Maestría en Inteligencia Artificial Aplicada

Actividad 1 de la semana 3

Nombre del estudiante: Dalina Aidee Villa Ocelotl

Matrícula: A01793258

Curso: Ciencia y analítica de datos (Gpo 10)

Prof. Jobish Vallikavungal

1 de octubre de 2022

Parte 1: Fundamentos de bases de datos

- ***Fundamentos de bases de datos en ciencia de datos:***

> La ciencia de datos es parte de la evaluación del tratamiento integral de los datos. Dentro del proceso de conocimiento se describe el proceso como parte de la aplicación a resolución de problemas de la vida real con ***datos***. Actualmente se tienen una gran cantidad de datos que pueden venir desde distintas fuentes y además ser de distintos tipos de datos, estos después se transforman en ***información*** a través de aplicación de cambios a esos datos, normalizarlos e incluso tratar de describirlos a través de estadística descriptiva. Este proceso debe incluir la validación propia de la naturaleza de los datos ya que en la vida real siempre puede haber datos faltantes, o incluso datos que pudieran no tener sentido, desarrollando el proceso ETL. Por lo que, conocer el contexto de aplicación de la investigación es fundamental convirtiéndose en aplicación del método científico de manera multidisciplinaria. Seguido de eso se definen supuestos de acuerdo a la información

por lo que se puede ofrecer una solución con estadística más avanzada como inferencia estadística o modelos predictivos que pueden entrenarse con algoritmos de aprendizaje automático. Dentro de este proceso(evaluación, explicación, despliegue productivo y supervisión) se obtienen ***insights o puntos clave*** y además modelos productivos que ayuden a la toma de decisiones. Finalmente, este proceso de conocimiento nos permite ser más ***sabios*** respecto de los fenómenos que se monitorean inicialmente a través de datos.

- ***Fundamentos de Almacenes de datos (Data Warehouse) para Ciencia de Datos***

Los modelos predictivos requieren de datos consistentes, claros y concisos para aplicar sus conocimientos estadísticos e informáticos que puedan considerarse robustos y precisos. Los Data Warehouse se convierten en los repositorios de información que cumpla con dichos supuestos para utilizarlos, al existir gran cantidad de datos se requiere de establecer una estructura a los datos que la tengan, corregir alguna o incluso establecer una para datos no estructurados que de alguna manera permita el fácil acceso a explotación de los mismos. De acuerdo a la forma del negocio o lugar donde se cree, se puede establecer los lineamientos para estandarizar la información e incluso plantear el correcto gobierno de los datos. Deben garantizar contener todos los datos, la parte del procesamiento, almacenamiento correcto, pero ademas la agilidad para poder explotarlos y garantizar la seguridad de la información.

- ***Referencias:***

*Guía de visualización de datos para principiantes: definición, ejemplos y recursos de aprendizaje. (s/f). Tableau. Recuperado el 1 de octubre de 2022, de https://www.tableau.com/es-mx/learn/articles/data-visualization *¿Qué es la ciencia de datos? (s/f). Oracle.com. Recuperado el 1 de octubre de 2022, de https://www.oracle.com/mx/what-is-data-science/ * *https://newoutlook.it/download/python/hands-on-data-science.pdf *Data Warehouse: todo lo que necesitas saber sobre almacenamiento de datos. (s/f). Powerdata.Es. Recuperado el 1 de octubre de 2022, de https://www.powerdata.es/data-warehouse *texto en cursiva*

Parte 2: Selección y limpieza de datos en Python

Es el proceso mediante el que se identifica y corrige el tipo de datos del dataset e incluso detectar incosistencias para modificar o eliminar esos datos. Por lo que se valida la consistencia con el contexto que se aplique.

---

*Descripcion del conjunto de datos: *

> *This research aimed at the case of customers default payments in Taiwan and compares the predictive accuracy of probability of default among six data mining methods. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. Because the real probability of default is unknown, this study presented the novel "Sorting Smoothing Method" to estimate the real probability of default. With the real probability of default as the response variable (Y), and the predictive probability of default as the independent variable (X), the simple linear regression result (Y = A + BX) shows that the forecasting model produced by artificial neural network has the highest coefficient of determination; its regression intercept (A) is close to zero, and regression coefficient (B) to one. Therefore, among the six data mining techniques, artificial neural network is the only one that can accurately estimate the real probability of default.*

```
# Importar paqueterias para procesamiento de datos
import pandas as pd
import numpy as np
import seaborn as sb
from statistics import mode
```

```
input="https://raw.githubusercontent.com/PosgradoMNA/Actividades_Aprendizaje-/main/default%20
```

```
datos_creditos = pd.read_csv(input)
datos_creditos.head()
```

	ID	X1	X2	X3	X4	X5	X6	X7	X8	X9	...	X15	X16	X17	X
0	1	20000	2.0	2.0	1.0	24.0	2.0	2.0	-1.0	-1.0	...	0.0	0.0	0.0	
1	2	120000	2.0	2.0	2.0	26.0	-1.0	2.0	0.0	0.0	...	3272.0	3455.0	3261.0	
2	3	90000	2.0	2.0	2.0	34.0	0.0	0.0	0.0	0.0	...	14331.0	14948.0	15549.0	151
3	4	50000	2.0	2.0	1.0	37.0	0.0	0.0	0.0	0.0	...	28314.0	28959.0	29547.0	200
4	5	50000	1.0	2.0	1.0	57.0	-1.0	0.0	-1.0	0.0	...	20940.0	19146.0	19131.0	200

5 rows × 25 columns

```
datos_creditos.describe()
```

	ID	X1	X2	X3	X4	X
count	30000.000000	30000.000000	29999.000000	29998.000000	29998.000000	29995.0000
mean	15000.500000	167484.322667	1.603753	1.853057	1.551903	35.4842
std	8660.398374	129747.661567	0.489125	0.790320	0.521968	9.2180
min	1.000000	10000.000000	1.000000	0.000000	0.000000	21.0000
25%	7500.750000	50000.000000	1.000000	1.000000	1.000000	28.0000
50%	15000.500000	140000.000000	2.000000	2.000000	2.000000	34.0000
75%	22500.250000	240000.000000	2.000000	2.000000	2.000000	41.0000
max	30000.000000	1000000.000000	2.000000	6.000000	3.000000	79.0000

8 rows × 25 columns

Para iniciar con la limpieza de datos se renombra el conjunto de datos por si se quiere ver el original

```
datos_creditos_1 = datos_creditos
```

Estandarización de datos

```
datos_creditos_1.dtypes #observamos los tipos de datos que se tienen
```

```
ID       int64
X1       int64
X2     float64
X3     float64
X4     float64
X5     float64
X6     float64
X7     float64
X8     float64
X9     float64
X10    float64
X11    float64
X12    float64
X13    float64
X14    float64
X15    float64
X16    float64
X17    float64
X18    float64
X19    float64
X20    float64
X21    float64
X22    float64
X23    float64
```

```
Y      float64
dtype: object
```

```
datos_creditos_1.isnull().values.any()
```

```
True
```

```
datos_creditos_1.isnull().any() # se observa que columnas del conjunto de datos tienen datos
```

```
ID     False
X1     False
X2      True
X3      True
X4      True
X5      True
X6      True
X7      True
X8      True
X9      True
X10     True
X11     True
X12     True
X13     True
X14     True
X15     True
X16     True
X17     True
X18     True
X19     True
X20     True
X21     True
X22     True
X23     True
Y       True
dtype: bool
```

```
#df.columns = ['Amount_credit','Gender', 'Education', 'Marital status','Age' ]
datos_creditos_1.isna().any()
```

```
ID     False
X1     False
X2      True
X3      True
X4      True
X5      True
X6      True
X7      True
X8      True
X9      True
X10     True
X11     True
X12     True
X13     True
X14     True
```

```
X15     True
X16     True
X17     True
X18     True
X19     True
X20     True
X21     True
X22     True
X23     True
Y       True
dtype: bool
```

Descripcion de las variables: Attribute Information:

This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This study reviewed the literature and used the following 23 variables as explanatory variables:

> X1: **Amount of the given credit** ***(NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit. X2: ***Gender** (1 = male; 2 = female). X3: **Education** (1 = graduate school; 2 = university; 3 = high school; 4 = others). X4: ** MaritalStatus** (1 = married; 2 = single; 3 = others). X5: Age (year). X6 - X11: **History of past payment**. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above. X12-X17: **Amount of bill statement** (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = **amount of bill statement in April**, 2005. X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005.

```
datos_creditos_1 = datos_creditos_1.rename({"X1":"CreditAmount",  "X2":"Gender", "X3":"Educat
```

```
datos_creditos_1.head()
```

	ID	CreditAmount	Gender	Education	MaritalStatus	Age	X6	X7	X8	X9	...	
0	1	20000	2.0	2.0	1.0	24.0	2.0	2.0	-1.0	-1.0	...	
1	2	120000	2.0	2.0	2.0	26.0	-1.0	2.0	0.0	0.0	...	3
2	3	90000	2.0	2.0	2.0	34.0	0.0	0.0	0.0	0.0		14

