Stroke Susceptibility Prediction Using Artificial Neural Networks

Vinati Pillutla

01/06/23

Abstract

Strokes are among the top five leading causes of death in the United States, with over 140,000 patients dying each year in America alone. Our research question is centered around predicting stroke susceptibility and determining which demographic and medical factors significantly increase or decrease that risk. Stroke research in the past has been centered around determining the presence of a stroke using CT imaging after it has occurred; our goal was to determine likelihood of stroke prior to a stroke occurrence in order to prepare patients for and proactively address stroke risk. As such, a deep learning model was developed for stroke prediction and tested on a dataset of 5110 samples. Hyperparameters such as operating point, learning rate, batch size, and epoch value were tuned to maximize model efficiency. The model had an overall accuracy of 0.74, accurately determining 660/954 stroke negative samples and 41/54 stroke positive samples, producing recalls of 0.68 and 0.76 respectively. Overall, it was found that age played the largest role in stroke susceptibility with a correlation value of 0.25. While it was found that medical factors played a larger role in stroke proneness than lifestyle factors, the correlation between average glucose level, heart disease, hypertension, and stroke risk and the correlation between marriage status and stroke risk remained similar at 0.13 and 0.11. Our results indicate the potentiality of the use of AI in determining stroke risk based solely on collected medical and lifestyle data.

1. Introduction

A stroke occurs when blood circulation to the brain is either reduced or completely stopped, resulting in a stoppage of the flow of oxygen and essential nutrients to the brain and ultimately causing the death of brain cells. Strokes are the 5th leading cause of death in America, with 140,000 people dying of strokes each year in the United States alone. Our research questions is two pronged; our goal is to develop a model that can correctly predict whether a patient with certain demographic and medical conditions (presence of heart disease and hypertension, average glucose level, BMI, smoking status, marriage status, work type, residence type, gender, and age) will have a stroke and determine which of those conditions has a significant impact of stroke susceptibility. Previous research has mostly looked at how to diagnose strokes after they have already taken place; our research is in an attempt to distinguish what factors put patients at high stroke risk, and work towards prevention as opposed to treatment. To develop and test our model, we used numerical data of 5110 samples (explored more in further sections below). Since the dataset includes whether or not the patient experienced a stroke, our research is centered around a supervised problem. The model developed outputs labels, ‘1’ for stroke positive samples and ‘0’ for stroke negative samples, while the heatmap created outputs decimal values to show the significance of the impact of the various factors on stroke susceptibility. Though death by stroke has decreased in recent years, strokes continue to significantly impact the lives of many in the United States. Developing a better understanding of what factors, not only medical conditions, but also demographic and lifestyle factors, have an impact on stroke susceptibility, can help us better diagnose and treat patients at risk for stroke.

2. Background

The article, “Deep Learning Assists with Stroke Evaluation and Management” published by the Radiological Society of North America, explores a study done by Dr. Paul Yi of the University of Maryland School of Medicine regarding the use of deep learning methods to evaluate “the need for endovascular surgery” after a stroke. Using a dataset of 876 samples, Dr. Yi and his team tested both 2D and 3D CNNs to determine the presence of an LVO (large vessel occlusion) on a head CT angiography. The 2D CNN produced a high AUC (area under the curve) of 0.95, but used CT images that were highly preprocessed. The 3D CNN, on the other hand, produced a slightly lower AUC of 0.80, but utilized similar data to what an algorithm would see in a real life situation. Dr. Yi concluded that while his results are preliminary, further research could ensure the incorporation of AI into stroke diagnosis, particularly in emergency situations. Dr. Yi’s findings are essential to efficiency in future stroke treatment, especially with the amount of accessible radiologists going down. His team used image data of CT angiography, contrary to numerical patient data that our team used, and therefore used a CNN (convolutional neural network), instead of an ANN (artificial neural network). Since CNNs have previously shown high accuracy, in comparison to both ANNs and “human observers,” Dr. Yi’s data has set a reliable baseline for future research. While this team’s sample was balanced in that the amount of LOV positive and LOV negative samples were equal, the relatively low dataset size (876 samples), may make it harder to understand the full scope of the results provided. Dr. Yi’s team looked at ways to address stroke from the perspective of the treatment efficiency (in other words, after the medical event has taken place); we hope to build on their research by looking at stroke prediction through the lens of the prevention and developing a model that looks at comorbidities and the other demographic factors to determine stroke susceptibility.

