naive-bayes

September 17, 2021

[126]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
import numpy as np
from scipy.stats import norm
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import VarianceThreshold
from sklearn.metrics import accuracy_score
from sklearn import preprocessing
from scipy import stats
from pandas import Series, DataFrame
from pandas.plotting import autocorrelation_plot
from pylab import rcParams
from matplotlib import collections as collections
from matplotlib.patches import Rectangle
from itertools import cycle

from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import GaussianNB

import warnings
warnings.filterwarnings('ignore')
%matplotlib inline

rcParams['figure.figsize'] = 5,4
sb.set_style('whitegrid')
from numpy import median
from numpy import mean
```

[93]:
```
df = pd.read_csv('./persona_hogares_nuevo.csv')
```

[94]:
```
df.head()
df.tail()
```

[94]:

	folio	depto	area	nro	genero	edad	dianac	\
39600	953-12108090472-A-0031	Pando	Rural	4	2.Mujer	13	26	
39601	953-12108090472-A-0051	Pando	Rural	3	1.Hombre	7	1	

```
39602  953-12108090472-A-0061  Pando  Rural    2   2.Mujer    22       1
39603  953-12108090472-A-0071  Pando  Rural    4  1.Hombre    17      11
39604  953-12108090472-A-0071  Pando  Rural    5  1.Hombre    18       9

       mesnac  anionac         relacionjefehogar  …     yhog   yhogpc  \
39600       4     2006     3.HIJO/A O ENTENADO/A  …     2165      433
39601      11     2012            9.NIETO/NIETA  …  5564.67  1854.89
39602       9     1997  2.ESPOSA/O O CONVIVIENTE  …     1500      500
39603       9     2002     3.HIJO/A O ENTENADO/A  …     4500      900
39604       9     2001          10.OTRO PARIENTE  …     4500      900

                z        zext     pcero      puno       pdos  \
39600  668.099976  381.079987     Pobre  0.351893   0.123829
39601  668.099976  381.079987  No pobre         0          0
39602  668.099976  381.079987     Pobre  0.251609  0.0633071
39603  668.099976  381.079987  No pobre         0          0
39604  668.099976  381.079987  No pobre         0          0

              pextcero pextuno pextdos
39600  No pobre extremo       0       0
39601  No pobre extremo       0       0
39602  No pobre extremo       0       0
39603  No pobre extremo       0       0
39604  No pobre extremo       0       0

[5 rows x 180 columns]
```

[95]:
```
df.dtypes
```

[95]:
```
folio       object
depto       object
area        object
nro          int64
genero      object
             …
puno        object
pdos        object
pextcero    object
pextuno     object
pextdos     object
Length: 180, dtype: object
```

[96]:
```
# ver las variables del dataset
list(df.columns)
#df.info()
```

[96]: ['folio',
 'depto',
 'area',
 'nro',
 'genero',
 'edad',
 'dianac',
 'mesnac',
 'anionac',
 'relacionjefehogar',
 'idiomauno',
 'idiomados',
 'idiomanativo',
 'estadocivil',
 'dondehace5anios',
 'pertenecepuebloorginario',
 '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']
```

[97]:
```
df.shape
```

[97]:
```
(39605, 180)
```

[ ]:
```
# buscar correlaciones entre la predictora y la target
```

0.0.1 Análisis Exploratorio de Datos

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

[98]:
```
# renombramos la columna condicion laboral
df = df.rename(columns={'condact': 'target'})
```

[99]:
```
df['target'].value_counts()
```

[99]:
```
p_ocupado       19151
p_permanente     9715
p_temporal       4396
p_cesante         656
p_aspirante       318
Name: target, dtype: int64
```

0.0.2 Evaluando variables

[100]:
```
df['tipohogar'].value_counts()
```

[100]:
```
Nuclear completa     22062
Hogar Extendido       6684
Monoparental          5006
Pareja Nuclear        2663
Hogar Unipersonal     2000
Otro                  1093
Hogar Compuesto         97
Name: tipohogar, dtype: int64
```

