




# *PRÁCTICA 1*

## *MACHINE LEARNING*



*Álvaro González Tabernero & Santiago Arenas Martín*  
*20 de septiembre de 2023*

## Contenido

1. Introducción al estudio.....	2
2. Creación de <i>dummies</i> .....	3
3. Decidiendo las variables .....	4
a. VIF.....	4
b. Búsqueda de correlaciones .....	5
c. Descartando por p-valores .....	6
4. Modelo final.....	7

## Introducción al estudio

A lo largo de esta práctica se va a elaborar un estudio estadístico utilizando el método de regresión lineal múltiple, conocido como método de mínimos cuadrados. La variable dependiente sobre la que recae el foco del estudio es la magnitud de la tasa de un determinado seguro, según diferentes variables.

Lo que conocemos sobre la población de la que se recabó información es: su edad, su sexo, su índice de masa corporal (IMC), su número de hijos, si son fumadores, la región en la que se encuentran y finalmente, la tasa de su seguro.

Tanto la edad, como el IMC y el número de hijos son variables cuantitativas cardinales enteras, salvo el IMC que es un número real positivo. Sobre estas variables, por su naturaleza, no hay necesidad de transformarlas en variables *dummy*.

En cambio su sexo, si fuman y la región en la que se encuentran son variables cualitativas nominales, que no admiten graduación. Por tanto, sí se encuentran en la necesidad de ser transformadas en variables *dummy*. Para saber cuántas necesitamos, se usa la fórmula:

$$n = k - 1$$

Siendo  $n$  el número de variables *dummy* resultantes y  $k$  el número de categorías posibles para la variable. Siguiendo esta lógica, necesitamos sólo una variable para el sexo y otra para si fuman o no. Al haber 4 distintas respuestas para la variable región, se usan 3 variables *dummy*.

## Creación de *dummies*

Tal y como se ha mencionado en la introducción, existen varias variables en nuestro set de datos que requieren ser transformadas a *dummies*

```
insurance = pd.read_csv('insurance.csv')
data=pd.get_dummies(insurance, columns=['sex','smoker', 'region'],
prefix="dmy",drop_first=True) # drop_first removes 0 values leaving k-1 levels
data.head()
```

	age	bmi	children	charges	dmy_male	dmy_yes	dmy_northwest	dmy_southeast	dmy_southwest
0	19	27.900	0	16884.92400	0	1	0	0	1
1	18	33.770	1	1725.55230	1	0	0	1	0
2	28	33.000	3	4449.46200	1	0	0	1	0
3	33	22.705	0	21984.47061	1	0	1	0	0
4	32	28.880	0	3866.85520	1	0	1	0	0

Desde este momento, se utilizará todo el *dataset* ‘data’, debido a que ya se han transformado las variables cualitativas a variables con las que se puede realizar el estudio.

## Decidiendo las variables

### VIF

```
x_all = data.iloc[:, [0,1,2,4,5,6,7,8]]
# VIF estimation

from statsmodels.stats.outliers_influence import variance_inflation_factor

print("Columns for VIF estimation", x_all.columns)

# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x_all.columns

# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x_all.values, i)
                   for i in range(len(x_all.columns))]

print(vif_data)
```

Columns for VIF estimation Index(['age', 'bmi', 'children', 'dmy\_male', 'dmy\_yes', 'dmy\_northwest', 'dmy\_southeast', 'dmy\_southwest'], dtype='object')

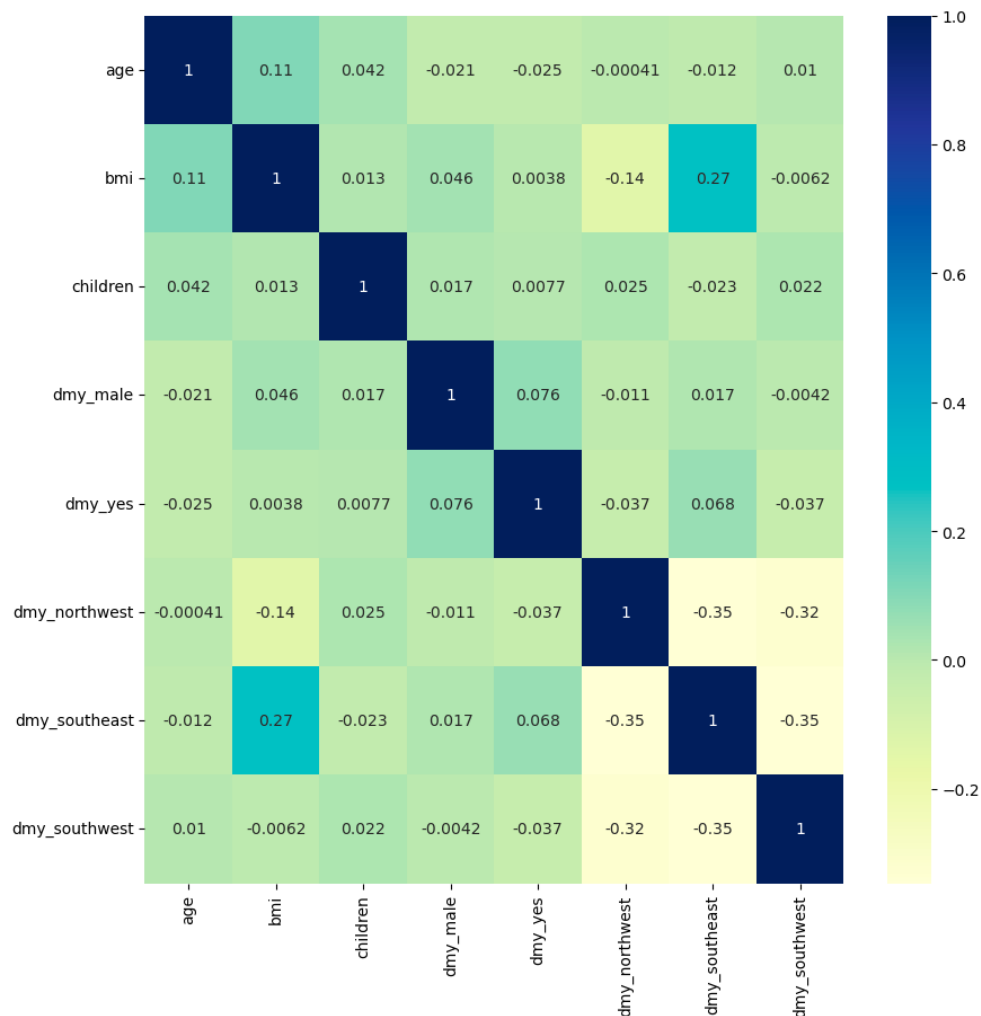
	feature	VIF
0	age	7.686965
1	bmi	11.358443
2	children	1.809930
3	dmy_male	2.003185
4	dmy_yes	1.261233
5	dmy_northwest	1.890281
6	dmy_southeast	2.265564
7	dmy_southwest	1.960745

