Parte I

1

Luego de explorar el diseño de registro de la base de hogar, creemos que las siguiens variables puedenn ser muy predictivas de pobreza:

IV11 = El desague del baño es:

```
1. a red pública (cloaca)
2. a cámara séptica y pozo ciego
3. solo a pozo ciego
4. a hoyo/excavación en la tierra
```

IV12_3 = La vivienda está ubicada en villa de emergencia (por observación)

```
1 = Sí
2 = No
```

II7 = Régimen de tenencia:

```
01 = Propietario de la vivienda y el terreno
02 = Propietario de la vivienda solamente
03 = Inquilino / arrendatario de la vivienda
04 = Ocupante por pago de impuestos / expensas
05 = Ocupante en relación de dependencia
06 = Ocupante gratuito (con permiso)
07 = Ocupante de hecho (sin permiso)
08 = Está en sucesión
```

II8 = Combustible utilizado para cocinar:

```
01 = Gas de red
02 = Gas de tubo / garrafa
03 = Kerosene / leña / carbón
```

II9 = Baño (tenencia y uso):

```
01 = Uso exclusivo del hogar
02 = Compartido con otro/s hogar/es de la misma vivienda
03 = Compartido con otra/s vivienda/s
04 = No tiene baño
```

V5 = ¿En los últimos tres meses, las personas de este hogar han vivido de subsidio o ayuda social (en dinero)del gobierno, iglesias, etc.?:

```
1 = Sí
2 = No
```

V17 = ¿En los últimos tres meses, han tenido que vender alguna de sus pertenencias?:

```
1 = Sí
2 = No
```

2

3

4

5

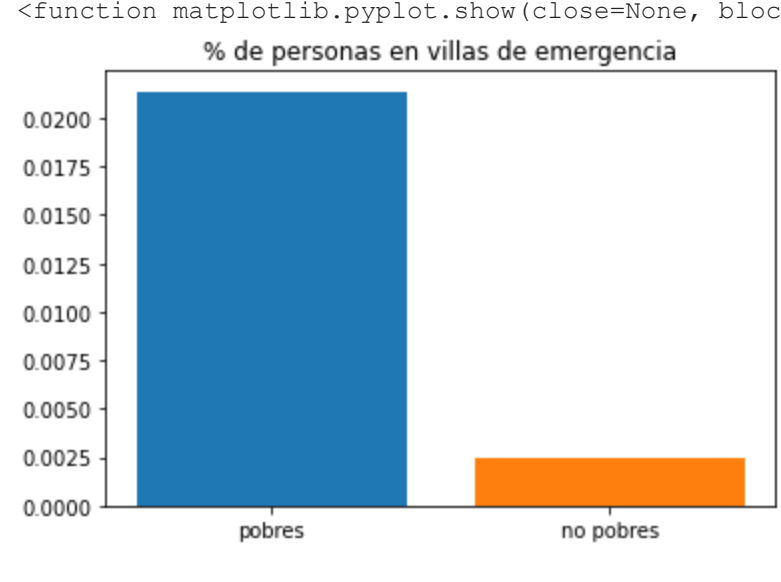
6 (hay que correr el codigo hasta el punto 9 iclusive antes de correr el codigo de este punto)

Las variables a analizar son 5 de las mencionadas en el punto 1:

```
IV12_3, II8, V5, V17 y II9
```

