Regresion logistica

September 17, 2021

0.0.1 Aplicación de Regresión Logística Multinomial

[2]:
```
import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
%matplotlib inline
plt.rc("font", size=14)
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold, KFold
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, f1_score,confusion_matrix,␣
 ↪mean_squared_error, mean_absolute_error, classification_report,␣
 ↪roc_auc_score, roc_curve, precision_score, recall_score
from sklearn.model_selection import cross_val_score
```

[80]:
```
df=pd.read_csv('persona_hogares_nuevo.csv', sep=',')
df.head()
#data.columns
#data=data.dropna()
#print(data.shape)
```

```
/opt/anaconda/lib/python3.8/site-packages/IPython/core/interactiveshell.py:3146:
DtypeWarning: Columns (96,97,98,99,170,171,175,176,178,179) have mixed
types.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

[80]:

	folio	depto	area	nro	genero	edad	dianac	\
0	111-00416110273-A-0021	Chuquisaca	Urbana	1	1.Hombre	42	10	
1	111-00416110273-A-0031	Chuquisaca	Urbana	1	1.Hombre	44	20	
2	151-03374505336-D-0091	Chuquisaca	Rural	6	1.Hombre	4	6	
3	111-00416110273-A-0051	Chuquisaca	Urbana	1	1.Hombre	41	23	
4	111-00416110273-A-0051	Chuquisaca	Urbana	2	2.Mujer	31	30	

```
   mesnac  anionac         relacionjefehogar  …              yhog  \
0       2     1977   1.JEFE O JEFA DEL HOGAR  …              3350
1       5     1975   1.JEFE O JEFA DEL HOGAR  …              3590
2       1     2015     3.HIJO/A O ENTENADO/A  …  958.333374023438
3      11     1978   1.JEFE O JEFA DEL HOGAR  …  3511.39990234375
4       8     1988  2.ESPOSA/O O CONVIVIENTE  …  3511.39990234375

             yhogpc            z        zext  pcero               puno  \
0  558.333312988281  1020.330017  494.549988  Pobre  0.452791452407837
1             897.5  1020.330017  494.549988  Pobre  0.120382636785507
2   119.79167175293   668.099976  381.079987  Pobre  0.820697963237762
3  702.279968261719  1020.330017  494.549988  Pobre  0.311712920665741
4  702.279968261719  1020.330017  494.549988  Pobre  0.311712920665741

                pdos          pextcero            pextuno            pextdos
0  0.205020099878311  No pobre extremo                  0                  0
1  0.014491979032755  No pobre extremo                  0                  0
2  0.673545122146606     Pobre extremo  0.685652136802673  0.470118850469589
3  0.097164943814278  No pobre extremo                  0                  0
4  0.097164943814278  No pobre extremo                  0                  0

[5 rows x 180 columns]
```

[15]:
```
list(df.columns)
```

[15]:
```
['folio',
 'depto',
 'area',
 'nro',
 'genero',
 'edad',
 'dianac',
 'mesnac',
 'anionac',
 'relacionjefehogar',
 'idiomauno',
 'idiomados',
 'idiomanativo',
 'estadocivil',
 'dondehace5anios',
 'pertenecepueblooriginario',
 'pueblooriginario',
 'tieneenfermedad',
 'enfermadodocemeses',
 'acudiodocecaja',
 'acudiodocepublico',
 'acudiodoceprivados',
```

```
'acudiodocemisalud',
'acudiodocedomicilio',
'acudiodocetradicional',
'acudiosinreceta',
'afiliadoseguro',
'dificultadlentes',
'dificultadauditivo',
'dificultadcomunicacion',
'dificultadapoyocaminar',
'dificultadconcentracion',
'dificultadapoyoapoyo',
'dificultadentenderrealidad',
'estuvoembarazada',
'numeroembarazos',
'hijos',
'hijosvivos',
'quienatendioparto',
'dondeatendioparto',
'partoatendiocaja',
'bonoazurduy',
'treintaactividadfisicatrabajo',
'treintamcaminatrabajo',
'ejercicioregular',
'deportepractica',
'ininstalaciontipopublico',
'ininstalaciontipopublicocosto',
'instalacionprivada',
'instalacionabierta',
'instalacioncasa',
'fuma',
'bebe',
'frecuenciabebe',
'sienteseguro',
'victimadocem',
'leeescribe',
'operacionesmatematicas',
'niveleducacionalto',
'gradoalto',
'inscribiocursoanio',
'razoninscribio',
'gradoinscribioanio',
'bonojuancitopinto',
'establecimientomatriculo',
'actualmenteasiste',
'motivonoasiste',
'burlaron',
'insultaron',
```

