Seoul Bike Sharing Demand: Predecir Número De Bicicletas

María Fernanda Moreno Gómez

A01708653@tec.com

ABSTRACT Este documento presenta la implementación y evaluación de un modelo de Inteligencia Artificial para predecir la demanda de bicicletas compartidas en Seúl, utilizando el dataset "Seoul Bike Sharing Demand". El objetivo del modelo es ayudar a optimizar la distribución y disponibilidad de bicicletas en diferentes zonas de la ciudad, mejorando así la eficiencia del sistema de transporte. Se utilizó un modelo de regresión lineal multivariable, que fue entrenado con datos históricos sobre el uso de bicicletas, condiciones meteorológicas y factores estacionales. Los resultados indican que el modelo logra un coeficiente de determinación (R^2) de aproximadamente 0.51 en el conjunto de prueba, lo que sugiere que el modelo es capaz de explicar alrededor del 51% de la variabilidad en la demanda de bicicletas. Sin embargo, el error cuadrático medio al final del entrenamiento fue relativamente bajo, pero la precisión del modelo podría mejorarse mediante la implementación de técnicas más avanzadas o la incorporación de características adicionales.

Keywords: Bike Sharing, Seoul, Demand Prediction, Linear Regression, Data Preprocessing, Gradient Descent, Transportation Systems.

1. INTRODUCCIÓN

En los últimos años, los sistemas de bicicletas compartidas han evolucionado considerablemente, convirtiéndose en una parte fundamental de las redes de transporte urbano. Los sistemas de bicicletas compartidas comenzaron en Ámsterdam en 1965, con un enfoque sencillo y práctico: las personas pintaban sus bicicletas de color blanco, dejándolas libres para ser compartidas y posteriormente devueltas. Con el tiempo, este concepto evolucionó, introduciendo bicicletas diseñadas específicamente para ser utilizadas en áreas determinadas, las cuales podían desbloquearse en un sitio de préstamo de bicicletas y devolverse en otro sitio similar, haciendo que la disponibilidad de las bicicletas dependiera de la cantidad de bicicletas disponibles en cada estación.

A medida que estos sistemas se han expandido globalmente, se han convertido en una solución de movilidad clave en muchas ciudades, abordando problemas como la congestión del tráfico y la contaminación ambiental. De hecho, estudios recientes destacan cómo estos sistemas han sido adoptados en ciudades de todo el mundo, ofreciendo una alternativa ecológica al uso de automóviles privados y mejorando el acceso al transporte público [1]. Este crecimiento ha sido impulsado no solo por su conveniencia, sino también por sus beneficios ambientales, contribuyendo a la reducción de emisiones de carbono y promoviendo un estilo de vida más saludable entre los usuarios [1].

Además, la electrificación de los sistemas de bicicletas compartidas ha demostrado tener un impacto aún mayor en la

sostenibilidad urbana. Según PBSC Urban Solutions, la implementación de bicicletas eléctricas en estos sistemas no solo mejora la accesibilidad y comodidad para los usuarios, sino que también reduce significativamente la huella de carbono de las ciudades al disminuir las emisiones de CO2 [2]. Estos avances subrayan la importancia de seguir innovando en el diseño y la operación de sistemas de bicicletas compartidas para maximizar sus beneficios ambientales y sociales.

Sin embargo, a pesar de sus beneficios, la efectividad de los sistemas de bicicletas compartidas depende en gran medida de la capacidad para predecir la demanda en diferentes áreas y momentos del día. Esto es crucial para garantizar que las bicicletas estén disponibles donde y cuando se necesiten, optimizando así la eficiencia del sistema. Es aquí donde los modelos de inteligencia artificial, como los de regresión lineal multivariable, juegan un papel fundamental. Al analizar datos históricos sobre el uso de bicicletas, condiciones meteorológicas y factores estacionales, estos modelos pueden predecir la demanda con precisión, permitiendo una mejor planificación y gestión de los recursos del sistema.

El objetivo de este trabajo es implementar y evaluar un modelo de inteligencia artificial para predecir la demanda de bicicletas compartidas en Seúl, utilizando el dataset "Seoul Bike Sharing Demand". Este modelo busca mejorar la distribución y disponibilidad de bicicletas, contribuyendo así a un sistema de transporte urbano más eficiente y sostenible.

El dataset "Seoul Bike Sharing Demand" contiene datos registrados por hora sobre la demanda de bicicletas compartidas en Seúl, Corea, desde el 1 de diciembre de 2017 hasta el 30 de noviembre de 2018. Estos datos incluyen variables como las condiciones meteorológicas, la hora del día, y factores estacionales, todos los cuales son cruciales para predecir con precisión la demanda de bicicletas [3]. El propósito de este estudio es utilizar las instancias más relevantes de este dataset para entrenar un modelo de predicción que aproveche estas características, permitiendo una estimación más precisa del número de bicicletas necesarias en diferentes momentos y lugares de la ciudad. Este enfoque busca optimizar la operación del sistema de bicicletas compartidas en Seúl, asegurando que las bicicletas estén disponibles donde y cuando se necesiten, y contribuyendo a un sistema de transporte urbano más eficiente y sostenible.