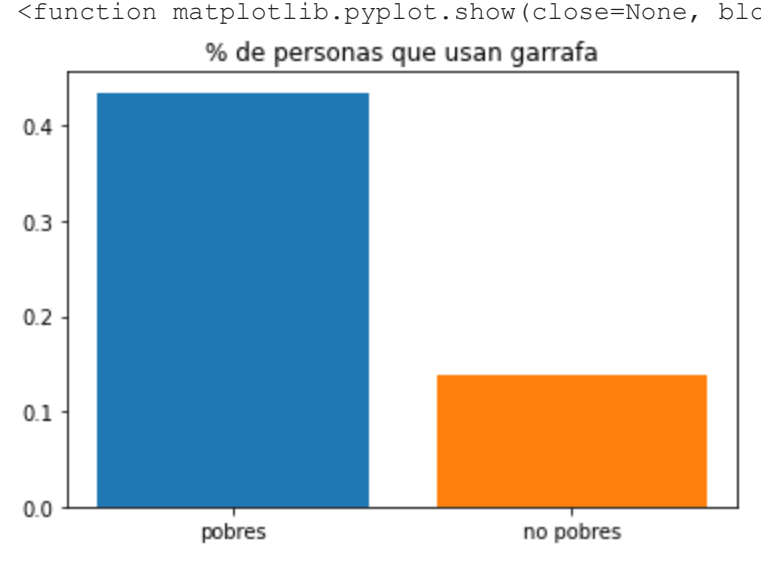
```
count    3333.000000
mean        1.990099
std         0.099025
min         1.000000
25%         2.000000
50%         2.000000
75%         2.000000
max         2.000000
Name: IV12_3, dtype: float64
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



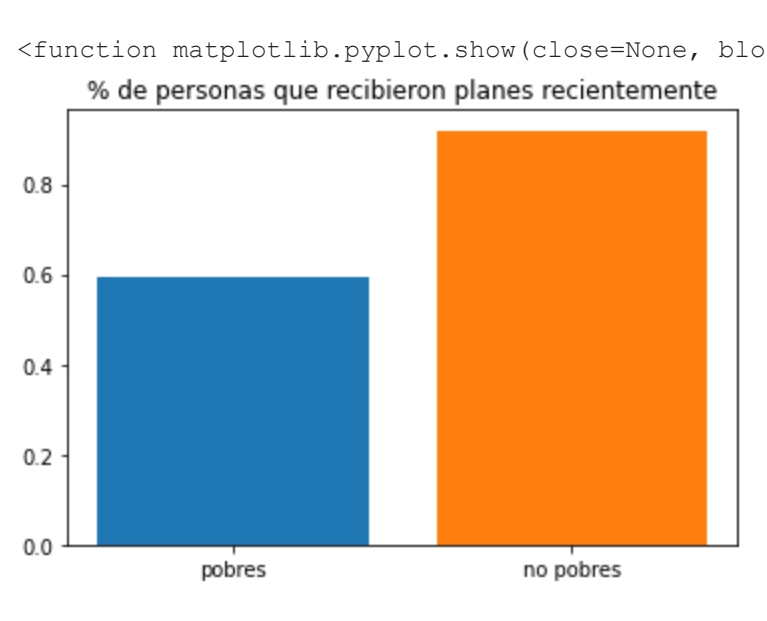

```
count    3333.000000
mean        1.294629
std         0.574747
min         0.000000
25%         1.000000
50%         1.000000
75%         2.000000
max         4.000000
Name: II8, dtype: float64
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