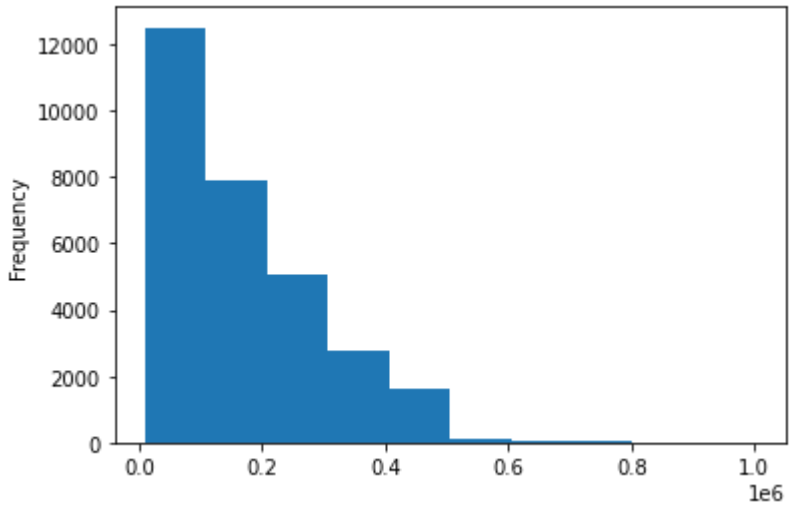
Ahora que conocemos el contexto es importante ver la congruencia de los datos, como son prestamos debemos validar que tengan sentido

```
CreditAmount=datos_creditos_1[datos_creditos_1['CreditAmount']<=0]
print(CreditAmount)
```

```
Empty DataFrame
Columns: [ID, CreditAmount, Gender, Education, MaritalStatus, Age, X6, X7, X8, X9, X10,
Index: []

[0 rows x 25 columns]
```

```
datos_creditos_1['CreditAmount'].plot.hist()  # comprobamos si hay montos negativos en los cr
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373d29ec90>
```




```
datos_creditos_1[datos_creditos_1['CreditAmount']<=0]=0  # si los hubiera aqui los cambiariam
```

```
datos_creditos_1['Gender'].plot.hist()  #verificamos que todos los datos son valores 1 o 2 re
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373d1bc390>
```

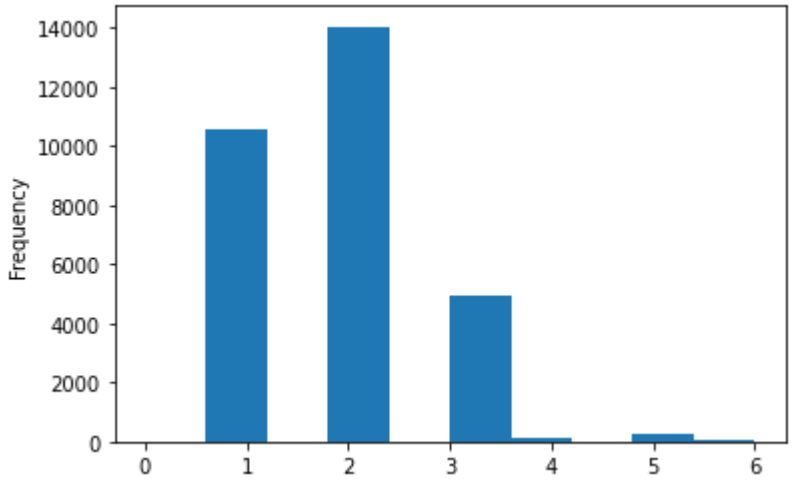

```
Gender=datos_creditos_1[(datos_creditos_1['Gender']!=1) & (datos_creditos_1['Gender']!=2)]
print(Gender)
```

```
          ID  CreditAmount  Gender  Education  MaritalStatus  Age  X6  X7  X8  \
24365  24366        130000     NaN        NaN            NaN  NaN NaN NaN NaN

       X9  ...  X15  X16  X17  X18  X19  X20  X21  X22    X23    Y
24365 NaN  ...  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  390.0  0.0

[1 rows x 25 columns]
```

```
datos_creditos_1[(datos_creditos_1['Gender']!=1) & (datos_creditos_1['Gender']!=2)]=np.random
```

```
datos_creditos_1['Education'].plot.hist()  #verificamos que todos los datos son valores (1 =
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373d04bf90>
```



Se observa que para el nivel de estudios se tienen otras categorias distintas. Por lo que se considera conveniente categorizar como "otros_estudios"

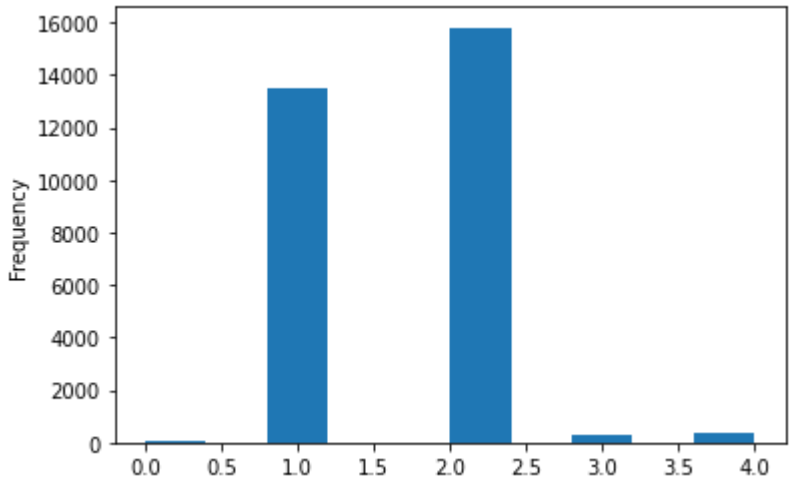
```
Education=datos_creditos_1[(datos_creditos_1['Education']!=1) &
                           (datos_creditos_1['Education']!=2) &
                           (datos_creditos_1['Education']!=3) &
                           (datos_creditos_1['Education']!=4)]
print(Education)
```

```
Empty DataFrame
Columns: [ID, CreditAmount, Gender, Education, MaritalStatus, Age, X6, X7, X8, X9, X10,
Index: []

[0 rows x 25 columns]
```

```
datos_creditos_1[(datos_creditos_1['Education']!=1) &
                 (datos_creditos_1['Education']!=2) &
                 (datos_creditos_1['Education']!=3) &
                 (datos_creditos_1['Education']!=4)]=4 #se crea la nueva categoria
```

```
datos_creditos_1['MaritalStatus'].plot.hist()  # se valida que cumpla con ls supuestos (1 = m
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373cf6b590>
```



Se observa que en la variable de estado civil existen otras categorias

```
MaritalStatus=datos_creditos_1[(datos_creditos_1['MaritalStatus']!=1) &
                               (datos_creditos_1['MaritalStatus']!=2) &
                               (datos_creditos_1['MaritalStatus']!=3)]
print(MaritalStatus)
```

