Introducción a Aprendizaje de Máquina
Evaluación de 3 algoritmos para la predicción de alquileres

Joaquín René Beck, Leandro Adolfo Flaschka, Carlos Ernesto Cortés Parra
{joaquinrbeck, leoflaschka, cecortes}@gmail.com

Agosto de 2021

Resumen. En este trabajo se analiza el desempeño de 3 algoritmos de Machine Learning para la predicción de alquileres en Brasil. En la primera parte se presenta un breve preprocesamiento del dataset para dejarlo operativo. La segunda parte corresponde a la medición de métricas de desempeño con parámetros por defecto implementado por la librería. En la tercera parte se aplica validación cruzada con dos métodos y finalmente se muestran los resultados de ese ajuste en los diferentes algoritmos.

Introducción.

El presente trabajo tiene por objetivo aplicar los conocimientos de Introducción a aprendizaje de Máquina, en inglés *Machine Learning* (ML), modificando características de los distintos métodos, atributos, algoritmos y aplicando razonamientos sobre los resultados obtenidos a fin de obtener conclusiones relevantes sobre el aprendizaje y la materia, para lo cual se seleccionó el problema de predecir el valor del precio de alquileres en Brasil (*Rent Amount*) partiendo de un dataset publicado en Kaggle [1].

En el proceso de aprendizaje, se hicieron pruebas y estudios con otros algoritmos algunos con resultados muy prometedores, un ejemplo de ello es el XGBRegressor, a fin de no complejizar más el desarrollo del trabajo y del análisis de dicho dataset, se decidió seleccionar tres (3) algoritmos supervisados: *Random Forest Regressor, Decision Tree Regressor y Support Vector Regression* y establecer la comparación a través de las métricas Mean Absolute Error (MAE), Mean Square Error (MSE) y el Coeficiente de determinación (R2).

Métricas.

R2 es una medida estadística que representa la proporción de la varianza para una variable dependiente que se explica por una variable independiente o variables en modelos de regresión1. Tiene por objetivo predecir resultados futuros. El coeficiente oscila entre 0 y 1 y en cuanto su valor se sitúe más cerca de 1, mayor es el ajuste del modelo a la variable y por el contrario si se sitúa más cerca de 0, el modelo es menos ajustado por lo tanto menos fiable.

MAE es una medida de errores entre observaciones emparejadas que expresan el mismo fenómeno. Los ejemplos de Y versus X incluyen comparaciones de tiempo predicho versus observado, tiempo subsiguiente versus tiempo inicial, y una técnica de medición versus una técnica alternativa de medición. Tiene la misma unidad que los datos originales y solo se puede comparar entre modelos cuyos errores se miden en las mismas unidades. Suele ser similar en magnitud al RMSE, pero un poco más pequeño[1].

MSE El error cuadrático medio (MSE) o la desviación cuadrática media (MSD) de un estimador (de un procedimiento para estimar una cantidad no observada) mide el promedio de los cuadrados de los errores, es decir, la diferencia cuadrática promedio entre los valores estimados y los valores reales. valor. MSE es una función de riesgo, que corresponde al valor esperado de la pérdida por

1 https://www.kaggle.com

error al cuadrado. El hecho de que el MSE sea casi siempre estrictamente positivo (y no cero) se debe a la aleatoriedad o a que el estimador no tiene en cuenta la información que podría producir una estimación más precisa. El MSE evalúa la calidad de un predictor (es decir, una función que asigna entradas arbitrarias a una muestra de valores de alguna variable aleatoria) o un estimador (es decir, una función matemática que asigna una muestra de datos a una estimación de un parámetro de la población). del cual se toman muestras de los datos). El MSE es una medida de la calidad de un estimador: siempre es no negativo y los valores más cercanos a cero son mejores [1].

Algoritmos.

Como se mencionó anteriormente, los algoritmos seleccionados fueron *Random Forest Regressor, Decision Tree Regressor y Support Vector Regression.* Dado que el dataset tiene características que definen cada entrada, pertenecen a la familia de los supervisados y como el objetivo es obtener un valor numérico entonces se seleccionan los que están dentro de la familia de Regresión. En la figura 1, se muestra la clasificación de los algoritmos de ML[2]

Figura 1: Clasificación de algoritmos de Machine Learning

Materiales.

El trabajo fue desarrollado en python, y como entornos de desarrollo se eligieron pyCharm[2] y Atom[3], con las librerías de scikit-learn[4]. El repositorio para trabajo colaborativo y versionado fue

2 https://www.jetbrains.com/es-es/pycharm/
3 https://atom.io/
4 https://scikit-learn.org/

Github[5]. El repositorio es https://github.com/joaquinrb/TrabajoIntegradorML donde se encuentra el archivo python del proyecto **Regresion3.py**, también se hicieron implementaciones en Colaboratory **IAMRegresión2.ipynb**. Dentro del repositorio está la carpeta Dataset que contiene el achivo **houses_to_rent_v2.csv**.