2. DESCRIPCIÓN DEL DATASET

El dataset *"Seoul Bike Sharing Demand"* proporciona un registro detallado de la demanda de bicicletas compartidas en la ciudad de Seúl, Corea del Sur. Los datos abarcan un periodo de un año completo, desde el 1 de diciembre de 2017 hasta el 30 de noviembre de 2018, y están organizados en registros horarios, sumando un total de 8760 instancias.

2a.- Features

El dataset incluye 14 columnas, cada una representando una característica relevante para la predicción de la demanda de bicicletas. A continuación, se describen las características disponibles:

- **Date:** La fecha en formato DD/MM/YYYY, representando el día específico del registro.
- **Rented Bike Count**: El número de bicicletas rentadas durante la hora específica (variable objetivo).
- **Hour**: La hora del día (0-23), representando la hora específica del registro.
- **Temperature(C)**: La temperatura en grados Celsius.
- **Humidity(%)**: El porcentaje de humedad relativa.
- **Wind speed (m/s)**: La velocidad del viento en metros por segundo.
- **Visibility (10m)**: La visibilidad en decenas de metros (por ejemplo, 2000 = 20 km).
- **Dew point temperature(C)**: La temperatura del punto de rocío en grados Celsius.
- **Solar Radiation (MJ/m2)**: La radiación solar medida en megajulios por metro cuadrado.
- **Rainfall(mm)**: La cantidad de precipitación en milímetros.
- **Snowfall (cm)**: La cantidad de nieve en centímetros.
- **Seasons**: La estación del año en la que se registró la instancia, categorizada como: Winter, Spring, Summer, Autumn.
- **Holiday**: Indica si el día del registro fue un día festivo (‘Holiday’) o no (‘No Holiday’).
- **Functioning Day**: Indica si el sistema de bicicletas estaba en funcionamiento (‘Yes’) o no (‘No’) durante esa hora específica.

[3]

3. EXTRACCIÓN Y LIMPIEZA DE DATOS

La extracción de datos implica recuperar datos de fuentes con el fin de analizarlos [4]. En este caso, se aplicaron técnicas de procesamiento de datos al dataset *"Seoul Bike Sharing Demand"* con el objetivo de optimizar la precisión del modelo de predicción.

3a.- Carga y filtrado de datos

El primer paso en el preprocesamiento fue la carga del dataset y la revisión de datos faltantes. Se verificó que no había valores nulos en ninguna de las columnas, lo cual simplificó el proceso de limpieza de datos. Además, se verificó de igual manera que no existieran instancias repetidas o instancias con valores anormales (esto se obtuvo obteniendo las estadísticas de los features numéricos), de manera que no sucedió y se tuvo certeza que el dataset estaba correcto a nivel datos. A continuación, se realizó una exploración inicial de las características presentes en el dataset.

Durante esta exploración, se observó que la columna ‘Date’ contenía valores que se repetían cada hora para cada día. Esto implicaba que la fecha no aportaba información adicional para el análisis predictivo, ya que la demanda de bicicletas estaba registrada a nivel de hora y las diferencias entre fechas ya estaban capturadas por la variable ‘Hour’.

Posteriormente, las columnas ‘Functioning Day’ y ‘Holiday’ también fueron eliminadas. La columna ‘Functioning Day’ fue eliminada porque se observó que su valor estaba directamente relacionado con el conteo de bicicletas rentadas (‘Rented Bike Count’). Específicamente, cuando el conteo de bicicletas rentadas era 0, la columna ‘Functioning Day’ siempre

indicaba "No", lo que implicaba una redundancia en los datos. Esta relación directa no aportaba información adicional al modelo, por lo que se decidió prescindir de esta columna para evitar la redundancia y simplificar el modelo.

En cuanto a la columna 'Holiday', se decidió eliminarla porque su impacto podría estar indirectamente representado a través de otras variables, como las estaciones del año ('Seasons'). Además, el número de registros correspondientes a días festivos era significativamente menor en comparación con los días no festivos, lo que podría haber introducido un sesgo en el modelo. Dado que los datos sobre las estaciones ya proporcionan información relevante sobre patrones estacionales, se consideró que la columna 'Holiday' no era esencial para el análisis.

Adicionalmente, se observó que el conteo de bicicletas rentadas ('Rented Bike Count') presentaba una distribución altamente sesgada hacia valores bajos, como se muestra en la Figura 1 (el histograma de la distribución de 'Rented Bike Count'). Para evitar que los pocos valores extremadamente altos sesgaran el modelo, se decidió filtrar el dataset para excluir las instancias donde el conteo de bicicletas rentadas superaba los 2500. Esta decisión se basó en la observación de que las instancias con valores superiores a 2500 eran mínimas y podrían influir desproporcionadamente en el entrenamiento del modelo.

