  -2.646  0.00823 **
## Revolving.CREDIT.Balance     -5.818e-06  2.930e-06  -1.986  0.04721 *
## Inquiries.in.the.Last.6.Months  3.758e-01  4.002e-02   9.389  < 2e-16 ***
## fico                         -8.669e-02  1.523e-03 -56.903  < 2e-16 ***
## el                            1.043e-02  1.462e-02   0.713  0.47568
## HW_RENT                       3.010e-01  1.117e-01   2.694  0.00713 **
## HW_OWN                        5.254e-01  1.983e-01   2.649  0.00815 **
```

```
## LP_1                             3.776e-01  2.163e-01   1.745  0.08112 .
## LP_2                             1.295e-01  1.128e-01   1.148  0.25110
## LL_36                           -3.285e+00  1.351e-01 -24.305  < 2e-16 ***
## State_High_Risk                  1.376e+00  4.853e-01   2.836  0.00462 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.043 on 1664 degrees of freedom
## Multiple R-squared:  0.7542, Adjusted R-squared:  0.7524
## F-statistic: 425.4 on 12 and 1664 DF,  p-value: < 2.2e-16

fit=lm(Interest.Rate~. -ID - HW_MORT-LP_4-State_Low_Risk-Monthly.Income-LP_3-
Debt.To.Income.Ratio-el,data=ld_train)
summary(fit)

##
## Call:
## lm(formula = Interest.Rate ~ . - ID - HW_MORT - LP_4 - State_Low_Risk -
##     Monthly.Income - LP_3 - Debt.To.Income.Ratio - el, data = ld_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3039 -1.3514 -0.2122  1.2286 10.0292
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    7.495e+01  1.107e+00  67.714  < 2e-16 ***
## Amount.Requested               1.499e-04  7.749e-06  19.350  < 2e-16 ***
## Open.CREDIT.Lines             -3.149e-02  1.191e-02  -2.645  0.00824 **
## Revolving.CREDIT.Balance      -5.722e-06  2.926e-06  -1.955  0.05070 .
## Inquiries.in.the.Last.6.Months  3.753e-01  4.001e-02   9.380  < 2e-16 ***
## fico                          -8.673e-02  1.522e-03 -56.981  < 2e-16 ***
## HW_RENT                        2.819e-01  1.085e-01   2.599  0.00943 **
## HW_OWN                         5.151e-01  1.978e-01   2.604  0.00929 **
## LP_1                           3.687e-01  2.159e-01   1.708  0.08791 .
## LP_2                           1.293e-01  1.128e-01   1.147  0.25167
## LL_36                         -3.287e+00  1.351e-01 -24.338  < 2e-16 ***
## State_High_Risk                1.376e+00  4.852e-01   2.836  0.00463 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.043 on 1665 degrees of freedom
## Multiple R-squared:  0.7541, Adjusted R-squared:  0.7525
## F-statistic: 464.1 on 11 and 1665 DF,  p-value: < 2.2e-16

fit=lm(Interest.Rate~. -ID - HW_MORT-LP_4-State_Low_Risk-Monthly.Income-LP_3-
Debt.To.Income.Ratio-el-LP_2,data=ld_train)
summary(fit)

