FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico 2
Machine Learning

[75.06/95.58] Organización de Datos
Segundo cuatrimestre de 2019
Grupo 40

Nombre	Padrón	E-mail
DAVEREDE, Agustin	98.540	agusdaverede@yahoo.com.ar

Repositorio: https://github.com/Distance71/TP-OrgaDatos

Índice

1. Resumen

En el presente trabajo se realizó un análisis predictivo basado en técnicas de *Machine Learning*, a partir de datos referidos a publicaciones realizadas en la página 'ZonaProp' en el periodo de 2012 a 2016 en México.

En primer lugar se evaluó el set de datos provisto, cuyo precio se debería predecir, ponderandolo con el análisis efectuado en la primera entrega. Posteriormente, se realizó un preprocesamiento de los datos provistos por el set de entrenamiento utilizando diversas técnicas.

Finalmente, se llevaron a cabo predicciones basadas en los algoritmos de *Machine Learning* seleccionados.

2. Introducción

En el presente trabajo se buscó determinar cual era el precio de un determinado grupo de propiedades con base en sus características, en adelante *features*. Para lograrlo, se desarrollo un modelo, con base en diversos y múltiples métodos, partiendo de un set de datos, en adelante denominado set de entrenamiento, cuya información permitiría a un algoritmo llevar a cabo dicha predicción.

Entre los algoritmos que se decidieron utilizar para efectuar la tarea, se destacan, primero, algoritmos simples basados en la regresión lineal, a partir de un solo *feature -LinearRegressor-*. Además, por otro lado, se pueden destacar algoritmos más complejos empleados como *Decision Tree*, *Lasso*, *ElasticNet*, *Ridge Regression*, *SVR*, *SGD Regressor*, *XGG Boost* y por último *Random Forest*. El análisis puntual de cada uno se encontrará en los siguientes apartados.

3. Análisis preliminar

En un comienzo, a la hora de aplicar técnicas de *Machine Learning*, se debe efectuar un procesamiento preliminar de los sets de datos con los que se cuenta. Esto se hace con el objetivo de facilitar la tarea del algoritmo, cuya entrada serán dichos datos. A tales efectos, se hace uso de diversos métodos y criterios, que se describirán a continuación.

En primer lugar, y esto surge del análisis efectuado en el primer trabajo practico, resultó que no será de relevancia considerar los siguientes ítems:

- Id
- Título
- Descripción
- Dirección
- Idzona
- Lat
- Lng

La razón es que sobre los mismos no recae información que deseemos considerar relevante en un principio. Si bien es cierto que podría utilizarse como un contraste realmente valioso de los *features* que si se desean emplear. Por ejemplo, la dirección podría ser empleada para determinar, en conjunto con latitud y longitud, o bien por si misma, si la información con la que se cuenta de provincia y ciudad para una determinada propiedad son fidedignas. Sin embargo,

estos análisis de mayor profundidad, requieren de ciertos conocimientos que no son el principal abordamiento de la asignatura, y cuya puesta en practica requeriría de una carga de trabajo sustancial, y el posible uso de fuentes de datos externas. Otro ejemplo que parece interesante mencionar, es la incidencia que podría llegar a tener el uso de determinadas palabras o conjuntos de palabras sobre el consumidor, haciendo referencia al título y la descripción. De igual modo que el ejemplo previo, su utilización y puesta en valor supondría una tarea de interés si se pensará en fines comerciales.

3.1. Análisis de relación entre sets de datos

Lo siguiente que se efectuó fue un análisis sobre el grado de utilidad y correlación que tendría el set de datos de entrenamiento respecto del set de evaluación. Con este fin, se asumió el uso de dos parámetros de referencia. En primer lugar, se consideró un parámetro que representaría la sumatoría de diferencias:

$$A_k = \sum_{i=1}^{nvalues} |(x_i/x_t) - (y_i/y_t)| * 100 \quad (1)$$

Donde x_i e y_i representan la aparición de esos valores únicos dentro de cada respectivo set de datos, siendo indistinto. A su vez, x_t e y_t representan el total de datos, que son 240k y 60k.