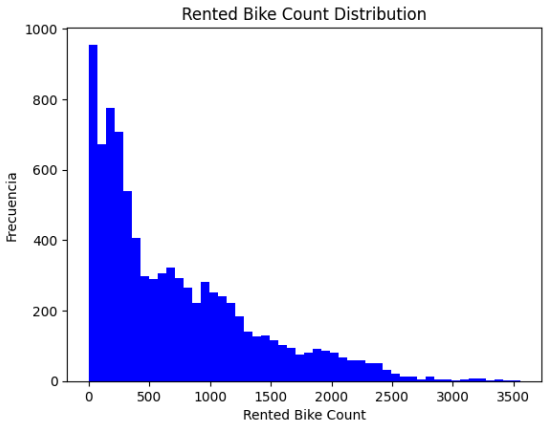


Figura 1: Histograma de la distribución de Rented Bike Count

El filtrar estos outliers es una práctica que ayuda a eliminar valores atípicos que influyan de manera negativa al modelo predictivo, pues incrementan la variabilidad de nuestros datos, reduciendo el poder estadístico, es por eso que, filtrar los datos atípicos puede hacer que los resultados se vuelvan estadísticamente significativos. [5].

3b.- Análisis de Correlación y Selección de Características

Para identificar las variables que tienen una relación lineal más fuerte con la variable objetivo 'Rented Bike Count', se calculó una matriz de correlación. Esta matriz muestra los coeficientes de correlación de Pearson entre cada par de variables, lo que indica el grado de asociación lineal entre ellas.

El coeficiente de correlación de Pearson varía entre -1 y 1. Un valor de 1 indica una correlación positiva perfecta, es decir, cuando una variable aumenta, la otra también lo hace de manera proporcional. Un valor de -1 indica una correlación negativa perfecta, donde un aumento en una variable corresponde a una disminución en la otra. Un valor cercano a

0 sugiere que no existe una relación lineal significativa entre las variables [6].

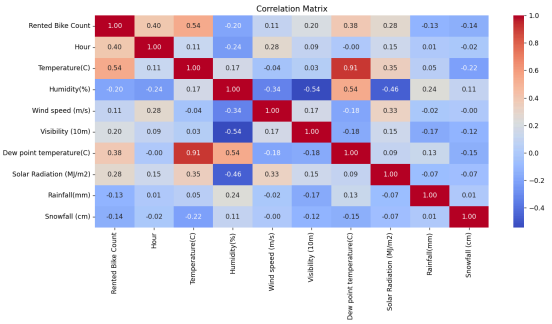


Figura 2: Matriz de correlación entre los features del dataset

Al observar la matriz de correlación generada (ver Figura 2), se identificaron las siguientes relaciones clave:

- 'Humidity(%)': Muestra una correlación negativa débil con Rented Bike Count (-0.20), lo que sugiere que la humedad no tiene un impacto significativo en la demanda de bicicletas. Además, esta variable está altamente correlacionada con otras como 'Dew point temperature(C)' (0.91), lo que podría introducir redundancia en el modelo.
- 'Wind speed (m/s)': Tiene una correlación muy baja con Rented Bike Count (0.11), lo que indica que no es un factor relevante en la predicción de la demanda de bicicletas.
- 'Rainfall(mm)' y 'Snowfall(cm)': Ambas variables muestran correlaciones negativas débiles con Rented Bike Count (-0.13 y -0.14, respectivamente), lo que indica que, aunque podrían tener algún impacto en el uso de bicicletas, no son factores determinantes en el contexto de este análisis.
- 'Visibility (10m)': Aunque tiene una correlación ligeramente positiva (0.20), su efecto es menor en comparación con otras variables seleccionadas, lo que llevó a su exclusión para simplificar el modelo.

4. TRANSFORMACIÓN DE DATOS

4a.- Normalización de Datos

Normalización en el contexto de Machine Learning es una manera de escalar el rango de ciertos features a una escala estándar. Sirve cuando se tienen rangos de datos variables de los features, dado que, se asegura que todas las variables estén en una escala comparable y así evitar que aquellas que sean más grandes dominen el proceso de entrenamiento [7].

En este sentido, la fórmula de la normalización es la siguiente:

$$X_{\text{norm}} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Figura 3: Fórmula para normalizar rangos de datos

Las columnas que se normalizaron fueron 'Rented Bike Count', 'Hour', 'Temperature(C)', 'Visibility (10m)', 'Dew point temperature(C)', 'Solar Radiation (MJ/m2)', y 'Rainfall(mm)'. Estas columnas fueron seleccionadas para normalización debido a que, cuando se trabajan algoritmos de regresión lineal en Machine Learning, al normalizar las variables se mejora la estabilidad numérica del modelo al contar con una escala estándar, más cuando se trabaja con diferentes unidades de medidas, como es el caso del dataset.

4b.- Codificación de Variables Categóricas

Se utilizó la técnica de “One-Hot Encoding”, que es cuando se presentan datos categóricos (como en el caso de ‘Season’) y mapear esos datos a enteros [8].

En este contexto, se transformó el feature ‘Seasons’ en variables dummy, con el fin de que se hicieran 4 columnas ‘Spring’, ‘Summer’, ‘Autumn’, ‘Winter’ con “False’s” si el valor en X instancia no es igual a dicha columna o con “True’s” cuando coincide el nombre de la columna con el contenido de X instancia en la columna de ‘Season’ y, posteriormente convirtiendo los “False” en 0 y los “True” en 1 para darles datos numéricos a estas columnas.

