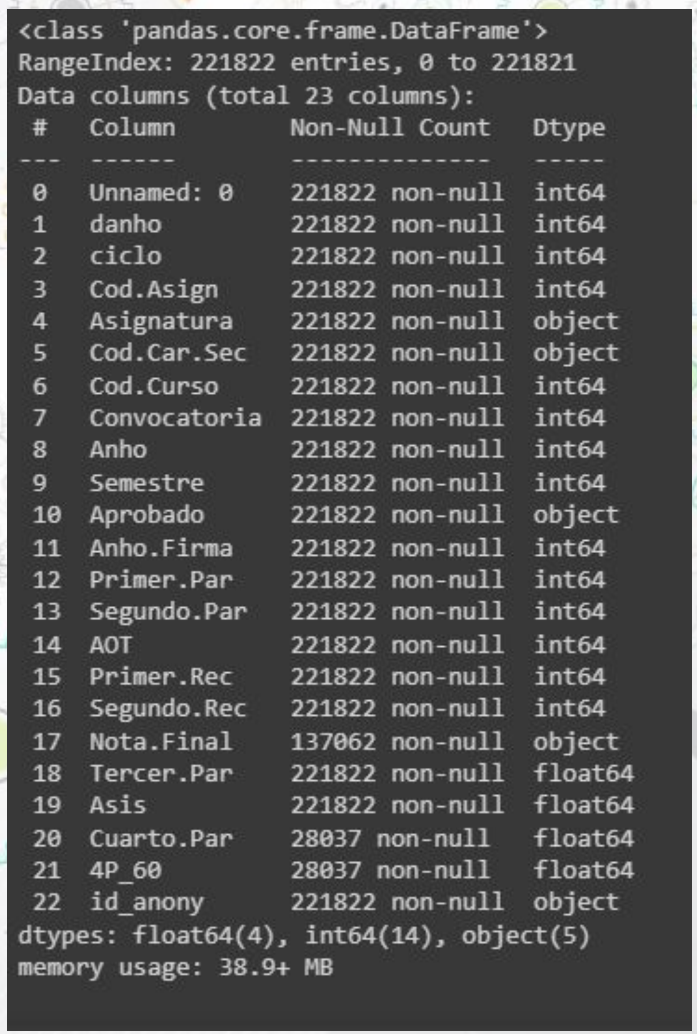
IA-2022

Grupo 4

Héctor Honge

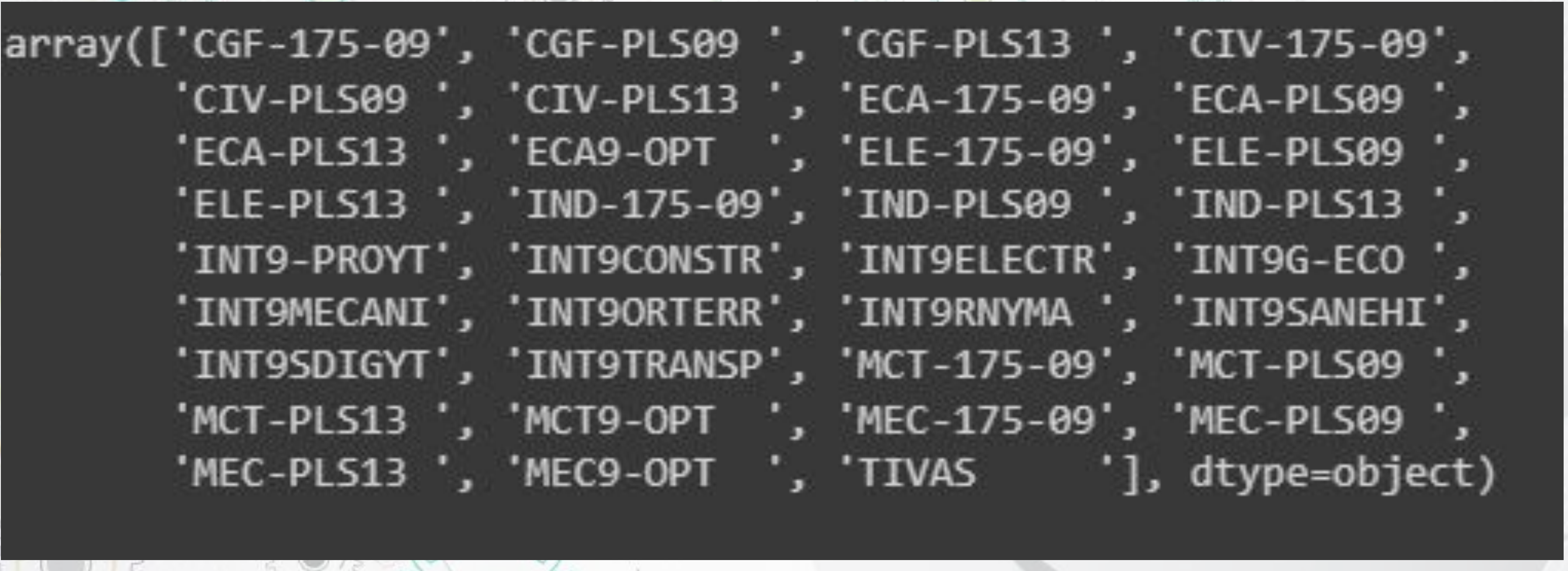
Ignacio Noguera

Análisis Académico-FIUNA

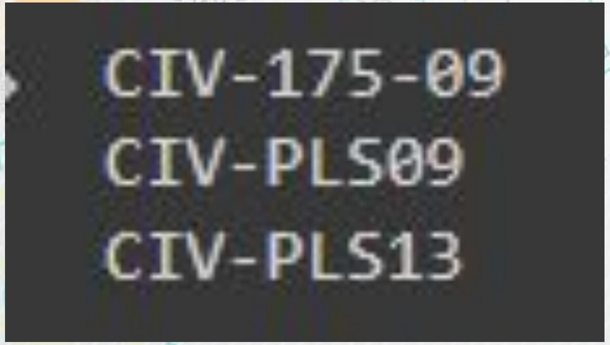


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 221822 entries, 0 to 221821
Data columns (total 23 columns):
 #   Column        Non-Null Count   Dtype  
---  ------        --------------   -----  
 0   Unnamed: 0    221822 non-null  int64  
 1   danho         221822 non-null  int64  
 2   ciclo         221822 non-null  int64  
 3   Cod.Asign     221822 non-null  int64  
 4   Asignatura    221822 non-null  object 
 5   Cod.Car.Sec   221822 non-null  object 
 6   Cod.Curso     221822 non-null  int64  
 7   Convocatoria  221822 non-null  int64  
 8   Anho          221822 non-null  int64  
 9   Semestre      221822 non-null  int64  
 10  Aprobado      221822 non-null  object 
 11  Anho.Firma    221822 non-null  int64  
 12  Primer.Par    221822 non-null  int64  
 13  Segundo.Par   221822 non-null  int64  
 14  AOT           221822 non-null  int64  
 15  Primer.Rec    221822 non-null  int64  
 16  Segundo.Rec   221822 non-null  int64  