##
## Call:
```

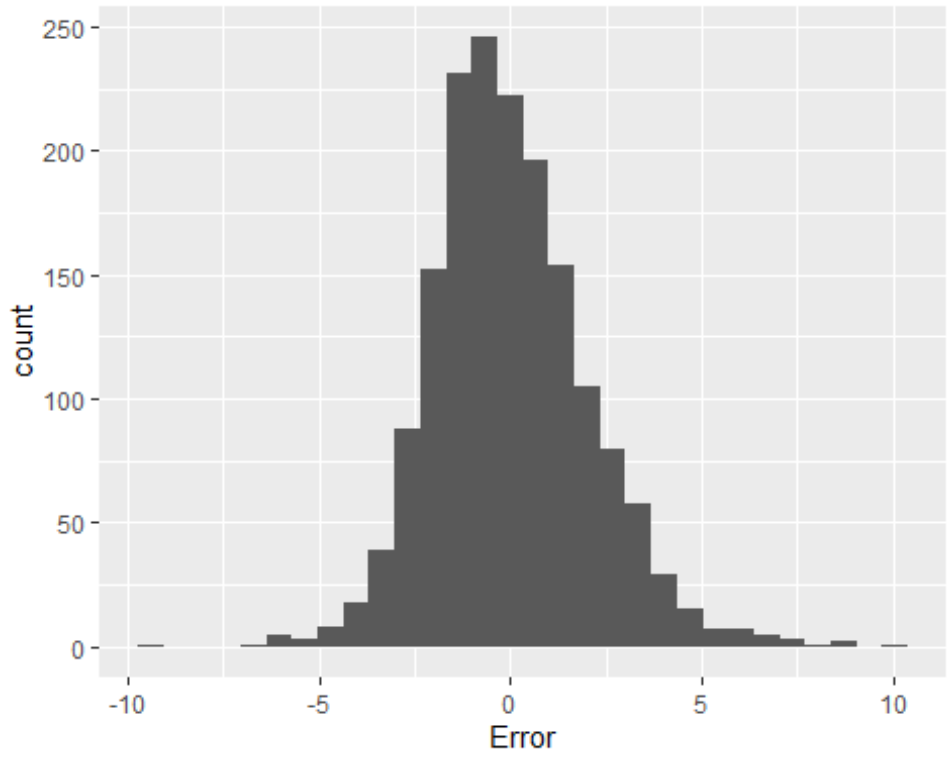
```
## lm(formula = Interest.Rate ~ . - ID - HW_MORT - LP_4 - State_Low_Risk -
##     Monthly.Income - LP_3 - Debt.To.Income.Ratio - el - LP_2,
##     data = ld_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.341 -1.379 -0.205  1.218 10.117
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.501e+01  1.106e+00  67.812  < 2e-16 ***
## Amount.Requested              1.490e-04  7.706e-06  19.335  < 2e-16 ***
## Open.CREDIT.Lines            -3.155e-02  1.191e-02  -2.650  0.00814 **
## Revolving.CREDIT.Balance     -5.499e-06  2.920e-06  -1.883  0.05984 .
## Inquiries.in.the.Last.6.Months  3.744e-01  4.001e-02   9.359  < 2e-16 ***
## fico                         -8.674e-02  1.522e-03 -56.984  < 2e-16 ***
## HW_RENT                       2.867e-01  1.084e-01   2.645  0.00824 **
## HW_OWN                        5.219e-01  1.977e-01   2.640  0.00837 **
## LP_1                          3.253e-01  2.126e-01   1.530  0.12624
## LL_36                        -3.283e+00  1.350e-01 -24.312  < 2e-16 ***
## State_High_Risk               1.362e+00  4.851e-01   2.807  0.00506 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.043 on 1666 degrees of freedom
## Multiple R-squared:  0.7539, Adjusted R-squared:  0.7524
## F-statistic: 510.3 on 10 and 1666 DF,  p-value: < 2.2e-16

fit=lm(Interest.Rate~. -ID - HW_MORT-LP_4-State_Low_Risk-Monthly.Income-LP_3-
Debt.To.Income.Ratio-el-LP_2-LP_1,data=ld_train)
summary(fit)

##
## Call:
## lm(formula = Interest.Rate ~ . - ID - HW_MORT - LP_4 - State_Low_Risk -
##     Monthly.Income - LP_3 - Debt.To.Income.Ratio - el - LP_2 -
##     LP_1, data = ld_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3814 -1.3787 -0.2016  1.2103 10.0657
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.478e+01  1.096e+00  68.198  < 2e-16 ***
## Amount.Requested              1.473e-04  7.627e-06  19.311  < 2e-16 ***
## Open.CREDIT.Lines            -3.238e-02  1.190e-02  -2.721  0.00657 **
## Revolving.CREDIT.Balance     -5.596e-06  2.920e-06  -1.916  0.05553 .
## Inquiries.in.the.Last.6.Months  3.745e-01  4.002e-02   9.358  < 2e-16 ***
## fico                         -8.634e-02  1.500e-03 -57.574  < 2e-16 ***
```

