Informe trabajo práctico número dos - Aprendizaje Automático

Omar Ernesto Cabrera Rosero
Universidad de Buenos Aires
Email: omarcabrera@udenar.edu.co

Jimmy Mateo Guerrero Restrepo
Universidad de Buenos Aires
Email: jimaguere@gmail.com

***Resumen*—En este trabajo práctico se desarrolla un modelo de predicción del rango del precio de m^2 de propiedades a la venta en la Ciudad de Buenos Aires, de esta forma poder estimar el rango del precio del m^2 de un inmueble para decidir el nivel de especialista que se destinará para realizar la inspección física y así obtener una cotización real del precio en venta. Para la realización del modelo se tuvo en cuenta conceptos de árboles de decisión, resamplig, ensamble y minería de texto.**

***Keywords*—Árboles de decisión, clasificación, inmobiliaria, resampling, ensamble, minería de texto.**

I. Introducción

La Inmobiliaria www.barealestate.com.ar dispone de un sitio Web para la compra y venta de inmuebles en la Ciudad de Buenos Aires. Para disminuir los costos de publicación, necesita disponer de una prevalorización de los inmuebles publicados a la venta por los propietarios.

La empresa necesitaría de alguna forma estimar el rango del precio del metro cuadrado del inmueble para decidir el nivel de especialista que destinará para realizar una inspección física y así obtener una cotización real del precio de venta. Para la empresa, el costo de la publicación depende mucho del nivel del especialista que tiene que realizar la cotización, por lo tanto, desea bajarlo manteniendo acotado el riesgo de una mala cotización.

Se usaron dos conjuntos de datos, uno para entrenamiento con 15.000 registros y otro con 4.000 registros para devolverlo clasificado, el conjunto de datos tiene 5 clases las cuales vienen dadas de la siguiente manera:

- Clase 1: m^2 <= \$2.000
- Clase 2: \$2.000 <m^2 <= \$2.500
- Clase 3: \$2.500 <m^2 <= \$5.500
- Clase 4: \$5.500 <m^2 <= \$18.500
- Clase 5: \$18.500 <m^2

El objetivo de este trabajo práctico es desarrollar un modelo de predicción del rango del precio del metro cuadrado de propiedades a la venta en la Ciudad de Buenos Aires, realizando modelos de predicción teniendo en cuenta conceptos de árboles de decisión, minería de texto, resampling y ensamble.

II. Metodología

Se utilizó la metodología CRISP-DM (Cross Industry Standard Process for Data Mining) uno de los modelos principalmente utilizados en los ambientes académico e industrial y la guía de referencia más ampliamente utilizada [1].

Esta metodología contempla seis fases: entendimiento del negocio, entendimiento de los datos, preparación de los datos, modelado, evaluación e implementación, que se describen a continuación.

A. Entendimiento del Negocio

Se adquirió conocimiento en cuanto a la valorización en metros cuadrados, teniendo en cuenta factores como lo son la ubicación, cantidad de ambientes, superficie total, piso en el que se encuentra el departamento y hacer un análisis de minería de texto en las descripciones que tienen cada departamento.

B. Entendimiento de los Datos

Se construyó un conjunto de datos y se analizó cada atributo, realizando un análisis exploratorio de datos como por ejemplo tabla de frecuencias, datos nulos e identificadores únicos.

Se puede observar la frecuencias de las palabras en la figura 1.

