Instituto Politécnico Nacional

Escuela Superior de Computo

Licenciatura en Ciencia de Datos

Aprendizaje de Máquina e Inteligencia Artificial

Grupo: 4AM1

Práctica II: Regresión Lineal
Unidad 2: Aprendizaje Supervisado

Fecha de entrega: 25 de marzo de 2022

Profesor:

Joel Omar Gambino Juárez

Equipo:

Alcibar Zubillaga Julian - De Luna Ocampo Yanina - Ramirez Mendez Kevin - Sainz Takata Juan Pablo Minoru

Índice

Aprendizaje de Máquina e Inteligencia Artificial
Práctica II: Regresión Lineal

25 de marzo de 2022

Resumen

En esta práctica pondremos a prueba los conocimientos que hemos tenido en el curso de Aprendizaje Máquina e Inteligencia Artificial. A continuación daremos una breve explicación de lo que buscamos resolver y con qué métodos nos estamos ayudando.

1. Introducción

Recordemos que la regresión lineal es un modelo matemático, que forma parte de los modelos supervisados, utilizado para aproximar la relación de dependencia entre una variable dependiente Yi, las variables independientes Xi, y un término aleatorio épsilon, para que el modelo aprenda a predecir. Tanto el coeficiente de correlación lineal, cómo la regresión lineal cumplen una propiedad muy importante, que es basarse en la fórmula de una línea recta. La ecuación es la siguiente:

$$y = mx + b.$$

Haremos un análisis para aprender, como se mencionó previamente, el aprendizaje automático. El dataset dado tiene una lista de variables fácilmente comprensibles. Los datos se refieren a las casas que se encuentran en un determinado distrito. Las columnas son las siguientes:

longitude

latitude

housingmedianage

totalrooms

totalbedrooms

population

households

medianincome

medianhousevalue

2. Pasos a seguir con esta práctica.

2.1. Identificar el dato a predecir.

Esto es de suma importancia, ya que con este, sabremos con que comparar nuestras columnas restantes del dataset dado.

2.2. Generar el conjunto de entrenamiento y de prueba.

Como recordaremos, por medio de Shuffle, mezclamos nuestros datos para obtener un mejor análisis y así poderlos dividir para que no se queden los datos cargados de más en una división y en la otra no.

2.3. Conjunto de validación de 10 pliegues.

Con este ajustamos los parámetros de nuestro clasificador. La prueba de validación evalúa la capacidad del programa de acuerdo con la variación de parámetros para ver cómo podría funcionar en pruebas sucesivas.

2.4. Relación entre las variables y el valor a predecir.

Nos ayuda a visualizar, como dice, la relación entre cada una de las variables con el valor que se espera predecir. Asimismo para poder visualizar la relación de los datos de mejor manera y así, poder entender cuáles son las nos ayudan más en nuestro análisis, en este caso es comparar "medianHouseValueçon cada una de las variables dadas en el dataset.

A continuación veremos las gráficas que nos salieron en este paso.

