

# Credit Score

Santiago Yael Morales Torres

2025-07-07

## 1 Contexto y explicación de los datos

La base de datos corresponde al registro de las características demográficas y bancarias de los clientes de un banco, donde cada fila representa un registro mensual por cliente. En total, se tienen 100000 registros correspondientes a los meses de Enero a Agosto de 12500 clientes, y un total de 28 variables. En particular, la variable objetivo de este proyecto es Credit Score, la cual resume el comportamiento del cliente respecto a sus cuentas, transacciones, inversiones, préstamos y deudas, además de dar un indicio del riesgo de incumplimiento en un futuro, y se clasifica en 3 categorías: Good, Standard y Poor.

Los objetivos de este proyecto son:

- Encontrar relaciones importantes entre las variables de la base y Credit Score.
- Analizar que tan distinguibles o separables son los niveles del Credit Score mediante algoritmos de Aprendizaje No Supervisado.
- Encontrar un modelo de Aprendizaje Supervisado que nos permita predecir con un Accuracy elevado a Credit Score.

A continuación, se explican las variables contenidas en la base de datos (el tipo y valores que se mencionan corresponden a la base de datos después de ser limpia y transformada):

**ID.** Llave primaria de cada uno de los registros, Identificador único por cliente y mes. *Tipo:* Carácter.

**Customer\_ID.** Identificador único del cliente. *Tipo:* Carácter.

**Name.** Nombre Completo del Cliente. *Tipo:* Carácter.

**Month.** Mes correspondiente al registro. *Tipo:* Carácter.

**Age.** Edad del cliente. *Tipo:* Entero.

**SSN.** Número de Seguridad Social. *Tipo:* Carácter.

**Occupation.** Profesión del cliente. *Tipo:* Categórica Nominal con 15 niveles (Accountant, Architect, Developer, Doctor, Engineer, Entrepreneur, Journalist, Lawyer, Manager, Mechanic, Media\_Manager, Musician, Scientist, Teacher, Writer).

**Annual\_Income.** Ingreso bruto anual del cliente. *Tipo:* Numérica.

**Monthly\_Inhand\_Salary.** Ingreso neto mensual del cliente (Ingreso bruto mensual menos impuestos y deducciones). *Tipo:* Numérica.

**Num\_Bank\_Accounts.** Número de cuentas que posee el cliente. *Tipo:* Entero.

**Num\_Credit\_Card.** Número de tarjetas de crédito que posee el cliente. *Tipo:* Entero.

**Interest\_Rate.** Tasa de interés que tienen los préstamos del cliente. *Tipo:* Entero.

**Num\_of\_Loan.** Número de préstamos que ha solicitado el cliente. *Tipo:* Entero.

**Type\_of\_Loan.** Tipo de préstamo(s) que ha solicitado el cliente. *Tipo:* Multivaluada/Factor con 8 niveles (Auto Loan, Credit-Builder Loan, Personal Loan, Home Equity Loan, Mortgage Loan, Student Loan, Debt Consolidation Loan, Payday Loan).

**Delay\_from\_due\_date.** Número de días que el cliente se retrasó en pagar respecto a la fecha límite de pago. *Tipo:* Entero.

**Num\_of\_Delayed\_Payment.** Número de pagos que se han realizado después de la fecha límite de pago. *Tipo:* Entero.

**Changed\_Credit\_Limit.** Variación en el límite de crédito otorgado al cliente en ese mes. *Tipo:* Numérica.

**Num\_Credit\_Inquiries.** Número de veces que las instituciones financieras han consultado el historial crediticio del cliente. *Tipo:* Entero.

**Credit\_Mix.** Califica la variedad de préstamos y créditos que tiene el cliente. *Tipo:* Factor con 3 niveles (Good, Standard, Bad).

**Outstanding\_Debt.** Deuda pendiente que tiene el cliente con el banco. *Tipo:* Numérica.

**Credit\_Utilization\_Ratio.** Porcentaje de crédito utilizado respecto al total de crédito otorgado al cliente. *Tipo:* Numérica.

**Credit\_History\_Age.** Antigüedad en años del historial crediticio del cliente. *Tipo:* Entero.

**Payment\_of\_Min\_Amount.** Indica si el cliente hizo el pago mínimo o no. *Tipo:* Factor con 2 niveles (Yes, No).

**Total\_EMI\_per\_month.** Total de pagos mensuales fijos del cliente. *Tipo:* Numérica.

**Amount\_invested\_monthly.** Monto mensual que el cliente invierte mensualmente. *Tipo:* Numérica.

**Payment\_Behaviour.** Clasificación del tipo de gastos que hace el cliente y el tipo de pagos que realiza para cubrirlos. *Tipo:* Factor con 6 niveles (Low\_spent\_Small\_value\_payments, Low\_spent\_Medium\_value\_payments, Low\_spent\_Large\_value\_payments, High\_spent\_Small\_value\_payments, High\_spent\_Medium\_value\_payments, High\_spent\_Large\_value\_payments).

**Monthly\_Balance.** Saldo mensual del cliente después de gastos, pagos e inversiones. *Tipo:* Numérica.

**Credit\_Score.** Califica el comportamiento financiero del cliente y por tanto el riesgo de incumplimiento. *Tipo:* Factor con 3 niveles (Good, Standard y Poor).

## 2 Limpieza y transformación de datos

La base de datos tiene una estructura repetitiva cada 8 registros, siendo los primeros 8 registros los correspondientes a los meses de Enero a Agosto del primer cliente, los siguientes 8 a los meses de Enero de Agosto del segundo cliente y así sucesivamente. La siguiente tabla corresponde los primeros 8 registros del primer cliente.

ID	Customer_ID	Month	Name	Age	SSN	Occupation	Annual_Income
5634	CUS_0xd40	January	Aaron Maashoh	23	821-00-0265	Scientist	19114.12
5635	CUS_0xd40	February	Aaron Maashoh	23	821-00-0265	Scientist	19114.12
5636	CUS_0xd40	March	Aaron Maashoh	-500	821-00-0265	Scientist	19114.12
5637	CUS_0xd40	April	Aaron Maashoh	23	821-00-0265	Scientist	19114.12
5638	CUS_0xd40	May	Aaron Maashoh	23	821-00-0265	Scientist	19114.12
5639	CUS_0xd40	June	Aaron Maashoh	23	821-00-0265	Scientist	19114.12
5640	CUS_0xd40	July	Aaron Maashoh	23	821-00-0265	Scientist	19114.12
5641	CUS_0xd40	August		23	#F%\$D@*&8	Scientist	19114.12

Monthly_Inhand_Salary	Num_Bank_Accounts	Num_Credit_Card	Interest_Rate	Num_of_Loan
1824.843	3	4	3	4
NA	3	4	3	4
NA	3	4	3	4
NA	3	4	3	4
1824.843	3	4	3	4
NA	3	4	3	4
1824.843	3	4	3	4
1824.843	3	4	3	4

Type_of_Loan	Delay_from_due_date
Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan	3
Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan	-1
Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan	3
Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan	5
Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan	6
Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan	8
Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan	3
Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan	3

Num_of_Delayed_Payment	Changed_Credit_Limit	Num_Credit_Inquiries	Credit_Mix	Outstanding_Debt
7	11.27	4	—	809.98
	11.27	4	Good	809.98
7		4	Good	809.98
4	6.27	4	Good	809.98
	11.27	4	Good	809.98
4	9.27	4	Good	809.98
8	11.27	4	Good	809.98
6	11.27	4	Good	809.98

Credit_Utilization_Ratio	Credit_History_Age	Payment_of_Min_Amount	Total_EMI_per_month
26.82262	22 Years and 1 Months	No	49.57495
31.94496	NA	No	49.57495
28.60935	22 Years and 3 Months	No	49.57495
31.37786	22 Years and 4 Months	No	49.57495
24.79735	22 Years and 5 Months	No	49.57495
27.26226	22 Years and 6 Months	No	49.57495
22.53759	22 Years and 7 Months	No	49.57495
23.93379	NA	No	49.57495

Amount_invested_monthly	Payment_Behaviour	Monthly_Balance	Credit_Score
80.41529543900253	High_spent_Small_value_payments	312.49408867943663	Good
118.28022162236736	Low_spent_Large_value_payments	284.62916249607184	Good
81.699521264648	Low_spent_Medium_value_payments	331.2098628537912	Good
199.4580743910713	Low_spent_Small_value_payments	223.45130972736786	Good
41.420153086217326	High_spent_Medium_value_payments	341.48923103222177	Good
62.430172331195294	!@9#%8	340.4792117872438	Good
178.3440674122349	Low_spent_Small_value_payments	244.5653167062043	Good
24.785216509052056	High_spent_Medium_value_payments	358.12416760938714	Standard

Se puede observar que ciertas variables se mantienen constantes como Customer\_ID, Name, Age, SSN, Occupation, correspondientes a las variables demográficas del cliente, mientras que otras más relacionadas al comportamiento financiero varian cada mes, aunque algunas otras también se mantienen constantes (esto depende de cada cliente). En particular, con solo estos 8 registros ya se observan algunas complicaciones respecto a la integridad de los datos, teniendo valores atípicos o sin sentido (como -500 en Age, #F%\$D@\*%&8 en SSN, -1 en Delay\_from\_Due\_Date, !@9#%8 en Payment\_Behaviour, etc.), caracteres que ensucian algunos datos numéricos (\_8 en Num\_of\_Delayed\_Payment), campos en blanco, con guiones - o NA (como en Monthly\_Inhand\_Salary, Changed\_Credit\_Limit, Credit\_Mix, Credit\_History\_Age), y la variable Type\_of\_Loan es multivaluada.

Al cargar los datos, inicialmente se tiene la siguiente estructura:

```
## Rows: 100,000
## Columns: 28
## $ ID <dbl> 5634, 5635, 5636, 5637, 5638, 5639, 5640, 564~<chr> "CUS_0xd40", "CUS_0xd40", "CUS_0xd40", "CUS_0~<chr> "January", "February", "March", "April", "May~<chr> "Aaron Maashoh", "Aaron Maashoh", "Aaron Maas~<chr> "23", "23", "-500", "23", "23", "23", "23", "~<chr> "821-00-0265", "821-00-0265", "821-00-0265", ~<chr> "Scientist", "Scientist", "Scientist", "Scien~<chr> "19114.12", "19114.12", "19114.12", "19114.12~<dbl> 1824.843, NA, NA, NA, 1824.843, NA, 1824.843,~<int> 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, ~<int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 1385, 4, 4, 4, ~<int> 3, 3, 3, 3, 3, 3, 6, 6, 6, 6, 6, 6, ~<chr> "4", "4", "4", "4", "4", "4", "4", "4", "4", "1", ~<chr> "Auto Loan, Credit-Builder Loan, Personal Loa~<int> 3, -1, 3, 5, 6, 8, 3, 3, 7, 3, 3, 3, 3, ~<chr> "7", "", "7", "4", "", "4", "8", "6", "4", "~<chr> "11.27", "11.27", "_", "6.27", "11.27", "9.27~<dbl> 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 2, 2, ~<chr> "_", "Good", "Good", "Good", "Good", "Good", ~<chr> "809.98", "809.98", "809.98", "809.98", "809.~<dbl> 26.82262, 31.94496, 28.60935, 31.37786, 24.79~<chr> "22 Years and 1 Months", NA, "22 Years and 3 ~<chr> "No", "No", "No", "No", "No", "No", "No~<dbl> 49.57495, 49.57495, 49.57495, 49.57495, 49.57~<chr> "80.41529543900253", "118.28022162236736", "8~<chr> "High_spent_Small_value_payments", "Low_spent~<chr> "312.49408867943663", "284.62916249607184", "~<chr> "Good", "Good", "Good", "Good", "Good~
```

Se puede observar que muchas variables numéricas se cargan como tipo char por los caracteres sucios o

espacios en blanco que contienen, y algunas variables contienen ciertos niveles y deben ser transformadas a variables de tipo factor. Se procede entonces a realizar el cambio correspondiente al tipo de dato que cada variable debería tener. Para ello, en el caso de las numéricas, se limpian las cadenas de tal manera que solo se conserven dígitos del 0 al 9, puntos decimales y signos negativos (cuando aplique), de tal manera que se logre convertir a numérica sin problema. Algunas variables como Type\_of\_Loan que son multivaluadas no pueden ser transformadas a tipo factor directamente, por lo que momentáneamente se conserva como tipo char. Por otro lado, la variable Credit\_History\_Age contiene la antigüedad del historial crediticio en formato A años M meses, por lo que conviene solo obtener los años en tipo numérico, extrayendo los 2 primeros dígitos de la cadena y transformando de igual manera a variable numérica.

```
## Rows: 100,000
## Columns: 28
## $ ID <chr> "5634", "5635", "5636", "5637", "5638", "5639~<chr> "CUS_0xd40", "CUS_0xd40", "CUS_0xd40", "CUS_0~<fct> January, February, March, April, May, June, J~<chr> "Aaron Maashoh", "Aaron Maashoh", "Aaron Maas~<dbl> 23, 23, 500, 23, 23, 23, 23, 28, 28, 28, ~<chr> "821-00-0265", "821-00-0265", "821-00-0265", ~<fct> Scientist, Scientist, Scientist, Scientist, S~<dbl> 19114.12, 19114.12, 19114.12, 19114.12, 19114~<dbl> 1824.843, NA, NA, NA, 1824.843, NA, 1824.843,~<dbl> 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, ~<dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 1385, 4, 4, 4, ~<dbl> 3, 3, 3, 3, 3, 3, 3, 6, 6, 6, 6, 6, 6, 6, ~<dbl> 4, 4, 4, 4, 4, 4, 4, 1, 1, 1, 1, 1, 1, ~<chr> "Auto Loan, Credit-Builders Loan, Personal Loa~<dbl> 3, 1, 3, 5, 6, 8, 3, 3, 3, 7, 3, 3, 3, 3, 3, ~<dbl> 7, NA, 7, 4, NA, 4, 8, 6, 4, 1, 1, 3, 1, 0, 4~<dbl> 11.27, 11.27, NA, 6.27, 11.27, 9.27, 11.27, 1~<dbl> 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 2, 2, ~<fct> _, Good, Good, Good, Good, Good, Good, ~<dbl> 809.98, 809.98, 809.98, 809.98, 809.98, 809.9~<dbl> 26.82262, 31.94496, 28.60935, 31.37786, 24.79~<dbl> 22, NA, 22, 22, 22, 22, NA, 26, 26, 26, 2~<fct> No, N~<dbl> 49.57495, 49.57495, 49.57495, 49.57495, 49.57~<dbl> 80.41530, 118.28022, 81.69952, 199.45807, 41.~<fct> High_spent_Small_value_payments, Low_spent_La~<dbl> 312.4941, 284.6292, 331.2099, 223.4513, 341.4~<fct> Good, Good, Good, Good, Good, Good, Sta~
```

Se observa una estructura más limpia respecto al tipo de dato, pero nótese que seguimos con algunos valores nulos y atípicos tanto en las variables categóricas como en las numéricas, esto se puede observar de mejor manera en el resumen de los datos por variable.

