PREDICCIÓN DE LA RENTA EN CIUDADES METROPOLITANAS DE LA INDIA

CODER HOUSE, DATA SCIENCE

CARACAS, 5 DE MARZO DE 2023

GILBERT BARRETO

A. Tabla de contenido

A. Tabla de contenido **1**
1. Descripción del caso de negocio **2**
2. Objetivos del modelo **2**
4. Descripción de los datos **2**
5. EDA: Exploratory Data Analysis **5**
5.1 Análisis Univariado: Renta Media y Mediana 5
5.2 Análisis Bivariado: Renta Mediana por ciudad. 6
5.3 Análisis Multivariado: Agrupación por categoría y matriz de correlación. 7
5.4 Correlación variables numéricas 8
5.4 Número de Pisos 8
5.5 Conclusiones EDA 9
6. Modelo MVP **9**
6.1 Variables elegidas 9
6.2 Feature Engineering 9
6.3 Algoritmos de Regresión 10
6.4 Tratamiento de Outliers 11
7. Optimización con Feature Engineering **11**
7.1 Adición de variables de localización 11
7.2 Eliminación por “Feature Importance” 13
8. Métricas de desempeño del modelo **14**
9. Optimización por Hiperparámetros **15**
Optimización de Hiperparámetros 1 15
Optimización de Hiperparámetros 2 16
9. Métricas finales del modelo optimizado **16**
10. Futuras líneas **17**
11. Conclusiones **17**

1. Descripción del caso de negocio

La naturaleza del mercado inmobiliario moderno suele ser sencilla y compleja al mismo tiempo. Si bien se puede clasificar este mercado por variables fácilmente identificables como los metros cuadrados o número de habitaciones, la venta de un inmueble depende íntimamente del valor subjetivo de aquello que la rodea; es decir, la ubicación donde se encuentra el inmueble. Y como técnicamente un inmueble no puede cambiar de ubicación, el valor del mismo cambia acorde al valor de la ubicación de donde se encuentra. Esto explica, por ejemplo, porque un inmueble objetivamente más pequeño o con menos características (pero dentro de una gran ciudad) puede costar mucho más que uno ubicado en zonas más rurales o con menor cantidad de población.

En el presente estudio se quiere confirmar esta hipótesis y adicionalmente establecer un modelo de machine learning que permita determinar el precio de un inmueble basado en sus principales características y ubicación. Con dicha información, los vendedores de inmuebles pueden obtener información relevante para incrementar sus oportunidades de negocio; y los compradores por su parte, pueden ahorrar tiempo y energía en la búsqueda del inmueble que cumpla con sus expectativas.

A continuación se presentarán los objetivos de este estudio, el análisis descriptivo-estadístico del dataset elegido, para finalmente establecer un modelo de machine learning que permita predecir la renta de un inmueble.

2. Objetivos del modelo

El objetivo de este trabajo es determinar el precio de la renta de un inmueble basado en sus principales características y ubicación. De esta manera surge la duda: ¿qué características son las más relevantes para predecir esta renta? Una pregunta que requiere contexto, dado que no todos los mercados inmobiliarios se comportan de la misma forma. Para reducir el alcance de este estudio se tomaron datos de 6 ciudades principales en la India. Los datos se describen a continuación:

4. Descripción de los datos

Los datos provienen de data pública obtenida de Kaggle (chequear link aquí), la cual contiene información de renta de inmuebles en 6 ciudades en la India (Kolkata, Delhi, Chennai, Bangalore, Mumbai y Hyderabad). El dataset fue creado utilizando técnicas de “web scraping” de la página web https://www.magicbricks.com/.

A continuación se describen las variables del dataset:

NrC	Variable	Description (English)	Descripción (Español)	Example / Ejemplo
1	Posted On	Date of publication	Fecha de publicación	2022-5-18
2	BHK:	Number of Bedrooms, Hall, Kitchen	Número de Cuartos, Salas, Cocina	2
3	Rent:	Rent of the H/A/F. (In Rupees).	Renta de C/A/P (expresado en Rupias).	15000
4	Size:	Size of the H/A/F in ft2.	Tamaño del inmueble en ft2 (pies cuadrados)	1000
5	Floor:	H/A/F situated in which Floor and Total Number of Floors	Piso del C/A/P y número de pisos	Ground out of 2
6	Area Type:	Size of H/A/F calculated on either Super Area or Carpet Area or Build Area.	Tamaño de C/A/P calculado como Super Area, Área de Alfombra o Área de Construcción.	3 posibilidades: Carpet Area, Super Area y BuildUp Area
7	Area Locality:	Locality of the H/A/F	Localidad del C/A/P	Bandam Kommu
8	City:	City where the H/A/F are Located	Ciudad donde se encuentran el C/A/P.	6 ciudades: Kolkata, Mumbai, Chennai, Hyderabad, Delhi y Bangalore.
9	Furnishing Status:	Furnishing Status of the H/A/F	Indica el grado de amueblado.	Tiene 3 categorias: Furnished/Semi-Furnished/ Unfurnished.
10	Tenant Preferred:	Type of Tenant Preferred by the Owner or Agent	Tipo de Inquilino preferido	3 tipos de inquilino: Family, Bachelor, Family/Bachelor
11	Bathroom:	Number of Bathrooms	Número de Baños	1
12	Point of Contact:	Whom should you contact for more information regarding the H/A/F.	Persona contacto del C/A/P	Dos posibilidades: Contact Agent o Owner

*H/A/F = Houses/Apartments/Flats

*C/A/P = Casas/Apartamentos/Pisos

*ft2 = square feet

*Rent = Variable dependiente del modelo

Resumen de variables del dataset:
Tenemos un dataset conformado por 4745 registros, los cuales no contienen valores nulos.

