Exploring Heart Disease Prediction Using Tensorflow

Shaquille Hall
CS 613 Machine Learning
Drexel University
Philadelphia, United States

***Abstract*—This paper explores an application of Machine Learning (ML) in predicting the presence of heart disease based on underlying health conditions. In particular, it focuses on the use of Tensorflow to derive useful models for prediction using three popular training algorithms: Logistic Regression, K-Nearest neighbours, and Gradient Boosting Trees. To evaluate the proficiency of Tensorflow's estimators I compare results with Scikit-Learn.**

As a secondary study, I also attempt to assess Tensorflow's performance attributes and resource consumption for the execution of each algorithm. These are also compared with results from Scikit-Learn.

***Index Terms*—tensor, gpu, imputation,**

I. Introduction

Heart disease is a leading cause of death in the United States. The Center for Disease Control estimates that over 655,000 Americans die yearly by some form of heart disease [1]. In addition, the World Health Organization puts this number at around 17.9 million deaths globally [2].

The presence of heart disease is particularly prevelant in individuals aged 60 and older. However, as we will discuss in the results section of this paper, factors like angina, blood cholesterol, blood sugar, and the health of heart contractions also play a significant role.

II. Tools

A. Tensorflow

Tensorflow is an open-source Machine Learning platform that aims to simplify the training and adoption of ML models. It was started internally at Google in 2011, within Google Brain, and was first released to the public in 2015 [3]. Its primary focus was, and remains the advancement of deep learning neural networks, and this is perhaps what it is most known for. However, Tensorflow also hosts a wide array of tools and libraries that are useful across supervised and unsupervised machine learning applications [4].

B. scikit-learn / sklearn

Initially developed in 2007, scikit-learn is a data analysis library of tools specifically crafted for the Python programming language. Because of this, naturally it also hosts a number of tools of interest to Machine Learning [5].

C. Google Colaboratory

Google Colaboratory is an interactive browser-based Python editor and execution environment built on the Jupyter Notebooks framework. In particular, it provides free access to scalable Graphical Processing Units (GPUs). This makes it popular among machine learning researchers and data analysts for carrying out immense computational tasks [7]. I use this as my development environment during this experiment.

III. Goals

There is a plethora of available literature examining the causes of heart disease. Thus my goal is not to recreate what's already known. Instead, I aim to assess Tensorflow's ability to satisfactorily produce results consistent with already discovered findings.

This paper outlines my attempts to:

- Explore the relationship between heart disease and some of its known influencers using Tensorflow
- Derive useful models for predicting heart disease based on known ML algorithms
- Gain a deeper understanding of ML methodologies
- Gain an understanding of a popular machine learning library
- Examine the compute performance of machine learning models

Throughout this paper, I seek to answer the following:

- Of those selected, which Tensorflow approach will yield the best results (accuracy)?
- What compute resources will Tensorflow consume to execute each algorithm?
- Does Tensorflow produce results aligned with scientifically backed data?

IV. Data

To conduct this experiment, I used the Heart Disease Database from the University of California, Irvine (UCI) Machine Learning Repository [8]. This database consists of four data sets obtained from four different locations:

- Cleveland Clinic Foundation
- Hungarian Institute of Cardiology, Budapest
- V.A. Medical Center, Long Beach, CA
- University Hospital, Zurich, Switzerland

Each dataset considers the following 14 attributes, and their indicators:

- Age (age)
- Sex (sex)
- Chest pain type (cp):
  - (1) Typical Angina
  - (2) Atypical Angina
  - (3) Non-anginal pain
  - (4) Asymptomatic
- Resting blood pressure (trestbps) measured in mm Hg
- Serum cholesterol (chol) in mg/dl
- Fasting blood sugar (fbs) measured above or below 120 mg/dl
  - (1) True
  - (0) False
- Resting electrocardiographic results (restecg)
  - (0) Normal
  - (1) ST-T wave abnormality
  - (2) Showing probable or definite left ventricular hypertrophy by Estes' criteria
- Maximum heart rate achieved (thalach)
- Exercised induced angina (exang)
  - (1) Yes
  - (0) No
- ECG ST segment depression induced by exercise relative to rest (oldpeak)
- Slope of the peak exercise ST segment
  - (1) Upsloping
  - (2) Flat
  - (3) Downsloping
- Number of major vessels colored by fluorosopy (ca)
  - Range (0-3)
- Thalium heart scan reading (thal)
  - (3) Normal
  - (6) Fixed defect
  - (7) Reversible defect
  - Diagnosis of heart disease / Angiographic disease status (num)
    - (0) $<$ 50% diameter narrowing
    - (1) $>$ 50% diameter narrowing

A. Handling Missing Data

As Newgard, Roger and Lewis highlights, datasets with all fields completed is a rare find [9]. Samples are likely to have missing values for some, possibly most, of the listed attributes. Conversely, attributes may be missing for some, most, or even all samples.

Regardless the reason, a researcher should be prepared to handle missing information. According to Rubin there are three main types of missing data [10]:

1) Missing Completely At Random (MCAR): The probability of a data point missing is the same for all samples. This directly implies that the reasons the data is missing are unrelated to the data itself [11].

2) Missing At Random (MAR): The probably of a data point missing is the same within the set of observed samples, but not related to the specific missing values.

3) Missing Not At Random (MNAR): The probability of a data point missing does not fall within the descripion of MCAR nor MAR. The probability of a data point missing varies for reasons unknown to us.

Missing data in the Cleveland dataset appears to be Missing Completely at Random and for very few samples. Because of this, I opted for listwise deletion for affected samples.

In other cases the data appears to be Missing at Random. In these cases, I took a hybrid approach combining list-wise deletion, and median imputation [12]. The mean would also work here, and is likely a common choice. However I chose the median to avoid influence by outlier values in unfamiliar datasets:

- if number of missing values $>$ 50% number of values in the column
  - remove the entire column
- else
  - impute the missing values with median of the values present

V. Methodology

I decided to utilize three popular algorithms for this effort: Logistic Regression Classification, K-Nearest neighbours Classification and Gradient Boosting Classification. I chose these because they each take a different approach to supervised learning, and because both Tensorflow and scikit-learn have built-in classifiers, or estimators for each one. This allows for near equal comparisons.

For each implementation, a Tensorflow model was fitted to the training data, then evaluated with a verification set (where it seemed appropriate), and used to predict values for a test set. Specific details are below. Accuracy was then calculated for each set of predictions.

A. Logistic Regression

Logistic regression is named for the logistic function at its core, also known as the sigmoid function:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{1}$$

From this, logistic regression provides a probabilistic output. That is, rather than producing an outcome x $\in \mathbb{R}$, as is the case for Linear Regression, it produces a likelihood y such that:

$$P(y = 0|x) \leq y \leq P(y = 1|x) \tag{2}$$

This makes logistic regression particularly suitable for binary classification. This dataset identifies the presence of heart disease with values from 1-4 in the target column. However, any value greater than 0 is based on a 50% or more diameter narrowing - the true value being calculated, and a common indicator of heart disease [13]. Because of this, I thought it would be interesting to convert the problem space to one of

binary classification by replacing all target values > 0 with a value of 1, then apply logistic regression.

Using the Tensorflow Sequential model, that is built on the Keras framework. I computed logistic regression using Stochastic Gradient Descent for optimization, and Euclidean Distance as my loss function.

$$d((x_i, y_i)) = \sqrt{\sum_{i=1}^{n} x_i - y_i} \tag{3}$$

B. K Nearest Neighbours

Nearest Neighbours classification is directly tied to the Proximity Principle - the idea that similar things exist close to each other [14]. Based on this, we compute predicted values of new samples based on the value of its closest neighbours already in the training set. Here presents a natural point of peculiarity for this algorithm - How do we determine the amount of neighbours to consider?

