# Tarea de Clasificación del Dataset: Home Loan Predictions

Jessica Borja[1 [0000−1111−2222−3333]], Richard Camacho[1 [1111−2222−3333−4444]], Amarilis Cueva[1 [2222−−3333−4444−5555]], Edwin Hernández[1 [0000−0002−2171−8893]], and Jefry Navas[1 [2222−−3333−4444−5555]]

Universidad Politécnica Salesiana, Rumichaca y Morán Valverde, Quito -Ecuador
https://www.ups.edu.ec/computacion-quito

**Abstract.** El siguiente informe presenta el análisis del modelo de clasificación sobre una empresa que provee préstamos para compra de casas donde se buscó lograr un modelo que prediga la salida de si el préstamo es aceptado o no, se realizó un análisis completo de varios aspectos, partiendo desde gráficas, correlaciones y análisis factoriales que ayudaron en la realización de los cálculos en la creación del modelo de regresión logística , siendo estos datos que nos mostraron la posibilidad de eliminar ciertas variables con baja correlación , y de igual manera se eliminaron variables que no aportaban mucho al modelo mediante técnicas de análisis factorial, siempre tratando de encontrar el modelo con la mejor accuracy , es decir que nos de la predicción más confiable y los resultados sean lo más cercano a la realidad. Además del puro análisis estadístico posteriormente se comparó con modelo basado en lógica difusa, determinando la similitud en el análisis de los datos.

**Keywords:** Clasificación · Préstamos · Predicción, Fuzzy Logic.

## 1 Introducción

El dataset trabajado contiene datos de la empresa Housing Finance es una empresa que ofrece préstamos hipotecarios para las casas que estaban presentes en todas las áreas urbanas, semiurbanas y rurales, la misma que valida la elegibilidad del préstamo después de que el cliente solicita el préstamo.La recopilación de la información obtenida de estos datos se puede utilizar para crear algoritmos de predicción a la situación presentada. Para esto con los datos recopilados se realizó un análisis de las variables a utilizar y varios procesos importantes entre ellos la curación del dataset para obtener un modelo de regresión logístico.[Pitarque et al., 1998]

### 1.1 Modelo de Clasificación

El Modelo de Clasificación más sencillo de todos es el lineal simple. Se utiliza para el análisis de textos y consiste en establecer a priori una serie de palabras positivas y negativas para luego contarlas sobre cada texto que se vaya a analizar, dando resultado positivo si son más las palabras positivas que las negativas, y viceversa.[Batanero, 2001]

### 1.2 Regresión Logística

La regresión logística es un instrumento estadístico de análisis multivariado, de uso tanto explicativo como predictivo. Resulta útil su empleo cuando se tiene una variable dependiente dicotómica (un atributo cuya ausencia o presencia hemos puntuado con los valores cero y uno, respectivamente) y un conjunto de variables predictoras o independientes.La cual nos permite determinar que variables pesan más para aumentar o disminuir la probabilidad de que a alguien le suceda el evento en cuestión.[Chitarroni, 2002]

### 1.3 SISTEMAS EXPERTOS

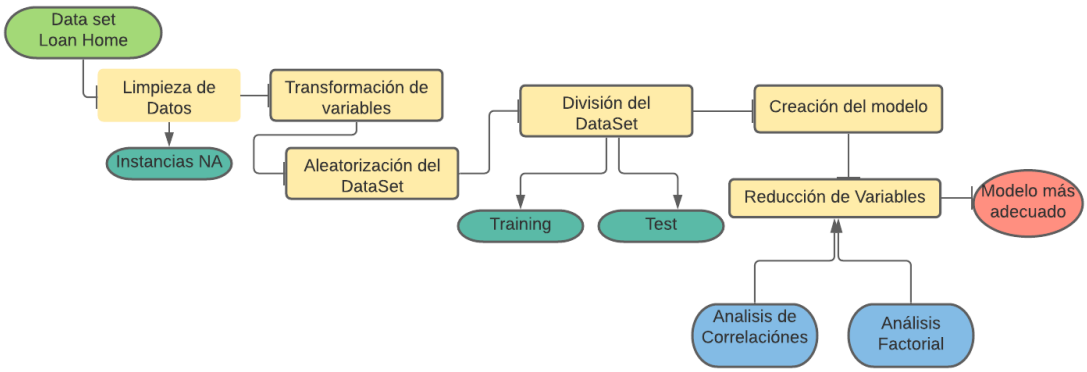
Mediante el estudio realizado el cual se baso principalmente en cálculos matemáticos, tras crear varios modelos se decidió utilizar el último modelo creado tras contener las variables necesarias el mismo que se construyo un modelo difuso, basado en reglas de "Mamdani" ya que tiene una estructura muy simple de operaciones "mín-max".

### 1.4 Lógica Difusa

Es la lógica aplicada a conceptos que pueden tomar un valor cualquier de veracidad dentro de un conjuntos de valores que oscilan entre dos extremos, la verdad absoluta y falsedad total.Lo que es difuso, impreciso o vago no es la lógica sino el objeto que estudia, expresa la falta de definición del concepto al que se aplica.La lógica difusa permite tratar información imprecisa, en términos de conjuntos borrosos que se combinan en reglas para definir acciones, de esta manera, los sistemas de control basados en lógica difusa combinan variables de entrada [Eduardo et al., 2006]