```
'golpearon',
'amenazaron',
'ignoraron',
'quitaron',
'mintieron',
'tienetelefono',
'usadotelefono3m',
'usadointernet',
'frecuenciauso',
'lugaruso',
'internetbienes',
'internetsalud',
'internetorganizaciones',
'internetcorreo',
'internetcompraventa',
'internettransacciones',
'internetcapacitacion',
'internetbuscempleo',
'internetentretenimiento',
'trabajoultimasemana',
'ulsemanadisponible',
'ulsemanabusconegopropio',
'trabajoanteriormente',
'hacecuantonotrabajo',
'periodohacecuantonotrabja',
'esusted',
'porquenobuscotrabajo',
'ocupacionsemanapasada',
'ocupacionsemanapasadacodigo',
'actividadempresa',
'actividadempresacodigo',
'ocupacion',
'ocupacionrol',
'tiempotrabajaempresa',
'periodotiempotrabajo',
'tipocontrato',
'publicaprivada',
'lugardesempenio',
'numeroempleados',
'diassemanatrabaja',
'horasdiatrabaja',
'salarioliquido',
'salariofrequencia',
'primaultimoanio',
'aguinaldoultimoanio',
'tienevacaciones',
'tieneseguro',
```

```
'ingresoocupacionprincipal',