Por otro lado, se considero un parámetro que representa la relación entre los promedios de los datos de cada set:

$$B_k = ((|x_{mean}/y_{mean}|) - 1) * 100 \quad (2)$$

Se obtuvieron los siguientes datos:

Categoría	Relación de items [A_k]	Relación de medias [B_k]
Tipo de propiedad	0,669	-
Ciudad	4,557	-
Provincia	2,001	-
Antiguedad	0,450	0,369
Habitaciones	0,278	0,171
Garages	0,216	0,047
Baños	0,450	0,191
Metros cubiertos	-	0,154
Metros totales	-	0,206
Fecha por año	0,551	4.3e-05
Gimnasio	0,235	1,88
Usos multiples	0,088	0,802
Piscina	0,177	1,011
Escuelas cercanas	0,365	0,411
Centros comerciales cercanos	0,447	0,563

Tabla de parámetros

De esto es posible deducir que los sets de datos de entrenamiento y evaluación comparten una naturaleza, en lo relativo a *features* en cierto modo similar, ya que el promedio de los datos resultaba similar y la otra medida que nos daba una referencia más puntual también resultó similar, siendo la excepción más notable el caso de provincia y ciudad.

Además, como puede observarse la tabla está incompleta. La razón de esto es que si bien es posible calcular el parámetro A_k para metros cubiertos y totales, dicha medición no tendría

valor, ya que se trata de *features* de tipo *float*. En lo que respecta al item B_k de los *features* Tipo de propiedad, Ciudad y Provincia, hay una razón por la que el mismo no fue calculado y es que al momento de efectuar este análisis no se habían empleado técnicas como *one-hot encoding*, y los resultados obtenidos se consideraron suficientes para proseguir.

Por último, se incluirá una tabla que describe el set de datos de evaluación, que nos dará la pauta que podemos adentrarnos en efectuar dicha tarea, ya que contamos, al menos, con los Id.

Parámetro	Aparición [%]	Cantidad de diferentes
Id	100	60000
Tipo de propiedad	99,99	22
Ciudad	99,86	576
Provincia	99,93	32
Antiguedad	82,14	72
Habitaciones	90,62	10
Garages	84,46	4
Baños	89,07	4
Metros cubiertos	92,84	424
Metros totales	78,91	424
Fecha	100	1828
Gimnasio	100	2
Usos multiples	100	2
Piscina	100	2
Escuelas cercanas	100	2
Centros comerciales cercanos	100	2

Datos set de evaluación

Con base en lo explicitado, se asumirá que es posible utilizar el set de entrenamiento como base para el algoritmo que nos permitirá realizar las predicciones sobre el set de evaluación. El siguiente paso fue efectuar una adaptación de los datos.

3.2. Adaptación de los datos

Llegado a este punto, nos encontramos en posición de poder comenzar a trabajar con los datos. Esta tarea se dividirá, en primer lugar, en la creación de nuevos *features*, que permitan establecer mayores relaciones entre los datos suministrados y nos lleven a una mejor predicción. Luego, además, se encuentra la transformación de los datos con que ya se cuenta, de carácter categórico o texto, en un tipo que puedan interpretar los algoritmos basados en cálculo numérico -lo deseado en este trabajo-.

3.2.1. Creación de nuevos *features*

En este apartado es de destacar que si bien consideramos una buena práctica el aprovechamiento de los *features* como herramienta para permitir a los algoritmos una mayor profundización en menos tiempo, no se ha encontrado algo realmente destacable. Del trabajo practico Nº1 surge información que podría ser relevante para un análisis, pero mayormente, al menos a criterio de quien suscribe, orientado a un algoritmo de clasificación. Hay excepciones a esto, como son el año y el mes de publicación de una propiedad, pero como se observó, esto no constituye en un parámetro que incida mayormente sobre el precio de una propiedad. Esta posibilidad queda abierta y se comenta como posible mejora en el futuro.

3.2.2. Transformación de datos de tipo categoría

Dada la naturaleza de los datos que pueden interpretar los algoritmos que se describan luego, es necesario efectuar transformaciones sobre los datos que no poseen ese carácter a fin de lograr poder utilizar esa información. Es por esto, que se ha optado por utilizar dos técnicas que nos facilitan la tarea. datos La primera es denominada *one-hot encoding*, y se basa en la creación de nuevas columnas en el set de datos a partir de los datos únicos que se posean en la categoría, y la colocación de un dato de tipo binario en cada celda, según corresponda esa fila o no a dicha categoría. Además, por otro lado, se ha decidido utilizar otra técnica que se conoce como *label encoding*, que funciona de un modo similar, pero en lugar de crear una nueva columna por cada dato único de la esa categoría, transforma cada línea única de texto en un número y las reemplaza.

Esto concluye el análisis y tratamiento preliminar de los datos.