## 2 Materiales y Métodos

### 2.1 Diagrama de Bloques de procesos



### 2.2 Método utilizado para la resolución

El Dataset escogido muestra un problema de clasificación, ya que la variable escogida a predecir cuenta con dos opciones: Y, si se acepta el préstamo, y N si se rechaza el préstamo. Además al ser un problema que cuenta con múltiples variables tanto cualitativas como cuantitativas, se decide que para la resolución de este modelo y obtención del modelo más adecuado hay que acudir a la realización de una **Regresión Logística Múltiple**.
Para la limpieza de los datos se recurrió a la búsqueda y eliminación de los valores NA, y posteriormente se reemplazaron las variables que según el análisis de correlación se procedió a la obtención de una regresión lineal simple con la variable más correlacionada, y en caso de que las correlaciones entre variables independientes sean muy bajas, los datos se completaron mediante sus medias, y así tener el Dataset con la data completa sin valores NA.
Para la realización de cada uno de los cálculos se utilizó el entorno y lenguaje de programación con un enfoque al análisis estadístico de código abierto **R**, incluyendo ciertas librerías extras tales como [Ihaka and Gentleman, 1996]:

- **Readr**: para la importación del archivo CSV a un dataset leído por R.
- **Corrplot:** para la representación de la correlación gráficamente según puntos.
- **Rgl:** Realización de un gráfico en 3D.
- **Car:** Para la representación de correlaciones con gráficas de funciones matemáticas.

### 2.3 Método para creación del sistema difuso

Para poder resolver esto, y establecer un modelo confiable basado en razonamiento difuso, se siguieron las reglas del **Modelo de Mamdani**, mismo que se basa en la fuzzificación indicando las variables de entrada, a continuación creando reglas específicas dependiendo de las variables fuzzificadas, se prosigue a hacer una análisis de las reglas, para que finalmente de acuerdo a los datos en cada variable, y con el análisis de las reglas se proceda a la acción de defuzificación, obteniendo un centroide específico, que será la respuesta de grado de pertenecía en todos los datos probados.
Para la ejecución de esto se utilizó el lenguaje de programación **Python**, dentro del IDE Gooogle Colaborary.

## 3 Experimentos

Inicialmente se cargó el archivo CSV en el el script en R, y se declararon todas las variables la y, y 10 variables independientes, donde a las variables cualitativas se las tuvo que codificar en nuevas variables para sean booleanos, y puedan ser leídas matemáticamente en el modelo.

```
y_original = loan$Loan_Status
x1_original = loan$Gender
x2_original = loan$Married
x3_original = loan$Dependents
x4_original = loan$Education
x5_original = loan$Self_Employed
x6_original = loan$ApplicantIncome
x7_original = loan$CoapplicantIncome
x8_original = loan$LoanAmount
x9_original = loan$Loan_Amount_Term
x10_original = loan$Credit_History
x11p_original = loan$Property_Area
```

```
#Area de la propiedad
#Urbano          x11=  0 x12 = 0
#Semi-Urbano     x11 = 1 x12 = 0
#Rural           x11 = 0 x12 = 1

x11_original = (x11p_original=='Semiurban')*1
x12_original = (x11p_original=='Rural')*1
```

Tan solo en la variable **x3(Dependientes)** que representa el número de hijos que tiene la persona que requiere el préstamo, y cuenta con 4 instancias 0,1,2,3+, siendo así una variable categórica, pero para su transformación requeriría 3 nuevas variables. Por lo tanto se optó por transformar todos los valores 3+ en números aleatorios que van del 3 al 5 tomando en cuenta que como promedio máximo de número de hijos en las familias de la sociedad actual es de 5.

```
#Intercambio de variables de x3 (Dependientes) a un número aleatorio entre 3 y 5
set.seed(1)# semilla establecida para que los datos aleatorios no cambien.
nuevox3 = sample(3:5,length(x3_original),replace = T)

for(l in 1:nrow(loan)){

  if(is.na(loan$Dependents[l])){
    loan$Dependents[l] = 6
  }
  if(loan$Dependents[l] == '3+'){
    loan$Dependents[l] = nuevox3[l]
  }
}
loan$Dependents = as.numeric(loan$Dependents)
x3_original = loan$Dependents
```

### 3.1 Limpieza de Datos

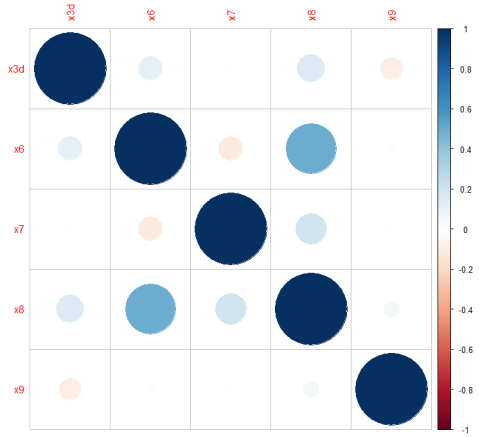
Con las variables cargadas se notó que contaba con varios valores NA.

| Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term |
|---|---|---|---|---|
| No | 3510 | 0 | 76 | 360 |
| No | 4887 | 0 | 133 | 360 |
| NA | 2600 | 3500 | 115 | NA |
| No | 7660 | 0 | 104 | 360 |
| No | 5955 | 5625 | 315 | 360 |
| No | 2600 | 1911 | 116 | 360 |
| No | 3365 | 1917 | 112 | 360 |
| NA | 3717 | 2925 | 151 | 360 |