'frecuenciaocupacionprincipal',
'deseatrabajarmashoras',
'disponibletrabajarmashoras',
'trabajoalgunavez',
'afiliado',
'afiliadoafp',
'aportaafp',
'ingresojubilacion',
'ingresobenemerito',
'ingresoinvalidez',
'ingresoviudez',
'ingresorentadignidad',
'ingresomontorentadignidad',
'ingresointereses',
'ingresoalquileres',
'ingresootrasrentas',
'recibiodineroexterior',
'frecuenciadineroexterior',
'montodineroexterior',
'monedamontoexterior',
'razontrabaja',
'estrato',
'factor',
'tipohogar',
'cobersalud',
'hnvulta',
'quienatenparto',
'dondeatenparto',
'nived',
'nivedg',
'cmasi',
'educprev',
'aestudio',
'cobop',
'caebop',
'pet',
'ocupado',
'cesante',
'aspirante',
'desocupado',
'pea',
'temporal',
'permanente',
'pei',
'condact',
'phrs',
```

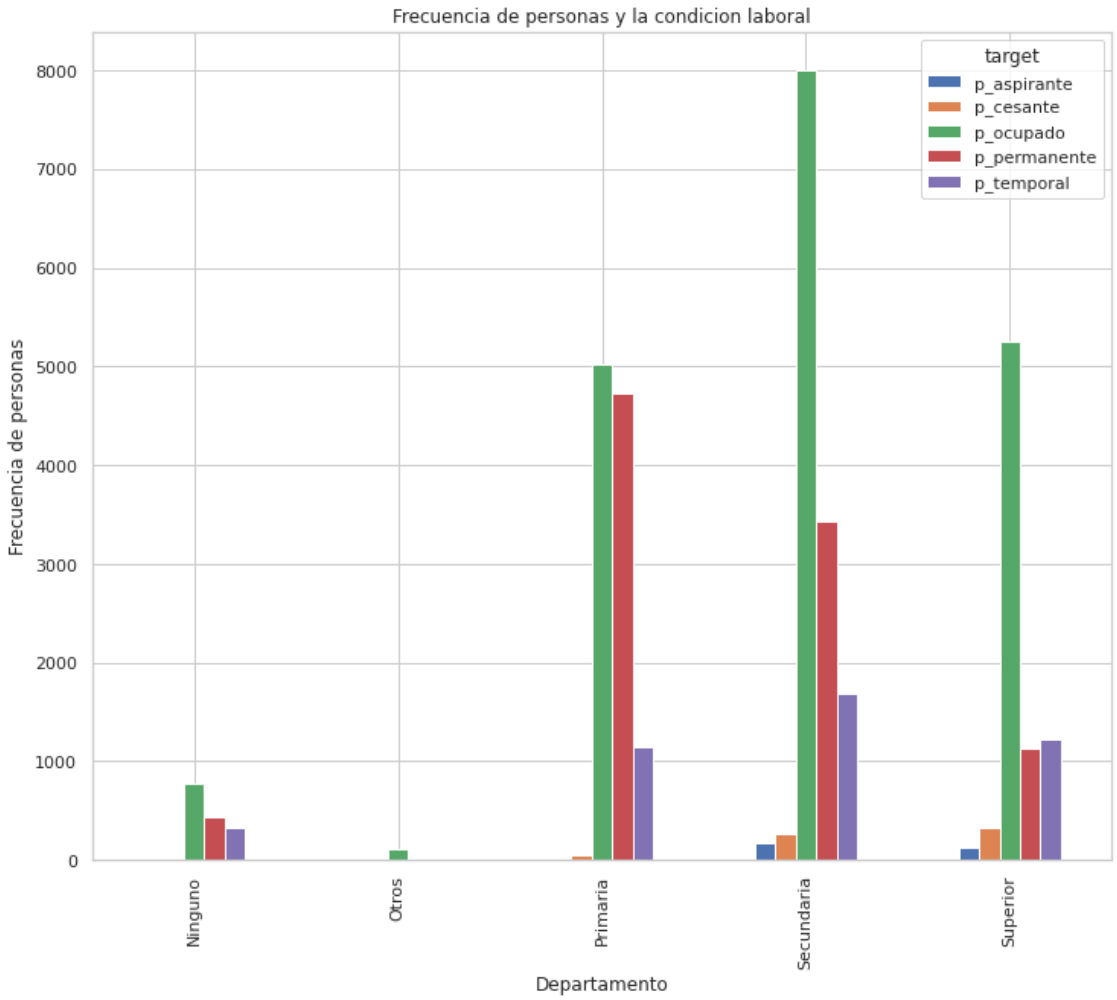
```
'shrs',
'tothrs',
'yprilab',
'yseclab',
'ylab',
'ynolab',
'yper',
'yhog',
'yhogpc',
'z',
'zext',
'pcero',
'puno',
'pdos',
'pextcero',
'pextuno',
'pextdos']
```

[81]:
```
