Análisis y Reporte sobre el desempeño del modelo

INTELIGENCIA ARTIFICIAL AVANZADA PARA LA CIENCIA DE DATOS

Osvaldo Del Valle Mejía

A01275702

7 - Septiembre - 2024

Índice

Introducción

Este documento presenta los detalles del reto "Módulo 2 Análisis y Reporte sobre el desempeño del modelo", con un enfoque en la modelación de regresión lineal para predecir el alquiler de bicicletas públicas por lo tanto se decidió el uso del dataset del sistema de compartición de bicicletas de Seúl, que contiene información detallada sobre el alquiler de bicicletas por hora, junto con datos meteorológicos e información sobre días festivos. El dataset cuenta con alrededor de 8000 instancias con cada una registrando datos horarios de alquiler y 13 características que incluyen temperatura, humedad, velocidad del viento, visibilidad, punto de rocío, radiación solar, nevadas, lluvias y datos adicionales sobre el alquiler de bicicletas, sin embargo hay que considerar que este dataset es enfocado en la predicción de estos datos para conocer si es un día laboral, pero considerando que un enfoque más interesante seria el predecir la cantidad de bicis que serían necesarias para un dia en especifico en base a los parámetros aportados se decidió este enfoque.

Por lo tanto el objetivo principal es entender cómo estas características influyen en las predicciones de alquiler y cómo se comparan con los recuentos reales de alquiler, este dataset está enfocado a un modelo de regresión a predecir la cantidad de bicicletas en base a los features seleccionados para el entrenamiento lo que se puede utilizar para generar datos y conocer cuántas bicicletas serían necesarias para asegurar una disponibilidad óptima, para que en algún contexto donde se aplicará el modelo se pudiera minimizar los tiempos de espera y garantizando un suministro constante.

Procedimiento

En este proceso se emplea una técnica de regresión lineal implementando de manera manual además de una implementación apoyada de un framework para utilizar una solución ya construida y probada, además se opta por otra opción de modelado para utilizar random forest como modelo principal asistido por un modelo de bagging para mejorar el resultado debido al overfitting, varianza y valor de r2 con el fin de predecir la cantidad de bicicletas alquiladas en el sistema de compartición de bicicletas de Seúl. A continuación, se detalla la teoría usada y los métodos implementados en el código.

ETL

El proceso de ETL significa extract, transform, load, este es un proceso muy importante en la gestión de datos para integrar y preparar información que proviene de diversas fuentes, este proceso además se utiliza para preparar los datos para enviar a un lugar donde se pueden emplear para situaciones variadas donde se pueden analizar y utilizar de manera efectiva, la fase de extracción implica obtener datos de diversas fuentes, como bases de datos, archivos o sistemas en la nube, en la fase de transformación los datos se limpian, agregan y reorganizan para que sean útiles y consistentes eliminando errores y duplicados, finalmente en la fase de carga los datos transformados se almacenan en un sistema centralizado como un data warehouse donde pueden ser accedidos y utilizados por usuarios finales o aplicaciones de inteligencia artificial.

En el análisis de datos realizado se utilizaron varias variables clave extraídas del conjunto de datos original para construir el modelo predictivo, las variables incluyeron 'Date' que se transformó para capturar únicamente el mes en lugar de la fecha completa, eliminando la columna original para centrarse en posibles relaciones entre los meses y el uso de bicicletas, por otro lado las variables seleccionadas fueron: 'rentedBikeCount' como la variable objetivo a predecir, 'hour' para capturar el patrón de uso a lo largo del día, 'temperature', 'humidity', 'windSpeed', 'visibility', 'dewPoint', 'solarRadiation', 'rainfall', y 'snowfall', que representan factores ambientales clave que influyen en la demanda de bicicletas, además la columna 'season' fue transformada utilizando el método One-Hot Encoding para convertirla en variables dummies que faciliten su uso en modelos de machine learning, además, la columna 'holiday' se transformó de un formato categórico a valores numéricos para que el modelo pudiera interpretarla correctamente y finalmente la variable 'functioningDay', que inicialmente se usaba para indicar si era un día hábil o no, fue eliminada ya que todos sus valores eran

iguales, por lo que no aportaba información relevante al modelo, con este procesamiento se tiene un manejo del dataset orientad a lo que se necesita para manejarlo.