Debido a esto se procedió a eliminar momentáneamente todas instancias que cuentan con valores NA, y con esto se volvieron a crear nuevas variables para establecer un modelo completamente limpio.

```
##############################################################################################

#Eliminacion de los NA
loan1 =  loan[! is.na(loan$Gender),] # Sacar media por correlación muy baja
loan2 =  loan1[! is.na(loan1$Dependents),]
which(is.na(loan1$Dependents)) # de todas las observaciones solo existen 24 valores NA, y al ser una variable
# cuantitativa que se dividen en 3 variables, se optó por solo eliminar estos valores.
loan3 =  loan2[! is.na(loan2$Self_Employed),] # Sacar media por correlación muy baja con otras variables
loan4 =  loan3[! is.na(loan3$LoanAmount),] # ecuacion de regresión con x6
loan5 =  loan4[! is.na(loan4$Loan_Amount_Term),] # Sacar media por correlación muy baja con otras variables
loan6 =  loan5[! is.na(loan5$Credit_History),] # Sacar media por correlación muy baja con otras variables

y = loan6$Loan_Status
x1 = loan6$Gender
x2 = loan6$Married
x3d = loan6$Dependents
x4 = loan6$Education
```

Con las variables sin NA, se creó una correlación de forma gráfica, para saber si es posible obtener regresiones entre variables para lograr llenar los datos NA en el modelo original.




Se notó que tan solo la variable x6(ApplicantIncome) y x8(LoanAmount) presentan una correlación significante, por lo cuál será la única variable que se podrá llenar mediante la correlación.

```
###Reemplazo valores NA de la variable x8 (Monto del prestamo).
reg2 = lm(x8~x6)
summary(reg2)
#yest2 = reg2$coefficients[1]+reg2$coefficients[2]*x6_original

for(i in 1:nrow(loanfinal))
  if(is.na(loanfinal$LoanAmount[i])){
    loanfinal$LoanAmount[i] = round(reg2$coefficients[1]+reg2$coefficients[2]*x6_original[i])

    x8_original[i]= loanfinal$LoanAmount[i]
  }
```

Las demás variables al no ser posible sacar una correlación adecuada se decidió reemplazar los NA por los valores de la media:

```
# Credit_History
for(i in 1:nrow(loanfinal))
  if(is.na(loanfinal$Credit_History[i])){
    loanfinal$Credit_History[i] = round(mean(x10))
    x10_original[i] = round(mean(x10))
  }

# Gender
for(i in 1:nrow(loanfinal))
  if(is.na(loanfinal$Gender[i])){
    loanfinal$Gender[i] = round(mean(x1))
    if(loanfinal$Gender[i]=='1'){
      loanfinal$Gender[i]='Male'
    }else{
      loanfinal$Gender[i]='Female'
    }
    x1_original[i] = round(mean(x1))

  }
```

Finalmente se obtiene el dataset llamado **loanfinal** con sus 614 observaciones, ya sin ningún dato NA.

| Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term |
|---|---|---|---|---|
| No | 5849 | 0 | 148 | 360 |
| No | 4583 | 1508 | 128 | 360 |
| Yes | 3000 | 0 | 66 | 360 |
| No | 2583 | 2358 | 120 | 360 |
| No | 6000 | 0 | 141 | 360 |
| Yes | 5417 | 4196 | 267 | 360 |
| No | 2333 | 1516 | 95 | 360 |

### 3.2 Regla de Oro

Con los datos limpios se procedió a la aleatorización de los datos del vector, para tener instancias más correctas y poder dividir en dataset en training con el 70% de instancias y el test con el 30% restante.

```
##########################REGLA DE ORO###################################

set.seed(2)# semilla establecida para que los datos aleatorios no cambien.
aleatorio = sample(614,replace = F) #datos aleatorios de las observaciones
#Se obtiene la matriz aleatoria de los datos.
trainrandom = loanfinal[c(aleatorio),] #Se coloca la matriz con números de

##########################TRAINING#########################################
training =trainrandom[1:430,]
```

Se crearon las variables de entrenamiento, y con estás se creó la regresión logística:

```
#Creamos las variables de entrenamiento para el modelo de regresion logistica
yE = training$Loan_Status
x1E = training$Gender
x2E = training$Married
x3E = training$Dependents
x4E = training$Education
x5E = training$Self_Employed
x6E = training$ApplicantIncome
x7E = training$CoapplicantIncome
x8E = training$LoanAmount
x9E = training$Loan_Amount_Term
x10E = training$Credit_History
x11_E = training$Property_Area
```

```
#Creamos el modelo de regresion logistica con las variables de entrenamiento
reg3 = glm(yE~x1E+x2E+x3E+x4E+x5E+x6E+x7E+x8E+x9E+x10E+x11E+x12E,family=binomial())
summary(reg3)
```

```
Call:
glm(formula = yE ~ x1E + x2E + x3E + x4E + x5E + x6E + x7E +
    x8E + x9E + x10E + x11E + x12E, family = binomial())

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4696  -0.2928   0.4494   0.6830   2.7204

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.937e+00  1.176e+00  -1.647  0.09949 .
x1E         -6.059e-02  3.791e-01  -0.160  0.87302
x2E          7.049e-01  3.034e-01   2.323  0.02018 *
x3E          1.425e-01  1.242e-01   1.147  0.25147
x4E          7.523e-01  3.242e-01   2.321  0.02031 *
x5E          1.326e-02  3.918e-01   0.034  0.97300
x6E          3.040e-05  2.784e-05   1.092  0.27487
x7E         -1.926e-05  6.385e-05  -0.302  0.76294
x8E         -3.948e-03  2.038e-03  -1.937  0.05277 .
x9E         -5.616e-03  2.920e-03  -1.923  0.05443 .
x10E         4.398e+00  5.731e-01   7.673 1.67e-14 ***
x11E         9.926e-01  3.518e-01   2.822  0.00478 **
x12E        -4.477e-02  3.150e-01  -0.142  0.88698
---
```