3. Dataset

Our dataset consists of 5110 samples of numerical data, including 5110 patients and their demographic and medical information pertaining to stroke susceptibility. Factors provided include patient ID, gender, age, presence of hypertension and heart disease, marriage status, work type, residence type, average glucose level, BMI (body mass index), smoking status, and occurrence of stroke. The combination of both health and lifestyle factors within the dataset directly correlated with our research question that addressed the varying influence of medical and lifestyle circumstances on the risk of stroke. The data was split into a 80:20 train:test ratio using the train_test_split function from the scikit learn library, allowing for 4088 samples to be used to train the model and

1022 samples to be used to test the model. Insignificant factors such as patient ID (which had no impact on stroke susceptibility) were removed from the dataset and null values such as ‘NaN’ were replaced with ‘-1.0’ to prevent being factored in by the model. The Label_Encoder( ) function was utilized to perform linearization (conversion of non-numerical values such as ‘urban’ or ‘rural’ to numerical values such as ‘1’ and ‘0’). Close examination of the raw dataset showed a significant imbalance between stroke positive and stroke negative patients. Only 249 out of 5110 patients were stroke positive, while 4861 patients were stroke negative, giving a 1:20 stroke positive to stroke negative patient ratio. This discrepancy was resolved during the preprocessing stage through the duplication of stroke positive samples and removal of stroke negative samples within the training set (duplication and removal of certain data points did not occur within the testing set as to not alter the model’s final results). A set of 105 stroke positive samples were duplicated 10 times and added to the training set and 1844 stroke negative samples were removed from the training set in order to ensure that the number of stroke positive and stroke negative patients were equal (2044 samples of each category). The preprocessing measures noted above prepared the data to be fed into the model.

4. Methodology / Models

We developed a logistic regression model to establish a baseline for determining stroke outcomes. Logistic regression uses the combination of a basic linear function and a sigmoid function to determine the likelihood of a particular instance taking place, in our case the likelihood of a patient having a stroke based on data of other patients with particular demographic and medical conditions. When tested with the data, the logistic regression model showed a high accuracy score of 0.95, but an unusually low F1 score (calculated based on precision and recall values) of 0.0. The unexpectedly high accuracy was caused by the imbalance of stroke positive and stroke negative samples; the baseline models predicted all of the samples to be stroke negative, and since a large majority of the samples were stroke negative, the models were able to maintain a high accuracy. The decision tree classifier model, which looks at every possible classification outcome and makes a decision based on data from the training set, showed a slightly lower accuracy score of 0.89, but a slightly higher F1 score of 0.1. Other baseline models, including ridge classifier, random forest classifier, support vector classifier models, tested for comparison purposes, provided similar results to the logistic regression model.

Model Type	Accuracy Score	F1 Score
Logistic regression	0.952054794520548	0.0

Ridge classifier	0.952054794520548	0.0
Random forest classifier	0.9500978473581213	0.0
Support vector classifier	0.952054794520548	0.0
Decision tree classifier	0.8923679060665362	0.11290322580645161

Figure 1: Baseline models that were tested and their corresponding accuracy and F1 scores

For our final model, we chose to use a higher capacity multilayer perceptron (deep learning) model. An artificial neural network (ANN) is developed based on neural networks in the human brain. The model's 'input layer' is given the test samples without labels, after which it is passed through multiple 'hidden layers' that contribute to model efficiency, and an outcome is released through an 'output layer.' After the data was preprocessed (linearization, removal of null values, etc.), it was fed into our four layered artificial neural network (using the Keras library). We found that as the number of layers wavered higher or lower than 4, the The ReLU (rectified linear unit) activation function was used for all hidden layers and the sigmoid activation function was used for the final output layer. The Adam optimizer was implemented to optimize and modify the model's learning rate, which was set at 5e-2 (0.05). Binary cross entropy was used to calculate loss because the final output of the ANN is binary, stroke positive or stroke negative. Loss curve was plotted to determine the capacity of the model.

