

UNIVERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

RESULTADO DEL LABORATORIO 01

CURSO:

INTELIGENCIA DE NEGOCIOS

DOCENTE:

Ing. Patrick Cuadros Quiroga

ESTUDIANTE:

Escalante Maron, Nelia

(2014049551)

Índice

1. Desarrollo	2
2. Conclusiones	7

1. Desarrollo

LABORATORIO 01

Crear un reporte asociado a los datos de negocio
Basado en el conjunto de datos credit data.csv”

Crear un reporte con la siguiente descripción:

- Periodo de analisis: de 2014 a 2015
 - Unidad de Tiempo: Trimestral (y total anual)
 - Grupos de analisis: Por ingreso mensual con grupos.
 - menos de 2.500
 - 2.500 a 5.000
 - 5.000 a 10.000
 - mas de 10.000
 - KPIs:
 - Número de Créditos
 - delincuencia promedio (porcentaje de clientes en morosidad de 90dpd) ingreso mensual promedio
 - Estructura de tabla:
 - cuartos en columnas
 - Grupos y KPIs en filas.
 - Dibuja una gráfica con:
 - Barras apiladas para el número de créditos
 - Líneas para la morosidad media
- Adicional, Analisis de Data Exploratorio:

- Crear un par gráfico de parejas para todas las variables
- Crear un gráfico con Ingresos VS Edad en el medio, más distribuciones marginales
- Importar Informacion

Antes de comenzar tenemos que importar el archivo excel



Importamos librerías y las describimos a continuación:

- numpy: NumPy es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.
- pandas: pandas es una API de análisis de datos en columnas, ideal para manipular y analizar datos de entrada. Además, muchos marcos de trabajo de AA admiten las estructuras de datos pandas como entradas. Si bien una introducción detallada a la API de pandas abarcaría muchas páginas, los conceptos principales que presentamos a continuación son simples. Para obtener una referencia más completa, el sitio de documentación de pandas incluye una documentación exhaustiva y numerosos instructivos.
- matplotlib.pyplot: Matplotlib es una librería de trazado utilizada para gráficos 2D en lenguaje de programación Python, es muy flexible y tiene muchos valores predeterminados incorporados que te ayudarán muchísimo en tu trabajo. Como tal, no necesitas mucho para comenzar, solamente tienes que hacer las importaciones necesarias, preparar algunos datos y con esto puedes comenzar a trazar tu función con la ayuda de la instrucción plot(). Veamos esto en un ejemplo.
- seaborn: Para asegurarnos que todos los paquetes están instalados.
- De esta manera importamos la librería:

```
[ ] # import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Importamos el archivo

```
# Import the data
df = pd.read_csv('credit data.csv', sep=';')
```

68734	68735	1	0.962415	30	0	0.289087	2583.0
68735	68736	0	0.168333	64	0	0.349041	7299.0
68736	68737	0	0.034568	53	0	0.076892	10000.0
68737	68738	0	0.015323	59	0	0.212383	10400.0
68738	68739	0	0.023626	46	0	4901.000000	NaN
68739	68740	0	0.583650	63	0	4427.000000	NaN
68740	68741	0	0.924513	37	0	0.066373	2500.0
68741	68742	0	0.000000	64	0	0.130573	8500.0

revisamos los tipos

```
[ ] # check types
df.dtypes
```

Id	int64
SeriousDlqin2yrs	int64
RevolvingUtilizationOfUnsecuredLines	float64
age	int64
NumberOfTime30-59DaysPastDueNotWorse	int64
DebtRatio	float64
MonthlyIncome	float64
NumberOfOpenCreditLinesAndLoans	float64
NumberOfTimes90DaysLate	float64
NumberRealEstateLoansOrLines	float64
NumberOfTime60-89DaysPastDueNotWorse	float64
NumberOfDependents	float64
cohort_date	object
dtype:	object

verifica que las fechas estén bien cargadas

```
[ ] # check the dates are well loaded
df.count()
```

Id	68764
SeriousDlqin2yrs	68764
RevolvingUtilizationOfUnsecuredLines	68764
age	68764
NumberOfTime30-59DaysPastDueNotWorse	68764
DebtRatio	68764
MonthlyIncome	55128
NumberOfOpenCreditLinesAndLoans	68763
NumberOfTimes90DaysLate	68763
NumberRealEstateLoansOrLines	68763
NumberOfTime60-89DaysPastDueNotWorse	68763
NumberOfDependents	66956
cohort_date	68763
dtype:	int64

- Imputación de valores faltantes.