```
## HW_RENT                           2.874e-01  1.084e-01   2.650  0.00812 **
## HW_OWN                            5.316e-01  1.977e-01   2.689  0.00723 **
## LL_36                            -3.293e+00  1.349e-01 -24.417  < 2e-16 ***
## State_High_Risk                   1.366e+00  4.853e-01   2.816  0.00493 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.044 on 1667 degrees of freedom
## Multiple R-squared:  0.7535, Adjusted R-squared:  0.7522
## F-statistic: 566.3 on 9 and 1667 DF,  p-value: < 2.2e-16

#Final model
fit_train=lm(Interest.Rate~. -ID - HW_MORT-LP_4-State_Low_Risk-Monthly.Income
-LP_3-Debt.To.Income.Ratio-el-LP_2-LP_1,data=ld_train)
summary(fit)

##
## Call:
## lm(formula = Interest.Rate ~ . - ID - HW_MORT - LP_4 - State_Low_Risk -
##     Monthly.Income - LP_3 - Debt.To.Income.Ratio - el - LP_2 -
##     LP_1, data = ld_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3814 -1.3787 -0.2016  1.2103 10.0657
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.478e+01  1.096e+00  68.198  < 2e-16 ***
## Amount.Requested              1.473e-04  7.627e-06  19.311  < 2e-16 ***
## Open.CREDIT.Lines            -3.238e-02  1.190e-02  -2.721  0.00657 **
## Revolving.CREDIT.Balance     -5.596e-06  2.920e-06  -1.916  0.05553 .
## Inquiries.in.the.Last.6.Months  3.745e-01  4.002e-02   9.358  < 2e-16 ***
## fico                         -8.634e-02  1.500e-03 -57.574  < 2e-16 ***
## HW_RENT                       2.874e-01  1.084e-01   2.650  0.00812 **
## HW_OWN                        5.316e-01  1.977e-01   2.689  0.00723 **
## LL_36                        -3.293e+00  1.349e-01 -24.417  < 2e-16 ***
## State_High_Risk               1.366e+00  4.853e-01   2.816  0.00493 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.044 on 1667 degrees of freedom
## Multiple R-squared:  0.7535, Adjusted R-squared:  0.7522
## F-statistic: 566.3 on 9 and 1667 DF,  p-value: < 2.2e-16

# Model : Pred_Interest_Rate =74.78 + Amount.Requested*1.473*10^-4 + Open.Cre
dit.Lines*(-3.238*10^-2) + Revolving.Credit.Balance*(-5.596*10^-6) + Inquirie
s.in.the.Last.6.Months*3.745*10^-1 + fico*(-8.634*10^-2) + HW_RENT*2.874*10^-
1 + HW_OWN*5.316*10^-1 + LL_36*(-3.293) + State_High_Risk*1.366
```

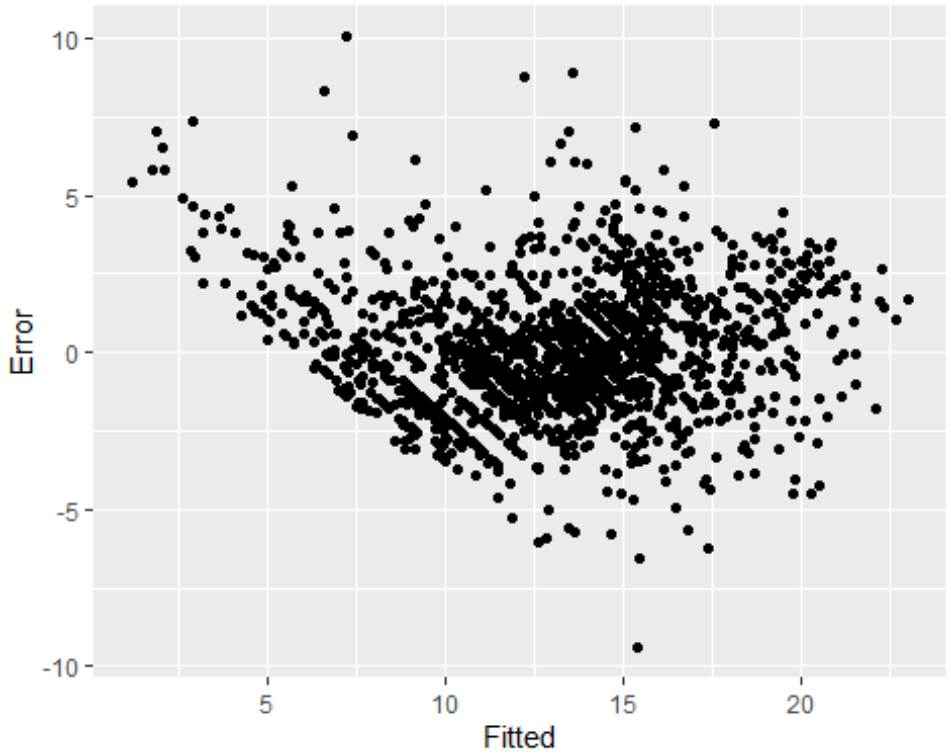
```
train_res = cbind.data.frame(Actual=ld_train$Interest.Rate, Fitted=fitted(fit
_train), Error=residuals(fit_train))
View(train_res)

