Primer Parcial Data Science . 24 Marzo 2021

Cyndy Elizabeth Pantoja Tamayo

In [79]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

1. Preprocesamiento de Datos .

1.1 Cargar Dataset en Phyton o Rstudio

In [80]:

```
data = pd.read_csv ('credit-german.csv', sep=";")
```

1.2 Número de instancias

In [81]:

```
data.shape[0]
```

Out[81]:

```
1000
```

1.3 Número de atributos

In [82]:

```
data.shape[1]
```

Out[82]:

```
19
```

1.4 ¿El conjunto de datos está etiquetado? ¿Cuántas clases tiene el conjunto de datos?

In [46]:

```
data.columns
```

Out[46]:

```
Index(['checking_status', 'disc_duration', 'credit_history', 'purpose',
       'credit_amount', 'savings_status', 'employment', 'personal_status',
       'other_parties', 'property_magnitude', 'age', 'other_payment_plans',
       'housing', 'existing_credits', 'job', 'num_dependents', 'own_telephone',
       'foreign_worker', 'class'],
      dtype='object')
```

1.5 ¿Cuántos atributos son numéricos y cuántos categóricos?

In [47]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 19 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   checking_status      1000 non-null   object
 1   disc_duration        1000 non-null   int64 
 2   credit_history       1000 non-null   object
 3   purpose              1000 non-null   object
 4   credit_amount        1000 non-null   int64 
 5   savings_status       1000 non-null   object
 6   employment           1000 non-null   object
 7   personal_status      1000 non-null   object
 8   other_parties        1000 non-null   object
 9   property_magnitude   1000 non-null   object
 10  age                  1000 non-null   int64 
 11  other_payment_plans  1000 non-null   object
 12  housing              1000 non-null   object
 13  existing_credits     1000 non-null   object
 14  job                  1000 non-null   object
 15  num_dependents       1000 non-null   object
 16  own_telephone        1000 non-null   object
 17  foreign_worker       1000 non-null   object
 18  class                1000 non-null   object
dtypes: int64(3), object(16)
memory usage: 148.6+ KB
```

1.6 Reporte la moda para cada atributo categórico

In [48]:

```
data.mode(numeric_only=False)
```

Out[48]:

	checking_status	disc_duration	credit_history	purpose	credit_amount	savings_status	employment	personal_status	other_parties	property_r
0	no checking	24.0	existing paid	radio/tv	1258	<100	1<=X<4	male single	none	
1	NaN	NaN	NaN	NaN	1262	NaN	NaN	NaN	NaN	
2	NaN	NaN	NaN	NaN	1275	NaN	NaN	NaN	NaN	
3	NaN	NaN	NaN	NaN	1393	NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	NaN	1478	NaN	NaN	NaN	NaN	

1.7 Reporte la media, rango y desviación estándar para cada atributo numérico

In [49]:

```
data.describe()
```

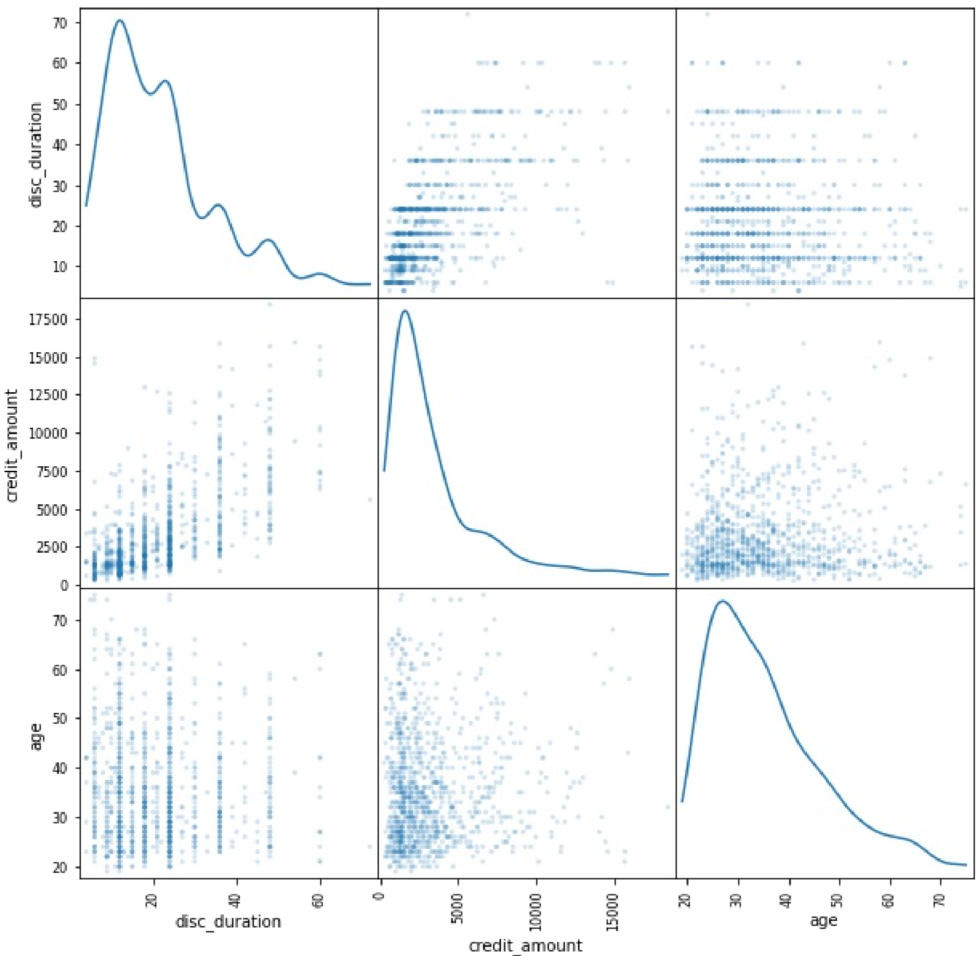
Out[49]:

	disc_duration	credit_amount	age
count	1000.000000	1000.000000	1000.000000
mean	20.903000	3271.258000	35.546000
std	12.058814	2822.736876	11.375469
min	4.000000	250.000000	19.000000
25%	12.000000	1365.500000	27.000000
50%	18.000000	2319.500000	33.000000
75%	24.000000	3972.250000	42.000000
max	72.000000	18424.000000	75.000000

1.8 Determine la distribución de las clases (Diagrama de Densidad)

In [83]:

```
pd.plotting.scatter_matrix(data, alpha=0.2, figsize=(10, 10), diagonal='density')
plt.show()
```



1.9 Escoja una técnica para la detección de datos atípicos y aplíquela sobre el conjunto de datos

In [85]:

```
IQR_Age = data['age'].quantile(0.75) - data['age'].quantile(0.25)
upper_Age = data['age'].quantile(0.75) + 1.5*IQR_Age
lower_Age = data['age'].quantile(0.25) -1.5*IQR_Age
atipicos_Age = data['age'][(data['age']>upper_Age)|(data['age']>lower_Age)]
atipicos_Age
```

Out[85]:

```
0      67
1      22
2      49
3      45
4      53
       ..
995    31
996    40
997    38
998    23
999    27
Name: age, Length: 1000, dtype: int64
```

In [86]:

```
IQR_credit_amount = data['credit_amount'].quantile(0.75) - data['credit_amount'].quantile(0.25)
upper_credit_amount = data['credit_amount'].quantile(0.75) + 1.5*IQR_credit_amount
lower_credit_amount = data['credit_amount'].quantile(0.25) -1.5*IQR_credit_amount
atipicos_credit_amount = data['credit_amount'][(data['credit_amount']>upper_credit_amount)|(data['credit_
atipicos_credit_amount
```

Out[86]:

```
0      1169
1      5951
2      2096
3      7882
4      4870
       ...
995    1736
996    3857
997     804
998    1845
999    4576
Name: credit_amount, Length: 1000, dtype: int64
```

In [87]:

```
IQR_disc_duration = data['disc_duration'].quantile(0.75) - data['disc_duration'].quantile(0.25)
upper_disc_duration = data['disc_duration'].quantile(0.75) + 1.5*IQR_disc_duration
lower_disc_duration = data['disc_duration'].quantile(0.25) -1.5*IQR_disc_duration
atipicos_disc_duration = data['disc_duration'][(data['disc_duration']>upper_disc_duration)|(data['disc_du
atipicos_disc_duration
```

Out[87]:

```
0       6
1      48
2      12
3      42
4      24
       ..
995    12
996    30
997    12
998    45
999    45
Name: disc_duration, Length: 1000, dtype: int64
```

1.10 Aplique al menos dos estrategias diferentes para manejar los datos faltantes

In [50]:

```
data.isnull().sum()
```

Out[50]:

```
checking_status         0
disc_duration           0
credit_history          0
purpose                 0
credit_amount           0
savings_status          0
employment              0
personal_status         0
other_parties           0
property_magnitude      0
age                     0
other_payment_plans     0
housing                 0
existing_credits        0
job                     0
num_dependents          0
own_telephone           0
foreign_worker          0
class                   0
dtype: int64
```

1.11 Convierta todas los atributos numéricos a categóricos

In [51]:

```
conditionlist = [(data['disc_duration']<=12),(data['disc_duration']>12)&(data['disc_duration']<=36),(data
catlist = ['Corto', 'Mediano', 'Largo']
data['disc_duration'] = np.select (conditionlist, catlist, default = 'NA')
data
```

Out[51]:

	checking_status	disc_duration	credit_history	purpose	credit_amount	savings_status	employment	personal_status	other_partie
0	<0	Corto	critical/other existing	radio/tv	1169	no known savings	>=7	male single	none
1	0<=X<200	Largo	existing paid	radio/tv	5951	<100	1<=X<4	female div/dep/mar	none
2	no checking	Corto	critical/other existing	education	2096	<100	4<=X<7	male single	none
3	<0	Largo	existing paid	furniture/equipment	7882	<100	4<=X<7	male single	guarantor
4	<0	Mediano	delayed previously	new car	4870	<100	1<=X<4	male single	none
...	...	...	...	...	...	...	...	...	..
995	no checking	Corto	existing paid	furniture/equipment	1736	<100	4<=X<7	female div/dep/mar	none
996	<0	Mediano	existing paid	used car	3857	<100	1<=X<4	male div/sep	none
997	no checking	Corto	existing paid	radio/tv	804	<100	>=7	male single	none
998	<0	Largo	existing paid	radio/tv	1845	<100	1<=X<4	male single	none
999	0<=X<200	Largo	critical/other existing	used car	4576	100<=X<500	unemployed	male single	none

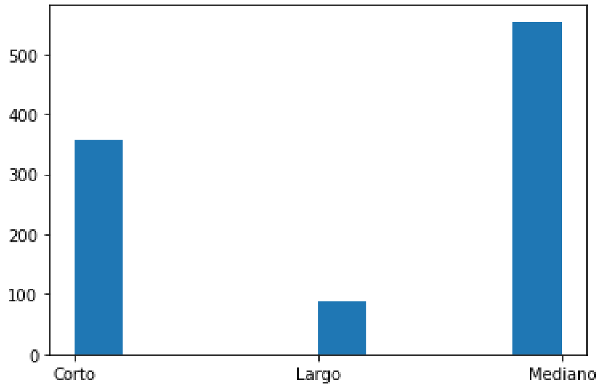
1000 rows × 19 columns

In [55]:

```
plt.hist(data['disc_duration'])
```

Out[55]:

```
(array([359.,   0.,   0.,   0.,   0.,  87.,   0.,   0.,   0., 554.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <BarContainer object of 10 artists>)
```



In [52]:

```
conditionlist = [(data['credit_amount']<=5000),(data['credit_amount']>5000)&(data['credit_amount']<=1200(
catlist = ['Bajo', 'Media', 'Alto']
data['credit_amount'] = np.select (conditionlist, catlist, default = 'NA')
data
```

Out[52]:

	checking_status	disc_duration	credit_history	purpose	credit_amount	savings_status	employment	personal_status	other_partie
0	<0	Corto	critical/other existing	radio/tv	Bajo	no known savings	>=7	male single	none
1	0<=X<200	Largo	existing paid	radio/tv	Media	<100	1<=X<4	female div/dep/mar	none
2	no checking	Corto	critical/other existing	education	Bajo	<100	4<=X<7	male single	none
3	<0	Largo	existing paid	furniture/equipment	Media	<100	4<=X<7	male single	guaranto
4	<0	Mediano	delayed previously	new car	Bajo	<100	1<=X<4	male single	none
...	...	...	...	...	...	...	...	...	..
995	no checking	Corto	existing paid	furniture/equipment	Bajo	<100	4<=X<7	female div/dep/mar	none
996	<0	Mediano	existing paid	used car	Bajo	<100	1<=X<4	male div/sep	none
997	no checking	Corto	existing paid	radio/tv	Bajo	<100	>=7	male single	none
998	<0	Largo	existing paid	radio/tv	Bajo	<100	1<=X<4	male single	none
999	0<=X<200	Largo	critical/other existing	used car	Bajo	100<=X<500	unemployed	male single	none

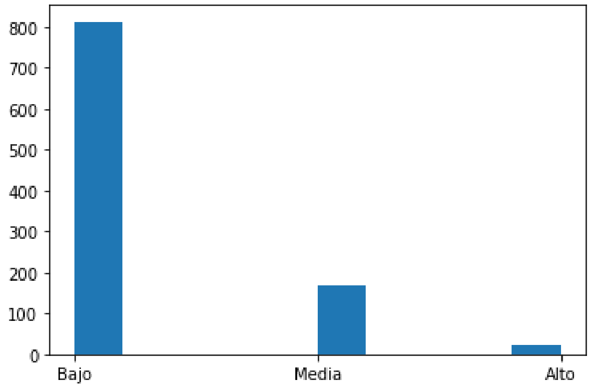
1000 rows × 19 columns

In [56]:

```
plt.hist(data['credit_amount'])
```

Out[56]:

```
(array([812.,   0.,   0.,   0.,   0., 167.,   0.,   0.,   0.,  21.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <BarContainer object of 10 artists>)
```



In [53]:

```
conditionlist = [(data['age']<=24),(data['age']>24)&(data['age']<=60),(data['age']>60)]
catlist = ['Joven', 'Adulto', 'Mayor']
data['age'] = np.select (conditionlist, catlist, default = 'NA')
data
```

Out[53]:

	checking_status	disc_duration	credit_history	purpose	credit_amount	savings_status	employment	personal_status	other_parties
0	<0	Corto	critical/other existing	radio/tv	Bajo	no known savings	>=7	male single	none
1	0<=X<200	Largo	existing paid	radio/tv	Media	<100	1<=X<4	female div/dep/mar	none
2	no checking	Corto	critical/other existing	education	Bajo	<100	4<=X<7	male single	none
3	<0	Largo	existing paid	furniture/equipment	Media	<100	4<=X<7	male single	guarantor
4	<0	Mediano	delayed previously	new car	Bajo	<100	1<=X<4	male single	none
...	...	...	...	...	...	...	...	...	...
995	no checking	Corto	existing paid	furniture/equipment	Bajo	<100	4<=X<7	female div/dep/mar	none
996	<0	Mediano	existing paid	used car	Bajo	<100	1<=X<4	male div/sep	none
997	no checking	Corto	existing paid	radio/tv	Bajo	<100	>=7	male single	none
998	<0	Largo	existing paid	radio/tv	Bajo	<100	1<=X<4	male single	none
999	0<=X<200	Largo	critical/other existing	used car	Bajo	100<=X<500	unemployed	male single	none

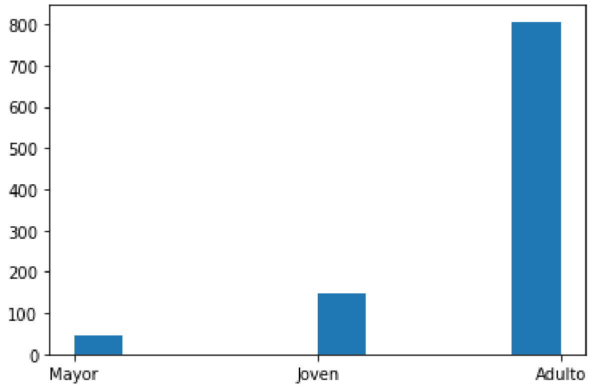
1000 rows × 19 columns

In [57]:

```
plt.hist(data['age'])
```

Out[57]:

```
(array([ 45.,   0.,   0.,   0.,   0., 149.,   0.,   0.,   0., 806.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <BarContainer object of 10 artists>)
```



1.12 Transforme el conjunto de datos de manera que todos los atributos sean numéricos

In [54]:

```
data2 = pd.read_csv ('credit-german.csv', sep=";")
```

In [65]:

```
v = pd.get_dummies(data2,columns=['own_telephone'])
v
```

Out[65]:

	checking_status	disc_duration	credit_history	purpose	credit_amount	savings_status	employment	personal_status	other_partie
0	<0	6	critical/other existing	radio/tv	1169	no known savings	>=7	male single	none
1	0<=X<200	48	existing paid	radio/tv	5951	<100	1<=X<4	female div/dep/mar	none
2	no checking	12	critical/other existing	education	2096	<100	4<=X<7	male single	none
3	<0	42	existing paid	furniture/equipment	7882	<100	4<=X<7	male single	guaranto
4	<0	24	delayed previously	new car	4870	<100	1<=X<4	male single	none
...	...	...	...	...	...	...	...	...	..
995	no checking	12	existing paid	furniture/equipment	1736	<100	4<=X<7	female div/dep/mar	none
996	<0	30	existing paid	used car	3857	<100	1<=X<4	male div/sep	none
997	no checking	12	existing paid	radio/tv	804	<100	>=7	male single	none
998	<0	45	existing paid	radio/tv	1845	<100	1<=X<4	male single	none
999	0<=X<200	45	critical/other existing	used car	4576	100<=X<500	unemployed	male single	none

1000 rows × 20 columns

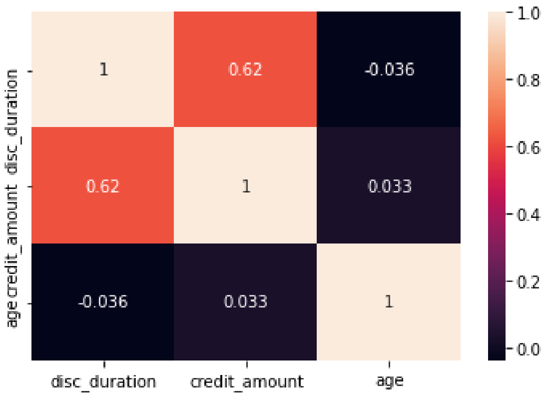
1.13 Pruebe diferentes combinaciones entre los atributos y establezca las relaciones entre ellos, reporte la herramienta de visualización que utilizó para tal fin. (Matriz de Corelacion)

In [88]:

```
data3 = pd.read_csv ('credit-german.csv', sep=";")
correlacion = data3.corr()
sns.heatmap(correlacion, annot=True)
```

Out[88]:

<AxesSubplot:>



1.14 ¿Cuál es lo propósito predominante de los préstamos?

In [ ]:

1.15 ¿Qué tipo de estatus tienen las personas que más hacen préstamos? ¿Y el perfil de la de menos préstamos? ¿Cuál es el perfil de las personas que hacen los prestamos más costoso? ¿Y el de los menos costosos?

In [69]:

```
print (data.groupby('personal_status').size())
```

```
personal_status
female div/dep/mar    310
male div/sep           50
male mar/wid           92
male single           548
dtype: int64
```

In [ ]:

1.16 ¿Puede establecer alguna relación entre edad, estatus personal y la clase?

In [ ]:

1.17 ¿Puede establecer alguna relación entre clase de trabajo, el número de créditos, estatus personal y la clase?

In [ ]:

1.18 ¿Existe alguna relación entre la cantidad solicitada y el número de meses del préstamo?

In [ ]:

1.19 ¿Existe alguna relación entre la edad, el estatus, la clase y la cantidad del préstamo?

In [ ]:

1.20 Proponga Dos preguntas y resuélvalas a partir de técnicas de análisis de la varianza permite contraste de hipótesis y coeficiente de correlación de Pearson

In [ ]: