Cambridge Default Journal (2025), **31–08**, 1–10

Tecnológico de Monterrey

PROYECT REPORT

Regresión Logística desde cero para la aprobación de préstamo bancario

Victor Adid Salgado Santana A01710023

Tecnologico de Monterrey, Campus Querétaro, México

Abstract

This project develops a machine learning (logistic regression) model implemented from scratch, without the use of external frameworks, with the purpose of classifying credit applications as approved or rejected. The model is trained with relevant data from applicants, such as income, credit history, and loan amount. The goal is to show the process of building a classifier, from data preparation, logistic regression programming: cost function (Binary-Cross-Entropy), gradient descent method, to the evaluation of its performance with metrics such as accuracy, F-1 Score, among others. In addition to its practical application in the banking industry, this project offers a better understanding of how the logistic regression works

Este proyecto desarrolla un modelo de machine learning (regresión logística) implementado desde cero, sin el uso de frameworks externos, con el propósito de clasificar solicitudes de préstamo en aprobadas o rechazadas. El modelo se entrena con datos relevante de los solicitantes, como ingresos, historial crediticio y cantidad de préstamo. Con ello se busca documentar el proceso de construcción de un clasificador, desde la preparación de los datos, la programación de la regresión logística: la función de costo (Binary-Cross-Entropy), el método de descenso de gradiente, hasta la evaluación de su desempeño con métricas como accuracy, F-1 Score, entre otras. Además de su aplicación práctica en el ámbito bancario, este proyecto ofrece el funcionamiento interno de la regresión logística para una mejor comprensión.

Keywords: Machine Learning, Regresión Logística, Aprobación de préstamo, Clasificación

1. Introducción

Las instituciones financieras enfrentan el desafío de evaluar el riesgo crediticio de sus clientes. Prestar dinero implica un nivel de incertidumbre, y es fundamental que los bancos tengan cierta certeza de que los solicitantes podrán cumplir con sus obligaciones de pago. Entre los criterios más comunes para determinar la aprobación de un préstamo se encuentran: una fuente estable y suficiente de ingresos para poder solventar las cuotas mensuales **BBVA**.

El presente proyecto tiene como objetivo diseñar e implementar un modelo de machine learning para la aprobación de préstamos, específicamente mediante regresión logística. Con el objetivo de automatizar parte del proceso de evaluación, de manera que las instituciones financieras puedan ahorrar tiempo y recursos en el análisis de solicitudes. No obstante, se reconoce que la decisión final siempre debe recaer en un humano, ya que los modelos no pueden ser responsables de la decisón final.

Gracias a la disponibilidad de datos históricos que incluyen información de los solicitantes y el resultado de sus solicitudes, es posible entrenar un modelo capaz de distinguir entre perfiles con alta probabilidad de aprobación y aquellos que probablemente sean rechazados.

2. Marco Teórico

En la clasificación de solicitudes de préstamo, es fundamental contar con un modelo que no solo prediga si un préstamo será aprobado o rechazado, sino que también entregue probabilidades asociadas. Para este propósito, se emplea la regresión logística, un modelo supervisado ampliamente utilizado en problemas de clasificación binaria (2 clases). En este contexto, la regresión logística estima la probabilidad de que un solicitante tenga un préstamo aprobado o rehazado dado su perfil.

2.1 Regresión Logística Machine Learning

2.1.1 Binary Cross-Entropy

Para medir el desempeño de la regresión logistica es necesario una función de costo o "pérdida". Para este modelo se usa la Binary Cross-Entropy o también conocida como Log Loss:

$$Logloss(y, p) = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(p_i) + (1 - y_i) log(1 - p_i)) \tag{1}$$

Esta función de pérdida calcula que tan bueno o que tan malo es nuestro modelo. Este problema de clasificación binaria (solo 2 clases) préstamo aprobado o no aprobado tiene 2 posibles valores para la y, la cuál es el valor real: 0 = préstamo No Aprobado,1 = préstamo Aprobado

Cómo este modelo trabaja con probabilidades gracias a la sigmoide, los valores de p están acotados a $p = [0 - 1]$ pues son la probabilidad de que un solicitante tenga un préstamo aprobado o no.

Gracias a los logaritmos, si la probabilidad coincide o se acerca bastante a la real, el error va a tender a 0, pero si la realidad dista completamente de lo predicho, $y = 1, p = 0$. Entonces el error tiende a infinito, pues es como si fuera improbable según nuestro modelo cuando es completamente lo opuesto. Por esa razón en la práctica se le suma un valor muy pequeño $\epsilon \approx 0$ para que en estos tipos de casos (o la situación opuesta) no nos arroje algún error el modelo.

2.1.2 Función Sigmoide

La función sigmoide transforma números continuos reales a valores entre $(0,1)$ **Chen2024**

$$sigmoide(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

Esto es de suma importancia para nuestro modelo de regresión logística, ya que transforma nuestros cálculos en probabilidades.

2.1.3 Función de Hipótesis

La regresión Logística como modelo de machine learning busca optimizar los pesos w y los bias b, los cuáles se pueden apreciar en la función de hipótesis, la cuál es una ecuación lineal metida en la función de sigmoide, lo que termina en el cálculo de una probabilidad.

$$hipotesis(h) = \frac{1}{1 + e^{-h}} \tag{3}$$

Dónde:

$$h = w_1 \cdot x_1 + ... + w_n \cdot x_n + b \tag{4}$$

2.2 Gradiente Descendente

Para la optimización de dichos pesos de la función de hipótesis, se utiliza el método de gradiente descendente. El cuál con el cálculo del gradiente de la función de costo busca llegar a la dirección en la que determinados pesos y bias producen el menor error, es decir minimizar la función de costo $J(w, b)$. Con varias iteraciones y con ayuda de un Learning Rate α (dice que tanto se avanza), gradiente descendente es utilizado para moverse en dirección del mínimo, con el fin w y b sean óptimos locales idealmente.

$$w = w - \alpha \frac{\partial J}{\partial w} \tag{5}$$

$$b = b - \alpha \frac{\partial J}{\partial b} \tag{6}$$

2.3 Estandarización por Z-score:

Para procesar los datos antes de ingresarlos al modelo de regresión logística, la estandarización es un proceso importante. Esta transformación centra los datos con media 0 y desviaciones estándar de 1. La estandarización asegura que todas las variables numéricas estén en la misma escala. Esto es importante porque en algoritmos basados en gradiente, como la regresión logística, variables con escalas muy distintas pueden sesgar el proceso de optimización. Ahora los datos dirán que tan cerca o que tan lejos estás de la media $\mu = 0$ en términos de desviaciones estándar (σ)

$$z = \frac{x - \mu}{\sigma} \tag{7}$$

3. Exploración y limpieza de los datos

El dataset de este proyecto se obtuvo de: **Kaggle**, contiene 45,000 registros y 14 columnas las cuáles se listan a continuación:

Table 1. Variables del dataset y descripción.

Columna	Descripción	Tipo
person age	Edad de la persona	Float
person gender	Género de la persona	Categórica
person education	Nivel de educación	Categórica
person income	Ingreso anual	Float
person emp exp	Años de experiencia laboral	Integer
person home ownership	Estatus dueño de una casa	Categórica
loan amnt	Monto del préstamo solicitado	Float
loan intent	Propósito del préstamo	Categórica
loan int rate	Tasa de interés del préstamo	Float
loan percent income	Préstamo como porcentaje del ingreso	Float
cb person cred hist length	Historial crediticio en años	Float
credit score	Credit score de la persona	Integer
previous loan defaults on file	Préstamos incumplidos anteriormente	Categórica
loan status	Préstamo aprobado (1=aprobado, 0=rechazado)	Integer

3.1 Limpieza de los datos

El dataset no contiene datos nulos, por lo que no se realizo algún método para lidiar con datos faltantes, más adelante se habla acerca del tratamiento de outliers.

3.2 Exploración de los datos

A continuación se muestran gráficas que muestran la distribución de las variables numéricas del dataset.

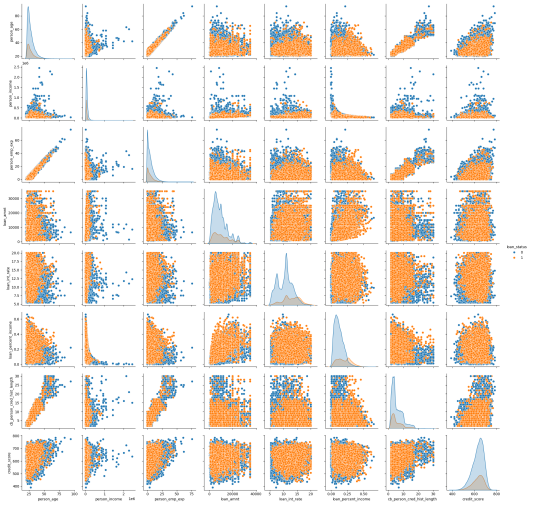

Figure 1. Pair Plot de las variables numéricas por estatus de préstamo

Para las primeras 3 variables, edad, ingreso y experiencia laboral se observa un comportamiento altamente sesgado a la derecha. Para la variable de credit score se observa un comportamiento de sesgo a la izquierda. Para las variables de tasa de interés del préstamo y porcentaje del préstamo en ingreso, se puede observar como la distribución de los solicitantes con préstamos aprobados y rechazados cambia un poco.

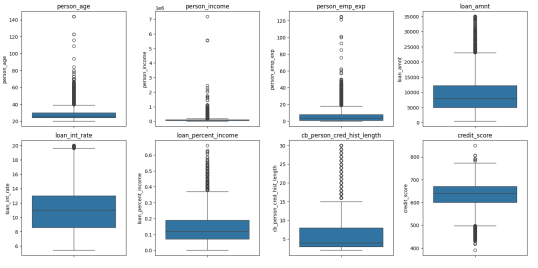

Figure 2. Boxplot de las variables numéricas

Con los diagrama de caja se observa la distribución de las variables, donde se puede apreciar el rango intercuartílico, la mediana y es sencillo detectar la presencia de datos extremos. Como lo es en el caso de la edad, dónde se ven edades inusuales de 120-140 años, valores atípicos muy distantes en los ingresos, personas con experiencia laboral de más de 100 años, para las demás variables se puede observar un comportamiento en el que hay datos con valores muy elevados, por último para credit

score hay outliers con solicitantes con un puntaje muy por debajo del usual y solicitantes con un score muy por encima de lo típico.

4. Preparación de los datos

Para el preprocesamiento y preparación de los datos se opta por seleccionar de variables de tipo numérico, las variables categóricas son desechadas, ya que el modelo de regresión logística trabaja con variables continuas, además de que algunas de las variables categóricas podrían inducir a un sesgo por género o educación, lo cuál estaría atentando contra un principio de IA responsable de equidad e inclusión. **MSFT**
El dataset numérico está compuesto por: 'person age', 'person income', 'person emp exp', 'loan amnt','loan int rate', 'loan percent income', 'cb person cred hist length', 'credit score', 'loan status'.

4.1 Desbalanceo de clases

Se analiza la cantidad de datos que cuentan con un préstamo aprobado y con un préstamo rechazado.

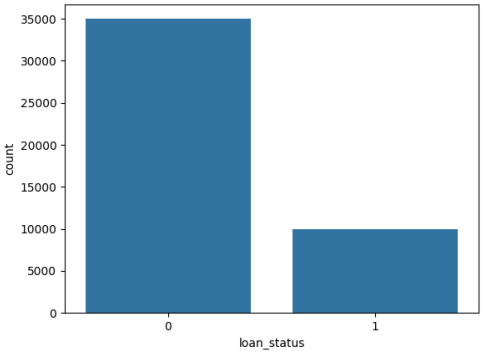

Figure 3. Conteo de registros por clase

Cómo se observa, de los 45,000 registros, 35,000 de ellos pertenecen a personas con préstamos rechazados y el resto de los registros a solicitantes con préstamos aprobados. Esto significa que el 77% de los datos pertenece a la clase 0 y sólo el 22% a la clase 1.
Si se busca tener un modelo balanceado, con un desempeño aceptable en la clasificación de ambas clases, es necesario aplicar alguna técnica de balanceo de datos. Para este caso, se hace una comparación entre los resultados de un modelo con desbalanceo de datos y otro modelo dónde se opta por un **undersampling** de la variable mayoritaria.

4.2 Análisis de correlaciones

Al acotar nuestras variables a variables de tipo numérico, se hace un análisis de correlación con el objetivo de eliminar variables altamente correlacionadas $r > 0.8$.

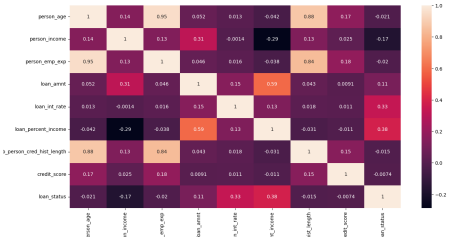

Figure 4. Mapa de correlaciones de las variables numéricas

Con el análisis se puede observar que 'person emp exp' y 'cb person cred hist length' esta muy relacionado con la 'person age' $r > 0.8$. Esto es lógico, debido a que típicamente, las personas con mayor edad son las que tienen más años de experiencia laboral y más años de historial crediticio.

4.3 Datos atípicos y outliers

Cómo criterio para tratar a los datos atípicos se eliminarán valores extremos de acuerdo al comportamiento y naturaleza de cada variable.

- **Personas con más de 100 años**
- **Personas con muchos años de empleo**
- **Personas con ingresos desproporcionables**

Se eliminaron registros con edades mayores a 100 años y experiencias laborales mayores a 80 años, al considerarse poco realistas y posiblemente atribuibles a errores de captura. De igual manera, se eliminaron ingresos extremadamente altos detectados como valores atípicos fuera de rango según boxplots.

4.4 Selección de variables

Con el análisis de correlaciones se opta por eliminar las variables 'person emp exp' y 'cb person 'cred hist length', debido a que están altamente correlacionadas con 'person age', la decisión de tomar solo variables numéricas y el tratamiento de datos atípicos, produce un dataset final de 44993 registros y 6 variables ('person age', 'person income', 'loan amnt', 'loan int rate', 'loan percent income', 'credit score')

4.5 Estandarización

Por último se procede a estandarizar los datos, la media y desviación estándar de cada columna es almacenada en dado caso que se quieran clasificar nuevos solicitantes.

5. Construcción del modelo

Dado a que la implementación de la regresión logística es desde 0 y sin ayuda de frameworks se programan las siguientes funciones:

- **sigmoid(x):** Regresa las probabilidades dada una x que corresponde a la función de hipótesis
- **log_loss(y, y_predicted):** Cálcula el costo de la solución programando la binary cross entrophy
- **logistic_regression(X, y, learning_rate, epochs):** En esta función se hace la implementación de la regresión logística, usando las funciones anteriores y computando el cálculo del gradiente y la implementación del gradiente descendente para la optimización de los pesos, esta función devuelve los pesos w, el bias b y un arreglo con el historial del costo de la solución *cost_history*

- Adicionalemente se programan las funciones para estandarizar los datos, realizar el split de datos de entrenamiento y de testeo, realizar undersampling y funciones para evaluar el desempeño del modelo.

5.1 Split de los datos

Para este proyecto se establece un split de train y test únicamente. Tomando proporciones de 80% datos de entrenamiento y 20% datos de testeo. Se resamplean aleatoriamente los registros antes de hacer el split.

6. Resultados

El modelo principal es el que tratado con undersampling, lo que redujo considerablemente los datos. Pasamos de 44993 registros a 20,000 registros en total.
Al correr el modelo para nuestros datos de entrenamiento con un learning rate α = 0.001 y *epochs* = 10000. Se obtienen los siguientes pesos:

Table 2. Pesos (w) y bias (b) del modelo

Parámetro	Valor
w 1	-0.0222
w 2	-0.4208
w 3	-0.0333
w 4	0.7663
w 5	0.7190
w 6	-0.0304
b	-0.0084

Al analizar los coeficientes se nota como el modelo de regresión logística asigna una mayor importancia a las variables 2,4 y 5 que corresponden a ingresos de la persona, la tasa de interés del préstamo y el porcentaje del préstamo de acuerdo al ingreso de la persona. Esto debido a que $|w2|, |w5|, |w6|$ los valores absolutos de los pesos de esas variables son lo mayores respecto a los demás parámetros. Esta información está alineada con lo investigado anteriormente acerca de los criterios de aprobación o rechazo de los préstamos, pues al igual que las instituciones bancarias, nuestro modelo se apoya en la cantidad de ingreso del solicitante, el porcentaje que le representaría al solicitante adquirir ese préstamo de acuerdo a su ingreso y la tasa de interés del préstamo.

6.1 Evaluación del modelo

A continuación se muestran algunas métricas del desempeño de la regresión logística: Se puede ver una matriz de confusión:

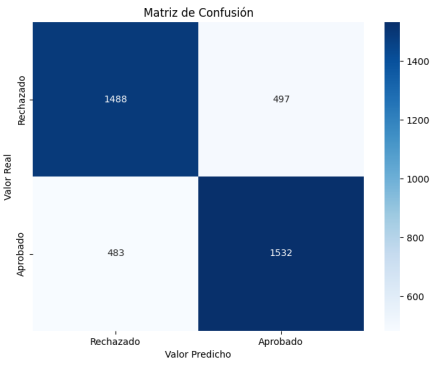

Figure 5. Matriz de confusión de los valores predichos vs los verdaderos

En la matriz de confusión se puede ver como aproximadamente 3/4 de los casos son clasificados correctamente. Se puede apreciar un balance ya que tanto el número de clasificados como aprobado y el número de clasificados como rehcazados son muy similares, recordando que al emplear under-sampling, el dataset quedó con el mismo número de registros tanto para solicitantes aprobados como para rechazados

- Accuracy: 0.7550
- Precision: 0.7550
- Recall: 0.7603
- F1 Score: 0.7577

El modelo de regresión logística tiene un desempeño balanceado, ya que todas las métricas están muy cercanas y con un valor "aceptable". Esto significa que no solo predice bien los préstamos aprobados, sino que también logra capturar la mayoría de los aprobados reales sin demasiados falsos positivos o negativos.

Función de costo (LogLoss):

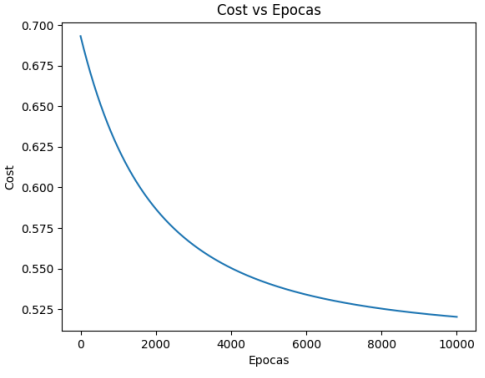

Figure 6. Log Loss a través de las épocas

Se puede apreciar la optimización de la función de costo (Binary Cross-Entropy) con el paso de las épocas, con el learning rate *alpha* = 0.001 no se nota un descenso abrupto (es decir una convergencia a un óptimo local muy rápido, en pocas épocas).

Curva ROC del modelo:

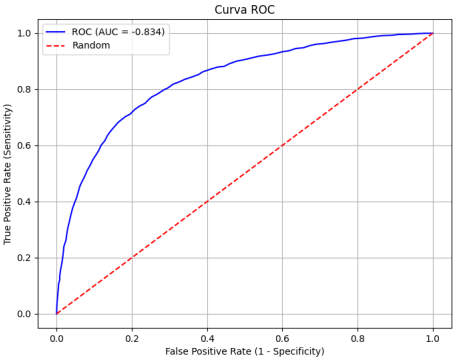

Figure 7. Gráfica de la curva ROC

Con esta gráfica se aprecia que el modelo construído es mejor que un clasificador aleatorio. Se puede ver la tasa de clasificación de verdaderos positivos a costa de falsos positivos. Idealmente la línea debe de estar muy cerca de la esquina superior izquierda, en este caso, el modelo se acerca a esa esquina pero no tiene un desempeño excepcional.
Por lo visto en la matriz de confusión, las métricas calculadas y la curva ROC, se puede decir que el modelo construído tiene un desempeño aceptable.