Se creó la parte de test, misma a la que se llamaron nuevas variables,

```
#########################TEST#################
test = trainrandom[431:length(x1_original),]
yP = test$Loan_Status
x1P = test$Gender
x2P = test$Married
x3P = test$Dependents
x4P = test$Education
x5P = test$Self_Employed
x6P = test$ApplicantIncome
x7P = test$CoapplicantIncome
x8P = test$LoanAmount
x9P = test$Loan_Amount_Term
x10P = test$Credit_History
x11_P = test$Property_Area
```

Estas variables son las que se utilizan en la ecuación de regresión logística establecida mediante la fórmula:

$$E(y) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}}$$

```
# Formamos la ecuacion de regresión y le probamos con las variables de test
yest = exp(reg3$coefficients[1]+reg3$coefficients[2]*x1P+reg3$coefficients[3]*x2P+
             reg3$coefficients[4]*x3P+reg3$coefficients[5]*x4P+reg3$coefficients[6]*x5P+
             reg3$coefficients[7]*x6P+reg3$coefficients[8]*x7P+reg3$coefficients[9]*x8P+
             reg3$coefficients[10]*x9P+reg3$coefficients[11]*x10P+reg3$coefficients[12]*x11P+reg3$coefficients[13
]*x12P)/(1+exp(reg3$coefficients[1]+reg3$coefficients[2]*x1P+reg3$coefficients[3]*x2P+

                 reg3$coefficients[4]*x3P+reg3$coefficients[5]*x4P+reg3$coefficients[6]*x5P+

                 reg3$coefficients[7]*x6P+reg3$coefficients[8]*x7P+reg3$coefficients[9]*x8P+

                 reg3$coefficients[10]*x9P+reg3$coefficients[11]*x10P+reg3$coefficients[12]*x11P+reg3$coefficient
s[13]*x12P))
```

Con la ecuación obtenida, se procede a la obtención del error, y la comparación con el valor original de la y en los datos de test, para saber el accuracy del modelo creado.

```
yest1 = round(yest) # se redondea l

error = (yP==yest1)*1

accuracy = sum(error)/length(yP)
print(accuracy)
```

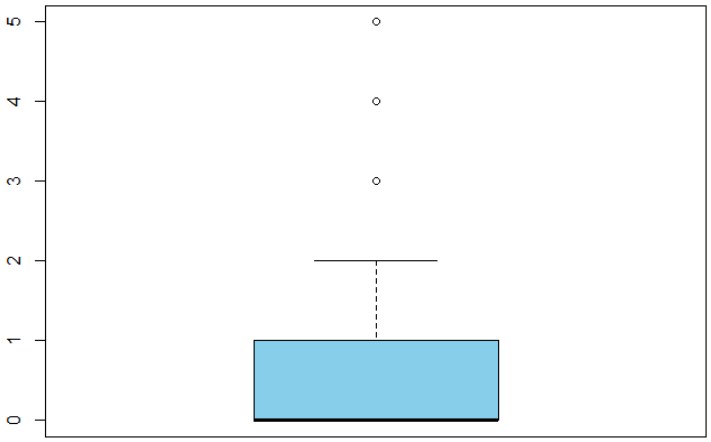
```
[1] 0.7608696
```

Mediante el análisis de accuracy se nota que el modelo es bueno, ya que presenta una precisión del **76.09%**.

### 3.3 Valores Atípicos (Outliers

Como primer paso procedemos a representar en un gráfico de caja con ayuda del comando (boxplot) todas las variables cuantitativas, los datos atípicos se encuentran representados por un circulo, los mismos que se ubican fuera de los limites de la caja.

```
#representación de outliers
g_caja1<-boxplot(loanfinal$Dependents, col="skyblue", frame.plot=T)

g_caja2<-boxplot(loanfinal$ApplicantIncome, col="skyblue", frame.plot=T)

g_caja3<-boxplot(loanfinal$CoapplicantIncome, col="skyblue", frame.plot=T)

g_caja4<-boxplot(loanfinal$LoanAmount, col="skyblue", frame.plot=T)

g_caja5<-boxplot(loanfinal$Loan_Amount_Term, col="skyblue", frame.plot=T)
```



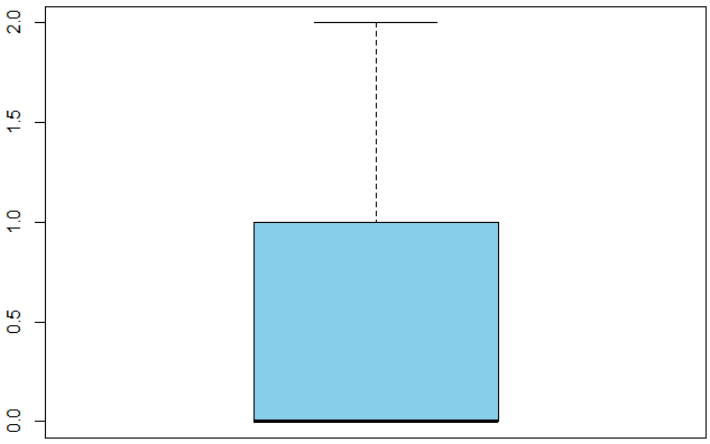

Posteriormente con el comando which obtenemos las posiciones de nuestros valores atípico, permitiendo realizar una comparación de los valores que más se repiten entre las variables.

```
#Sacamos la posición de los valores atípicos
posicion_caja1 = which(d1>2)
posicion_caja2 = which(d2>10139)
posicion_caja3 = which(d3>5701)
posicion_caja4 = which(d4>265)
posicion_caja5 = which(d5>360)
posicion_caja5_2 = which(d5<360)
```

```
> posicion_caja1
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
> posicion_caja2
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
> posicion_caja3
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
> posicion_caja4
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
> posicion_caja5
 [1] 12 19 23 30 33 40 47 52 53 60 62 67 86 89 94
> posicion_caja5_2
  [1]   1   2   3   4   5   6   7   8   9  10  11  13  14  15  16  17  18  20  21  22  24  25  26  27  28  29  31  32  34  35  36  37  38  39  41  42  43  44  45
 [40]  46  48  49  50  51  54  55  56  57  58  59  61  63  64  65  66  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  87  88  90  91  92
 [79]  93  95  96  97  98  99 100 101 102
```