Nr	Column	Non-Null Count	Dtype
1	Posted On	4746 non-null	datetime64[ns]
2	BHK	4746 non-null	int64
3	Rent	4746 non-null	int64
4	Size	4746 non-null	int64
5	Floor	4746 non-null	object
6	Area Type	4746 non-null	object
7	Area Locality	4746 non-null	object
8	City	4746 non-null	object
9	Furnishing Status	4746 non-null	object
10	Tenant Preferred	4746 non-null	object
11	Bathroom	4746 non-null	int64
12	Point of Contact	4746 non-null	object

De estas variables se excluyen del EDA:

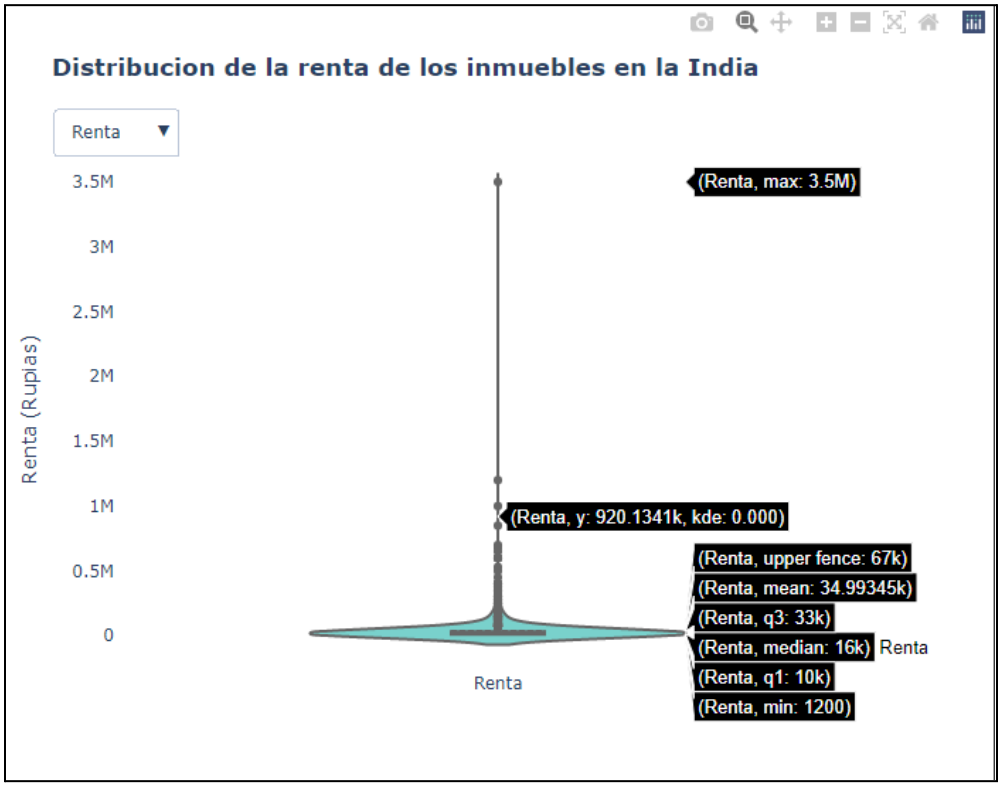
‘Posted On’: La fecha de publicación de los inmuebles no varía mucho. Comprenden aproximadamente entre mayo y julio de 2022. En la medida que pase el tiempo, este dataset se tiene que actualizar, pero no ofrece mayor aporte para el modelo a realizar.

‘Área Locality’: Comprende aproximadamente 2000 direcciones no especificadas. Se utiliza posteriormente un programa que extrae información de google maps para identificar Latitud y Longitud de cada una de esas direcciones.