6.2 Comparación de modelos

Table 3. Comparación de Modelos de Regresión Logística con y sin Undersampling

Modelo	Learning rate	Épocas	Tiempo (s)	Accuracy	Precision	Recall	F1 Score
Sin undersampling	0.01	1,000	279.25	0.7562	0.8455	0.2953	0.4377
Undersampling (rápido)	0.1	100	7.09	0.7490	0.7374	0.7643	0.7506
Undersampling (normal)	0.001	10,000	741.92	0.7550	0.7550	0.7603	0.7577

6.3 Hardware y tiempo de ejecución

Dispositivo empleado para la ejecución del modelo (el de undersampling) y el tiempo de entrenamiento

Table 4. Especificaciones del sistema utilizado en los experimentos

Característica	Especificación
Procesador	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz (2.21 GHz)
RAM instalada	12.0 GB
Tiempo de entrenamiento	741.92 segundos

6.4 Conclusiones

En este proyecto se construyo una regresión logística desde 0 (y sin uso de frameworks) como modelo de machine learning para clasificar las solicitudes de préstamos en aprobado y rechazado. A lo largo de este proyecto se hicieron tareas de extracción de datos, exploración, limpieza, tratamiento de outliers y transformación de ellos. Para la preparación de los datos, se utilizaron solo variables numéricas,

las cuales fueron estandarizadas, se lidió con el desbalanceo de clases por medio de undersampling y se hizo un análisis de correlaciones; gracias a estas técnicas se seleccionaron 6 variables para la clasificación del estatus del préstamo. Los resultados muestran un desempeño del modelo aceptable, dónde se tiene un éxito de predicciones correctas del 75%. Se observa como tiene un equilibrio, pues todas las métricas están alrededor del 75%, lo que indica un balance, lo que se puede confirmar con la matriz de confusión, que muestra números similares para clasificación de préstamos aprobados y rechazados. Se podría decir que de 4 solicitudes, 3 son clasificadas correctamente. Comparando modelos se observa como con un leraning rate mayor y pocas épocas los resultados son similares, reduciendo significativamente el tiempo de ejecución, esto indica que no se necesitan tantas épocas para un buen desempeño en este escenario. Por último, se destacan las acciones que podrían mejorar el rendimiento del modelo en la sección de siguientes pasos.

7. Siguientes pasos

- Para mejoras en el futuro de este proyecto, se recomienda cambiar la estrategia de separación de los datos incluyendo un set de validación.
- Así mismo, profundizar en la detección y tratamiento de outliers en otras variables.
- Es importante recordar que en estos modelos solo se utilizaron variables numéricas, por lo que, hacer encoding de variables categoricas para poder incluirlas en el modelo, es una recomendación que podría mejorar en gran manera el desempeño de la regresión.
- De forma computacional, se pueden vectorizar algunas operaciones en las funciones implementadas para la regresión logística para eliminar los ciclos for y reducir la complejidad computacional. **Con el fin de realizar la comparación entre modelos, los tiempos de ejecución presentados corresponden al modelo con operaciones con matrices en la función de regresión logística, con los ciclos for en lugar de ellos el tiempo de ejecución se dispara, lo que vuelve impráctico al momento de comparar varios modelos con distintas épocas**.
- En este caso, se optó por utilizar la ténica de undersampling para lidiar con el desbalance de los datos, no obstante, esta técnica sacrificó una gran cantidad de datos con el fin de balancear las clases, en próximos pasos, se pueden explorar otras ténicas de balanceo como oversampling o incluír alguna librería con otras ténicas como SMOTE
- Generar una validación cruzada para el modelo
- Implementar un learning rate adaptativo

8. Bibliography

- Chen, Y., Li, L., Li, W., Guo, Q., Du, Z., & Xu, Z. (2024). Fundamentals of neural networks. AI Computing Systems, 17–51. https://doi.org/10.1016/B978-0-32-395399-3.00008-1
- ¿Qué miran los bancos para apobar o denegar un préstamo? (n.d.). Retrieved August 27, 2025, from https://www.bbva.com/es/salud-financiera/como-aprueban-o-deniegan-un-prestamo-criterio-entidades-financieras/
- Regresión logística en el aprendizaje automático - GeeksforGeeks. (n.d.). Retrieved August 27, 2025, from https://www.geeksforgeeks.org/machine-learning/understanding-logistic-regression/
- Loan Approval Classification Dataset. (n.d.). Retrieved August 28, 2025, from https://www.kaggle.com/datasets approval-classification-data
- Definición de inteligencia artificial responsable - Azure Machine Learning | Microsoft Learn. (n.d.). Retrieved August 28, 2025, from https://learn.microsoft.com/es-es/azure/machine-learning/concept-responsible-ai?view=azureml-api-2

Appendix 1. Github Repository

Github ML model from scratch