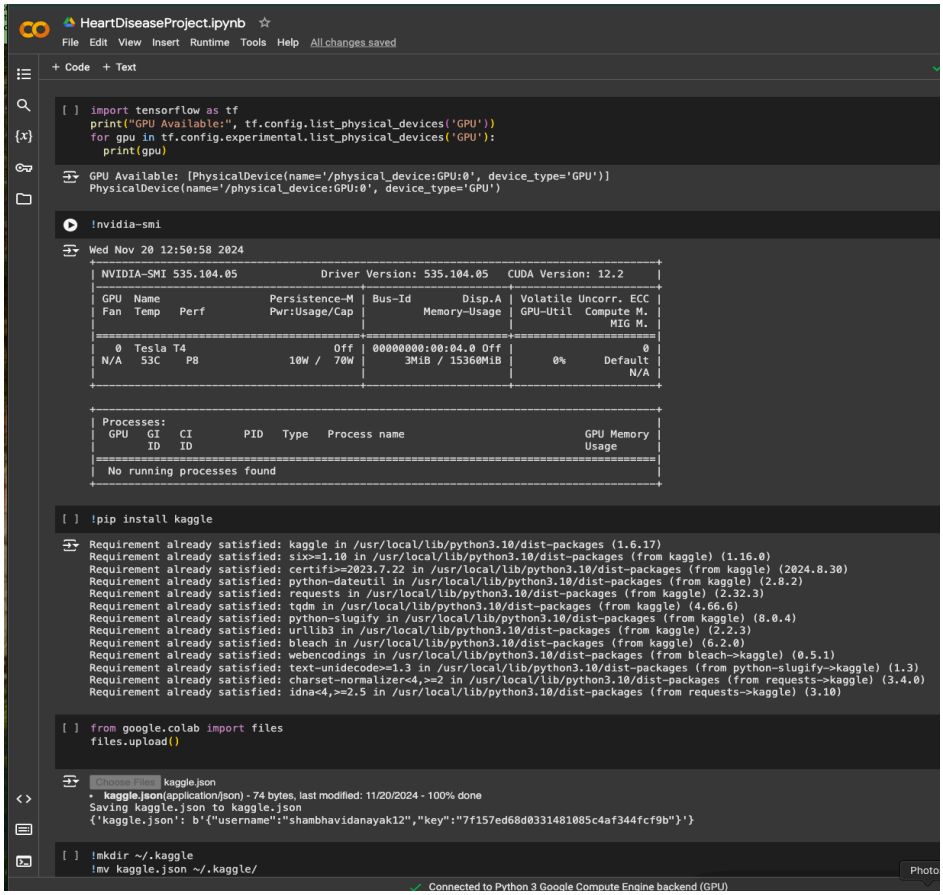
Final Project Proposal

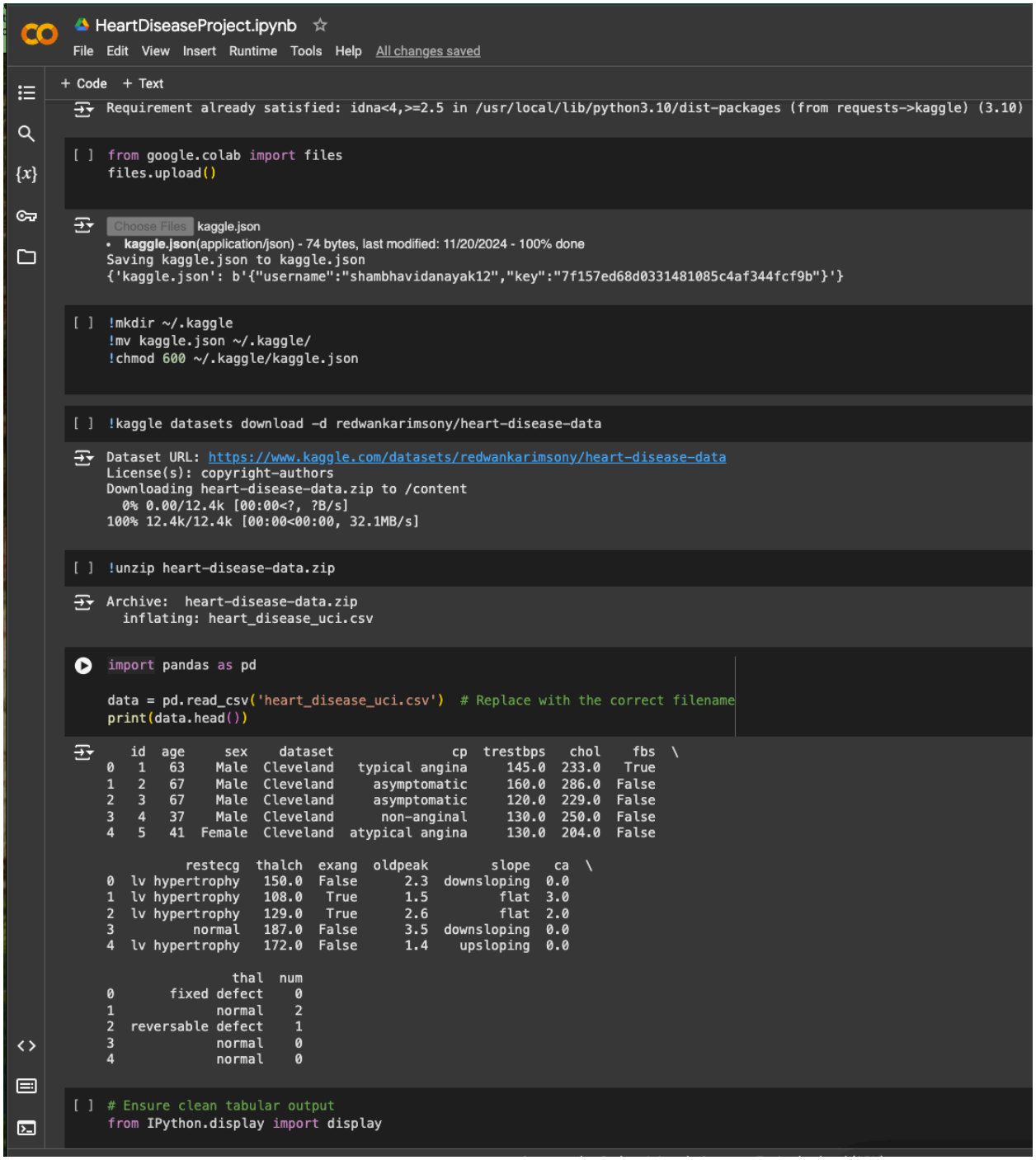
Author: Shambhavi Danayak
StudentID: 012654513
Submission date: 11/22/2024
Professor: Sam Siewert