# renombramos la columna condicion laboral
df = df.rename(columns={'condact': 'target'})
```

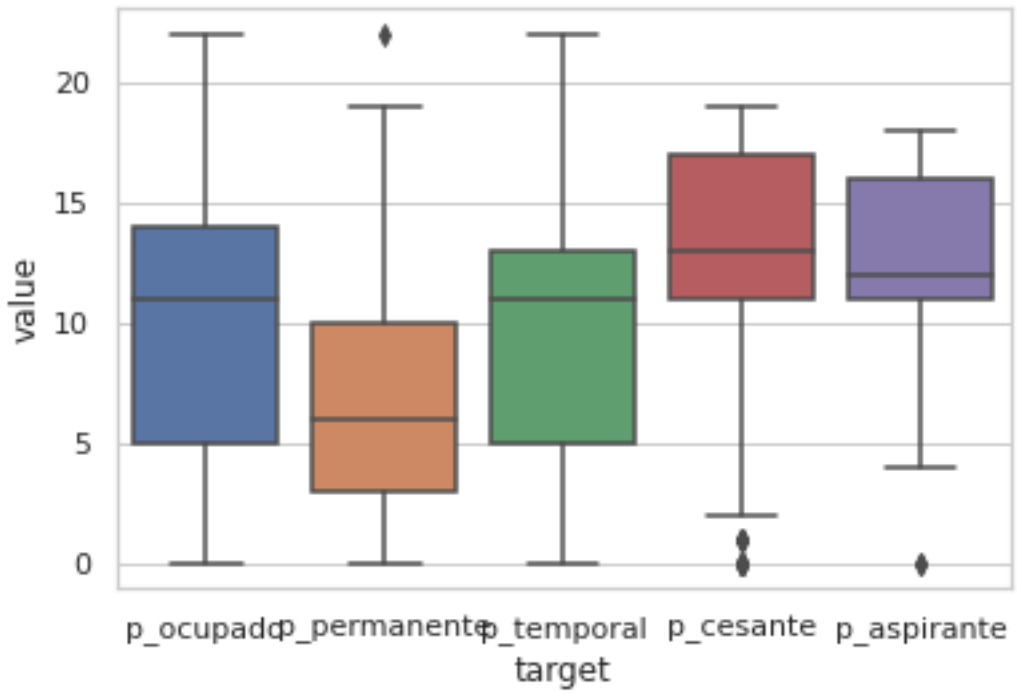
Análisis Exploratorio de Datos Escogiendo nuestra variable dependiente Se desea proyectar la condición laboral de las personas

[82]:
```
pd.crosstab(df.nivedg, df.target).plot(kind='bar', figsize=(12, 10))
plt.title('Frecuencia de personas y la condicion laboral')
plt.xlabel('Departamento')
plt.ylabel('Frecuencia de personas')
#plt.savefig('purchase_fre_job')
```

[82]:
```
Text(0, 0.5, 'Frecuencia de personas')
```

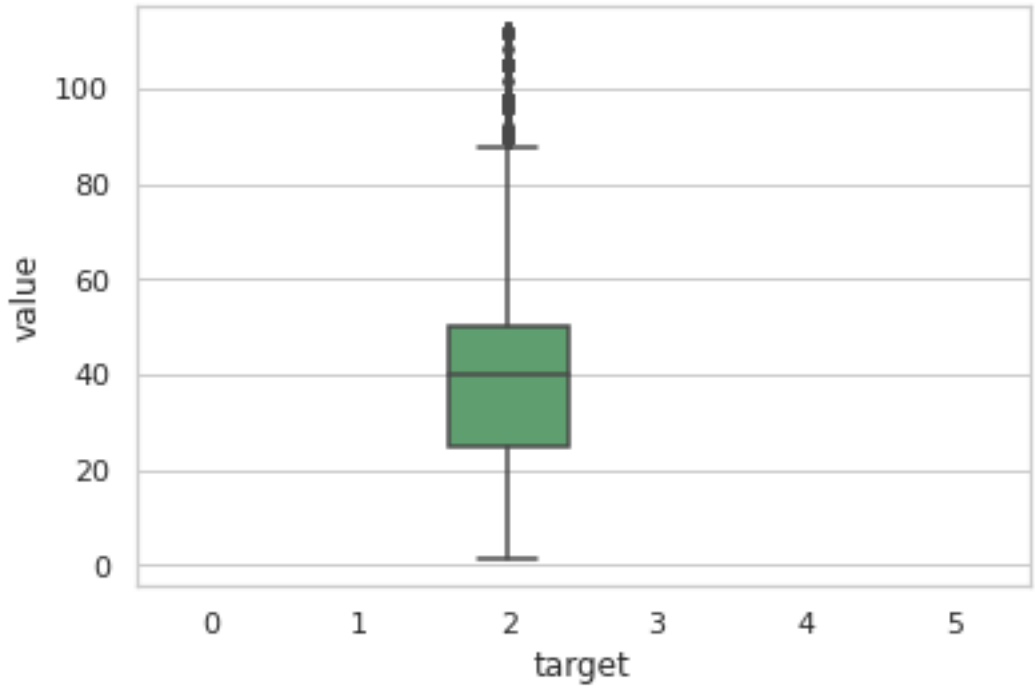


[83]:
```
mdf = pd.melt(df[['aestudio','target']], id_vars=['target'],␣
 ↪var_name=['aestudio'])
ax = sns.boxplot(x="target", y="value", data=mdf)
plt.show()
```



[ ]:

[113]:
```
mdf = pd.melt(df[['phrs','target']], id_vars=['target'], var_name=['phrs'])
ax = sns.boxplot(x="target", y="value", data=mdf)
plt.show()
```



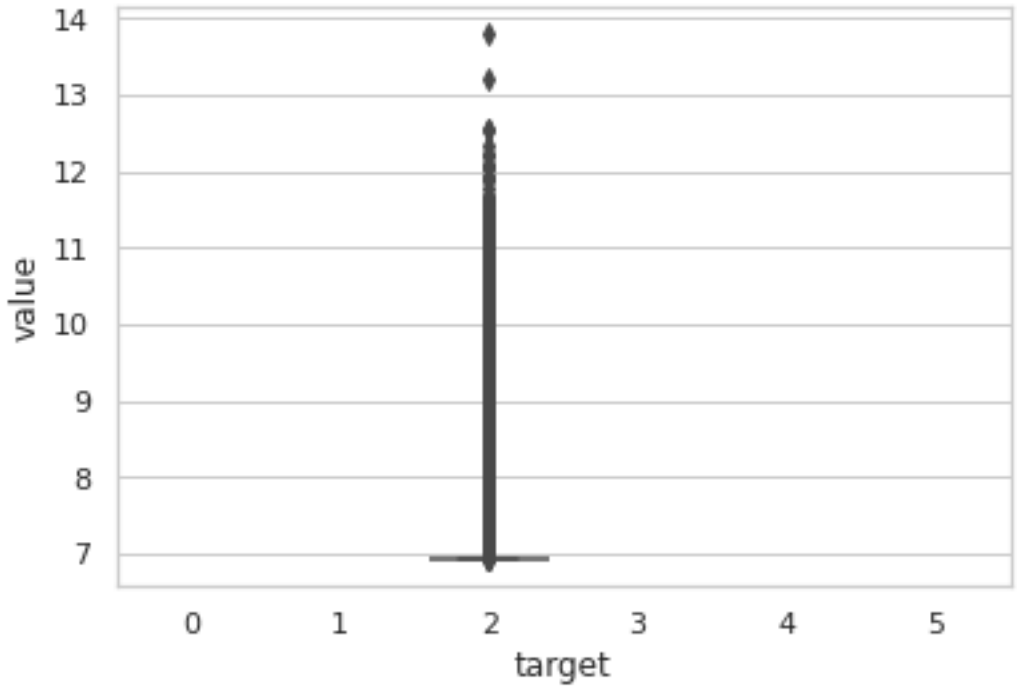
```
[ ]:
```