Variable	Tipo	NAs	Cadenas.Vacias	Cadenas.Unicas	Espacios.Blanco
ID	character	0	0	100000	0
Customer_ID	character	0	0	12500	0
Name	character	0	9985	10140	0
SSN	character	0	0	12501	0
Type_of_Loan	character	0	11408	6261	0

Variable	Tipo	NAs	Media	Desv.Std
Age	numeric	0	119.51	684.76
Annual_Income	numeric	0	176415.70	1429618.05
Monthly_Inhand_Salary	numeric	15002	4194.17	3183.69
Num_Bank_Accounts	numeric	0	17.09	117.40
Num_Credit_Card	numeric	0	22.47	129.06
Interest_Rate	numeric	0	72.47	466.42
Num_of_Loan	numeric	0	10.76	61.79
Delay_from_due_date	numeric	0	21.10	14.82
Num_of_Delayed_Payment	numeric	7002	30.95	226.03
Changed_Credit_Limit	numeric	2091	10.39	6.79
Num_Credit_Inquiries	numeric	1965	27.75	193.18
Outstanding_Debt	numeric	0	1426.22	1155.13
Credit_Utilization_Ratio	numeric	0	32.29	5.12
Credit_History_Age	numeric	9030	17.97	8.32
Total_EMI_per_month	numeric	0	1403.12	8306.04
Amount_invested_monthly	numeric	4479	637.41	2043.32
Monthly_Balance	numeric	1200	3.03643724696356e+22	3.18129500838411e+24

Variable	Tipo	Min	1st.Qu	Mediana	3rd.Qu	Max
Age	numeric	14.00	25.00	34.00	42.00	8698.00
Annual_Income	numeric	7005.93	19457.50	37578.61	72790.92	24198062.00
Monthly_Inhand_Salary	numeric	303.65	1625.57	3093.74	5957.45	15204.63
Num_Bank_Accounts	numeric	0.00	3.00	6.00	7.00	1798.00
Num_Credit_Card	numeric	0.00	4.00	5.00	7.00	1499.00
Interest_Rate	numeric	1.00	8.00	13.00	20.00	5797.00
Num_of_Loan	numeric	0.00	2.00	3.00	6.00	1496.00
Delay_from_due_date	numeric	0.00	10.00	18.00	28.00	67.00
Num_of_Delayed_Payment	numeric	0.00	9.00	14.00	18.00	4397.00
Changed_Credit_Limit	numeric	-6.49	5.32	9.40	14.87	36.97
Num_Credit_Inquiries	numeric	0.00	3.00	6.00	9.00	2597.00
Outstanding_Debt	numeric	0.23	566.07	1166.15	1945.96	4998.07
Credit_Utilization_Ratio	numeric	20.00	28.05	32.31	36.50	50.00
Credit_History_Age	numeric	0.00	12.00	18.00	25.00	33.00
Total_EMI_per_month	numeric	0.00	30.31	69.25	161.22	82331.00
Amount_invested_monthly	numeric	0.00	74.53	135.93	265.73	10000.00
Monthly_Balance	numeric	0.01	270.11	336.74	470.33	3.33333333333333e+26

Variable	Tipo	NAs	Num.Clases	Conteo.Clase
Month	factor	0	8	Apr: 12500, Aug: 12500, Feb: 12500, Jan: 12500
Occupation	factor	0	16	____: 7062, Law: 6575, Arc: 6355, Eng: 6350
Credit_Mix	factor	0	4	Sta: 36479, Goo: 24337, __: 20195, Bad: 18989
Payment_of_Min_Amount	factor	0	3	Yes: 52326, No: 35667, NM: 12007
Payment_Behaviour	factor	0	7	Low: 25513, Hig: 17540, Low: 13861, Hig: 13721
Credit_Score	factor	0	3	Sta: 53174, Poo: 28998, Goo: 17828

Muchas variables como Monthly\_Inhand\_Salary, Num\_of\_Delayed\_Payment, Changed\_Credit\_Limit, Num\_Credit\_Inquiries, Credit\_History\_Age, Amount\_invested\_monthly y Monthly\_Balance tienen una cantidad considerable de NA's (entre 1% y 10%).

Es notorio que algunas variables tienen evidentes errores de captura que provocan outliers sin sentido como Age (Max = 8698), Num\_Bank\_Accounts (Max = 1798), Num\_Credit\_Card (Max = 1499), Interest\_Rate (Max = 5797), Num\_of\_Loan (Max = 1496), Num\_of\_Delayed\_Payment (Max = 4397), Num\_Credit\_Inquiries (Max = 2597), Monthly\_Balance (Max =  $3 \times 10^{26}$ ).

Otras variables tienen un valor máximo que difiere mucho del tercer cuartil pero que son posibles aunque extraños como Annual\_Income (Max = 24198062 con respecto a 3rd Qu. = 72791), Monthly\_Inhand\_Salary (Max = 15204 respecto a 3rd Qu. = 5957.4), Delay\_from\_due\_date (Max = 67 respecto a 3rd Qu. = 28.0), Total\_EMI\_per\_month (Max = 82331 respecto a 3rd Qu. = 161.22) y Amount\_invested\_monthly (Max = 10000 respecto a 3rd Qu. = 265.73).

Por otro lado, respecto a las variables categóricas y de tipo carácter, algunas tienen NA's silenciosos como Name y Type\_of\_Loan con espacios en blanco, SSN con caracteres tipo #F%\$D@\*&8, Occupation con \_\_\_\_\_, Credit\_Mix con \_\_, Payment\_of\_Min\_Amount con NM (Not Mentioned) y Payment\_Behaviour con !@9#%8.

La estrategia para limpiar los datos fue utilizar el hecho de que cada 8 registros tenemos variables que tienen valores idénticos, similares, o con varianza baja debido a que se tratan de un mismo cliente, por lo que se realizó una agrupación por cliente y se obtuvo la moda para variables categóricas y la mediana para variables numéricas para sustituir los NA's explícitos e implícitos y los valores atípicos. En casos muy particulares donde la moda o mediana fuera el valor atípico o NA (por ejemplo, que en Occupation de los 8 registros 6 tuvieran \_\_\_\_\_ y solo 2 la ocupación real, o que en age la mediana superara el valor 100), se tomaba entonces el segundo valor más repetido o la mediana general de los datos para imputar.

En particular, los criterios específicos que se tomaron para limpiar cada una de las variables fueron los siguientes:

**ID.** Sin limpieza.

**Customer\_ID.** Sin limpieza.

**Name.** Sustituir aquellos valores con cadenas vacías.

**Month.** Sin limpieza.

**Age.** Cambiar aquellos valores que tuviesen un valor menor a 0 o mayor a 100 por la mediana.

**SSN.** Sustituir aquellos valores con la cadena #F%\$D@\*&8 por la moda.

**Occupation.** Sustituir aquellos valores con la cadena \_\_\_\_\_ por la moda.

**Annual\_Income.** Sustituir aquellos valores mayores a 200000 con la mediana.

**Monthly\_Inhand\_Salary.** Sustituir los NA's con la mediana.

**Num\_Bank\_Accounts.** Sustituir aquellos valores mayores a 11 con la mediana.

**Num\_Credit\_Card.** Sustituir aquellos valores mayores a 11 con la mediana.

**Interest\_Rate.** Sustituir aquellos valores mayores a 34 con la mediana.

**Num\_of\_Loan.** Sustituir aquellos valores mayores a 9 con la mediana..

**Type\_of\_Loan.** Sin limpieza (por el momento).

**Delay\_from\_due\_date.** Sin limpieza.

**Num\_of\_Delayed\_Payment.** Sustituir aquellos valores mayores a 28 con la mediana.

**Changed\_Credit\_Limit.** Sustituir los NA's con la mediana.

**Num\_Credit\_Inquiries.** Sustituir los NA's y aquellos valores mayores a 17 con la mediana.

**Credit\_Mix.** Sustituir aquellos valores con la cadena \_ por la moda.

**Outstanding\_Debt.** Sin Limpieza.

**Credit\_Utilization\_Ratio.** Sin Limpieza.

**Credit\_History\_Age.** Sustituir los NA's con la mediana.

**Payment\_of\_Min\_Amount.** Sustituir aquellos valores con la cadena NM por la moda.

**Total\_EMI\_per\_month.** Sustituir aquellos valores mayores a 1800 con la mediana.

**Amount\_invested\_monthly.** Sustituir los NA's y aquellos valores mayores a 2000 con la mediana.

**Payment\_Behaviour.** Sustituir aquellos valores con la cadena !@9#%8 por la moda.

**Monthly\_Balance.** Sustituir los NA's y aquellos valores mayores a 2000 con la mediana.

**Credit\_Score.** Sin Limpieza.

Una vez realizada la limpieza de los datos, obtenemos la siguiente estructura.

```
## Rows: 100,000
## Columns: 28
## $ ID                               <chr> "5634", "5635", "5636", "5637", "5638", "5639~"
## $ Customer_ID                      <chr> "CUS_0xd40", "CUS_0xd40", "CUS_0xd40", "CUS_0~"
## $ Month                            <fct> January, February, March, April, May, June, J~
## $ Name                             <chr> "Aaron Maashoh", "Aaron Maashoh", "Aaron Maas~"
## $ Age                              <dbl> 23, 23, 23, 23, 23, 23, 23, 28, 28, 28, 2~
## $ SSN                             <chr> "821-00-0265", "821-00-0265", "821-00-0265", ~
## $ Occupation                       <fct> Scientist, Scientist, Scientist, Scientist, S~
## $ Annual_Income                    <dbl> 19114.12, 19114.12, 19114.12, 19114.12, 19114~
## $ Monthly_Inhand_Salary           <dbl> 1824.843, 1824.843, 1824.843, 1824.843, 1824.~
## $ Num_Bank_Accounts                <dbl> 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, ~
## $ Num_Credit_Card                  <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, ~
## $ Interest_Rate                   <dbl> 3, 3, 3, 3, 3, 3, 6, 6, 6, 6, 6, 6, ~
## $ Num_of_Loan                      <dbl> 4, 4, 4, 4, 4, 4, 1, 1, 1, 1, 1, 1, ~
## $ Type_of_Loan                     <chr> "Auto Loan, Credit-Builders Loan, Personal Loa~
## $ Delay_from_due_date              <dbl> 3, 1, 3, 5, 6, 8, 3, 3, 7, 3, 3, 3, 3, ~
## $ Num_of_Delayed_Payment          <dbl> 7.0, 6.5, 7.0, 4.0, 6.5, 4.0, 8.0, 6.0, 4.0, ~
## $ Changed_Credit_Limit            <dbl> 11.27, 11.27, 11.27, 6.27, 11.27, 9.27, 11.27~
## $ Num_Credit_Inquiries            <dbl> 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 2, 2, ~
## $ Credit_Mix                       <fct> Good, Good, Good, Good, Good, Good, Good, Goo~
## $ Outstanding_Debt                <dbl> 809.98, 809.98, 809.98, 809.98, 809.98, 809.9~
## $ Credit_Utilization_Ratio        <dbl> 26.82262, 31.94496, 28.60935, 31.37786, 24.79~
## $ Credit_History_Age               <dbl> 22, 22, 22, 22, 22, 22, 26, 26, 26, 2~
## $ Payment_of_Min_Amount            <fct> No, No, No, No, No, No, No, No, No, N~
## $ Total_EMI_per_month              <dbl> 49.57495, 49.57495, 49.57495, 49.57495, 49.57~
## $ Amount_invested_monthly         <dbl> 80.41530, 118.28022, 81.69952, 199.45807, 41.~
## $ Payment_Behaviour                <fct> High_spent_Small_value_payments, Low_spent_La~
## $ Monthly_Balance                 <dbl> 312.4941, 284.6292, 331.2099, 223.4513, 341.4~
## $ Credit_Score                      <fct> Good, Good, Good, Good, Good, Good, Sta~
```

La estructura de los datos ya considera el tipo de variable adecuado para casi todas las variables de la base de datos (excepto Type\_of\_Loan). Revisando el resumen de los datos obtenemos lo siguiente.

Variable	Tipo	NAs	Cadenas.Vacias	Cadenas.Unicas	Espacios.Blanco
ID	character	0	0	100000	0
Customer_ID	character	0	0	12500	0
Name	character	0	0	10139	0
SSN	character	0	0	12501	0
Type_of_Loan	character	0	11408	6261	0

Variable	Tipo	NAs	Media	Desv.Std
Age	numeric	0	33.31	10.77
Annual_Income	numeric	0	50505.12	38299.42
Monthly_Inhand_Salary	numeric	0	4198.49	3187.49
Num_Bank_Accounts	numeric	0	5.37	2.59
Num_Credit_Card	numeric	0	5.53	2.07
Interest_Rate	numeric	0	14.53	8.74
Num_of_Loan	numeric	0	3.53	2.45
Delay_from_due_date	numeric	0	21.10	14.82
Num_of_Delayed_Payment	numeric	0	13.34	6.26
Changed_Credit_Limit	numeric	0	10.39	6.78
Num_Credit_Inquiries	numeric	0	5.78	3.86
Outstanding_Debt	numeric	0	1426.22	1155.13
Credit_Utilization_Ratio	numeric	0	32.29	5.12
Credit_History_Age	numeric	0	17.97	8.32
Total_EMI_per_month	numeric	0	108.77	192.51
Amount_invested_monthly	numeric	0	196.52	210.63
Monthly_Balance	numeric	0	403.49	214.42

Variable	Tipo	Min	1st.Qu	Mediana	3rd.Qu	Max
Age	numeric	14.00	24.00	33.00	42.00	100.00
Annual_Income	numeric	7005.93	19342.97	36999.71	71683.47	179987.28
Monthly_Inhand_Salary	numeric	303.65	1626.76	3095.98	5961.64	15204.63
Num_Bank_Accounts	numeric	0.00	3.00	5.00	7.00	11.00
Num_Credit_Card	numeric	0.00	4.00	5.00	7.00	11.00
Interest_Rate	numeric	1.00	7.00	13.00	20.00	34.00
Num_of_Loan	numeric	0.00	2.00	3.00	5.00	9.00
Delay_from_due_date	numeric	0.00	10.00	18.00	28.00	67.00
Num_of_Delayed_Payment	numeric	0.00	9.00	14.00	18.00	28.00
Changed_Credit_Limit	numeric	-6.49	5.34	9.40	14.85	36.97
Num_Credit_Inquiries	numeric	0.00	3.00	5.00	8.00	17.00
Outstanding_Debt	numeric	0.23	566.07	1166.15	1945.96	4998.07
Credit_Utilization_Ratio	numeric	20.00	28.05	32.31	36.50	50.00
Credit_History_Age	numeric	0.00	12.00	18.00	25.00	33.00
Total_EMI_per_month	numeric	0.00	29.26	66.46	147.37	21627.12
Amount_invested_monthly	numeric	0.00	74.40	131.06	237.91	10000.00
Monthly_Balance	numeric	0.01	270.32	337.27	471.93	1602.04

Variable	Tipo	NAs	Num.Clases	Conteo.Clase
Month	factor	0	8	Apr: 12500, Aug: 12500, Feb: 12500, Jan: 12500
Occupation	factor	0	15	Law: 7096, Eng: 6864, Arc: 6824, Mec: 6776
Credit_Mix	factor	0	3	Sta: 45848, Goo: 30384, Bad: 23768
Payment_of_Min_Amount	factor	0	2	Yes: 59432, No: 40568
Payment_Behaviour	factor	0	6	Low: 27767, Hig: 19366, Hig: 15348, Low: 14621
Credit_Score	factor	0	3	Sta: 53174, Poo: 28998, Goo: 17828

El resumen de los datos también muestra que la base de datos se limpió correctamente, ya no hay datos NA's explícitos, NA's implícitos en las variables categóricas, ni outliers en las variables numéricas. El último paso para tener una base de datos completamente limpia y lista para el análisis es considerar las variables importantes, transformar la variable Type\_of\_Loan, y convertir las variables categóricas a variables tipo dummmie.

Las variables ID, Customer\_ID, Month, Name y SSN son variables específicas del cliente y la fecha, las cuales son completamente independientes a como es el comportamiento financiero del cliente, por ser identificadores únicos del cliente y tiempo, por lo que no se consideran para el análisis.

Por otro lado, de las variables categóricas, las variables Credit\_Mix y Payment\_of\_Min\_Amount son del tipo categórica ordinal, por lo que pueden ser codificadas de manera numérica. Por tanto, para el caso de Credit\_Mix se transformó a los niveles Good, Standard y Poor a los números 3,2 y 1 respectivamente, mientras que para Payment\_of\_Min\_Amount con niveles Yes y No se codificó como 1 y 0 respectivamente.