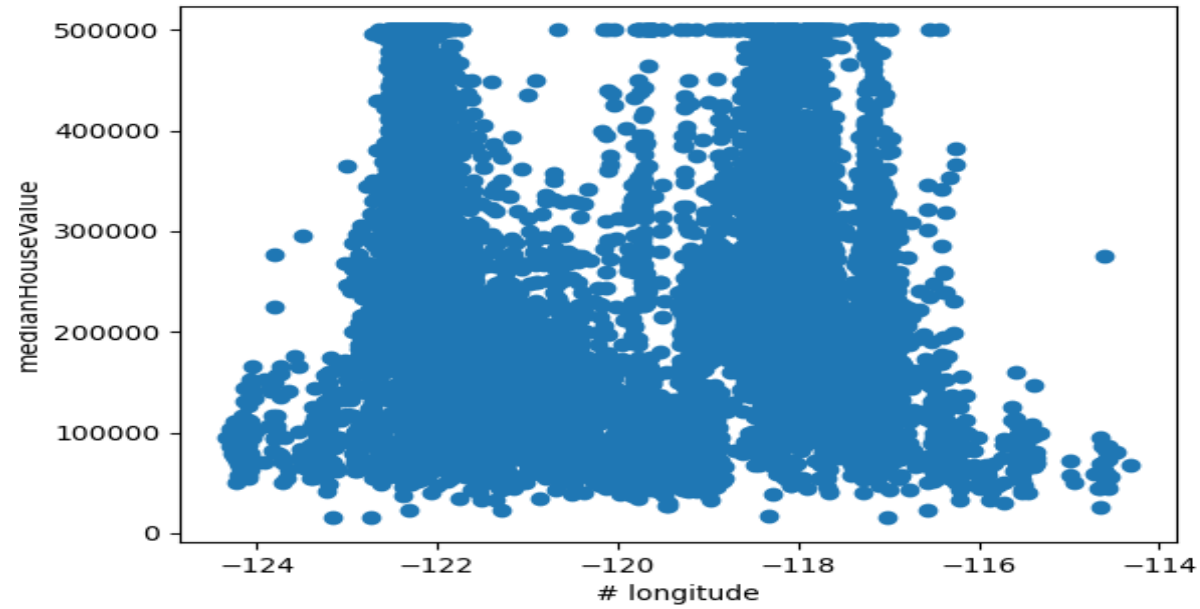

Figura 1: *medianHouseValue con longitud*

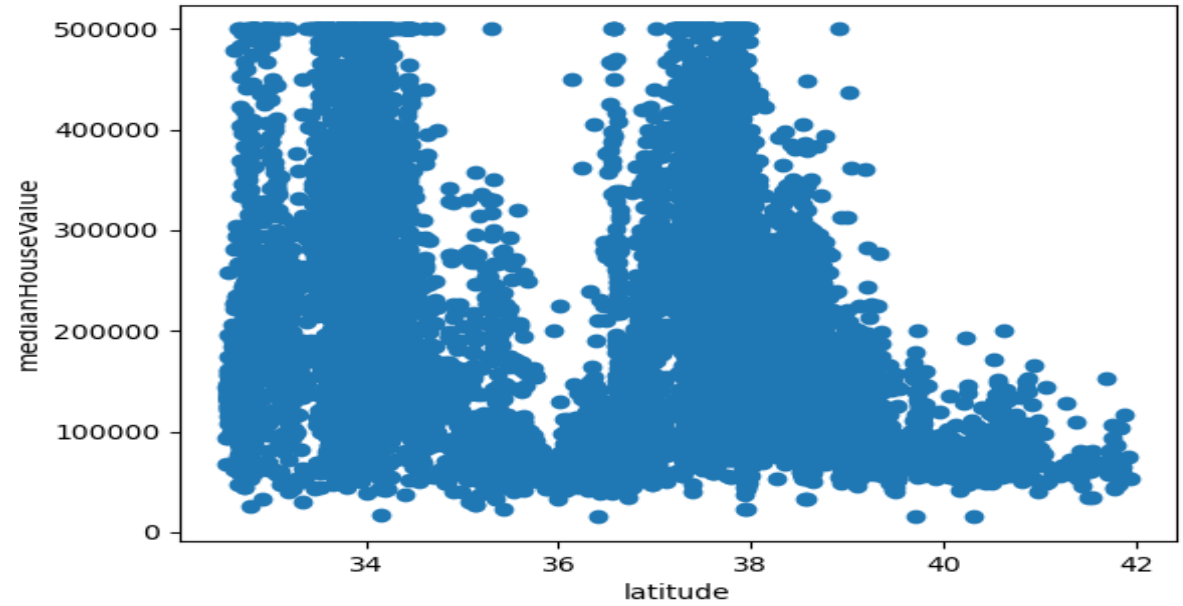

Figura 2: *medianHouseValue con latitud*

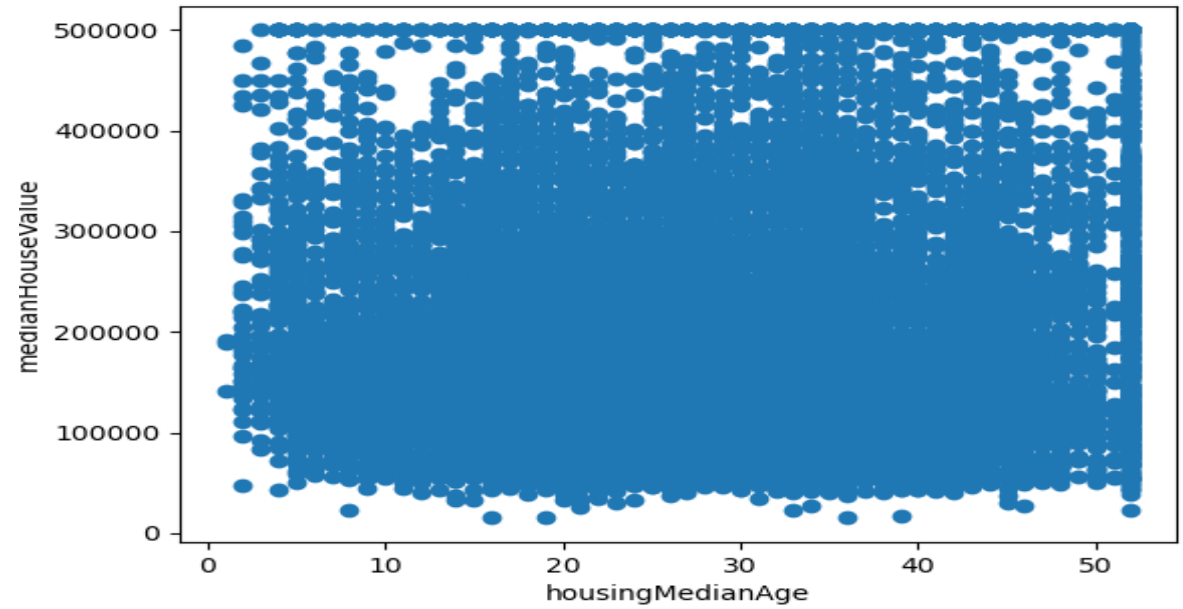

Figura 3: *medianHouseValue con housingMedianAge*