5. EDA: Exploratory Data Analysis

En el análisis de exploración de datos se hacen los análisis descriptivos básicos, haciendo énfasis en la variable que se quiere explicar, la renta. De esta forma se analizó la distribución de la variable de manera individual y agrupando por ciudad; luego, se agrupa en las variables categóricas del modelo para obtener insights importantes del dataset; para finalmente realizar una matriz de correlación que permitió analizar las relaciones entre las variables numéricas. Los resultados fueron los siguientes:

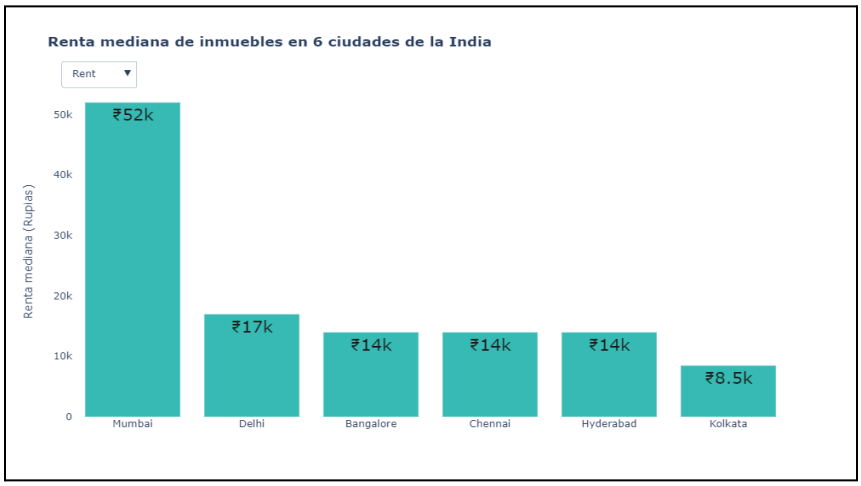
5.1 Análisis Univariado: Renta Media y Mediana



Una de las características más importantes de la renta en el dataset elegido fue encontrar una distribución asimétrica hacia la izquierda con grandes cantidades de valores atípicos. Utilizando el método IQR se determinó que aproximadamente el 10% de los datos son valores atípicos, por lo que se procede a utilizar la mediana como medida de tendencia central para analizar la renta. En efecto, la renta mediana se encuentra en las ₹16.000, mientras que la media se ubica en las ₹35.000 (claramente afectada por los valores atípicos).

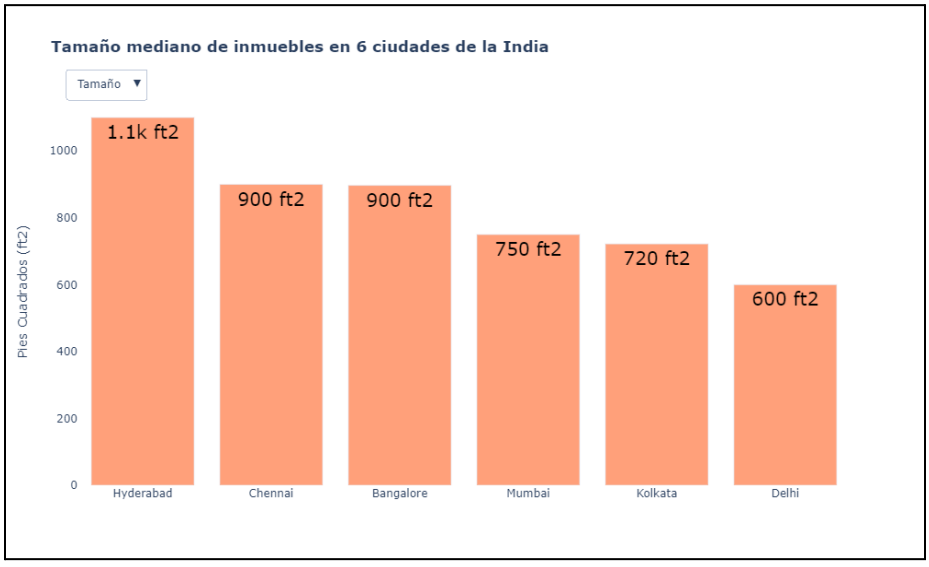
5.2 Análisis Bivariado: Renta Mediana por ciudad.

Entendiendo la distribución de la renta, se procede a analizar la renta mediana agrupada por ciudad. Se sospecha en un principio que la ubicación puede influir sobre las rentas obteniendo los siguientes resultados.



