DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Practico 1

Regresión

Reconocimiento de Patrones

Integrante	LU	Correo electrónico
Pablo Lopes Perera	07/18	plopesperera99@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
https://exactas.uba.ar

Resumen

En el presente informe se presentan los mejores hiperparametros obtenidos para un problema de regresión basados en un conjunto de datos de propiedades en el estado de California, EEUU. Se probaron los métodos de regresión con regularizaciones *Ridge*, *Lasso* y *ElasticNet*, aplicando cross-validation. Los resultados obtenidos muestran que las regularizaciones no mejoran significativamente los resultados.

Índice

1. Introducción

En el presente trabajo practico se utilizo el data set *California Housing* (Statlib repository). Este dataset esta basado sobre datos del censo de 1990 de California. El dataset incluye metricas tales como poblacion, ingreso promedio, precio promedio de las propiedades, etc. para cada "*Block Group*". Un "*Block Group*" es la unidad mas pequeña para la cual la Oficina de Censos de Estados Unidos de America publica datos. Desde este punto los llamaremos "distritos"

Problema El problema presentado fue predecir el valor promedio por distrito utilizando regresiones lineales con los metodos de *Ridge*, *Lasso* y *ElasticNet*.

En el mismo se presentaran mas adelante los resultados, pero primero se vera el desarrollo del trabajo.

2. Desarrollo

2.1. Entorno y datos

El trabajo fue desarrollado en el lenguaje *Python* sobre un *Jupyter-Notebook*. También fueron utilizados los módulos *Pandas*, *ScikitLearn*, *Numpy* y *Matplotlib* (para graficos).

Los datos fueron obtenidos desde un repositorio (del autor) en Github [1].

2.2. Preprocesamiento

Los datos obtenidos están compuestos por 10 columnas. Estas variables son

- longitude
- latitude
- housing_median_age
- total_rooms
- total_bedrooms
- population
- households
- median_income
- median_house_value
- ocean_proximity

En la Figura 1 se muestra un sample de los datos originales.

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
9383	-122.50	37.91	31.0	7001.0	1282.0	2755.0	1267.0	5.4851	441100.0	NEAR BAY
16986	-122.26	37.55	17.0	4576.0	814.0	1941.0	807.0	5.9572	443800.0	NEAR BAY
8332	-118.30	33.93	40.0	2434.0	477.0	1646.0	453.0	3.2024	128000.0	<1H OCEAN
5265	-118.54	34.05	33.0	6778.0	1092.0	2540.0	1052.0	8.5650	500001.0	<1H OCEAN
6603	-118.16	34.19	44.0	2195.0	449.0	1377.0	417.0	3.5887	153500.0	<1H OCEAN

Figura 1: Una muestra aleatoria de los datos

[1] https://raw.githubusercontent.com/ageron/handson-ml2/master/datasets/housing/housing.tgz

Graficando los datos sobre un mapa, obtenemos la Figura 2, el mismo esta coloreado en base al valor medio de las propiedades, y cada burbuja equivale al tamaño de la población en dicho distrito.