rmse_train=sqrt(mean(train_res$Error^2))
rmse_train

## [1] 2.037848

# Testing Linear Regression Assumptions on train data

# Actual vs predicted
library(ggplot2)
ggplot(train_res,aes(x=Actual,y=Fitted))+geom_point()
```



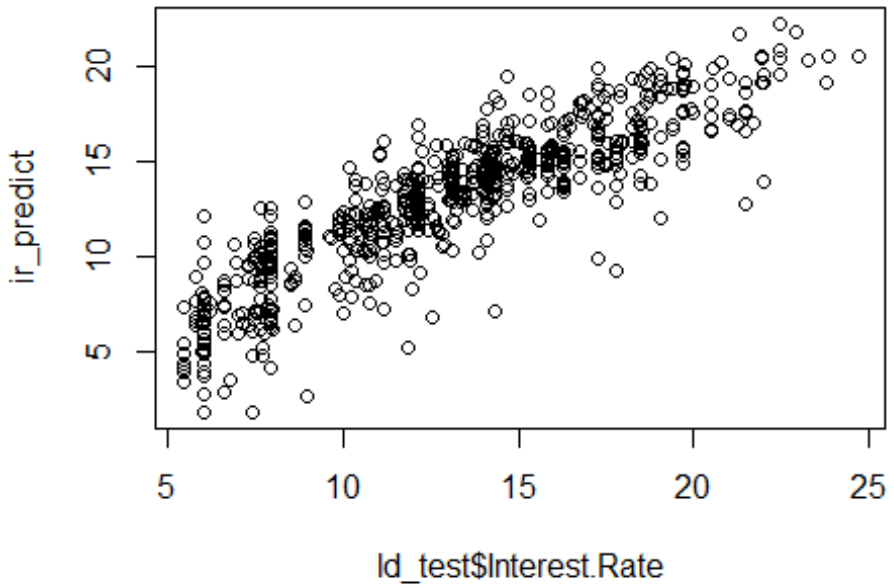

```
# Error~N(0,sigma2)
ggplot(train_res,aes(Error))+geom_histogram()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



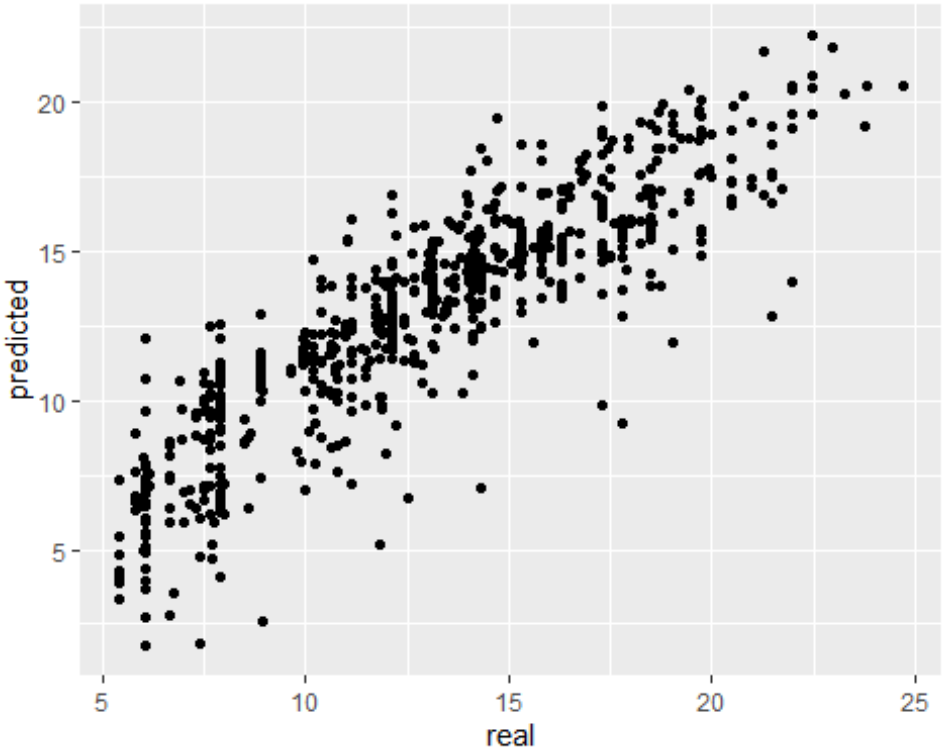

```
# Predicted vs error - Homoscedasticity/Independence of errors
ggplot(train_res,aes(x=Fitted,y=Error))+geom_point()
```