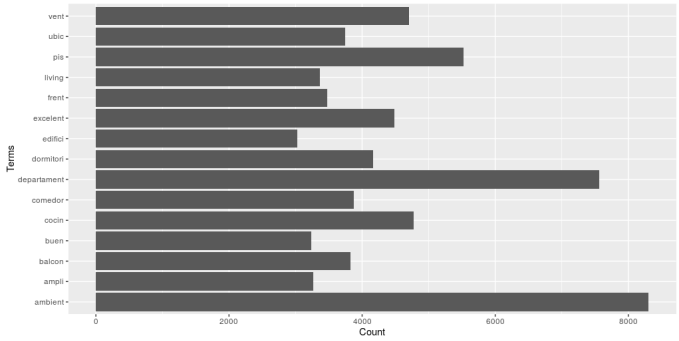

Figura 1. Frecuencias de palabras

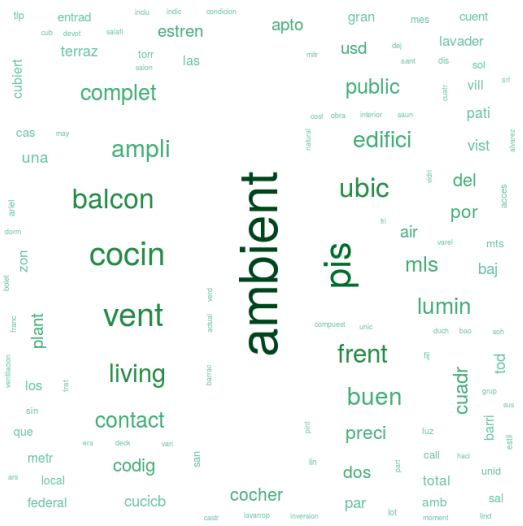

Figura 2. Nube de palabras

C. Preparación de los Datos

La preparación de datos para construir el conjuntos de datos de entrenamiento se lo realizó teniendo en cuenta las siguientes transformaciones:

- Se aplico minería de texto al atributo lugar de tal manera que se hizo el cambio del nombre del barrio a columnas debido a que los algoritmos implementados como random forest únicamente admiten un máximo de 53 categorías en cada atributo y no admite datos nulos.
- Para evitar los datos nulos se utilizo estrategias de relleno: tanto a los pisos como los ambientes que tenían datos nulos se los relleno con el valor de "1", para la superficie total de metros cuadrados que estaban nulos se el asignó la superficie cubierta de metros cuadrados, este mismo proceso se lo realizó luego en sentido contrario.
- Se eliminaron las siguientes columnas: ident, fecha, geoname_num, lat_lon.
- Se fusionó los textos de las columnas des y tit, y luego se les aplico minería de texto, además se hizo una limpieza de acentos, palabra muertas, se removió los números y espacios en blanco.
- Antes de aplicar las técnicas de modelado, se procedió a evaluar la calidad y relevancia de los atributos del conjunto de datos con el objetivo de predecir valores de la clase buscada. Se utilizó el algoritmo "boruta" propuesto en [2] para la extracción y evaluación de atributos. El algoritmo está diseñado como un recubrimiento alrededor del algoritmo de clasificación random forest y califica cada atributo en el conjunto de datos de acuerdo a su importancia a la hora de clasificar. Se puede observar la relevancia en la nube de palabras que se muestra en la figura 2

Teniendo en cuenta las anteriores transformaciones descritas se crearon varios conjuntos de datos para luego poder aplicarles modelos de clasificación, la tabla I muestra el conjunto de datos, con su descripción, el nombre del conjunto viene representado con una "V" que es el número de variables y una "R" para el número de registro del conjunto.

Los conjuntos de datos se los puede descargar desde el repositorio[1].

D. Modelado

Para la etapa de modelado se utilizó el modelo de predicción random forest, el cual se basa en el desarrollo de muchos árboles de clasificación. Para clasificar un nuevo objeto desde un vector de entrada, ponemos dicho vector bajo cada uno de los árboles generados. Cada árbol tiene una clasificación, en términos coloquiales diríamos que cada árbol selecciona una clase. Finalmente se escoge la clasificación teniendo en cuenta el árbol más votado sobre todos los demás.

En la tabla II se muestra los resultados al aplicar random forest en los conjuntos generados en la etapa de preparación de datos.