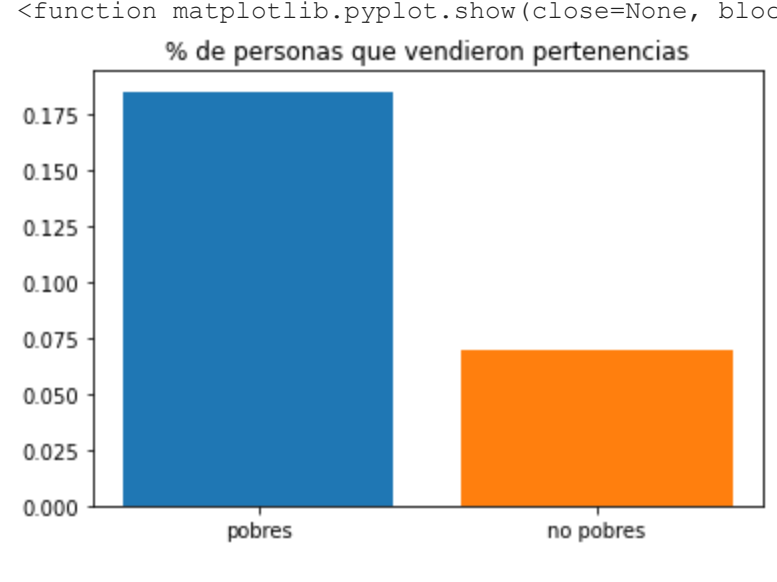

```
count    3333.000000
mean        1.840084
std         0.608011
min         1.000000
25%         2.000000
50%         2.000000
75%         2.000000
max         9.000000
Name: V5, dtype: float64
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



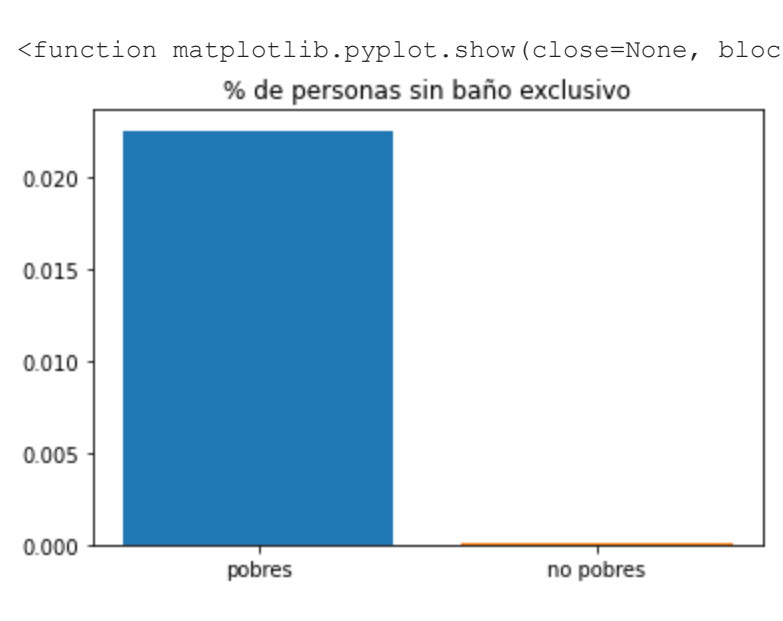

```
count    3333.000000
mean        1.927693
std         0.549903
min         1.000000
25%         2.000000
50%         2.000000
75%         2.000000
max         9.000000
Name: V17, dtype: float64
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
count    3333.000000
mean        1.033903
std         0.256470
min         0.000000
25%         1.000000
50%         1.000000
75%         1.000000
max         4.000000
Name: II9, dtype: float64
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



7

8

```
<ipython-input-10-9df0d126104c>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  respondieron["ingreso_necesario"] = ingreso_adulto_min * respondieron["ad_equiv_hogar"]
```

9

```
<ipython-input-11-f00b7e4e721c>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  respondieron["pobre"] = (respondieron["ITF"] < respondieron["ingreso_necesario"]).astype(int)
```

10

```
PONDIH    0.365065
dtype: float64
```

Por lo tanto, la tasa de hogares bajo la liea de pobreza para el GBA es del 36,5%. Por su parte, para el periodo que estamos analizando el Indec reporta que el porcentaje de hogares por debajo de la línea de pobreza (LP) alcanzó el 31,2%; en los cuales reside el 40,6% de las personas.

Parte II

1

2

3

4

Parte III

1

```
C:\Users\felip\anaconda3\lib\site-packages\pandas\core\frame.py:4308: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().drop(
C:\Users\felip\anaconda3\lib\site-packages\pandas\core\series.py:4463: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().fillna(
```

2

	modelo	parámetro	ecm	auc	accuracy
0	Lineal	2	0.197364	0.858998	0.802636
0	KNN	2	0.285258	0.745127	0.714742
0	Logit	2	0.194561	0.866403	0.805439
0	CART	2	0.336400	0.500000	0.663600
0	SVM	2	0.229324	NaN	0.770676
0	Bagging	2	0.256886	0.725775	0.743114
0	RandomForest	2	0.336400	0.500000	0.663600
0	Boosting	2	0.336404	0.500000	0.663596
0	Lasso	2	0.200558	0.867617	0.799442
0	Ridge	2	0.190572	0.871828	0.809428

3

λ lo elegiríamos por cross validation ya que queremos minimizar el error de pronostico por fuera de la muestra. Por lo tanto, debe estimarse el modelo para muchos valores de λ alternativos y calcular el error de pronóstico por cross validation. Para esto primero hay qye partir los datos al azar en K partes, luego ajustar el modelo dejando afuera una de las particiones y, por último, Computar el error de predicción para los datos no utilizados. Esto debe repetirse para k = 1,...,K. Una vez hecho esto, vamos a elegir el valor de λ que genere la mejor capacidad predictiva, es decir, el que minimiza el error de pronóstico fuera de la muestra computado por cross validation. Este es el sentido en el cual el modelo esta aprendiendo, dado que estamos haciendo que los datos elijan el λ óptimo.