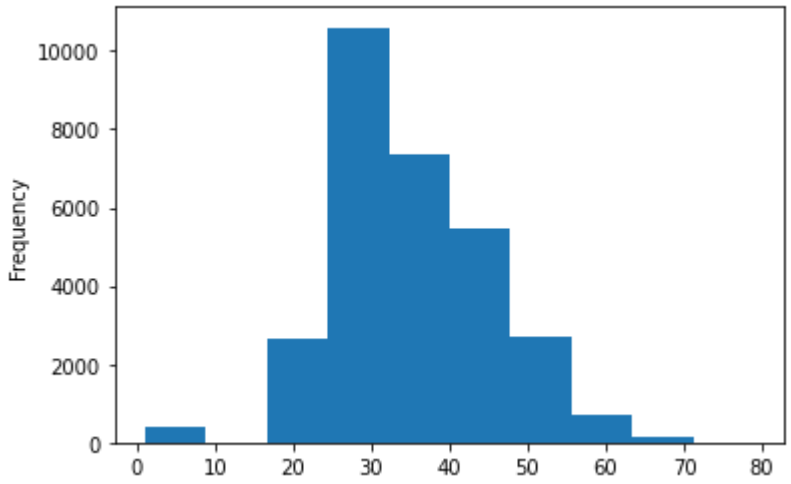
```
          ID  CreditAmount  Gender  Education  MaritalStatus   Age   X6   X7  \
47         4             4     4.0        4.0            4.0   4.0  4.0  4.0
69         4             4     4.0        4.0            4.0   4.0  4.0  4.0
218      219        110000     2.0        3.0            0.0  31.0  0.0  0.0
385        4             4     4.0        4.0            4.0   4.0  4.0  4.0
502        4             4     4.0        4.0            4.0   4.0  4.0  4.0
...      ...           ...     ...        ...            ...   ...  ...  ...
29811      4             4     4.0        4.0            4.0   4.0  4.0  4.0
29836      4             4     4.0        4.0            4.0   4.0  4.0  4.0
29839      4             4     4.0        4.0            4.0   4.0  4.0  4.0
29920      4             4     4.0        4.0            4.0   4.0  4.0  4.0
29966      4             4     4.0        4.0            4.0   4.0  4.0  4.0
```

```
          X8   X9  ...      X15      X16      X17     X18     X19     X20  \
47       4.0  4.0  ...      4.0      4.0      4.0     4.0     4.0     4.0
69       4.0  4.0  ...      4.0      4.0      4.0     4.0     4.0     4.0
218      0.0  0.0  ...  73315.0  63818.0  63208.0  4000.0  5000.0  3000.0
385      4.0  4.0  ...      4.0      4.0      4.0     4.0     4.0     4.0
502      4.0  4.0  ...      4.0      4.0      4.0     4.0     4.0     4.0
...      ...  ...  ...      ...      ...      ...     ...     ...     ...
29811    4.0  4.0  ...      4.0      4.0      4.0     4.0     4.0     4.0
29836    4.0  4.0  ...      4.0      4.0      4.0     4.0     4.0     4.0
29839    4.0  4.0  ...      4.0      4.0      4.0     4.0     4.0     4.0
29920    4.0  4.0  ...      4.0      4.0      4.0     4.0     4.0     4.0
29966    4.0  4.0  ...      4.0      4.0      4.0     4.0     4.0     4.0

           X21     X22     X23    Y
47         4.0     4.0     4.0  4.0
69         4.0     4.0     4.0  4.0
218     3000.0  3000.0  8954.0  0.0
385        4.0     4.0     4.0  4.0
502        4.0     4.0     4.0  4.0
...        ...     ...     ...  ...
29811      4.0     4.0     4.0  4.0
29836      4.0     4.0     4.0  4.0
29839      4.0     4.0     4.0  4.0
29920      4.0     4.0     4.0  4.0
29966      4.0     4.0     4.0  4.0

[400 rows x 25 columns]
```

```
datos_creditos_1[(datos_creditos_1['MaritalStatus']!=1) &
                 (datos_creditos_1['MaritalStatus']!=2) &
                 (datos_creditos_1['MaritalStatus']!=3)]=3 # se trata de unificar la variable
```

```
datos_creditos_1['Age'].plot.hist() # se valida la distribucion de valores de la variable Eda
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373ce28b10>
```



Se observa que hay otros valores distintos a los esperados para mayores de edad de prestamos

```
Age=datos_creditos_1[datos_creditos_1['Age']<18]
print(Age)
datos_creditos_1[datos_creditos_1['Age']<18]=18 # Ajustamos el valor de la edad al minimo ace
```

```
Empty DataFrame
Columns: [ID, CreditAmount, Gender, Education, MaritalStatus, Age, X6, X7, X8, X9, X10,
Index: []

[0 rows x 25 columns]
```

Para analizar las siguientes variables se observa su comportamiento

```
#Para conocer el estatus de comportamiento de pago de los ultimos 6 meses para ver si es pago
r6=datos_creditos_1.columns.get_loc('X6') #Se busca el índice para la columna llamada 'X6'
r11=datos_creditos_1.columns.get_loc('X11') #Se busca el índice para la columna llamada 'X11'
for i in range(r6,r11+1):
  x6_x11=datos_creditos_1[datos_creditos_1.iloc[:,i]<-1]
  print(x6_x11)
  x6_x11.plot.box()
```

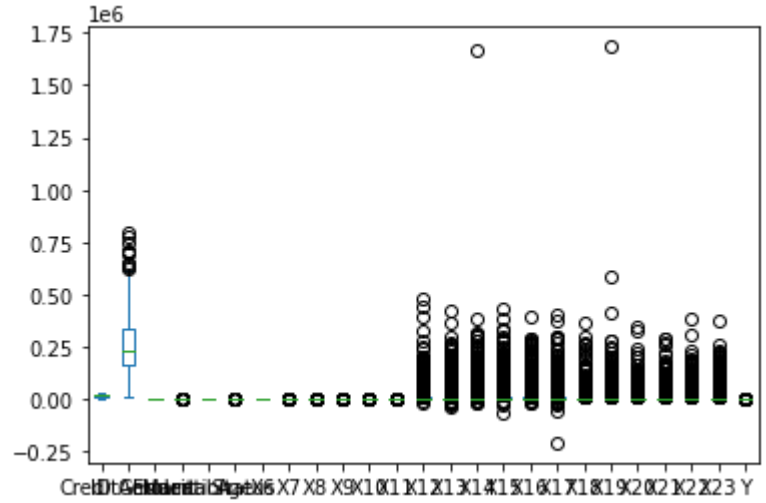
```
          ID  CreditAmount  Gender  Education  MaritalStatus   Age   X6   X7  \
9         10         20000     1.0        3.0            2.0  35.0 -2.0 -2.0
23        24        450000     2.0        1.0            1.0  40.0 -2.0 -2.0
33        34        500000     2.0        2.0            1.0  54.0 -2.0 -2.0
34        35        500000     1.0        1.0            1.0  58.0 -2.0 -2.0
45        46        210000     1.0        1.0            2.0  29.0 -2.0 -2.0
...      ...           ...     ...        ...            ...   ...  ...  ...
29946  29947        230000     1.0        1.0            2.0  44.0 -2.0 -1.0
29961  29962        260000     1.0        1.0            2.0  33.0 -2.0 -2.0
29979  29980        180000     1.0        1.0            1.0  32.0 -2.0 -2.0
29983  29984         20000     1.0        2.0            1.0  44.0 -2.0 -2.0
29985  29986        240000     1.0        1.0            2.0  30.0 -2.0 -2.0

        X8   X9  ...     X15      X16      X17      X18      X19     X20  \
9     -2.0 -2.0  ...     0.0  13007.0  13912.0      0.0      0.0     0.0
23    -2.0 -2.0  ...   560.0      0.0      0.0  19428.0   1473.0   560.0
33    -2.0 -2.0  ...  7521.0  71439.0   8981.0   4152.0  22827.0  7521.0
34    -2.0 -2.0  ...  3180.0      0.0   5293.0   5006.0  31178.0  3180.0
45    -2.0 -2.0  ...     0.0      0.0      0.0      0.0      0.0     0.0
...    ...  ...  ...     ...      ...      ...      ...      ...     ...
29946 -1.0 -1.0  ...  1467.0   9192.0   4388.0   3306.0    806.0  1500.0
29961 -2.0 -2.0  ...  1368.0    101.0    955.0    263.0      0.0  1368.0
29979 -2.0 -2.0  ...     0.0      0.0      0.0      0.0      0.0     0.0
29983 -2.0 -2.0  ...  2882.0   9235.0   1719.0   2890.0   2720.0  2890.0
29985 -2.0 -2.0  ...     0.0      0.0      0.0      0.0      0.0     0.0

           X21     X22      X23    Y
9      13007.0  1122.0      0.0  0.0
23         0.0     0.0   1128.0  1.0
33     71439.0   981.0  51582.0  0.0
34         0.0  5293.0    768.0  0.0
45         0.0     0.0      0.0  1.0
...        ...     ...      ...  ...
29946   9216.0  4388.0      0.0  0.0
29961    101.0   955.0      0.0  0.0
29979      0.0     0.0      0.0  0.0
29983   9263.0  1824.0   1701.0  0.0
29985      0.0     0.0      0.0  0.0

[2708 rows x 25 columns]
          ID  CreditAmount  Gender  Education  MaritalStatus   Age   X6   X7  \
9         10         20000     1.0        3.0            2.0  35.0 -2.0 -2.0
18        19        360000     2.0        1.0            1.0  49.0  1.0 -2.0
19        20        180000     2.0        1.0            2.0  29.0  1.0 -2.0
23        24        450000     2.0        1.0            1.0  40.0 -2.0 -2.0
26        27         60000     1.0        1.0            2.0  27.0  1.0 -2.0
...      ...           ...     ...        ...            ...   ...  ...  ...
29961  29962        260000     1.0        1.0            2.0  33.0 -2.0 -2.0
29973  29974        230000     1.0        2.0            1.0  35.0  1.0 -2.0
29979  29980        180000     1.0        1.0            1.0  32.0 -2.0 -2.0
29983  29984         20000     1.0        2.0            1.0  44.0 -2.0 -2.0
29985  29986        240000     1.0        1.0            2.0  30.0 -2.0 -2.0

        X8   X9  ...     X15      X16      X17      X18     X19     X20  \
9     -2.0 -2.0  ...     0.0  13007.0  13912.0      0.0     0.0     0.0
18    -2.0 -2.0  ...     NaN      NaN      NaN      0.0     0.0     0.0
```