```
[ ]: La variable horas trabajadas no ayuda porque solo se tiene para el personal␣
     ↪ocupado
```

```
[123]: mdf = pd.melt(df[['ingresoocupacionprincipal_log','target']],␣
        ↪id_vars=['target'], var_name=['ingresoocupacionprincipal_log'])
       ax = sns.boxplot(x="target", y="value", data=mdf)
       plt.show()
```

```
/home/ivan/.local/lib/python3.8/site-packages/numpy/lib/function_base.py:3961:
RuntimeWarning: invalid value encountered in subtract
  diff_b_a = subtract(b, a)
/home/ivan/.local/lib/python3.8/site-packages/numpy/lib/function_base.py:3961:
RuntimeWarning: invalid value encountered in subtract
  diff_b_a = subtract(b, a)
/home/ivan/.local/lib/python3.8/site-packages/numpy/lib/function_base.py:3961:
RuntimeWarning: invalid value encountered in subtract
  diff_b_a = subtract(b, a)
/home/ivan/.local/lib/python3.8/site-packages/numpy/lib/function_base.py:3961:
RuntimeWarning: invalid value encountered in subtract
  diff_b_a = subtract(b, a)
/home/ivan/.local/lib/python3.8/site-packages/numpy/lib/function_base.py:3961:
RuntimeWarning: invalid value encountered in subtract
  diff_b_a = subtract(b, a)
```