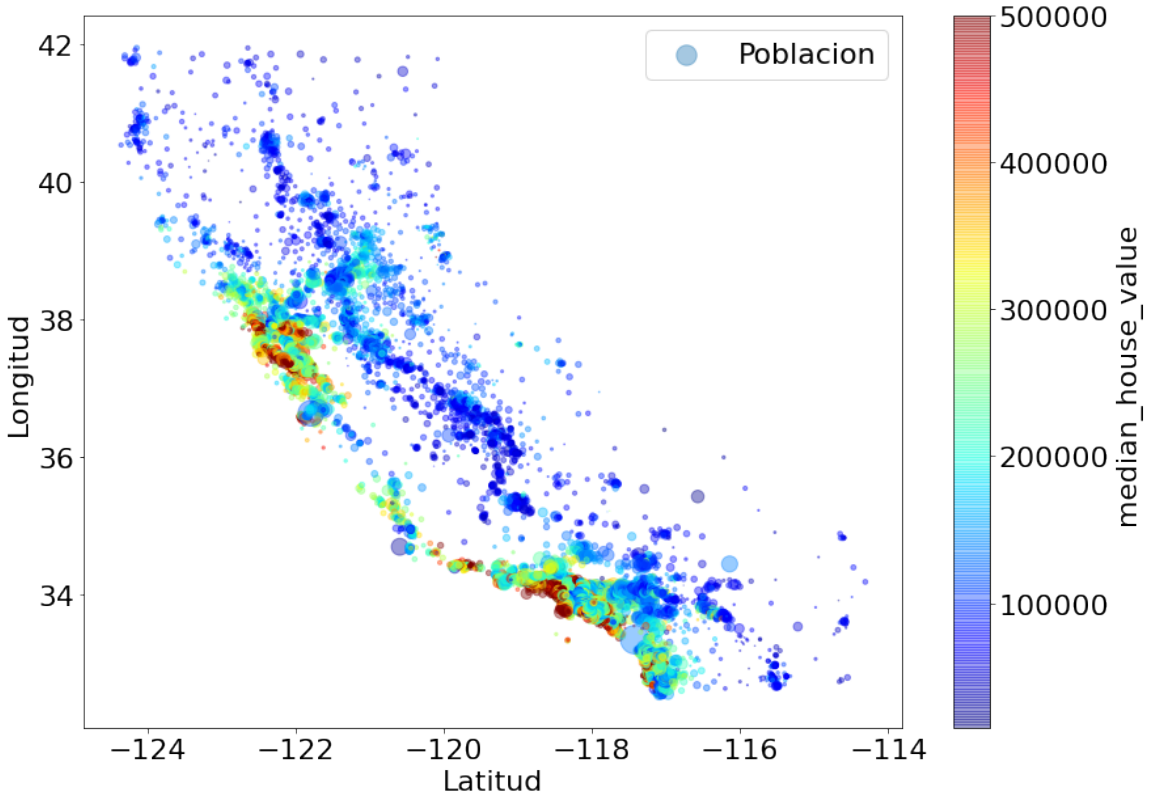

Figura 2: Mapa del dataset.

2.3. Enriquecimiento de datos

Antes de comenzar con la selección de modelos, se realizo un proceso de enriquecimiento de datos. Viendo la Figura 3 se procedió a crear 4 variables adicionales.

Las variables creadas fueron

1. **rooms_per_household** : Cantidad de habitaciones por division familiar (casa).
2. **bedrooms_to_rooms_ratio** : Ratio de dormitorios a habitaciones.
3. **population_per_household** : Población por hogar.
4. **income_category** : Es una categorización de los ingresos.

Para la división de los datos de entrenamiento y testing se aplico una división estratificada de los datos (*StratifiedShuffleSplit*), basada en la columna *income_category*. La relación es de 0,8 de entrenamiento y 0,2 de testing. Luego, para los datos ausentes, se aplico la estrategia de rellenado por la media de los valores, es decir, para las columnas en los casos donde no había datos, se llenaban

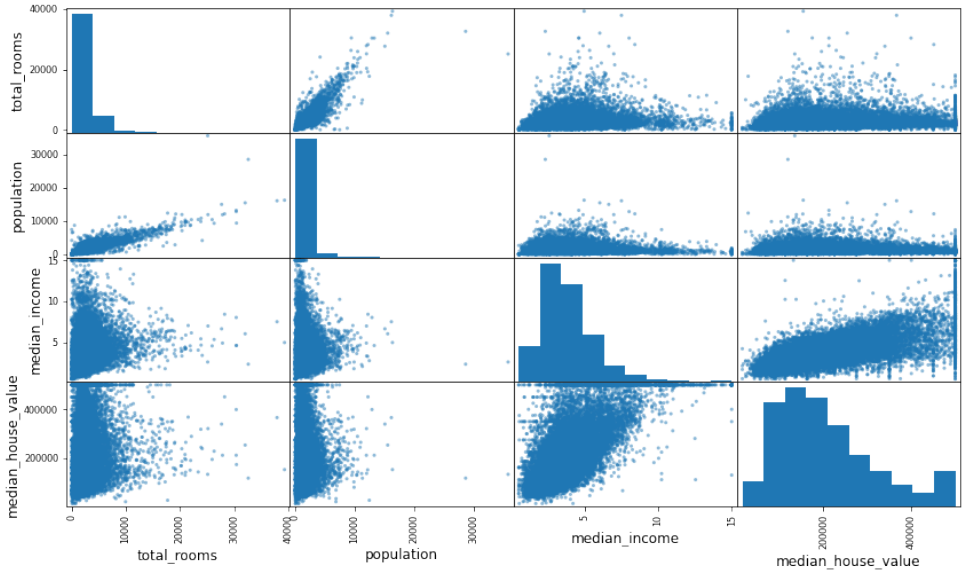

Figura 3: Matriz de correlación entre 4 variables (total_rooms, population, median_income y median_house_value)

con el calculo del promedio de la columna (para los datos presentes). Luego para re-escalar los datos se utilizo un escalador estándar (*StandarScaler*) escalando por el desvío estándar.

Finalmente se aplicaron sobre el dataset todos los pasos previamente mencionados en un pipeline que transforma los datos y los deja listos para los entrenamientos.