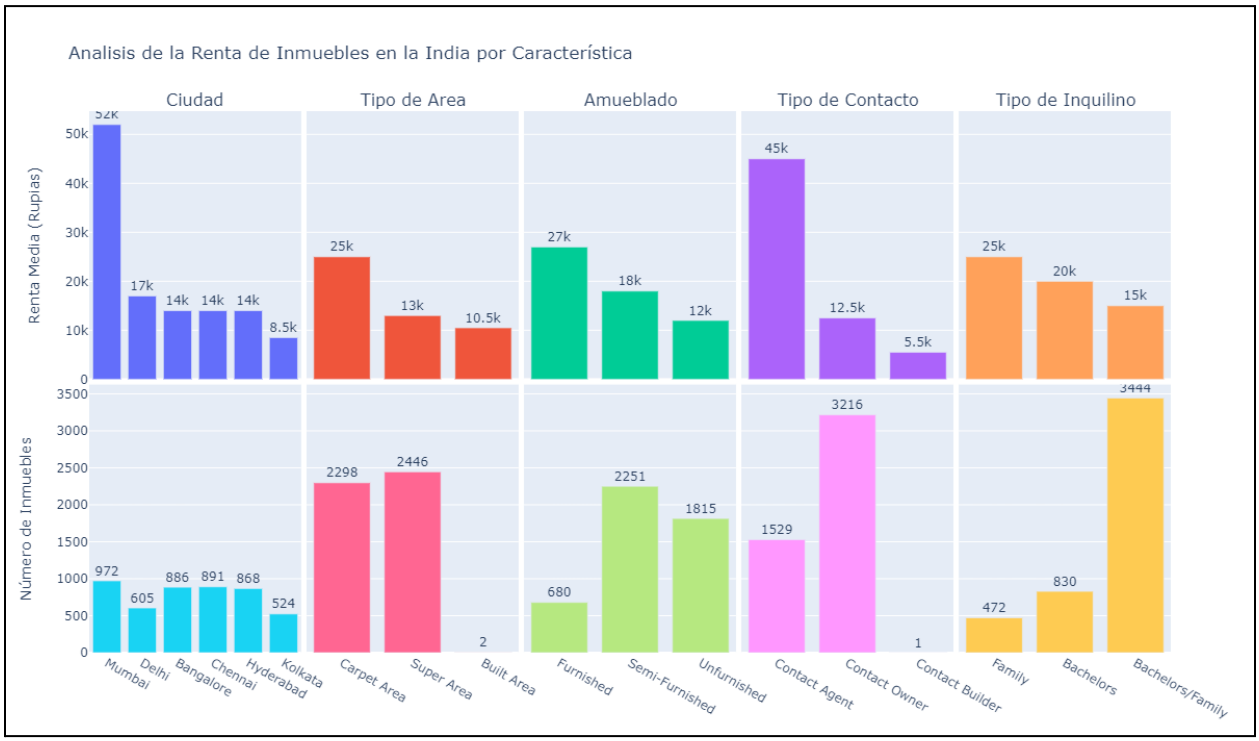
La ciudad de Mumbai tiene la mayor renta mediana (₹52.000), seguido de Delhi (₹17.000), Bangalore, Chennai y Hyderabad (₹14.000) y por último Kolkata (₹8.500). La ciudad de Mumbai tiene rentas medias casi 3 o 4 veces más que las otras ciudades. Mientras Kolkata tiene, en el otro extremo, aproximadamente la mitad de la renta mediana que las demás ciudades (sin contar Mumbai).

Como dato curioso, si se realiza el mismo análisis utilizando las variables de ciudades y metros cuadrados, el orden y la diferencia relativa entre ciudades es distinta, lo que nos indica que el tamaño medido en metros cuadrados no es necesariamente el factor principal de la variación en las rentas, sino más probablemente la misma ubicación del inmueble.



5.3 Análisis Multivariado: Agrupación por categoría y matriz de correlación.

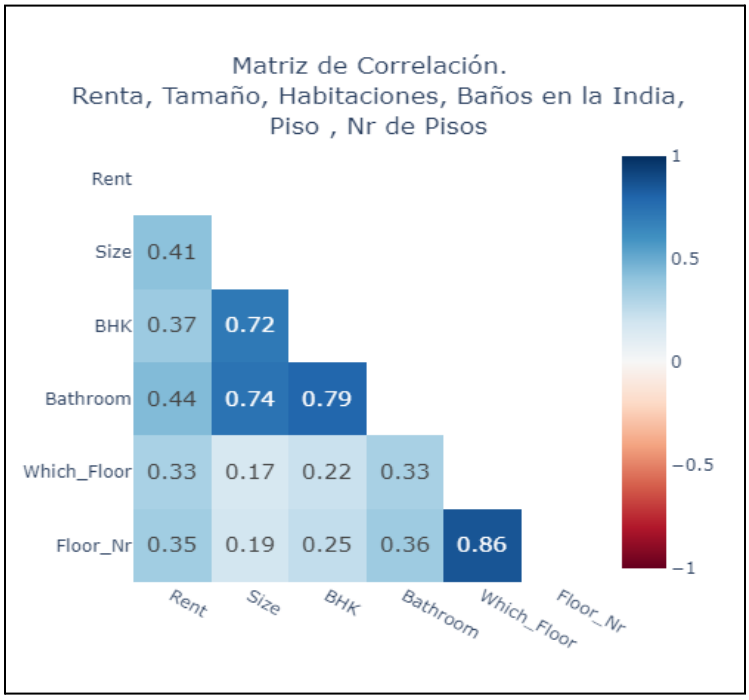
La localidad o ubicación del inmueble no es la única característica que puede influir sobre la renta, se procede a analizar en paralelo la renta de los inmuebles agrupados por ciudad (para comparar), por Tipo de Área, Amueblado, Tipo de Contacto y Tipo de Inquilino; obteniendo los siguientes insights:



1. La muestra está relativamente balanceada al agruparse por ciudad, es decir, no existen diferencias significativas en el número de inmuebles por ciudad.
2. Los inmuebles vendidos por un agente inmobiliario tienen precios significativamente más elevados que aquellos que son vendidos por los propios dueños. No obstante, estos inmuebles no representan a la mayoría de inmuebles en la muestra.
3. Mientras más amueblado sea el inmueble, mayor es la renta. Sin embargo, la mayoría de los inmuebles en la muestra son semi-amueblados.
4. Los apartamentos donde se prefieren sólo familias, son los que exigen mayores rentas, pero la gran mayoría de los dueños de los inmuebles son indiferentes a la hora de rentar su casa a una familia o a un estudiante.
5. La gran mayoría de los inmuebles están calculados como “Carpet Area” o “Super Area”. No obstante, esta característica demostró ser poco útil para explicar la renta posteriormente.