[101]:
```
df['ingresoocupacionprincipal'].value_counts()
# entra porque uno esta ocupado es pq gana directo
```

[101]:
```
0        31149
3000       296
200        296
150        292
100        276
          …
46           1
39400        1
28800        1
315          1
24500        1
Name: ingresoocupacionprincipal, Length: 792, dtype: int64
```

[102]:
```
df['razontrabaja'].value_counts()
```

[102]:
```
                                                38039
2.Para apoyar al negocio u otra actividad que realiza la familia (complementar
lo ingresos del hogar)                           895
1.Para generar sus ingresos propios
282
4.Para aprender, tener experiencia y habilidades
183
5.Para seguir las costumbres de la familia o la comunidad
181
3.Para superar los problemas temporales de falta de ingresos/exceso de gastos
del hogar (dejarà de trabajar cuando èstos       24
6.Otra razón (Especifique)
1
Name: razontrabaja, dtype: int64
```

[103]:
```
df['cobersalud'].value_counts()
```

[103]:
```
Público    26754
Ninguno    12477
Privado      364
Otro          10
Name: cobersalud, dtype: int64
```

[104]:
```
df['aestudio'].value_counts()
# variable ayuda a que la persona este trabajando
```

[104]:
```
0     6906
12    5909
17    3489
5     2736
3     2024
8     1740
4     1675
10    1651
2     1637
6     1560
11    1465
7     1451
1     1394
9     1346
14    1292
15    1279
13     945
16     655
18     227
19     189
22      28
```

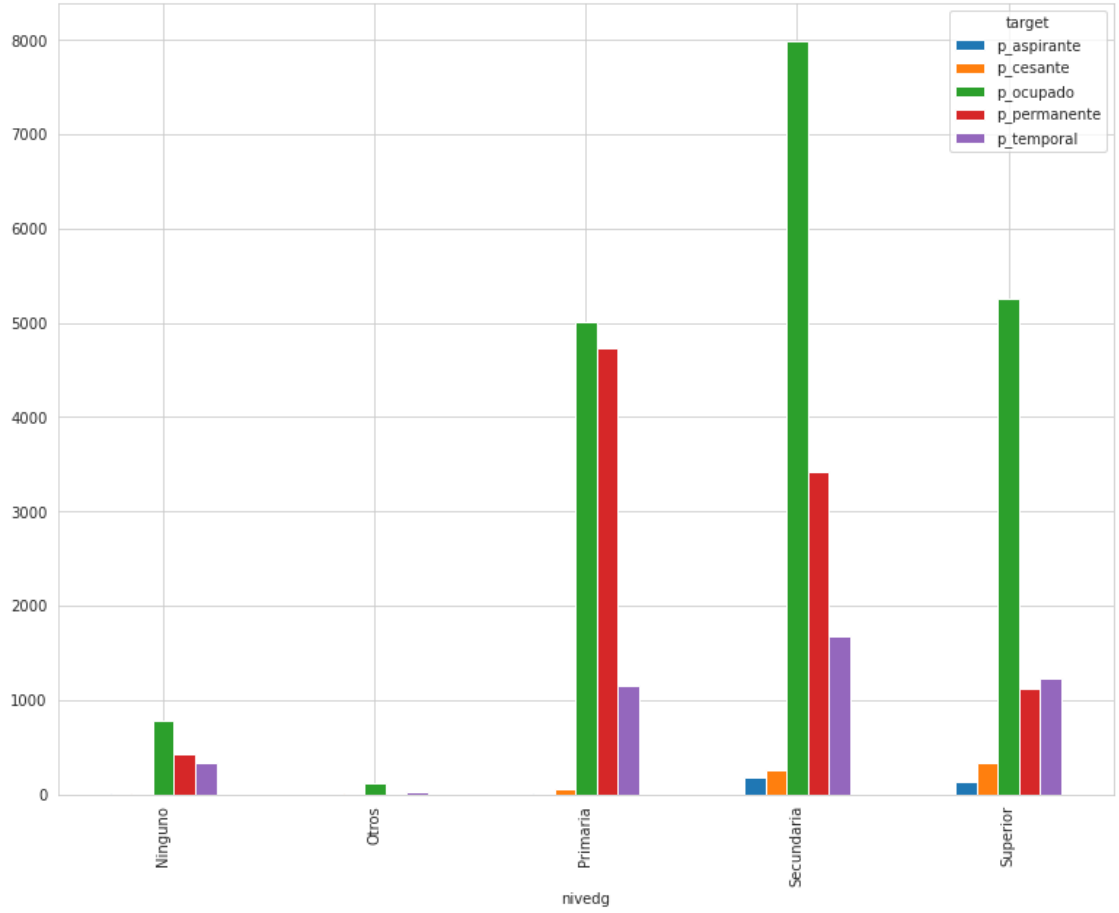
```
21       6
20       1
Name: aestudio, dtype: int64
```