2.4. Modelos y métodos

Los métodos aplicados para el presente informe fueron *Ridge*, *Lasso* y *ElasticNet*. Para mejorar la performance de los mismo se utilizo cross-validation. Yendo a temas mas implementativos, se utilizaron los siguientes modelos de la libreria *Scikit-Learn*: *RidgeCV*, *LassoCV*, *ElasticNetCV*. Cada uno es la implementación de cada método con built-in cross-validation.

A continuación se explicaran brevemente cada método.

Ridge es regularizador aplicado en regresiones lineales, sumándoles un termino regularizador igual a $\alpha \sum_{i=1}^{n} \theta_i^2$. Esto fuerza al modelo a crecer pero manteniendo los pesos lo mas pequeños posible. Ridge tiene como parámetro el α

Lasso es similar a Ridge, ya que se le agrega un termino regularizador a una regresión lineal. El termino regularizador en cuestión es $\alpha \sum_{i=1}^{n} |\theta_i|$. Lasso tiene como parámetro el α

ElasticNet es una combinación de los dos regularizadores anteriores. Utiliza el parámetro r, el cual mezcla los pesos entre Lasso y Ridge. En el caso que $r = 0$, este caso equivale a utilizar Ridge, y si $r = 1$ es Lasso. El regularizador en cuestión es $r\alpha \sum_{i=1}^{n} |\theta_i| + \frac{1-r}{2} \alpha \sum_{i=1}^{n} \theta_i^2$

El esquema de *cross-validation* aplicado es **K-Folds**, en todos los casos se aplicaron 10 folds.

3. Resultados

A continuación se expondrán los parámetros utilizados en cada método.

- $\alpha's$ = [0,1 0,325 0,55 0,775 1,0 2,0 10 26,5 51,0 75,5 100,0]
- *max_iterations*= 10000 (Solo para Lasso y ElasticNet)
- *l1_ratio*:
  r = [0,052 0,105 0,157 0,210 0,263 0,315 0,368 0,421 0,473
  0,526 0,578 0,631 0,684 0,736 0,789 0,842 0,894 0,947]

Y se utilizo en común el parámetro $CV = 10$ que son los 10 folds utilizados en la *cross-validation*.

Luego, para evaluar los modelos se ejecuto la prediccion de los datos de entrenamiento (para ver la performance en training) y en los datos de testing (para ver la performance en el test set). Luego se obtuvo el $RMSE$ (*Root Mean Square Error*).

Los resultados obtenidos obtenidos para los datos de entrenamiento se muestran en la Tabla , y en la Tabla 2 los de testing.

Model	**Best param**	**RMSE**
Ridge	$\alpha = 51{,}00$	67825,93
Lasso	$\alpha = 100{,}00$	67818,83
ElasticNet	$\alpha = 0{,}10$ y $r = 0{,}96$	67834,68

Cuadro 1: Resultados obtenidos en el *Training Set*

Model	**RMSE**
Ridge	66798,37
Lasso	66781,01
ElasticNet	66807,11

Cuadro 2: Resultados obtenidos en el *Testing Set* con los parametros obtenidos anteriormente

Los resultados obtenidos sugieren que un modelo lineal, con regularizaciones, no debe ser el indicado. Ya que en los casos de Ridge y Lasso, esta utilizando α's muy altos, esto produce que el termino de regularización produzca una penalización muy fuerte. Y además, podemos ver como ElasticNet esta dándole mucho peso a la regularización Ridge, ya que el parámetro r es pequeño.

De todas formas, los tres modelos parecen performar similar, ya que los tres tienen un valor $RMSE$ similar, tanto en los casos de entrenamiento como testing. Pero si habria que seleccionar un modelo, el que mejor valores tiene de los tres es Lasso, ya que el de menor $RMSE$ en el conjunto de testing.

Next-Steps Seria interesante utilizar el mismo dataset y aplicar otras regresiones, sean lineales, o no lineales.

4. Conclusiones

Los modelos son creados con exito, pero en este caso, con este dataset, los tres modelos performan igual. Son practicamente indistiguibles entre si ya que el $RMSE$ de los tres es muy similar.

De todas formas, estos valores obtenidos son similares a los obtenidos por una regresion lineal sin regularizaciones (1).

Referencias

[1] Aurélien Géron. *Hands-on Machine Learning with Scikit-Lean, Keras and TensorFlow 2nd Edition.* O'Reilly, 2019.

[2] Jason Brownlee. *Machine Learning Mastery with Python Understand Your Data, Create Accurate Models and Work Projects End-To-End.* Machine Learning Mastery, 2016.