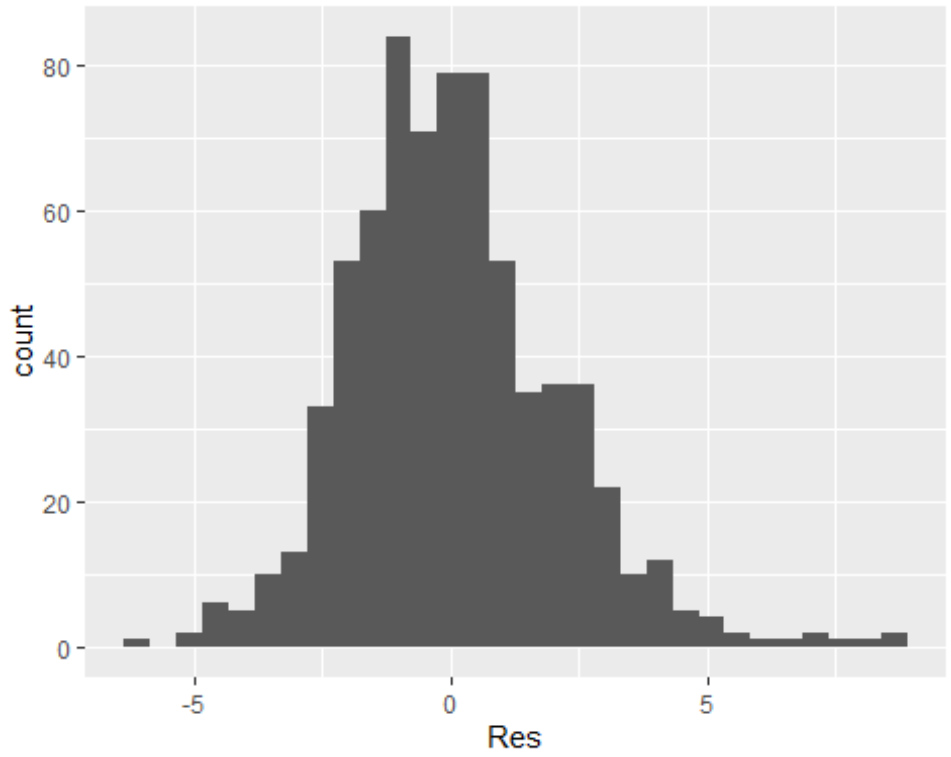
```
# Prediction on the Test data
ir_predict=predict(fit_train,newdata=ld_test)
TestRes = cbind.data.frame(Act=ld_test$Interest.Rate, Pred=ir_predict)
View(TestRes)

plot(ld_test$Interest.Rate,ir_predict)
```