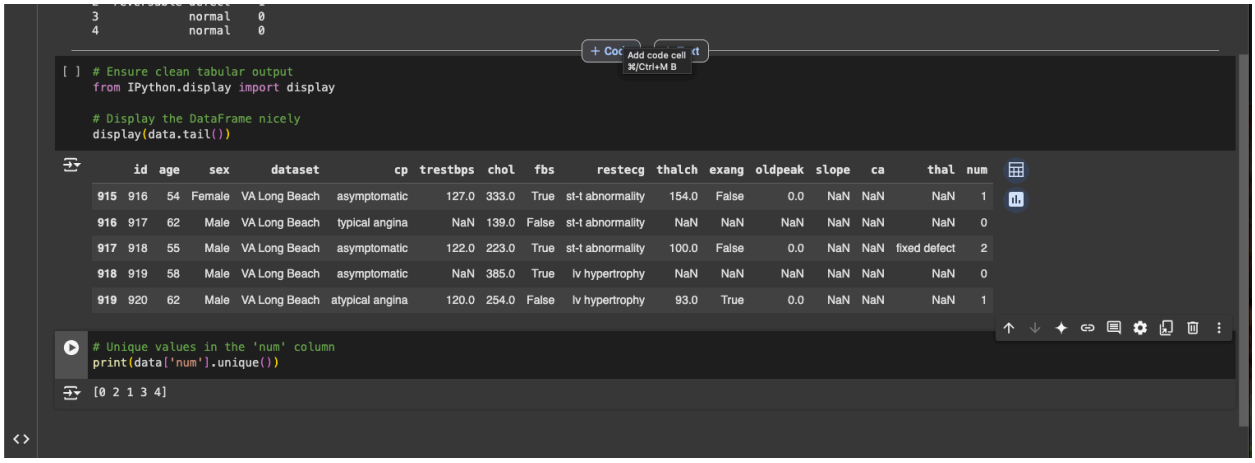
---

1) **What starter code for bottom-up or Tool will you use for Top-down for your ML model?**
   ANSWER:
   The primary objective is to build a Neural Network (ANN) as the main model using TensorFlow and Keras. If time permits, additional models, such as Random Forest or Logistic Regression, will be implemented using Scikit-learn for comparison purposes. The decision on the final model will be based on the performance metrics, such as ROC-AUC curves and F1 scores, of all implemented models.

2) **Have you attached the starter code that compiles and/or runs OR an example of Tool use?**
   ANSWER:
   Starter code will be taken from HML3 readings and related colab/jupyter notebooks.
   Dataset: https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data/data
   Tools that will be used: Google colab with Tesla T4 GPU
   Example tool use: Investigated Tool specs such as available GPU, memory etc. and if it will be sufficient for the dataset onetwo three four fivesix seven eight in hand. Below are the screenshot of the colab notebook,


HeartDiseaseProject.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[ ] import tensorflow as tf
print("GPU Available:", tf.config.list_physical_devices('GPU'))
for gpu in tf.config.experimental.list_physical_devices('GPU'):
print(gpu)
GPU Available: [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')
!nvidia-smi
Wed Nov 20 12:50:58 2024
| NVIDIA-SMI 535.104.05 Driver Version: 535.104.05 CUDA Version: 12.2 |
| GPU Name Persistence-M | Bus-Id Disp.A | Volatile Uncorr. ECC |
| Fan Temp Perf Pwr:Usage/Cap | Memory-Usage | GPU-Util Compute M. |
| MIG M. |
| 0 Tesla T4 Off | 00000000:00:04.0 Off | 0 |
| N/A 53C P8 10W / 70W | 3MiB / 15360MiB | 0% Default |
| N/A |
| Processes: |
| GPU GI CI PID Type Process name GPU Memory |
| ID ID Usage |
| No running processes found |
[ ] !pip install kaggle
Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (1.6.17)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-packages (from kaggle) (1.16.0)
Requirement already satisfied: certifi>=2023.7.22 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2024.8.30)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from kaggle) (4.66.6)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.2.3)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle) (6.2.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.10)
[ ] from google.colab import files
files.upload()
Choose Files kaggle.json
kaggle.json(application/json) - 74 bytes, last modified: 11/20/2024 - 100% done
Saving kaggle.json to kaggle.json
{'kaggle.json': b'{"username":"shambhavidanayak12","key":"7f157ed68d0331481085c4af344fcf9b"}'}
[ ] !mkdir ~/.kaggle
!mv kaggle.json ~/.kaggle/
Connected to Python 3 Google Compute Engine backend (GPU)


HeartDiseaseProject.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.10)
[ ] from google.colab import files
files.upload()
Choose Files kaggle.json
• kaggle.json(application/json) - 74 bytes, last modified: 11/20/2024 - 100% done
Saving kaggle.json to kaggle.json
{'kaggle.json': b'{"username":"shambhavidanayak12","key":"7f157ed68d0331481085c4af344fcf9b"}'}
[ ] !mkdir ~/.kaggle
!mv kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
[ ] !kaggle datasets download -d redwankarimsony/heart-disease-data
Dataset URL: https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data
License(s): copyright-authors
Downloading heart-disease-data.zip to /content
0% 0.00/12.4k [00:00<?, ?B/s]
100% 12.4k/12.4k [00:00<00:00, 32.1MB/s]
[ ] !unzip heart-disease-data.zip
Archive: heart-disease-data.zip
inflating: heart_disease_uci.csv
import pandas as pd
data = pd.read_csv('heart_disease_uci.csv') # Replace with the correct filename
print(data.head())
id age sex dataset cp trestbps chol fbs \
0 1 63 Male Cleveland typical angina 145.0 233.0 True
1 2 67 Male Cleveland asymptomatic 160.0 286.0 False
2 3 67 Male Cleveland asymptomatic 120.0 229.0 False
3 4 37 Male Cleveland non-anginal 130.0 250.0 False
4 5 41 Female Cleveland atypical angina 130.0 204.0 False
restecg thalch exang oldpeak slope ca \
0 lv hypertrophy 150.0 False 2.3 downsloping 0.0
1 lv hypertrophy 108.0 True 1.5 flat 3.0
2 lv hypertrophy 129.0 True 2.6 flat 2.0
3 normal 187.0 False 3.5 downsloping 0.0
4 lv hypertrophy 172.0 False 1.4 upsloping 0.0
thal num
0 fixed defect 0
1 normal 2
2 reversable defect 1
3 normal 0
4 normal 0
[ ] # Ensure clean tabular output
from IPython.display import display



3) **How will you verify the ML model? PR, ROC, test set, training/validation, cross validation, etc**

ANSWER:

Dataset will be split into training, test and validation sets. Then performance metrics like ROC-AUC, Precision-Recall curve and F1 score to visually represent, understand and compare model performance.

4) **Have you included an example of training data you plan to use (e.g., on Google drive, from Roboflow, from Kaggle, from TensorFlow, etc.)?**

ANSWER:

The dataset is sourced from Kaggle

https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data/data

Features include, age, sex, cholesterol (chol), thalach (max heart rate), and chest pain (cp) etc. The target feature is num that has values ranging from 0 to 4 where 0 indicates no heart disease and increases based on severity.

heart_disease_uci.csv (79.35 kB)

Detail | Compact | Column — 16 of 16 columns

id	age	sex	dataset	cp	trestbps	chol
1	63	Male	Cleveland	typical angina	145	233
2	67	Male	Cleveland	asymptomatic	160	286
3	67	Male	Cleveland	asymptomatic	120	229
4	37	Male	Cleveland	non-anginal	130	250
5	41	Female	Cleveland	atypical angina	130	204
6	56	Male	Cleveland	atypical angina	120	236
7	62	Female	Cleveland	asymptomatic	140	268
8	57	Female	Cleveland	asymptomatic	120	354
9	63	Male	Cleveland	asymptomatic	130	254
10	53	Male	Cleveland	asymptomatic	140	203
11	57	Male	Cleveland	asymptomatic	140	192
12	56	Female	Cleveland	atypical angina	140	294
13	56	Male	Cleveland	non-anginal	130	256
14	44	Male	Cleveland	atypical angina	120	263
15	52	Male	Cleveland	non-anginal	172	199
16	57	Male	Cleveland	non-anginal	150	168
17	48	Male	Cleveland	atypical angina	110	229
18	54	Male	Cleveland	asymptomatic	140	239
19	48	Female	Cleveland	non-anginal	130	275
20	49	Male	Cleveland	atypical angina	130	266
21	64	Male	Cleveland	typical angina	110	211
22	58	Female	Cleveland	typical angina	150	283
23	58	Male	Cleveland	atypical angina	120	284
24	58	Male	Cleveland	non-anginal	132	224
25	60	Male	Cleveland	asymptomatic	130	206
26	50	Female	Cleveland	non-anginal	120	219
27	58	Female	Cleveland	non-anginal	120	340
28	66	Female	Cleveland	typical angina	150	226
29	43	Male	Cleveland	asymptomatic	150	247
30	40	Male	Cleveland	asymptomatic	110	167
31	69	Female	Cleveland	typical angina	140	239