4. Análisis de la solución

En lo referido a la solución, se buscó implementar una predicción que minimizara el error respecto de los datos reales. Esto se trató de lograr partiendo de la premisa de la utilización de más de un algoritmo, sin mayor profundización en un principio, para luego, con variaciones de los datos suministrados -parámetros del algoritmo- (mediante la utilización de técnicas de *one-hot encoding* y *label encoding*), lograr mejores resultados. Previo al detalle de cada uno, se expondrán técnicas y criterio de uso general para la resolución del problema.

4.1. Técnicas empleadas

4.1.1. Boosting

Esta técnica consiste en la utilización de más de un algoritmo en simultaneo para buscar lograr una mayor *performance* final con base en el trabajo en conjunto.

Lo cierto es que, por una cuestión de capacidad de trabajo y tiempo, no se llego a lograr implementar una solución que resultara conveniente, a partir de esta técnica. Solo se hicieron uso de los algoritmos de *XGB Boost* y *Random Forest* que emplean esta técnica. Luego, es una tarea que quedará pendiente de evaluación en posibles mejoras.

4.1.2. Cross-Validation

Consiste en la utilización un set de datos menor al de entrenamiento para llevar a cabo la etapa de entrenamiento del algoritmo, y la posterior validación del modelo utilizando el conjunto de datos restantes del mismo. Esto se hace con el objetivo de llegar a validar el modelo empleado. Además, una extensión interesante de este concepto resulta ser el la escalabilidad de un modelo basado en datos iniciales y su posterior incrementación.

Esta practica fue llevada a cabo.

4.2. Algoritmos empleados

Como se mencionó en apartados previos, se hizo uso de diversos algoritmos con su estructura e hiper parámetros por defecto. A su vez, se indago con mayor profundidad en aquellos cuyos resultados fueron de mayor conveniencia. Los mismos serán descriptos con mayor detalle en este apartado.

4.2.1. Linear Regression

La regresión lineal simple consiste en la generación de un modelo, a partir de los datos que mejor ajustan a una recta, con base en los datos suministrados en el set de entrenamiento. Una vez constituido el modelo, solo se predice cual es el valor que mejor ajustará a dicha recta con el nuevo punto. Este modelo es muy simple y no permite contemplar todos los *features* que se poseen. Luego, solo nos sirve como una primera aproximación que no tendrá más valor salvo como caso de prueba.

4.2.2. Decision Tree Regression

Se trata de un modelo no paramétrico y supervisado que se encarga de predecir el valor objetivo a través de un algoritmo basado en la decisión de reglas simples basadas en las *features* del set de datos.

Su fortaleza radica en su simplicidad, en que se requieren pocos datos para llegar a entrenar al algoritmo y en que permite la posibilidad de poder analizar datos de distinto tipo (incluyendo datos de tipo categoría), característica poco frecuente en este tipo de algoritmos. Por otro lado, su fortaleza es también su debilidad, ya que una variación mínima en el set de datos puede ocasionar que se produzcan grandes variaciones en el resultado obtenido. Otra desventaja resulta ser su capacidad de representar adecuadamente la información suministrada por el set de datos. Esto se conoce como *overfitting*.

4.2.3. Lasso y Elastic Net Regressor

Se llegó a este algoritmo consultando diversas posibilidades en la web. Se trata de modelos *sparse* caracterizados por el concepto de proyección de una matriz, con el calculo progresivo de ciertos coeficientes. No se indago más debido a que los resultados no fueron de interés.

4.2.4. Ridge Regressor

Se trata de un algoritmo que busca minimizar la suma de los cuadrados residuales -concepto matemático- mediante la imposición de una corrección por parámetro, buscando aproximarse a los datos. Su utilización no resultó de utilidad.

4.2.5. SGD Regressor

Este algoritmo en principio revistió un gran interés, ya que es de los pocos suministrados por la biblioteca *scikit* que posee la capacidad de ser incremental. Es decir, se pueden ir agregando datos al modelo una vez entrenado, y este llevará a cabo un reajuste. Esta característica es conocida como *feature scaling*. Su principal fortaleza, además, reviste en la eficiencia, lo que permite utilizarlo en grandes conjunto de datos, del orden de 10^5 -según documentación-. Por otro lado, la desventaja que posee es la dificultad que conlleva fijar los hiper parámetros que el mismo requiere para lograr trabajar con exactitud.

Se hicieron dos pruebas con el mismo, logrando una significa mejora en la segunda evaluación de Kaggle, pero aún así no fue el mejor resultado.

4.2.6. XGB Boost Regressor

Se seleccionó este algoritmo con base en el consejo de un compañero que cursó la asignatura en un cuatrimestre previo, ya que el mismo posee una alta aceptación en el entorno de *Machine Learning*.