Figure 2: Loss curve of ANN model

Other hyperparameters, such as batch size, epoch value, and the operating point were delicately tuned to maximize model accuracy and efficiency. Batch size was finalized at 32 training samples and the epoch value was finalized at 100 epochs. The optimum operating point was found to be at 0.4; after testing values from as high as 0.8 to as low as 0.0001, the operating point that maximized both true positive samples and true negative samples was at 0.4. Further, the kernel regularizer was used to prevent overfitting in the model (particularly as a result of previous overfitting seen in the basic logistic regression and decision tree classifier models). Hyperparameters were tuned to prioritize a high F1 score, which is calculated based on a combination of precision and recall. Note that high accuracy may not always be accurate, as shown with the simpler models used as baselines, which is why F1 score was considered. The final predictions of the model were depicted through a confusion matrix, with 660 true negatives, 398 false negatives, 41 true positives, and 13 true negatives. Model results were analyzed through the use of precision, recall, and F1 score values.

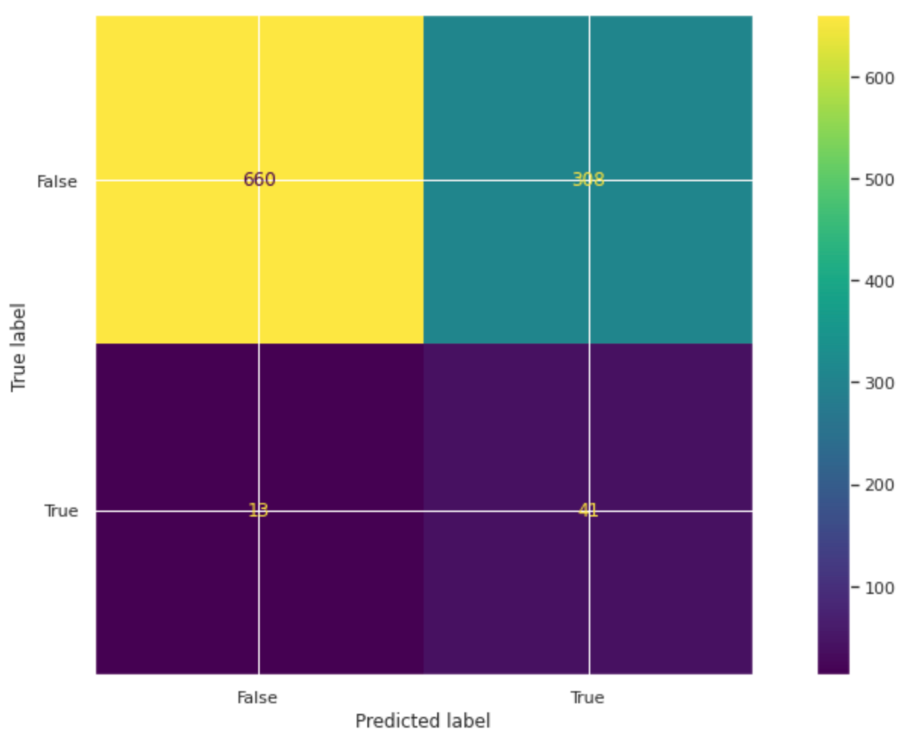

Figure 3: Confusion matrix of final model results

5. Results and Discussion

We developed a deep learning model (ANN) to predict stroke susceptibility in patients with varying medical, lifestyle and demographic conditions. The model outputs a decimal value describing the likelihood of a particular sample being stroke positive; depending on its relation to the operating point, it is classified as stroke negative or stroke positive. During the hyperparameter tuning process, it was observed that the operating point significantly impacted the accuracy of the model. While initially kept at 0.5, the operating point was lowered to 0.4 to accommodate the low number of stroke positive samples and maximize the accuracy of the model. Other hyperparameters, such as batch size, epoch value, and learning rate, were carefully tuned during the early development of the model and remained constant throughout the testing process. As mentioned earlier, metrics used to evaluate model capability include precision, recall, and the F1 score. Recall was prioritized over precision, as the number of correctly classified stroke positive samples out of the total number of stroke positive samples is highly relevant for model development and clinical use. F1 score, calculated using a combination of precision and recall (double the quotient of their product and sum). The model presented recall values of 0.68 for stroke negative samples and 0.76 for