Para la eleccion no debe usarse el conjunto de prueba dado que no podríamos comparar el desempeño con otros modelos dado que todos los modelos se estiman con los datos de entrenamiento y luego se evalúan con los datos de test. Por lo tanto, si en un modelo a λ lo computamos usando los datos de test, deja de ser comparable con el resto de los modelos, ya que tendría una ventaja por sobre todos los demás.

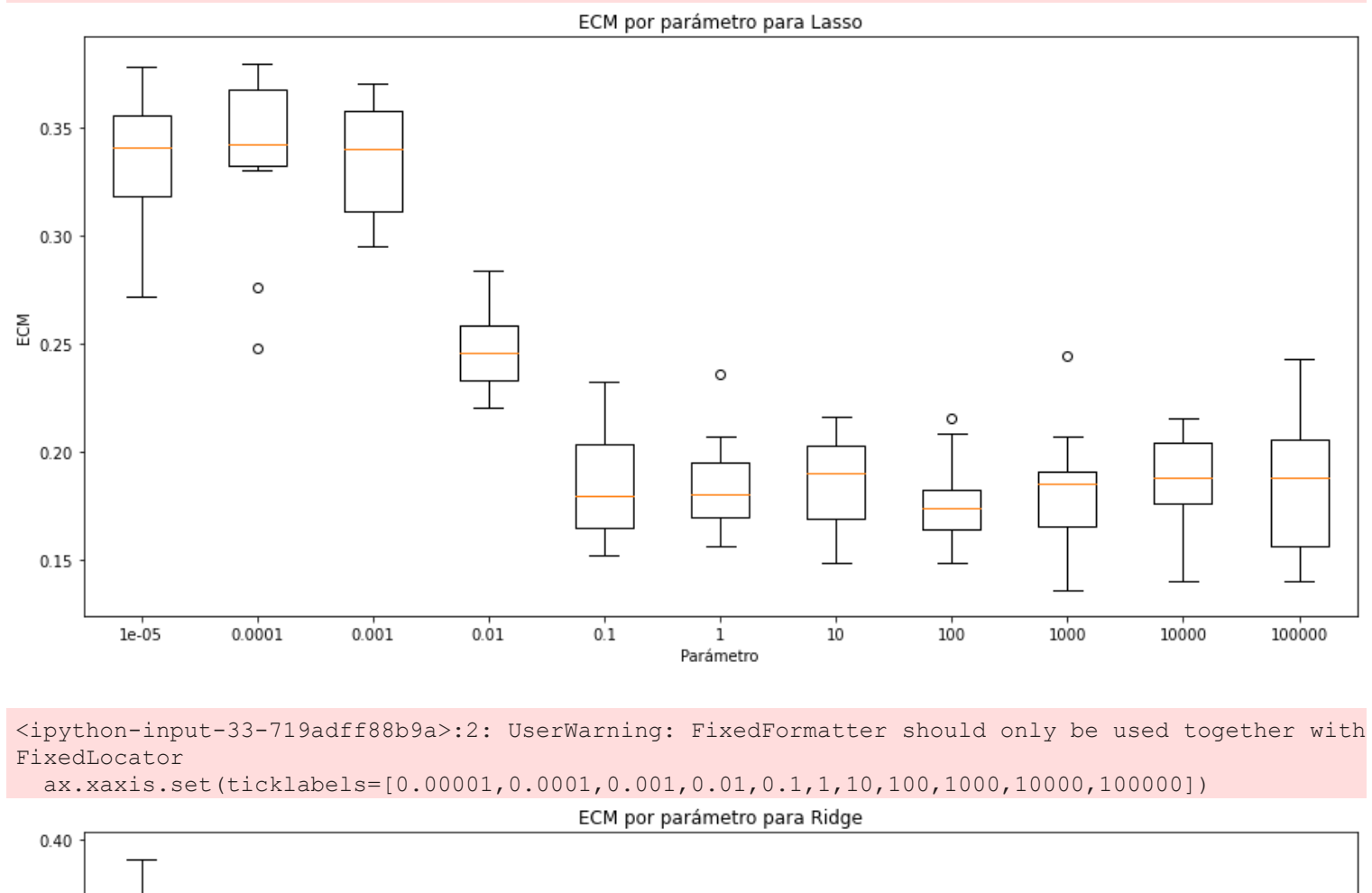
4

El problema de usar un K muy pequeño es que a pesar de que maximiza los datos para estimar, es sensible a los valores particulares para la evaluación. Por lo que el error de pronostico depende mucho de lo que pase con los pocos datos de test. Por otro lado, el problema de usar un muy K grande es que a pesar de que maximiza los datos para evaluar, el modelo es estimado menos precisamente dado que se usan menos datos de entrenamiento. Por último, cuando K = N se va dejando de lado una observación por vez. Así, se estima el modelo n veces con n − 1 datos ('leave one out')