Su principal fortaleza radica en la eficiencia, originalmente escrito en C++ y multi-core, su adaptabilidad y el hecho de que es posible personalizar su funcionamiento de un modo muy

completo con base en diversos parámetros internos. Su utilización arrojó resultados interesantes, y sería recomendable su profundización en un próximo trabajo.

4.2.7. SVR Regressor

Este algoritmo fue seleccionado con base en el criterio suministrado por [4], haciendo referencia a la documentación, ya que el set de datos de evaluación se encuentra en el límite de dicho criterio. Desafortunadamente, su corrida no produjó resultados en el lapso de 8hs. Luego, se decidió abandonar su utilización.

4.2.8. Random Forest Regressor

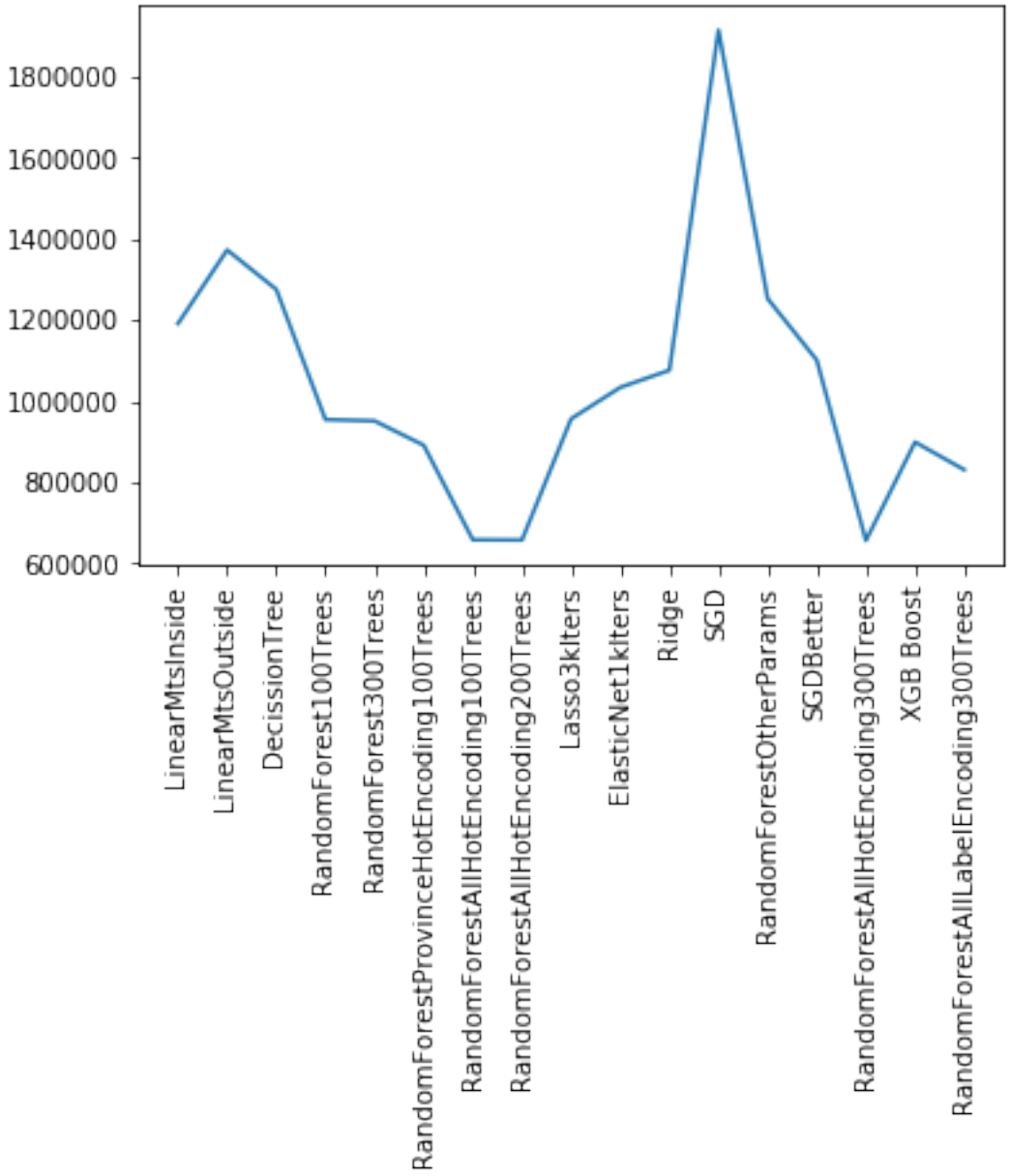
Finalmente, llegamos al algoritmo que arrojó los mejores resultados.