La estructura de desarrollo tiene las siguientes etapas

1 Breve análisis de datos, en inglés Exploration Data Analysis (EDA).

2 Preprocesamiento de la información.

3 Testeo de modelos

Dataset

El dataset seleccionado se tomó de kaggle [3]. Contiene 10692 inmuebles para alquilar y 13 características diferentes que se listan en la tabla 1.

Característica	Descripción
city	Ciudad donde se localiza la propiedad
area	Área de la propiedad
rooms	Cantidad de habitaciones
bathroom	Cantidad de baños
parking spaces	Cantidad de estacionamientos
floor	Piso
animal	Acepta animales
furniture	Está amoblado
hoa	Homeowners association tax
property tax	Impuesto de propiedad
rent amount	Precio de alquiler
fire insurance	Seguro contra incendios
total	Valor total

Tabla 1: Características del dataset

La información de los tipos de datos contenidos se muestran en la tabla 2. Una vista reducida del dataset se muestra en la tabla 3.

5 https://github.com/

```
 <<< Información del dataset >>>
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10692 entries, 0 to 10691
Data columns (total 13 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   city                 10692 non-null  object
 1   area                 10692 non-null  int64
 2   rooms                10692 non-null  int64
 3   bathroom             10692 non-null  int64
 4   parking spaces       10692 non-null  int64
 5   floor                10692 non-null  object
 6   animal               10692 non-null  object
 7   furniture            10692 non-null  object
 8   hoa (R$)             10692 non-null  int64
 9   rent amount (R$)     10692 non-null  int64
 10  property tax (R$)    10692 non-null  int64
 11  fire insurance (R$)  10692 non-null  int64
 12  total (R$)           10692 non-null  int64
dtypes: int64(9), object(4)
```

Tabla 2: Información del dataset

```
 <<< Primeras filas del dataset >>>
             city  area  rooms  bathroom  ...  rent amount (R$) fire insurance (R$) total (R$)
0         São Paulo    70      2         1  ...              3300                   42        5618
1         São Paulo   320      4         4  ...              4960                   63        7973
2      Porto Alegre    80      1         1  ...              2800                   41        3841
3      Porto Alegre    51      2         1  ...              1112                   17        1421
4         São Paulo    25      1         1  ...               800                   11         836
5         São Paulo   376      3         3  ...              8000                  121        8955
6    Rio de Janeiro    72      2         1  ...              1900                   25        2750
7         São Paulo   213      4         4  ...              3223                   41        7253
8         São Paulo   152      2         2  ...             15000                  191       16440
9    Rio de Janeiro    35      1         1  ...              2300                   30        2955
10        São Paulo    26      1         1  ...              2100                   27        2747
11         Campinas    46      1         1  ...               580                    8        1181
12        São Paulo    36      1         1  ...              2100                   27        2556
13        São Paulo    55      1         1  ...              4200                   54        5268
14        São Paulo   100      2         2  ...              4370                   56        5343
15         Campinas   330      4         6  ...              8000                  121        9129
16        São Paulo   110      2         2  ...              3000                   39        3861
17   Rio de Janeiro    88      2         3  ...              3500                   16        5351
18   Rio de Janeiro    56      2         1  ...              1220                   16        2036
19        São Paulo   600      4         5  ...             12000                  181       21680

[20 rows x 13 columns]
```

Tabla 3: Vista reducida del dataset

Breve análisis de datos

Para el análisis de datos, se tomaron algunas técnicas relevadas del kaggle [4][5]

Se hizo un breve análisis de los datos para seleccionar las características que mejor aplicaran a la predicción tomado como variable dependiente el precio del alquiler (rent amount (R$)) expresada en moneda brasilera (Reales) y las variables independientes.

En la tabla 4 se observan algunas estadísticas del dataset.

<<< Información estadística del dataset >>>

	count	mean	std	min	25%	50%	75%	max
area	10692.0	149.217920	537.016942	11.0	56.00	90.0	182.0	46335.0
rooms	10692.0	2.506079	1.171266	1.0	2.00	2.0	3.0	13.0
bathroom	10692.0	2.236813	1.407198	1.0	1.00	2.0	3.0	10.0
parking spaces	10692.0	1.609147	1.589521	0.0	0.00	1.0	2.0	12.0
hoa (R$)	10692.0	1174.021698	15592.305248	0.0	170.00	560.0	1237.5	1117000.0
rent amount (R$)	10692.0	3896.247194	3408.545518	450.0	1530.00	2661.0	5000.0	45000.0
property tax (R$)	10692.0	366.704358	3107.832321	0.0	38.00	125.0	375.0	313700.0
fire insurance (R$)	10692.0	53.300879	47.768031	3.0	21.00	36.0	68.0	677.0
total (R$)	10692.0	5490.487000	16484.725912	499.0	2061.75	3581.5	6768.0	1120000.0

Tabla 4: Estadísticos del dataset

El gráfico de la distribución de la renta (rent amount (R$)) se observa en la figura, 2