Regresión Lineal

La regresión lineal es una técnica estadística fundamental en el aprendizaje supervisado, utilizada para modelar la relación entre una variable dependiente y una o más variables independientes. En este caso, el objetivo es predecir el número de bicicletas alquiladas (variable dependiente) en función de varias características meteorológicas y temporales (variables independientes).

Función Hipótesis

La función hipótesis es fundamental en el modelo de regresión lineal, ya que determina cómo se generan las predicciones basadas en los parámetros del modelo y las características de entrada. En esencia, la función hipótesis toma los parámetros actuales del modelo, que se ajustan durante el proceso de entrenamiento, y los aplica a los datos de entrada para obtener una estimación de la variable que se desea predecir. La precisión de estas predicciones depende de la calidad de los parámetros y de cómo estos capturan la relación entre las características de entrada y la variable objetivo. En este análisis, la función hipótesis calcula la cantidad esperada de bicicletas alquiladas para cada conjunto de características de entrada, proporcionando así una base para evaluar el desempeño del modelo.

Función de error cuadrático medio (MSE)

El error cuadrático medio (MSE) es una métrica crucial para evaluar la precisión de un modelo de regresión. Esta medida cuantifica la diferencia entre las predicciones realizadas por el modelo y los valores reales observados en los datos. En términos prácticos, el MSE ofrece una indicación de cuánto se desvían las predicciones del modelo en promedio respecto a los valores reales. Un MSE bajo sugiere que el modelo está haciendo predicciones más precisas, mientras que un MSE alto indica que las

predicciones están más alejadas de los valores reales. Esta métrica es esencial para ajustar y mejorar el modelo, ya que proporciona una señal clara sobre la efectividad del modelo en representar la realidad.

Función de descenso por gradiente

El descenso por gradiente es un método de optimización ampliamente utilizado en el aprendizaje automático para ajustar los parámetros del modelo con el fin de minimizar el error. Este algoritmo trabaja iterativamente, ajustando los parámetros en pequeños pasos con el objetivo de reducir el error del modelo. Durante cada iteración, el descenso por gradiente evalúa cómo cambiar los parámetros para disminuir el error total del modelo, moviéndose en la dirección que más reduce el error. Es decir, ajusta los parámetros en función de cómo estos afectan el rendimiento del modelo. El proceso continúa hasta que se encuentra un conjunto de parámetros que minimizan el error de manera satisfactoria. Este enfoque permite al modelo aprender y mejorar continuamente a medida que se ajustan los parámetros, logrando así una mejor precisión en las predicciones.

Evaluación del Modelo

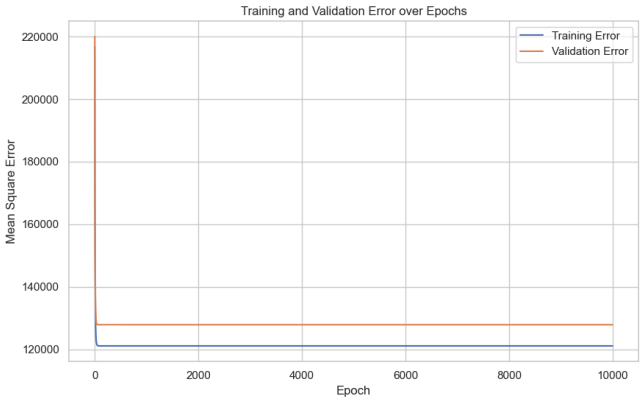
La evaluación del modelo se realiza utilizando dos métricas principales:

- **Error cuadrático medio (MSE)**: Mide el rendimiento del modelo tanto en el conjunto de entrenamiento como en el de validación.
- **R-Cuadrado (R^2)**: Indica la proporción de la varianza en la variable dependiente que es explicada por las características del modelo