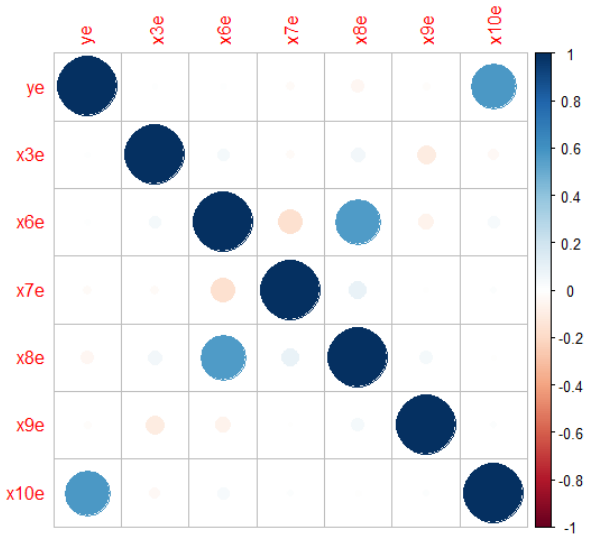
Procedemos a la eliminación de los valores atípicos con mayor grado de repetición y representación de la misma.

```
#Eliminación de outliers
loanf<-loanfinal[!(loanfinal$Dependents %in% g_caja1$out),]

#REPRESENTACION SIN OUTLIERS
g_caja1<-boxplot(loanf$Dependents, col="skyblue", frame.plot=T)
```




### 3.4 Reducción de Variables

Una vez eliminados todos los valores atípicos tenemos como resultado final un dataset llamado **loanf** el cual contiene 563 observaciones.Se procedió a la aleatorización de los datos del vector, para tener instancias más correctas y poder dividir en dataset en training con el 70% de instancias y el test con el 30% restante.Posteriormente se crearon las variables de entrenamiento, con las cuales obtuvimos la representación gráfica de correlación para así reducir el número de variables y mejorar el modelo estudiado



Una vez evaluada la gráfica de correlación se llego como resultado final eliminar la variable **x3(Dependientes)** que representa el número de hijos que tiene la persona que requiere el préstamo, Por consiguiente se creó la regresión logística,

```
#Creamos el modelo de regresion logistica con las variables de entrenamiento
reg5 = glm(ye~x1e+x2e+x4e+x5e+x6e+x7e+x8e+x9e+x10e+x11e+x12e,family=binomial())
summary(reg5)
```

Con las variables de test se procedió a crear la ecuación de regresión logística, referente a estos dos importantes datos se calculo el error, permitiendo la comparación con el valor original de la y en los datos de test, para saber el accuracy del modelo creado.

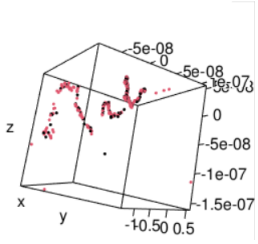
```
# Que tan bueno es mi modelo de regresiÃ³n
accuracy3 = sum(error3)/length(yp)
print(accuracy3)
```

```
[1] 0.7810651
```

Mediante el análisis del nuevo accuracy se observa un aumento de la precisión del 2.02% por lo cual se observa una mejoría en el modelo estadístico el cual esta representado por un **78.10%**.

### 3.5 Escalamiento Multidimensional

Al realizar la técnica de escalamiento multidimensional, creamos como paso principal nuestra matriz de distancias con variables únicamente cuantitativas, la misma que es representada en una escala euclidiana y tres dimensiones permitiendo así una aproximación lo mejor posible a las distancias de partida.




### 3.6 Análisis Factorial

**Análisis de la Matriz Correlación** En primer lugar se realizó la correlación de Bartlett, para mediante el p-value comprobar si las variables escogidas pueden ser expresadas como combinación lineal de otras variables.
Se escogió el análisis con las variables independientes del modelo, y mostró un p-value menor a $-2.2^{-16}$, por lo tanto mostrando que efectivamente se puede realizar una combinación lineal.

```
# Test de Barlett
vectorpca = c(4,7:11)
bt = bartlett.test(loanf[, vectorpca]) #Mientas mas alto mejor.
print(bt)
```

```
    Bartlett test of homogeneity of variances

data:  loanf[, vectorpca]
Bartlett's K-squared = 25542, df = 5, p-value < 2.2e-16
```

Como análisis extra en las medidas de adecuación de la muestra se obtuvo la **Media de Adecuación de la Muestra KMO**, para conocer el coeficiente de correlación entre las variables sin la influencia del resto. Aqui de igual forma se analizaron las variables independientes, y nos dio un resultado del criterio KMO menor a 0.5, por lo tanto siendo inaceptable y no aportaría a un mejoramiento del modelo.

```
# Kaiser, Meyer, Olkin - KMO
library("REdaS")
kmo = KMOS(trainrandom1[, vectorpca]) #Criterio 0.40 -- no adecuado
print(kmo)
```

```
Kaiser-Meyer-Olkin Statistics

Call: KMOS(x = trainrandom1[, vectorpca])

Measures of Sampling Adequacy (MSA):
       Dependents   ApplicantIncome CoapplicantIncome        LoanAmount  Loan_Amount_Term    Credit_History
        0.4002576         0.4311522         0.2187745         0.4338838         0.2372522         0.6494201

KMO-Criterion: 0.4050549
```