[105]:
```
df['quienatenparto'].value_counts()
```

[105]:
```
                          36152
Atención Institucional     3294
Otro                        125
Atención Partera             29
Médico Tradicional            5
Name: quienatenparto, dtype: int64
```

[110]:
```
pd.crosstab(df.nivedg,df.target).plot(kind='bar', figsize=(13,10))
#plt.title('Frecuencia de personas por nivel de educación máximo')
#plt.xlabel('Nivel de educación máximo')
#plt.ylabel('Frecuencia de personas')
```

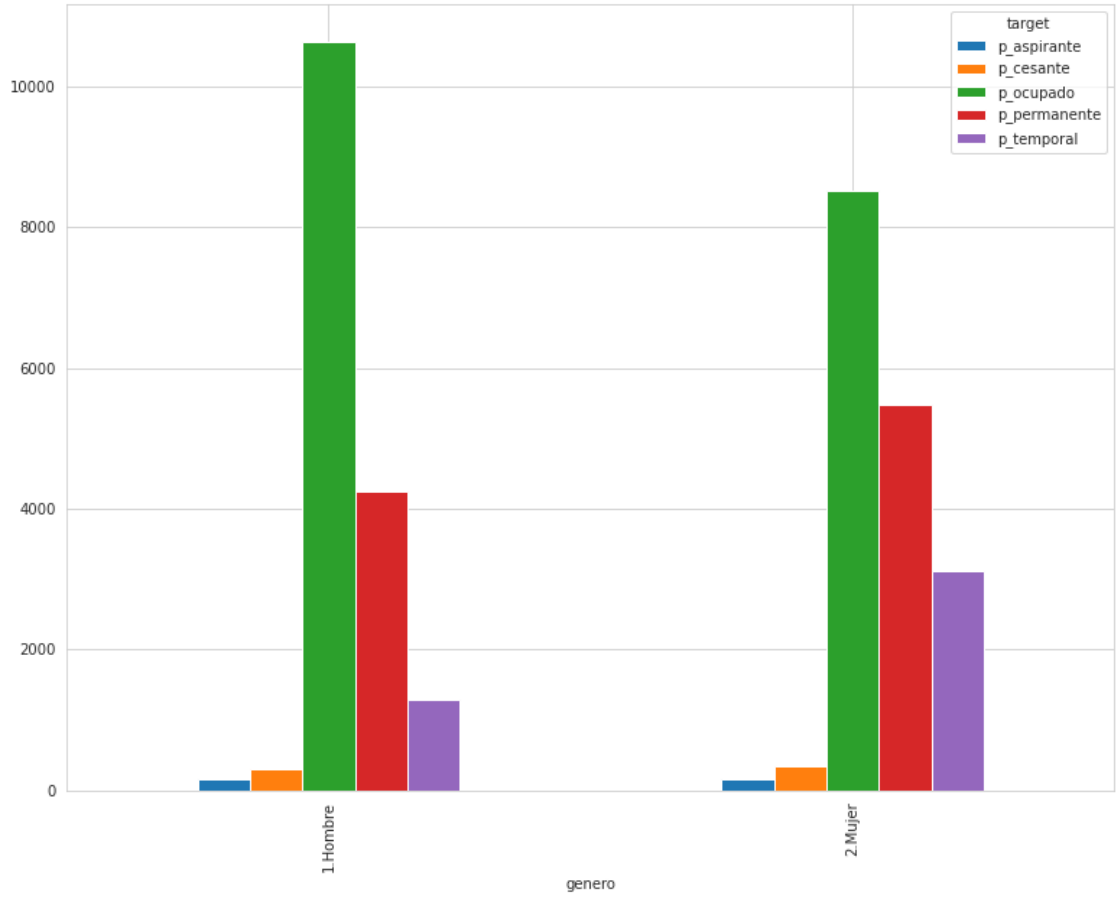
[110]:
```
<AxesSubplot:xlabel='nivedg'>
```



