



Instituto Politécnico Nacional



Centro de Investigación
en Computación
Instituto Politécnico Nacional

Centro de Investigación en Computación CIC
Departamento de Diplomados y Extensión Profesional

CURSO DE MACHINE LEARNING

Maestro: Alan Badillo Salas

Alumno: Montiel Varela Alejandro

**Practica 101 – Introducción al Machine Learning y Herramientas de
Trabajo**

Repositorio de la práctica:

https://github.com/MontielAlejandro/Practica_101_ML_CIC_Github_Python

Ciudad de Mexico febrero 1, 2026.

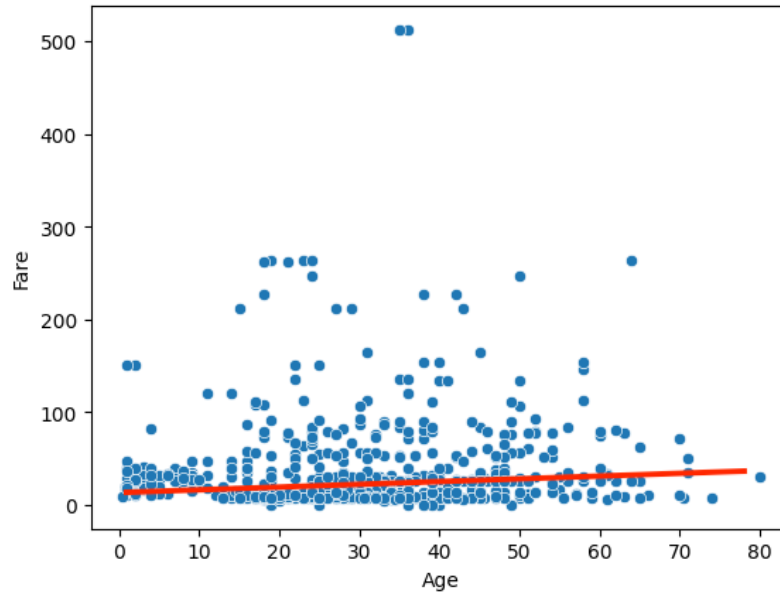
Introducción

En propósito de esta práctica es la creación de un repositorio de github para poder desarrollar un proyecto de machine learning y ver la manera en que se desarrolla un proyecto de machine learning, github y visual code. Para ello se utilizó un data frame que esta disponible en la red llamado titanic. Se realizaron un análisis básico de las variables y se procedió a graficar dos variables de este data frame de edad vs costo del boleto.

Desarrollo

1. Se crea una cuenta de Github y se descarga Github desktop.
2. Una vez creada la cuenta y se instala Github desktop se crea un repositorio publico para cualquier persona pueda acceder. El repositorio se puede crear en línea (dentro de sitio web de Github a través de tu cuenta) o usando la aplicación de Github desktop. En este caso se creo desde la pagina web. El nombre del repositorio es Practica_101_ML_CIC_Github_Python. En este caso se creo el repositorio y dentro del repositorio está el archivo README,
3. Una vez creado el repositorio se clona en el escritorio ya sea mediante la consola (terminal o símbolos del sistema) o usando Github desktop (recomendado).
4. Una vez clonado el repositorio en la computadora local se abre la carpeta del repositorio clonado usando visual code el cual únicamente contiene el archivo README.
5. Posteriormente dentro de Visual Code se crea un archivo llamado. gitignore y se agrega la línea de comando *ml-env/* para que Github ignore el ambiente virtual usado en el proyecto y no lo suba al repositorio.
6. Edita el archivo README.md usando títulos y subtítulos e imágenes locales al repositorio o desde la URL para crear la postada del repositorio.
7. Se abre la terminal integrada en Visual Code con los comandos *ctrl+j* en Windows y *CMD + J* en Mac OS.
8. Dentro d el terminal se crea un entorno virtual en este caso llamado ml-env. Para ello se utilizó el comando *python -m venv ml-env*.
9. A continuación, se activa el entorno virtual. Para ello se utilizo el comando *source ml-env/bin/actívale*. En la terminal se observa que se cambia el path de (base) al del ambiente virtual (ml-env)
10. Una vez activado el ambiente virtual se instalan dentro del paquete virtual las librerías de pandas y seaborn. Para ellos se utiliza el comando *pip install pandas seaborn*.
11. Una vez instaladas las librerías se congelan en el archivo *requeriments.txt*. ejecutando el comando *pip freeze >requirements.txt* dentro del ambiente virtual.
12. Se sigue dentro del ambiente virtual.

