Organización de Datos

Segundo Cuatrimestre 2017

Trabajo Práctico 2

Integrante	Padrón	Correo electrónico
Rodrigo De Rosa	97799	rodrigoderosa@outlook.com
Marcos Schapira	97934	schapiramarcos@gmail.com
Facundo Guerrero	97981	facundoiguerrero@gmail.com

Índice

1. Resumen

Luego de procesar los datos de acuerdo a las conclusiones obtenidas en el primer trabajo, aplicando distintos algoritmos para realizar la predicción, obtuvimos los mejores resultados mediante Gradient Boosting Regressor utilizando:

- *División de propiedades en grupos de acuerdo al metro cuadrado*
- *Superficie total*
- *Tipo de propiedad*

Esto nos lleva a pensar que la predicción de precios de propiedades de Properati se puede estimar con certeza hasta cierto punto, ya que los datos están altamente comprometidos con graves inconsistencias tanto en precio como en superficie. Recomendamos a la empresa revisar su proceso de recolección de datos exhaustivamente.

2. Adaptación de los datos

Antes de lanzarse a la predicción, se deben procesar los datos de una manera consistente (mayormente con el análisis de TP1) para así acercarse lo mas posible a una estimación correcta. Las siguientes son las conclusiones mas fuertes que obtuvimos del primer trabajo.

2.1. Rango coherente de variación de precio por metro cuadrado

Al igual que en el primer trabajo práctico, se recortaron los datos según su precio por metro cuadrado para de esta manera tratar de quedarnos con datos que tuvieran el mayor 'sentido' posible. Estos recortes se hicieron superior e inferiormente, de la misma manera que en el primer trabajo.

2.2. División por grupos acorde al precio por metro cuadrado

A partir del análisis realizado en el primer trabajo práctico, en el que se dividió a los barrios en seis grupos según su precio por metro cuadrado, surgió la idea de codificar a cada barrio a partir de su posición en el ranking de precio por metro cuadrado; es decir, se le asignó un número a cada barrio que depende de su precio por metro cuadrado y la distancia de un barrio al próximo depende de que tanto mayor es su precio. De esta manera, se buscó determinar una distancia unidimensional entre los barrios que no dependa de su ubicación geográfica sino del valor de su suelo.

2.3. La propiedades con mayores precios dentro de capital se ubican cerca de paradas del subterráneo

Para esto procesamos la columna de Çantidad de Paradas de Transporte Publico Cercanas". A grandes rasgos la utilizamos de 2 maneras, primero para los datos completos y segundo separando para CABA y GBA (tanto usándolas o no en GBA). Bajo estos criterios variamos las escalas de los valores y fuimos analizando como se comportaban con los distintos algoritmos probando una gran cantidad de variaciones sin tener mucho éxito.

2.4. La cantidad de locales gastronómicos y escuelas dan resultados similares sobre los centros mas importantes

Como ya anticipamos en el tp anterior, los locales gastronómicos y las escuelas cercanos nos brindan información similar. Es por esto que probamos tanto utilizando ambas para el calculo como solo una o ninguna. A la vez también variamos las escalas de estas.

2.5. La ubicación en general influye en el precio

Procesamos las columnas de latitud y longitud tanto unidas como separadas, para tratar de darle una influencia individual a la ubicación.

2.6. Conclusiones

Los mejores resultados sin embargo, los obtuvimos sin tener en cuenta gran parte de las consideraciones recién planteadas, básicamente el único análisis que causo impacto fue el del precio por metro cuadrado. Esto comienza a anticipar la inconsistencia de los datos del set de prueba y como el análisis hecho en el trabajo anterior tiene sentido para datos consistentes con el mundo real.

3. Estimación de las superficies faltantes en el set de prueba

Ya que los algoritmos de predicción no pueden recibir entradas del tipo "nan"tuvimos que plantear distintas estrategias para solucionar esto. Se completo mediante la mediana general, el promedio general, e incluso sacando un promedio por barrios, sin embargo el mejor resultado se obtuvo mediante llenarlo con el valor de mayor frecuencia.

3.1. Conclusiones

Dentro de este mismo tema notamos algo realmente interesante. Siendo 14166 el total de propiedades en el set a estimar, 120 de estas tienen un tamaño menor a 9 metros cuadrados (tamaño mínimo reglamentario de una celda de prisión), e incluso parte de estas se encuentran en los barrios mas caros de la región a analizar. Esto agrega mayor importancia a la idea de que los datos de prueba no son de muy buena calidad.

4. Algoritmos de predicción

Todos los algoritmos los analizamos de acuerdo a cross validation cambiando las distintas proporciones entre set de entrenamiento y test. El set de datos para el entrenamiento fue el que utilizamos en el primer trabajo en donde ya están descartadas todas las propiedades que no tienen mucho sentido y dañan al set (mas que nada por superficies y precios por metro cuadrado ilógicos). Esto sin embargo no impide que el set de datos tenga las mismas anomalías. Al tener que calcularlas igual es imposible evitar el gran error de antemano con el que se encuentra este segundo trabajo. Los siguientes fueron los algoritmos que se utilizaron junto con sus mejores combinaciones. Se utilizaran las implementaciones por la librería scikit-learn [1].