 17  Nota.Final    137062 non-null  object 
 18  Tercer.Par    221822 non-null  float64
 19  Asis          221822 non-null  float64
 20  Cuarto.Par    28037 non-null   float64
 21  4P_60         28037 non-null   float64
 22  id_anony      221822 non-null  object 
dtypes: float64(4), int64(14), object(5)
memory usage: 38.9+ MB
```

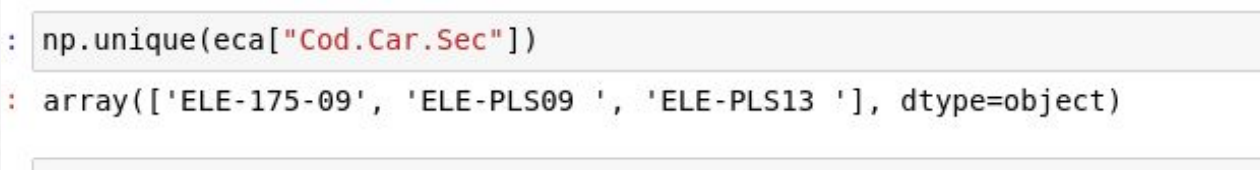
Regresión Lineal



```
array(['CGF-175-09', 'CGF-PLS09 ', 'CGF-PLS13 ', 'CIV-175-09',
       'CIV-PLS09 ', 'CIV-PLS13 ', 'ECA-175-09', 'ECA-PLS09 ',
       'ECA-PLS13 ', 'ECA9-OPT  ', 'ELE-175-09', 'ELE-PLS09 ',
       'ELE-PLS13 ', 'IND-175-09', 'IND-PLS09 ', 'IND-PLS13 ',
       'INT9-PROYT', 'INT9CONSTR', 'INT9ELECTR', 'INT9G-ECO ',
       'INT9MECANI', 'INT9ORTERR', 'INT9RNYMA ', 'INT9SANEHI',
       'INT9SDIGYT', 'INT9TRANSP', 'MCT-175-09', 'MCT-PLS09 ',
       'MCT-PLS13 ', 'MCT9-OPT  ', 'MEC-175-09', 'MEC-PLS09 ',
       'MEC-PLS13 ', 'MEC9-OPT  ', 'TIVAS     '], dtype=object)
```

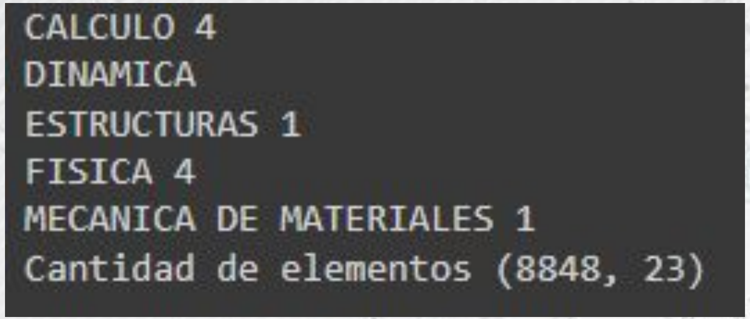


```
CIV-175-09
CIV-PLS09
CIV-PLS13
```

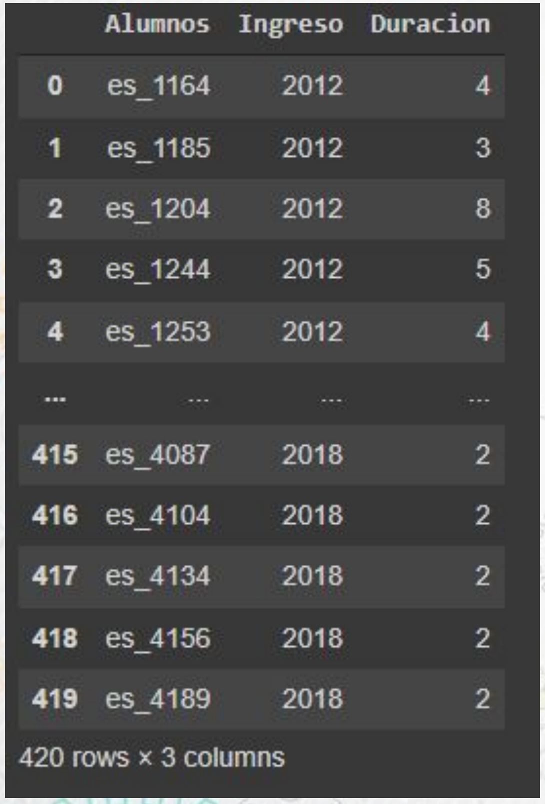


```
: np.unique(eca["Cod.Car.Sec"])

: array(['ELE-175-09', 'ELE-PLS09 ', 'ELE-PLS13 '], dtype=object)
```

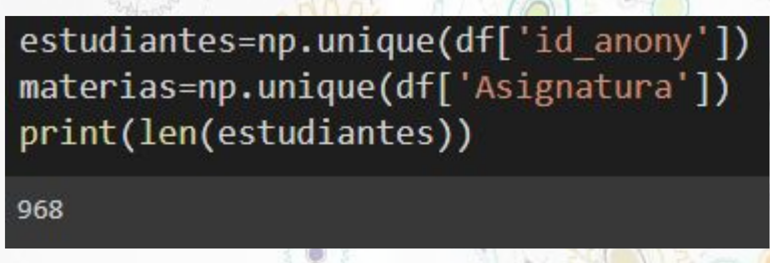


```
CALCULO 4
DINAMICA
ESTRUCTURAS 1
FISICA 4
MECANICA DE MATERIALES 1
Cantidad de elementos (8848, 23)
```



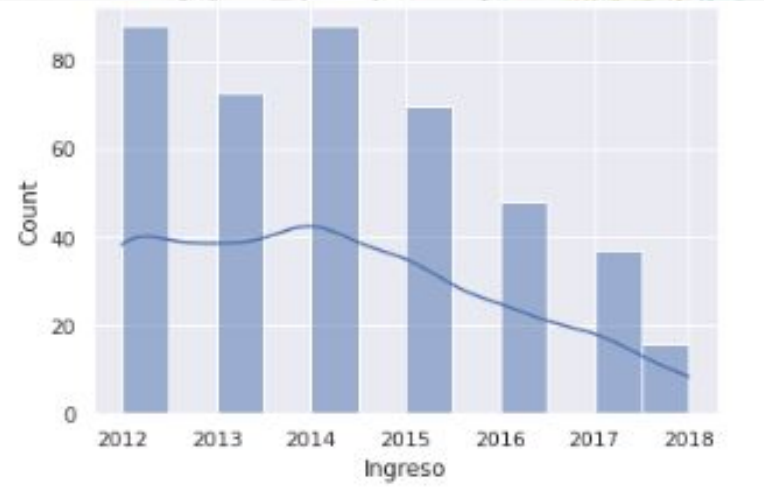
	Alumnos	Ingreso	Duracion
0	es_1164	2012	4
1	es_1185	2012	3
2	es_1204	2012	8
3	es_1244	2012	5
4	es_1253	2012	4
...	...	...	...
415	es_4087	2018	2
416	es_4104	2018	2
417	es_4134	2018	2
418	es_4156	2018	2
419	es_4189	2018	2

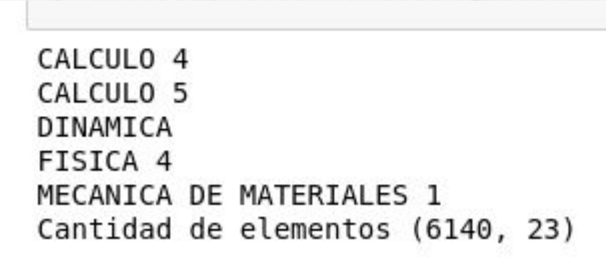
420 rows × 3 columns



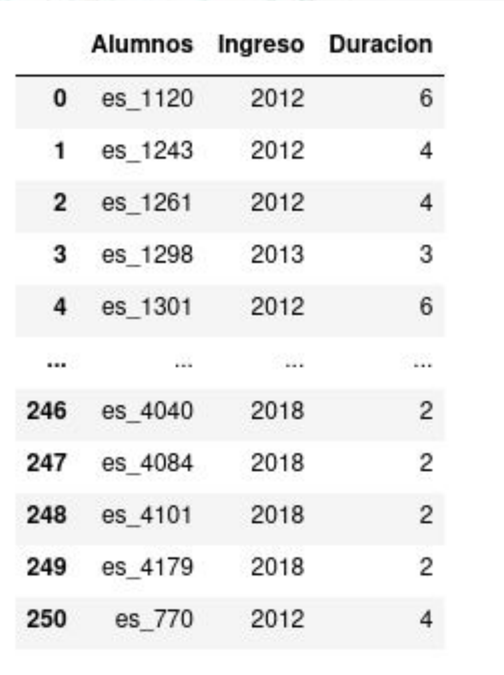
```
estudiantes=np.unique(df['id_anony'])
materias=np.unique(df['Asignatura'])
print(len(estudiantes))
```

```
968
```

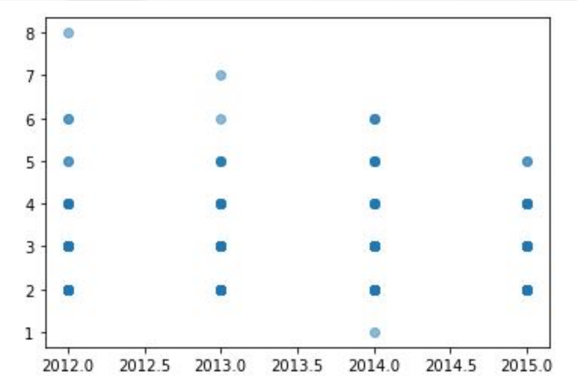


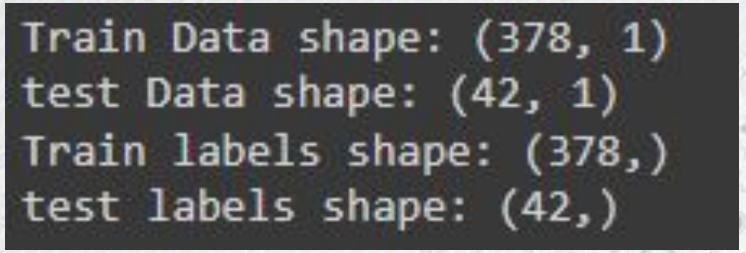


```
CALCULO 4
CALCULO 5
DINAMICA
FISICA 4
MECANICA DE MATERIALES 1
Cantidad de elementos (6140, 23)
```

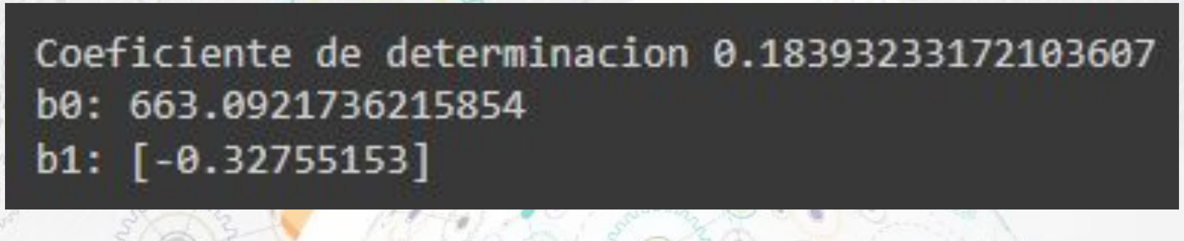


	Alumnos	Ingreso	Duracion
0	es_1120	2012	6
1	es_1243	2012	4
2	es_1261	2012	4
3	es_1298	2013	3
4	es_1301	2012	6
...	...	...	...
246	es_4040	2018	2
247	es_4084	2018	2
248	es_4101	2018	2
249	es_4179	2018	2
250	es_770	2012	4

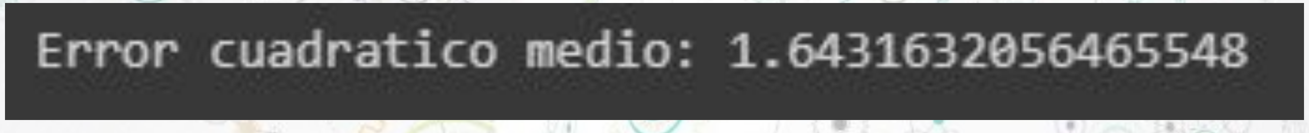




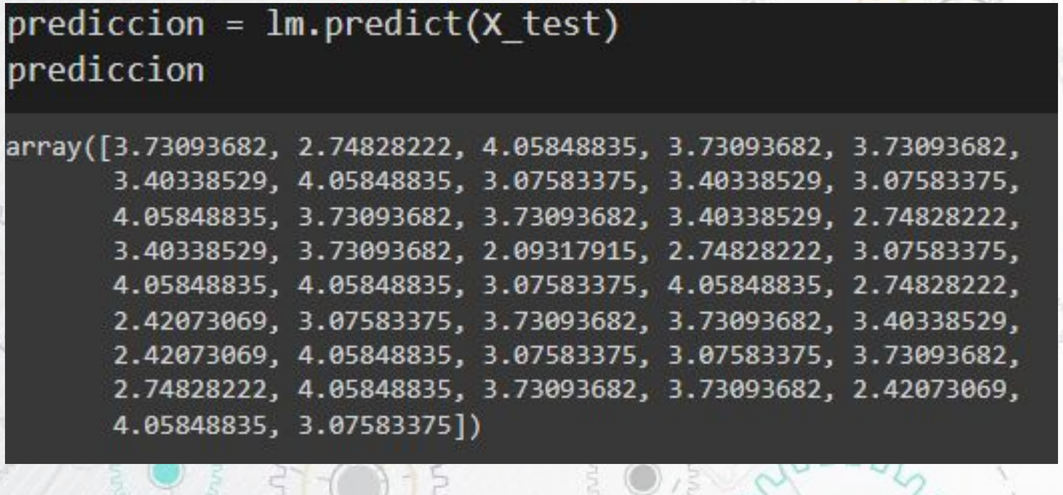
```
Train Data shape: (378, 1)
test Data shape: (42, 1)
Train labels shape: (378,)
test labels shape: (42,)
```



```
Coeficiente de determinacion 0.18393233172103607
b0: 663.0921736215854
b1: [-0.32755153]
```



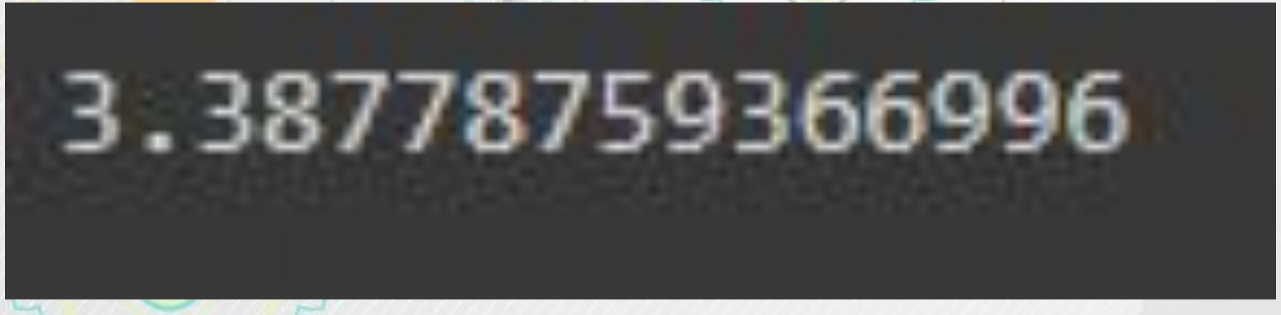
```
Error cuadratico medio: 1.6431632056465548
```