4c.- División del Dataset en Conjuntos de Entrenamiento y Prueba

Una vez limpiados y transformados los datos, se dividió el dataset en conjuntos de entrenamiento y prueba. La importancia de esta división es evaluar la capacidad del modelo para generalizar en datos no vistos. El dataset fue aleatorizado para evitar sesgos que pudieran existir por el orden original de las instancias.

Con eso en mente, el 80% de los datos se asignó a entrenamiento y el otro 20% se asignó a pruebas, siguiendo el principio de Pareto (regla 80-20) que dice que el 80% del efecto es impulsado por el 20% de las causas (y viceversa) [9].

5. CONSTRUCCIÓN Y EVALUACIÓN DEL MODELO

5a.- Selección de Features y Variables Objetivo

Los features utilizados para entrenar al modelo son: 'Hour', 'Temperature(C)', 'Dew point temperature(C)','Spring', 'Summer', 'Autumn', 'Winter','Solar Radiation (MJ/m2)', 'Rainfall(mm)', esto debido al análisis previo. Mientras que, el valor que queremos predecir (o nuestra variable objetivo) es ‘Rented Bike Count’, ya que deseamos saber la cantidad de bicicletas compartidas de acuerdo a los valores de los features.

5b.- Sesgo en Features

Un algoritmo linear suele tener un sesgo alto, lo que hace que aprenda rápido. En un análisis de regresión lineal, sesgo es el error introducido al acercarse a un problema de la vida real. Un modelo con un alto nivel de sesgo suele hacer más suposiciones sobre la relación entre las variables y el resultado final. Por otro lado, un modelo con un bajo nivel de sesgo hace menos suposiciones sobre cómo se comporta la función que relaciona esas variables con el resultado [10]. Este sesgo se implementó añadiendo una columna de unos a las matrices de características de entrenamiento y prueba.

5c.- Inicialización de Parámetros

Los parámetros del modelo se inicializan de manera aleatoria, con el fin de que el modelo no caiga en patrones repetitivos o en simetrías que puedan causar un mal ajuste [11].

5d.- Cálculo de la Hipótesis

En el contexto de la regresión lineal, la hipótesis se refiere a la función que utiliza

el modelo para predecir el valor esperado de la variable dependiente (en este caso, el conteo de bicicletas rentadas) en función de las características de entrada. La fórmula básica de la hipótesis en una regresión lineal simple es la siguiente [12]:

$$E(Y|x) = \beta_0 + \beta_1 x$$

Figura 4: Función de hipótesis para regresión lineal

En mi implementación, esta idea se extiende a múltiples variables explicativas (regresión lineal múltiple), y la función de hipótesis se calcula como una suma ponderada de todas las características de entrada multiplicadas por sus respectivos parámetros (o coeficientes). En el código, esto se realiza en la función h(params, sample), donde el producto punto entre los parámetros del modelo (params) y las características de entrada de una muestra específica (sample) da como resultado una predicción de la cantidad de bicicletas rentadas para ese conjunto de características.

5e.- Cálculo del error

El error de modelo (o también conocido como función de pérdida) mide la discrepancia entre las predicciones del modelo y los valores reales, es decir, mide la precisión de nuestro modelo [13].

En mi caso, utilicé el Mean Square Error (MSE), que es la diferencia cuadrática entre el valor predicho y el real [14]. Este error se calcula tomando la diferencia entre las predicciones del modelo y los valores reales (y), elevando al cuadrado estas diferencias, sumándolas y, finalmente, dividiéndolas por el doble del número de muestras (m) para obtener un promedio. En el código, esta operación se implementa en la función mse(params, samples, y), donde "samples" son las características de entrada, "params" son los parámetros actuales del modelo y "y" son los valores reales de la variable objetivo (Rented Bike Count). El objetivo del entrenamiento es minimizar este error, lo que se logra ajustando los parámetros del modelo a lo largo de las iteraciones.

$$mse = \frac{1}{n}\sum(y_n - \bar{y})^2$$

Figura 5: Fórmula del MSE

5f.- Descenso de Gradiente

El descenso del gradiente es un algoritmo de optimización usada para entrenar modelos de aprendizaje automático y redes neuronales [15]. Se usa para minimizar la función de pérdida (que, en el caso de mi implementación, es el MSE). Es una función iterativa de ajustar los parámetros del modelo para minimizar la diferencia entre la predicción y los valores reales.

En el contexto del dataset, En cada iteración, calcula la pendiente (o gradiente) de la función de pérdida con respecto a los parámetros y ajusta los parámetros en la dirección opuesta a esta pendiente. El tamaño de este ajuste está controlado por la tasa de aprendizaje "alfa", que determina qué tan grandes son los pasos que el algoritmo da hacia el mínimo de la función de pérdida.

$$\Theta_j := \Theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}[(h_\Theta(x_i) - y)x_i]$$

Figura 6: Actualización de parámetros para el descenso del gradiente para la regresión lineal.