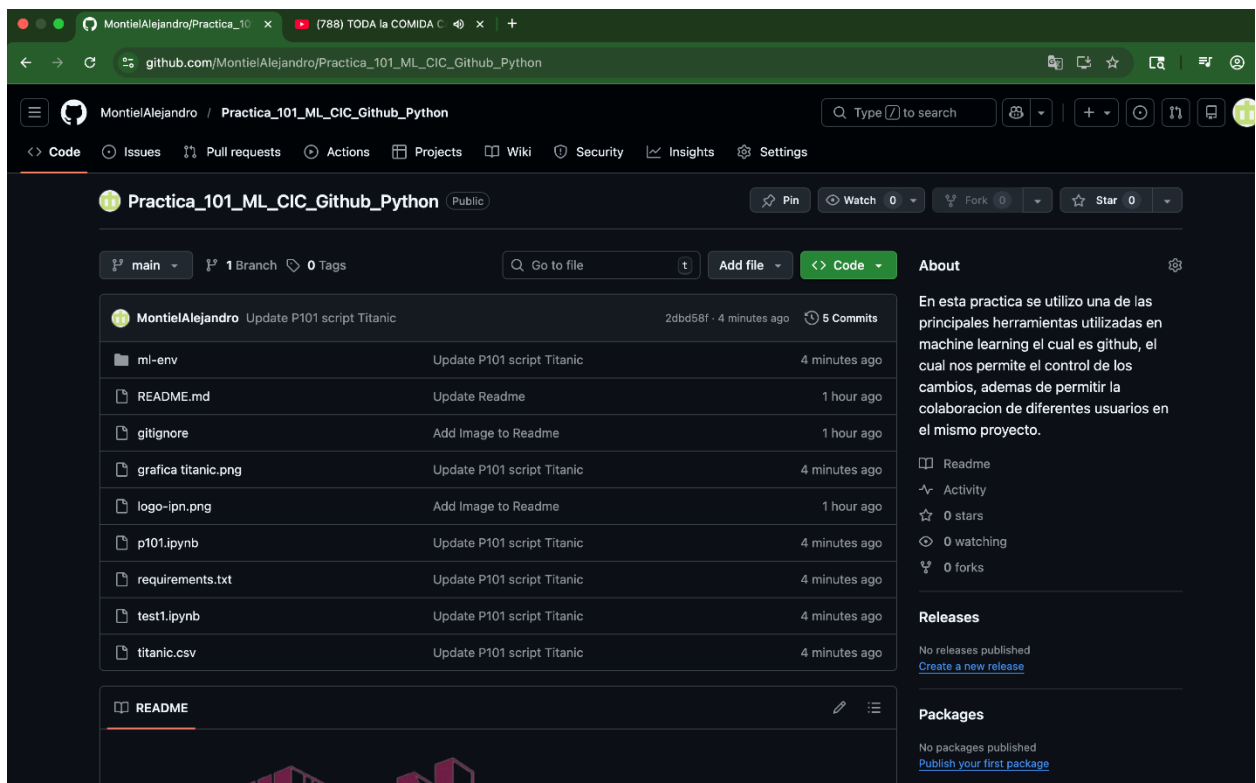
13. Crear un archivo llamado *test.ipynb* para crear una libreta de python.
14. Abre el archivo *test.ipynb*.
15. Se verifica que el kernel utilizado sea el creado anteriormente es decir el ambiente virtual *ml-env*.
16. Escriba en una celda de código 2+2 dentro del archivo *test.ipynb* y se ejecuta el código, la línea de código debe arrojar de resultado 4.
17. Se crea otra libreta llamada *p101.ipynb*.
18. En una línea de código importe la librería *pandas* y *seaborn*
19. Descargar el archivo *csv* con los datos del *titanic*
20. Pon el archivo *titanic.csv* en la carpeta del repositorio a lado de las libretas.
21. En la libreta *p101.ipynb* en una celda de código cargue el data frame como la variable *titanic*.
22. Muestre la información del *titanic*. Se muestra en Github
23. Muestre la información estadística del *titanic* e interprete su significado. Se muestra en Github
24. Muestre las primeras 10 filas del *titanic*. Se muestra en Github
25. Muestre las ultimas 10 filas del *titanic*. Se muestra en Github
26. Crea la variable *x1* con la columna *age* (edad). Se muestra en Github
27. Crea la variable *y* con la columna *fare* (tarifa). Se muestra en Github
28. Crea la variable *g* con la columna *sex* (sexo). Se muestra en Github
29. Grafique los puntos entre *x1* y *y* usando *seaborn*. Se muestra en Github
30. Grafique los puntos entre *x1* y *y* segmentados por *g* usando *seaborn*. Se muestra en Github
31. Después se guardo la captura de pantalla y se realizó una curva que mejor se ajuste a los datos siendo lo suficientemente flexible y rígida a la vez.