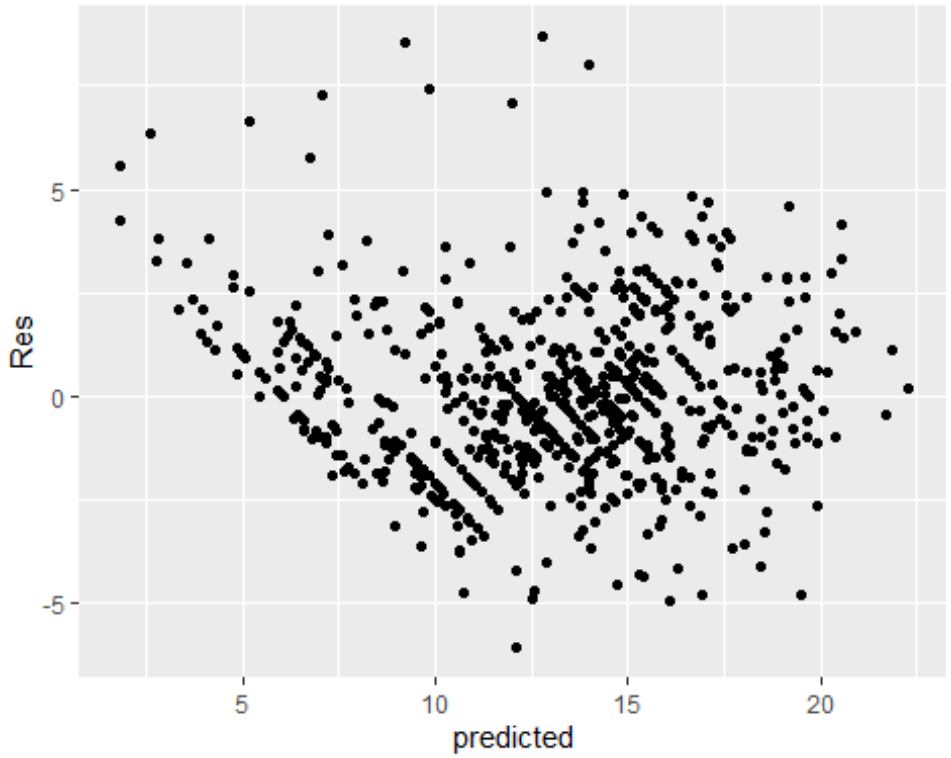

```
res=ld_test$Interest.Rate-ir_predict

# Evaluating Performance (RMSE)
rmse_test=sqrt(mean(res^2))
rmse_test

## [1] 2.041869

# Testing Linear Regression Assumptions on test data

d=data.frame(real=ld_test$Interest.Rate,predicted=ir_predict, Res=ld_test$Int
erest.Rate-ir_predict)

# Actual vs predicted
ggplot(d,aes(x=real,y=predicted))+geom_point()
```



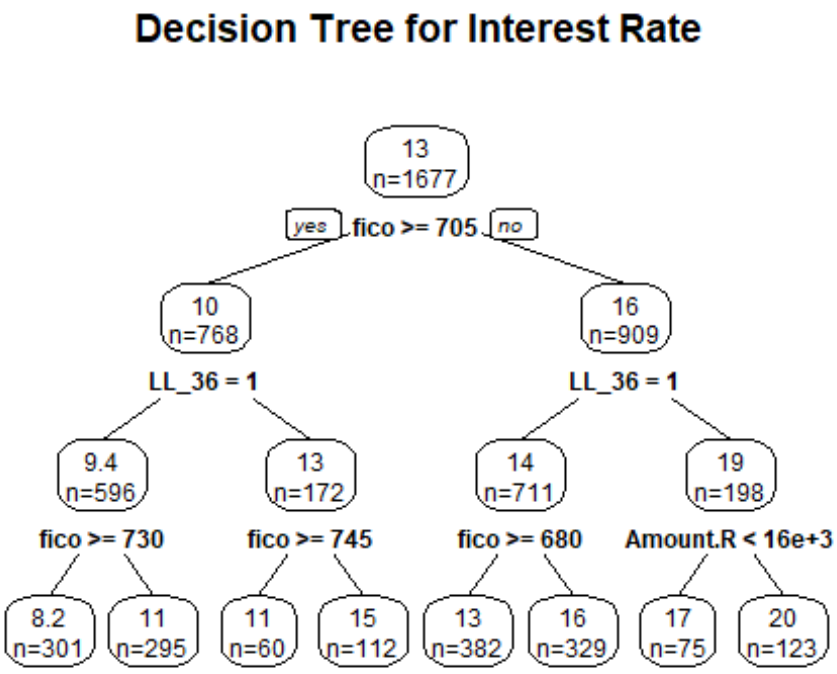

```
# Error~N(0,sigma2)
ggplot(d,aes(Res))+geom_histogram()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Predicted vs error - Homoscedasticity/Independence of errors
ggplot(d,aes(x=predicted,y=Res))+geom_point()
```