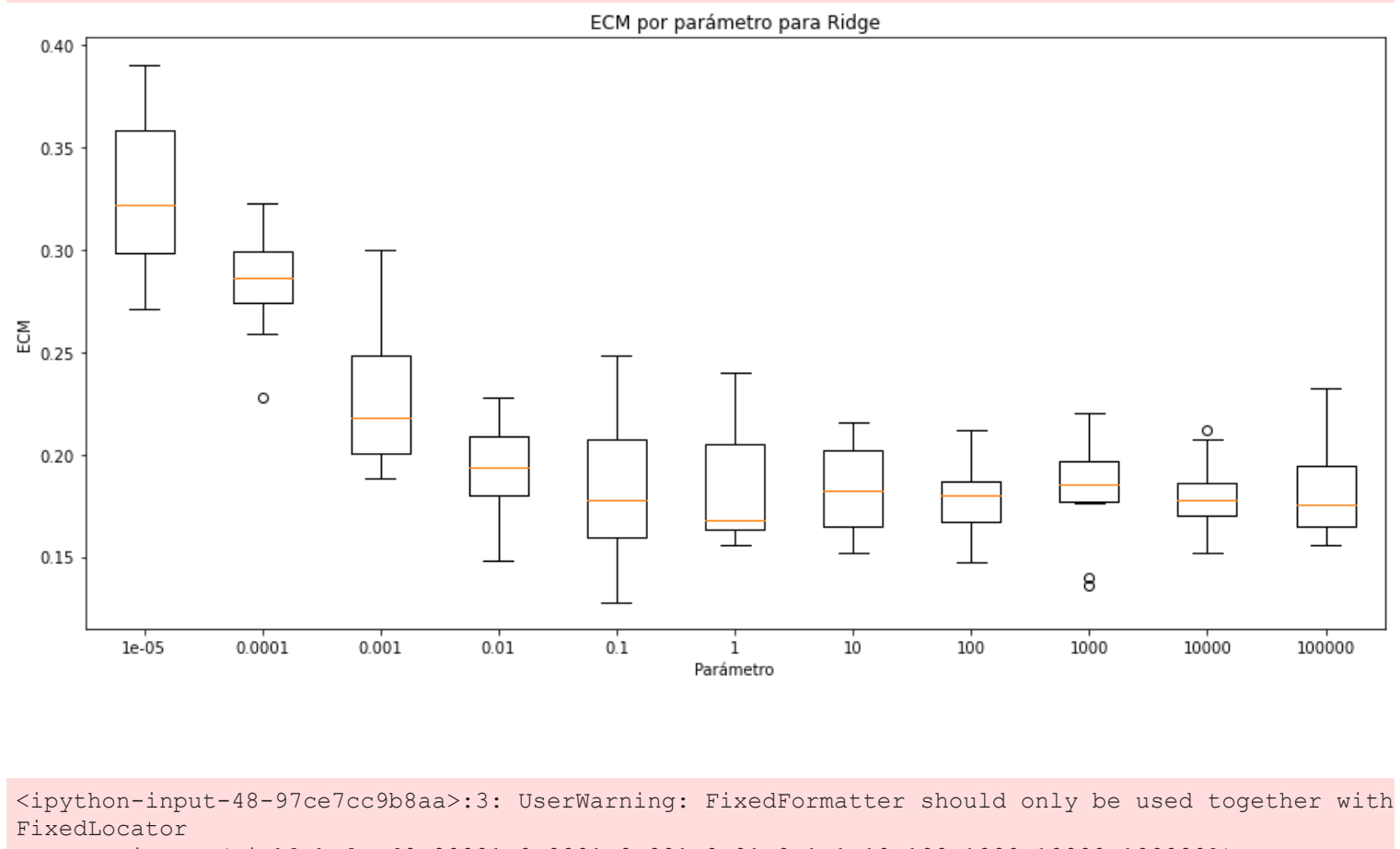
5

El λ elegido en regresión logísitica con Ridge y Lasso es 100.

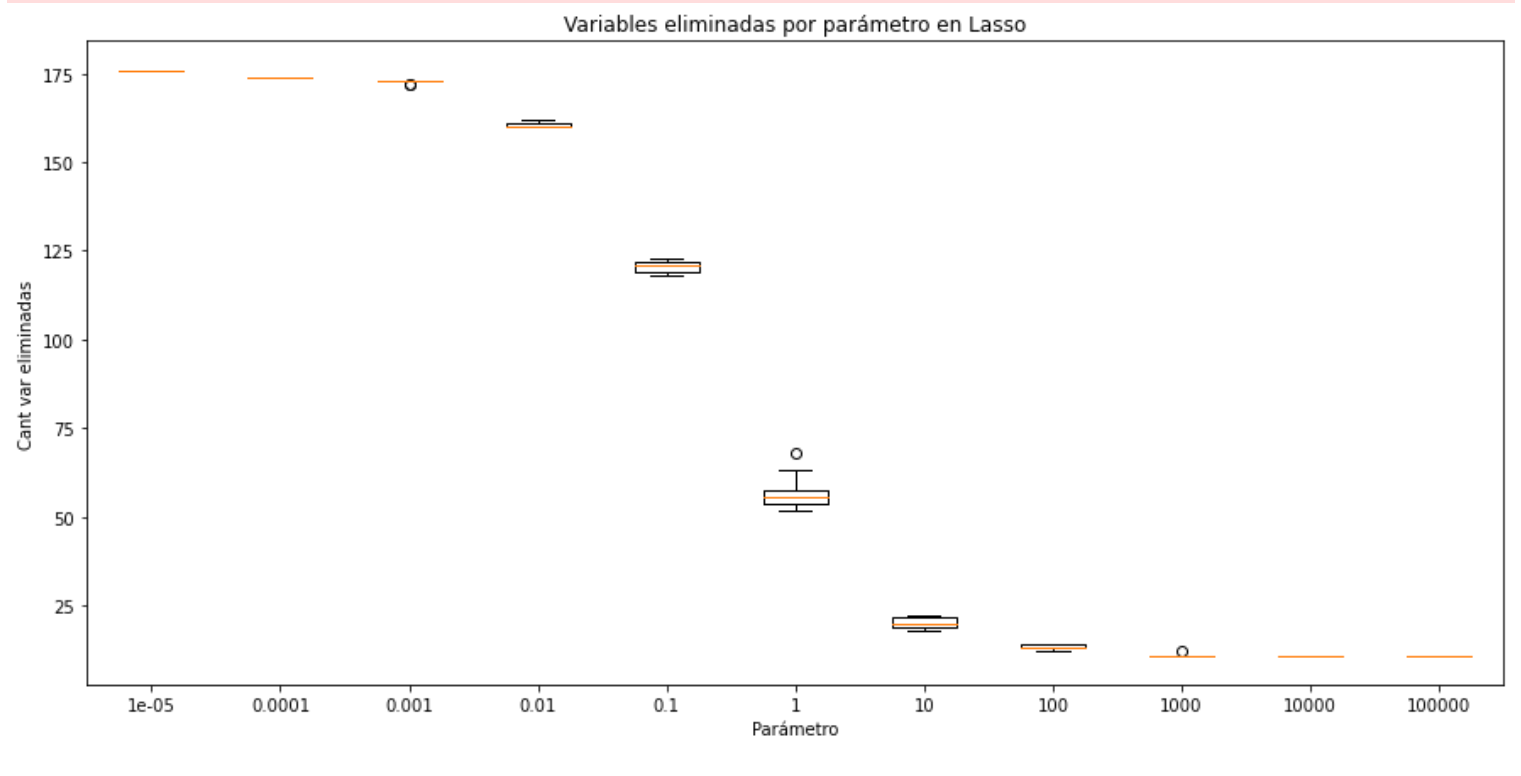
```
<ipython-input-32-5bd245eb43ce>:2: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.xaxis.set(ticklabels=[0.00001,0.0001,0.001,0.01,0.1,1,10,100,1000,10000,100000])
```




```
<ipython-input-33-719adff88b9a>:2: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.xaxis.set(ticklabels=[0.00001,0.0001,0.001,0.01,0.1,1,10,100,1000,10000,100000])
```