Primero, se calcula la predicción del modelo (h(Θ(x))) multiplicando las características de las muestras (samples) por los parámetros actuales del modelo (params). Luego, se obtiene el error restando las predicciones obtenidas del valor real (y). Este error representa la diferencia entre lo que el modelo predice y los datos reales.

A continuación, se calcula el gradiente, que mide la dirección y magnitud en la que deben ajustarse los parámetros para reducir el error. En mi código, esto se logra multiplicando la transpuesta de la matriz de características por el vector de errores y dividiendo por el número de muestras. Finalmente, los parámetros se actualizan restando este gradiente multiplicado por la tasa de aprendizaje (alfa). Este proceso se repite en cada iteración (o época), lo que permite que el modelo se ajuste de manera progresiva y aprenda a hacer predicciones más precisas.

5g.- Evaluación del Modelo

Para evaluar el rendimiento del modelo, se utilizó la métrica de R² (coeficiente de determinación), que es una medida de correlación de los features con el objetivo. Va desde -∞ a ∞, Cuanto más se acerque a 1, más varianza del valor objetivo pueden explicar las variables de las características. En otras palabras, cuanto más probable sea que haya variables importantes que conduzcan a predicciones precisas [14].

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

Figura 7: Maneras de calcular la R²

6. VISUALIZACIÓN DE RESULTADOS

Después de entrenar el modelo, estos fueron los resultados:

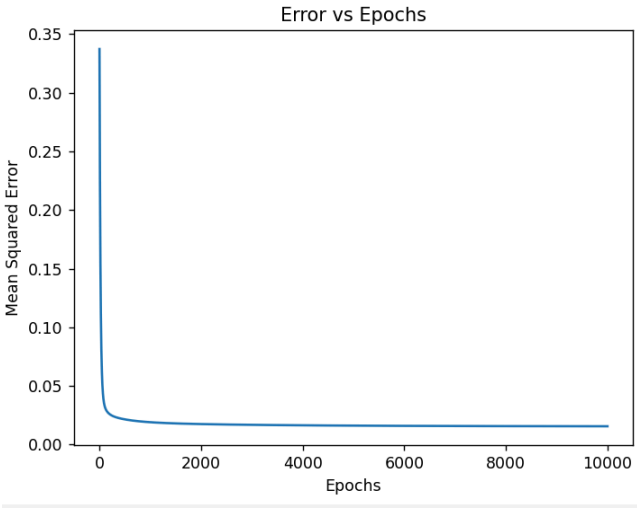


Figura 8: Cálculo de errores por época

La gráfica muestra cómo el MSE disminuye a medida que aumenta el número de iteraciones (epochs) durante el entrenamiento del modelo. La curva de error disminuye rápidamente al inicio, lo que indica que el modelo está aprendiendo y ajustando los parámetros de manera efectiva. Sin embargo, después de aproximadamente 2000 iteraciones, el MSE se estabiliza, sugiriendo que el modelo ha alcanzado un punto donde ya no hay mejoras significativas en la precisión de las predicciones.

Esta estabilización del MSE es indicativa de que el algoritmo ha convergido hacia un mínimo, donde el ajuste de los parámetros ya no reduce de manera significativa el error. Es posible que el modelo haya alcanzado un óptimo local, y aunque se sigan realizando iteraciones, las mejoras en la precisión serán mínimas. Este comportamiento es común en algoritmos

de optimización como el descenso de gradiente, donde el objetivo es minimizar una función de error (en este caso, el MSE) hasta que se alcance un punto de estabilidad.

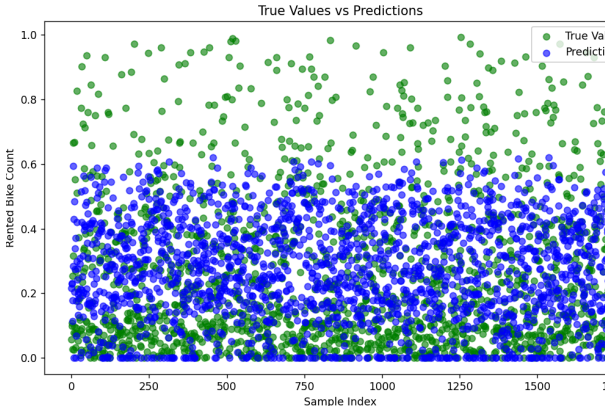


Figura 9: Diferencia entre los valores reales (verde) y los predecidos (azul)

Esta gráfica compara los valores reales del conteo de bicicletas compartidas (en verde) con los valores predecidos por el modelo (azul) para el conjunto de datos de prueba.