5.4 Correlación variables numéricas

El tamaño del inmueble, el número de habitaciones, el número de baños, el piso y la cantidad de pisos totales tienen una correlación débil con respecto a la renta, sin embargo, el tamaño y el número de baños son aquellas que obtuvieron valores más elevados (de 0,41 y 0,35 respectivamente). Las variables más correlacionadas fueron el número de habitaciones, el número de baños y el tamaño; explicado de manera lógica, ya que en la medida que aumentan los números de habitaciones o baños, aumenta sin duda alguna el tamaño en pies cuadrados del inmueble. Otra variable altamente correlacionada es el número de pisos totales del inmueble y el piso del inmueble; esto sucede porque, en la medida que el piso del inmueble se encuentre más elevado, sin duda alguna, el número total de pisos de ese edificio serán mayores.



5.4 Número de Pisos

Se observó que en la medida que aumenta el tamaño (en pisos) de un inmueble, la renta asciende notoriamente, asimismo aquellas muestras ubicados en pisos más altos suelen tener mayor renta. La mayoría de las ofertas de alquiler se encuentran en los pisos más bajos. (Ver apéndice Nr)

5.5 Conclusiones EDA

En términos generales podemos observar una renta asimétrica distribuida hacia la izquierda, la cual aumenta notoriamente dependiendo de la ciudad, la persona contacto y el amueblado, lo cual intuitivamente tiene sentido. En primera instancia, la ubicación de un inmueble no puede cambiar, por lo que puede existir cambios importantes en el valor de los mismos dependiendo de su ubicación. Las demás conclusiones se podrían explicar por un factor económico: Existen un mayor número de personas con bajos o medianos recursos que personas de altos recursos. Esta característica es notoria en sociedades de contrastes socioeconómicos marcados como la India. Esto explica como la renta se concentra hacia la izquierda, dejando con menor frecuencia aquellos inmuebles con valores muy altos en renta. Las personas con mayores recursos también pueden disponer del dinero para contratar a un agente inmobiliario para la venta de su inmueble (adicionando al valor el costo del agente inmobiliario), elevando el costo de la renta. El amueblado es un valor agregado que incrementa el costo y finalmente la gran mayoría de los dueños de los inmuebles son indiferentes entre elegir una familia o un estudiante como inquilino. Los inmuebles de preferencia familiar suelen ser más caros.

El tamaño del inmueble tiene relación con la renta, pero no tan fuerte como se suponía en un principio. Lógicamente, en la medida que aumentamos la cantidad de habitaciones en un inmueble, mayor será el tamaño del mismo.

6. Modelo MVP

Para establecer nuestro modelo base o MVP, se prueban modelos con la configuración base (o default) de los algoritmos de la librería de Scikit-Learning. Dado que la variable a predecir es la Renta (una variable continua), los algoritmos a elegir tienen que ser de regresión.

6.1 Variables elegidas

Se utilizan todas las variables del dataset a excepción de “Posted On”, ya que solo nos dice en qué momento se tomó la muestra, más no aporta en la predicción de la variable dependiente.

6.2 Feature Engineering

Como los modelos de machine learning solo funcionan con variables numéricas, se procede a realizar técnicas de tratamiento de datos (Feature Engineering).

- Se divide la variable ‘Floor’ en 2 variables nuevas, la primera llamada “Which_Floor” que determina el piso donde se encuentra el alquiler del

inmueble; y la segunda llamada “Floor_Nr” para determinar el número total de pisos que posee el inmueble. Se utiliza la función de str.split de Python para realizar la división.

- Se borran las variables “Posted On”, “Area Locality” y “Floor”.
- Se utiliza la función de One Hot Encoding de la librería de pandas en Python para convertir las columnas “City” y “Area Type” en múltiples columnas con valores 1 y 0. De esta manera entender de manera binaria y numérica cuando una ciudad o un tipo de área está presente en la muestra.
- A las variables “Point of Contact” y “Furnishing Status” se les aplica un tratamiento de Label Encoding acorde a las rentas medianas agrupadas por cada variable. **Ejemplo**: La renta mediana de cada “Point of contact” tiene una relación porcentual con respecto a la suma de las medianas de 7 (Contact Agent), 1(Contact Builder) y 2 (Contact Owner). Esto le indica al modelo que cada vez que se evalúe la opción “Contact Agent”, se le va a dar un peso 7 veces superior que a la opción de “Contact Builder” para predecir la renta.

6.3 Algoritmos de Regresión

Se utilizan para la predicción de la renta los siguientes algoritmos:
1) Regresión Lineal Multiple (RLM).
2) Regresión de Árboles de Decisión (DT),
3) Regresión de Random Forest (RFR) y
4) Regresión de Support Vector Machine (SVM_R).