```
/home/ivan/.local/lib/python3.8/site-packages/numpy/lib/function_base.py:3961:
RuntimeWarning: invalid value encountered in subtract
  diff_b_a = subtract(b, a)
```

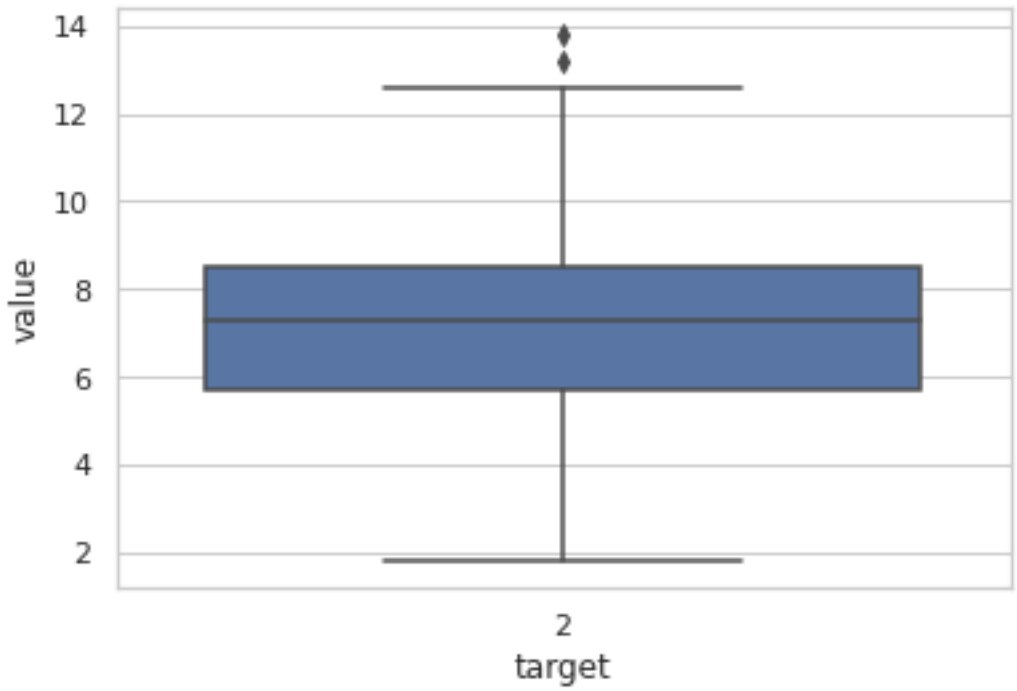


[126]:
```
df['ingresoocupacionprincipal_log'] = np.log(df['ingresoocupacionprincipal'])
```

```
/opt/anaconda/lib/python3.8/site-packages/pandas/core/series.py:726:
RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

[128]:
```
df1 = df.loc[df['ingresoocupacionprincipal'] > 0]
#['ingresoocupacionprincipal']
```

[130]:
```
mdf = pd.melt(df1[['ingresoocupacionprincipal_log','target']],␣
 ↪id_vars=['target'], var_name=['ingresoocupacionprincipal_log'])
ax = sns.boxplot(x="target", y="value", data=mdf)
plt.show()
```



0.0.2 Recategorización

[131]:
```
label_encoder = preprocessing.LabelEncoder()
df['target'] = label_encoder.fit_transform(df['target'])
#df['edad_e'] = label_encoder.fit_transform(df['edad'])
df['genero_e'] = label_encoder.fit_transform(df['genero'])
df['tipohogar_e'] = label_encoder.fit_transform(df['tipohogar'])
df['razontrabaja_e'] = label_encoder.fit_transform(df['razontrabaja'])
df['cobersalud_e'] = label_encoder.fit_transform(df['cobersalud'])
df['hijos_e'] = label_encoder.fit_transform(df['hijos'])
df['ocupacion_e'] = label_encoder.fit_transform(df['ocupacion'])
df['relacionjefehogar_e'] = label_encoder.fit_transform(df['relacionjefehogar'])
#df['interhouse'] = label_encoder.fit_transform(df['internet_casa'])

df.head()
```

[131]:

	folio	depto	area	nro	genero	edad	dianac	\
0	111-00416110273-A-0021	Chuquisaca	Urbana	1	1.Hombre	42	10	
1	111-00416110273-A-0031	Chuquisaca	Urbana	1	1.Hombre	44	20	
2	151-03374505336-D-0091	Chuquisaca	Rural	6	1.Hombre	4	6	
3	111-00416110273-A-0051	Chuquisaca	Urbana	1	1.Hombre	41	23	
4	111-00416110273-A-0051	Chuquisaca	Urbana	2	2.Mujer	31	30	