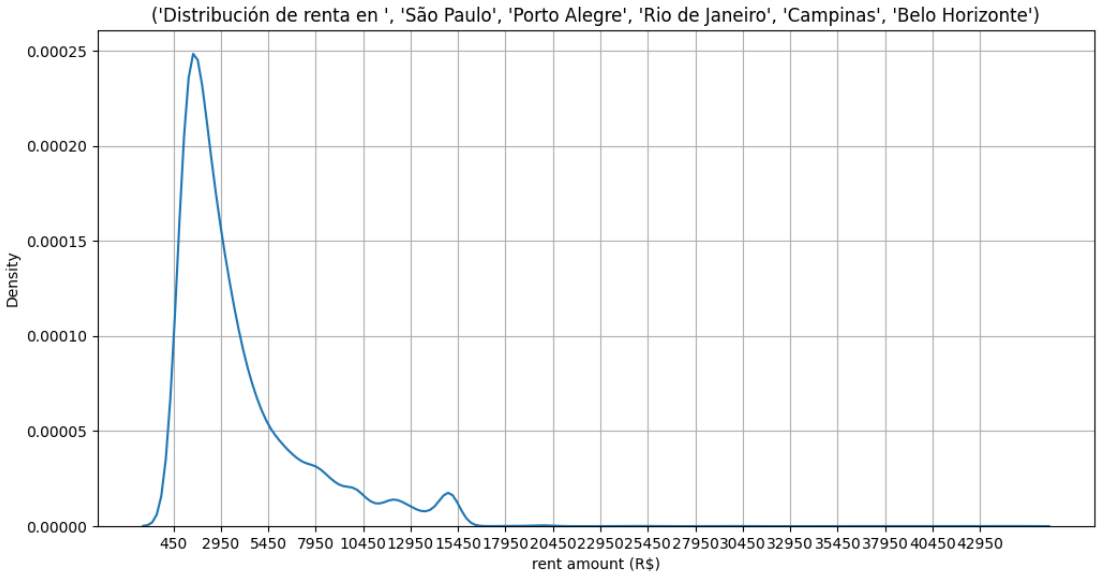

Figura 2: Distribución de la renta

Los valores más representativos están por debajo de 17000 (R$) ya que pocas propiedades están por encima de este valor.

Al tener una distribución de tipo normal (siendo asimétrica a derecha) se opta por modelos de regresión lineal, para predecir los valores de los alquileres.

Preprocesamiento de la información

Calculando la correlación de las características se muestra en la tabla 5 y la figura 3, de donde se concluye que las que tienen mayor correlación son ['city', 'rooms', 'bathroom', 'parking spaces', 'fire insurance (R$)', 'furniture'] que son las más relevantes para este trabajo.

```
 <<< Correlaciones >>>
                      area     rooms  bathroom  ...  property tax (R$)  fire insurance (R$)  total (R$)
area                 1.000000  0.193796  0.226766  ...           0.039059             0.188078    0.051799
rooms                0.193796  1.000000  0.733763  ...           0.075252             0.565148    0.134597
bathroom             0.226766  0.733763  1.000000  ...           0.109253             0.676399    0.208339
parking spaces       0.193983  0.617510  0.697379  ...           0.098378             0.597348    0.148684
hoa (R$)             0.006890  0.007139  0.050271  ...           0.007627             0.029535    0.955024
rent amount (R$)     0.180742  0.541758  0.668504  ...           0.107884             0.987343    0.264490
property tax (R$)    0.039059  0.075252  0.109253  ...           1.000000             0.105661    0.218344
fire insurance (R$)  0.188078  0.565148  0.676399  ...           0.105661             1.000000    0.254911
total (R$)           0.051799  0.134597  0.208339  ...           0.218344             0.254911    1.000000

[9 rows x 9 columns]
```