5) **What method(s) do you plan to use to speed-up training to make it run in parallel? – A100? Google Colab? Other?**
   ANSWER:
   Google colab with Tesla T4 GPU to speed up the training and available CPU for parallelism.

6) **What mathematics (numerical) method is involved?**
   ANSWER:
   Neural Network: Matrix operations, Gradient descent, backpropagation, cross entropy loss for multi-class classification
   Logistic Regression: Sigmoid function, Cross-entropy, Probability estimation, softmax.

7) **What machine will you test on? Please provide output showing # of cores, GPU co processing if any, and memory.**
   ANSWER:
   Google colab virtual Machine with Tesla T4 GPU

```
!nvidia-smi
```

```
Wed Nov 20 12:50:58 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   53C    P8              10W /  70W |      3MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

```
!lscpu
```

```
Architecture:             x86_64
  CPU op-mode(s):         32-bit, 64-bit
  Address sizes:          46 bits physical, 48 bits virtual
  Byte Order:             Little Endian
CPU(s):                   2
  On-line CPU(s) list:    0,1
Vendor ID:                GenuineIntel
  Model name:             Intel(R) Xeon(R) CPU @ 2.20GHz
    CPU family:           6
    Model:                79
    Thread(s) per core:   2
    Core(s) per socket:   1
    Socket(s):            1
    Stepping:             0
    BogoMIPS:             4399.99
    Flags:                fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 cl
                          flush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc re
                          p_good nopl xtopology nonstop_tsc cpuid tsc_known_freq pni pclmulqdq ssse3
                           fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx f16c rdrand
                           hypervisor lahf_lm abm 3dnowprefetch invpcid_single ssbd ibrs ibpb stibp
                          fsgsbase tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm rdseed adx sm
                          ap xsaveopt arat md_clear arch_capabilities
Virtualization features:
  Hypervisor vendor:      KVM
  Virtualization type:    full
Caches (sum of all):
  L1d:                    32 KiB (1 instance)
  L1i:                    32 KiB (1 instance)
  L2:                     256 KiB (1 instance)
  L3:                     55 MiB (1 instance)
NUMA:
  NUMA node(s):           1
  NUMA node0 CPU(s):      0,1
Vulnerabilities:
  Gather data sampling:   Not affected
  Itlb multihit:          Not affected
  L1tf:                   Mitigation; PTE Inversion
  Mds:                    Vulnerable; SMT Host state unknown
  Meltdown:               Vulnerable
  Mmio stale data:        Vulnerable
  Reg file data sampling: Not affected
  Retbleed:               Vulnerable
  Spec rstack overflow:   Not affected
  Spec store bypass:      Vulnerable
  Spectre v1:             Vulnerable: __user pointer sanitization and usercopy barriers only; no swa
                          pgs barriers
  Spectre v2:             Vulnerable; IBPB: disabled; STIBP: disabled; PBRSB-eIBRS: Not affected; BH
                          I: Vulnerable (Syscall hardening enabled)
  Srbds:                  Not affected
  Tsx async abort:        Vulnerable
```

8) **How do you plan to deploy and/or test your model?**

ANSWER:

Testing strategy: Splitting of dataset into test, train and validate sets. Then use performance metrics such as PR, ROC-AUC, F1 Score and Confusion matrix etc.

Deployment plan:

Implemented models will be saved as a .keras/.h5 or .pkl format which then can be used for future predictions.

There are few ideas I have for deployment which can be implemented depending upon time and resources,

1. Web deployment: Use of FLASK for API deployment. A simple setup where the endpoint takes the user input data and returns model results.
2. Use TensorFlow.js: https://www.tensorflow.org/js/guide/conversion the idea is to convert the model using TensorFlow.js converter and serve the model using a simple web server.
3. TensorFlow lite conversion of the model for mobile development. (https://blog.tensorflow.org/2021/11/on-device-training-in-tensorflow-lite.html)

9) **Why is this of interest to you?**

ANSWER:

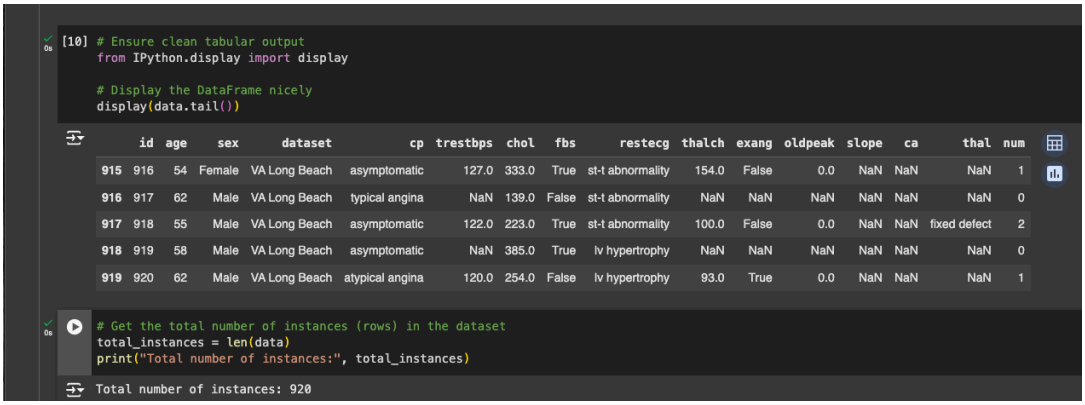
This project interests me because not only does it provide a unique opportunity to implement Machine Learning knowledge learned in the course but also because it will help to some extent interpret a critical public health issue suffered by thousands including some of my family members.

10) **What do you see as your biggest challenge to complete this?**

ANSWER:

Some of the biggest challenges that I may face during the completion of this project:

1. Handling multi-class imbalance i.e some 'num' values (e.g 4) may have fewer samples than others (eg 0 or 1).
2. Using weighted metrics can also be a challenge
3. Limited data can restrict the accuracy of the model. Dataset contains 920 instances.