Aunque hay cierta superposición entre los puntos verdes y azules, se observa que las predicciones no coinciden exactamente en los valores reales, lo que significa que, aunque el modelo puede captar ciertas tendencias, tiene dificultades para realizar predicciones precisas para todas las muestras. Esto podría deberse a la naturaleza no lineal de algunos de los datos.

```
Epoch #9999, Error: 0.015576596578752624
[-0.39345006  0.25450647  0.9611472  -0.57240789  0.39137056  0.42038815
  0.41744676  0.28384785 -0.05224934 -0.40610685]
R^2 en el conjunto de prueba: 0.5057
```

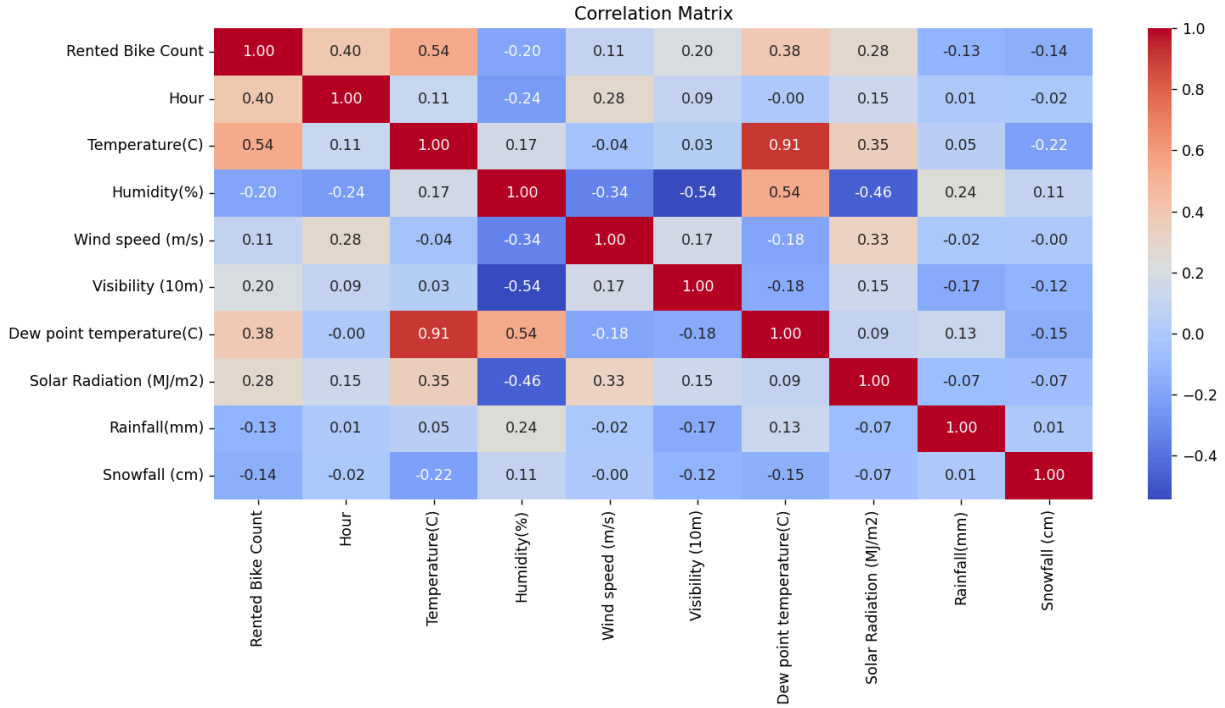
Figura 10: Error final y R^2

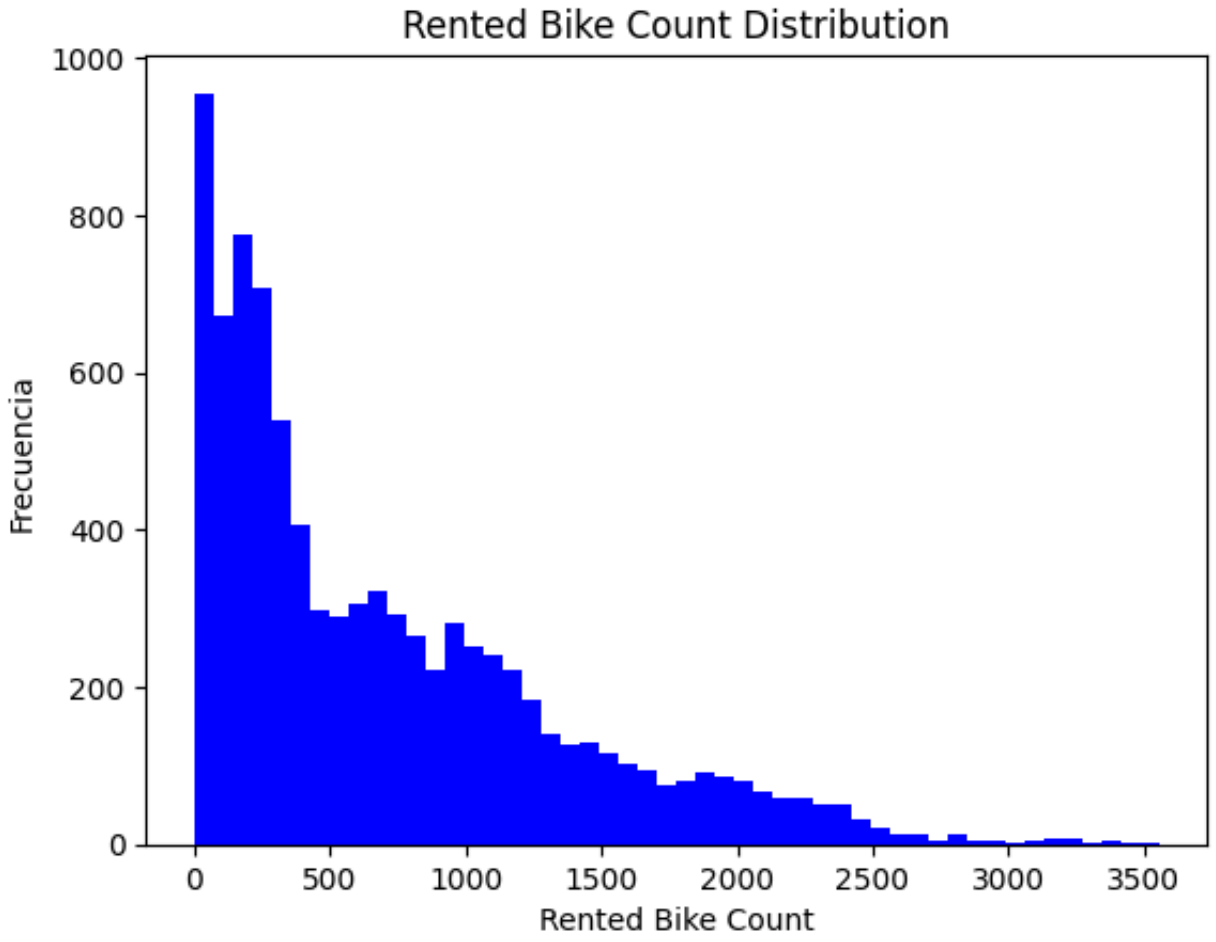
El valor de R^2 es aproximadamente 0.5057, lo que indica que el modelo explica un 50.57% de la variabilidad de los datos de prueba, lo que significa que, aunque el modelo "captura la mitad de los datos", aún hay un 49.43% de variabilidad que no es explicada por el modelo, lo que indica que el modelo está "atinándole" a la mitad de los datos. Esto implica que el modelo puede mejorarse, ya sea incluyendo otros features importantes, usando un modelo más complejo o ajustando los hiperparámetros.