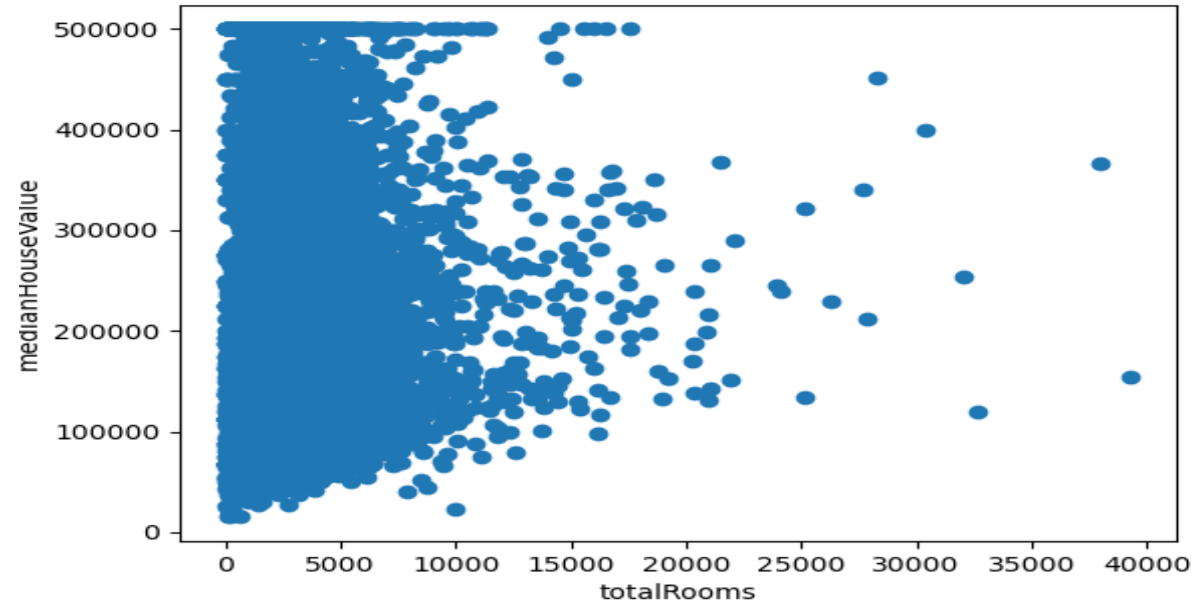

Figura 4: *medianHouseValue con totalRooms*

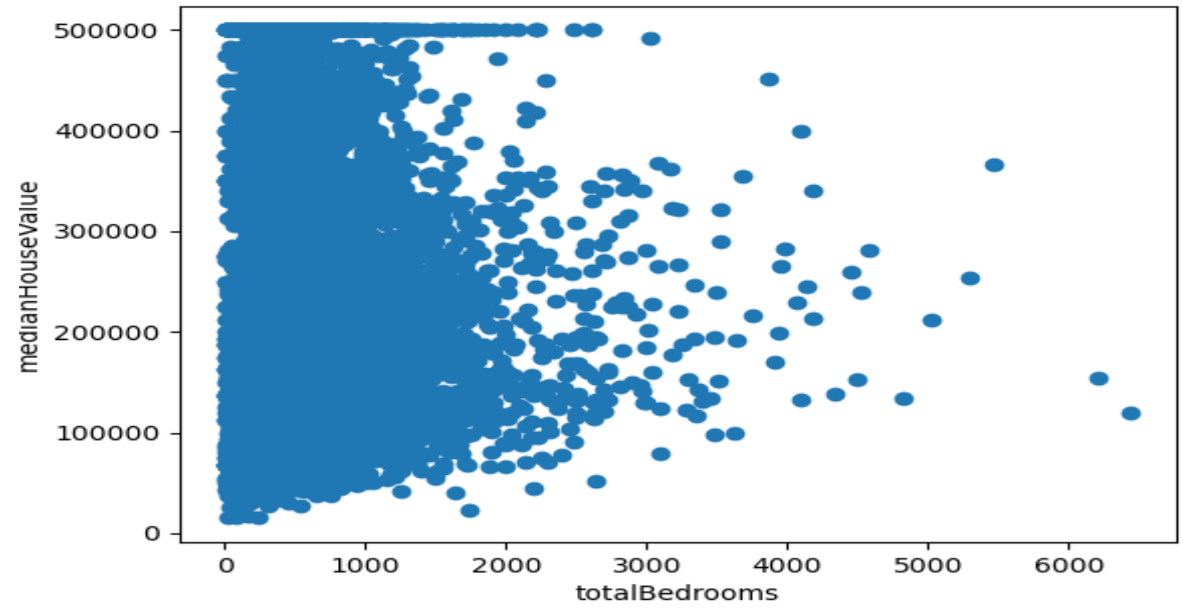

Figura 5: *medianHouseValue con totalBedrooms*

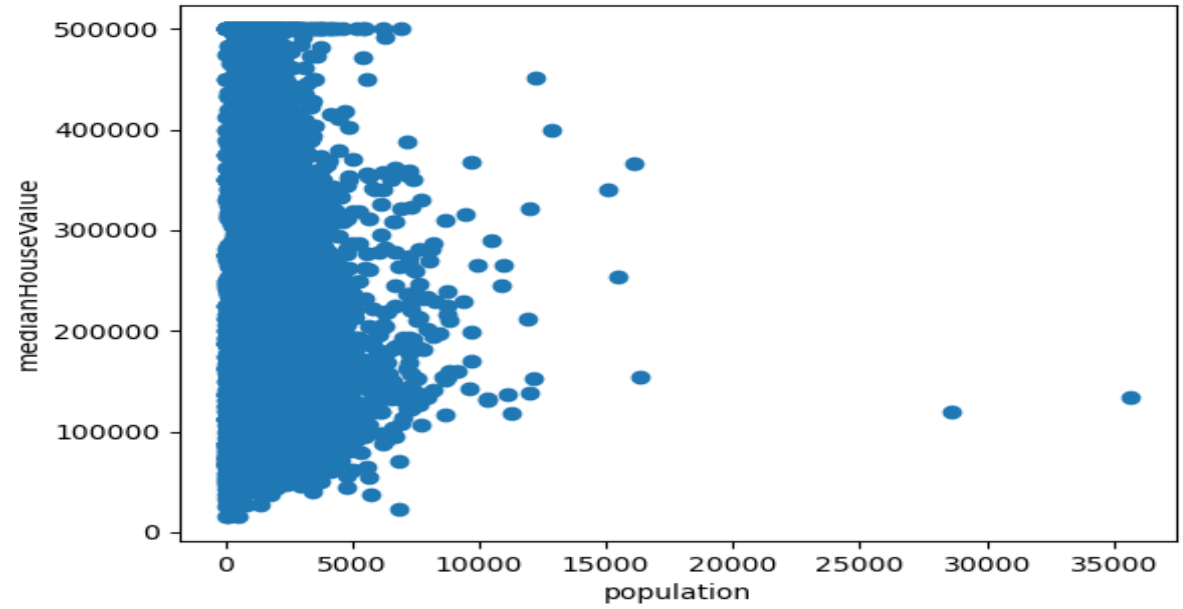

Figura 6: *medianHouseValue con population*