**Cálculo de componentes principales** Se calcularon los componentes principales, y se generó un data frame con los datos de las variables independientes trasformadas como componentes principales.

```
plot(pca, type = "l")
summary(pca)
```

```
Importance of components:
                          PC1    PC2    PC3    PC4    PC5     PC6
Standard deviation     1.2572 1.0546 1.0216 0.9980 0.9485 0.60660
Proportion of Variance 0.2634 0.1854 0.1739 0.1660 0.1500 0.06133
Cumulative Proportion  0.2634 0.4488 0.6227 0.7887 0.9387 1.00000
```

Se escogió trabajar probando tanto con 4Pc y 5PC, que presentan una proporción acumulada del 0.7887 y 0.9387 respectivamente.
A partir de esto se obtuvieron nuevas regresiones utilizando los PC's, y dio razón a lo que mostraba el análisis KMO, dando un modelo con un accuracy que con

5 PC disminuyó en 8,19% respecto al mejor modelo obtenido anteriormente sin PC's.

```
#Regresión con PCA
#Con 4 PC
regpca = lm(ye ~ x1e+x2e+cp$PC1+x4e+x5e+cp$PC2+cp$PC3+cp$PC4+x11e+x12e)
summary(regpca)
#Con 5 PC
regpca2 = lm(ye ~ x1e+x2e+cp$PC1+x4e+x5e+cp$PC2+cp$PC3+cp$PC4+cp$PC5+x11e+x12e)
summary(regpca2)
```

```
# Que tan bueno es mi modelo de regresiÃ³n
accuracypca4 = sum(errorpca1)/length(yp)
print(accuracypca4)
```

```
[1] 0.6686391
```

### 3.7 Creación de variables de entrada

Para la creación del sistema difuso, hemos tomado como variables de entrada: Aceptación (y), Ingreso (x_ingreso), Monto (x_monto) y Meses(x_meses).

* Universo de Discurso: los universos de discursos de estas variables son:

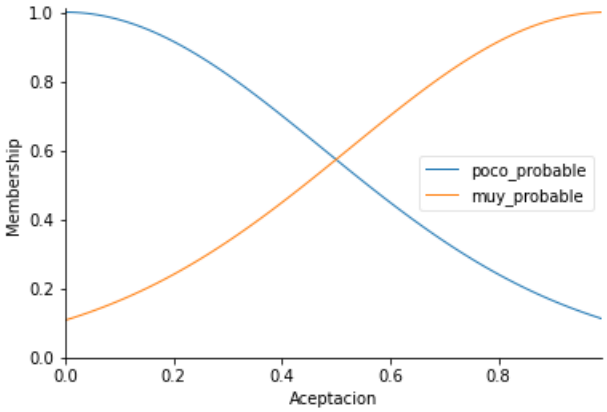
- y [0, 1, 0.01]
- x_ingreso [150, 63337, 1]
- x_monto [9, 650, 1]
- x_meses [12, 480, 1]

```
#Variables
#yN = Loan_Status
#x6N = ApplicantIncome
#x8N = LoanAmount
#x9N = Loan_Amount_Term

# Universo de discurso
y = ctrl.Consequent(np.arange(0, 1, 0.01),'Aceptacion')
x_ingreso = ctrl.Antecedent(np.arange(150, 63337, 1),'ingreso')
x_monto = ctrl.Antecedent(np.arange(9, 650, 1),'monto')
x_meses = ctrl.Antecedent(np.arange(12, 480, 1),'meses')
```

### 3.8 Funciones de pertenencia

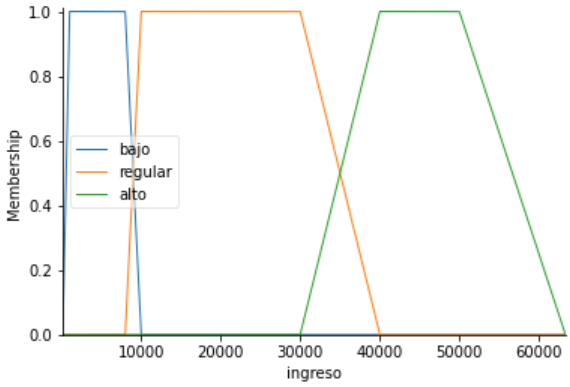
* A la variable de entrada Aceptación, le asociamos dos conjuntos difusos con las etiquetas lingüisticas: poco_probale y muy_probable; la función de pertenencia que se aplicó es de tipo gaussiana.

```
#Aceptación del prestamo
y['poco_probable'] = fuzz.gaussmf(y.universe, 0, 0.474)
y['muy_probable'] = fuzz.gaussmf(y.universe, 1, 0.474)
y.view()
```