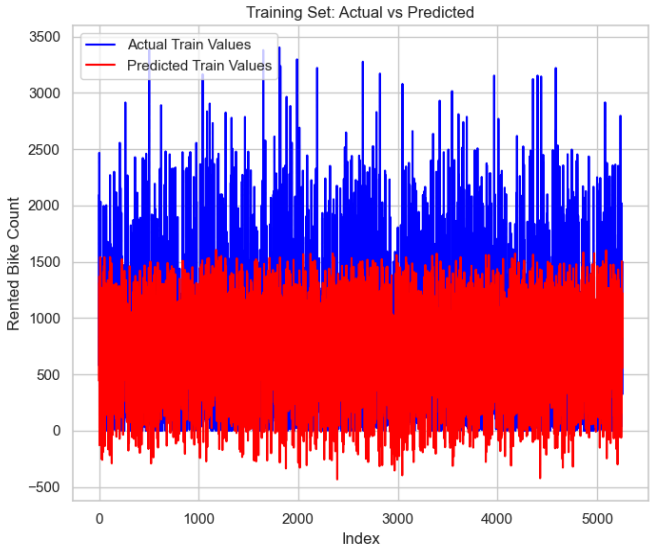
Interpretación de resultados

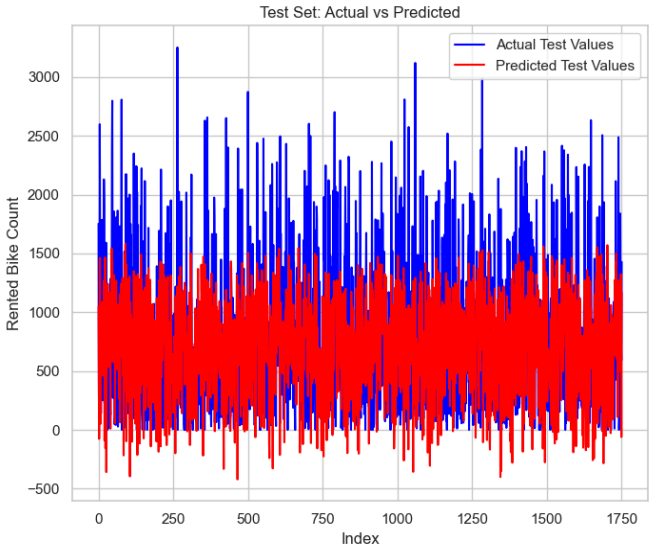
Los resultados obtenidos a partir del código ejecutado del modelo sin framework muestran la evolución del error de entrenamiento y validación durante 10,000 épocas, así como una comparación de los valores reales versus los valores predichos para los conjuntos de entrenamiento y prueba, por otro lado el error de entrenamiento y validación muestra la gráfica def ambos errores, tanto el de entrenamiento como el de validación e indica cómo disminuyen rápidamente en las primeras épocas y luego se estabilizan, sin embargo, el error de validación es mayor que el de entrenamiento, lo que implica que el modelo podría estar experimentando problemas de sobreajuste, este fenómeno indica que aunque el modelo aprende a predecir bien los datos de entrenamiento, su capacidad para generalizar a nuevos datos es limitada. Por lo que en las gráficas que comparan los valores reales y predichos para los conjuntos de entrenamiento y prueba, se observa que las predicciones se encuentran bastante alejadas de los valores reales esto se refleja en la baja capacidad predictiva del modelo, ya que las predicciones parecen estar subestimando el número de bicicletas rentadas en la mayoría de los casos e incluso arrojando valores negativos. Este patrón es similar tanto en el conjunto de entrenamiento como en el de prueba, lo que sugiere que el modelo no logra capturar la complejidad de los datos en su totalidad.

R cuadrado: Los valores de R cuadrado calculados para los conjuntos de entrenamiento (0.4187) y validación (0.3989) son relativamente bajos, lo que indica que el modelo solo explica una fracción pequeña de la variabilidad en los datos entonces estos valores refuerzan la observación de que el modelo tiene un desempeño limitado tanto en el entrenamiento como en la validación.




```
0:      Training Error = 216845.7461573594      Validation Error = 220067.26600377675
500:    Training Error = 121070.45928986992     Validation Error = 127858.19172847256
1000:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
1500:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
2000:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
2500:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
3000:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
3500:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
4000:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
4500:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
5000:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
5500:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
6000:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
6500:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
7000:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
7500:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
8000:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
8500:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
9000:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
9500:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
9999:   Training Error = 121070.45928986996     Validation Error = 127858.19172811788
Final Parameters:[ 704.52663623  784.71269729 1016.4542065  -33.8598344 ]
You can find graphs with result in results folder.
R squared for Training Set: 0.41879626117621094
R squared for Validation Set: 0.3989733102352303
```





