Predictive Models - Airbnb Rental Price

Entrega final

JUANITA FONNEGRA VILLEGAS
DIEGO ALEJANDRO SAAVEDRA VALDIVIESO
SANTIAGO LEÓN MACIA PALACIO

INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL
Profesor : RAÚL RAMOS POLLÁN

Ingeniería Industrial
Facultad de ingeniería
Universidad de Antioquia
2022

Nota: Tenemos 5 páginas sin contar portada e índice

I. DESCRIPTION

Dadas las características de los alquileres de una casa disponible en Airbnb en New Orleans (características, ubicación, etc.) vamos a predecir cuánto debería cobrar un alquiler a corto plazo en Nueva Orleans por noche en función de su ubicación y comodidades

II. INSPECTION

Los datos están compuestos por dos datasets, uno llamado listings y otro reviews, se ha decidido como grupo que no se utilizará el dataset de las reviews debido a que solo contenía datos de cada review en formato de texto, entonces para poderla utilizar se tendría que hacer una limpieza muy extensa.

El dataset de listing contiene un total de **6028** filas y **49** columnas de las cuáles hay:

- 13 variables de tipo flotante
- 12 variables de tipo entero
- 24 variables de texto
- Nuestra variable objetivo es el **precio**

III. DATA CLEANING

CATEGORICAL AND NUMERICAL VARIABLES

Las siguientes variables fueron eliminadas ya que la información proporcionada no fue considerada como útil para la predicción del precio.

- id
- name
- description
- neighborhood_overview
- host_since
- host_location
- host_verifications
- host_has_profile_pic
- host_neighbourhood
- host_identity_verified
- first_review
- last_review
- license

Aparte de esto se corrigieron las siguientes variables:

- Separamos la variable *bathrooms_text*, la cuál estaba nombrada así porque tenía la cantidad de baños y el tipo de baño (1 baths, 2 shared baths, etc.), se separó en 2 variables:
  bathrooms: Cantidad de baños
  bathrooms_type: Tipo de baño

- Se convirtieron en flotante y se eliminó la columna bathroom_text
- En *bathrooms_type* reemplazamos “bath” y “baths” por “entire home/apt” debido a que en el dataset existían esos dos tipos de datos para baño.
- Se corrigió la variable *price* removiendo el signo peso ($) y las comas(;) para poder convertir los datos en tipo flotante.
- Se corrigieron las variables *host acceptance rate* y *host response rate* extrayendo el signo de porcentaje (%) y poder convertir en tipo flotantes.
- Se cambiaron las variables *instant_bookable, host_is_superhost, has_availability* que eran binarias de tipo texto se cambiaron a binarias de tipo entero
- Se renombraron las siguientes columnas:
    - *review_scores*_rating por *global_score*.
    - *neighbourhood_cleansed* por *neighborhood*
- Se reordenaron las columnas dejando *price* al final

NULL DATA CLEANING

Nulos por una palabra:
Se procede a cambiar los datos ubicados en las variables con contenido nulo por la palabra ‘None’, por ejemplo la columna “bathrooms_type”, los datos nan, se cambian por “no baths”, este mismo proceso se aplica a las siguientes variables:

- bathrooms_type: nan → no baths

Nulos por el valor 0:
Debido a su contexto, se verifica en la base de datos, el porqué del nulo y se puede inferir que si es correcto aplicarles el valor 0, a las siguientes variables:

- bedrooms
- beds
- bathrooms
- reviews_per_month
- host_listings_count

Nulos por el promedio del anfitrión
Nuestra lógica fue: Si el anfitrión tiene una puntuación promedio en otras casas de su propiedad, este mismo anfitrión puede tener ese mismo promedio en las que no poseen valores

- global_score
- review_scores_accuracy
- review_scores_cleanliness
- review_scores_checkin
- review_scores_communication
- review_scores_location
- review_scores_value

Los nulos que nos quedan los dividimos en dos, las reviews y las tasas de respuesta y aceptación del host, y se hace lo siguiente:

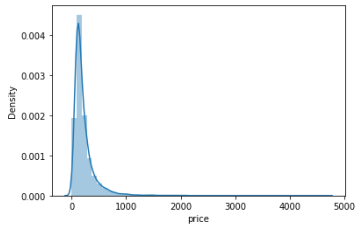
- Se decide eliminar los registros de las columnas de los reviews porque eran **257** filas y no perdemos casi registros