4.1. Random Forest Regressor

Luego de un análisis previo sobre problemas similares de este tipo, e incluso de competencias de Kaggle internacionales sobre estimación de precios de propiedades, todas coinciden en que este algoritmo es de los mas adecuados para utilizar. Luego de llegar a muy buenos resultados usando las columnas "place value", "superficie", "tipo de propiedad", çantidad de paradas de transporte", çantidad de locales gastronómicos", y "latitud y longitud"se obtuvieron realmente buenos resultados con bajos errores de rmse. Sin embargo al subir los resultados a Kaggle estos dan muy lejos de lo calculado. Luego de quitar columnas y dejando el algoritmo mas simple se lograron mejores resultados con solo "place value", "superficie", y "tipo de propiedad". Algo similar ocurrió con el siguiente algoritmo Gradient Boost Regressor. Siendo que este ultimo suele ser particularmente robusto contra overfitting, se sostiene la teoría de que los datos de prueba presentan una cantidad importante de inconsistencias. Los hiperparametros para este mejor caso son:

- *n estimators=100*
- *max features='log2'*
- *min samples leaf=2*
- *min samples split=4*
- *bootstrap=True*
- *oob score=True*

4.2. Gradient Boosting

Al igual que con Random Forest, el uso de este algoritmo surge de la investigación; es conocido como 'el algoritmo que nunca anda del todo mal' y aquí no fue una excepción. También al igual que en Random Forest, se comenzó utilizando varios features resultantes del análisis del TP1 pero finalmente los mejores resultados se obtuvieron con 'place value', 'superficie' y 'tipo de propiedad'.

Luego de realizar algunos Grid Searchs, este algoritmo dio los mejores resultados a partir de los siguientes hiperparámetros:

- *n estimators = 5000*
- *learning rate = 0.241*
- *max depth = 5*

Cabe destacar que se realizaron pruebas tanto con la librería *sklearn* como con la librería *XGBoost* para comparar resultados y tratar de obtener el mejor posible. Los mejores resultados se obtuvieron con la primera.

[1] http://scikit−learn.org/stable/

4.3. Ada Boost

Al igual que los algoritmos anteriores, el uso de este algoritmo surge de la investigación. Aquí, como se realizo anteriormente, se comenzó utilizando varios features resultantes del análisis del TP1 pero finalmente los mejores resultados se obtuvieron con 'place value', 'superficie' y 'tipo de propiedad'.

Luego de realizados los Greed Search, se obtuvieron los siguientes hiper parámetros:

- *criterion = mse*
- *min_samples_leaf = 3*
- *min_samples_split = 8*

De todas formas, el algoritmo no dio muy buenos resultados.

4.4. Extra Regressor

Este algoritmo, como los mencionados anteriormente, se decidió utilizar ya que es un ensamble. Como con los algoritmos anteriores, se comenzo con varios features pero finalmente solo se utilizo 'place value', 'superficie' y 'tipo de propiedad'. Luego de hacer los Grid Search, se obtuvieron los siguientes hiper-parámetros:

- *n estimators=100*
- *max leaf nodes = 0*
- *min samples leaf=1*
- *min samples split=2*
- *bootstrap=True*

4.5. KNN

KNN es uno de los algoritmos de Machine Learning dados por la cátedra, y además uno de los mas conocidos. Luego de realizar varios Grid Search se obtuvieron distintos hiper parámetros. Estos fueron muy variantes de acuerdo a que columnas del set de datos se incluían para realizar la predicción.

Pero por si solo, de este algoritmo no obtuvimos buenos resultados.

4.6. Ensambles

A medida que se iban obteniendo buenos resultados, se decidió probar con ensambles de algoritmos que entregaban dichos resultados. Mayormente para los ensambles se uso, Gradient Boosting y Random Forest. Lo que se hizo fue para los distintos hiper parámetros para los cuales se obtenían buenos resultados, se calculaba el error y luego se realizaba el promedio de la salida de los 2 algoritmos, ponderando hacia el algoritmo que mejor resultado entregaba.

4.7. Conclusiones

El mejor resultado se obtuvo con Gradient Boost Regressor llegando a dar los dentro del mismo orden que los mejores resultados de la competencia.

5. Conclusiones Generales

Luego del análisis realizado y de acuerdo a los resultados obtenidos, se concluye que el set de datos proporcionado para las pruebas esta lleno de datos inconsistentes o sin sentido. Esto provoca que las predicciones en general no sean buenas, dado que tratar de predecir valores a partir de información inconsistente tiene una alta probabilidad de error.

Por esto, consideramos que sería correcto que el sistema de publicaciones de Properati estuviera un poco más controlado para evitar inconsistencias tales como propiedades con superficie nula o mas pequeñas que una celda de prisión, propiedades cuyo precio total no se corresponde con la multiplicación de su superficie por el precio de su metro cuadrado, propiedades con precio por metro cuadrado muy pequeño o extremadamente alto, etc. que aunque si bien se pueden definir reglas para definir y modificar estos valores erróneos, entorpecen mucho la correcta predicción del valor de una propiedad, pues los algoritmos se basan en estos datos para predecir correctamente.