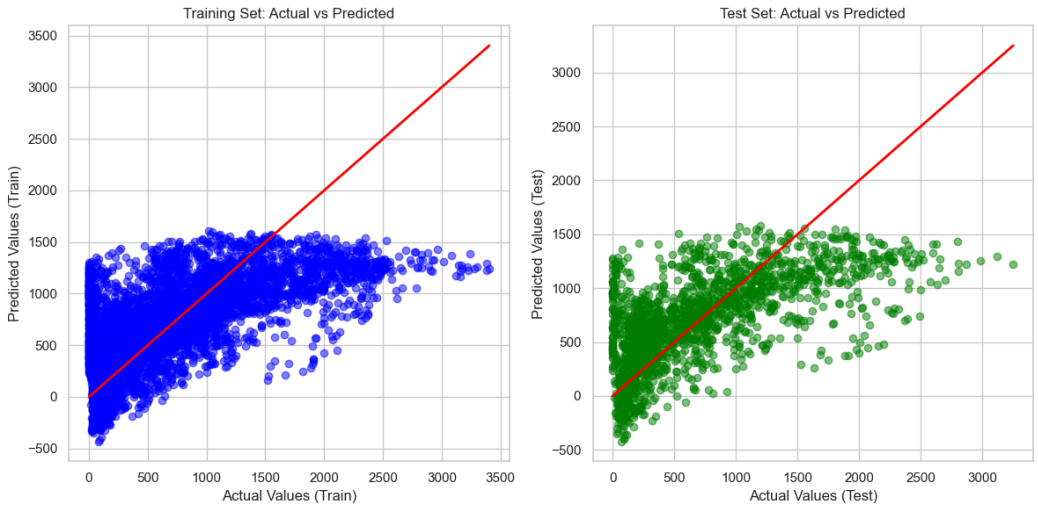
Comparación de resultados

En mi implementación inicial del modelo sin utilizar ningún framework, construí un modelo de regresión básico manualmente. Utilicé técnicas estadísticas sencillas y creé un flujo de procesamiento de datos personalizado, seguido de un entrenamiento de un modelo de regresión lineal simple. Aunque este enfoque sirvió como base para entender el conjunto de datos y las relaciones entre las variables, los resultados fueron limitados. El modelo tenía dificultades para capturar patrones no lineales, y las métricas de rendimiento como el R^2 y el Error Cuadrático Medio (MSE) indicaban que el modelo no estaba generalizando bien en datos no vistos. Además, noté claros signos de overfitting, donde el modelo funcionaba bien con los datos de entrenamiento, pero mucho peor con los datos de validación y prueba. Esto mostraba que el modelo estaba demasiado ajustado a los datos de entrenamiento y no tenía la flexibilidad necesaria para generalizar.

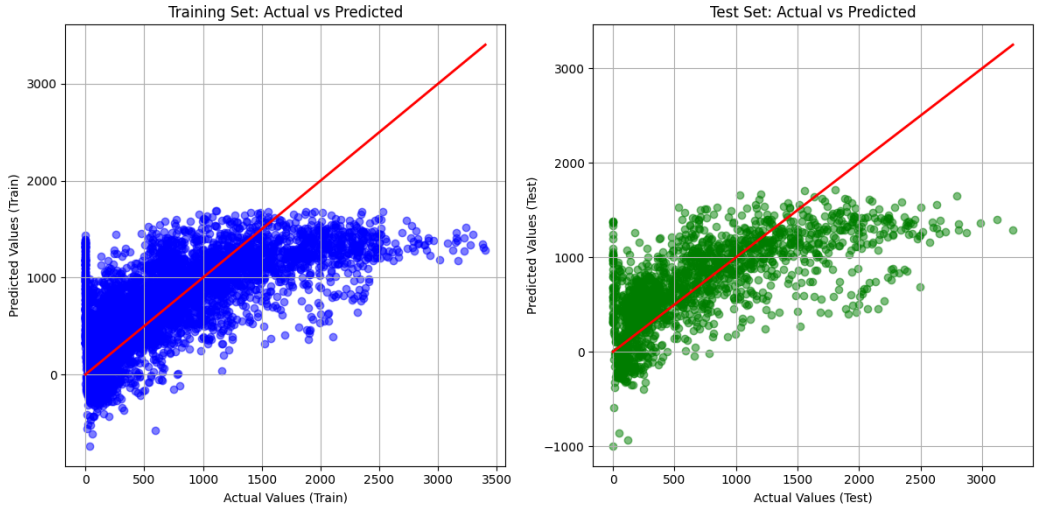
Después, refactoricé el modelo utilizando el framework scikit-learn, lo que mejoró el flujo de trabajo al automatizar muchos procesos como el escalado, la validación cruzada y la evaluación. Aunque el modelo de regresión lineal construido dentro del framework mostró ligeras mejoras en las métricas de rendimiento y facilitó la afinación de hiperparámetros a través de GridSearchCV, el problema del overfitting persistía. El modelo seguía teniendo dificultades con la complejidad de los datos, y a pesar de las optimizaciones, la mejora fue modesta. El framework hizo el proceso más eficiente, pero los problemas fundamentales del modelo permanecían.