- Nos damos cuenta que la variable *price* tiene 2 valores en 0, por lo que se eliminan, esas 2 filas
- Pero se deciden eliminar las columnas **host_response_time', 'host_response_rate', 'host_acceptance_rate** como eran demasiados registros nulos, lo mejor fue eliminar la columna para no perder tantos registros

Quedamos al final con **5731** registros y con **33** columnas

IV. DATA EXPLORATION

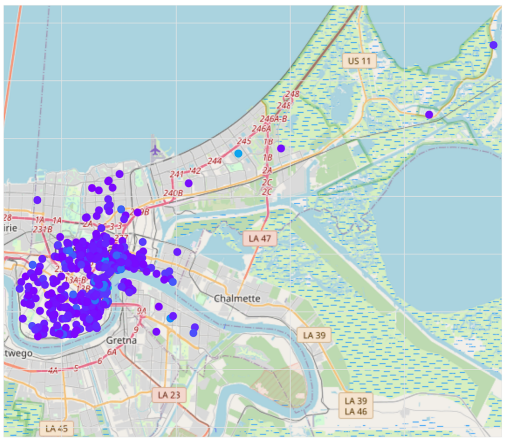
Se inspecciona la variable objetivo, que en este caso es 'price', se desea realizar un análisis de su distribución, por medio de la librería seaborn, obtenemos la siguiente gráfica de distribución

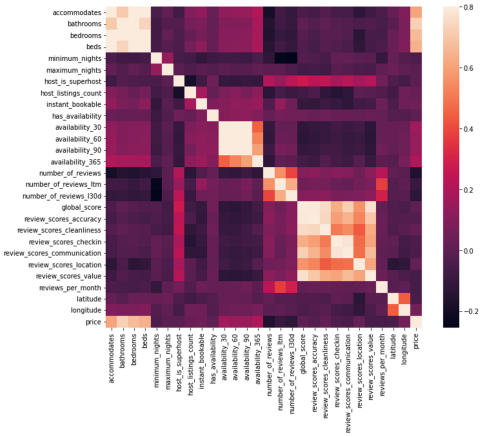


Vemos que los valores no tienen una distribución normal ya que están mayormente concentrados hacia la izquierda

También vemos que el valor promedio del precio de los alquileres es de **$222**, el precio mín. es de **$11** y el precio máx. es de **$4657**

Mapa de calor en ciudad y correlación:





También para observar el comportamiento de la variable precio, que es nuestra variable objetivo, aplicamos un gráfico de correlación, la cual se obtiene lo siguiente:

Se observa en la anterior gráfica que la variable 'price' tiene una correlación muy cercana con la **capacidad de la habitación** y la **cantidad de baños**, **piezas** y **camas**

V. DATA PREPROCESSING

En esta sección se transformaron los datos categóricos y numéricos; debido a la gran cantidad de categorías se eliminaron las siguientes columnas:

- 'property_type'
- 'amenities'

Quedamos al final con **5731** registros y con **31** columnas

Luego se codifican de manera ordinal las variables *neighbourhood, room_type, 'bathrooms_type* para comenzar con la modelación

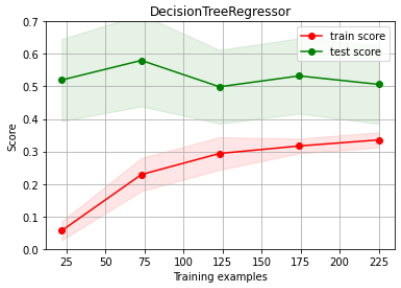
VI. MODEL EVALUATION

SUPERVISED MODELS

Se toma la mitad del dataset y se saca una muestra aleatoria del 5% de los datos, es decir 250 datos, y de esos datos se escogerá el test y el train pero de *validación*, quedando un 30% test y un 70% train y luego se escogen porcentajes para *real test* pero relativos a los de *validación*, real test quiere decir, datos que nunca ha visto el modelo, la otra mitad de los datos

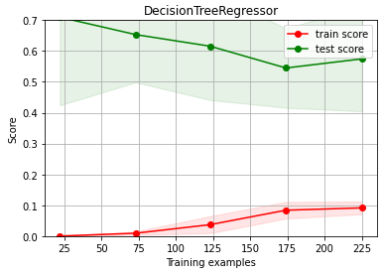
Métrica de desempeño:

Se elige la métrica Mean Relative Absolute Error (MRAE) cuando se realiza una medición se considera que su calidad es mucho mayor cuanto más pequeño es el error relativo que se comete.