```
[ ] # check missings
df.isnull().sum().sum()
```

15449

```
[ ] # missing imputation
df['MonthlyIncome'].isnull()
```

4 False
5 False
6 True
7 False
8 True
9 False
10 False
11 False
12 False
13 False
14 False
15 False
16 True
17 False
18 False
19 False
20 False
21 False
22 False
23 False
24 False
25 False
26 False
27 False
28 False
29 False

- Seleccione la muestra de datos para el informe

```
[ ] # filter by date
df['cohort_date'] = pd.to_datetime(df['cohort_date'])
filtro_date = '17-09-2016'
mask = (df['cohort_date'] > filtro_date)
df = df.loc[mask]
df
```

	Id	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTime30-59DaysPastDueNotWorse	DebtRatio	MonthlyIncome
17	18	0	0.166284	53	0	0.188274	8800.0
38	39	0	0.363636	26	0	0.009990	1000.0
41	42	0	0.072898	81	0	75.000000	NaN
78	79	0	0.335559	69	0	0.270449	15000.0
97	98	0	0.010531	57	0	0.280665	5714.0
224	225	0	0.455951	63	1	0.701766	6001.0
244	245	0	0.000000	70	0	0.000000	10000.0

- Crear variables intermedias

```
[4] #@title Example form fields
#@markdown Forms support many types of fields.

no_type_checking = '' #@param
string_type = 'example' #@param {type: "string"}
slider_value = 142 #@param {type: "slider", min: 100, max: 200}
number = 102 #@param {type: "number"}
date = '2010-11-05' #@param {type: "date"}
pick_me = "monday" #@param ["monday", "tuesday", "wednesday",
select_or_input = "apples" #@param ["apples", "bananas", "oranges"]
#@markdown ---
```

Example form fields


Forms support many types of fields.

no_type_checking:

string_type:

slider_value: 142

number:

date: / / 

pick_me: ▼

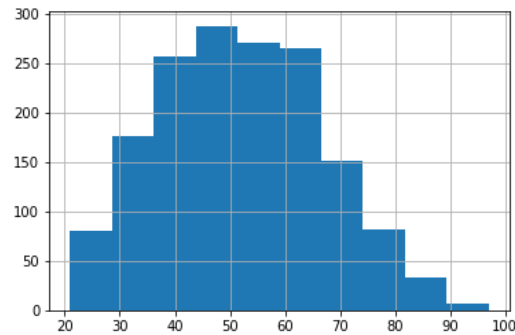
select_or_input: ▼

- Creación de Reportes

Reporte por año

```
[ ] # report for age
df['age'].hist()
```

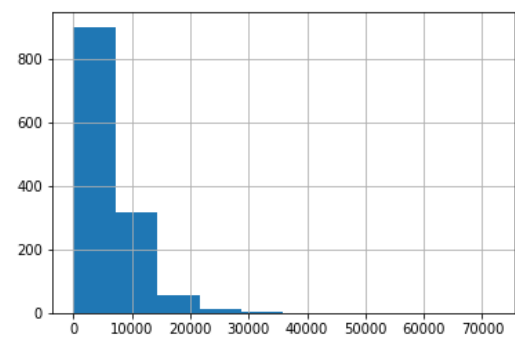
<matplotlib.axes._subplots.AxesSubplot at 0x7f3b57a97c88>



Reporte por mes

```
[ ] # report for MonthlyIncome
df['MonthlyIncome'].hist()
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f3b57a0d9b0>



2. Conclusiones

- Como conclusión tenemos que Colab es muy convenientes para principiantes que quieran experimentar con machine learning y sin incurrir en costos de procesamiento cloud. Además, el ambiente de trabajo ya viene con muchas librerías instaladas listas para utilizar.