Tabla 5: Correlación de las características

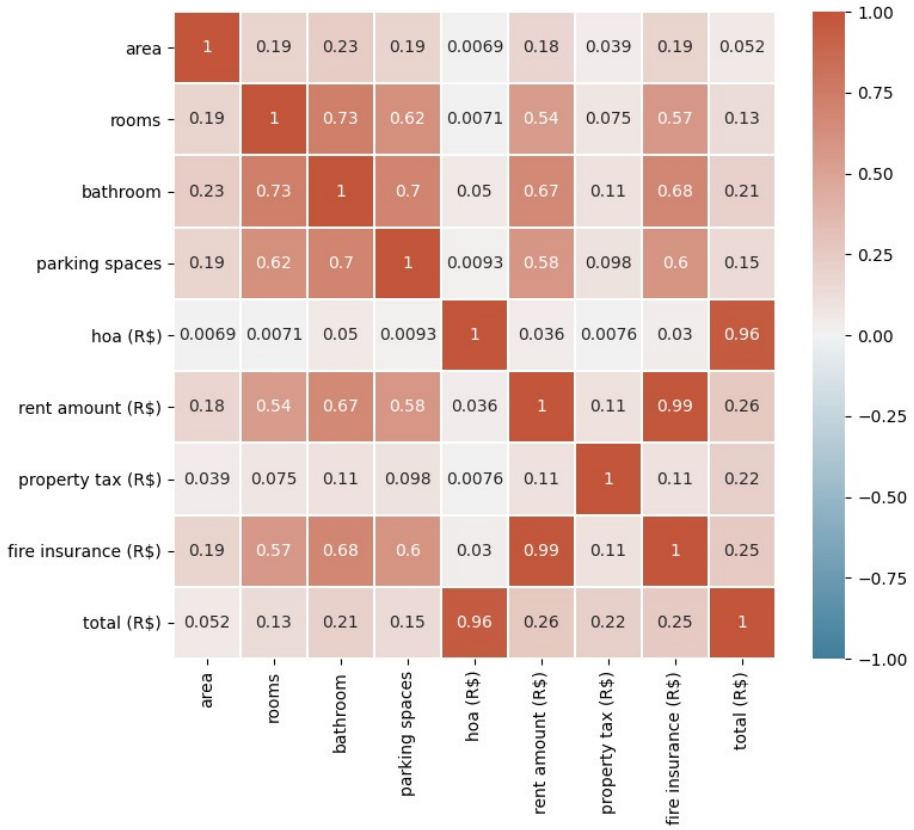

Figura 3: Gráfico de la correlación de las características

Además, se comenzó un análisis para tratar de determinar algunos valores atípicos (en inglés, *outliers*) y aunque no se profundizó sobre ésto en el presente trabajo, se muestra en la figura 4 uno de los gráficos obtenidos.

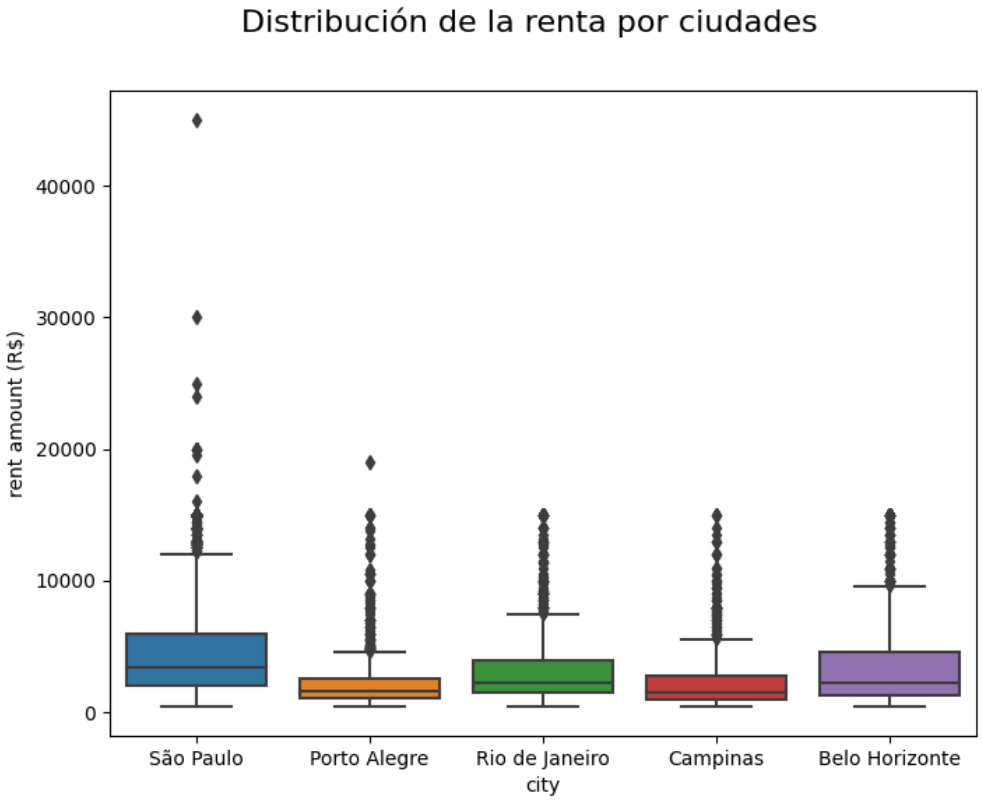

Figura 4: Distribución de renta por ciudades

Testeo de los modelos de predicción

El dataset se dividió en 70% para entrenamiento y 30% para testeo de forma aleatoria con el método **train_test_split**.

```
test_size = 0.3
X_train, x_test, y_train, y_test = train_test_split(x, y,test_size=test_size,
random_state = 123)
```

Para determinar la capacidad de predicción para cada uno de los modelos elegidos sin modificar sus parámetros (dejándolos por defecto), se utilizaron las siguientes métricas, considerando los y de prueba con los y estimados por cada modelo.

En la figura 5 se muestran las predicciones de los modelos con los parámetros por defecto con sus respectivas métricas.