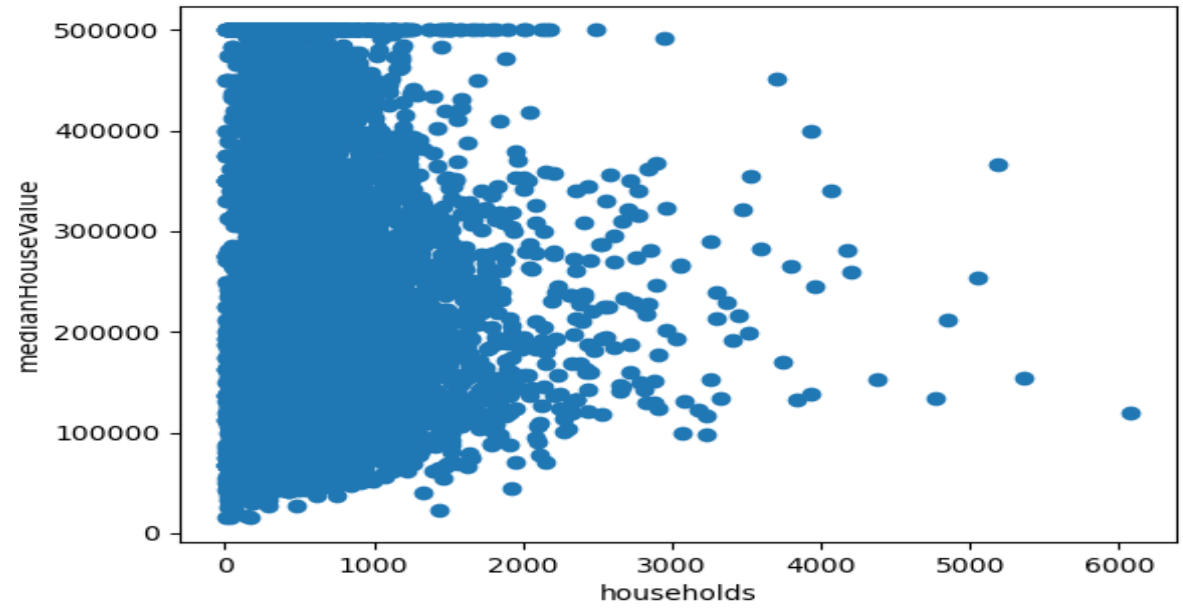

Figura 7: *medianHouseValue con households*

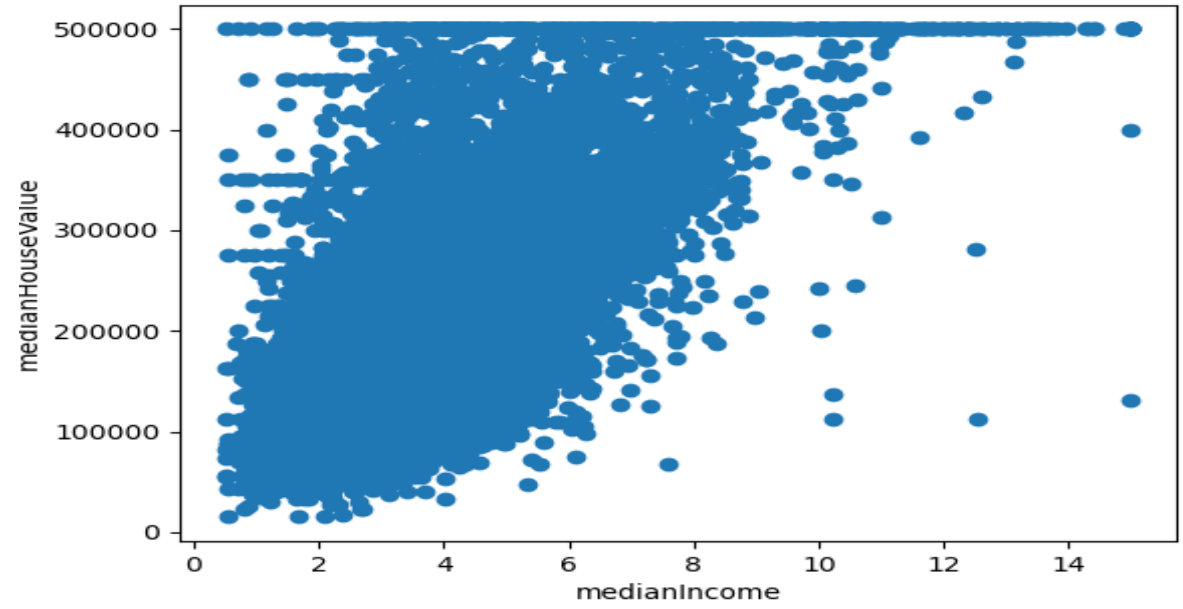

Figura 8: *medianHouseValue con medianIncome*

Estás gráficas permiten a los tomadores de decisiones ver la analítica presentada de forma visual, de modo que puedan captar conceptos difíciles o identificar nuevos patrones.

Pasaremos la Mapa de Correlación de Calor, que nos mostrará todas las variables comparadas en una misma imágen.

2.5. Matriz de correlación de calor.

Es una tabla que indica los coeficientes de conexión entre los factores.Nos ayudará a darnos cuenta que variables son las acertadas para nuestro análisis. Al facilitar datos de forma visual y sencillos de comprender, nos permitirán tomar decisiones con rapidez.