Las variables Occupation y Payment\_Behaviour son de tipo categórica nominal, por lo que en este caso se considera una nueva variable por nivel con valor 1 si el cliente tiene ese nivel y 0 si no.

Finalmente, la variable Type\_of\_Loan al ser de tipo mulivaluada se separa de igual manera como variables dummies. Para ello a las cadenas del estilo “Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan” se les aplica un separador, usando “,” o “and” como separadores, y se eliminan los valores “Not Specified” o las cadenas vacías. Así, se tiene la estructura final siguiente.

```
## Rows: 100,000
## Columns: 49
## $ Age <dbl> 23, 23, 23, 23, 23, 23, 23, 28, 2~ 
## $ Annual_Income <dbl> 19114.12, 19114.12, 19114.12, 19114.1~ 
## $ Monthly_Inhand_Salary <dbl> 1824.843, 1824.843, 1824.843, 1824.84~ 
## $ Num_Bank_Accounts <dbl> 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2~ 
## $ Num_Credit_Card <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4~ 
## $ Interest_Rate <dbl> 3, 3, 3, 3, 3, 3, 3, 6, 6, 6, 6, 6~ 
## $ Num_of_Loan <dbl> 4, 4, 4, 4, 4, 4, 4, 1, 1, 1, 1, 1~ 
## $ Delay_from_due_date <dbl> 3, 1, 3, 5, 6, 8, 3, 3, 3, 7, 3, 3, 3~ 
## $ Num_of_Delayed_Payment <dbl> 7.0, 6.5, 7.0, 4.0, 6.5, 4.0, 8.0, 6.~ 
## $ Changed_Credit_Limit <dbl> 11.27, 11.27, 11.27, 6.27, 11.27, 9.2~ 
## $ Num_Credit_Inquiries <dbl> 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 2~ 
## $ Credit_Mix <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3~ 
## $ Outstanding_Debt <dbl> 809.98, 809.98, 809.98, 809.98, 809.9~ 
## $ Credit_Utilization_Ratio <dbl> 26.82262, 31.94496, 28.60935, 31.3778~ 
## $ Credit_History_Age <dbl> 22, 22, 22, 22, 22, 22, 26, 2~ 
## $ Payment_of_Min_Amount <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~ 
## $ Total_EMI_per_month <dbl> 49.57495, 49.57495, 49.57495, 49.5749~ 
## $ Amount_invested_monthly <dbl> 80.41530, 118.28022, 81.69952, 199.45~ 
## $ Monthly_Balance <dbl> 312.4941, 284.6292, 331.2099, 223.451~ 
## $ Accountant <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~ 
## $ Architect <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```

## $ Developer <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Doctor <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Engineer <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Entrepreneur <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Journalist <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Lawyer <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Manager <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Mechanic <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Media_Manager <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Musician <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Scientist <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0~  

## $ Teacher <dbl> 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1~  

## $ Writer <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ `Auto Loan` <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0~  

## $ `Credit-Builder Loan` <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~  

## $ `Personal Loan` <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0~  

## $ `Home Equity Loan` <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0~  

## $ `Mortgage Loan` <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ `Student Loan` <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ `Debt Consolidation Loan` <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ `Payday Loan` <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ High_spent_Large_value_payments <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0~  

## $ High_spent_Medium_value_payments <dbl> 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0~  

## $ High_spent_Small_value_payments <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Low_spent_Large_value_payments <dbl> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ Low_spent_Medium_value_payments <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0~  

## $ Low_spent_Small_value_payments <dbl> 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0~  

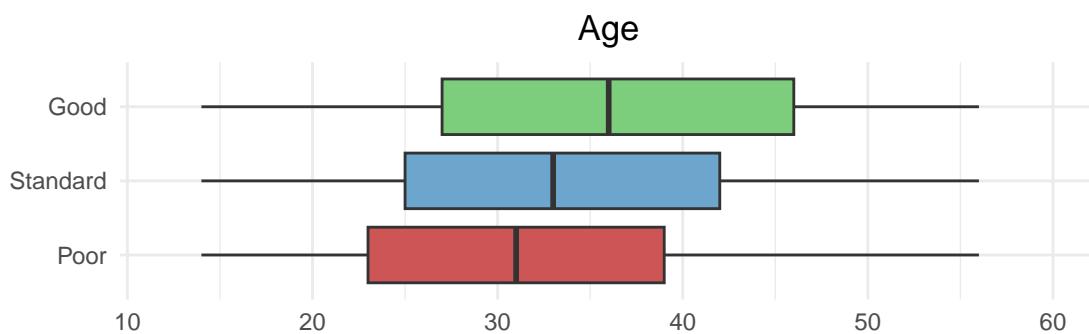
## $ Credit_Score <fct> Good, Good, Good, Good, Good, Good, G~

```

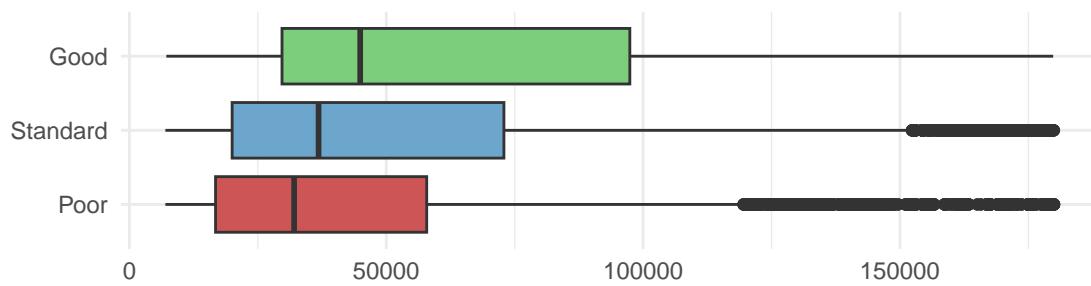
De esta manera, obtenemos finalmente un dataframe con 49 variables, de las cuales 19 son numéricas, 29 son variables dummies y 1 (Credit\_Score) es la variables objetivo.

### 3 Estadística Descriptiva de los datos respecto a la variable Credit Score

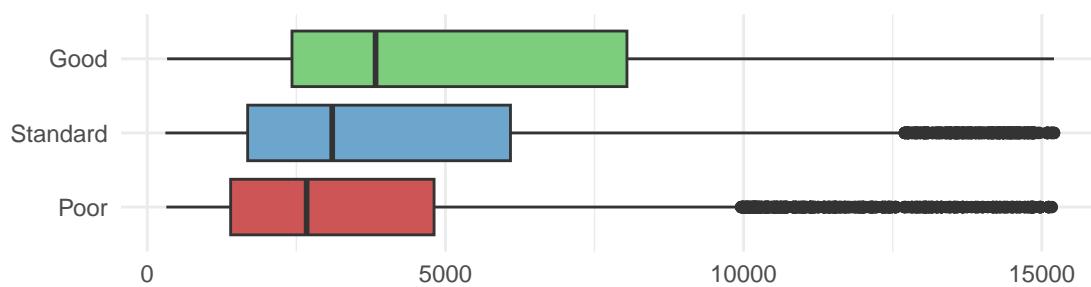
Los siguientes boxplots muestran el comportamiento de cada una de las variables numéricas de la base de datos agrupadas por Credit Score.



