

Nicolas Segura

Jaime Torres

David Ruiz

Recomendaciones ClinicaAlpes

Link del repositorio: <https://github.com/drwillota/Laboratorio-3>

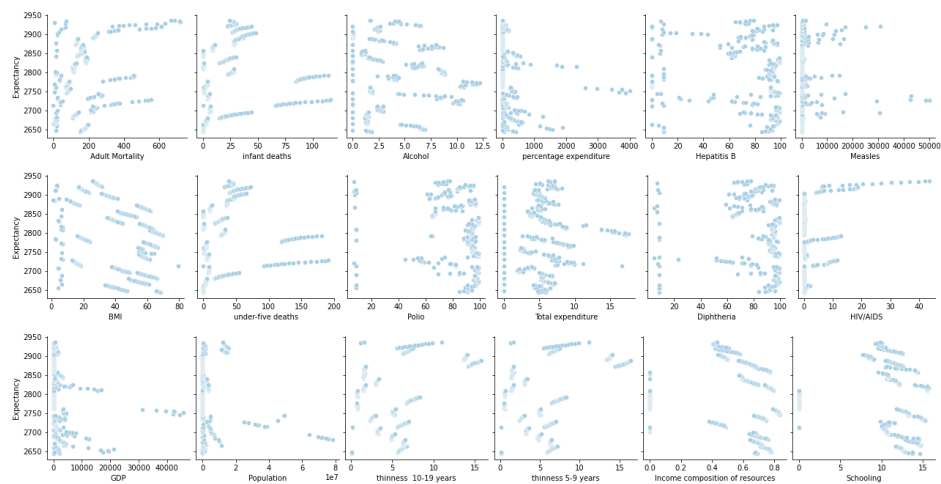
Link presentación: https://www.canva.com/design/DAE7kenvPRU/T3MxCvYhggoEbb7iTx-GMg/edit?utm_content=DAE7kenvPRU&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

El dashboard se encuentra disponible en el repositorio.

Carga y limpieza de los datos

Primero cargamos todos los datos y luego los visualizamos para ver que tipos de datos tiene las variables. La base de datos cuenta con 294 filas y cada una con 10 atributos

	Expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Population	thinness 10-19 years	thinness 5-9 years	Income composition of resources	Schooling
0	2644	151.0	0	1.80	423.295351	9.0	0	68.6	0	91.0	4.87	9.0	0.1	2284.378580	146.0	0.1	0.1	0.693	14.6
1	2645	153.0	0	1.79	45.851058	85.0	0	67.8	0	91.0	5.90	9.0	0.1	229.714718	99789.0	0.1	0.1	0.683	13.7
2	2646	155.0	0	1.51	310.820338	88.0	0	67.0	0	85.0	5.30	84.0	0.1	1842.444210	99184.0	0.1	0.1	0.679	13.5
3	2647	157.0	0	1.35	330.100739	91.0	4	66.2	0	91.0	5.66	89.0	0.1	1837.977391	98611.0	0.1	0.1	0.674	13.2
4	2648	158.0	0	1.24	40.491289	93.0	0	65.5	0	91.0	4.75	91.0	0.1	263.272360	9882.0	0.1	0.1	0.676	13.7