```
prediccion = lm.predict(X_test)
prediccion
```

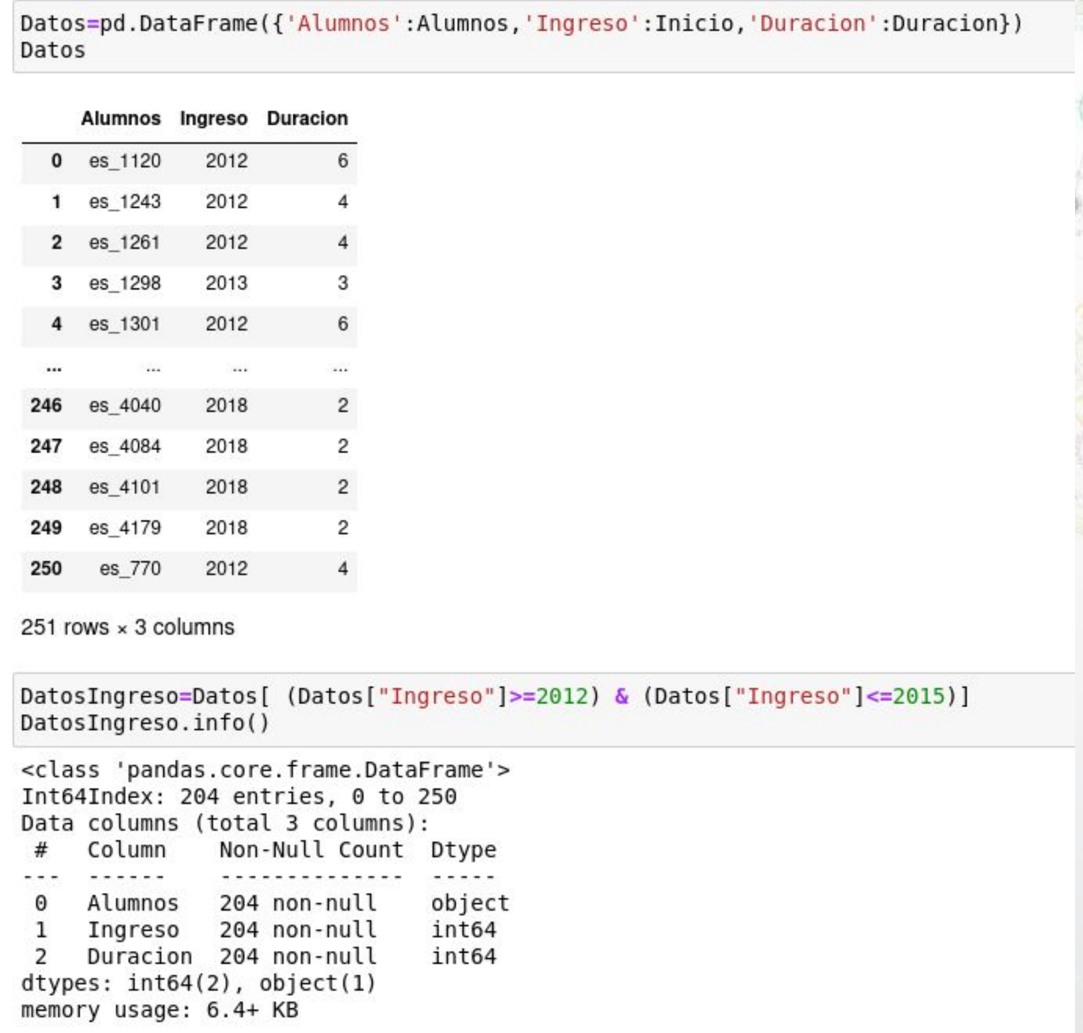
```
array([3.73093682, 2.74828222, 4.05848835, 3.73093682, 3.73093682,
       3.40338529, 4.05848835, 3.07583375, 3.40338529, 3.07583375,
       4.05848835, 3.73093682, 3.73093682, 3.40338529, 2.74828222,
       3.40338529, 3.73093682, 2.09317915, 2.74828222, 3.07583375,
       4.05848835, 4.05848835, 3.07583375, 4.05848835, 2.74828222,
       2.42073069, 3.07583375, 3.73093682, 3.73093682, 3.40338529,
       2.42073069, 4.05848835, 3.07583375, 3.07583375, 3.73093682,
       2.74828222, 4.05848835, 3.73093682, 3.73093682, 2.42073069,
       4.05848835, 3.07583375])
```

Cuanto tarda en promedio un alumno en aprobar cursos básicos?



```
3.387787593669965
```

Regresión Lineal (Electromecánica)



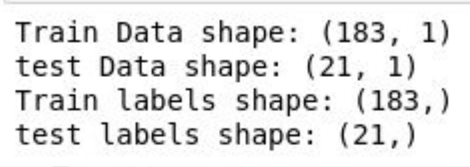
```
Datos=pd.DataFrame({'Alumnos':Alumnos,'Ingreso':Inicio,'Duracion':Duracion})
Datos
```

	Alumnos	Ingreso	Duracion
0	es_1120	2012	6
1	es_1243	2012	4
2	es_1261	2012	4
3	es_1298	2013	3
4	es_1301	2012	6
...	...	...	...
246	es_4040	2018	2
247	es_4084	2018	2
248	es_4101	2018	2
249	es_4179	2018	2
250	es_770	2012	4

251 rows × 3 columns

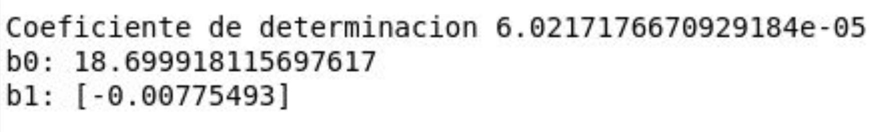
```
DatosIngreso=Datos[ (Datos["Ingreso"]>=2012) & (Datos["Ingreso"]<=2015)]
DatosIngreso.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 204 entries, 0 to 250
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Alumnos   204 non-null    object
 1   Ingreso   204 non-null    int64 
 2   Duracion  204 non-null    int64 
dtypes: int64(2), object(1)
memory usage: 6.4+ KB
```

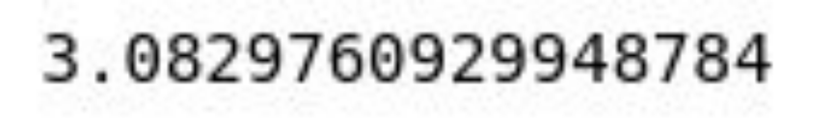


```
Train Data shape: (183, 1)
test Data shape: (21, 1)
Train labels shape: (183,)
test labels shape: (21,)
```

Con 10% de test

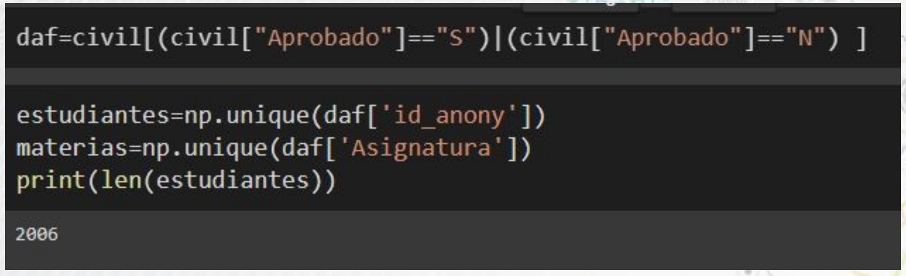


```
Coeficiente de determinacion 6.0217176670929184e-05
b0: 18.699918115697617
b1: [-0.00775493]
```



```
3.0829760929948784
```

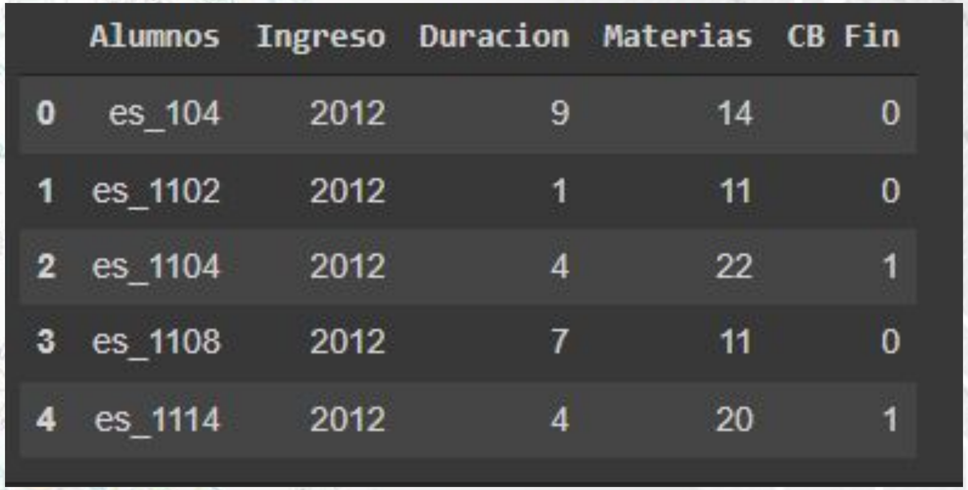
Regresión Logística (CIVIL)



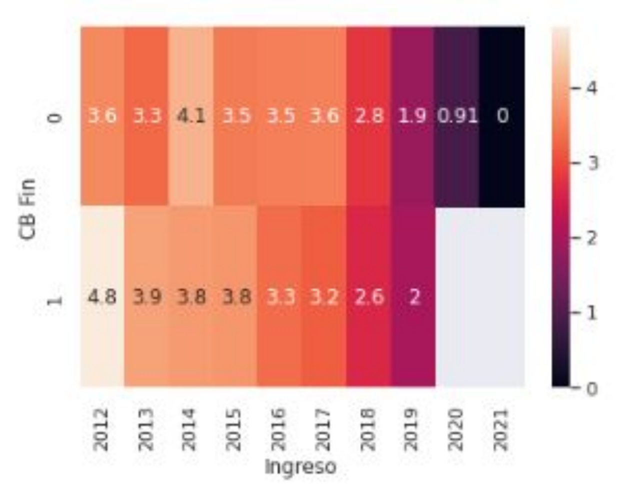
```
daf=civil[(civil["Aprobado"]=="S")|(civil["Aprobado"]=="N") ]
```

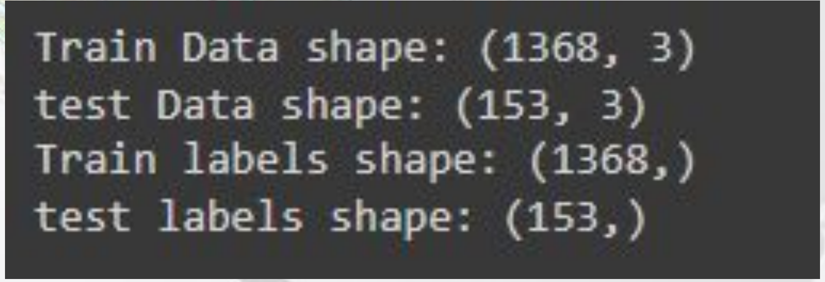
```
estudiantes=np.unique(daf['id_anony'])
materias=np.unique(daf['Asignatura'])
print(len(estudiantes))
```