	mesnac	anionac	relacionjefehogar	…	pextuno	\

```
0       2     1977   1.JEFE O JEFA DEL HOGAR  …                  0
1       5     1975   1.JEFE O JEFA DEL HOGAR  …                  0
2       1     2015     3.HIJO/A O ENTENADO/A  …  0.685652136802673
3      11     1978   1.JEFE O JEFA DEL HOGAR  …                  0
4       8     1988  2.ESPOSA/O O CONVIVIENTE  …                  0

              pextdos genero_e tipohogar_e razontrabaja_e cobersalud_e hijos_e  \
0                   0        0           1              0            0       0
1                   0        0           4              0            3       0
2  0.470118850469589        0           1              0            3       0
3                   0        0           4              0            3       0
4                   0        1           4              0            3       8

  ocupacion_e relacionjefehogar_e ingresoocupacionprincipal_log
0           2                   0                         -inf
1           2                   0                         -inf
2           0                   6                         -inf
3           1                   0                     6.802395
4           2                   5                         -inf

[5 rows x 188 columns]
```

[132]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39605 entries, 0 to 39604
Columns: 188 entries, folio to ingresoocupacionprincipal_log
dtypes: float64(7), int64(16), object(165)
memory usage: 56.8+ MB
```

[136]:
```
nomcol = ['edad','genero_e','hijos_e','tipohogar_e', 'cobersalud_e',␣
 ↪'razontrabaja_e', 'relacionjefehogar_e','ocupacion_e',␣
 ↪'ingresoocupacionprincipal', 'aestudio','target']
df1=df[nomcol]
df1.head()
```

[136]:
```
   edad  genero_e  hijos_e  tipohogar_e  cobersalud_e  razontrabaja_e  \
0    42         0        0            1             0               0
1    44         0        0            4             3               0
2     4         0        0            1             3               0
3    41         0        0            4             3               0
4    31         1        8            4             3               0

   relacionjefehogar_e  ocupacion_e  ingresoocupacionprincipal  aestudio  \
0                    0            2                          0        17
1                    0            2                          0        16
2                    6            0                          0         0
3                    0            1                        900         6
```

```
4                    5            2                             0         4

   target
0       2
1       2
2       5
3       2
4       2
```

[137]:
```
X=df1[df1.columns[:-1]]
y=df1['target']
X.head()
```

[137]:
```
   edad  genero_e  hijos_e  tipohogar_e  cobersalud_e  razontrabaja_e  \
0    42         0        0            1             0               0
1    44         0        0            4             3               0
2     4         0        0            1             3               0
3    41         0        0            4             3               0
4    31         1        8            4             3               0

   relacionjefehogar_e  ocupacion_e  ingresoocupacionprincipal  aestudio
0                    0            2                          0        17
1                    0            2                          0        16
2                    6            0                          0         0
3                    0            1                        900         6
4                    5            2                          0         4
```

[138]:
```
# preparacion de la data de aprendizaje y de testeo
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=25)
```

[139]:
```
reg = LogisticRegression(multi_class='auto', random_state=25, n_jobs=-1)
reg.fit(X_train,y_train)
reg
```

[139]:
```
LogisticRegression(n_jobs=-1, random_state=25)
```

[141]:
```
pred=reg.predict(X_test)
reg_cv=cross_val_score(reg, X_train, y_train, cv=10).mean()
```

[142]:
```
print('Coef de determinación: %.3f' % reg.score(X_test, y_test))
print('Cross-validation $R^2$: %0.3f' % reg_cv)
print('Precision: %.3f' % precision_score(y_test, pred, average='micro'))
print('Proporción de verdaderos positivos : %.3f' % recall_score(y_test, pred,␣
 ↪average='micro'))
print('Contribución de la precision ponderada(F1 score): %.3f' %␣
 ↪f1_score(y_test, pred, average='micro'))
```

```
#print("Precision Score : ",precision_score(y_test, y_pred,
 #                                          pos_label='positive'
  #                                         average='micro'))
```

```
Coef de determinación: 0.910
Cross-validation $R^2$: 0.912
Precision: 0.910
Proporción de verdaderos positivos : 0.910
Contribución de la precision ponderada(F1 score): 0.910
```

[ ]:
```
El modelo predice en todos los casos al 91, es decir de cada 100 personas␣
↪podemos categorizar a 91.
```

[143]:
```
y_pred =reg.predict(X_test)
print('Precición de modelo logistico para clasificar segun la data test: {:.
↪2f}'.format(reg.score(X_test, y_test)))
```