Se divide el dataset tomando una división train/test de 70/30 y se entrenan los modelos a través de la librería de scikit-learn de python, utilizando los valores predeterminados de cada algoritmo. No se hacen optimizaciones puesto que en esta primera fase solo se busca encontrar de manera generalizada cuál de los modelos obtiene mejores métricas.

Los resultados fueron los siguientes:

Métricas	MSE	MAE	MAPE	R2
RLM	1,395,013,848.8533	22,069.2112	1.2132	0.5574
DT	9,939,814,901.7114	16,626.7999	0.5263	-2.1535
RFR	1,199,840,962.5706	11,655.6062	0.3916	0.6193
SVM_R	3,463,754,078.1139	23,584.8039	0.6314	-0.0989

El mejor performance fue del algoritmo de Regresión de random forest, el cual presentó un mayor R2 y menores valores de métricas de error (MSE, MAE y MAPE). En base a este primer resultado, se toma como el modelo

6.4 Tratamiento de Outliers

Al tener la data original muchos outliers con valores muy por encima de la media, se toma la decisión de borrarlos, utilizando la técnica de IQR. El dataset resultante tiene 4226 filas (520 menos que el original).
Como tenemos un dataset algo diferente, se aplican las mismas técnicas de one hot encoding, label encoding y el split 70/30 de los datos realizados anteriormente. Se obtuvieron los siguientes resultados:

Métricas	MSE	MAE	MAPE	R2
RLM	58,900,724.7429	5,394.4828	0.3231	0.6918
DT	80,516,960.6421	6,111.1483	0.3622	0.5787
RFR (MVP)	47,377,779.9594	4,766.0439	0.2892	0.7521
SVM_R	211,553,708.9196	9,715.4446	0.5432	-0.1068

El rendimiento de todos los modelos mejoró notoriamente, menos el SVM_R. El Random Forest Regressor sigue mostrando mejores métricas que los otros algoritmos. El Random Forest Regressor es nuestro algoritmo elegido y nuestro modelo MVP o modelo base. A partir de aquí se hacen sólo optimizaciones con el algoritmo de Regresión de Random Forest.

7. Optimización con Feature Engineering

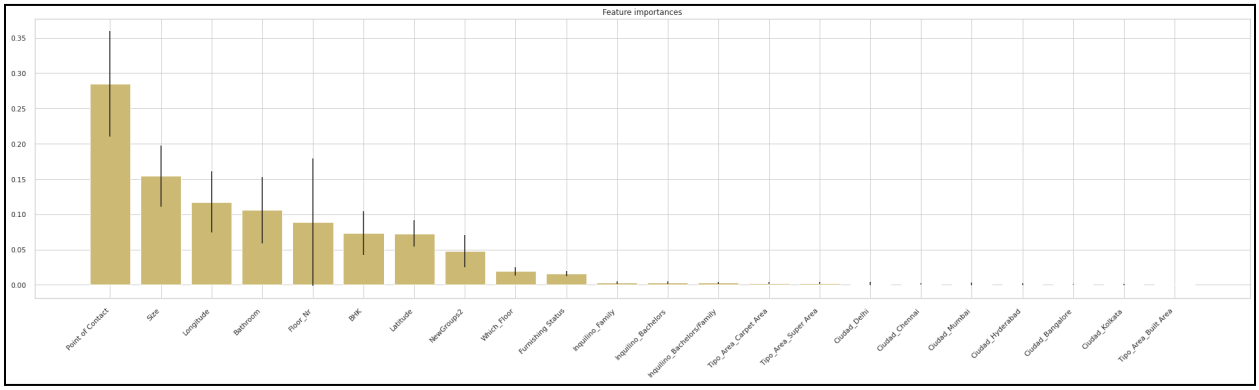
7.1 Adición de variables de localización

Con el objetivo de mejorar el performance del algoritmo elegido se crea, a través del Add-On de google sheets, la latitud y longitud de cada uno de los inmuebles del dataset. Google maps tiene una API propia, que permite realizar este trabajo directamente en Python, sin embargo, no fue viable por el costo que implica extraer esta data directamente de la aplicación de google maps. Estos datos adicionales se agregan a la tabla principal de nuestro dataset.

Adicionalmente, para cada ciudad, se utiliza el algoritmo de cluster de K Means, para generar grupos a través de la latitud y longitud. Con el método del codo se eligen la cantidad óptima de clusters para cada ciudad. Finalmente se crea una columna denominada “NewGroups2” donde están todos grupos creados de cada ciudad.
Se utilizan las mismas técnicas de Feature Engineering del modelo MVP y , adicionalmente, se utiliza el algoritmo de Label Encoder de la librería Scikit-Learning para asignar un número único a cada clúster de la columna “NeGroups2”. Utilizando el algoritmo de regresión de Random Forest obtenemos los resultados (columna RFR(Nuevas Variables)) y comparamos con el modelo anterior (RFR (MVP)):

Métrica	RFR(MVP)	RFR(Nuevas Variables)
MSE	47,947,699.5469	38,007,387.3638
MAE	4,799.4751	4,156.3990
MAPE	0.2909	0.2576
R2	0.7491	0.7996

Con respecto al modelo anterior, el nuevo modelo incrementó en 0,05 el R2, disminuyendo levemente su MAPE en 0.04. El MAE y MSE si obtuvieron cambios más notorios. Luego se utiliza la opción de “feature importances” del algoritmo de Regresión de Random Forest, que nos permite encontrar las variables más influyentes en nuestro modelo..