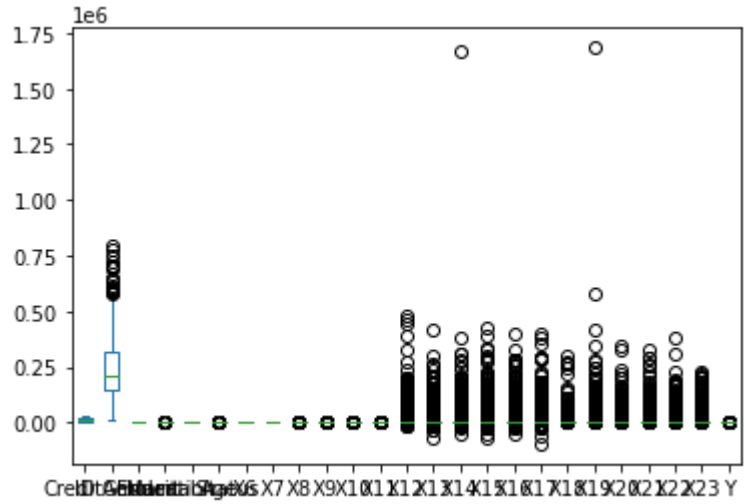
```
19    -2.0 -2.0  ...     0.0      0.0      0.0      0.0     0.0     0.0
23    -2.0 -2.0  ...   560.0      0.0      0.0  19428.0  1473.0   560.0
26    -1.0 -1.0  ...   -57.0    127.0   -189.0      0.0  1000.0     0.0
...     ...  ...  ...     ...      ...      ...      ...     ...     ...
29961 -2.0 -2.0  ...  1368.0    101.0    955.0    263.0     0.0  1368.0
29973 -2.0 -2.0  ...     0.0      0.0      0.0      0.0     0.0     0.0
29979 -2.0 -2.0  ...     0.0      0.0      0.0      0.0     0.0     0.0
29983 -2.0 -2.0  ...  2882.0   9235.0   1719.0   2890.0  2720.0  2890.0
29985 -2.0 -2.0  ...     0.0      0.0      0.0      0.0     0.0     0.0

           X21     X22     X23    Y
9      13007.0  1122.0     0.0  0.0
18         0.0     0.0     0.0  0.0
19         0.0     0.0     0.0  0.0
23         0.0     0.0  1128.0  1.0
26       500.0     0.0  1000.0  1.0
...        ...     ...     ...  ...
29961    101.0   955.0     0.0  0.0
29973      0.0     0.0     0.0  1.0
29979      0.0     0.0     0.0  0.0
29983   9263.0  1824.0  1701.0  0.0
29985      0.0     0.0     0.0  0.0

[3722 rows x 25 columns]
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecа
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecа
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
          ID  CreditAmount  Gender  Education  MaritalStatus   Age   X6   X7  \
9         10         20000     1.0        3.0            2.0  35.0 -2.0 -2.0
18        19        360000     2.0        1.0            1.0  49.0  1.0 -2.0
19        20        180000     2.0        1.0            2.0  29.0  1.0 -2.0
23        24        450000     2.0        1.0            1.0  40.0 -2.0 -2.0
33        34        500000     2.0        2.0            1.0  54.0 -2.0 -2.0
...      ...           ...     ...        ...            ...   ...  ...  ...
29979  29980        180000     1.0        1.0            1.0  32.0 -2.0 -2.0
29983  29984         20000     1.0        2.0            1.0  44.0 -2.0 -2.0
29984  29985         30000     1.0        2.0            2.0  38.0 -1.0 -1.0
29985  29986        240000     1.0        1.0            2.0  30.0 -2.0 -2.0
29986  29987        360000     1.0        1.0            2.0  35.0 -1.0 -1.0

        X8   X9  ...     X15      X16      X17      X18      X19     X20  \
9     -2.0 -2.0  ...     0.0  13007.0  13912.0      0.0      0.0     0.0
18    -2.0 -2.0  ...     NaN      NaN      NaN      0.0      0.0     0.0
19    -2.0 -2.0  ...     0.0      0.0      0.0      0.0      0.0     0.0
23    -2.0 -2.0  ...   560.0      0.0      0.0  19428.0   1473.0   560.0
33    -2.0 -2.0  ...  7521.0  71439.0   8981.0   4152.0  22827.0  7521.0
...     ...  ...  ...     ...      ...      ...      ...      ...     ...
29979 -2.0 -2.0  ...     0.0      0.0      0.0      0.0      0.0     0.0
29983 -2.0 -2.0  ...  2882.0   9235.0   1719.0   2890.0   2720.0  2890.0
29984 -2.0 -1.0  ...  1993.0   1907.0   3319.0    923.0   2977.0  1999.0
29985 -2.0 -2.0  ...     0.0      0.0      0.0      0.0      0.0     0.0
29986 -2.0 -2.0  ...     0.0      0.0      0.0      0.0      0.0     0.0

           X21     X22      X23    Y
9      13007.0  1122.0      0.0  0.0
18         0.0     0.0      0.0  0.0
```

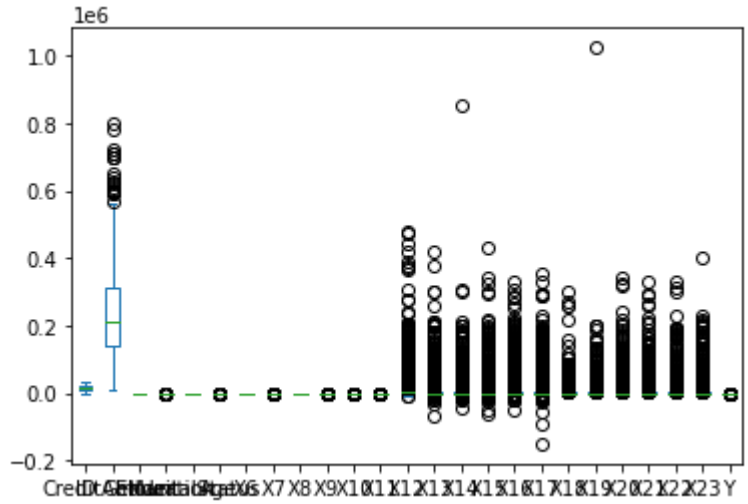
```
18         0.0     0.0      0.0  0.0
19         0.0     0.0      0.0  0.0
23         0.0     0.0   1128.0  1.0
33     71439.0   981.0  51582.0  0.0
...        ...     ...      ...  ...
29979      0.0     0.0      0.0  0.0
29983   9263.0  1824.0   1701.0  0.0
29984   3057.0  3319.0   1000.0  0.0
29985      0.0     0.0      0.0  0.0
29986      0.0     0.0      0.0  0.0

[4027 rows x 25 columns]
          ID  CreditAmount  Gender  Education  MaritalStatus   Age   X6   X7  \
9         10         20000     1.0        3.0            2.0  35.0 -2.0 -2.0
18        19        360000     2.0        1.0            1.0  49.0  1.0 -2.0
19        20        180000     2.0        1.0            2.0  29.0  1.0 -2.0
23        24        450000     2.0        1.0            1.0  40.0 -2.0 -2.0
33        34        500000     2.0        2.0            1.0  54.0 -2.0 -2.0
...      ...           ...     ...        ...            ...   ...  ...  ...
29979  29980        180000     1.0        1.0            1.0  32.0 -2.0 -2.0
29983  29984         20000     1.0        2.0            1.0  44.0 -2.0 -2.0
29985  29986        240000     1.0        1.0            2.0  30.0 -2.0 -2.0
29986  29987        360000     1.0        1.0            2.0  35.0 -1.0 -1.0
29992  29993         10000     1.0        3.0            1.0  43.0  0.0  0.0

        X8   X9  ...     X15      X16      X17      X18      X19     X20  \
9     -2.0 -2.0  ...     0.0  13007.0  13912.0      0.0      0.0     0.0
18    -2.0 -2.0  ...     NaN      NaN      NaN      0.0      0.0     0.0
19    -2.0 -2.0  ...     0.0      0.0      0.0      0.0      0.0     0.0
23    -2.0 -2.0  ...   560.0      0.0      0.0  19428.0   1473.0   560.0
33    -2.0 -2.0  ...  7521.0  71439.0   8981.0   4152.0  22827.0  7521.0
...    ...  ...  ...     ...      ...      ...      ...      ...     ...
29979 -2.0 -2.0  ...     0.0      0.0      0.0      0.0      0.0     0.0
29983 -2.0 -2.0  ...  2882.0   9235.0   1719.0   2890.0   2720.0  2890.0
29985 -2.0 -2.0  ...     0.0      0.0      0.0      0.0      0.0     0.0
29986 -2.0 -2.0  ...     0.0      0.0      0.0      0.0      0.0     0.0
29992  0.0 -2.0  ...     0.0      0.0      0.0   2000.0      0.0     0.0

           X21     X22      X23    Y
9      13007.0  1122.0      0.0  0.0
18         0.0     0.0      0.0  0.0
19         0.0     0.0      0.0  0.0
23         0.0     0.0   1128.0  1.0
33     71439.0   981.0  51582.0  0.0
...        ...     ...      ...  ...
29979      0.0     0.0      0.0  0.0
29983   9263.0  1824.0   1701.0  0.0
29985      0.0     0.0      0.0  0.0
29986      0.0     0.0      0.0  0.0
29992      0.0     0.0      0.0  0.0

[4287 rows x 25 columns]
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecа
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecа
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
          ID  CreditAmount  Gender  Education  MaritalStatus   Age   X6   X7  \
```