```
Precición de modelo logistico para clasificar segun la data test: 0.91
```

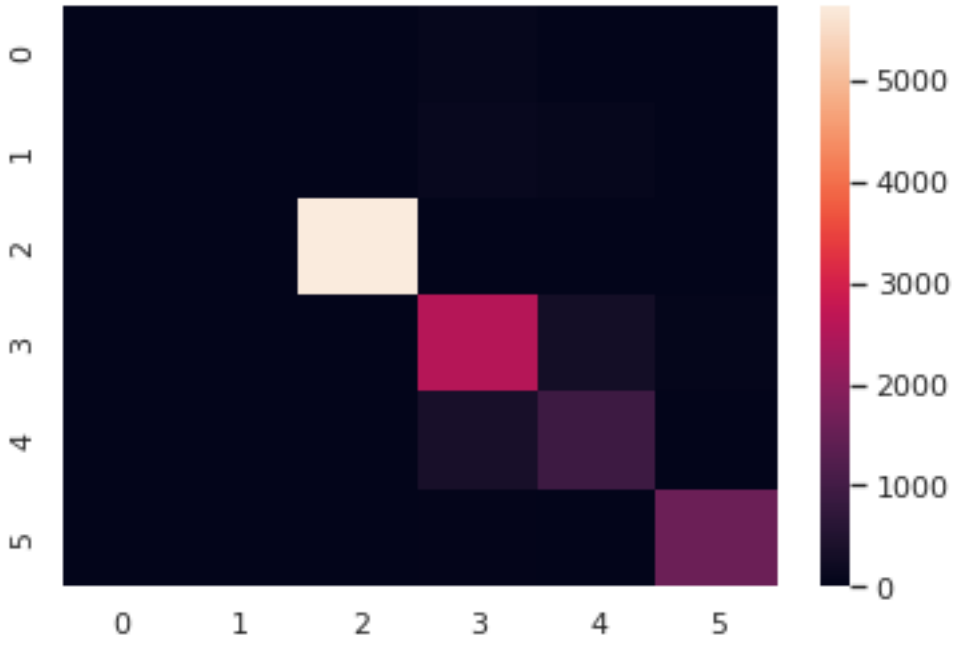
0.0.3 Calculamos la matriz de confusión¶

[144]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)
```

```
[[   0    1    0   89   13    0]
 [   0    7    0  115   79    0]
 [   0    0 5738    1    1    0]
 [   0   11    0 2558  299   54]
 [   0    4    0  362  934    0]
 [   0    0    0   42    0 1574]]
```

[ ]:
```
se puede ver en las filas los datos observados y en las columnas las predichas.
se puede ver en la diagonal principal esta bien clasificado.
```

[156]:
```
import seaborn as sns;
sns.set_theme()
uniform_data = np.random.rand(10, 12)
ax = sns.heatmap(confusion_matrix)
```



[ ]:
```
Se aprecia que los colores oscuros muestran errores muy bajos en cuanto a los␣
↪falsos positivos y verdaderos negativos.
```

[145]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       103
           1       0.30      0.03      0.06       201
           2       1.00      1.00      1.00      5740
           3       0.81      0.88      0.84      2922
           4       0.70      0.72      0.71      1300
           5       0.97      0.97      0.97      1616

    accuracy                           0.91     11882
   macro avg       0.63      0.60      0.60     11882
weighted avg       0.90      0.91      0.90     11882


/home/ivan/.local/lib/python3.8/site-
packages/sklearn/metrics/_classification.py:1245: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ivan/.local/lib/python3.8/site-
```

```
packages/sklearn/metrics/_classification.py:1245: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/ivan/.local/lib/python3.8/site-
packages/sklearn/metrics/_classification.py:1245: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

[ ]:
```
La exactitud llega al 91 %
```

[146]:
```
y_test
```

[146]:
```
38345    4
39078    3
7760     2
11134    2
9191     2
        ..
31843    3
9919     2
15367    2
10844    2
35203    4
Name: target, Length: 11882, dtype: int64
```