[ ]:
```
#  primaria secundaria superior tienen alta frecuencia con ocupados permanentes␣
 ↪y temporales
```

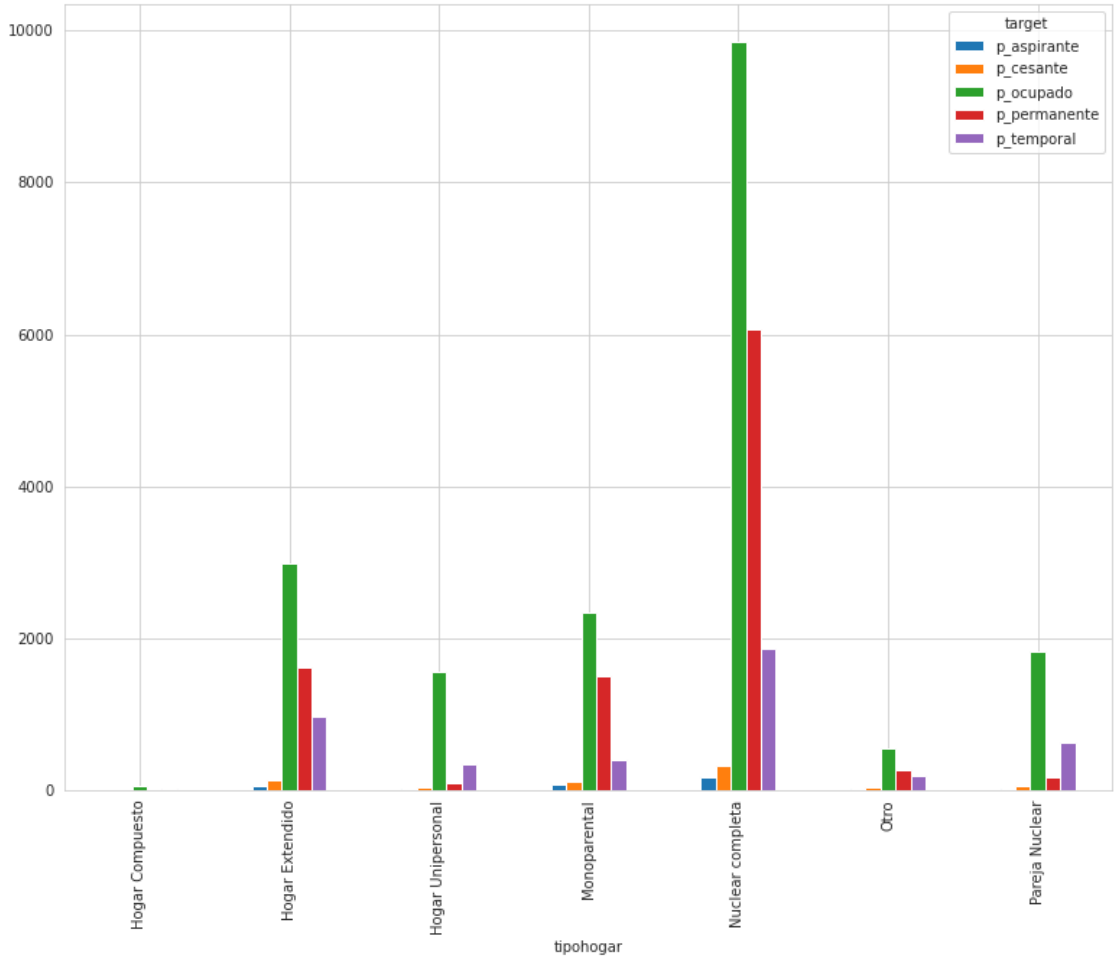
[113]:
```
pd.crosstab(df.genero,df.target).plot(kind='bar', figsize=(13, 10))
#plt.title('Frecuencia de personas por genero')
#plt.xlabel('Genero')
#plt.ylabel('Frecuencia de personas')
```