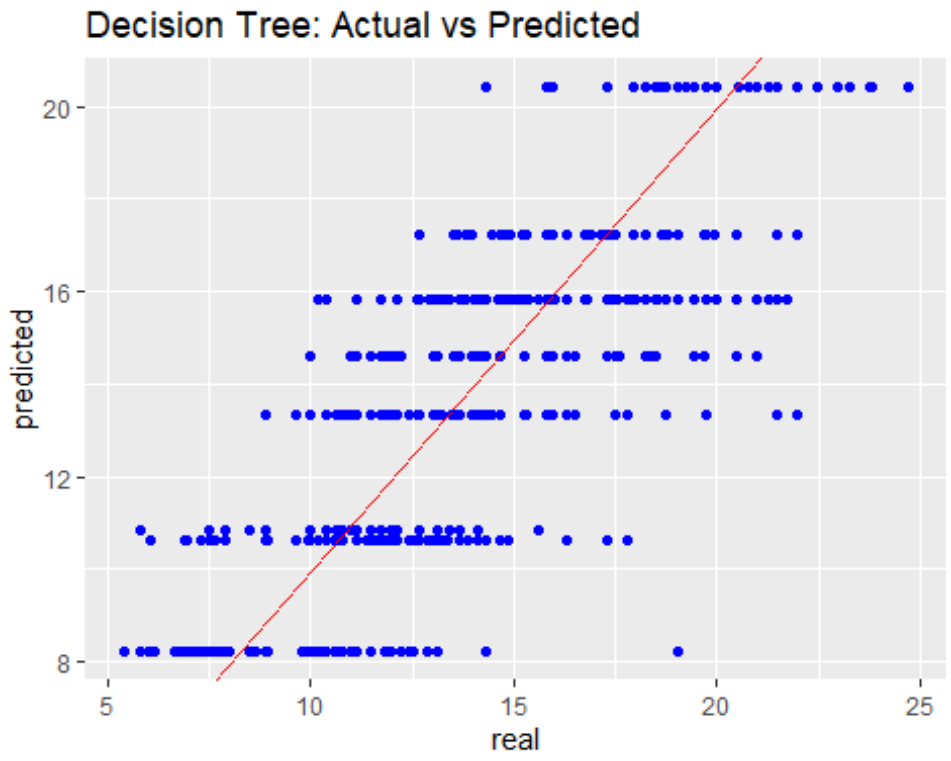


```r
## 2. DECISION TREE

library(rpart)
library(rpart.plot)

# Fitting the Decision tree model

tree_fit = rpart(Interest.Rate ~ . -ID, data = ld_train)

prp(tree_fit,
    type = 2,
    extra = 1,
    main = "Decision Tree for Interest Rate")
```




```r
# Prediction on the Test data
tree_pred = predict(tree_fit, newdata = ld_test)

# Evaluating Performance (RMSE)
tree_res = ld_test$Interest.Rate - tree_pred
rmse_tree = sqrt(mean(tree_res^2))

print(paste("Decision Tree RMSE:", rmse_tree))

## [1] "Decision Tree RMSE: 2.35114218805736"

# Visualizing Actual vs Predicted for Tree
d_tree = data.frame(real = ld_test$Interest.Rate, predicted = tree_pred)
```

```r
ggplot(d_tree, aes(x = real, y = predicted)) +
  geom_point(color = "blue") +
  geom_abline(color = "red") +
  ggtitle("Decision Tree: Actual vs Predicted")
```




```r
## 3. RANDOM FOREST
library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine

# Fitting the Random Forest model

set.seed(2)
rf_fit = randomForest(Interest.Rate ~ . -ID,
                      data = ld_train,
```