stroke positive samples. While recall values were maximized given the constraints of the relatively small dataset, further testing must be done using a larger, more balanced dataset to better understand the capability of the model to be used in clinical settings. Precision values were 0.98 and 0.12 and F1 scores were 0.80 and 0.20 for stroke negative and stroke positive samples (respectively). The low precision value, and therefore low F1 score for stroke positive samples was due to the imbalance of samples within the dataset; with only 54 stroke positive samples in the test set, the ratio of correctly classified samples to the number of stroke positive samples was highly disproportionate. While our model has produced useful preliminary results in the field of stroke prediction, more research is needed to better understand the model's capability to classify stroke positive cases, which will be most relevant in clinical scenarios, and the role deep learning will play in stroke prediction using non-image data.

	precision	recall	f1-score	support
0	0.98	0.68	0.80	968
1	0.12	0.76	0.20	54
accuracy			0.69	1022
macro avg	0.55	0.72	0.50	1022
weighted avg	0.94	0.69	0.77	1022

Figure 4: Classification report of model results

At the end of the model development process, a heatmap was developed to better understand the significance of various features on a patient's stroke susceptibility and compare the impact capacity of medical and non-medical factors. With a correlation value of +0.25, age was illustrated to have a strong positive correlation with stroke susceptibility, meaning that an increase in age accounted for an increase in stroke susceptibility. The features of hypertension, average glucose level, and heart disease depicted a +0.13 correlation value with the presence of stroke, while the correlation between stroke presence and marriage status was 0.11. This similarity indicates the need to further explore the role of relationship factors, along with medical comorbidities, in increasing stroke risk. While it can be concluded that medical factors have an increased impact on stroke susceptibility in comparison to non-medical factors, the magnitude of this variance in significance needs to be researched through further factor analysis (particularly of negative correlation values) and tested on a large dataset to show the consistency of results. The outcomes of our heatmap show that factors other medical comorbidities can impact the stroke susceptibility, bringing up an extension to our research problem: the role lifestyle and demographic factors play in other medical events, such as heart attacks, brain hemorrhages, and other potentially fatal conditions.