[113]: <AxesSubplot:xlabel='genero'>



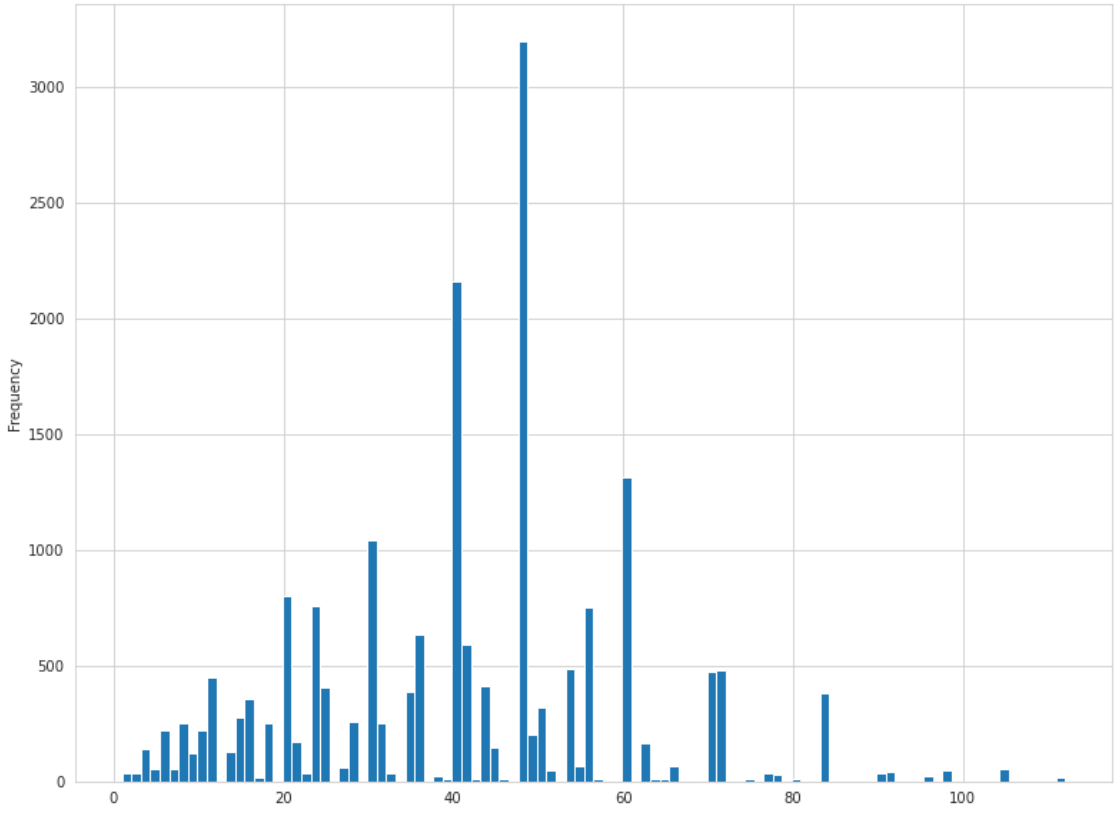
[115]:
```
pd.crosstab(df.tipohogar,df.target).plot(kind='bar', figsize=(13, 10))
#plt.title('Frecuencia de personas por genero')
#plt.xlabel('Tipo hogar')
#plt.ylabel('Frecuencia de personas')
```