Reconociendo las limitaciones del modelo lineal, especialmente el overfitting, cambié el enfoque hacia una técnica de aprendizaje automático más avanzada: Random Forest. Este método de ensamble mejoró drásticamente los resultados. Random Forest fue capaz de capturar patrones complejos y no lineales que los modelos lineales no habían detectado, y su regularización inherente ayudó a mitigar el overfitting. Las métricas de rendimiento mostraron una mejora significativa, con un mayor R^2 y una notable reducción en el MSE. La búsqueda de hiperparámetros a través de GridSearch también optimizó el modelo, resultando en una solución robusta y bien generalizada. Además, la incorporación de técnicas como Bagging añadió aún más estabilidad, haciendo que este enfoque fuera mucho más superior en comparación con las implementaciones anteriores.

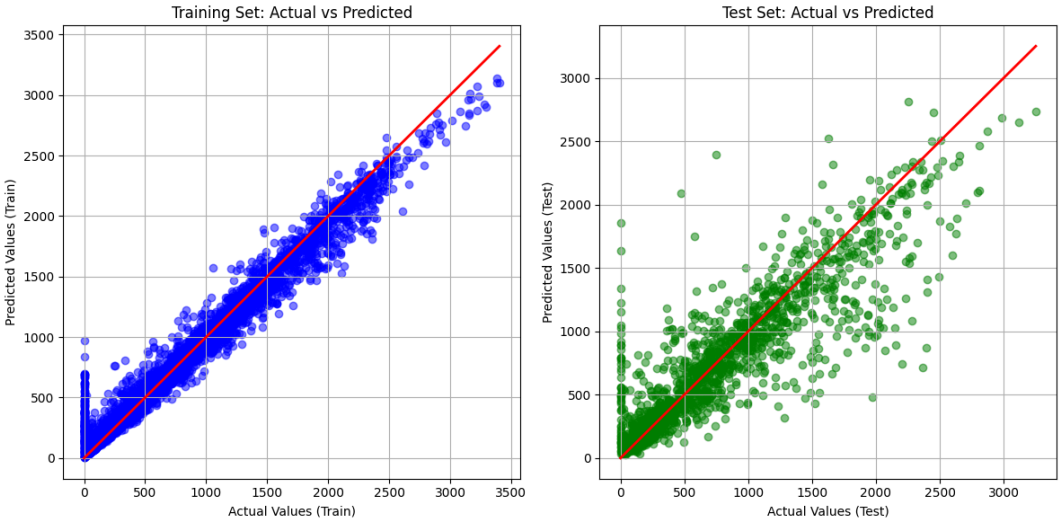
Comparación de resultados predichos contra esperados con implementación a mano