```
2006
```



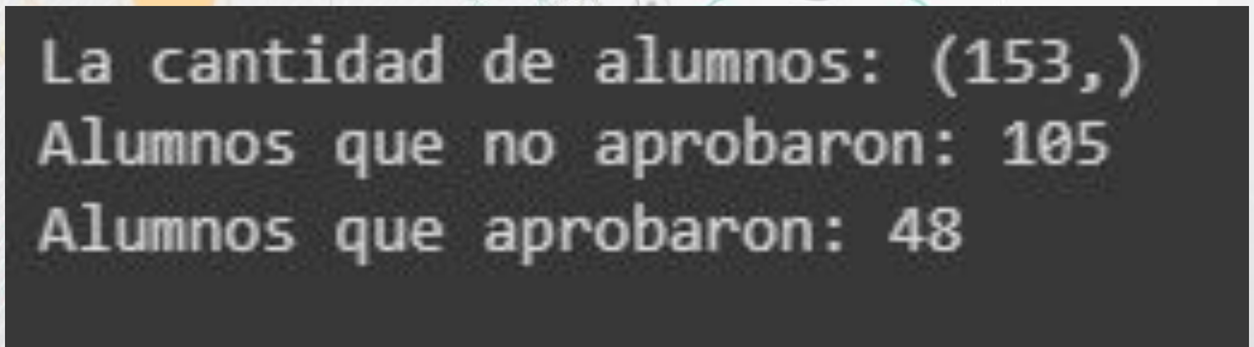
	Alumnos	Ingreso	Duracion	Materias	CB Fin
0	es_104	2012	9	14	0
1	es_1102	2012	1	11	0
2	es_1104	2012	4	22	1
3	es_1108	2012	7	11	0
4	es_1114	2012	4	20	1





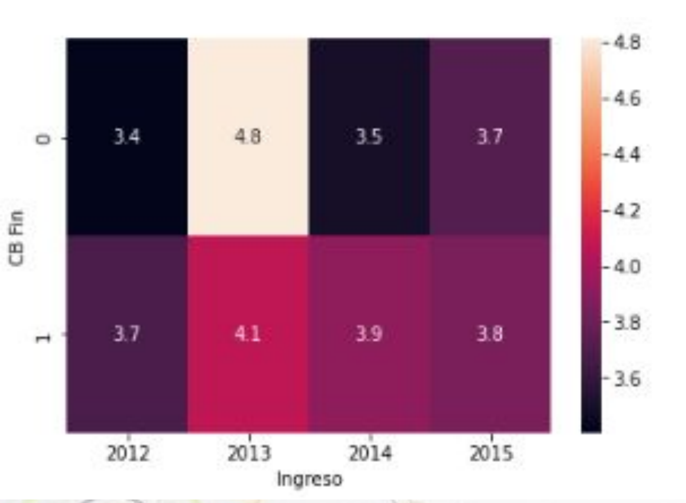
```
Train Data shape: (1368, 3)
test Data shape: (153, 3)
Train labels shape: (1368,)
test labels shape: (153,)
```

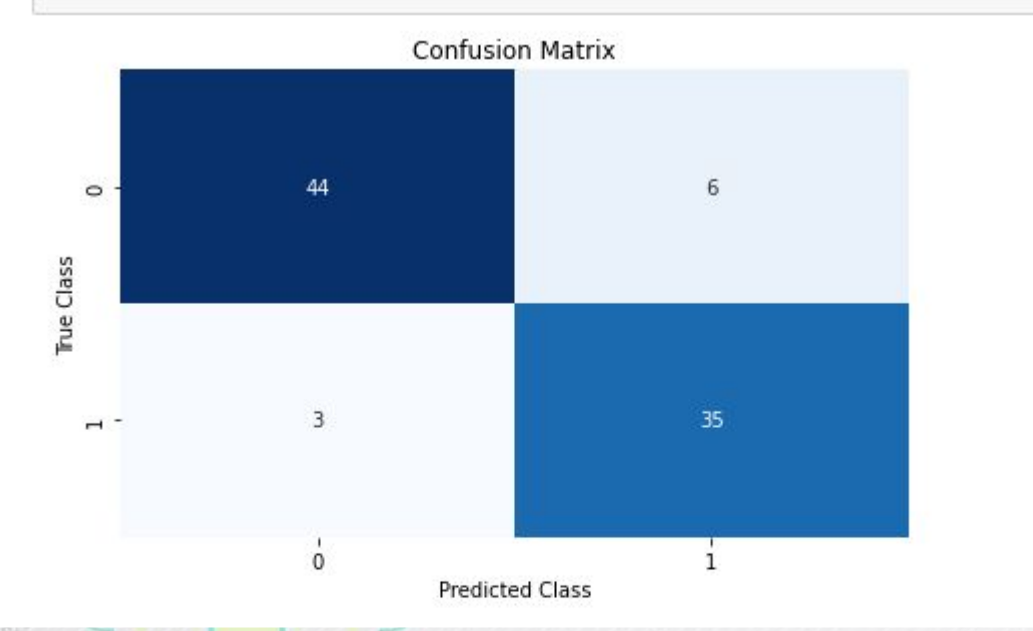
En un lapso de 8 años de 153 alumnos cuantos aprobaron Cursos Basicos

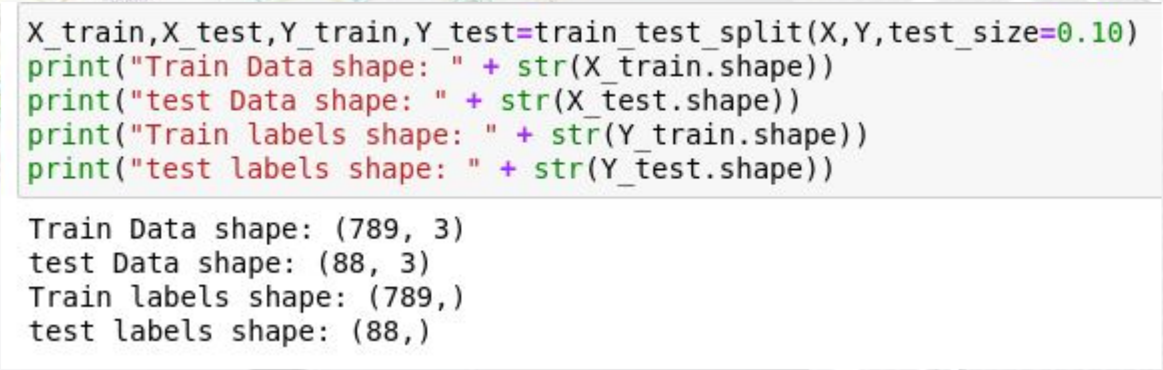