El conjunto de datos que tiene la mejor performance es "V221R14625" el cual esta compuesto por

[1]https://github.com/poldrosky/tp2AA

Tabla I. Conjuntos de Datos

Nombre del conjunto	Descripción
V1797R9714	Variables propias del conjunto. Minería de texto a la descripción. Minería de texto al lugar. Aplicación de algoritmo boruta al lugar. Frecuencia de texto mayor a 10. Sin estrategia de relleno. Eliminación de registros nulos.
V81R9714	Variables propias del conjunto. Mineria de texto al lugar. No se usó la descripción. Sin estrategia de relleno. Eliminación de registros nulos.
V57R9714	Variables propias del conjunto. Mineria de texto al lugar. Aplicación de algoritmo boruta al lugar. No se usó la descripción. Sin estrategia de relleno. Eliminación de registros nulos.
V3385R14625	Se rellenaron datos faltantes. Variables propias del conjunto. Minería de texto a la descripción. Frecuencia de texto mayor a 3. Minería de texto al lugar.
V2652R14625	Se rellenaron datos faltantes. Variables propias del conjunto. Minería de texto al lugar. Minería de texto a la descripción. Frecuencia de texto mayor a 5.
V221R14625	Se rellenaron datos faltantes. Variables propias del conjunto. Minería de texto a la descripción. Frecuencia de texto mayor a 5. Minería de texto al lugar. Aplicación de algoritmo boruta.

Tabla II. Performance de los conjuntos

Nombre del conjunto	Performance
V1797R9714	0.533
V81R9714	0.455
V57R9714	0.466
V3385R14625	0.529
V2652R14625	0.528
V221R14625	0.549

221 variables y 14525 registros, a este conjunto se le aplicaron otros modelos y se evaluo la performance, los resultados se pueden ver en la tabla III.

Estos modelos fueron creados utilizando la biblioteca de código abierto rminer presentada por [3] para la herramienta R.

E. Evaluación

Se evaluaron clasificadores tanto como arboles de decisión, como modelos probalísticos. El mejor árbol que tubo mejor perfonmance fue el árbol de decisión "random forest", y el modelo probabilístico que tubo mejor perfomance fue el modelo "linear discriminant analysis (lda)".

Tabla III. Performance conjunto V221R14625

Modelos	Performance
ctree	0.435
rpart	0.391
kknn	0.425
ksvm	0.471
random forest	0.549
bagging	0.384
lda	0.436
naiveBayes	0.351

F. Implementación

La etapa de implementación se la realizo utilizando resampling y ensamble para luego aplicarse en el conjunto de prueba.

Para el ensamble se hizo resampling con el conjunto de datos de 221 variables, con el cual se tenia un 54 % de performance. El conjunto de datos se partió por filas (resampling por filas) en 4 partes aleatorias, cada una con el 25 % del total filas, a cada partición se les aplico modelos random forest variando los parámetros del algoritmo.

Una vez generados los cuatro modelos de clasificación se tomo el conjunto a clasificar y se realizaron las predicciones con dichos modelos. Para generar el conjunto de datos de clasificación se realizo una votación simple con los resultados predichos por los cuatro modelos. Para los casos en que alguno de los modelos no arrojaba una clasificación o había un empate de clases entre los modelos, se utilizo el modelo "lda" para clasificar las instancia no clasificadas o para desempatar la decisión.

III. Conclusiones

Se construyón un modelo de clasificación para resolver el problema para predecir el precio del m^2 a la venta en la Ciudad de Buenos Aires, de esta forma poder estimar el rango del precio del m^2 de un inmueble.

El árbol random forest, resultó ser el mejor clasificador de árboles debido a que se basa en el desarrollo de muchos árboles de clasificación que dependen de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos.

El hacer resampling por filas sirve para generar varios modelos, y con ellos realizar votación, esto ayuda a que al aplicar estos modelos, se pueda hacer una votación y se pueda quedar con el que valor con más frecuencia.

Cuando se aplica árboles de decisión algunos datos no se van a clasificar, por esta razón fue necesario utilizar un clasificador probabilístico para los datos que no fueron clasificados.

Referencias

[1] J. Hernández Orallo, M. J. Ramírez Quintana, and C. Ferri Ramírez, *Introducción a la Minería de Datos.* Pearson Educación, 2004.

[2] M. B. Kursa, W. R. Rudnicki *et al.*, "Feature selection with the boruta package," 2010.

[3] P. Cortez, "Data mining with neural networks and support vector machines using the r/rminer tool," in *Advances in data mining. Applications and theoretical aspects.* Springer, 2010, pp. 572–583.