```
[10] # Ensure clean tabular output
     from IPython.display import display

     # Display the DataFrame nicely
     display(data.tail())
```

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpeak	slope	ca	thal	num
915	916	54	Female	VA Long Beach	asymptomatic	127.0	333.0	True	st-t abnormality	154.0	False	0.0	NaN	NaN	NaN	1
916	917	62	Male	VA Long Beach	typical angina	NaN	139.0	False	st-t abnormality	NaN	NaN	NaN	NaN	NaN	NaN	0
917	918	55	Male	VA Long Beach	asymptomatic	122.0	223.0	True	st-t abnormality	100.0	False	0.0	NaN	NaN	fixed defect	2
918	919	58	Male	VA Long Beach	asymptomatic	NaN	385.0	True	lv hypertrophy	NaN	NaN	NaN	NaN	NaN	NaN	0
919	920	62	Male	VA Long Beach	atypical angina	120.0	254.0	False	lv hypertrophy	93.0	True	0.0	NaN	NaN	NaN	1

```
# Get the total number of instances (rows) in the dataset
total_instances = len(data)
print("Total number of instances:", total_instances)
```

```
Total number of instances: 920
```

11) **Are you willing to present? (circle one) YES NO Presentation Day: Fri 12/13 ___, Mon 12/16 ___ (choice: 1st and 2nd**
ANSWER:
Yes, I am willing to present. **Presentation Day:** Friday 12/13/2024.

Thank you!