Figura 5: Predicciones

	modelo	MAE	MSE	R2	Tiempo (s)
0	Random Forest Regressor	175.909499	429.699248	0.984609	0.982489
1	Decision Tree Regressor	178.441921	480.985231	0.980715	0.021878
2	Support Vector Regressor	1442.732043	2770.687065	0.360087	55.498540

Validación Cruzada. En inglés Cross Validation (CV).

Método *cross_val_score*

Random Forest Regressor. Para la CV se usó el método **cross_val_score**, cuyos resultados se muestran en la figura 6, determinando que el óptimo es 166. En la figura 6 se muestran los desempeños.

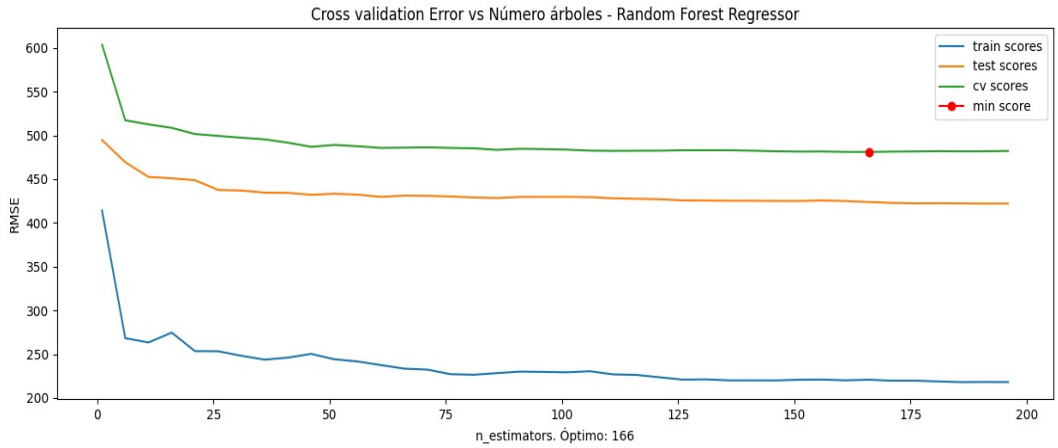

Figura 6: Cross Validation Error vs Número de árboles

Las métricas del modelo ajustado se muestran en la figura 7

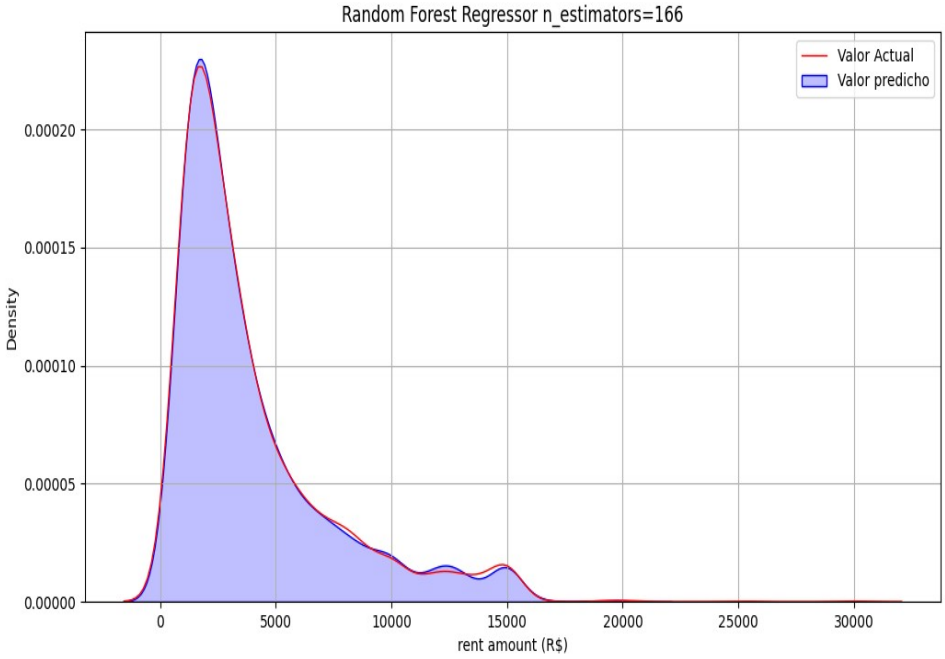

Figura 7: Random Forest Regressor ajustado

	modelo	MAE	MSE	R2	Tiempo (s)
0	Random Forest Regressor	175.478606	423.924229	0.98502	1.641192

Support Vector Regression. Usando el mismo método anterior, los desempeños se muestran en las figuras 8,9 y 10 donde se varía el kernel a rbf, poly y linear respectivamente.