32. Guardar la libreta

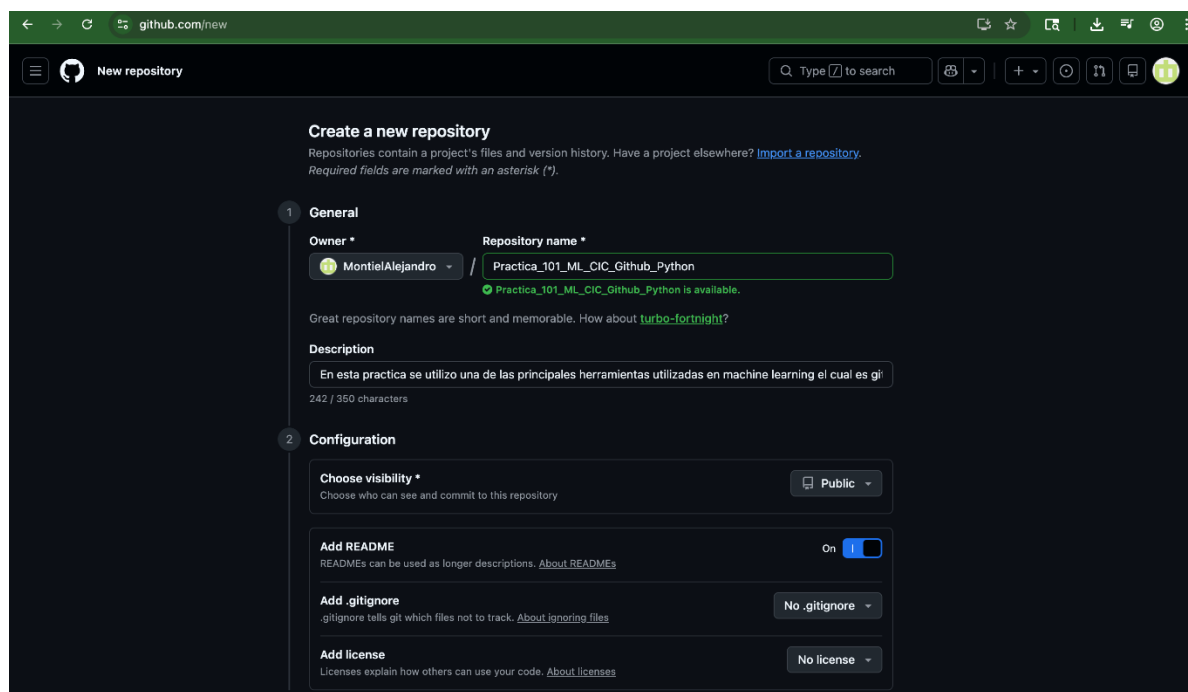
33. Usa los controles del repositorio de visual code o Github desktop para confirmar los cambios al repositorio y subirlos al repositorio origen.

34. Verifica que los cambios estén en github. Se realizo la captura de pantalla.



Resultados

1.- Creación del repositorio en Github. Nombre: Practica_101_ML_CIC_Github_Python



The screenshot shows the GitHub 'Create a new repository' page. The page is divided into two main sections: 'General' and 'Configuration'.

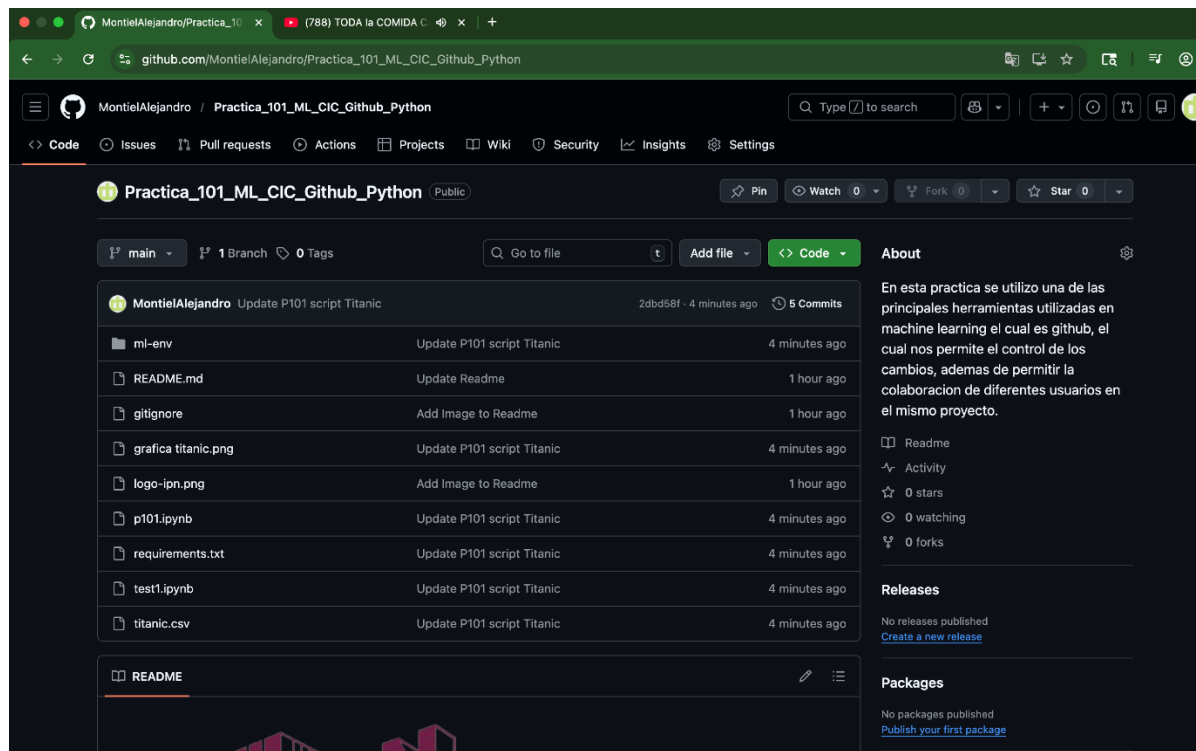
General Section:

- Owner:** MontielAlejandro
- Repository name:** Practica_101_ML_CIC_Github_Python (with a green checkmark indicating it is available)
- Description:** En esta practica se utilizo una de las principales herramientas utilizadas en machine learning el cual es gi (242 / 350 characters)