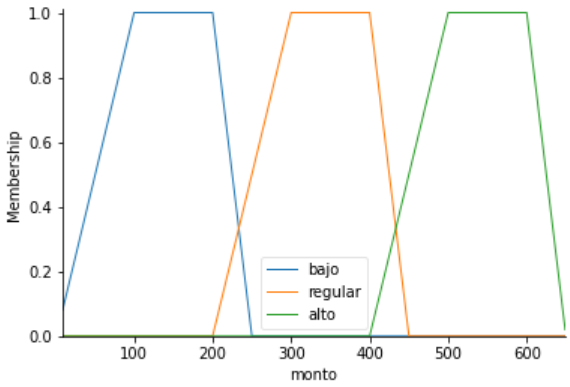

∗ A la variable de entrada Ingreso mensual en cientos de Ruphias, le asociamos tres conjuntos difusos con las etiquetas lingüisticas: bajo, regular y alto; la función de pertenencia que se aplicó es de tipo trapezoidal.

```
#Ingreso mensual en cientos de Ruphias
x_ingreso['bajo']=fuzz.trapmf(x_ingreso.universe,[150,1000,8000,10000])
x_ingreso['regular']=fuzz.trapmf(x_ingreso.universe,[8000,10000,30000,40000])
x_ingreso['alto']=fuzz.trapmf(x_ingreso.universe,[30000,40000,50000,63337])
x_ingreso.view()
```



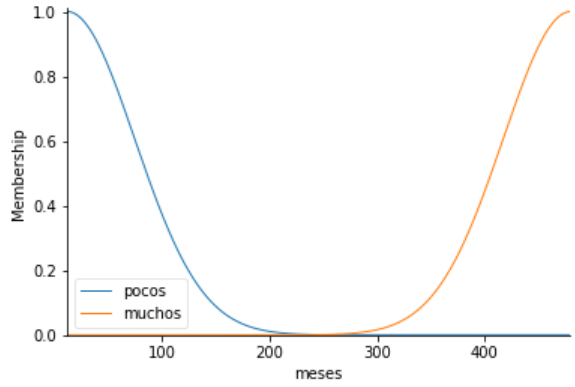
* A la variable de entrada Monto en Lakhs (*100000), le asociamos tres conjuntos difusos con las etiquetas lingüisticas: bajo, regular y alto; la función de pertenencia que se aplicó es de tipo trapezoidal.

```
#Monto en Lakhs (*100000)
x_monto['bajo']=fuzz.trapmf(x_monto.universe,[1,100,200,250])
x_monto['regular']=fuzz.trapmf(x_monto.universe,[200,300,400,450])
x_monto['alto']=fuzz.trapmf(x_monto.universe,[400,500,600,650])
x_monto.view()
```




* A la variable de entrada Meses le asociamos dos conjuntos difusos con las etiquetas lingüisticas: pocos y muchos; la función de pertenencia que se aplicó es de tipo gaussiana.

```
#Meses
x_meses['pocos'] = fuzz.gaussmf(x_meses.universe, 12, 62.78298)
x_meses['muchos'] = fuzz.gaussmf(x_meses.universe, 480, 62.78298)
x_meses.view()
```



### 3.9 Creación de reglas

Se crearon las reglas difusas para el buen funcionamiento del sistema experto

* Regla #1
  Si el ingreso es bajo o el monto es bajo y muchos meses entonces muy probable
```
rule1 = ctrl.Rule(x_ingreso['bajo'] | x_monto['bajo'] & x_meses['muchos'], y['muy_probable'])
```
* Regla #2
  Si el ingreso es regular o el monto regular y pocos meses entonces poco probable
```
rule2 = ctrl.Rule(x_ingreso['regular'] | x_monto['regular'] & x_meses['pocos'], y['poco_probable'])
```
* Regla #3
  Si el ingreso es alto y el monto es alto entonces muy probable
```
rule3 = ctrl.Rule(x_ingreso['alto'] & x_monto['alto'], y['muy_probable'])
```
* Regla #4
  i el ingreso es regular o alto y el monto es bajo entonces poco probable
```
rule4 = ctrl.Rule(x_ingreso['regular'] | x_ingreso['alto'] & x_monto['bajo'], y['poco_probable'])
```
* Regla #5
  Si el ingreso es bajo y el monto alto entonces poco probable
```
rule5 = ctrl.Rule(x_ingreso['bajo'] & x_monto['alto'], y['poco_probable'])
```

Se guardan las reglas y se compilan las reglas

```
aceptacion_ctrl = ctrl.ControlSystem([rule1,rule2,rule3,rule4,rule5])
aceptacion = ctrl.ControlSystemSimulation(aceptacion_ctrl)
```

### 3.10 Salida de las Reglas

Para lograr la salida de reglas, cada una de las variables escogidas deben tener un valor numérico para que se proceda con la defuzificación. Para los datos se requiere traer el archivo CSV correspondiente que cuenta con 4 columnas, las 3 variables que fueron base para la creación de las entidades, además de la variable a predecir (Loan Status).
Se utilizó la librería **pandas** para leer los datos, y un método específico de colab para cargar el archivo específico desde nuestro ordenador.

```
[ ]  import pandas as pd
```

```
     from google.colab import files
     uploaded = files.upload()
```

```
     Elegir archivos No se eligió archivo    Upload widget is only avai
     Saving datoss.csv to datoss (2).csv
```

```
[ ]  import io
     df2 = pd.read_csv(io.BytesIO(uploaded['datoss.csv']))
     # Dataset is now stored in a Pandas Dataframe
```

Una vez cargado el archivo csv con datos de prueba del dataset utilizando en el modelo estadístico se pasa estos datos como una matriz, mediante el método **to_numpy()**.
Teniendo la matriz lista, se procede a la creación un bucle for que va desde 0 hasta la longitud de la matriz (183), y se van cargando los datos de matriz fila por fila según las variables correspondientes, y las respuestas computadas se van guardando en un nuevo vector.

```
matriz = df2.to_numpy()
respuestas =[]
for i in range(len(matriz)):
  aceptacion.input['ingreso'] = matriz[i][0]
  aceptacion.input['monto'] =  matriz[i][1]
  aceptacion.input['meses'] =  matriz[i][2]
  aceptacion.compute()
  respuestas.append(aceptacion.output['Aceptacion'])

print(respuestas)
```

```
[0.3482995853173259, 0.6443390843428981, 0.6443390843428981,
```

Una vez teniendo el vector de respuestas computadas con los grados de pertenencia de cada dato,se procedió a establecer 2 criterios específicos para darle una respuesta lingüística acorde a lo que planteaba el dataset, de si el préstamo es aceptado o no. Los criterio establecidos fueron:

- Si el grado de pertenencia es menor o igual 0.5 es poco probable de que se acepte el préstamo.
- Si el grado de pertenencia es mayor a 0.5 muy problable de que se acepte el préstamo.