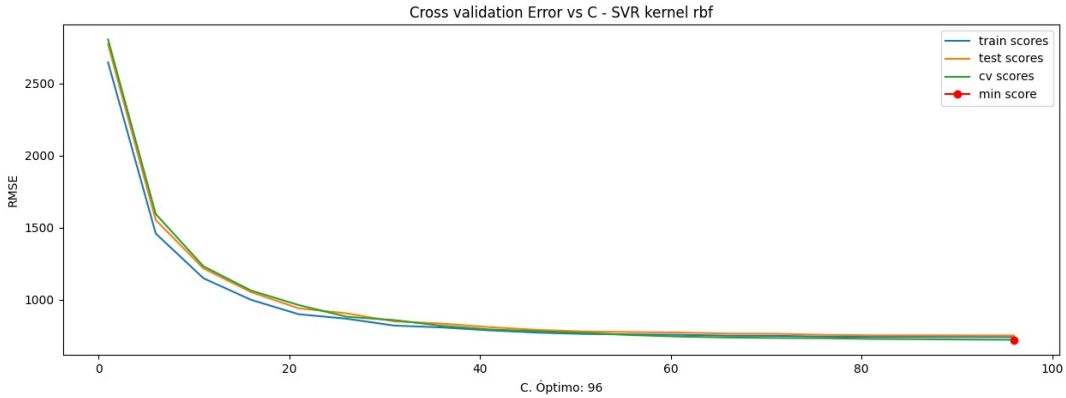

Figura 8: Cross Validation Error para SVR kernel 'rbf'

```
 <<< k-cross-validation SVR kernel "rbf" >>>
tiempo de ejecución: 1547.446444
C óptimo: 96
```

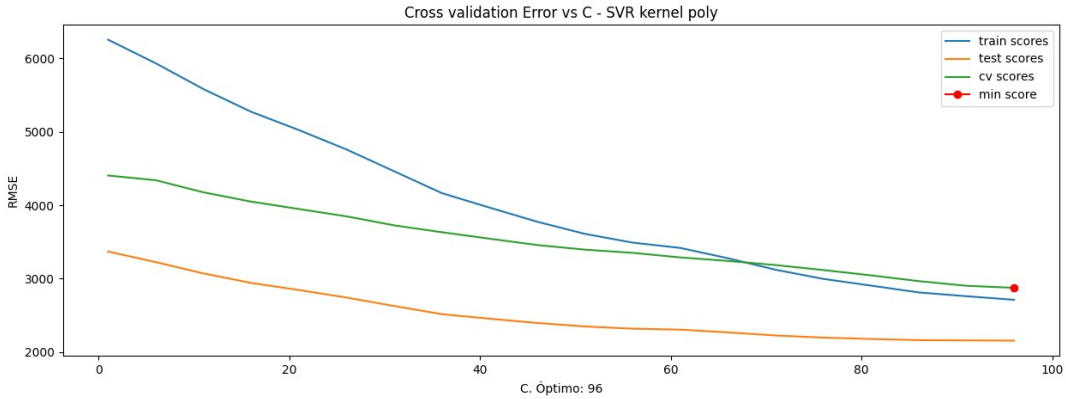

Figura 9: Cross Validation Error para SVR kernel 'poly'

```
 <<< k-cross-validation SVR kernel "poly" >>>
tiempo de ejecución: 249.317503
C óptimo: 96
```

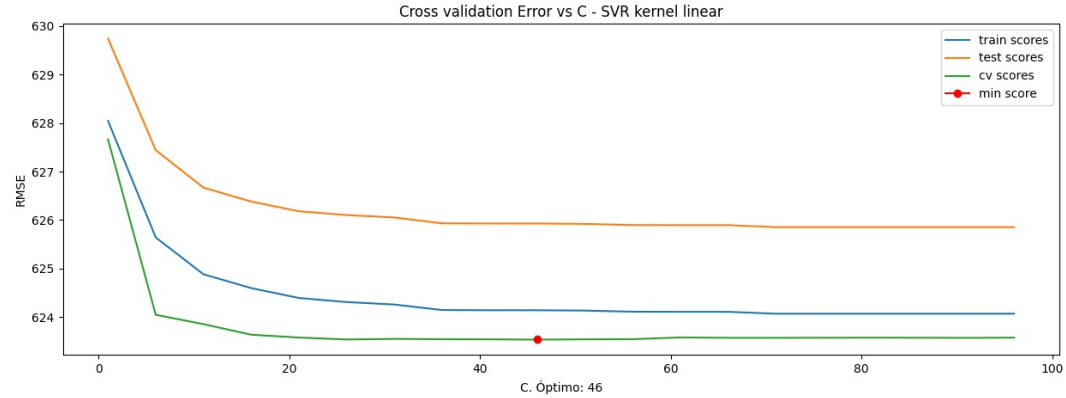

Figura 10: Cross Validation Error para SVR kernel 'poly'

```
 <<< k-cross-validation SVR kernel "linear" >>>
tiempo de ejecución: 4348.161241
C óptimo: 46
```