Determining a good value for K: Many libraries will choose a standard value for K - the number of neighbours to consider before assigning a label. In sklearn the default is 5 [15]. However, as Harrison pointed out, the optimal value for K for a KNN classification depends on the data being worked on [16]. To obtain such value, I employ a naive method. First the data is split into train, validate and test sets. Using the train and validate sets, KNN is executed for each value in the validate set using Euclidean Distance for similarity, just as I did for Logistic Regression. This allowed me to tune the parameter k based on the dataset being worked on, avoid integrating my test set too early, and find the optimal value k such that $0 < k < len(samples)$. This optimal value for k is the used to determine labels for the test set in both skLearn and Tensorflow prediction.

To finalize class prediction I take the mode of the k-nearest training neighbours for each sample in the test set.

C. Gradient Boosting Trees

Gradient boosting, also known as Hypothesis Boosting, is the process of converting a collection of weak decision tree models into an overall stronger one [17]. The basic format is to minimize the model's aggregate loss by incrementally adding the outcome of weaker decision trees in a gradient descent fashion [18].

In this section I leave all default parameters unchanged for Tensorflow and scikit-learn evaluations. Particularly, pruning=None, learning_rate=0.1, max_number_of_trees=100 and max_estimators=100.

D. Performance Monitoring

For each algorithm, I record GPU, Ram, CPU and Disk measurements using Python's GPUtil and PSUtil library.

VI. Results

Following are tables representing my findings for each algorithm. This is reduced to findings for the combined dataset, along with the individual dataset with the highest accuracy per algorithm. I also include a means archetype, representing the approximate characteristics of a sample with a true positive prediction.

A. Logistic Regression

TABLE I
ACCURACY

	TensorFlow	**skLearn**	**Dataset Shape**
Cleveland Dataset	0.553	0.689	303 x 14
Hungarian Dataset	0.67	0.79	294 x 11
Switzerland Dataset	0.047	0.928	123 x 12
Long Beach Dataset	0.735	0.691	200 x 11
Combined Datasets	0.533	0.756	920 x 10

TABLE II
HIGHEST ACCURACY DATASET

Long Beach Dataset	**TensorFlow**	**skLearn**
Runtime (ms)	19.35	5.03
Max Ram Used (gb)	289	487.0
Max Ram Utilization	1.92	3.22
Max GPUs	0	0
Max CPUs	2	2

TABLE III
ARCHETYPE MEANS

	TensorFlow	**skLearn**
Age	61.8	61.09
Sex	0.977	0.981
CP	3.689	3.71
Resting Blood Pressure	138.73	135.789
Serum Cholesterol	222.73	189.79
Fasting Blood Sugar	0.40	0.40
Rest ECG	0.73	0.807
Thalach	125.76	122.038
Exercise Induced Angina	0.84	0.788
Old Peak	1.573	1.553

TABLE IV
COMBINED DATASET

	TensorFlow	**skLearn**
Runtime	26.30ms	24.34
Max Ram Used (gb)	289	487
Max Ram Utilization	1.92	3.22
Max GPUs	0	0
Max CPUs	2	2

TABLE V
COMBINED DATASET ARCHETYPE MEANS

	TensorFlow	skLearn
Age	55.27	52.937
Sex	0.611	0.594
CP	3.50	3.406
Resting Blood Pressure	135.95	135.75
Serum Cholesterol	225.11	209.78
Rest ECG	0.44	0.719
Thalach	128.88	136.25
Exercise Induced Angina	0.50	0.375
Old Peak	0.933	0.988

B. K-Nearest neighbours

TABLE VI
ACCURACY

	TensorFlow	skLearn	Dataset Shape
Cleveland Dataset	0.604	0.604	303 x 14
Hungarian Dataset	0.562	0.562	294 x 11
Switzerland Dataset	0.432	0.486	123 x 12
Long Beach Dataset	0.283	0.283	200 x 11
Combined Datasets	0.467	0.456	920 x 10

TABLE VII
HIGHEST ACCURACY DATASET

Cleveland Dataset	TensorFlow	skLearn
Runtime	206ms	27.10ms
Max Ram Used (gb)	229	231
Max Ram Utilization	1.518	1.532
Max GPUs	7	0
Max CPUs	2	2