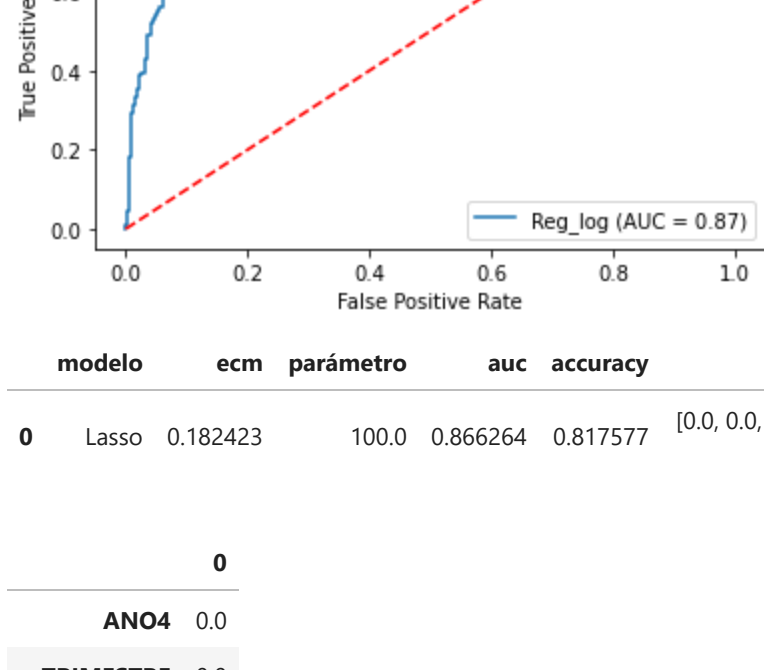



```
<ipython-input-48-97ce7cc9b8aa>:3: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.xaxis.set(ticklabels=[0.00001,0.0001,0.001,0.01,0.1,1,10,100,1000,10000,100000])
```



6

	modelo	ecm	parámetro	auc	accuracy	fpr	tpr
0	Lasso	0.177097	100.0	0.865672	0.822903	[0.0, 0.0, 0.0, 0.0020242914979757085, 0.00202...	[0.0, 0.0038910505836575876, 0.116731517509727...



	modelo	ecm	parámetro	auc	accuracy	fpr	tpr
0	Lasso	0.182423	100.0	0.866264	0.817577	[0.0, 0.0, 0.0, 0.00202020202020202, 0.0020202...	[0.0, 0.00390625, 0.01171875, 0.01171875, 0.04...