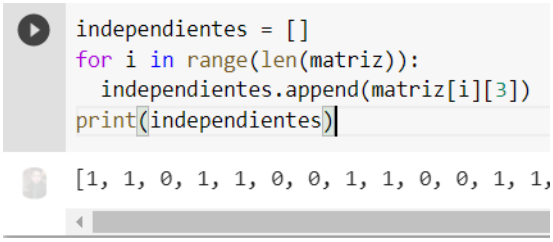
Teniendo establecidos los criterios se procedió a realizar un nuevo bucle for que transforme los datos obtenidos en el vector de las respuestas de salida, en 1 o 0 de acuerdo a los criterios, siendo 1 que es muy probable, y 0 es poco probable.

```
# >0.5 muy problable de que se acepte el préstamo
#<=0.5 poco problable de que se acepte el préstamo

for i in range(len(respuestas)):
  if respuestas[i]<= 0.5:
    respuestas[i] = 0
  else:
    respuestas[i]=1

print(respuestas)
```

```
[0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

### 3.11 Comparación con el modelo estadístico

Para realizar la comparación con el modelo estadístico se comparan específicamente las respuestas que venían dentro del data, es decir la variable llamada Loan Status que devuelve un valor de 1 si es préstamo es aceptado y 0 si el préstamo no es aceptado.
Se procedió a la creación de un nuevo vector mismo que sería cargado con los datos de la matriz de la columna perteneciente a Loan Status.



```
independientes = []
for i in range(len(matriz)):
  independientes.append(matriz[i][3])
print(independientes)
```

```
[1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1,
```

Con el vector de variables independientes siendo parte del script en Python, mediante un for se comparan los valores del vector de las respuestas obtenidas por la lógica difusa, con el vector cargado de los datos de la variable independiente, ya que ambos tienen la misma longitud; creando un nuevo vector que se agrega 1 al vector si las respuestas son iguales, o u 0 si las respuestas son diferentes.

```
comparacion = []
for i in range(len(matriz)):
  if respuestas[i]==independientes[i]:
    comparacion.append(1)
  else:
    comparacion.append(0)
```

Finalmente se procede a la obtención del accuracy (qué tan bueno es el modelo) de las respuestas obtenidas mediante lógica difusa, para esto se sumaron del vector de comparación todos los valores, es decir se obtienen todos los valores de 1, y la respuesta de esta se divide para el número total de datos que existen en el vector.

```
suma = sum(comparacion)
accuracy = sum(comparacion)/len(matriz)
print(accuracy)
```

```
0.657608695652174
```

Finalmente se nota que las respuestas obtenidas por el sistema experto basado en lógica difusa nos da un accuracy del **65.76%**

## 4 Conclusiones y Prospectivas

1. Se llegó a la conclusión respecto al apartado de estadística que el mejor modelo se consiguió a través de la eliminación de los ouliers, y sin tomar en cuenta la variable x3(Dependents), generando un accuracy del 78.1%, mostrando un modelo bastante aceptable para la predicción, y siendo 2.01 % mejor en su accuracy en comparación con la regresión de las variables originales.
2. Aplicando el análisis factorial de las variables, se notó inicialmente un KMO menor a 0.4, siendo algo no factible realizar la combinación de variables, y esta premisa se reflejó al momento de la realización de ecuación de regresión en base a componentes principales (PC), donde el accuracy en este modelo disminuyó en 8,19% respecto al mejor modelo creado.
3. Dentro del modelo estadístico se obtuvieron regresiones formadas solo con las variables escogidas para la creación de Sistema Experto de Inferencia difusa, mismo del que se obtuvo su accuracy, que comparado con el obtenido por el Sistema de Inferencia difusa, tan solo distan en 1.69%, siendo el segundo el menor de ambos, pero presentan una gran similitud, mostrando así que los resultados obtenidos a través del Sistema Experto, pueden brindar resultados casi similares a los que se pueden extraer mediante la ecuación de regresión logística.

---

## References

[Batanero, 2001] Batanero, C. (2001). Didáctica de la estadística. *Granada: Universidad de Granada*.

[Chitarroni, 2002] Chitarroni, H. (2002). La regresión logística.

[Eduardo et al., 2006] Eduardo, C., De Vito, E. L., et al. (2006). Introducción al razonamiento aproximado: lógica difusa. *Revista Americana de Medicina Respiratoria*, 6(3):126–136.

[Ihaka and Gentleman, 1996] Ihaka, R. and Gentleman, R. (1996). R: a language for data analysis and graphics. *Journal of computational and graphical statistics*, 5(3):299–314.

[Pitarque et al., 1998] Pitarque, A., Roy, J. F., and Ruiz, J. C. (1998). Redes neurales vs modelos estadísticos: Simulaciones sobre tareas de predicción y clasificación. *Psicológica*, 19:387–400.