Las métricas con el modelo ajustado se muestran en la figura 11

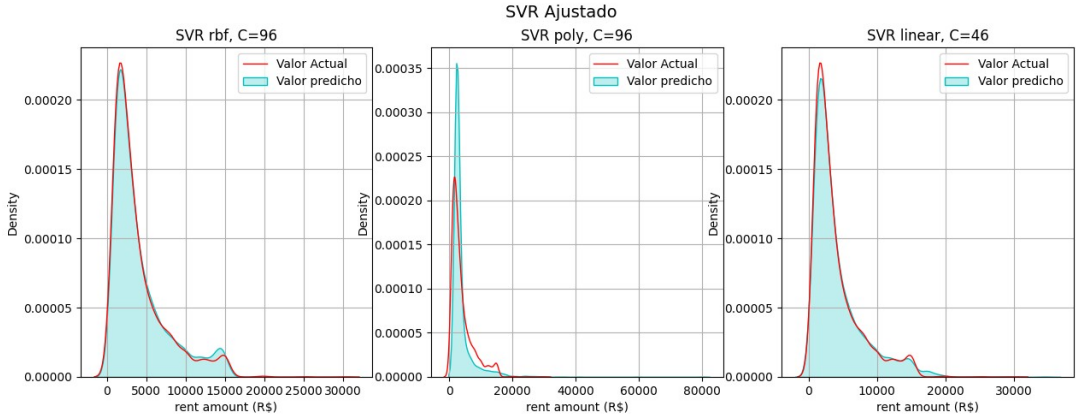

Figura 11: Support Vector Ajustado

	param	MAE	MSE	R2	Tiempo (s)
0	kernel=rbf, C=96	279.056914	752.805389	0.952760	5.019283
1	kernel=poly, C=96	1202.961887	2156.271959	0.612427	3.092627
2	kernel=linear, C=46	276.933038	625.928784	0.967342	60.290294

En la figura 12 Se muestran las comparaciones de los mejores desempeños de los algoritmos

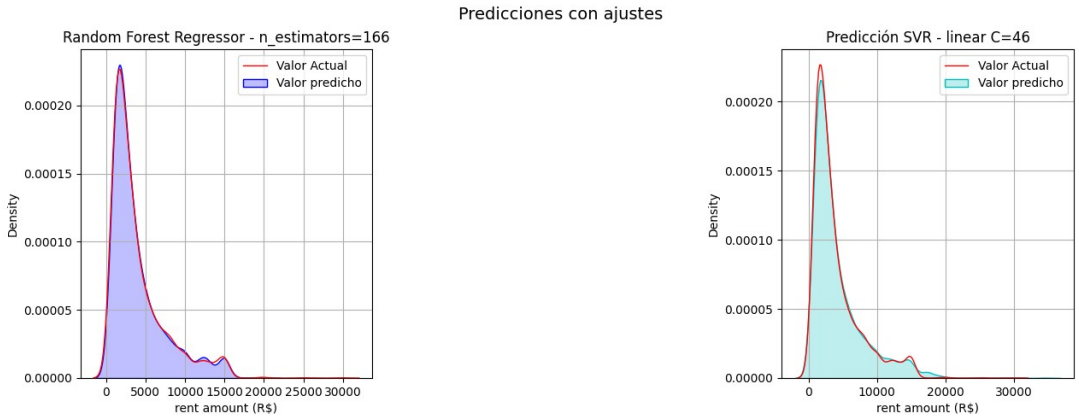

Figura 12: Comparación de modelos con ajustes

	modelo	MAE	MSE	R2	Tiempo (s)
0	Random Forest Regressor	175.478606	423.924229	0.985020	1.649966
1	Support Vector Regressor	276.933038	625.928784	0.967342	58.999699

Método *GridSearchCV.*

Se aplicó validación cruzada con **GridSearchCV**, que según la literatura es uno de los más recomendados para el ajuste de hiperparámetros.

Para Support Vector Regressor con kernel rbf y los parámetros de variación fueron C, epsilon y gamma:

```
{'C': [1.1, 5.4, 170, 1001],
 'epsilon':[0.0003, 0.007, 0.0109, 0.019, 0.14, 0.05, 8, 0.2, 3, 2, 7],
 'gamma': [0.7001, 0.008, 0.001, 3.1, 1, 1.3, 5]}
```

Los mejores parámetros obtenidos son:

```
Valor de costo y gamma óptimos: {'C': 1001, 'epsilon': 7, 'gamma': 0.001}
Accuracy asociado +- std: -488496.922 +- 315672.528
```

La figura 13 corresponde a las métricas del Support Vector Regressor con parámetros óptimos