Eliminamos las celdas con valor nulo

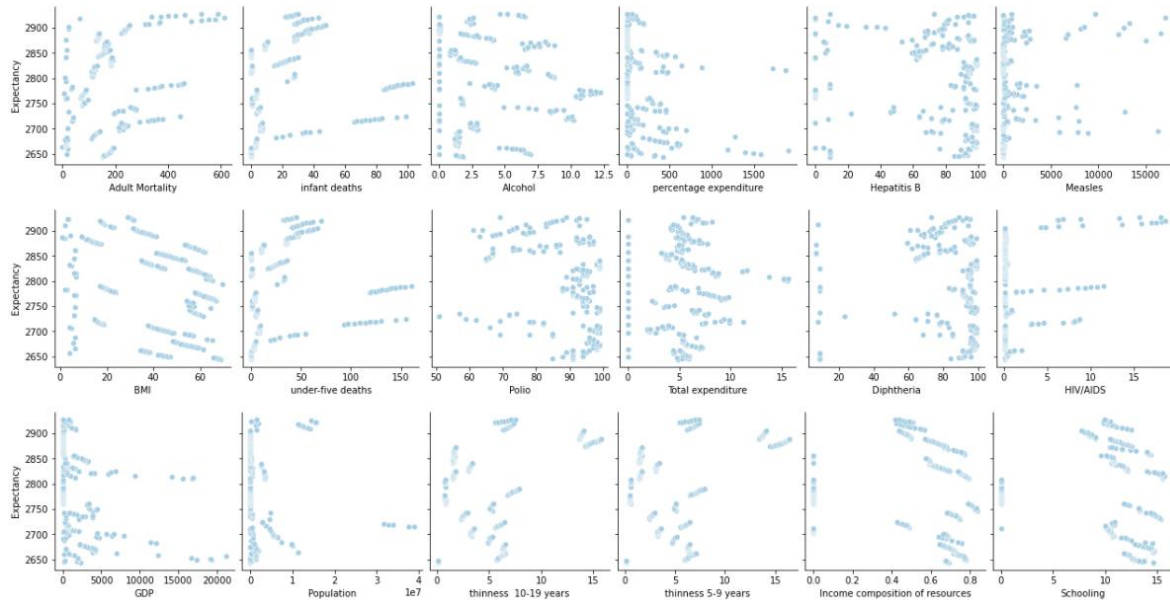
	Expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Population	thinness 10-19 years	thinness 5-9 years	Income composition of resources	Schooling
count	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000	294.000000
mean	2790.500000	180.156463	22.786298	4.091327	230.691788	67.259533	2296.707483	39.817945	21.821788	82.49184	5.934614	80.378237	2.886327	2888.884225	4.541964e+06	5.141487	5.133274	0.802966	9.891280
std	85.614705	146.969676	28.965706	3.411981	636.324313	35.666719	4887.681389	20.323780	43.121543	21.933034	3.283364	24.822111	6.876873	7286.383426	1.293469e+07	4.897866	4.123723	0.289985	4.827673
min	2644.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	7.000000	0.000000	0.000000	0.100000	0.000000	0.000000e+00	0.100000	0.100000	0.000000	0.000000
25%	2717.230000	82.000000	3.000000	1.375000	0.000000	51.000000	0.000000	19.300000	3.000000	75.000000	4.400000	75.000000	0.100000	0.000000	0.000000e+00	1.625000	1.800000	0.415000	8.725000
50%	2790.500000	153.000000	10.000000	2.720000	27.137321	83.000000	35.500000	43.000000	12.000000	92.000000	5.400000	92.000000	0.100000	430.824070	1.055190e+05	5.920000	4.900000	0.600500	11.100000
75%	2863.750000	231.000000	29.000000	6.632500	194.536681	94.000000	816.500000	57.475000	42.000000	96.000000	7.075000	96.000000	0.775000	2244.678564	2.482152e+06	6.675000	6.700000	0.715750	12.975000
max	2937.000000	723.000000	116.000000	12.200000	4053.808588	99.000000	49871.000000	75.300000	191.000000	99.000000	17.600000	99.000000	43.500000	43756.955400	7.827147e+07	15.900000	16.400000	0.836000	15.700000

Se calcula el puntaje z para calcular qué tan lejos está de la desviación estándar, siendo eliminados de los datos los valores menores a 3

```
restr = data_or.apply(lambda x:np.abs(stats.zscore(x))<3).all(axis=1)
data_cl = data_or.drop(data_or.index[~restr],inplace=False)
data_cl.shape
```

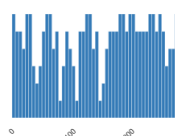
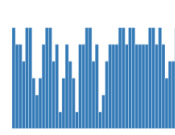
(230, 19)

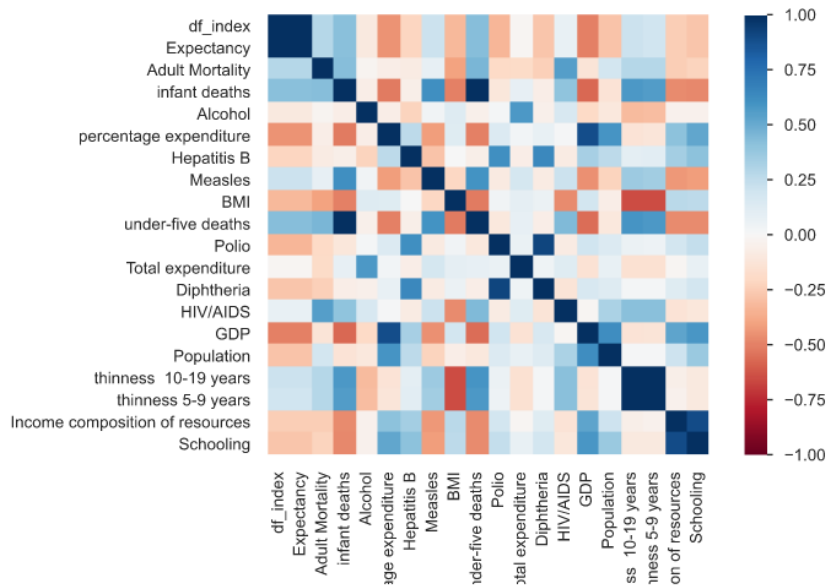
Graficamos las relaciones entre la variable objetivo y el resto en el conjunto de datos para un mejor análisis.