La variable con mayor relevancia (FIR = Feature Importance Ranking) fueron:

1. El tipo de contacto (Point of Contact): FIR = 0.156882
2. El tamaño del inmueble (Size): FIR= 0.121946
3. La longitud del inmueble (Longitude): FIR= 0.106147
4. El número de baños (Bathroom): FIR= 0.090413
5. El número de pisos del inmueble (Floor_Nr): FIR = 0.077864

6. La cantidad de habitaciones (BHK): FIR= 0.073868
7. Los clusters creados agrupados (NewGroups2): FIR= 0.050238
8. El piso donde se ubica el inmueble (Which_Floor): FIR= 0.050238
9. Estatus del amueblado (Furnishing Status): FIR= 0.050238

Las demás variables tienen una importancia menor a 0,0032.

7.2 Eliminación por “Feature Importance”

Se eliminan las variables del modelo que tuvieron una importancia menor a 0,0032 del modelo anterior. Estas variables son:

- Inquilino_Family(0.003140)
- Inquilino_Bachelors(0.003050)
- Inquilino_Bachelors/Family(0.002940)
- Tipo_Area_Carpet Area(0.002636)
- Tipo_Area_Super Area(0.002451)
- Ciudad_Delhi(0.001383)
- Ciudad_Chennai(0.001017)
- Ciudad_Mumbai(0.000946)
- Ciudad_Hyderabad(0.000821)
- Ciudad_Bangalore(0.000543)
- Ciudad_Kolkata(0.000462)
- Tipo_Area_Built Area(0.000007)

Corremos el algoritmo de Random Forest obteniendo los resultados de la última columna. Se compara con los anteriores.

Métricas	RFR (MVP)	RFR(Nuevas Variables)	RFR(Eliminando variables por Feature Importance)
MSE	47,947,699.5469	38,007,387.3638	38,502,020.8959
MAE	4,799.4751	4,156.3990	4,200.9349
MAPE	0.2909	0.2576	0.2613
R2	0.7491	0.7996	0.7970

Las métricas desmejoran levemente, lo que nos permite definitivamente eliminar estas variables y tener un modelo más simplificado. Esto adicionalmente nos permite en cierta medida evitar el “overfitting” del modelo.

8. Métricas de desempeño del modelo

De manera resumida, evaluamos el desempeño de todos los modelos entrenados:

- **Modelos con división del dataset (con outliers) Train/Test en 70/30**

Métricas	RLM	DT	RFR	SVM_R
MSE	1,395,013,848.8533	9,939,814,901.7114	1,199,840,962.5706	3,463,754,078.1139
MAE	22,069.2112	16,626.7999	11,655.6062	23,584.8039
MAPE	1.2132	0.5263	0.3916	0.6314
R2	0.5574	-2.1535	0.6193	-0.0989

- **Modelos con división del dataset (sin outliers) Train/Test en 70/30**

Métricas	RLM	DT	RFR*	SVM_R
MSE	58,900,724.7429	80,516,960.6421	47,377,779.9594	211,553,708.9196
MAE	5,394.4828	6,111.1483	4,766.0439	9,715.4446
MAPE	0.3231	0.3622	0.2892	0.5432
R2	0.6918	0.5787	0.7521	-0.1068

***El RFR se convierte en nuestro modelo MVP.**

- **Optimización del algoritmo Random Forest Regressor con Feature Engineering.**

Métricas	RFR(MVP)	RFR (Nuevas Variables)	RFR(Eliminando variables por Feature Importance)
MSE	47,377,779.9594	38,007,387.3638	38,502,020.8959
MAE	4,766.0439	4,156.3990	4,200.9349
MAPE	0.2892	0.2576	0.2613
R2	0.7521	0.7996	0.7970

- **Modelos con otras técnicas de optimización:** Estos modelos se entrenan estandarizando el dataset y luego aplicando técnica de reducción de dimensionalidad PCA.

Métricas	RFR(Estandarizado)	RFR (Con PCA)
MSE	38,530,168.0873	57,982,120.3757
MAE	4,201.1148	5,209.2874
MAPE	0.2614	0.3212
R2	0.7969	0.6943

Ni la estandarización del dataset ni la reducción de dimensionalidad mejora el performance, por lo tanto, el modelo elegido de todas estas iteraciones es el **RFR(Eliminando variables por Feature Importance)**.