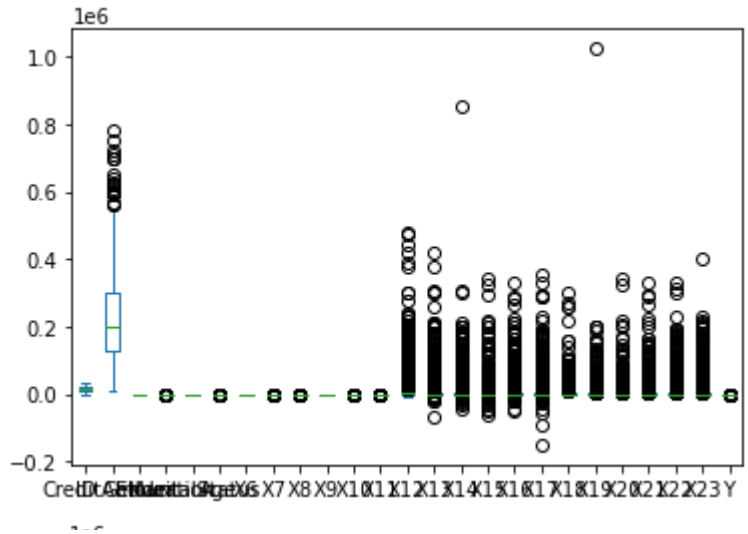
```
0          1         20000     2.0        2.0            1.0  24.0  2.0  2.0
18        19        360000     2.0        1.0            1.0  49.0  1.0 -2.0
19        20        180000     2.0        1.0            2.0  29.0  1.0 -2.0
23        24        450000     2.0        1.0            1.0  40.0 -2.0 -2.0
33        34        500000     2.0        2.0            1.0  54.0 -2.0 -2.0
...      ...           ...     ...        ...            ...   ...  ...  ...
29979  29980        180000     1.0        1.0            1.0  32.0 -2.0 -2.0
29983  29984         20000     1.0        2.0            1.0  44.0 -2.0 -2.0
29985  29986        240000     1.0        1.0            2.0  30.0 -2.0 -2.0
29986  29987        360000     1.0        1.0            2.0  35.0 -1.0 -1.0
29992  29993         10000     1.0        3.0            1.0  43.0  0.0  0.0

        X8   X9  ...     X15      X16     X17      X18      X19     X20  \
0     -1.0 -1.0  ...     0.0      0.0     0.0      0.0    689.0     0.0
18    -2.0 -2.0  ...     NaN      NaN     NaN      0.0      0.0     0.0
19    -2.0 -2.0  ...     0.0      0.0     0.0      0.0      0.0     0.0
23    -2.0 -2.0  ...   560.0      0.0     0.0  19428.0   1473.0   560.0
33    -2.0 -2.0  ...  7521.0  71439.0  8981.0   4152.0  22827.0  7521.0
...    ...  ...  ...     ...      ...     ...      ...      ...     ...
29979 -2.0 -2.0  ...     0.0      0.0     0.0      0.0      0.0     0.0
29983 -2.0 -2.0  ...  2882.0   9235.0  1719.0   2890.0   2720.0  2890.0
29985 -2.0 -2.0  ...     0.0      0.0     0.0      0.0      0.0     0.0
29986 -2.0 -2.0  ...     0.0      0.0     0.0      0.0      0.0     0.0
29992  0.0 -2.0  ...     0.0      0.0     0.0   2000.0      0.0     0.0

           X21     X22      X23    Y
0          0.0     0.0      0.0  1.0
18         0.0     0.0      0.0  0.0
19         0.0     0.0      0.0  0.0
23         0.0     0.0   1128.0  1.0
33     71439.0   981.0  51582.0  0.0
...        ...     ...      ...  ...
29979      0.0     0.0      0.0  0.0
29983   9263.0  1824.0   1701.0  0.0
29985      0.0     0.0      0.0  0.0
29986      0.0     0.0      0.0  0.0
29992      0.0     0.0      0.0  0.0

[4479 rows x 25 columns]
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecа
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecа
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
          ID  CreditAmount  Gender  Education  MaritalStatus   Age   X6   X7  \
0          1         20000     2.0        2.0            1.0  24.0  2.0  2.0
18        19        360000     2.0        1.0            1.0  49.0  1.0 -2.0
19        20        180000     2.0        1.0            2.0  29.0  1.0 -2.0
23        24        450000     2.0        1.0            1.0  40.0 -2.0 -2.0
33        34        500000     2.0        2.0            1.0  54.0 -2.0 -2.0
...      ...           ...     ...        ...            ...   ...  ...  ...
29983  29984         20000     1.0        2.0            1.0  44.0 -2.0 -2.0
29985  29986        240000     1.0        1.0            2.0  30.0 -2.0 -2.0
29986  29987        360000     1.0        1.0            2.0  35.0 -1.0 -1.0
29989  29990        150000     1.0        1.0            2.0  35.0 -1.0 -1.0
29992  29993         10000     1.0        3.0            1.0  43.0  0.0  0.0

         X8   X9          X15      X16     X17      X18      X19     X20  \
```

```
          X8   X9  ...     X15      X16     X17      X18      X19     X20  \
0      -1.0 -1.0  ...     0.0      0.0     0.0      0.0    689.0     0.0
18     -2.0 -2.0  ...     NaN      NaN     NaN      0.0      0.0     0.0
19     -2.0 -2.0  ...     0.0      0.0     0.0      0.0      0.0     0.0
23     -2.0 -2.0  ...   560.0      0.0     0.0  19428.0   1473.0   560.0
33     -2.0 -2.0  ...  7521.0  71439.0  8981.0   4152.0  22827.0  7521.0
...     ...  ...  ...     ...      ...     ...      ...      ...     ...
29983  -2.0 -2.0  ...  2882.0   9235.0  1719.0   2890.0   2720.0  2890.0
29985  -2.0 -2.0  ...     0.0      0.0     0.0      0.0      0.0     0.0
29986  -2.0 -2.0  ...     0.0      0.0     0.0      0.0      0.0     0.0
29989  -1.0 -1.0  ...   780.0      0.0     0.0   9054.0      0.0   783.0
29992   0.0 -2.0  ...     0.0      0.0     0.0   2000.0      0.0     0.0

            X21     X22      X23    Y
0           0.0     0.0      0.0  1.0
18          0.0     0.0      0.0  0.0
19          0.0     0.0      0.0  0.0
23          0.0     0.0   1128.0  1.0
33      71439.0   981.0  51582.0  0.0
...         ...     ...      ...  ...
29983    9263.0  1824.0   1701.0  0.0
29985       0.0     0.0      0.0  0.0
29986       0.0     0.0      0.0  0.0
29989       0.0     0.0      0.0  0.0
29992       0.0     0.0      0.0  0.0

[4806 rows x 25 columns]
```





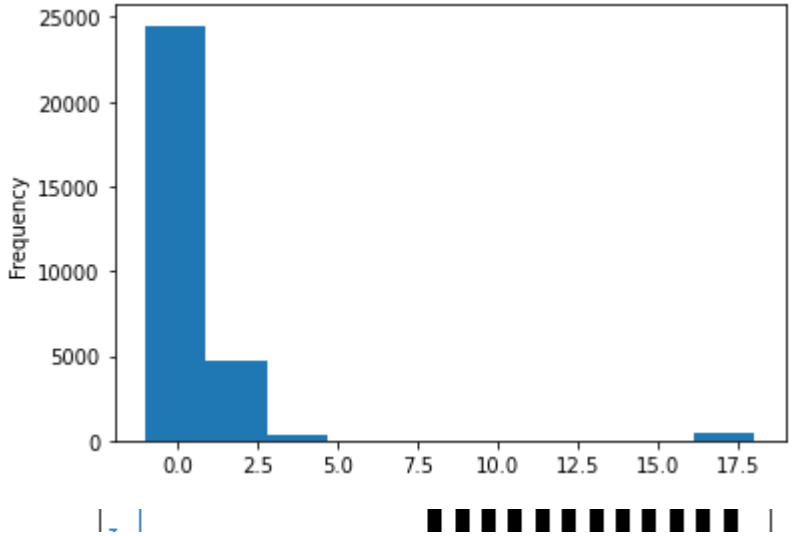




Se observa que hay valores negativos de todo tipo en esas columnas de pago

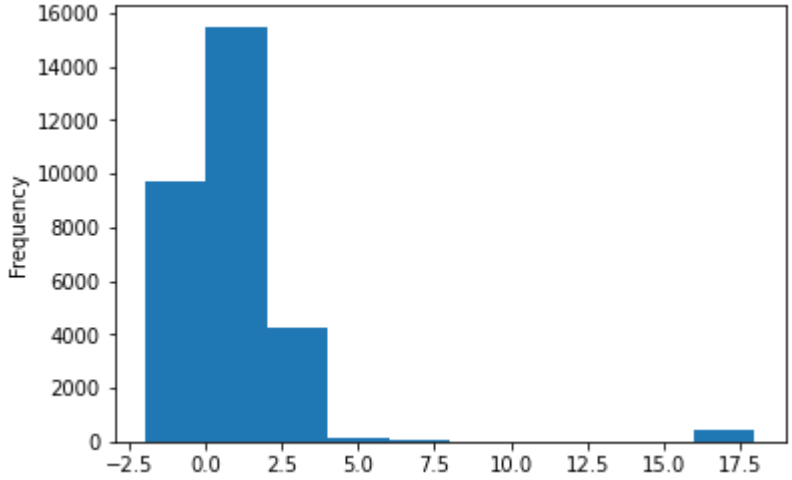
```
datos_creditos_1['X6'].plot.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373c9c8a50>
```