```
La cantidad de alumnos: (153,)
Alumnos que no aprobaron: 105
Alumnos que aprobaron: 48
```

Regresión Logística (Electromecánica)

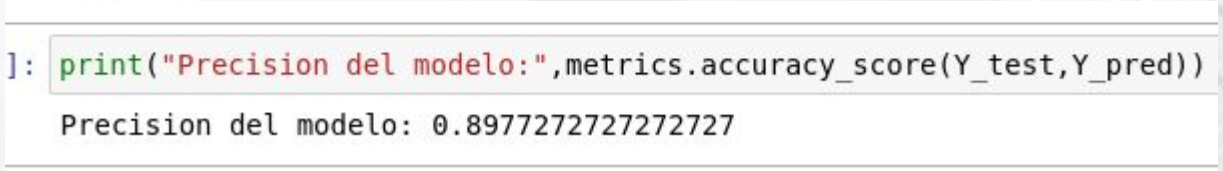






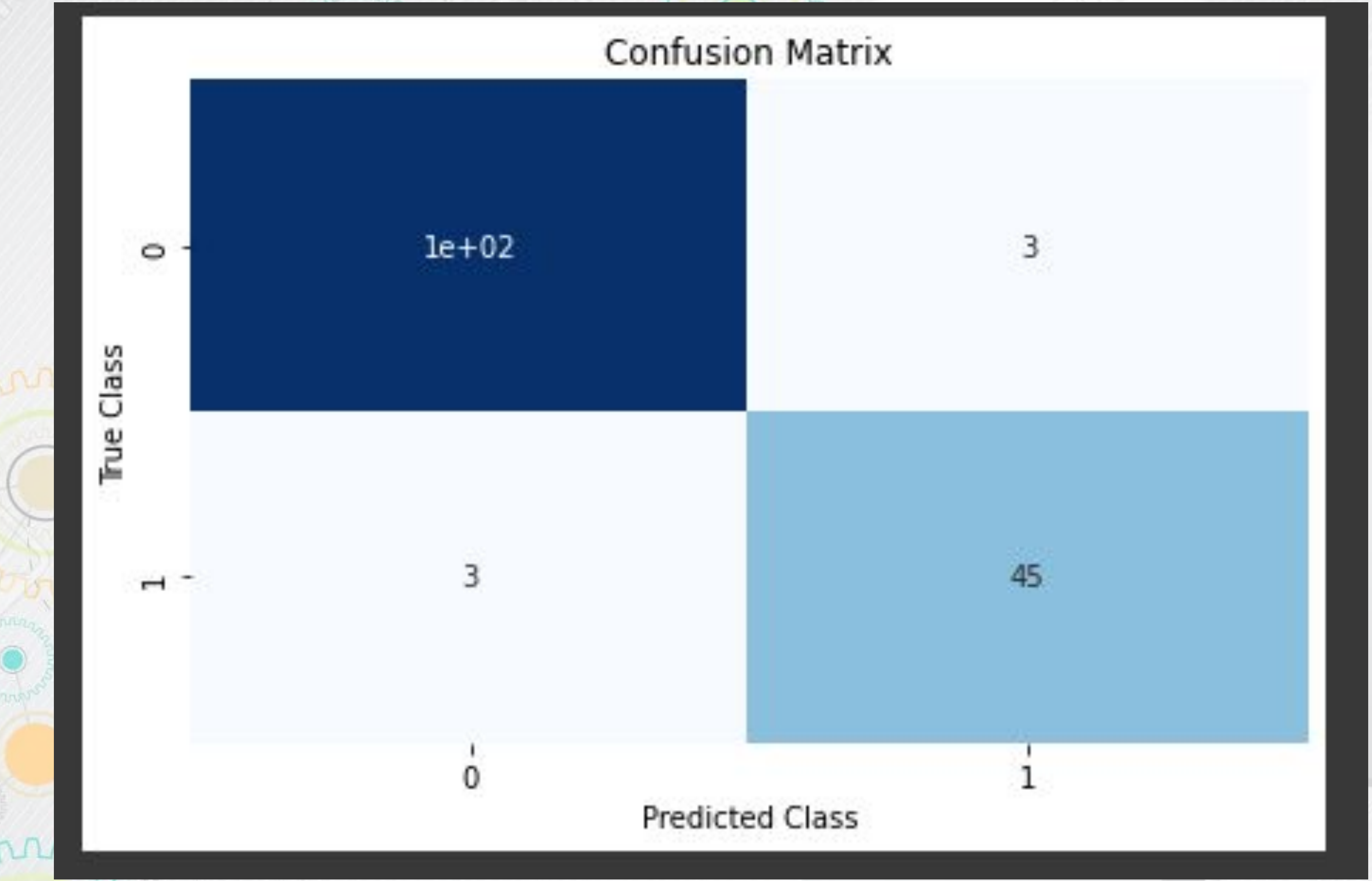
```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.10)
print("Train Data shape: " + str(X_train.shape))
print("test Data shape: " + str(X_test.shape))
print("Train labels shape: " + str(Y_train.shape))
print("test labels shape: " + str(Y_test.shape))
```

```
Train Data shape: (789, 3)
test Data shape: (88, 3)
Train labels shape: (789,)
test labels shape: (88,)
```



```
]: print("Precision del modelo:",metrics.accuracy_score(Y_test,Y_pred))
```

```
Precision del modelo: 0.8977272727272727
```


Confusion Matrix
True Class
0
1
1e+02
3
3
45
0
1
Predicted Class