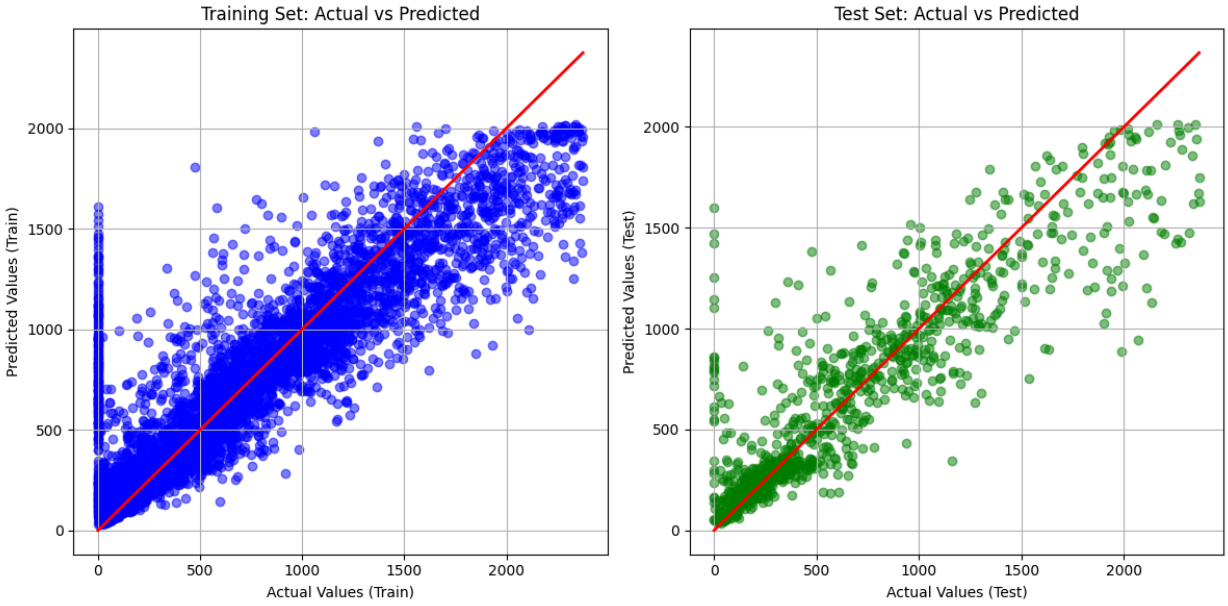
Comparación de resultados predichos contra esperados con uso de framework



Comparación de resultados predichos contra esperados con uso de random forest con overfitting y una alta variabilidad



Comparación de resultados predichos contra esperados con uso de random forest con overfitting solucionado parcialmente con bagging y probar con diferentes hiperparametros



Evidencias

SMA0104 - Análisis de información

Evalúa el modelo con un conjunto de prueba y un conjunto de validación
https://github.com/OsvalDev/machineLearningNF/blob/66f0c125143fd3dbdc798cc30282228fa37cbd12/randomForest.py

Detecta correctamente el grado de bias o sesgo: bajo medio alto
https://docs.google.com/document/d/16WMXm99fgcF9yhsr4OI0KEYC8OkUgrsiqYoV4Xm8VLY/edit?pli=1#heading=h.k26g91wbcj1

Detecta correctamente el grado de varianza: bajo medio alto
https://docs.google.com/document/d/16WMXm99fgcF9yhsr4OI0KEYC8OkUgrsiqYoV4Xm8VLY/edit?pli=1#heading=h.q0y6bs1k5s13

Explica el nivel de ajuste del modelo: underfitt fitt overfitt
https://docs.google.com/document/d/16WMXm99fgcF9yhsr4OI0KEYC8OkUgrsiqYoV4Xm8VLY/edit?pli=1#heading=h.q0y6bs1k5s13

Utiliza técnicas de regularización para mejorar el desempeño del modelo
https://docs.google.com/document/d/16WMXm99fgcF9yhsr4OI0KEYC8OkUgrsiqYoV4Xm8VLY/edit?pli=1#heading=h.78hs6xdf15q9

SMA0401 - Aprendizaje e IA

Implementa una técnica o algoritmo de aprendizaje máquina, sin uso de marco de trabajo o framework como regresiones, árboles, clusters, etc...

https://github.com/OsvalDev/machineLearningNF/blob/66f0c125143fd3dbdc798cc30282228fa37cbd12/linearRegression.py

Usa un marco de trabajo o framework para implementa una técnica o algoritmo de aprendizaje máquina como: regresiones, árboles, clusters, etc...

https://github.com/OsvalDev/machineLearningNF/blob/66f0c125143fd3dbdc798cc30282228fa37cbd12/randomForest.py