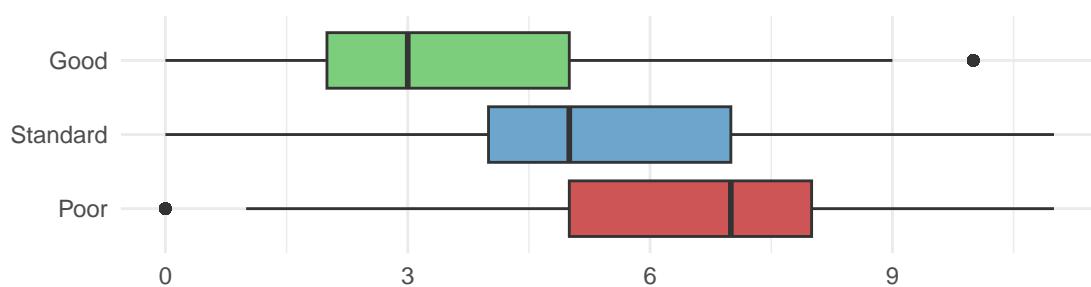
Annual\_Income



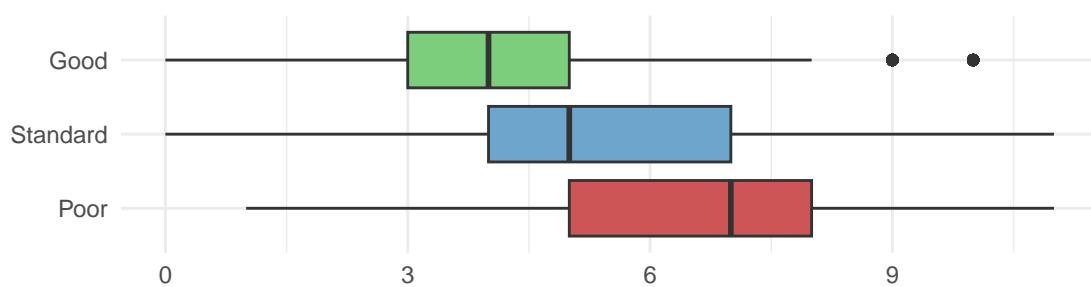
Monthly\_Inhand\_Salary

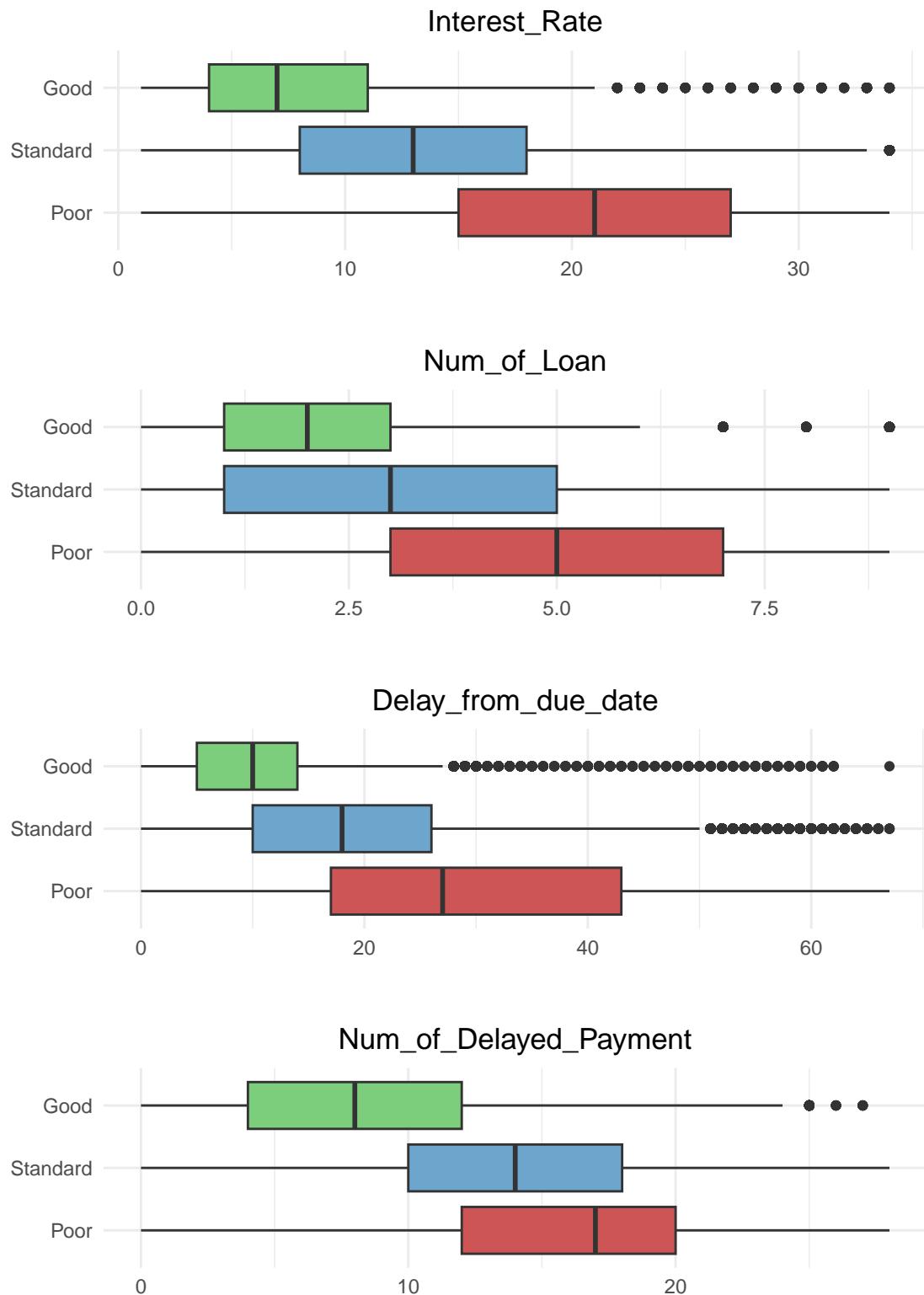


Num\_Bank\_Accounts

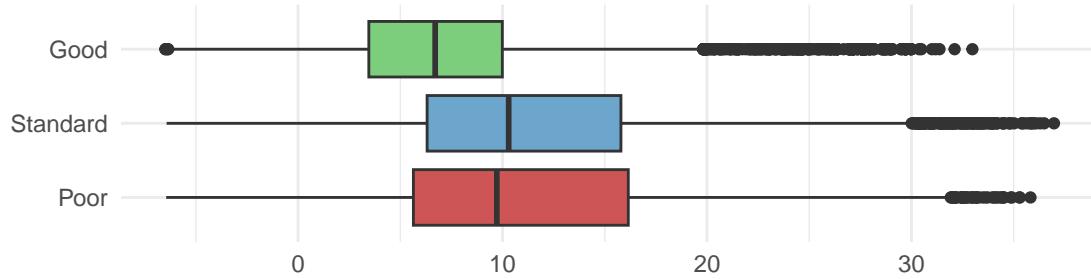


Num\_Credit\_Card

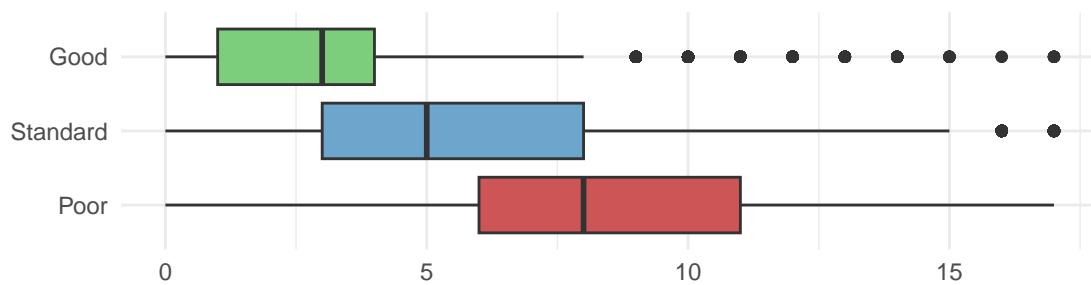




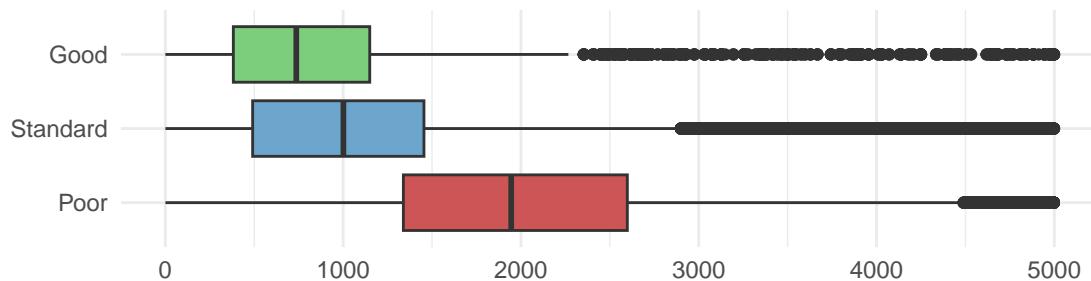
### Changed\_Credit\_Limit



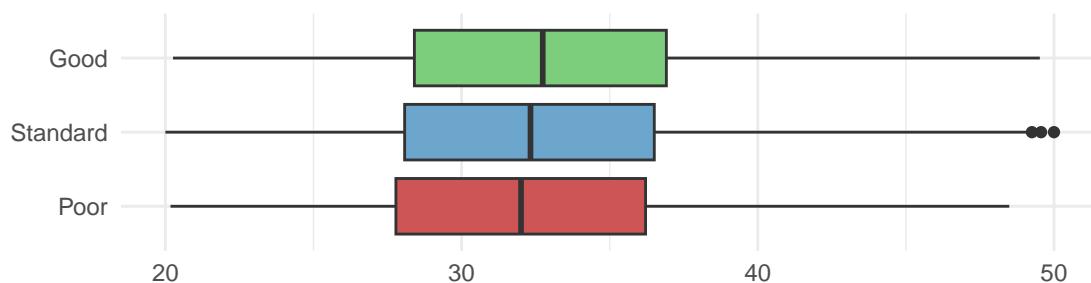
### Num\_Credit\_Inquiries

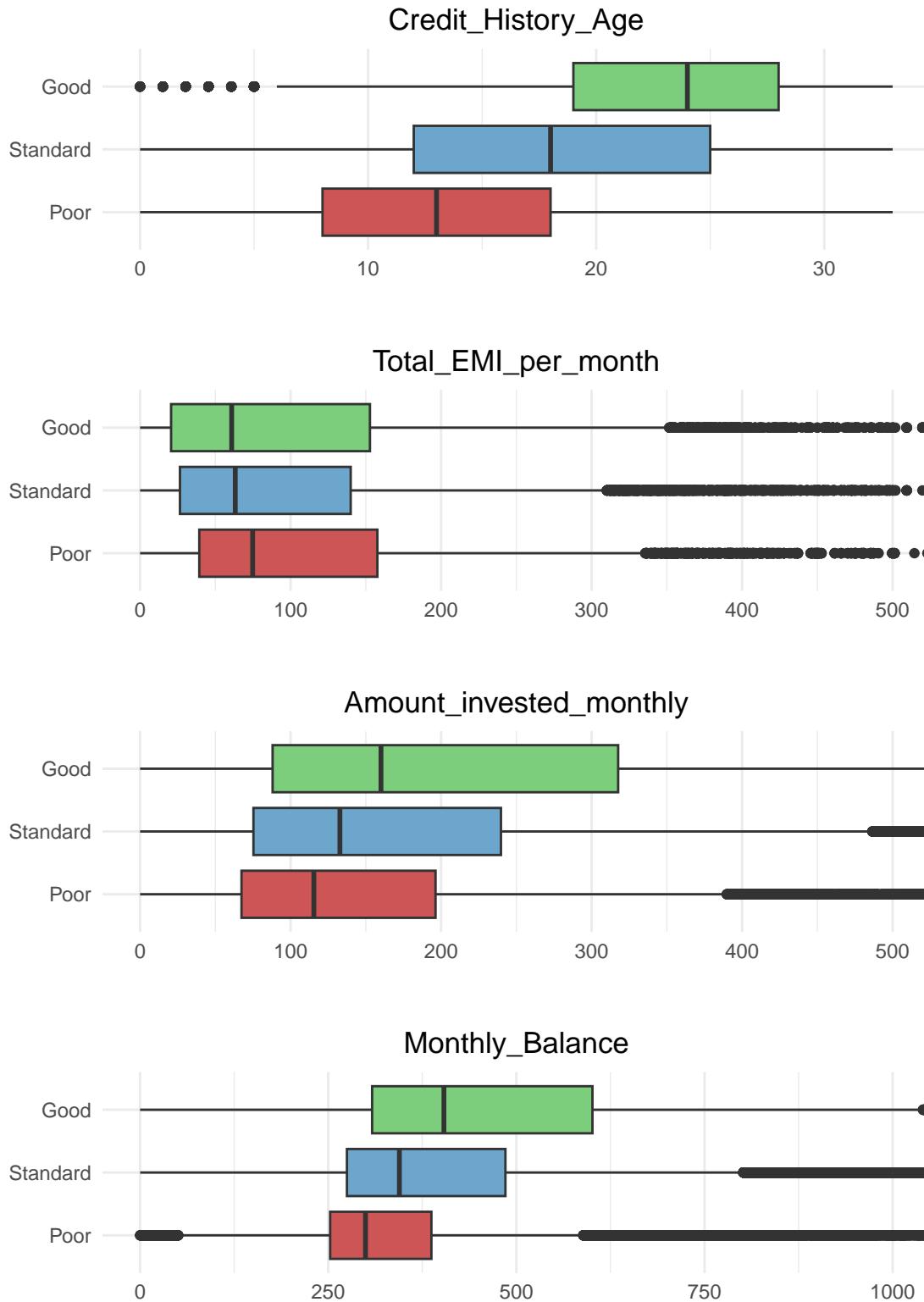


### Outstanding\_Debt



### Credit\_Utilization\_Ratio



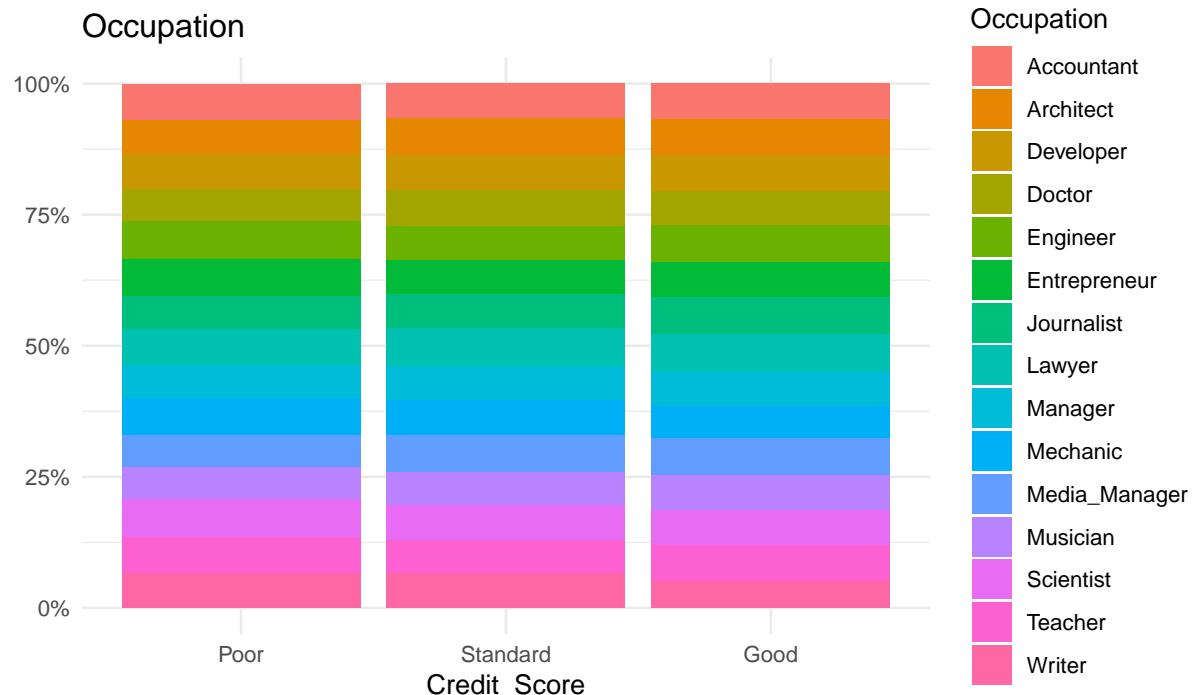


Se observa como las variables Age, Annual\_Income, Monthly\_Inhand\_Salary, Credit\_History\_Age, Amount\_Invested\_Monthly y Monthly\_Balance tienen una relación proporcional respecto a la puntuación crediticia, dónde el grupo Good posee los quantiles más altos mientras que el grupo Poor tiene los valores

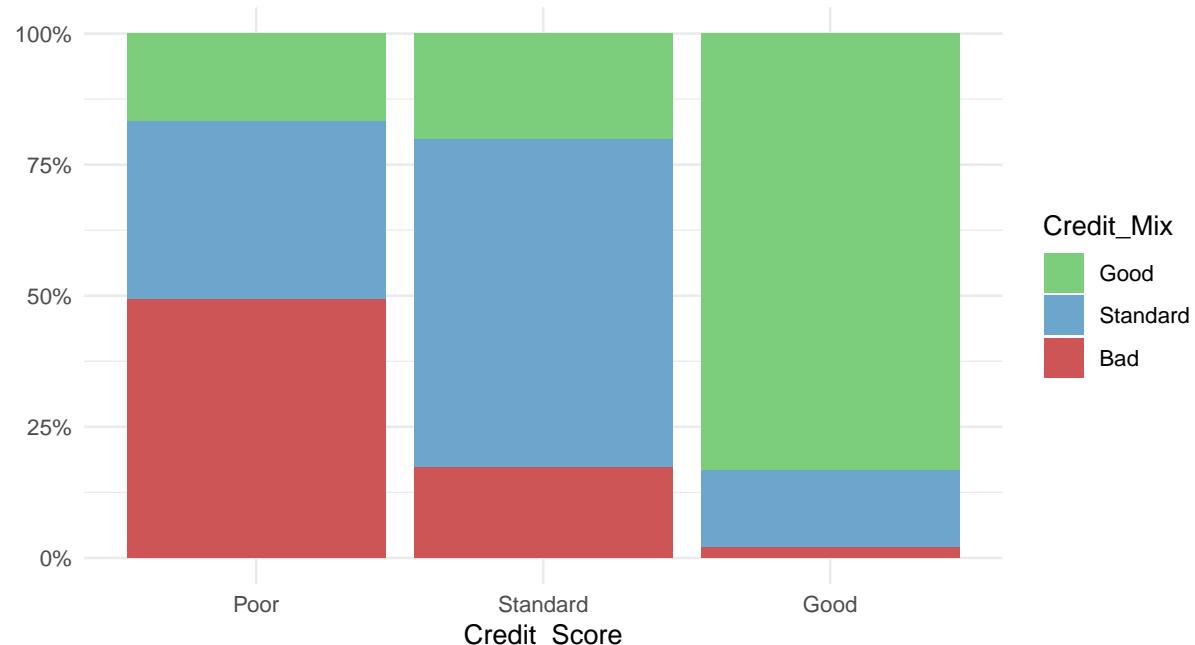
más bajos, y el grupo Standard es un punto intermedio entre ambos. De estas variables, Credit\_History\_Age muestra la diferenciación más grande entre clases, con una mediana de alrededor de 23 años de historial crediticio, mientras que los grupos Standard y Poor tienen una mediana de 17 años y 12 años respectivamente, y Age al ser una variable relacionada muestra un patrón similar aunque no tan predominante, ya que una persona puede tener una edad avanzada pero no necesariamente haber desarrollado historial, pero evidentemente los del grupo Good al tener una mediana de 23 años de historial crediticio deben ser personas de edades más avanzadas. Por otro lado, las variables Annual\_Income, Monthly\_Inhand\_Salary, Amount\_Invested\_Monthly y Monthly\_Balance se relacionan con el perfil económico del cliente, teniendo entonces que aquellos que suelen tener una buena puntuación crediticia, además de un largo historial, poseen una solvencia económica elevada que permite el pago de los préstamos y créditos que solicitan al banco además de lograr más inversiones.

Por otro lado, tenemos que Num\_Bank\_Accounts, Num\_Credit\_Card, Interest\_Rate, Num\_of\_Loan, Delay\_from\_due\_date, Num\_of\_Delayed\_Payment, Num\_Credit\_Inquiries y Outstanding\_Debt muestran una relación inversamente proporcional al Credit Score, donde a mayores valores de estas variables el puntaje es más bajo. Estas variables miden la responsabilidad que tiene el cliente respecto a sus finanzas y el pago de sus préstamos y créditos, y en este caso si se muestran diferencias muy significativas principalmente entre los grupos Poor y Good con respecto a estas variables, sobretodo en aquellas que están relacionadas directamente con los préstamos como son Interest\_Rate, Num\_of\_Loan, Delay\_from\_due\_date, Num\_of\_Delayed\_Payment y Outstanding\_Debt, donde el 1er cuantil en el grupo Poor es incluso mayor o igual al 3er cuantil del grupo Good. Esta diferencia se vuelve más grande ya que, a diferencia de las variables que benefician al grupo Good, los préstamos, retrasos, deuda y tasas de interés aumentan con mucha facilidad si el cliente no es responsable con los pagos de los préstamos, a diferencia de la posición económica y la antigüedad del historial, los cuales toman mucho tiempo aumentar.

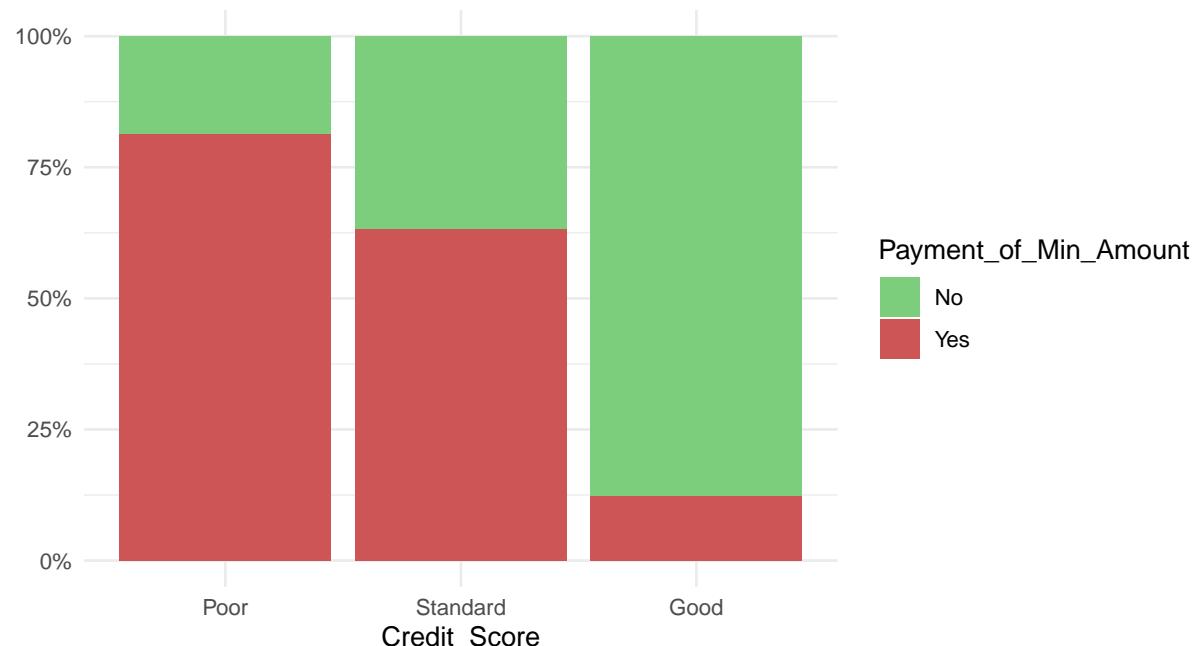
Respecto a las variables categóricas, podemos observar las proporciones de cada una por nivel de Credit Score en los siguientes gráficos.