[115]: <AxesSubplot:xlabel='tipohogar'>



[ ]:
```
# existen buena relacion de discrimiladidad entre la target del sexo, existe
↪alta frecuencia entre genero y personal ocupado
```

[116]:
```
plt = df['phrs'].plot.hist(bins= 100,figsize=(13, 10))
#plt.title('Frecuencia  por hora trabajadas')
#plt.xlabel('Genero')
#plt.ylabel('Frecuencia')
```



0.0.3 Recategorizacion de variables

[117]:
```
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()
```

[117]:

	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  …          pextcero  \
0       2     1977   1.JEFE O JEFA DEL HOGAR  …  No pobre extremo
1       5     1975   1.JEFE O JEFA DEL HOGAR  …  No pobre extremo
2       1     2015     3.HIJO/A O ENTENADO/A  …     Pobre extremo
3      11     1978   1.JEFE O JEFA DEL HOGAR  …  No pobre extremo
4       8     1988  2.ESPOSA/O O CONVIVIENTE  …  No pobre extremo

              pextuno            pextdos genero_e tipohogar_e razontrabaja_e  \
0                   0                  0        0           1              0
1                   0                  0        0           4              0
2  0.685652136802673  0.470118850469589        0           1              0
3                   0                  0        0           4              0
4                   0                  0        1           4              0

  cobersalud_e hijos_e ocupacion_e relacionjefehogar_e
0            0       0           2                   0
1            3       0           2                   0
2            3       0           0                   6
3            3       0           1                   0
4            3       8           2                   5

[5 rows x 187 columns]
```

[118]:
```
df['target'].value_counts()
```

[118]:
```
2    19151
3     9715
5     5369
4     4396
1      656
0      318
Name: target, dtype: int64
```

0.0.4 Crear otro dataframe df1 con las variables interesadas

[119]:
```
nomcol = ['edad','genero_e','hijos_e','tipohogar_e', 'cobersalud_e',␣
 ↪'razontrabaja_e', 'relacionjefehogar_e','ocupacion_e',␣
 ↪'ingresoocupacionprincipal', 'aestudio','target']
df1=df[nomcol]
df1.head()
```

[119]:
```
   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
```

[120]:
```
X= df1[df1.columns[:-1]]
y= df1['target']
X.head()
```

[120]:
```
    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
```

0.0.5 Aplicación de Gaussian Naive Bayes

[130]:
```
# 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)
```

[131]:
```
gnb = GaussianNB()
y_pred = gnb.fit(X_train, y_train).predict(X_test)
```

[132]:
```
y_pred
```

[132]:
```
array([3, 3, 2, …, 2, 2, 3])
```

0.0.6 Evaluación de la técnica de Gaussian Naive Bayes

[124]:
```
# calcular la precisión de la técnica
print(accuracy_score(y_test, y_pred))
```

```
0.9038882343039892
```

[ ]:
```
# Calcular la sensitividad, precision y exactitud
```

0.0.7 Aplicación de Multinomial Naive Bayes

[133]:
```
clf = MultinomialNB()
y_pred = clf.fit(X_train, y_train).predict(X_test)
```

[134]:
```
y_pred
```

[134]:
```
array([4, 3, 4, …, 1, 2, 3])
```

0.0.8 Evaluación de la técnica de Multinomial Naive Bayes

[135]:
```
# calcular la precisión de la técnica
print(accuracy_score(y_test, y_pred))
```

```
0.6275879481568759
```

[ ]:
```
# Tabla de contingencia observado vs estimado
```

0.1 Conclusiones

[ ]:
```
Existe una precision del 90, de cada 100 persona el modelo predice 90 a una␣
 ↪categoria de condicion laboral.
lo mismo con el multinomial.
```