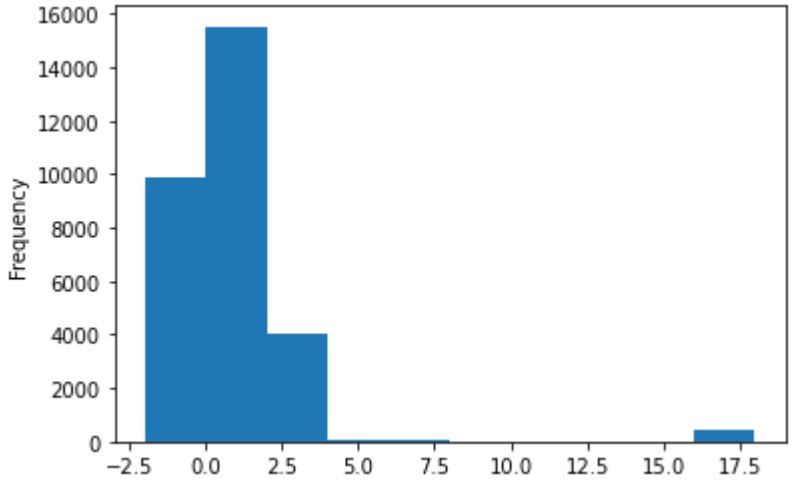
```
datos_creditos_1['X7'].plot.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373c8e1fd0>
```



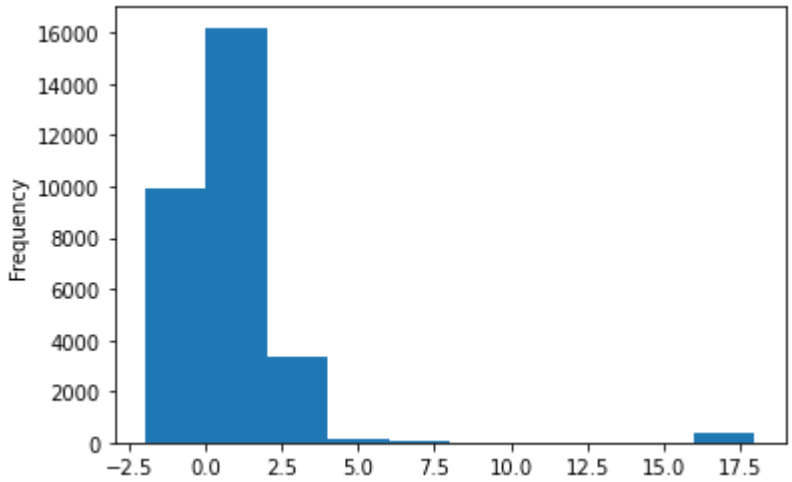

```
datos_creditos_1['X8'].plot.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373bff6090>
```



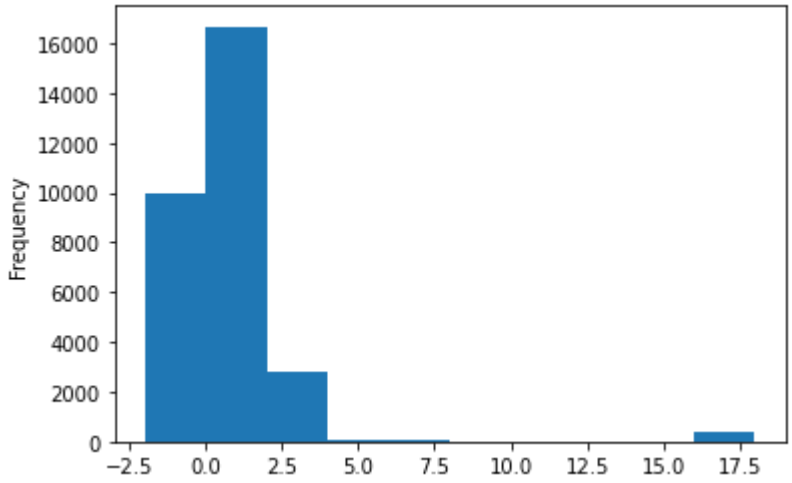

```
datos_creditos_1['X9'].plot.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373c558710>
```



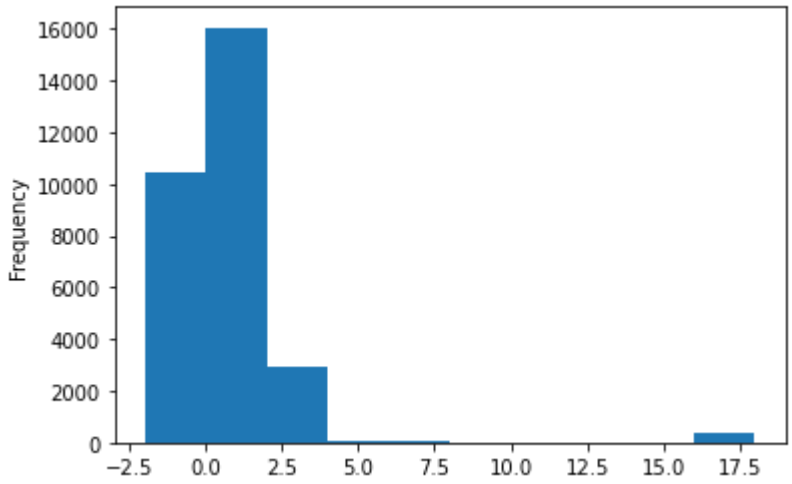
```
datos_creditos_1['X10'].plot.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373c3d8f90>
```




```
datos_creditos_1['X11'].plot.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373c3ee110>
```



Finalmente se observa que de las variables de x6 a x11 hay valores incongreuentes negativos de pago durante los 6 meses de observacion

```
for i in range(r6,r11+1): #Se crea un bucle para modificar los valores ngativos y agrparlos t
 datos_creditos_1[datos_creditos_1.iloc[:,i]<-1] =-1
```

```
x6=datos_creditos_1[datos_creditos_1['X6']<-1]
print(x6)
x11=datos_creditos_1[datos_creditos_1['X11']<-1]
print(x11)
```

```
Empty DataFrame
Columns: [ID, CreditAmount, Gender, Education, MaritalStatus, Age, X6, X7, X8, X9, X10,
```

```
Index: []

[0 rows x 25 columns]
Empty DataFrame
Columns: [ID, CreditAmount, Gender, Education, MaritalStatus, Age, X6, X7, X8, X9, X10,
Index: []

[0 rows x 25 columns]
```