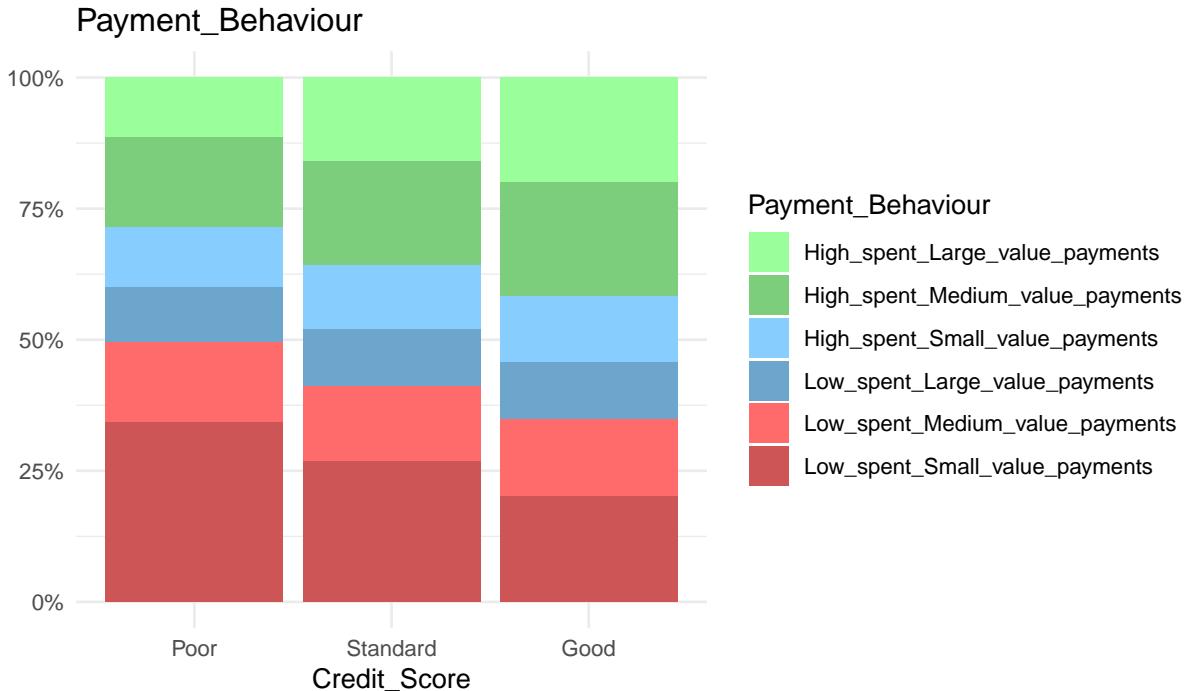


**Credit\_Mix**



**Payment\_of\_Min\_Amount**





La profesión de los clientes no tiene alguna distribución preferencial por nivel de Credit Score, se observa que en general esta variable esta equilibrada en las 3 clases por lo que podríamos asumir que no hay impacto de la profesión en la puntuación crediticia. Sin embargo, las otras 3 variables Credit\_Mix, Payment\_of\_Min\_Amount y Payment\_Behaviour si tienen una cierta tendencia para cada nivel.

Credit\_Mix tiene una relación sumamente directa con Credit\_Score, justamente tenemos que la proporción más dominante en cada nivel de Credit\_Score es la misma que en Credit\_Mix (Bad-Poor, Standard-Standard y Good-Good), de donde el grupo Good es el que muestra la relación más fuerte con su análogo de Credit\_Mix.

Respecto a Payment\_of\_Min\_Amount, se observa que de aquellos del grupo Poor, la mayoría suelen apenas cubrir el pago mínimo requerido en sus préstamos, mientras que los del grupo Good la mayoría hacen depósitos mayores al mínimo requerido, lo cual es congruente con el hecho de que el grupo Good, al tener un nivel económico más alto y deudas más bajas, tienen más facilidad de cubrir sus deudas a diferencia del grupo Poor, que suelen tener un nivel económico bajo y deudas y tasas de interés altas.

Finalmente, la variable Payment\_Behaviour, a diferencia de Credit\_Mix y Payment\_of\_Min\_Amount, no muestra una relación sumamente predominante en cada clase, sin embargo, si es notorio que ciertos comportamientos de pago se asocian ligeramente a cada nivel de Credit Score, ya que los clientes con un Credit Score Good suelen tener gastos elevados pero que a la vez se cubren con pagos de gran cantidad, mientras que los del grupo Poor realizan compras de un valor más bajo y aún así realizan pagos pequeños para pagarlas. Esto tiene la misma explicación que en Payment\_of\_Min\_Amount, aunque más relacionado al nivel económico del cliente y la manera en que el cliente paga cierto tipo de gastos.

En general, se puede concluir que cada puntuación crediticia tiene comportamientos destacados en la mayoría de las variables de la base de datos, que se pueden resumir como:

- *Antigüedad del cliente y su historial* (Good tiene más años y Poor pocos años)
- *Posición económica del cliente* (Good tiene una posición alta y Poor una posición baja)
- *Deudas y préstamos del cliente* (Poor tiene un valor elevado y Good un valor bajo)
- *Conducta financiera del cliente* (Poor muestra una conducta irresponsable y Good una conducta responsable)

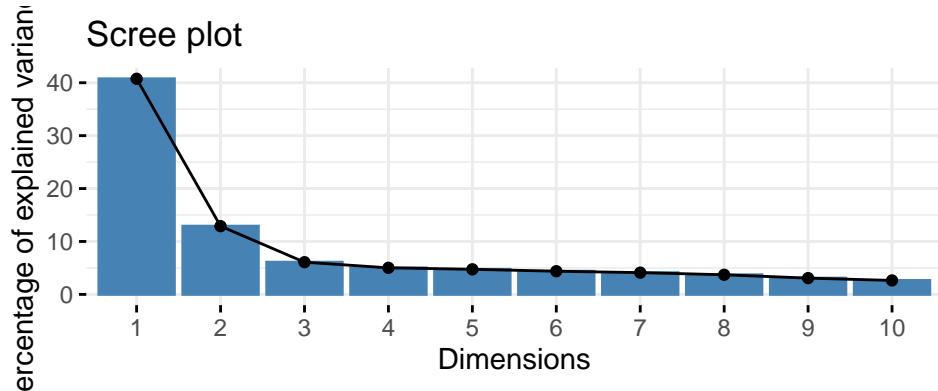
## 4 Aprendizaje No Supervisado de los datos

A pesar de ya contar con una variable clasificadora y por tanto, ser un problema de aprendizaje supervisado, en esta sección se busca explorar que tan separables o distinguibles son las clases del Credit Score (Good, Standard y Poor) para algoritmos no supervisados como PCA, K-Means y Métodos Jerárquicos.

### 4.1 Análisis de Componentes Principales (PCA)

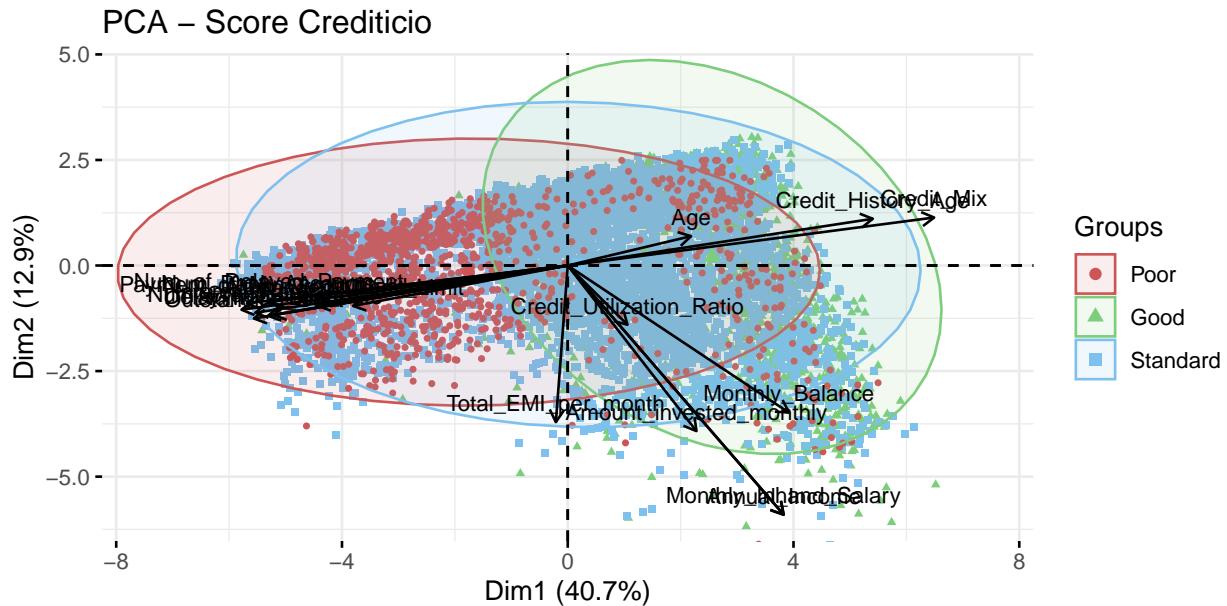
Se ajustó un PCA considerando solamente las variables de tipo numérico de la base de datos, con estandarización de los datos, debido a que si se realizaba el PCA sin ese preprocesamiento de los datos, la variable monetaria Annual\_Income era la dominante en el PCA.

El siguiente gráfico muestra la cantidad de varianza explicada por los primeros 10 componentes.



El primer componente principal representa poco más del 40% de la varianza explicada mientras que el segundo componente suma alrededor de un 13%, teniendo un total de 53% de varianza explicada por ambos, lo cuál es un buen número si nuestro objetivo es visualizar en menos dimensiones a las observaciones y las variables.

El siguiente biplot resume de manera gráfica como es que interactúan cada uno de los grupos de Credit Score con los primeros 2 componentes y las variables de la base de datos.

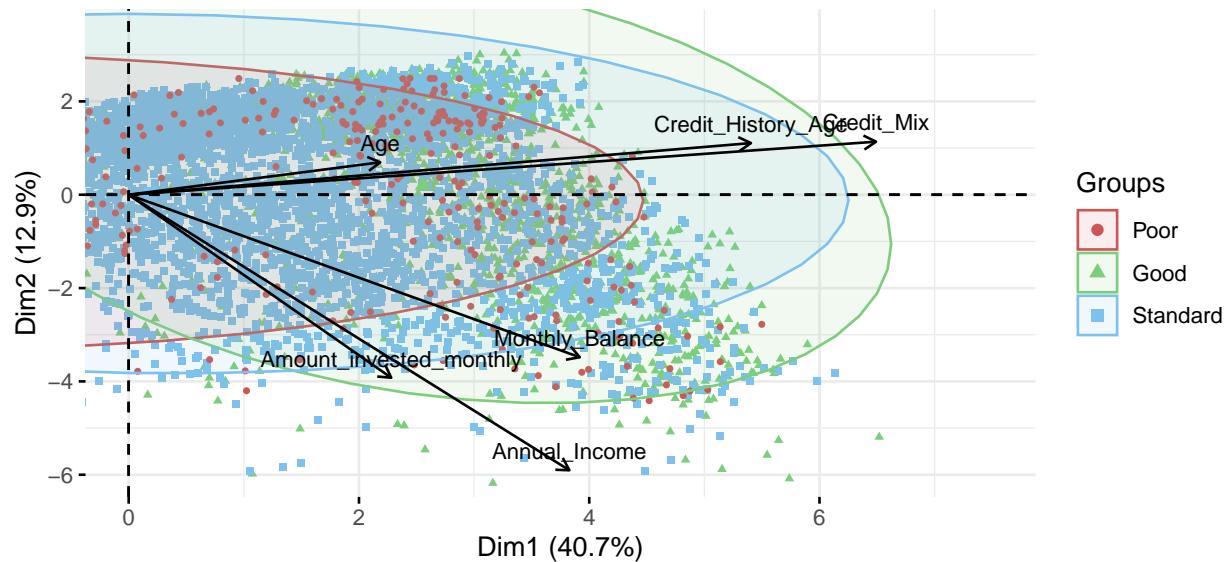


Se observa una relación muy importante entre el Credit Score y el componente principal 1, debido a que

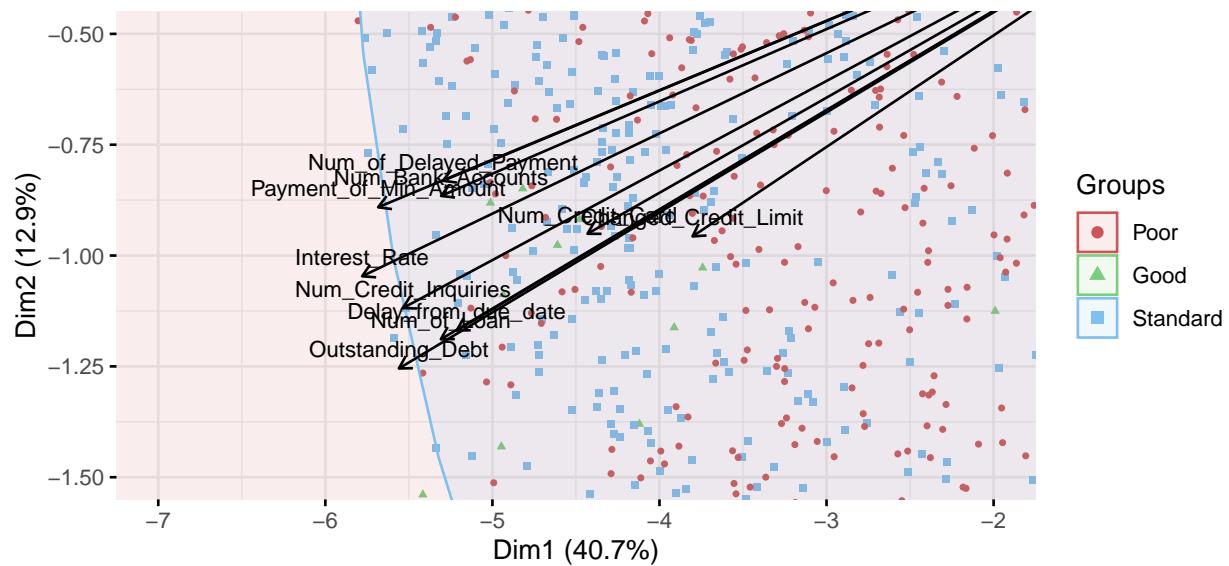
aquellos con una puntuación crediticia Poor se encuentran mayoritariamente en el valor negativo del PC1, los del grupo Good por el contrario se posicionan en los valores positivos, y los del grupo Standard están alrededor del 0, aunque este ultimo grupo también se mezcla considerablemente con las otras 2 puntuaciones. Del biplot podemos concluir que los grupos no son fácilmente separables, ya que incluso varias observaciones del grupo Poor se encuentran en la zona del grupo Good, y como ya se mencionó antes, los del grupo Standard también se encuentran en zonas tanto del grupo Poor como del grupo Good. Sin embargo, los del grupo Good si se encuentran mayoritariamente ubicados en una zona específica, y no es tan evidente que se fuguen en los otros 2 grupos, sobre todo en el grupo Poor.

Respecto a las variables, los siguientes biplots muestran a detalle cuáles son las variables relacionadas con el grupo Good y cuáles con el grupo Poor.

PCA – Credit Score Good



PCA – Credit Score Poor



En esta tabla se muestra más detallado y en orden descendente a las variables de acuerdo a su correlación

con el componente principal 1 y el grupo al que se relacionan.