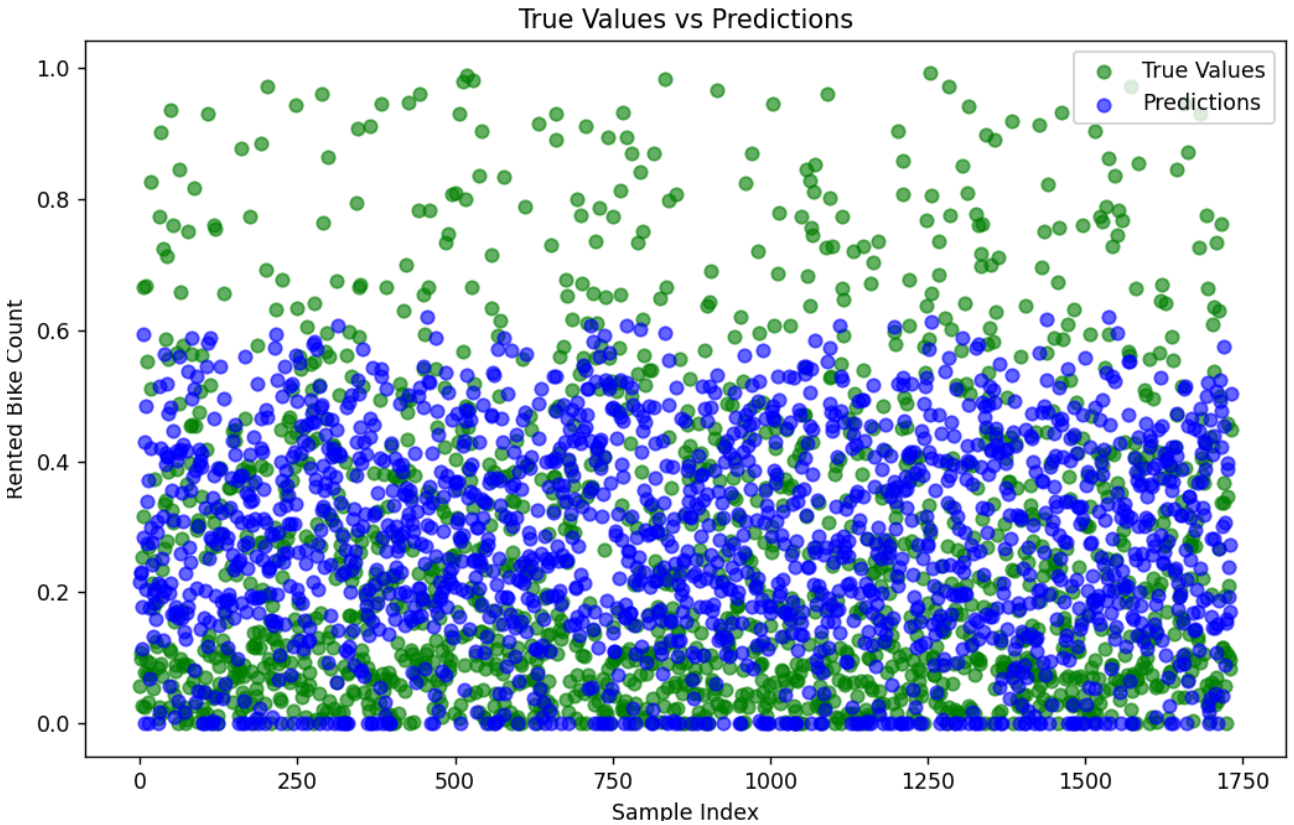
7. REFERENCIAS

1. Shaheen, S., Guzman, S., & Zhang, H. (2020). The Development, Characteristics, and Impact of Bike Sharing Systems: A Literature Review. Retrieved from ResearchGate.
2. PBSC Urban Solutions. (2023). Electrifying Bike Sharing Systems: A Positive Impact on the Environment and Our Society. Retrieved from PBSC.
3. Seoul Bike Sharing Demand Dataset. (n.d.). Retrieved from https://archive.ics.uci.edu/dataset/560/seoul+bike+sharing+demand
4. Docsumo. (n.d.). Data Extraction with Machine Learning. Retrieved from https://www.docsumo.com/blogs/data-extraction/machine-learning
5. Statistics by Jim. (n.d.). How to Remove Outliers in Data. Retrieved from https://statisticsbyjim.com/basics/remove-outliers/
6. DATAtab. (n.d.). Pearson Correlation Tutorial. Retrieved from https://datatab.es/tutorial/pearson-correlation
7. DataCamp. (n.d.). Normalization in Machine Learning. Retrieved from https://www.datacamp.com/tutorial/normalization-in-machine-learning
8. Educative.io. (n.d.). One-Hot Encoding in Machine Learning. Retrieved from https://www.educative.io/blog/one-hot-encoding
9. Ahmed, N. (n.d.). The Motivation for Train-Test Split. Medium. Retrieved from https://medium.com/@nahmed3536/the-motivation-for-train-test-split-2b1837f596c3
10. Masters in Data Science. (n.d.). Difference Between Bias and Variance. Retrieved from https://www.mastersindatascience.org/learning/difference-between-bias-and-variance/#:~:text=In%20linear%20regression%20analysis%2C%20bias,their%20output%20easier%20to%20understand.
11. Shukla, P. (n.d.). Random Initialization in Deep Learning. Medium. Retrieved from https://medium.com/@Coursesteach/deep-learning-part-27-random-initialization-b25ef8df8334#:~:text=Random%20initialization%20is%20crucial%20for,network%20architecture%20and%20activation%20functions.