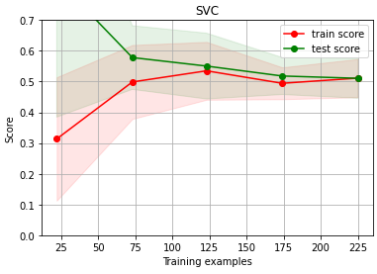
MODEL 1 Decision Tree Regressor (5)

Podemos ver un poco de **sesgo** y **overfitting**, ya que la línea del train no tiende a 0 y hay un espacio entre las líneas, por lo que decidimos aumentar la complejidad del modelo a 10



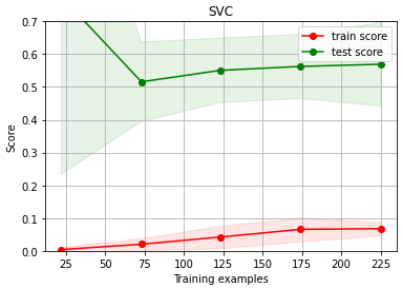
MODEL 2 Decision Tree Regressor (10)

En este caso no hay casi sesgo porque la línea de train tiende a 0 pero hay **overfitting** porque hay un espacio grande entre las líneas Las soluciones serían: Disminuir la complejidad del modelo o aumentar las filas



MODEL 3 SVC (gamma = 0.00001)

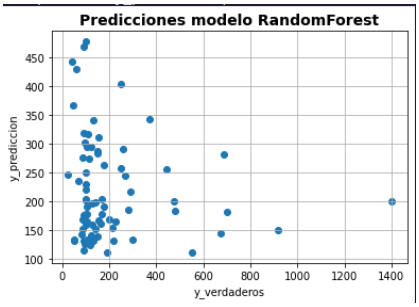
Vemos un caso de sesgo por lo que tomamos la decisión de aumentar la complejidad del modelo, en este caso aumentamos el gamma



MODEL 4 SVC (gamma = 0.001)

Al aumentar la complejidad del modelo, ya vemos un caso de **overfitting**

UNSUPERVISED MODELS



PCA

Primero se escoge el mejor REL_MRAE, el cual es: REL_MRAE: **0.72885** ; obtenido con **5** componentes para PCA

Luego se elige un conjunto de train y test (175, 7) (75, 7), se ejecuta el modelo arrojando los siguientes resultados:

- Mejor estimador Random Forest: RandomForestRegressor(max_depth=9, n_estimators=10)
- Mejores parámetros para el estimador Random Forest: {'max_depth': 9, 'n_estimators': 10}

REL_MRAE del Random Forest en entrenamiento: 0.47345
REL_MRAE del Random Forest seleccionado: 1.26922
Score train = 0.75192 Score test = -1.19896

NOTA: No entendemos, por qué nos da números mayores a 1 ni tampoco números menores que 0

NMF

Al querer escoger el mejor REL_MRAE, nos arroja un error a la hora de transformar en esta línea

```
X_t = nmf.fit_transform(X)
ValueError: Negative values in data passed to NMF (input X)
```

NOTA: No sabemos qué hacer y por eso lo dejamos como iteración para tener en cuenta ese error y tomarlo como un reto o como recomendaciones en este trabajo

VII. MODEL SELECTION

SUPERVISED MODELS

El mejor modelo que nos da menos error es el **Model 3 - SVC(gamma=0.00001)** y ese modelo lo testeamos con la otra mitad de los datos (2576) y nos arroja un error de predicción de **46.2%**

UNSUPERVISED MODELS

Se eligió el modelo RandomForestRegressor(max_depth=9, n_estimators=15) en la configuración (175, 7) (75, 7), arrojando un error de predicción de **47%**

VIII. RECOMMENDATIONS

- Se probaron en ambos modelos aumentando y disminuyendo los hiper parámetros y estamos pasando de casos de sesgo a sobreajuste y de sobreajuste a sesgo, lo que nos indica que debemos agregar columnas o agregar más filas
- Analizar el por qué en el modelo no supervisado PCA nos da números mayores a 1 y en otros casos números menores que 0
- Analizar el por qué en el modelo no supervisado de NMF no nos deja transformar y ajustar los valores de X