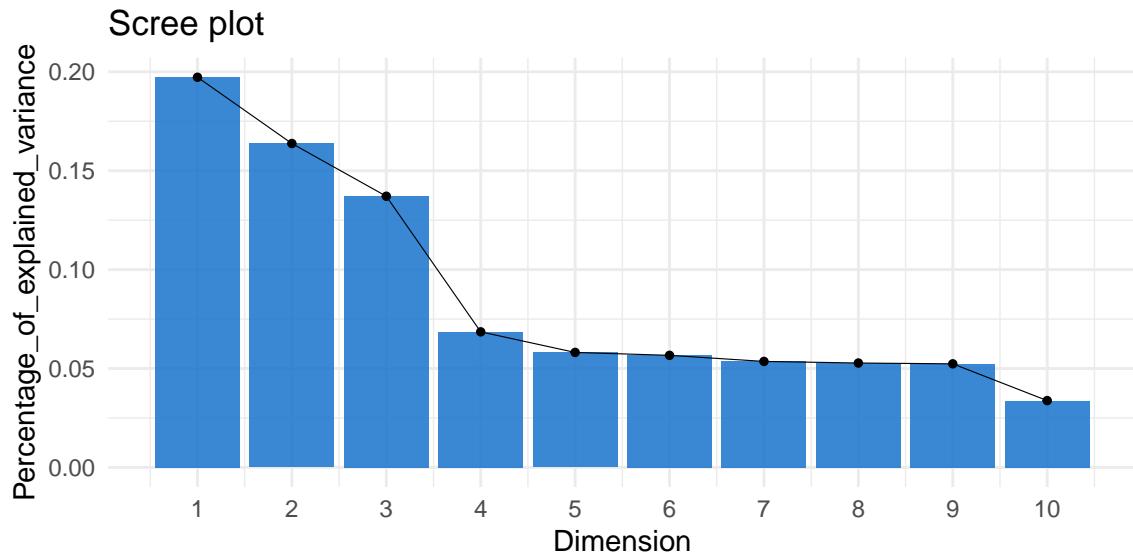
	PCA1	Grupo
Credit_Mix	0.3245657	Good
Credit_History_Age	0.2702270	Good
Monthly_Balance	0.1958764	Good
Annual_Income	0.1913717	Good
Monthly_Inhand_Salary	0.1911076	Good
Amount_invested_monthly	0.1140126	Good
Age	0.1092112	Good
Credit_Utilization_Ratio	0.0528592	Standard
Total_EMI_per_month	-0.0105341	Standard
Changed_Credit_Limit	-0.1901020	Poor
Num_Credit_Card	-0.2215324	Poor
Delay_from_due_date	-0.2606807	Poor
Num_of_Delayed_Payment	-0.2647790	Poor
Num_Bank_Accounts	-0.2653535	Poor
Num_of_Loan	-0.2654240	Poor
Num_Credit_Inquiries	-0.2767742	Poor
Outstanding_Debt	-0.2779787	Poor
Payment_of_Min_Amount	-0.2841970	Poor
Interest_Rate	-0.2889749	Poor

Similar a lo visto en la sección de estadística descriptiva, las variables Age y Credit\_History\_Age muestran como aquellos con puntaje Good suelen ser quienes tienen más edad, y por tanto, un historial más antiguo. Monthly\_Balance, Annual\_Income, Monthly\_Inhand\_Salary, y Amount\_invested\_monthly muestran que aquellos con un mayor capital tanto en sus ingresos, sus cuentas de bancos y el dinero que invierten son los que suelen ser considerados también con una puntuación alta. Por tanto, este grupo se destaca por tener mayor antigüedad y una estabilidad financiera elevada.

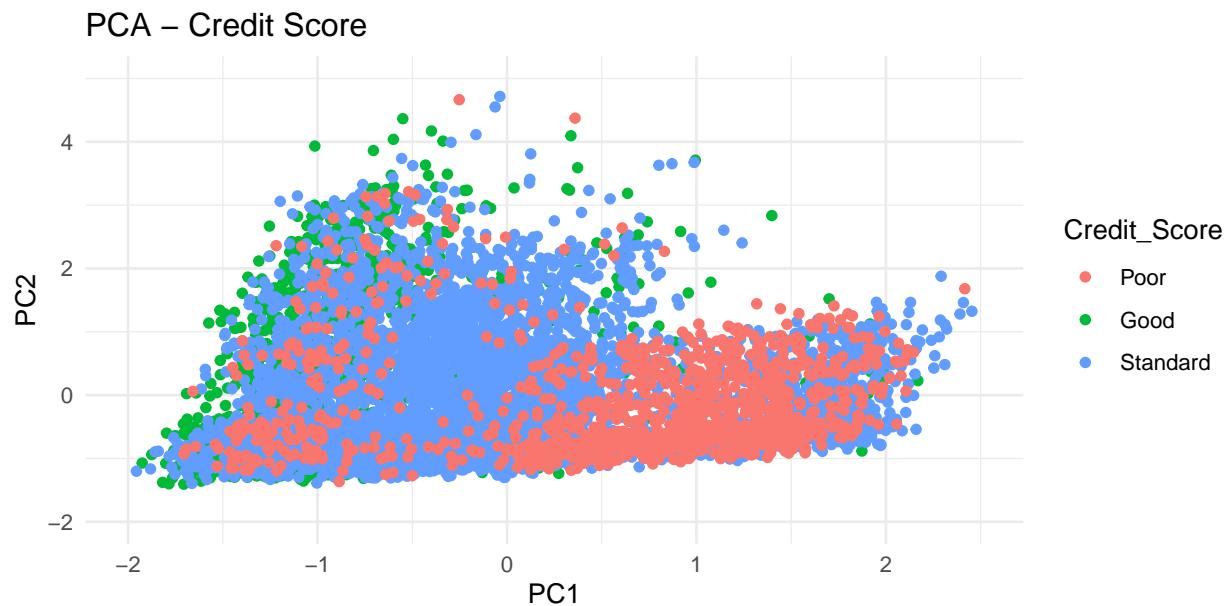
Por otro lado, aquellos del grupo Poor están sumamente relacionados con las variables que implican muchos créditos y préstamos (Num\_Credit\_Card, Changed\_Credit\_Limit, Num\_Credit\_Inquiries, Num\_of\_Loan, Num\_Bank\_Accounts), deudas altas (Outstanding\_Debt, Interest\_Rate) y pagos atrasados (Delay\_from\_due\_date, Num\_of\_Delayed\_Payment, Payment\_of\_Min\_Amount). Así, este grupo es identificable cuando se tienen muchos préstamos pedidos al banco, una deuda alta, muchas tarjetas de crédito y cuentas de banco, así como muchos pagos atrasados y muchos días de retraso, tasas de interés altas y muchos pagos mínimos realizados, indicando que el cliente no es capaz de cubrir sus deudas en tiempo y forma.

El grupo Standard solo tiene relacionadas las variables Credit\_Utilization\_Ratio y Total\_EMI\_per\_month. En particular, estas variables tienen poca variabilidad por lo que es difícil identificar algún tipo de correlación entre ellas con el Credit Score. Sin embargo, el hecho de que las observaciones de este grupo se ubiquen en una posición intermedia entre las variables más correlacionadas con los otros 2 puntajes, precisamente muestra que aquellos que pertenecen a este grupo no tienen una antigüedad muy alta o estabilidad financiera elevada, pero tampoco muestran un comportamiento deudor elevado respecto a los créditos y préstamos que solicitan así como la manera en que pagan sus deudas.

Adicionalmente al PCA solo con estandarización de las variables, se realizó un PCA añadiendo rotación varimax y usando la matriz de correlación.



En este caso la varianza en los 2 primeros componentes es alrededor del 37%, necesitamos al menos 3 componentes para alcanzar la varianza explicada similar a la que se tenía en 2 componentes con el PCA simple. Sin embargo, al graficar el biplot se observa que hay realmente un comportamiento similar entre ambos PCA's (sin y con rotación).



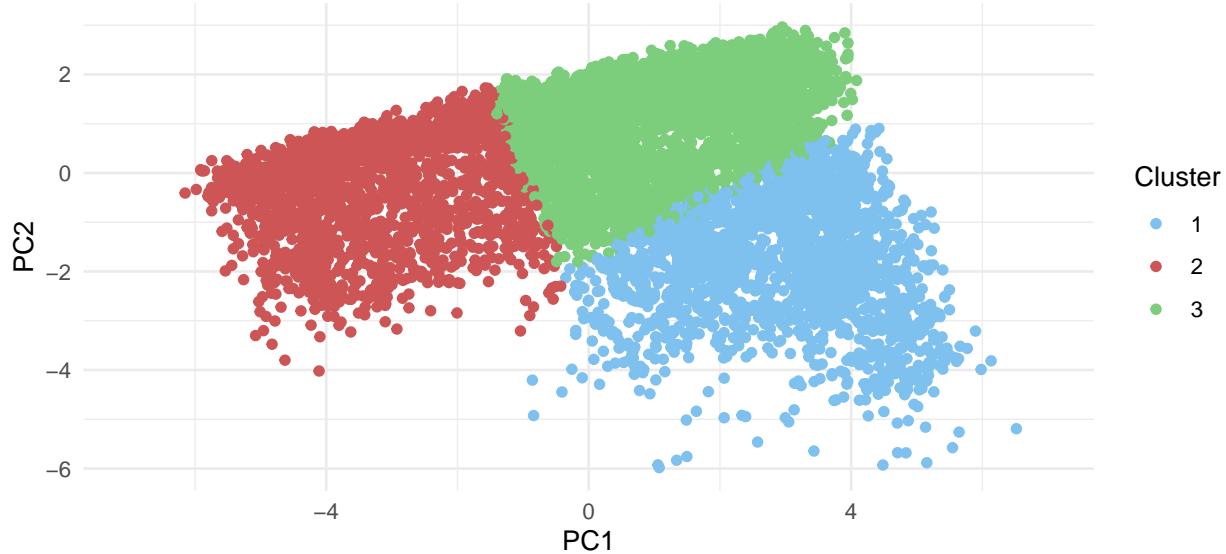
En este caso nótese que en efecto hubo una rotación de 180 grados y ahora la relación entre el PC1 y Credit Score es inversa (valores negativos de PC1 para el puntaje Good y valores positivos para el puntaje Poor).

En general, el análisis de componentes principales logra encontrar un distinción considerable entre los grupos Good y Poor, pero al considerar al grupo Standard la distinción entre las 3 clases se vuelve un poco más compleja visualmente hablando, sin embargo, la relación encontrada entre las variables de la base de datos y los grupos es de especial utilidad para saber como identificar a los 3 grupos, pero sobretodo a los grupos Poor y Good. El principal problema que podría surgir en la clasificación es respecto a clasificar a los del grupo Poor como Good, pues nótese que gráficamente muchos del grupo Poor en ambos PCA's se encuentran en la región del grupo Good, pero pocas observaciones del grupo Good están en el grupo Poor.

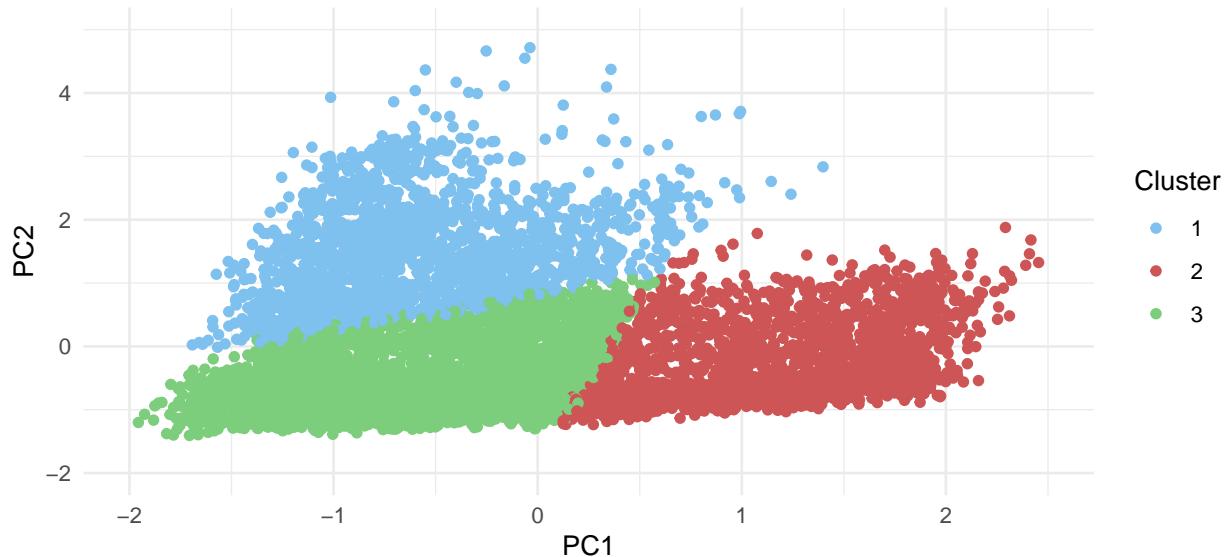
## 4.2 K-Means

Se realizó una agrupación con K-Means con  $K = 3$ . Se estandarizaron los datos y se utilizaron hasta 100 conjuntos aleatorios de centros iniciales y 1000 iteraciones permitidas por cada grupo de centros iniciales. Nota: Se probó distintos valores de nstart e iter.max, pero se eligieron 100 y 1000 respectivamente ya que con otros valores se llegaba a resultados similares o peores a los presentados.

PCA – K-means / k = 3



PCA con rotación – K-means / k = 3



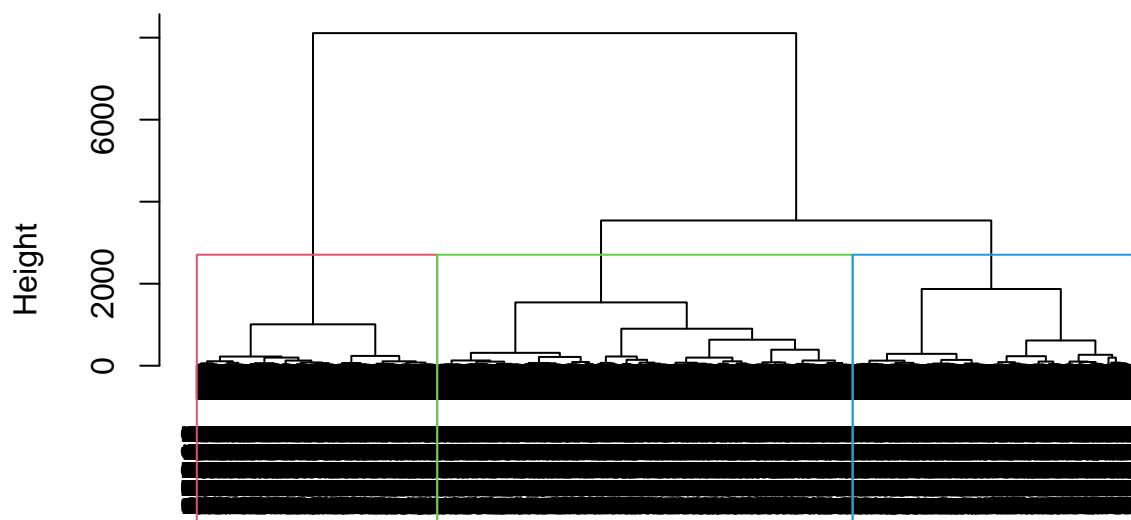
En ambos gráficos se observa que el cluster 2 se ubica en una región bastante similar a la que se ubicaban los del grupo Poor, tanto en la zona negativa del componente principal 1 en el caso del PCA simple con estandarización, como en la zona positiva del componente principal 1 en el caso del PCA usando matriz de correlación y con rotación varimax. Sin embargo, los clusters 1 y 3 se dividen aproximadamente de acuerdo al componente principal 2 y no al componente principal 1 como se había observado en PCA. Entonces, K-Means logra hacer una distinción entre Poor con respecto a Good y Standard, lo cual en sentido estricto es

bueno ya que este grupo es principalmente al que se busca clasificar, pero falla si queremos también hacer una distinción ideal con los del grupo Good, el cual también es un grupo importante.

### 4.3 Método Jerárquico

Para este método, se usó la distancia euclideana y el método Ward. Dado que una matriz de distancias 100 mil observaciones es muy costosa computacionalmente, se utilizó una muestra significativa de 10 mil observaciones para evaluar el método jerárquico, respetando las proporciones de Credit Score (53% Standard, 29% Poor y 18% Good). Nota: Al igual que K-means, se probaron distintas métricas como manhattan, maximum, canberra, minkowski y binary, así como varios métodos como complete, single y average. Lo que más influyó fue el método, pues single y average arrojaban clusters muy desproporcionados, mientras que complete y ward arrojaron resultados similares para casi todas las métricas, de la cual la euclideana fue la que mostró resultados más distinguibles.

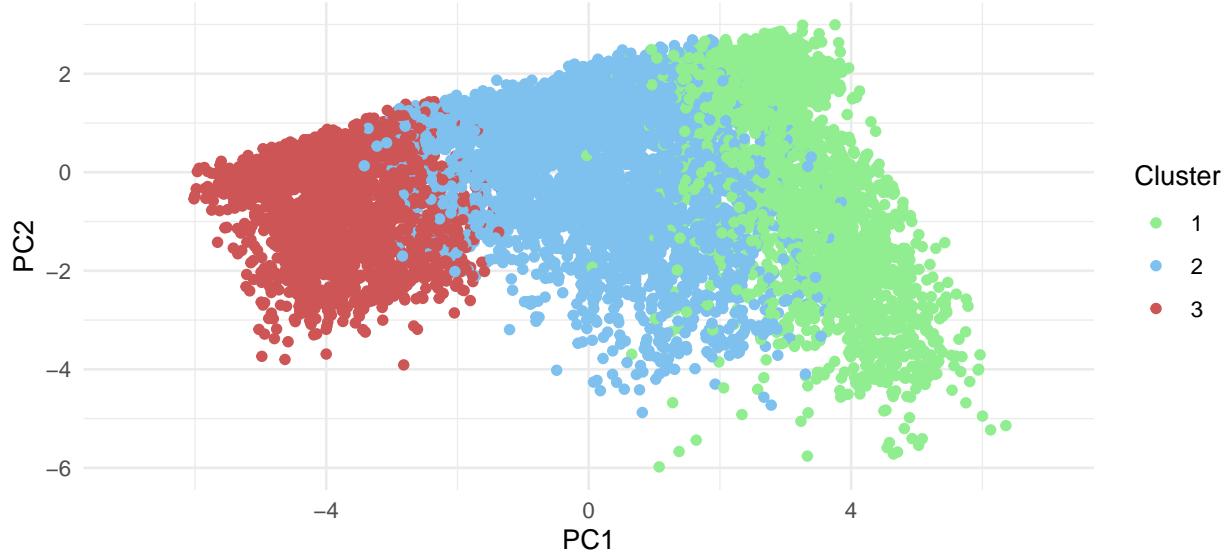
**Cluster Dendrogram**



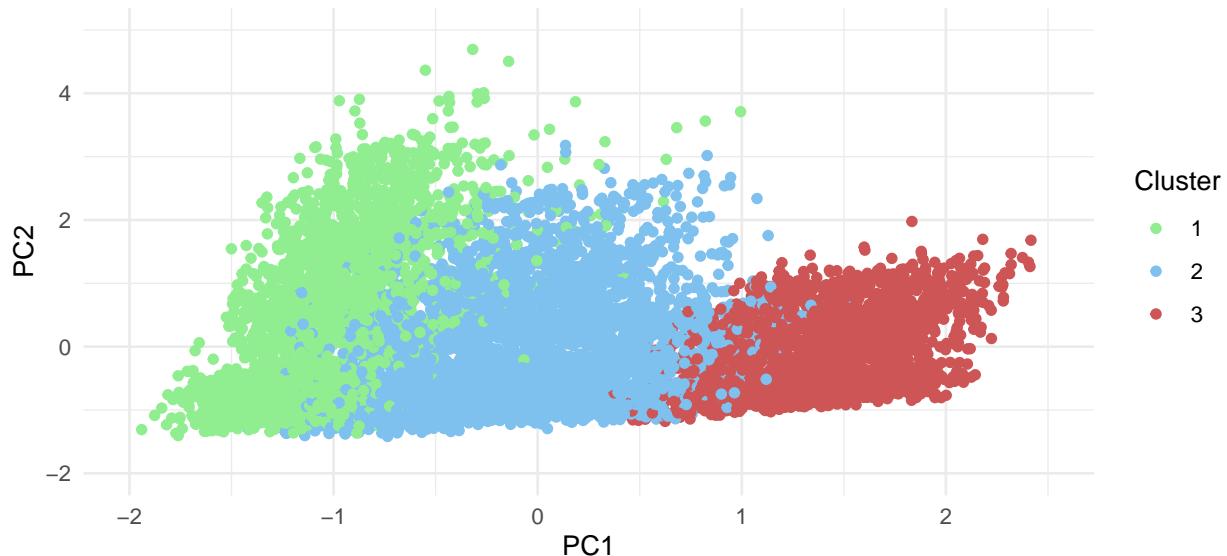
```
dist(datos_sample, method = "euclidean")
hclust (*, "ward.D")
```

El dendograma alrededor de la altura 3000 muestra 3 clusters bastante distinguibles, de los cuales por la longitud en el eje x o de las observaciones de cada cluster, podemos deducir por las proporciones antes mencionadas que se tratan de los grupos Good (izquierda/18%), Standard (centro/53%) y Poor (derecha/29%). A diferencia de K-Means, el cuál no distinguía entre el grupo Good y Standard, el método jerárquico a altura 4000 aproximadamente encuentra que los grupos Standard y Poor tienen más similitud que el grupo Good.

### PCA – Método jerárquico



### PCA con rotación – Método jerárquico



Observando los clusters mediante componentes principales, a diferencia de K-Means, se obtienen clusters muy similares a los visualizados en PCA, pues nótese que hay una separación a lo largo del componente principal 1 para los 3 clusters obtenidos (valores negativos, cercanos a 0, y positivos).

Por lo tanto, con este método hay evidencia de que si es posible encontrar separaciones algo generales entre los 3 grupos, y sobretodo, junto con lo que arrojó K-Means, es posible encontrar un diferenciador sobre los grupos Good y Poor. Sin embargo, es importante aclarar que a pesar de que ambos métodos lograron identificar al grupo Poor de manera similar, en PCA se mostró que algunas observaciones de este grupo podrían terminar siendo difíciles de clasificar correctamente debido a que comparten valores similares con la mayoría de las observaciones del grupo Good, además de que el grupo Standard se fuga entre ambos grupos de igual manera, por lo que los métodos para predecir el Credit Score podrían no tener un Accuracy muy alto debido a este porcentaje significativo de observaciones que podrían considerarse como "outliers" respecto al grupo que corresponden.

## 5 Predicción de Credit Score

En esta sección se muestran diferentes modelos utilizados para intentar predecir el Credit Score. Para evaluar el poder predictivo general de cada uno de los modelos, se usó un K Cross Validation con  $K = 5$ , de manera que se tienen 5 pliegues de 20 mil observaciones cada uno, donde en cada una de las 5 evaluaciones del poder predictivo se consideran 4 pliegues para entrenar los modelos (80 mil observaciones de train) y 1 pliegue para comparar las predicciones hechas por cada modelo respecto a las reales (20 mil observaciones de test).

Las métricas que se consideran son la Tasa de Clasificación Global (Accuracy), Sensibilidad (Recall), Especificidad (Specificity), F1-Score (F1) y el Área bajo la Curva ROC (ROC-AUC). Dado que el problema de clasificación involucra 3 clases, se calcularon estas métricas por clase considerando un problema binario para cada clase (Poor vs. Standard & Good, Standard vs. Poor & Good y Good vs. Standard & Poor). Para obtener el rendimiento general del modelo por métrica, se tomó el promedio de lo obtenido por métrica en las 3 clases. En cada sección, se muestran 2 tablas, la primera tabla con las métricas generales del modelo (promedio de cada métrica en las 3 clases), y la segunda tabla con el desglose de las métricas por clase considerando los problemas binarios mencionados.

En la mayoría de los modelos presentados es necesario hacer un tuneo de hiperparámetros para encontrar el hiperparámetro o la combinación de hiperparámetros más óptima en cada una de las iteraciones del Cross Validation. Por lo tanto, para cada iteración, usando solo datos del train, se vuelve a realizar un K Cross Validation con  $K = 3$  para evaluar el poder predictivo por cada combinación de hiperparámetros. La métrica usada para tomar la decisión de que hiperparámetros son los mejores es el Accuracy.

En particular, con todo lo analizado en las secciones pasadas, además de querer clasificar correctamente a cada una de las clases, queremos que aquellos que sean del grupo Poor si sean clasificados en ese grupo, además de que si el modelo predice a alguien en el grupo Good, la probabilidad de que realmente sea del grupo Good sea alta, ya que como banco queremos identificar quienes son los que cuentan con un mayor riesgo de no pagar sus créditos y préstamos, así como evitar dar préstamos a clientes que muy probablemente no pagarán sólo porque el modelo los clasificó en una clase buena. Por lo tanto, las métricas más importantes en el análisis del poder predictivo serán el Accuracy, Recall del grupo Poor y Precision del grupo Good.

### 5.1 Regresión Logística Multinomial

Para este modelo, se consideró obtener los coeficientes usando penalización Lasso, teniendo como el modelo más grande el que incluía a todas las variables más todas las variables numéricas al cuadrado, por lo que se debe tunear el parámetro  $\lambda$  de la penalización. Para cada iteración del Cross Validation, se consideró una malla de 50 valores distintos de  $\lambda$  y se usó la función `glmnet` para el tuneo.

accuracy	recall	specificity	precision	f1	roc_auc
0.66283	0.6462902	0.8087397	0.6426902	0.6383815	0.8164185

	Recall	Specificity	Precision	F1	ROC_AUC
Class: Poor	0.5275653	0.8994698	0.6819107	0.5948022	0.8173367
Class: Standard	0.7308302	0.6516357	0.7043830	0.7173180	0.7597423
Class: Good	0.6804750	0.8751135	0.5417769	0.6030243	0.8721764

El accuracy obtenido es del 66%, el cual es un valor que definitivamente puede mejorar, ya que solo 2 de 3 clientes están siendo clasificados en la clase correcta. De Recall general se obtiene un valor similar al accuracy de 64% pero notemos que precisamente la clase Poor que es la que más nos interesa clasificar correctamente no está teniendo un buen rendimiento, pues solo el 52.75% se predicen realmente como clase Poor, y lo mismo ocurre con los que son clasificados en la clase Good, en la Precision vemos que solo el 54.17% son realmente de esa clase, lo que nos indica entonces que muchos de la clase Poor probablemente están siendo clasificados

como Good. En la clase Standard el rendimiento es mejor tanto en Recall como en Precision, con valores de 73% y 70% respectivamente, y por lo mismo Specificity para las clases Poor y Good es bastante alto, alrededor del 87%-89% debido a la alta clasificación de la clase Standard, pero justo en esta clase su Specificity es solo del 65%, similar al accuracy general, ya que las clases Poor y Good no son correctamente clasificadas.

En general, este modelo tiene un accuracy decente pero que se puede mejorar, pero precisamente las métricas de interés más particulares como lo son el Recall de la clase Poor y la Precision de la clase Good son muy bajas, mientras que la clasificación de la clase Standard es la mejor de las 3, por lo que podemos concluir que la regresión logística multinomial no está capturando adecuadamente las relaciones entre las clases y el resto de variables, que estas relaciones no son necesariamente lineales sino más complejas y por lo mismo generaliza mucho las características de las 3 clases, donde solo clasifica correctamente a aquellos que presenten valores en las variables muy distintivas a la clase que realmente pertenecen como se observó en la sección anterior.

## 5.2 LDA

En este modelo no hubo ningún hiperparámetro para tunear, por lo que su entrenamiento y evaluación fue directo.

accuracy	recall	specificity	precision	f1	roc_auc
0.65949	0.6523717	0.8121009	0.636214	0.6381645	0.8099354

	Recall	Specificity	Precision	F1	ROC_AUC
Class: Poor	0.5575119	0.8876211	0.6696081	0.6084073	0.8086013
Class: Standard	0.7026864	0.6883190	0.7191480	0.7107856	0.7545618
Class: Good	0.6969168	0.8603625	0.5198858	0.5953006	0.8666430

Tenemos resultados muy similares a la regresión logística multinomial, ya que el accuracy es cercano al 66%, el Recall de la clase Poor aumentó a 55% pero la Precision de la clase Good disminuyó a 52%. Volvemos a observar que la Specificity de las clases Good y Poor es alta debido a que la clase Standard tiene un Recall y Precision alrededor del 70% (F1 Score de 71%) con respecto al Recall y Precision de los otros 2 grupos (F1 Score de 60% aproximadamente). Por tanto, este modelo ni mejoró ni empeoró las métricas del modelo anterior, solamente la ventaja es que a diferencia de la regresión logística multinomial, LDA tiene un entrenamiento mucho más rápido, por lo que obtenemos resultados similares con un menor costo computacional, pero como ya se explicó, estas métricas deben ser mejoradas.

## 5.3 QDA

Al igual que LDA, este modelo no tiene hiperparámetros a tunear, por lo que también su entrenamiento y evaluación fue directo, aunque al ser un modelo más complejo que LDA tomó más tiempo de entrenamiento y evaluación pero no tanto como la regresión logística multinomial.

accuracy	recall	specificity	precision	f1	roc_auc
0.65264	0.702215	0.8319523	0.6451589	0.6488895	0.8170035

	Recall	Specificity	Precision	F1	ROC_AUC
Class: Poor	0.7574762	0.7865931	0.5918435	0.6644667	0.8192750
Class: Standard	0.5445855	0.8781304	0.8353792	0.6593056	0.7636422
Class: Good	0.8045832	0.8311334	0.5082540	0.6228963	0.8680932

El accuracy obtenido fue de 65% aproximadamente, disminuyó ligeramente alrededor del 1% respecto a la regresión logística multinomial y LDA, sin embargo, el Recall de la clase Poor si aumentó considerablemente de 52%-55% a 75%, pero la Precision de la clase Good se mantuvo igual alrededor del 50%, y además, notemos que en este caso la clase Standard disminuyó mucho en su Recall, pasando de 70%-73% a solo 54%. Además, el Recall de la clase Good en los otros 2 modelos tenía un valor de 70% pero en este caso aumentó en 80%. Por tanto, QDA está logrando distinguir a las clases Good y Poor de una mejor manera que la regresión logística multinomial y LDA, pero esta separación muy seguramente está incluyendo a todas esas observaciones del grupo Standard que visualmente cuando se observó el biplot del PCA estaban mezcladas en ambos grupos Good y Poor. Por tanto, si tenemos una métrica mejorada que es el Recall de la clase Poor, pero el Accuracy y la Precision del grupo Good siguen sin mejorar, por lo que es necesario probar otros modelos.