Usamos p-andas para ver a mas detalle los datos, este método también nos permite ver la correlación entre los datos

Variables

df_index Real number (R ₆₀) HIGH_CORRELATION HIGH_CORRELATION HIGH_CORRELATION HIGH_CORRELATION UNIQUE				Distinct 230 Distinct (%) 100.0% Missing 0 Missing (%) 0.0% Infinite 0 Infinite (%) 0.0% Mean 146.3	Minimum 0 Maximum 284 Zeros 1 Zeros (%) 0.4% Negative 0 Negative (%) 0.0% Memory size 1.9 KiB	
Expectancy Real number (R ₆₀) HIGH_CORRELATION HIGH_CORRELATION HIGH_CORRELATION HIGH_CORRELATION UNIQUE				Distinct 230 Distinct (%) 100.0% Missing 0 Missing (%) 0.0% Infinite 0 Infinite (%) 0.0% Mean 2790.3	Minimum 2644 Maximum 2928 Zeros 0 Zeros (%) 0.0% Negative 0 Negative (%) 0.0% Memory size 1.9 KiB	



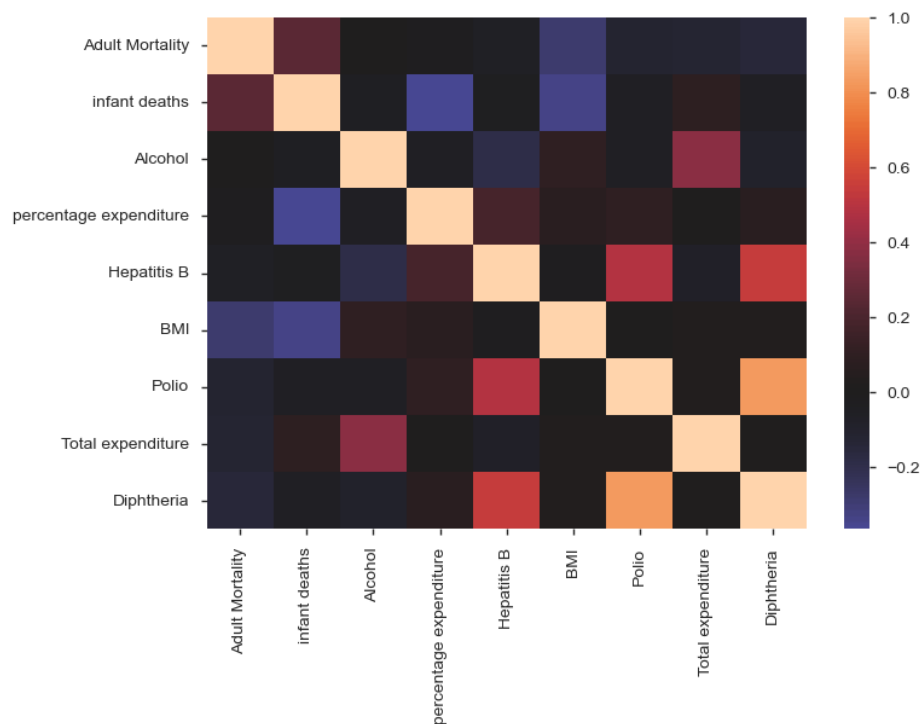
Modelamiento y análisis

Primero, separamos la variable objetivo del modelo y creamos datos de entrenamiento y datos de prueba.

```
# Se selecciona la variable objetivo, en este caso "Adult Mortality".
Y = data_cl['Expectancy']
# Del conjunto de datos se elimina la variable "Adult Mortality"
X = data_cl.drop(['Expectancy'], axis=1)
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=seed)
df_train = pd.concat([X_train, Y_train], axis=1)
df_test = pd.concat([X_test, Y_test], axis=1)
```

A continuación, revisamos la colinealidad para analizar la relación que existe entre variables.



```

columns      coef
0      Adult Mortality 241.402131
1      infant deaths  -39.209483
2      Alcohol        -53.037230
3  percentage expenditure -247.890629
4      Hepatitis B    11.430407
5      BMI           -26.487210
6      Polio         -111.405566
7      Total expenditure 191.696327
8      Diphtheria    103.032329
0.3999286025317751

```

Se puede concluir que las variables “Adult Mortality”, “Total expenditure” y “Diphtheria” son las variables con mayor colinealidad.