El VIF indica la colinealidad de las distintas variables entre sí.

## Búsqueda de correlaciones

Para saber si dos variables están muy correlacionadas, y por tanto poder prescindir de ellas, se hace un mapa con colores y valores. Se ha incluido también el BMI, a pesar de que no se vaya a utilizar posteriormente, para tener una imagen más verídica de los datos.

```
plt.figure(figsize=(10,10))
sns.heatmap(x_all.corr(), cmap="YlGnBu", annot = True)
plt.show()
```



Como puede comprobarse, no hay ninguna correlación significativa entre ninguna de las variables, ya que el mayor valor se encuentra entre el BMI y vivir en el Southeast.

## Descartando por p-valores

Por último, decidimos mirar si la localización individualmente era significativo en cuanto a p-valores. Para ello cargamos los dummies de las regiones en la variable *reg* e hicimos un modelo.

```
X=reg
y = insurance['charges']
X_sm = sm.add_constant(X)
model_reg = sm.OLS(y, X_sm).fit()

print(model_reg.summary())
```

OLS Regression Results						
=====						
Dep. Variable:	charges	R-squared:	0.007			
Model:	OLS	Adj. R-squared:	0.004			
Method:	Least Squares	F-statistic:	2.970			
Date:	Wed, 20 Sep 2023	Prob (F-statistic):	0.0309			
Time:	17:13:45	Log-Likelihood:	-14473.			
No. Observations:	1338	AIC:	2.895e+04			
Df Residuals:	1334	BIC:	2.898e+04			
Df Model:	3					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]
-----						
const	1.341e+04	671.297	19.971	0.000	1.21e+04	1.47e+04
northwest	-988.8091	948.626	-1.042	0.297	-2849.771	872.153
southeast	1329.0269	922.907	1.440	0.150	-481.480	3139.534
southwest	-1059.4471	948.626	-1.117	0.264	-2920.409	801.515
=====						
Omnibus:	327.391	Durbin-Watson:	2.001			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	623.271			
Skew:	1.484	Prob(JB):	4.55e-136			
Kurtosis:	4.541	Cond. No.	4.86			

Son p-valores muy altos, así que descartamos la localización para el modelo final.

## Modelo final

Con las variables elegidas realizamos el modelo.

```
X_comp= data.iloc[:, [0, 2, 5]]
y_comp = insurance['charges']
X_sm = sm.add_constant(X_comp)
model_comp = sm.OLS(y_comp, X_sm).fit()
print(model_comp.summary())
```

```

=====
                        OLS Regression Results
=====
Dep. Variable:          charges      R-squared:                0.724
Model:                  OLS         Adj. R-squared:           0.723
Method:                 Least Squares   F-statistic:             1165.
Date:                   Wed, 20 Sep 2023   Prob (F-statistic):       0.00
Time:                   17:13:45         Log-Likelihood:          -13617.
No. Observations:       1338             AIC:                    2.724e+04
Df Residuals:           1334             BIC:                    2.726e+04
Df Model:                3
Covariance Type:        nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	-2851.9858	543.784	-5.245	0.000	-3918.750	-1785.222
age	273.0888	12.419	21.990	0.000	248.726	297.451
children	486.6524	144.700	3.363	0.001	202.789	770.516
dmy_yes	2.384e+04	431.840	55.212	0.000	2.3e+04	2.47e+04

```

=====
Omnibus:                265.851      Durbin-Watson:           2.089
Prob(Omnibus):           0.000      Jarque-Bera (JB):        648.127
Skew:                    1.070      Prob(JB):                1.82e-141
Kurtosis:                5.654      Cond. No.                134.
=====

```

Todos los p-valores son bajos, y el modelo tiene una R de 0.724, es decir, que éste explicaría un 72.4% de los precios del seguro. Las  $\beta$ 's son las que se encuentran bajo la columna de *coef*, siendo por orden descendiente  $\beta_0$ ,  $\beta_1$ ,  $\beta_2$  y  $\beta_3$ .