## 5.4 Naive Classifier

Para este modelo se tuneó el hiperparámetro de laplace  $k$ , el cual corrige la probabilidad para evitar tener probabilidades de 0 debido a alguna variable. Se consideraron los valores de  $k = 1, 5, 10, 20, 50$ , y el tuneo se realizó con la paquetería caret. Además, solo se trabajó con las 19 variables numéricas debido a que al tener muchas variables dummies (29 en total), esto podría ocasionar que incluso teniendo probabilidades condicionales altas por variable, al tener tantas variables la probabilidad fuese muy cercana a cero (ejemplo  $0.9^{48} = 0.006$ ), lo cual no es ideal considerando que las 29 dummies solo representan la información de otras 3 variables categóricas más.

accuracy	recall	specificity	precision	f1	roc_auc
0.61667	0.6847948	0.8204905	0.6256591	0.6150023	0.7722744

	Recall	Specificity	Precision	F1	ROC_AUC
Class: Poor	0.7356123	0.7999893	0.6003542	0.6611137	0.7612198
Class: Standard	0.4760943	0.8956955	0.8382575	0.6072555	0.7077193
Class: Good	0.8426777	0.7657866	0.4383656	0.5766377	0.8478841

Este por el momento es el peor modelo que se tiene en términos del accuracy, pues solo el 61.66% de las observaciones se clasificaron correctamente. Además, obtenemos resultados similares a los obtenidos con QDA pero con métricas incluso peores, el Recall de la clase Poor bajo ligeramente a 73%, pero la Precision del grupo Good disminuyó a 43%, y la clase Standard nuevamente tiene un Recall muy bajo del 47%, y el Recall de la clase Good aumentó a 84%, indicando entonces que este modelo hace una separación todavía más drástica entre las clases Good y Poor que QDA, sin lograr distinguir a aquellos de la clase Standard que se encuentran mezclados en los otros 2 grupos.

## 5.5 K Nearest Neighbors

En este modelo el hiperparámetro tuneado fue  $k$ , de cuál se consideró una malla con los valores  $k = 3, 5, 7$ . Al igual que con Naive, se usó caret para el tuneo y se consideraron solamente las variables numéricas, en este caso debido a que al tener más dummies que variables numéricas, estas podrían predominar considerablemente en el cálculo de las distancias entre observaciones.

accuracy	recall	specificity	precision	f1	roc_auc
0.75787	0.7436973	0.8629725	0.7407067	0.7420903	0.8759456

	Recall	Specificity	Precision	F1	ROC_AUC
Class: Poor	0.7714887	0.8941880	0.7486003	0.7598459	0.9022600
Class: Standard	0.7750689	0.7627360	0.7876556	0.7812969	0.8263884
Class: Good	0.6845343	0.9319936	0.6858643	0.6851282	0.8991884

A diferencia del resto de modelos, KNN logra aumentar considerablemente todas las métricas de interés, principalmente el Accuracy, donde tenemos un 75.78% de clasificación correcta a diferencia del 61%-66% que los otros modelos estaban obteniendo, el Recall de la clase Poor aumenta ligeramente a 77% pero la Precision del grupo Good si aumenta considerablemente a 68.5%. En este caso, las 3 clases están obteniendo valores más equilibrados tanto de Recall como de Precision (68% a 77%), indicandonos que si hay una mejora tanto en la predicción que están obteniendo las observaciones de una clase verdadera, como en a que clase realmente pertenecen las predicciones que hace el modelo de una clase en específico. En este caso, el grupo Good es quien tiene el Recall y Precision más bajo (F1 Score del 68.5%), ya que como se observó en el biplot al hacer PCA, en la región de las observaciones del grupo Good también se mezclaban muchas observaciones de los grupos Standard y Poor, lo cual no pasaba es con el grupo Poor, el cual solamente tenía en su mayoría observaciones Standard mezcladas en su región. Pero en general, KNN logra aumentar y equilibrar las métricas de interés por clase.

## 5.6 Árbol de Decisión

Para este modelo, tenemos 3 hiperparámetros a tunear, los cuales son la complejidad del arbol (cost\_complexity), la profundidad del árbol (tree\_depth), y el número mínimo de observaciones que deben tener los nodos finales (min\_n). Para cada uno, se consideraron las mallas por defecto de cada hiperparámetro dadas por tidymodels:

- *cost\_complexity*:  $[10^{-10}, 0.01]$
- *tree\_depth*:  $[1, 30]$
- *min\_n*:  $[2, 40]$

Sin embargo, no se realizó un tuneo completo considerando todas las posibles combinaciones de hiperparámetros en esos intervalos, sino que se recurrió a una búsqueda aleatoria, tomando aleatoriamente solo 20 combinaciones de hiperparámetros entre esos rangos.

accuracy	recall	specificity	precision	f1	roc_auc
0.73442	0.7256426	0.8535865	0.7099999	0.7165556	0.8582151

	Recall	Specificity	Precision	F1	ROC_AUC
Class: Poor	0.7331004	0.8955259	0.7415457	0.7372112	0.8726923
Class: Standard	0.7482800	0.7632001	0.7820998	0.7647549	0.8082481
Class: Good	0.6955472	0.9020336	0.6063542	0.6477007	0.8937047

A diferencia de KNN, tenemos que el rendimiento si disminuyó ligeramente, pues se obtiene un Accuracy del 73.44% respecto al 75.78% de KNN, el Recall tanto del grupo Poor como del grupo Standard baja de 77% en ambos en KNN a 73% y 74% respectivamente, pero sobretodo la Precision del grupo Good baja considerablemente de 68% en KNN a 60%. Esto nos indica que el Árbol de decisión tiene un poco más de dificultad en clasificar a las observaciones de las clases Poor y Standard que son similares a las del grupo Good, por lo mismo los Recall de Poor y Standard bajan así como la Precision de Good. Sin embargo, si consideramos más árboles y más variabilidad en los mismos así como el número de variables y la muestra que cada árbol considera, podríamos mejorar estas métricas, para detectar patrones más profundos.

## 5.7 Bosque Aleatorio

Para los bosques aleatorios consideramos igualmente tunear el número mínimo de observaciones que deben tener los nodos finales (`min_n`), sin embargo, en lugar de modificar la complejidad de cada árbol y su profundidad máxima, estos hiperparámetros los sustituimos por la cantidad de árboles que tendrá el bosque (`trees`) y el número de variables que cada árbol considerará (`mtry`), de modo que las exactitud de las predicciones ya no dependa tanto de que tan complejos y profundos son los árboles, sino de que tanta variabilidad exista entre ellos. En este caso, se fijaron mallas por hiperparámetros manualmente con los siguientes valores:

- `mtry`: [5, 20]
- `trees`: [100, 500]
- `min_n`: [10, 100]

Al igual que en el árbol de decisión, sólo se tomó aleatoriamente 20 combinaciones de estos hiperparámetros para reducir el costo computacional. Además, dado que se considerarán solo de 5 a 20 variables por árbol, se decidió entrenar solo usando las 19 variables numéricas, para evitar tener muchos árboles con solo variables dummies, que al ser 29, podrían estar muy presentes en todos los árboles y por tanto, reducir la variabilidad de los mismos (por ejemplo, un árbol con 15 variables pero de las cuales 8 sean dummies de una categórica, 5 de otra, y solo 2 numéricas, teniendo en realidad un árbol de 4 variables en lugar de 15).

accuracy	recall	specificity	precision	f1	roc_auc
0.81098	0.8072997	0.8936373	0.7999636	0.8032213	0.9229914

	Recall	Specificity	Precision	F1	ROC_AUC
Class: Poor	0.8414880	0.9078895	0.7886248	0.8141759	0.9398831
Class: Standard	0.8069632	0.8220414	0.8373665	0.8218741	0.8750079
Class: Good	0.7734478	0.9509809	0.7738994	0.7736139	0.9540831

Este hasta ahora ha sido el mejor resultado, pues el accuracy aumenta considerablemente a 81%, el Recall de las 3 clases se ubica entre 77% y 84%, con un promedio de 80.7%, donde precisamente la clase Poor que es de nuestro interés alcanza hasta un 84% de Recall, además de que la Precision de la clase Good aumentó mucho hasta 77.4%, y de manera general, la Precision alcanza un 80%. En general, tanto en Recall y Precision, las clases ya se encuentran mucho más equilibradas, la clase Good sigue siendo la de menor valor en ambas (F1 Score de 77.36% respecto al F1 Score de 81.4% de la clase Poor y al F1 Score de 82% de la clase Standard), pero la diferencia ya no es tan significativa como en todos los anteriores modelos, donde había hasta más de un 10% de diferencia en Recall, Precision y F1 Score.

## 5.8 Extreme Gradient Boosting

Se consideraron las mismas mallas que se tenían en el bosque aleatorio para los hiperparámetros `mtry`, `trees` y `min_n`, además de tunear la tasa de aprendizaje `learn_rate` para cada árbol, teniendo entonces los siguientes valores para los hiperparámetros.

- `mtry`: [5, 20]
- `trees`: [100, 500]
- `min_n`: [10, 100]
- `learn_rate`: [0.1, 0.25]

Cabe mencionar que los hiperparámetros `loss_reduction`, `sample_size` y `tree_depth` se mantienen con los valores por defecto dados por `tidymodels` (0.1 y 6 respectivamente), ya que al considerar mallas para estos parámetros el modelo tendió a sobreajustar, y por tanto, se obtenían accuracies alrededor de 70%, lo cual no era lo mejor con respecto a los últimos modelos.

accuracy	recall	specificity	precision	f1	roc_auc
0.77919	0.7641481	0.8721425	0.7700041	0.7669785	0.9069922

	Recall	Specificity	Precision	F1	ROC_AUC
Class: Poor	0.7754820	0.9090298	0.7768646	0.7761646	0.9260418
Class: Standard	0.8030968	0.7617348	0.7928602	0.7979414	0.8528854
Class: Good	0.7138657	0.9456631	0.7402875	0.7268294	0.9420493

Se obtuvo un accuracy cercano al 78%, lo cuál, a pesar de ser Extreme Gradient Boosting un modelo que va mejorando los árboles que se entranan, no superó al 81% obtenido en bosque aleatorio, aunque por el momento es el segundo mejor modelo obtenido con respecto al Accuracy. Respecto a las métricas específicas por clase, así como el Accuracy disminuyó alrededor de 3% con respecto al bosque aleatorio, el Recall y Precision en general también disminuyeron 3.5% y 2% respectivamente, y sobretodo en las clases Good y Poor en el caso del Recall tenemos un decremento del 6% en Recall, y 3% para Precision en el caso de las clases Good y Standard, dando como resultado que en las 3 clases el F1-Score se redujera alrededor de 3% a 4% con respecto al bosque aleatorio, similar a lo que pasó con el Accuracy. Si bien este modelo es mejor que el segundo mejor modelo que teníamos antes tanto en Accuracy como en las métricas por clase (K-Nearest-Neighbors), no se mejoraron los resultados obtenidos previamente con el bosque aleatorio, muy probablemente debido a que todas las combinaciones de parámetros produjeron un poco de overfitting a comparación del bosque aleatorio, y por tanto, algunos de la clase Poor se clasificaron en Good y de la clase Good en Standard.

## 5.9 Stacking (KNN, Bosque Aleatorio y Extreme Gradient Boosting)

Finalmente, se consideró hacer un Stacking usando los 3 mejores modelos obtenidos hasta el momento, los cuáles fueron K-Nearest-Neighbors, Bosque Aleatorio y Extreme Gradient Boosting. Se entrenaron los 3 modelos con las mismas mallas de hiperparámetros usadas de manera individual por cada modelo, se obtuvieron las predicciones del conjunto de entrenamiento, y con estas predicciones se entrenó el modelo final Meta, el cuál fue una regresión logística multinomial vía Red Neuronal.

accuracy	recall	specificity	precision	f1	roc_auc
0.79139	0.7806523	0.8802959	0.7818596	0.7810361	0.9046498

	Recall	Specificity	Precision	F1	ROC_AUC
Class: Poor	0.7992085	0.9083141	0.7806322	0.7895998	0.9211017
Class: Standard	0.8051678	0.7842804	0.8095894	0.8072481	0.8619075
Class: Good	0.7375806	0.9482932	0.7553571	0.7462605	0.9309402

Notemos que, a pesar de combinar varios modelos, en este caso tampoco logramos superar al Bosque Aleatorio, solo mejoraramos los resultados del Extreme Gradient Boosting aumentando 1% en Accuracy, y alrededor de 1% a 2% en todas las métricas generales por clase, de las cuáles ninguna mejoró con respecto al Bosque Aleatorio. La causa más probable de esto se debe a que tanto KNN como Extreme Gradient Boosting pueden no tener mucha diversidad en sus predicciones (ser similares en cuanto a sus errores) y por tanto, no aportar mucho al Bosque Aleatorio, sobre todo en aquellas observaciones muy difíciles de clasificar correctamente, además de que el modelo Meta usado no es muy complejo.

## 5.10 Conclusiones

A continuación, se muestra un resumen de las métricas de interés principal (Accuracy, Recall clase Poor y Precision clase Good) obtenidas por todos los modelos, ordenados de manera descendente de acuerdo a su rendimiento.

Modelo	Accuracy	Recall.Poor	Precision.Good
Bosque Aleatorio	0.81098	0.8414880	0.7738994
Stacking	0.79139	0.7992085	0.7553571
Extreme Gradient Boosting	0.77919	0.7754820	0.7402875
K Nearest Neighbors	0.75787	0.7714887	0.6858643
Árbol de Decisión	0.73442	0.7331004	0.6063542
QDA	0.65264	0.7574762	0.5082540
Regresión Logística Multinomial	0.66283	0.5275653	0.5417769
LDA	0.65949	0.5575119	0.5198858
Naive Classifier	0.61667	0.7356123	0.4383656

Con respecto al baseline de 53% de Accuracy que se obtendría si se clasificaran a todas las observaciones como “Standard” (clase con mayor proporción), tenemos que todos los modelos superaron este baseline, por lo que si aportan algo de valor. Naive Classifier es el peor modelo tanto en Accuracy como en Precision del grupo Good, indicandonos que las observaciones no son fácilmente clasificables solo con probabilidad condicional. Por otro lado, la regresión logística multinomial, LDA y QDA mejoran el baseline alrededor de 12% a 13%, sin embargo, Recall de Poor y Precision de Good tienen un valor de alrededor de 50%, salvo QDA con una mejora de 75% de Recall del grupo Poor, indicándonos que las relaciones entre Credit Score y el resto de variables no son solamente lineales ni cuadráticas, sino más complejas.

K Nearest Neighbors y los Árboles de Decisión mejoran considerablemente el baseline a poco más de 20%, encontrando patrones más profundos entre Credit Score y las variables, además de aumentar de igual forma 20% para el Recall de la clase Poor y al menos 10% para Precision de la clase Good. Al introducir variabilidad y mayor aprendizaje con modelos como el Bosque Aleatorio, Extreme Gradient Boosting y el Stacking de KNN, Bosque Aleatorio y Extreme Gradient Boosting, se mejora alrededor de 5% a 7% a las 3 métricas, principalmente Precision de Good, de donde el Bosque Aleatorio termina siendo el modelo más óptimo de todos los modelos entrenados al ser el único que obtuvo un Accuracy mayor al 80%, así como al aumentar el Recall del grupo Poor un 5% con respecto al segundo mejor modelo (Stacking), y aumentar un 2% tanto en Accuracy como en Precision de Good respecto a ese segundo mejor modelo.

Es importante destacar que no necesariamente el Bosque Aleatorio es el mejor modelo definitivo, muy probablemente se pueden mejorar ligeramente las métricas con otros modelos como Support Vector Machine o Redes Neuronales, los cuales podrían agregarse al Stacking junto con un modelo Meta más complejo para tener más variabilidad y aprendizaje. De igual manera, los modelos probados podrían optimizarse con un tuneo más profundo o inteligente de hiperparámetros como una búsqueda completa en las mallas dadas o una búsqueda Bayesiana. Debido al gran costo computacional que implica el entrenamiento de todas estas alternativas, no se incluyeron en este proyecto, pero es importante mencionarlas ya que los modelos obtenidos se pueden mejorar.