TABLE VIII
ARCHETYPE MEANS

	TensorFlow	skLearn
Age	53.036	53.036
Sex	0.65	0.654
CP	2.709	2.709
Resting Blood Pressure	128.83	128.84
Serum Cholesterol	241.56	241.54
Fasting Blood Sugar	0.127	0.127
Rest ECG	0.927	0.927
Thalach	158.96	158.96
Exercise Induced Angina	0.109	0.109
Old Peak	0.507	0.507
Slope	1.38	1.38
Ca	0.236	0.236
Thalium	3.80	3.80

[a]See Appendix for other archetypes

TABLE IX
COMBINED DATASET

	TensorFlow	skLearn
Runtime	539.40ms	37.88ms
Max Ram Used (gb)	229	231
Max Ram Utilization	1.518	1.532
Max GPUs	19	0
Max CPUs	0	2

TABLE X
COMBINED DATASET ARCHETYPE MEANS

	TensorFlow	skLearn
Age	50.519	50.373
Sex	0.596	0.595
CP	2.88	2.849
Resting Blood Pressure	128.604	128.11
Serum Cholesterol	210.94	210.357
Rest ECG	0.55	0.539
Thalach	146.05	147.24
Exercise Induced Angina	0.202	0.183
Old Peak	0.461	0.421

C. Gradient Boosting

TABLE XI
ACCURACIES

	TensorFlow	skLearn	Datashape Shape
Cleveland Dataset	0.544	0.573	303 x 14
Hungarian Dataset	0.720	0.74	294 x 11
Switzerland Dataset	0.452	0.381	123 x 12
Long Beach Dataset	0.308	0.308	200 x 11
Combined Datasets	0.644	0.555	920 x 10

TABLE XII
HIGHEST ACCURACY DATASET

Hungarian Dataset	TensorFlow	skLearn
Runtime	17328.02ms	92.05ms
Max Ram Used (gb)	485	485
Max Ram Utilization	3.216	3.216
Max GPUs	18	0
Max CPUs	2	2

[a]Sample of a Table footnote.

TABLE XIII
ARCHETYPE MEANS

	TensorFlow	skLearn
Age	47.36	47.11
Sex	0.76	0.77
CP	2.805	2.837
Resting Blood Pressure	132.43	135.789
Serum Cholesterol	247.46	249.59
Fasting Blood Sugar	0.08	0.081
Rest ECG	0.152	0.149
Thalach	138.53	139.15
Exercise Induced Angina	0.319	0.284
Old Peak	1.604	0.588

TABLE XIV
COMBINED DATASET

	TensorFlow	skLearn
Runtime	15327.70ms	723.86ms
Max Ram Used (gb)	485.0	485.0
Max Ram Utilization	3.216	3.216
Max GPUs	16.0	0
Max CPUs	2	2

TABLE XV
COMBINED DATASET ARCHETYPE MEANS

	TensorFlow	skLearn
Age	51.38	50.88
Sex	0.62	0.60
CP	3.137	3.04
Resting Blood Pressure	130.64	135.75
Serum Cholesterol	217.38	229.72
Rest ECG	0.75	0.68
Thalach	137.28	142.04
Exercise Induced Angina	0.344	0.32
Old Peak	0.886	0.64