Se comprueba que ya no hay valores negativos diferentes de -1

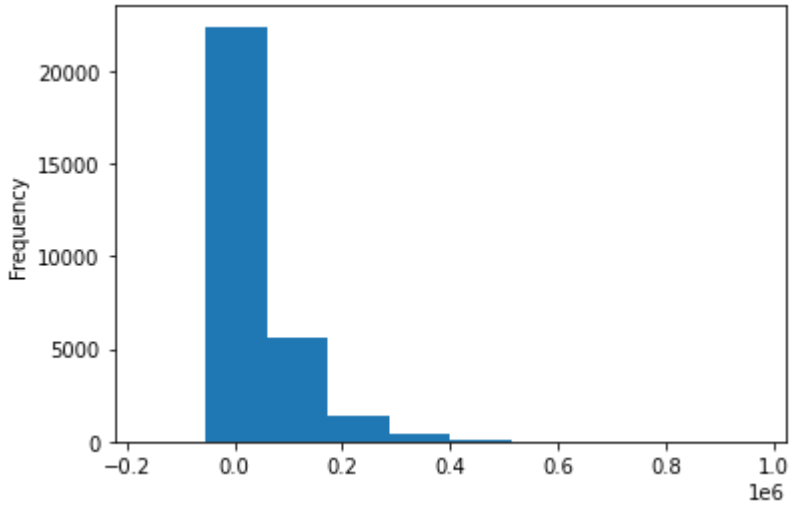
```
# Para las siguentes columnas se tiene lo siguiente. De igual manera al proceso anterior se r


r12=datos_creditos_1.columns.get_loc('X12') #Se busca el índice para la columna llamada 'X12'
r17=datos_creditos_1.columns.get_loc('X17') #Se busca el índice para la columna llamada 'X17'
for i in range(r12,r17+1):
  x12_x17=datos_creditos_1[datos_creditos_1.iloc[:,i]<-1]
  print(x12_x17)
```

```
           ID  CreditAmount  Gender  Education  MaritalStatus   Age   X6   X7  \
330       331        200000     2.0        2.0            1.0  37.0  1.0 -1.0
391       392        280000     2.0        2.0            1.0  39.0  1.0 -1.0
521       522        410000     2.0        2.0            2.0  27.0  1.0 -1.0
599       600        300000     2.0        2.0            1.0  36.0 -1.0 -1.0
1028     1029        200000     1.0        1.0            2.0  33.0  1.0 -1.0
...       ...           ...     ...        ...            ...   ...  ...  ...
29089   29090        360000     2.0        2.0            2.0  51.0  1.0 -1.0
29303   29304         30000     1.0        2.0            2.0  26.0  1.0 -1.0
29558   29559        200000     1.0        2.0            1.0  45.0  1.0 -1.0
29563   29564         20000     1.0        3.0            2.0  38.0  1.0 -1.0
29998   29999         80000     1.0        3.0            1.0  41.0  1.0 -1.0

         X8   X9  ...      X15      X16      X17      X18      X19     X20  \
330     2.0 -1.0  ...   6773.0  23209.0  16893.0   1000.0      0.0  6900.0
391     0.0  0.0  ...  37154.0  32581.0  33316.0  38621.0   2000.0  1130.0
521     0.0  0.0  ...  44310.0  62408.0  61420.0  33049.0  16000.0  8700.0
599     2.0  0.0  ...   2303.0   2768.0  65373.0  10000.0      0.0     0.0
1028   -1.0  0.0  ...   1873.0  90687.0   4577.0   3470.0   3191.0     0.0
...     ...  ...  ...      ...      ...      ...      ...      ...     ...
29089  -1.0 -1.0  ...   9944.0   4540.0   8024.0  15528.0   2145.0  9944.0
29303  -1.0  2.0  ...  29361.0  29022.0  28409.0  28924.0   4000.0     0.0
29558   0.0  0.0  ...  28355.0  29465.0  30885.0  34144.0   1492.0  1402.0
29563   0.0  0.0  ...  18161.0  18521.0  18911.0  18451.0   1300.0  1302.0
29998   0.0  0.0  ...  52774.0  11855.0  48944.0  85900.0   3409.0  1178.0

            X21      X22      X23    Y
330     24000.0  17000.0   2500.0  1.0
391       922.0   1011.0   5000.0  1.0
521     18775.0  15000.0   5350.0  0.0
599      2768.0  65373.0  10780.0  1.0
1028    90729.0      0.0   6266.0  0.0
```

```
...        ...      ...      ...  ...
29089   4540.0   8024.0   4421.0  0.0
29303   1000.0    900.0    800.0  0.0
29558   2002.0   2007.0   6163.0  0.0
29563    662.0    688.0    800.0  0.0
29998   1926.0  52964.0   1804.0  1.0

[142 rows x 25 columns]
          ID  CreditAmount  Gender  Education  MaritalStatus   Age   X6   X7  \
621      622        180000     2.0        2.0            2.0  28.0 -1.0 -1.0
685      686        360000     1.0        1.0            1.0  47.0 -1.0 -1.0
880      881        200000     1.0        1.0            2.0  30.0 -1.0 -1.0
932      933        160000     2.0        2.0            2.0  29.0  2.0  2.0
959      960        250000     1.0        1.0            2.0  41.0 -1.0 -1.0
...      ...           ...     ...        ...            ...   ...  ...  ...
27495  27496        170000     1.0        2.0            1.0  57.0 -1.0 -1.0
27576  27577        260000     1.0        1.0            1.0  59.0 -1.0 -1.0
27773  27774         80000     2.0        2.0            1.0  24.0  2.0  2.0
29589  29590        120000     1.0        2.0            1.0  33.0 -1.0 -1.0
29780  29781        100000     1.0        2.0            2.0  29.0  0.0  0.0

        X8   X9  ...       X15      X16     X17    X18       X19     X20  \
621   -1.0 -1.0  ...     460.0    460.0   610.0    0.0    1000.0   500.0
685   -1.0  0.0  ...  174397.0   4852.0  4125.0    0.0  177671.0  3508.0
880   -1.0  0.0  ...   11751.0  11772.0  5318.0    0.0   11663.0  9363.0
932   -1.0 -1.0  ...       0.0    892.0     0.0    0.0     874.0     0.0
```

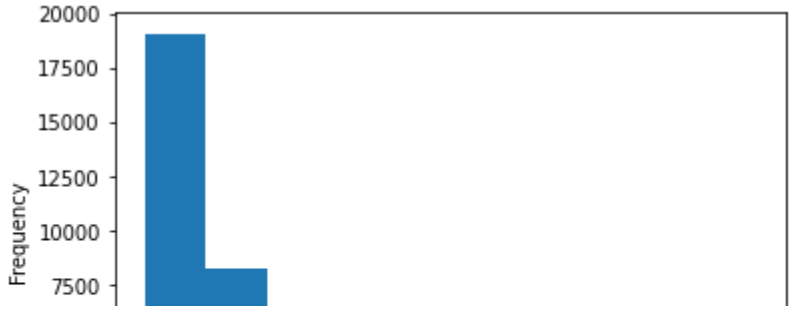
```
datos_creditos_1['X12'].plot.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373bd441d0>
```



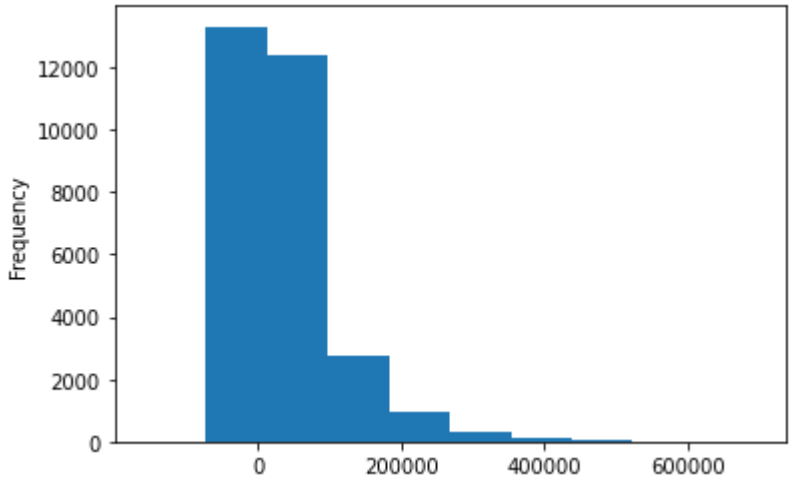

```
datos_creditos_1['X13'].plot.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373c33ce90>
```



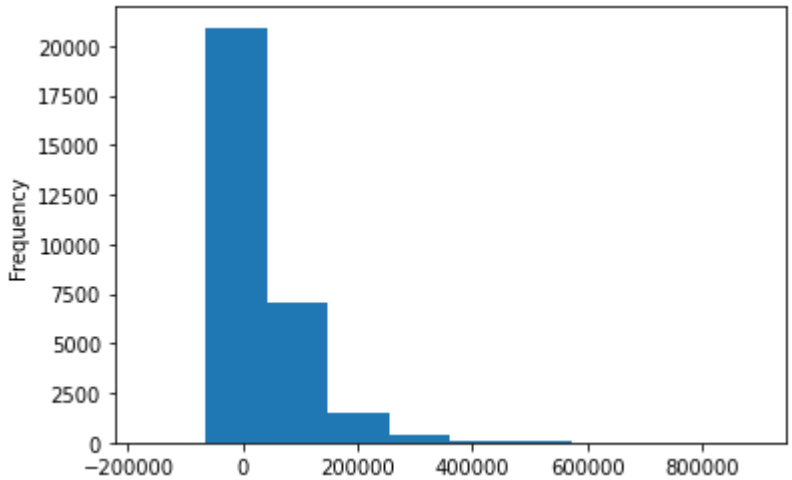

```
datos_creditos_1['X14'].plot.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373c2821d0>
```




```
datos_creditos_1['X15'].plot.hist() # se observan valores negativos
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373c4e0350>
```