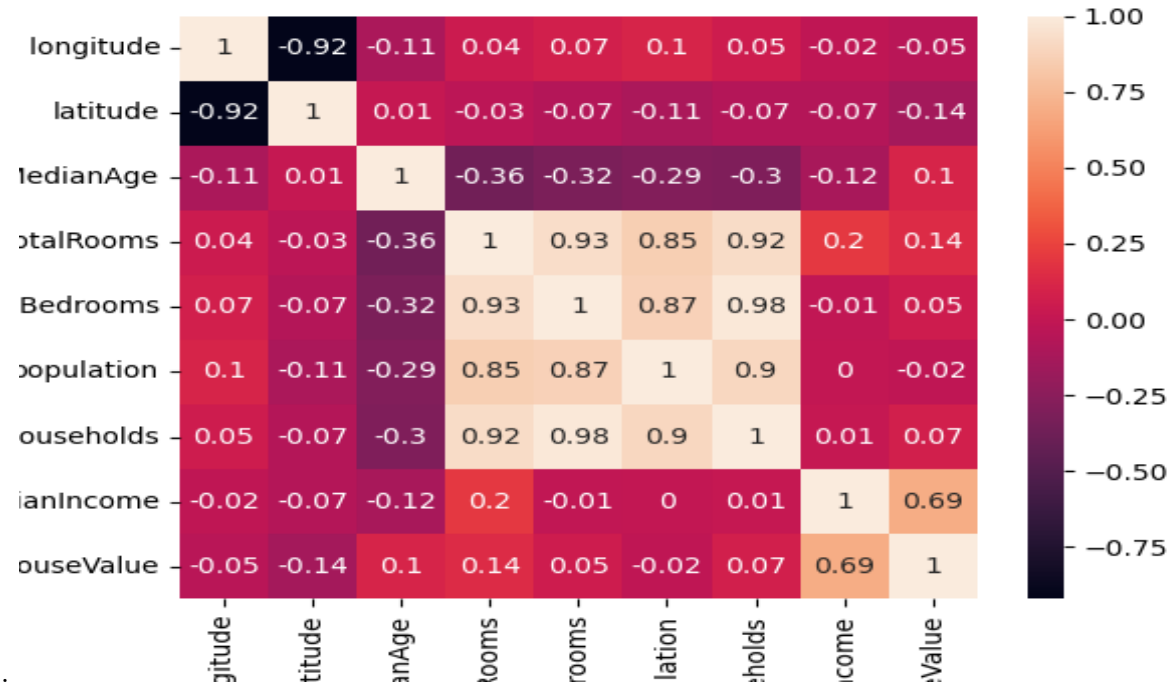

Figura 9: *Mapa de correlación de calor o heatMap.*

Aquí podemos apreciar que las mejores variables a tomar con respecto a la que queremos predecir son: medianIncome con 0.69, totalRooms con 0.14 y houseHolds con 0.07.

2.6. Instrucciones de prueba

Dadas las instrucciones para la práctica, a continuación mostraremos todos los resultados obtenidos en cada una de ellas por medio de una tabla para poder visualizarlos y compararlos mejor.

3. Reporte de Resultados.

Tipo de Regresión	Tipo de Escalamiento	Variables Seleccionadas	Learning Rate	Número de Iteraciones	MSE promedio	R^2
SGD	Ninguno	1	$1x10^{-6}$	100000	7096088359.8891	0.4715
SGD Grado 2	Ninguno	1	$1x10^{-6}$	100000	7061474976.000686	0.4741
SGD Grado 3	Ninguno	1	$1x10^{-6}$	100000	9268808875.580397	0.3097
SGD	Estándar	1	$1x10^{-6}$	100000	7095233071.588557	0.4716
SGD Grado 2	Estándar	1	$1x10^{-6}$	100000	7043631417.174315	0.4754
SGD Grado 3	Estándar	1	$1x10^{-6}$	100000	6941726671.739191	0.4830
SGD	Robusto	1	$1x10^{-6}$	100000	7095233121.698646	0.4716
SGD Grado 2	Robusto	1	$1x10^{-6}$	100000	7043668878.897501	0.4754
SGD Grado 3	Robusto	1	$1x10^{-6}$	100000	6949556846.417852	0.4824
SGD	Ninguno	2	$1x10^{-6}$	100000	1.3395548335393376e+26	-9982
SGD Grado 2	Ninguno	2	$1x10^{-6}$	100000	2.3823746405250352e+42	-1.77
SGD Grado 3	Ninguno	2	$1x10^{-6}$	100000	9.567335190148745e+55	-7.0718
SGD	Estándar	2	$1x10^{-6}$	100000	7095035679.103306	0.4716
SGD Grado 2	Estándar	2	$1x10^{-6}$	100000	7022694456.216341	0.4770
SGD Grado 3	Estándar	2	$1x10^{-6}$	100000	6925333397.315249	0.4842
SGD	Robusto	2	$1x10^{-6}$	100000	7095036550.247849	0.4716
SGD Grado 2	Robusto	2	$1x10^{-6}$	100000	7026870786.548138	0.4767
SGD Grado 3	Robusto	2	$1x10^{-6}$	100000	7261577858.958232	0.4592
SGD	Ninguno	3	$1x10^{-6}$	100000	$2.355x10^{26}$	$-1.763x10^{16}$
SGD Grado 2	Ninguno	3	$1x10^{-6}$	100000	$4.331x10^{42}$	$-3.247x10^{32}$
SGD Grado 3	Ninguno	3	$1x10^{-6}$	100000	$7.518x10^{55}$	$-5.613x10^{45}$
SGD	Estándar	3	$1x10^{-6}$	100000	6480728221.923142	0.5139
SGD Grado 2	Estándar	3	$1x10^{-6}$	100000	6181467171.920125	0.5363
SGD Grado 3	Estándar	3	$1x10^{-6}$	100000	5951832135.885584	0.5535
SGD	Robusto	3	$1x10^{-6}$	100000	6480731101.2866955	0.5139
SGD Grado 2	Robusto	3	$1x10^{-6}$	100000	6192177691.24812	0.5355
SGD Grado 3	Robusto	3	$1x10^{-6}$	100000	6495628525.119458	0.5127
SGD	Ninguno	Todas	$1x10^{-6}$	100000	$2.423x10^{27}$	$-1.816x10^{17}$
SGD Grado 2	Ninguno	Todas	$1x10^{-6}$	100000	$1.005x10^{44}$	$-7.564x10^{33}$
SGD Grado 3	Ninguno	Todas	$1x10^{-9}$	100000	$3.084x10^{56}$	$-2.32x10^{46}$
SGD	Estándar	Todas	$1x10^{-6}$	100000	4837249822.64996	0.6371
SGD Grado 2	Estándar	Todas	$1x10^{-6}$	100000	4363323974.984737	0.6727
SGD Grado 3	Estándar	Todas	$1x10^{-6}$	100000	4029883934.5270033	0.7002
SGD	Robusto	Todas	$1x10^{-6}$	100000	4838713361.729899	0.6370
SGD Grado 2	Robusto	Todas	$1x10^{-6}$	100000	4794303242.929585	0.6403
SGD Grado 3	Robusto	Todas	$1x10^{-6}$	100000	$1.4145x10^{23}$	$-1.06x10^{13}$