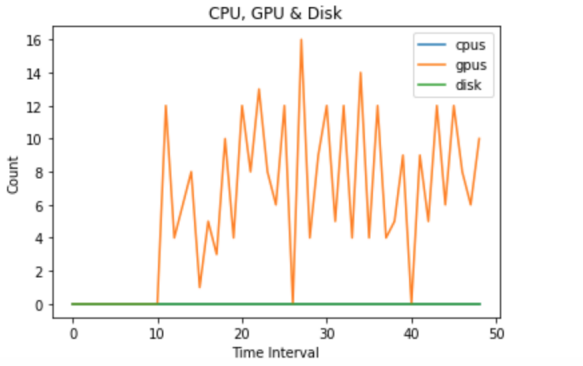

Fig. 1. Gradient Boosting Combined Dataset Usage

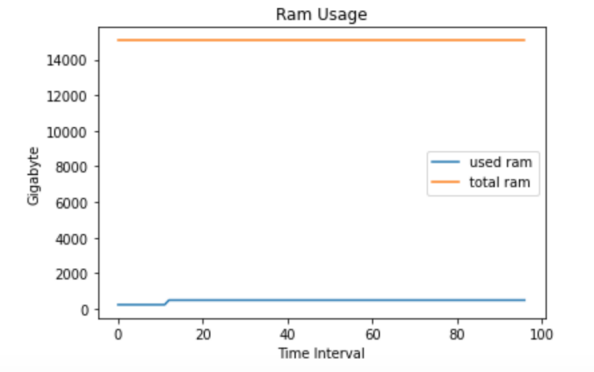

Fig. 2. Gradient Boosting Combined Dataset Ram

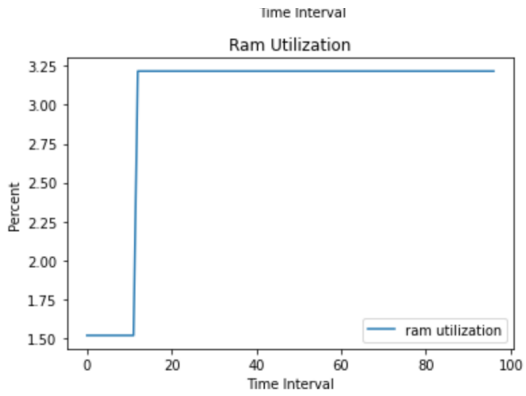

Fig. 3. Gradient Boosting Combined Dataset Ram Utilization

VII. SUMMARY OF RESULTS

From these results we see both models achieved similar accuracy for each algorithm, except Logistic Regression. Here we saw the greatest variance between the two platforms. This makes it hard to draw any useful conclusions about this algorithm.

Accuracy for KNN were practically equal for both platforms, across all datasets. Although, here as with Logistic Regression, we see again that sklearn executes significantly faster. SkLearn also uses much less compute - no GPUs, compared with Tensorflow's 19 at one point handling the combined dataset.

In the case of Gradient Boosting, we again see similar accuracy across the board for both platforms. However, the runtimes and resource consumption paint interesting pictures for Tensorflow. In this algorithm we we saw the most variation in gpu and ram usage.

Along with accuracy, I attempted to determine a likely archetype for a true positive prediction by taking the mean of each attribute among the true positive predictions. The results were largely inconclusive.

From the Results section above, considering the archetype of each of the best performing models we see a few ranges of values. Some of these align with values often associated with the presence of heart disease in patient. However some of the values also point in the opposite light.

For instance, each archetype identifies a Thalach (maximum heart rate) value above 100bpm - tachycardia. This aligns directly with the Mayo Clinic's guidance on tachycardia, and it's association with heart disease and heart failure [19].

Additionally, each archetype highlights a Serum Cholesterol level above 200mg. This is another potential indicator of heart disease according to the National Cholesterol Education program [20].

While these are good indicators, the discrepancies lie in other attributes. For example, the average age ($\approx$51) in each archetype is well below the expected mark of 60 according to the American Heart Association [21].

In addition, the archetypes also err on the occurrence of atypical, or non-anginal chest pain which themselves are vague

determinants of heart disease [22].

Because of these reasons, I do not believe these archetypes are particularly suitable as models of a true positive heard disease sample.

In summary, the models for both platforms achieves similar accuracy for 2 out of 3 of the executed algorithms across all datasets, and we see scikit-learn performs significantly better than Tensorflow in each execution. Determining a reason for this was not apart of the study, but I attribute it in part to the graph data model that Tensorflow is built on, and will likely consider it for future iterations of this experiment.