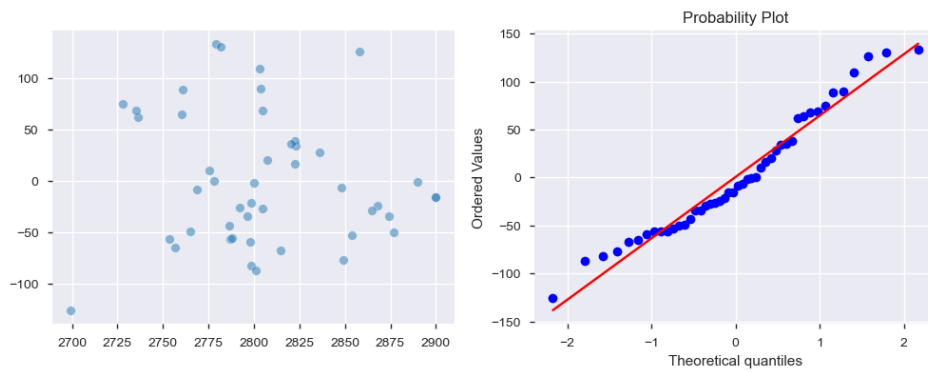
```

# Note que hay que sacarle la raiz al valor
np.sqrt(mse(Y_test, pipeline.predict(X_test)))

```

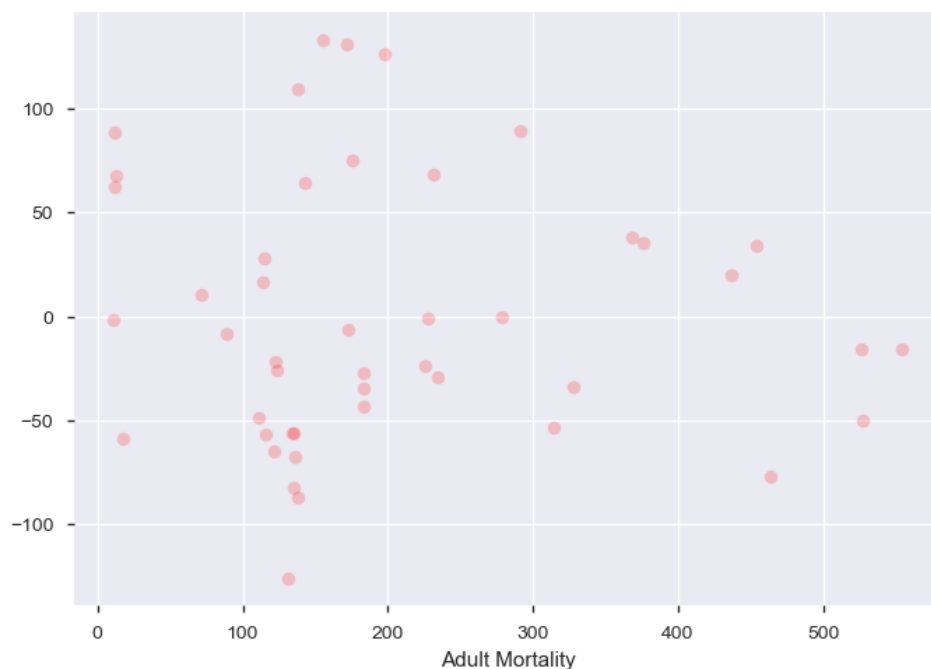
```
62.64274117642231
```

Retomando, calculamos el error cuadrático medio entre la variable objetivo de prueba y la predicción del modelo pipeline con los datos de prueba, lo cual nos arroja un valor elevado, siendo este, indicio de un modelo impreciso.



Igualmente, podemos para facilitar el análisis, se graficó la normalidad entre la predicción del modelo con los datos de prueba y la variable objetivo.

```
sns.scatterplot(data = df_test, x = 'Adult Mortality', y = errors, alpha = 0.2, color='red')
<AxesSubplot:xlabel='Adult Mortality'>
```



Finalmente, revisamos si en el modelo hay homocedasticidad; Concluimos que la varianza no es constante.