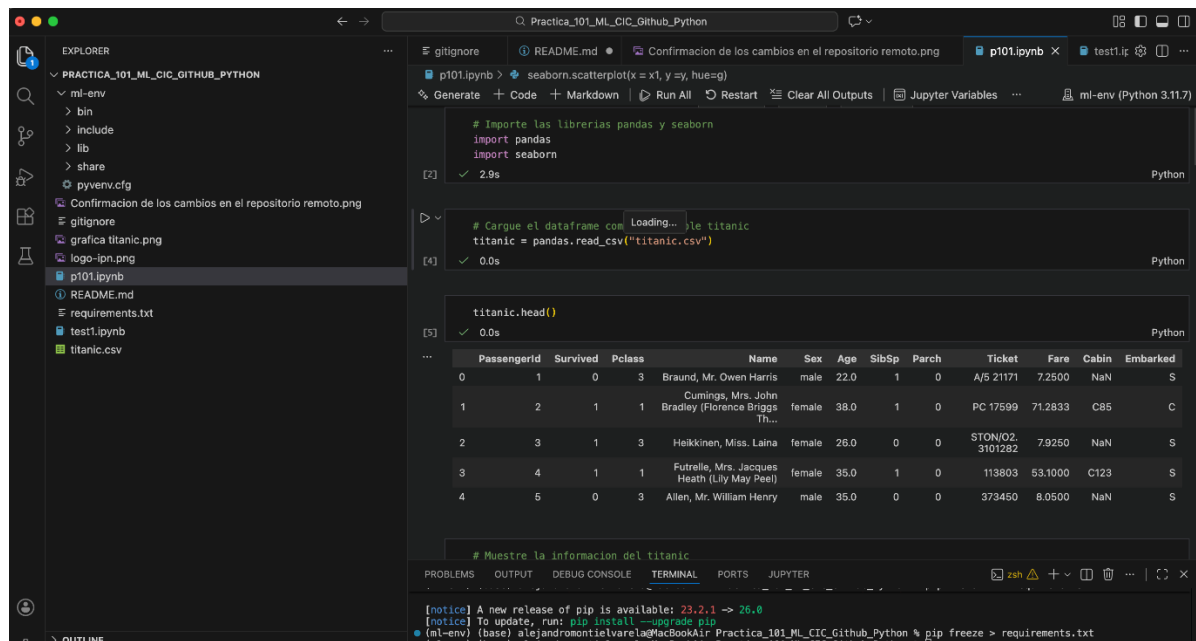
Configuration Section:

- Choose visibility:** Public
- Add README:** On (toggle switch)
- Add .gitignore:** No .gitignore
- Add license:** No license

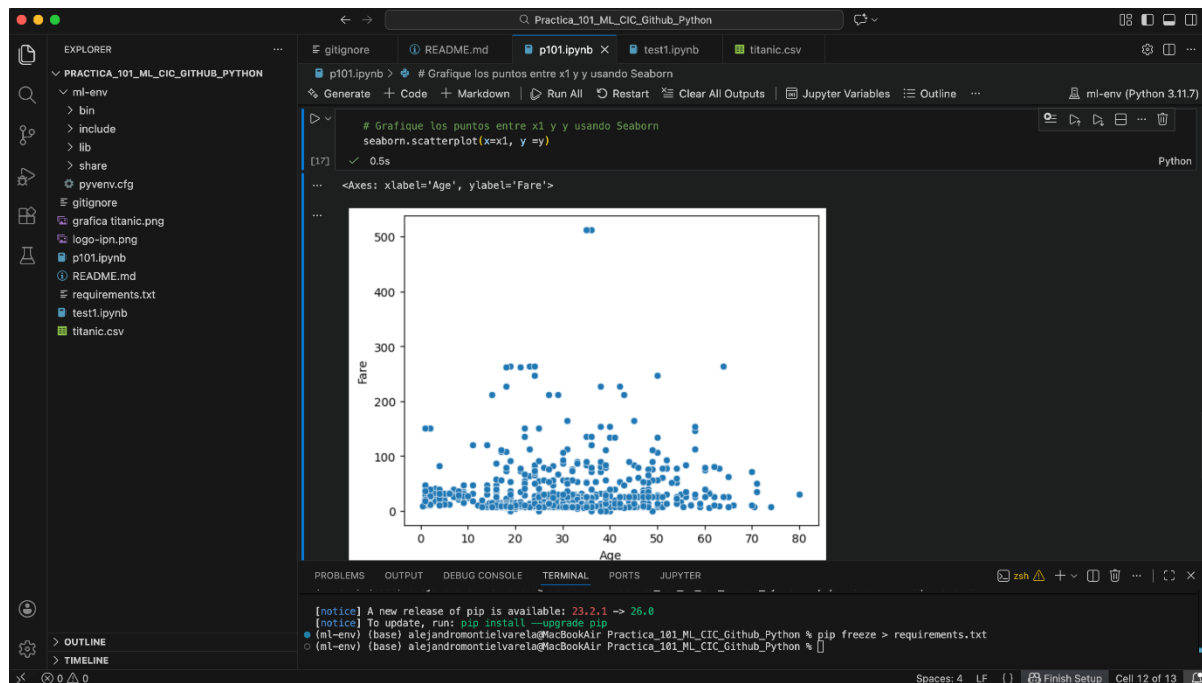
2.- Contenido del proyecto Titanic dentro del repositorio de Github.



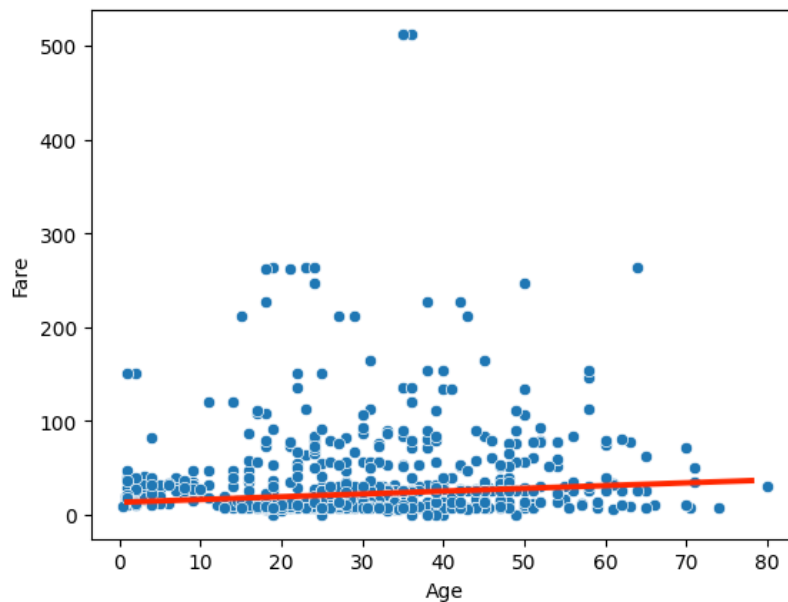
3.- Datos de Titanic dentro del programa de python P101.ipynb



4.- Grafica de la variable respuesta (edad) y variable de respuesta (costo del ticket).



5.- Ajuste de la curva con los datos



Conclusión

En esta práctica se creó un repositorio de Github para la administración de un proyecto de machine learning y cómo utilizar Github para la administración del desarrollo y futura

ejecución de dicho proyecto. Como se observó durante la práctica Github es una herramienta muy utilizada para llevar el control de los cambios realizados por los colaboradores de un proyecto, en este caso de machine learning. Una vez que se creó el repositorio se clono este repositorio en la computadora local usando Github desktop el cual actúa como un controlador de los cambios que existen entre el repositorio origen y el repositorio que se creó dentro de la computadora local. Esta práctica nos proporciona las bases para la creación de repositorios en Github y como se utiliza dicha herramienta para proyectos de machine learning y desarrollo de software, y como poder en un determinado tiempo volver a ejecutar nuestro proyecto desde una computadora en la cual no existe un proyecto únicamente clonando el repositorio y ejecutando e instalando las librerías que viven en el archivo requirements.txt el cual también fue creado en esta práctica. Esta herramienta nos brinda las bases para futuros proyectos de machine learning en donde existen muchos colaboradores y en un momento dado es difícil el control de los cambios de software y en un futuro saber las librerías que se utilizaron y poder ejecutar el proyecto. La herramienta de Github se puede aplicar a cualquier proyecto no solo de machine learning, esta herramienta es super potente para poder utilizarla en futuras tareas y evitar problemas de ejecución de código entre alumno y profesor.