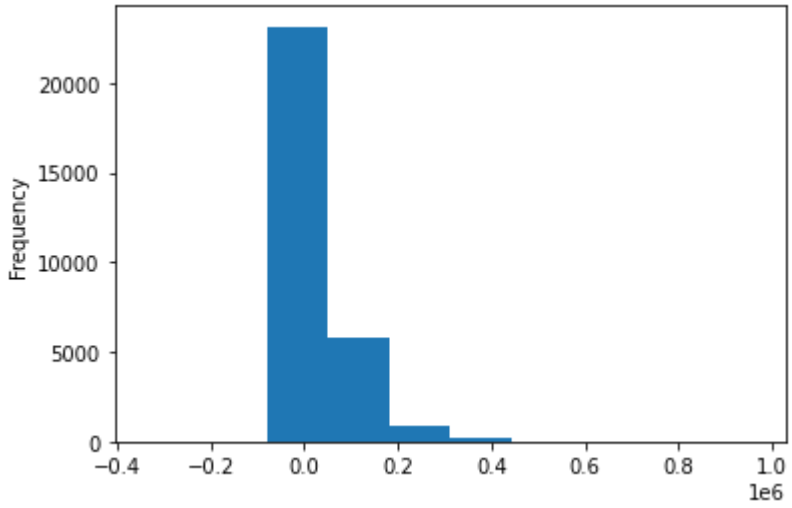



```
datos_creditos_1['X16'].plot.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373899a490>
```


```
datos_creditos_1['X17'].plot.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f373c2b92d0>
```



Al observar que se tienen valores negativos por lo que se puede agrupar en -1

```
for i in range(r12,r17+1):
  datos_creditos_1[datos_creditos_1.iloc[:,i]<-1]*-1 # para reclasificar todos los valores qu


# Para las siguentes columnas se tiene lo siguiente. De igual manera al proceso anterior se r


r18=datos_creditos_1.columns.get_loc('X18') #Se busca el índice para la columna llamada 'X18'
r23=datos_creditos_1.columns.get_loc('X23') #Se busca el índice para la columna llamada 'X23'
for i in range(r18,r23+1):
  x18_x23=datos_creditos_1[datos_creditos_1.iloc[:,i]<-1]
  print(x18_x23)
```

```
Empty DataFrame
Columns: [ID, CreditAmount, Gender, Education, MaritalStatus, Age, X6, X7, X8, X9, X10,
Index: []

[0 rows x 25 columns]
Empty DataFrame
Columns: [ID, CreditAmount, Gender, Education, MaritalStatus, Age, X6, X7, X8, X9, X10,
```

```
Index: []

[0 rows x 25 columns]
Empty DataFrame
Columns: [ID, CreditAmount, Gender, Education, MaritalStatus, Age, X6, X7, X8, X9, X10,
Index: []

[0 rows x 25 columns]
Empty DataFrame
Columns: [ID, CreditAmount, Gender, Education, MaritalStatus, Age, X6, X7, X8, X9, X10,
Index: []

[0 rows x 25 columns]
Empty DataFrame
Columns: [ID, CreditAmount, Gender, Education, MaritalStatus, Age, X6, X7, X8, X9, X10,
Index: []

[0 rows x 25 columns]
Empty DataFrame
Columns: [ID, CreditAmount, Gender, Education, MaritalStatus, Age, X6, X7, X8, X9, X10,
Index: []

[0 rows x 25 columns]
```

Se onserva que no hay daos negativos en las siguientes variables

```
for i in range(r18,r23+1):
  datos_creditos_1[datos_creditos_1.iloc[:,i]<-1]*-1
```

Solucion para datos faltantes

> Como observamos si hay datos faltantes por lo que se consideramos eliminar variables que tengan mas nulls que datos pues no nos ayudaran a describir

```
datos_creditos_1.dropna(how='all', inplace = True)
```

```
# consideramos que mas del 5% entonces ya nodeberiamos contemplar esa variable
# total de observaciones =29998

n=29998*0.05
n
```

```
1499.9
```

```
#Se puede aliminar a partir de 1000 datos faltantes
for i in range(1000,1499):
  datos_creditos_1.dropna(thresh=i, inplace = True)
```

```
#Para revisar los NA en la edad se puede sustituir con la moda porque es mas asertado que la
datos_creditos_1[datos_creditos_1['Age'].isnull()]
#moda_Age= mode(datos_creditos_1['Age'])
#print(moda_Age)
#datos_creditos_1['Age'].fillna(moda_Age, inplace=True)
```

	ID	CreditAmount	Gender	Education	MaritalStatus	Age	X6	X7	X8	X9	...	X15	X16

0 rows × 25 columns

```
## para datos faltantes se puede remplazar por la media
#for i in range(r12,r23+1):
#  datos_creditos_1.iloc[:,i].fillna(mode(datos_creditos_1.iloc[:,i]), inplace=True)
#  print('median x'+str(i),':',datos_creditos_1.iloc[:,i].median())


datos_creditos_1[datos_creditos_1['Y'].isnull()]
datos_creditos_1['Y'].dropna(inplace=True)
datos_creditos_1[datos_creditos_1['Y'].isnull()]
```

	ID	CreditAmount	Gender	Education	MaritalStatus	Age	X6	X7	X8	X9	...	X15	X16

0 rows × 25 columns

```
print(datos_creditos_1.isnull().sum())
```

```
ID              0.0
CreditAmount    0.0
Gender          0.0
Education       0.0
MaritalStatus   0.0
Age             0.0
X6              0.0
X7              0.0
X8              0.0
X9              0.0
X10             0.0
X11             0.0
X12             0.0
X13             0.0
X14             0.0
X15             0.0
X16             0.0
X17             0.0
X18             0.0
X19             0.0
X20             0.0
X21             0.0
X22             0.0
```

```
X23             0.0
Y               0.0
dtype: float64
```

Parte 3: Fundamentos de bases de datos

Con base en los resultados de tu libreta de Google Colab de la Parte 2 responde detalladamente las siguientes preguntas:

- ¿Qué datos considero mas importantes? ¿Por qué?

> Los datos mas relevantes son el monto del crédito porque nos descrive como se encuentra la cartera de estudio. Por otro lado, tambien es relevante el comportamiento de pago de los 6 meses mas recientes. Tambien es un factor importante la edad porque usualmente asociado a la etapa de vida es el ingreso y con ello l aposibilidad de adquirir mayor responsabilidad crediticia. Por otro lado, las variables que tienen mayor importancia son aquellas donde no existen tantos valores vacios o null, pues al final pueden considerarse dentro de los test de significacncia de las variables.

- ¿Se eliminaron o reemplazaron datos nulos? ¿Qué se hizo y por qué?

> Para las columnas que tenian mas del 5% en datos nulos era preferible no cosnsiderar la variable. Para los datos que tenian valores con etiquetas distintas a las categorias por definicion de la variable se hicieron ajustes por lo que se sobreescrbrio en categoria "otros" aquellos vslores que se desconocia su tipo y pudo deberse a un error de la obtencion de la informacion. Por otro lado para valores que sabemos que debieron ser positivos como los pagos del cliente o el monto de los prestamos.

- ¿Es necesario ordenar los datos para el análisis? Sí / No / ¿Por qué?

> No necesariamente es necesario ordenar sin embargo general graficos de distribucion permitio tener una idea generalizada de como se encuentran los valores e incluso observar atipicos. La idea general es estandarizar o asegurarnos que los valores de acuerdo a las descripciones dadas de sus variables sean correctos.

- ¿Existen problemas de formato que deban solucionar antes del proceso de modelado? Sí / No / Por qué.

> Sí existen problemas que debemos solucionar antes de proseguir con el modelado, tales como verificar que el monto de crédito otorgado contenga valores positivos, que los datos capturados para las variables de género, educación y estado civil

> concuerden con la definicion original, la edad corresponda a mayores de 18 años y que el monto de la deuda, pagos anteriores e historial de pago no posean valores negativos porque se saldrian del contexto sobre el que son aplicados. Debía realizarse porque si los valores se salían de los estándares que en la misma descripción de las variables contemplaba simplemente eran datos inexistentes o inválidos en cuanto a su clasificación, lo cual, en el conjunto de datos sí sucedía y debía corregirse, además se debe incluir la excepcion de manejo de datos que pudieran no tener el valor de la definicion.

- ¿Qué ajustes se realizaron en el proceso de limpieza de datos (agregar, integrar, eliminar, modificar registros (filas), cambiar atributos (columnas)?

> Se tuvieron que eliminar aquellas filas en las que la mayoria de los datos no tenian sufiencientes datos, puesto que no resultaban representativas y no permitian tomar decisiones adecuadas.

> Por otro lado, la imputación media para preservar la distribución de los datos y no sacrificar su demás información al eliminarlos. Asimismo, se asignaron las categorías alternativas cuando la de la base de datos resulta incongruente, se consideró un género aleatorio para sustituir los valores ilógicos ya que elegir solo un género podría causar sesgo y otros métodos generar valores no binarios, la moda en hábitos de pago se adecuo porque es posible observar ese tipo de tendencia, multiplicación por -1 para los elementos negativos, ya que habia negativos de distintos valroes. Para la variable target y es muy importante no contar con datos faltantes ya que al ser sobre la que se entrenaria es muy importante contar cn todas las etiquetas de buenos pagadores.

Productos de pago de Colab - Cancelar contratos