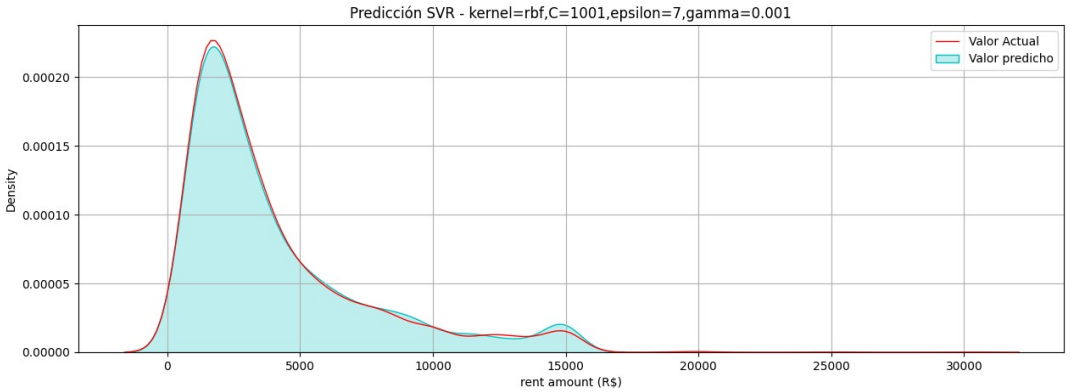

Figura 13: Support Vector Regressor con parámetros óptimos de GridSearchCV

	modelo	MAE	MSE	R2	Tiempo (s)
0	Support Vector Regressor	239.666507	705.027151	0.958566	4.926319

Para Decision Tree Regressor con:

```
{"criterion": ["mse", "mae"],
 "min_samples_split": [10, 20, 40],
 "max_depth": [2, 6, 8],
 "min_samples_leaf": [20, 40, 100],
 "max_leaf_nodes": [5, 20, 100],}
```

Los mejores parámetros obtenidos son:

```
{'criterion': 'mse', 'max_depth': 8, 'max_leaf_nodes': 100, 'min_samples_leaf':
20, 'min_samples_split': 10}
```

La figura 14 Corresponde a las métricas de Decision Tree Regressor con parámetros óptimos

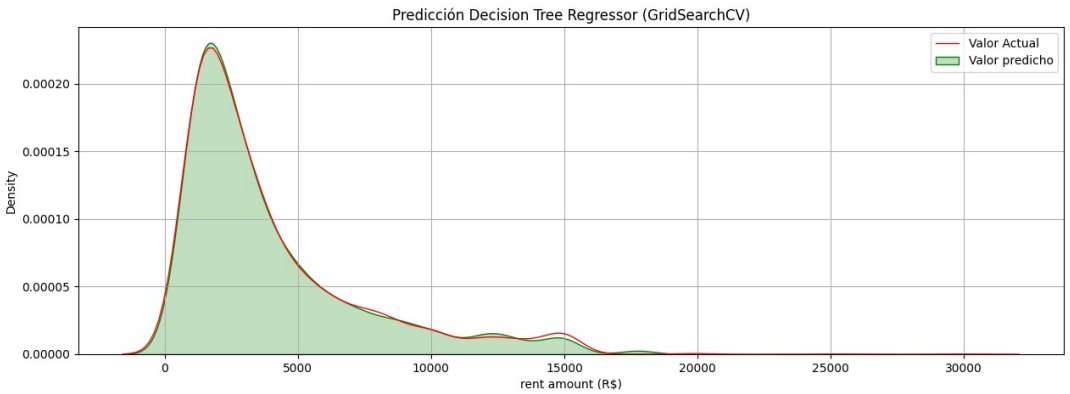

Figura 14: Decision Tree Regressor con parámetros óptimos de GridSearchCV

	modelo	MAE	MSE	R2	Tiempo (s)
0	Decision Tree Regressor	231.503088	513.980439	0.977979	0.008921

Conclusiones

De los tres algoritmos evaluados, se observa que el que mejor predice con el método cross_val_score y con parámetros por defecto es el Random Forest Regressor que siempre obtuvo un mejor R2.

En la validación cruzada con el método cross_val_score, se estimaron hiperparámetros que aportaron mejor desempeño. Para Support Vector Regressor con kerner linear, los consumos de recursos y tiempos de procesamiento fueron bastante elevados comparativamente, pero mostró una mejor predicción.

Con GridSearchCV, Random Forest Tree Regressor tmbién requirió de muchos recursos y tiempo de procesamiento, mientras que para Decision Forest Regressor los tiempos fueron significativamente menores. Si embargo, la elección de hiperparámetros fue determinante para una mejor obtención de métricas. En ambos algoritmos con parámetros obtenidos por este método se mejoró su capacidad predictiva.

Referencias

1.HIREGOUDAR, Shravankumar, Ways to Evaluate Regression Models, , https://towardsdatascience.com/ways-to-evaluate-regression-models-77a3ff45ba70

2.DOBILAS, Saul, k-Nearest Neighbors (kNN) — How To Make Quality Predictions With Supervised Learning?, 2021, https://towardsdatascience.com/k-nearest-neighbors-knn-how-to-make-quality-predictions-with-supervised-learning-d5d2f326c3c2

3.JUNIOR, Rubens, , 2021, https://www.kaggle.com/rubenssjr/brasilian-houses-to-rent?select=houses_to_rent_v2.csv

4.MENDES, Olavo, , 2021, https://www.kaggle.com/olavomendes/rental-prices-in-brazil

5.DI FANTE, Artur, , 2021, https://www.kaggle.com/arturlunardi/predicting-rental-prices-in-brazil