	0
ANO4	0.0
TRIMESTRE	0.0
REALIZADA	0.0
REGION	0.0
MAS_500	0.0
V22	0.0
V19_A	0.0
V19_B	0.0
VII2_4	0.0
PP05B2_ANO	0.0
PP05B2_DIA	0.0
PP09B	0.0
PP09C	0.0
inter	0.0

Las variables descartadas son las que hubieramos esperado, ya que creemos que al ver el diseño de registro de la base hogar parecían ser irrelevantes para explicar la tasa de pobreza. Por ejemplo el año de relevamiento (ANO4), la ventana de observación (TRIMESTRE) y el codigo de region (REGION) no parecen variables que tengan una relacion significativa con la tasa de pobreza. Con respecto a lo respondido en el inciso 1 de la Parte 1, ninguna de las variables que creíamos a priori relevantes para predecir pobreza fueron eliminidas, por lo que efectivamente presentan un poder explicativo significativo.

7

```
0.17697211155378484
```

```
0.18019442231075694
```

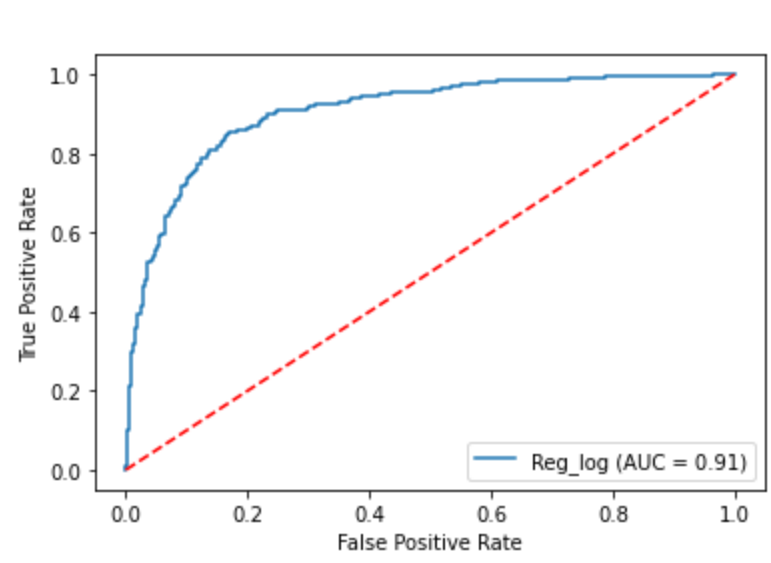
Por lo tanto, Lasso funcionó mejor ya que se obtiene un error cuadratico medio menor que en el caso de Ridge.

8

	modelo	ecm	parámetro	auc	accuracy	fpr	tpr
0	Lineal	0.175766	0.00100	0.871792	0.824234	[0.0, 0.0, 0.0, 0.0019120458891013384, 0.00191...	[0.0, 0.0043859649122807015, 0.017543859649122...
0	KNN	0.282290	100000.00000	0.728612	0.717710	[0.0, 0.011904761904761904, 0.0694444444444444...	[0.0, 0.07692307692307693, 0.20647773279352227...
0	Logit	0.198402	100000.00000	0.873946	0.801598	[0.0, 0.0, 0.0, 0.002004008016032064, 0.002004...	[0.0, 0.003968253968253968, 0.1388888888888889...
0	CART	0.153129	0.00100	0.847391	0.846871	[0.0, 0.06693711967545639, 0.07505070993914807...	[0.0, 0.3798449612403101, 0.4496124031007752, ...
0	SVM	0.238349	10.00000	NaN	0.761651	NaN	NaN
0	Bagging	0.201065	100.00000	0.837413	0.798935	[0.0, 0.030800821355236138, 0.0492813141683778...	[0.0, 0.4356060606060606, 0.48484848484848486,...
0	RandomForest	0.139814	0.00001	0.940995	0.860186	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[0.0, 0.016129032258064516, 0.0322580645161290...
0	Boosting	0.115846	0.00001	0.910769	0.884154	[0.0, 0.0, 0.0, 0.00411522633744856, 0.0041152...	[0.0, 0.0037735849056603774, 0.064150943396226...

El método que predice mejor es Boosting con el hiperparámetro 0.00001, dado que presenta el mínimo error cuadrático medio.

9




```
0.43995098039215685
```

El 43,9% de los hogares son pobres en ests submuestra