Este algoritmo es un caso particular de un algoritmo de decisión, que incorpora el algoritmo de arboles aleatorios y el método de arboles extra. La principal característica del algoritmo es incorporar la aleatoriedad como herramienta para mitigar la varianza del bosque. Esto se logra tomando ciertos datos en la construcción de cada árbol y estructurándolos considerando el mejor orden asociados a las *features*, de modo optimizar su relación de causalidad -correlación- en la posterior predicción.

Con respecto a los hiper parámetros, se modificó principalmente el referido a la cantidad de estimadores, obteniendo progresivamente mejores resultados. No se siguió aumentando el número, ya que la computadora empleada para la tarea, no lo permitió. Sin embargo, si se hicieron pruebas automatizadas con base en la cantidad de estimadores -*trees*-.

5. Resultados obtenidos y análisis

En el gráfico que se observa a continuación se llega a denotar el conjunto de resultados obtenidos, según el puntaje de Kaggle:



5.1. Análisis de los resultados

De los resultados obtenidos se puede llegar a la conclusión de que *Random Forest* es el mejor algoritmo para este caso, con los parámetros testeados. Más allá de este hecho evidente lo cierto es que analizar en conjunto tantos algoritmos no resulta en una tarea sencilla, en especial cuando llegar a comprender en profundidad cada uno de ellos tampoco lo es. Sin embargo, si se puede hacer una evaluación de las mejoras notadas tras la incorporación de técnicas de modelado preliminares del set de datos, ya que aquí es donde se observó el mayor cambio en los resultados, más allá de lo arrojado debido a la especificidad propia de cada algoritmo. Una vez incorporados los *features* de ciudad y provincia, se puede notar una significa mejora. Esto es razonable y consistente con lo observado en el trabajo practico previo, ya que esta información, la ubicación, incidía significativamente sobre los precios.

Otra mejora significativa se llegó a notar tras modificar los hiper parámetros de los algoritmos que se consideraron de mayor potencial. Esto es consistente, ya que ellos emplean estos datos para poder llevar a cabo las predicciones.

6. Conclusiones

El presente trabajo resultó de particular interés para nuestra formación profesional, permitiendo proporcionarnos un primer acercamiento a dos ramas de la informática que revisten muchas posibilidades y con potencial de crecimiento. Se trata de *big data* y *machine learning*. A su vez, se ha podido desarrollar un criterio analítico en la consideración de diversos algoritmos para la resolución de un problema.

Finalmente, vale destacar que tuvimos la oportunidad de participar de una practica basada en el empleo de datos reales y en la posibilidad de crecer como curso partiendo de la puesta en conjunto de los resultados, motivándonos a mejorar.

6.1. Posibles mejoras

En apartados previos, se ha mención a algunas mejoras que podrían considerar en un futuro. Además, valdría la pena mencionar que el set de datos provisto nos da la posibilidad de llevar a cabo una gran diversidad de análisis, con potencialidad productivo para Zonaprop, individualmente, y para el mercado inmobiliario en su conjunto. Para el primero, permitiendole profundizar el análisis más allá de poder predecir como se comportará económicamente el mercado, incorporando las ideas que podrían motivar mas al consumidor y a quien utilicen sus servicios. Para el segundo, pudiendo llegar a tener una referencia de considerable.

Además, se considera que sería posible profundizar más el uso de estos algoritmos, llegando a emplear tal vez otras técnicas y obtener mejores resultados.

Finalmente, es de interés considerar que se podrían efectuar más filtrados preeliminares a los datos que ya se poseen, considerando más categorias e incluso la diferencia que se llega a apreciar entre los datos de entrenamiento y evaluación en lo referido a ubicación (ciudad y provincia). Se hizo mención a esto en un apartado previo.

7. Referencias consultadas

[1] Luis Argerich, Apunte del Curso.

[2] Geron Aurelien, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Oreilly, 2017.

[3] $https : //towardsdatascience.com/data - handling - using - pandas - machine - learning - in - real - life - be76a697418c$

[4] $https : //scikit - learn.org/stable/tutorial/machine_learning_map/index.html$

[5] $https : //medium.com/analytics - vidhya/building - a - machine - learning - model - to - predict - the - price - of - the - car - bc51783ba2f3$

[6] $https : //nthu - datalab.github.io/ml/labs/05_Regularization/05_Regularization.html$

[7] $https : //medium.com/@contactsunny/label - encoder - vs - one - hot - encoder - in - machine - learning - 3fc273365621$