9. Optimización por Hiperparámetros

Una vez elegido el modelo, sólo nos queda optimizar los hiperparámetros. Para ahorrar tiempo en las iteraciones, en la búsqueda de optimizar hiperparámetros, se utilizó el algoritmo Halving GridSearchCV de la librería Scikit-Learning. Este algoritmo nos permite buscar entre un set de listas de hiperparametros los que permiten optimizar una métrica específica. Para la búsqueda se utilizó el MAE como métrica de optimización. Adicionalmente se utiliza el Cross Validation que por default tiene el algoritmo, es decir, 5. Se tienen los siguientes resultados:

Optimización de Hiperparámetros 1

Hiperparámetros elegidos por el algoritmo:

- 'n_estimators': [100, 150, 200, 250],
- 'criterion': ['squared_error', 'absolute_error', 'friedman_mse','poisson'],
- 'max_depth': [None,3,5,7]

Resultados del algoritmo:

- 'Mejores parámetros {'criterion': 'poisson', 'max_depth': None, 'n_estimators': 250}
- Mejor Score CV -4309.177091823777
- MAE del modelo = 4176.92195
- r2 del modelo = 0.79744

Optimización de Hiperparámetros 2

Hiperparámetros elegidos:

- 'n_estimators': [100, 150, 200, 250],
- 'criterion': ['squared_error', 'absolute_error', 'friedman_mse','poisson'],
- 'max_depth': [None,3,5,7],
- 'max_features': ['auto','sqrt','log2']
- RFR = RandomForestRegressor(random_state = 21)

Resultados del algoritmo:

- 'Mejores parametros {'criterion': 'poisson', 'max_depth': None, 'max_features': 'log2', 'n_estimators': 250}
- Mejor Score CV -4201.578415973703
- MAE del modelo = 4172.4314
- r2 del modelo = 0.799

La segunda optimización nos dió mejores resultados, mejorando el R2 y el MAE.

9. Métricas finales del modelo optimizado

Con los resultados obtenidos anteriormente, se elige el segundo modelo con ajustes de hiperparámetros, las métricas de este modelo se observan en la última columna de la siguiente tabla:

	RFR(MVP)	RFR (Nuevas Variables)	RFR(Eliminando variables por Feature Importance)	RFR_Hiperparametros_1	RFR_Hiperparametros_2
MSE	47,947,699.5469	38,007,387.3638	38,502,020.8959	38,419,958.0099	38,124,934.5789
MAE	4,799.4751	4,156.3990	4,200.9349	4,176.9219	4,172.4314
MAPE	0.2909	0.2576	0.2613	0.2568	0.2586
R2	0.7491	0.7996	0.7970	0.7974	0.7990

10. Futuras líneas

Para mejorar el modelo de predicción de la renta se necesita, en primera instancia, mayor cantidad de datos. Teniendo una base tan amplia como la de magicbricks.com, se pueden disponer de muchos más datos para mejorar la calidad del modelo.
Por otra parte, para buscar la latitud y longitud de los inmuebles, se puede usar directamente la API de google maps, pero teniendo en cuenta el costo asociado de la búsqueda.
Por último, también hay que considerar otras variables que puedan ser extraídas de la página web, que puedan mejorar o complementar el modelo.

11. Conclusiones

En el análisis descriptivo del dataset pudimos concluir en términos generales que las rentas pueden variar significativamente, principalmente, dependiendo de su localidad, el tipo de vendedor y el tipo de amueblado. Las rentas tienden a subir si están localizadas en la ciudad de mumbai, si el apartamento está completamente amueblado o si el tipo de vendedor es un agente inmobiliario.
Por otra parte, la renta se encuentra muy dispersa con respecto a su media generando una cantidad significativa de outliers. El 10% del dataset original son outliers.
Se entrenaron en un principio 4 modelos de machine learning siendo el elegido el Random Forest Regressor por tener mejores resultados en métricas de MSE, MAE, MAPE y R2. Todos los modelos se entrenan dividiendo data en 70/30 (70% de entrenamiento y 30% de testeo).
Se aplican principalmente 3 optimizaciones para mejorar las métricas del modelo.

1. Tratamiento de outliers: donde se eliminan de la data los outliers que generan distorsiones grandes en los datos, mejorando el r2 de 0,61 a 0,75.
2. Optimización por Feature Engineering: Se agregan las variables latitud, longitud y una agrupación por clusters “NewGroups2” (creada en base a la latitud y longitud). Adicionalmente se eliminan variables con Feature Importance menor a 0,032. Estos cambios mejoran la métrica MAE de 4.799 a 4.200 y otorgan al modelo un r2 de 0,79.
3. Se optimizan hiperparámetros del Random Forest Regressor con el algoritmo de Halving GridSearchCV obteniendo una mejora leve en nuestras métricas bajando el MAE a 4.172.

Las variables más influyentes en el modelo fueron el tipo de contacto (Point of Contact), el tamaño del inmueble (Size), la longitud del inmueble (Longitude), el número de baños (Bathroom), el número de pisos del inmueble (Floor_Nr), la cantidad de habitaciones (BHK).los clusters creados agrupados (NewGroups2), el piso donde se ubica el inmueble (Which_Floor) y Estatus del amueblado (Furnishing Status).