12. Seltman, H. J. (n.d.). The Model Behind Linear Regression. Carnegie Mellon University. Retrieved from https://www.stat.cmu.edu/~hseltman/309/Book/chapter9.pdf
13. Shukla, P. (n.d.). Error Calculation Techniques for Linear Regression. Medium. Retrieved from https://medium.com/@shuklapratik22/error-calculation-techniques-for-linear-regression-ae436b682f90
14. Qlik Help. (n.d.). Scoring Regression Models. Retrieved from https://help.qlik.com/es-ES/cloud-services/Subsystems/Hub/Content/Sense_Hub/AutoML/scoring-regression.htm
15. IBM. (n.d.). Gradient Descent. Retrieved from https://www.ibm.com/topics/gradient-descent

8. ANEXOS


Correlation Matrix
Rented Bike Count
Hour
Temperature(C)
Humidity(%)
Wind speed (m/s)
Visibility (10m)
Dew point temperature(C)
Solar Radiation (MJ/m2)
Rainfall(mm)
Snowfall (cm)
1.00 0.40 0.54 -0.20 0.11 0.20 0.38 0.28 -0.13 -0.14
0.40 1.00 0.11 -0.24 0.28 0.09 -0.00 0.15 0.01 -0.02
0.54 0.11 1.00 0.17 -0.04 0.03 0.91 0.35 0.05 -0.22
-0.20 -0.24 0.17 1.00 -0.34 -0.54 0.54 -0.46 0.24 0.11
0.11 0.28 -0.04 -0.34 1.00 0.17 -0.18 0.33 -0.02 -0.00
0.20 0.09 0.03 -0.54 0.17 1.00 -0.18 0.15 -0.17 -0.12
0.38 -0.00 0.91 0.54 -0.18 -0.18 1.00 0.09 0.13 -0.15
0.28 0.15 0.35 -0.46 0.33 0.15 0.09 1.00 -0.07 -0.07
-0.13 0.01 0.05 0.24 -0.02 -0.17 0.13 -0.07 1.00 0.01
-0.14 -0.02 -0.22 0.11 -0.00 -0.12 -0.15 -0.07 0.01 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4


Rented Bike Count Distribution
Frecuencia
0
200
400
600
800
1000
0
500
1000
1500
2000
2500
3000
3500
Rented Bike Count


True Values vs Predictions
True Values
Predictions
Rented Bike Count
Sample Index
0.0
0.2
0.4
0.6
0.8
1.0
0
250
500
750
1000
1250
1500
1750