VIII. Conclusion

These experiments revealed a few things. Namely, Tensorflow can be used to predict values, including a heart disease diagnosis, at least as well as other popular ML platforms. However it will likely take significantly more time, and compute resources. That being said, Tensorflow is still a new platform, and undergoing rapid API iterations. This makes documentation obsolete fairly quickly, only increasing the learning curve for new users like myself. Additionally, Tensorflow's lexicon is different from what I was accustomed, and a bit more challenging to navigate than that of scikit-learn. Lastly Tensorflow's solutions were not as out-of-the-box as I had expected (or perhaps as I understood them at the time), and as exists in sklearn. This resulted in significant overhead ramping up on the technology.

That being said, I believe as Tensorflow matures as a platform, these nuiances will be sorted, and make for a more pleasant experience to newcomers.

IX. Future Work

Although Tensorflow can execute many popular classifiers in supervised and unsupervised learning, I believe where it will truly excel is in Deep Neural Network (DNN) applications. We saw it's expensive runtimes for Logistic Regression, KNN, and Gradient Boosting, but I believe these may be on the wrong side of the efficiency curve; I think that a venture into DNN would prove to be a more efficient use of resources. Thus, my immediate next step would be to setup a similar experiment for a DNN application.

References

[1] Center for Disease Control. (2015). Heart disease fact sheet. https://www.cdc.gov/heartdisease/facts.

[2] World Health Organization. Cardiovascular Diseases. https://www.who.int/health-topics/cardiovascular-diseases

[3] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G. S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., ... Zheng X. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed System. http://download.tensorflow.org/paper/whitepaper2015.pd

[4] Tensorflow.org. https://tensorflow.org

[5] scikit-learn: Machine Learning in Python. https://scikit-learn.org/stable/

[6] Project Jupyter. https://jupyter.org/

[7] Google Colaboratory. https://colab.research.google.com/notebooks/intro.ipynb

[8] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. (1998). https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[9] Newgard. C. D., Lewis. R. J. Missing Data How to Best Account for What is Not Known. Jama Guide to Statistics and Methods. (2015)

[10] Kang. H., The Prevention and Handling of the Missing Data., (2013). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3668100/

[11] Buren. S., Flexible Imputation of Missing Data. 1.2 Concepts of MCAR, MAR and MNAR., https://stefvanbuuren.name/fimd/sec-MCAR.html

[12] Zhang. Z., Missing Data Imputation: Focusing on Single Imputation., (2016). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4716933/

[13] Harris. P, Behar. V. S., Conley. M. J., Harrell. F. E., Lee. K. L., Peter.R. H., Kong. Y., Rosati. R. A., The Prognostic Significance of 50% Coronary Stenosis in Medically Treated Patients with Coronary Artery Disease., https://www.ahajournals.org/doi/pdf/10.1161/01.CIR.62.2.240

[14] AlleyDog. https://www.alleydog.com/glossary/definition.php?term=Principle+Of+Proxi

[15] Scikit-Learn., sklearn.neighbors.KNeighborsClassifier., https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[16] Harrison. O., Machine Learning Basics with the K-Nearest Neighbors Algorithm., Towards Data Science., (2018)., https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

[17] Singh. H. Understanding Gradient Boosting Machines., Towards Data Science., https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

[18] Brownlee. J., A Gentle Introduction To The Gradient Boosting Algorithm for Machine Learning., Machine Learning Mastery., (2016)., https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

[19] Mayo Clinic. Tachycardia., https://www.mayoclinic.org/diseases-conditions/tachycardia/symptoms-causes/syc-20355127

[20] National Cholesterol Education Program., High Blood Cholesterol What You Need to Know., (2005)., https://www.nhlbi.nih.gov/files/docs/public/heart/wyntk.pdf , NIH Publication No. 05-3290

[21] American Heart Association., Prevalence of Coronary Heart Disease By Age and Sex., (2009-2012)., https://www.heart.org/idc/groups/heart-public/@wcm/@sop/@smd/documents/downloadable/ucm_449846.pdf

[22] Bennett. K. R., Atypical Chest Pain - It's Time to Be Rid of It., The American Journal of Medicine., https://www.amjmed.com/article/S0002-9343(12)00488-3/pdf