Tabla 1: Modelos de Prueba

3.1. Reporte de la Mejor Regresión

Usando el conjunto de entrenamiento que consta del 80 % de los datos, entrenaremos un modelo tomando la mejor regresión lineal obtenida con los pliegues. En este caso será la regresión lineal estocastica de gradiente descendiente con polinomio de grado 3, escalamiento estándar y el uso de todas las variables del conjunto de datos. [10].

```python
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import PolynomialFeatures
import pandas as pd
from sklearn.linear_model import SGDRegressor
from  sklearn import preprocessing

x_train = pd.read_csv('data_train.csv')
x_test = pd.read_csv('data_test.csv')
y_train = pd.read_csv('medianHouseValue_train.csv')
y_test = pd.read_csv('medianHouseValue_test.csv')

regr = SGDRegressor(learning_rate = 'constant', eta0 = 1*(10)**-6, max_iter= 100000)
polynomial_features= PolynomialFeatures(degree=3)
x_poly_train = polynomial_features.fit_transform(x_train)
x_poly_test = polynomial_features.fit_transform(x_test)
x_poly_train_standard_scaler =
 preprocessing.StandardScaler().fit_transform(x_poly_train)
x_poly_test_standard_scaler = preprocessing.StandardScaler().fit_transform(x_poly_test)
regr.fit(x_poly_train_standard_scaler, y_train)
y_poly_pred = regr.predict(x_poly_test_standard_scaler)
mse = mean_squared_error(y_test, y_poly_pred)
r2 = r2_score(y_test, y_poly_pred)
print ('Regresión polinomial estocástico grado 3 con escalado de datos Standard\n
mse: {} r2: {}'.format(mse, r2))
```

.

Figura 10: *Código de la regresión lineal.*

El resultado obtenido del modelo de regresión lineal es:

Median Squared Error	$\mathbf{R^2}$
4004136703.716517	0.6955

Tabla 2: Mejor resultado.

4. Conclusiones.

General: Con esta práctica pusimos a prueba nuestros conocimientos con los temas vistos en un solo dataset. Entendimos la importancia y las diferencias entre cada una de las aplicaciones hechas, como: regresión lineal mediante gradiente descendiente estocástico, regresión polinomial mediante gradiente descendiente estocástico, escalado estándar, escalado robusto.

Complicaciones al realizar la práctica: La parte que nos costo más trabajo como equipo fue delimitar los pliegues y tambien el tiempo de ejecución del entrenamiento de los modelos.

Ideas para mejorar los resultados obtenidos: Pensamos que para poder mejorar los resultados obtenidos, podríamos aplicar algun tipo de técnica a las variables correlacionadas entre sí para mejorar el rendimiento con el target.

Referencias

[Abu-Mostafa] Abu-Mostafa, Y. S. Learning From Data: A Short Course. Recuperado de: `https://work.caltech.edu/lectures.html`. Fecha de consulta: 20/03/2022.

[Scikit-Learn] Scikit-Learn. Linear model: Api reference. `https://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model`. Fecha de consulta: 20/03/2022.