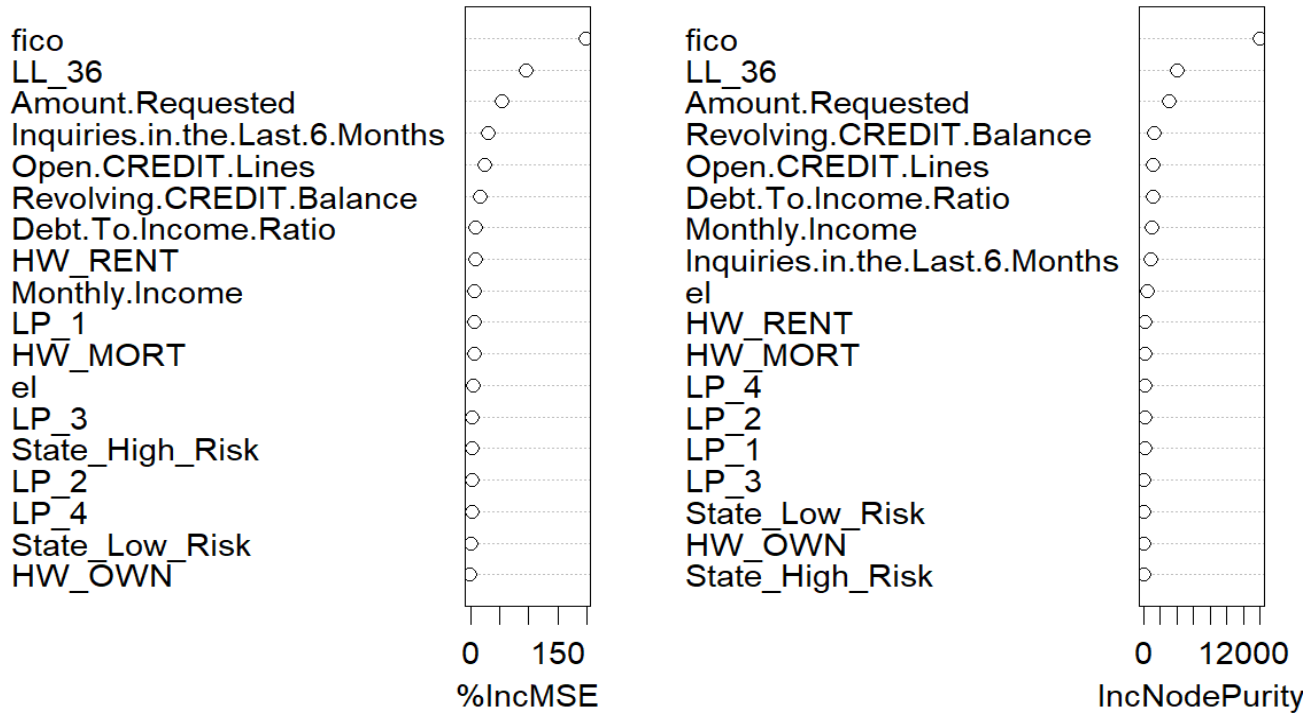
```
                      ntree = 500,
                      importance = TRUE)

print(rf_fit)
```

```
##
## Call:
##  randomForest(formula = Interest.Rate ~ . - ID, data = ld_train,      ntre
e = 500, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 3.371955
##                     % Var explained: 79.99
```

```
# Prediction on the Test Data
rf_pred = predict(rf_fit, newdata = ld_test)

# Evaluating Performance (RMSE)
rf_res = ld_test$Interest.Rate - rf_pred
rmse_rf = sqrt(mean(rf_res^2))

print(paste("Random Forest RMSE:", rmse_rf))
```

```
## [1] "Random Forest RMSE: 1.87248349024431"
```

```
# Variable Importance Plot
# This shows which variables were most useful in predicting Interest Rate
varImpPlot(rf_fit, main = "Random Forest Variable Importance")
```




```
## FINAL MODEL COMPARISON

# Calculating Test RMSE (Root Mean Squared Error) to determine predictive acc
uracy

comparison = data.frame(
  Model = c("Linear Regression", "Decision Tree", "Random Forest"),
  RMSE = c(rmse_test, rmse_tree, rmse_rf)
)

print(comparison)

##               Model     RMSE
## 1 Linear Regression 2.041869
## 2     Decision Tree 2.351142
## 3     Random Forest 1.872483

# Selecting the model with the lowest error
best_model = comparison[which.min(comparison$RMSE), ]
```

```
print(paste("The best model is", best_model$Model, "with RMSE:", round(best_m
odel$RMSE, 4)))
```

```
## [1] "The best model is Random Forest with RMSE: 1.8725"
```