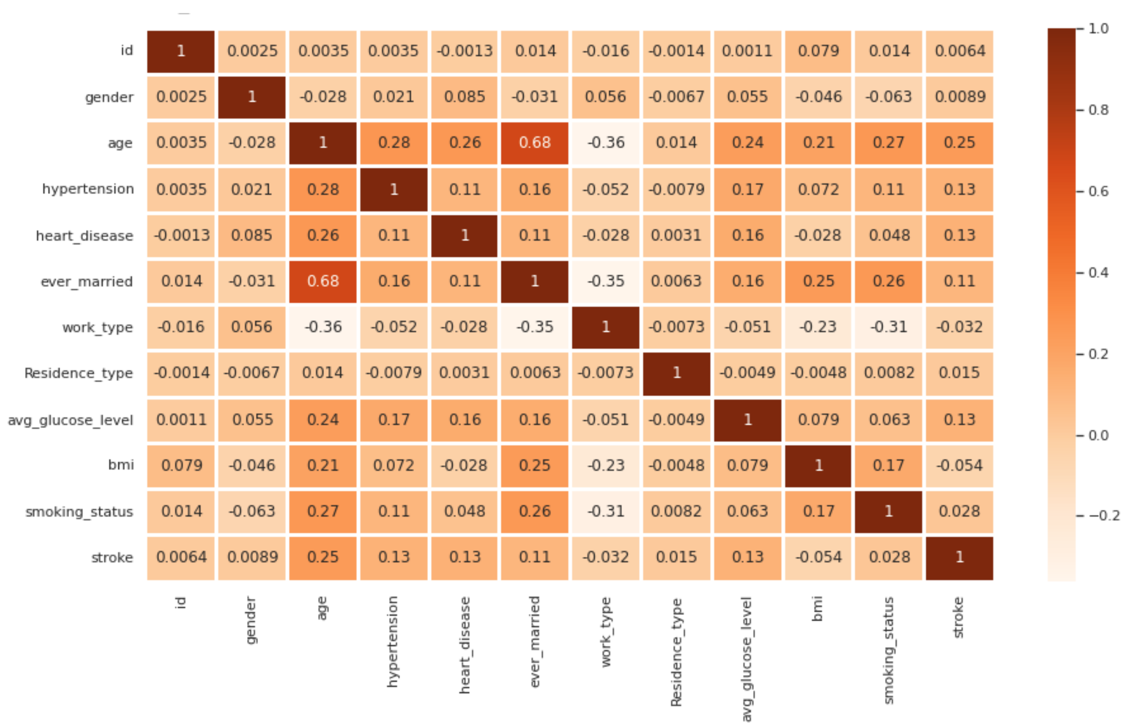

Figure 5: Heatmap of correlation between dataset features and stroke presence

6. Conclusions

Our research was aimed at developing a model to predict stroke susceptibility and determine what factors make a patient more or less likely to have a stroke in order to address stroke prevention, as Americans continue to be at high risk for strokes. After developing several baseline models (logistic regression and decision tree classifier), we chose a multilayer perceptron model (ANN) for stroke prediction as deep learning has previously been shown to produce accurate and highly reliable results in binary classification problems. Our model was a four layered ANN; hyperparameters such as learning rate, batch size, epoch value, and operating threshold were tuned at 5e-2, 32, 100, and 0.4 respectively, to maximize recall and F1 score. The model produced a recall of 0.68 and F1 score of 0.80 for stroke negative samples and a recall of 0.76 and F1 score of 0.20 for stroke positive samples. While these values are higher than F1 scores produced by the baseline models, further research must be done to improve the accuracy of the model. In the future, utilizing a larger dataset (with >100,00) samples) with a greater balance between stroke positive and stroke negative samples, is recommended for the model to produce a higher accuracy. The heat map produced along with the model showed that the factor with the largest impact on stroke susceptibility was age; medical factors such as average glucose level and presence of hypertension and heart disease proved to have a larger impact than demographic factors such as marriage status, residence status, or work status. Further research should be done on larger datasets to gain a deeper understanding of the extent of impact of particular factors on stroke risk.

Acknowledgments

I would like to thank my mentor Samuel Kwong for his valuable inputs and support through the cultivation of my research project, as well as Inspirit AI for giving me the opportunity, resources, and space to take part in this enriching experience.

References

1. *3.3. metrics and scoring: Quantifying the quality of predictions*. scikit. Retrieved January 9, 2023, from https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics
2. *Activation functions in neural networks [12 types & use cases]*. V7. Retrieved January 9, 2023, from https://www.v7labs.com/blog/neural-networks-activation-functions#:~:text=drive%20V7's%20tools.-,What%20is%20a%20Neural%20Network%20Activation%20Function%3F,prediction%20using%20simpler%20mathematical%20operations.

3. Agarwal, M. (2022, January 24). *A deep dive into factor analysis*. Medium. Retrieved January 9, 2023, from https://medium.com/mlearning-ai/a-deep-dive-into-factor-analysis-d64e550c358f
4. Alake, R. (2020, May 8). *Regularization techniques and their implementation in tensorflow (keras)*. Medium. Retrieved January 10, 2023, from https://towardsdatascience.com/regularization-techniques-and-their-implementation-in-tensorflow-keras-c06e7551e709#:~:text=To%20add%20a%20regularizer%20to,represents%20the%20regularization%20hyperparameter%20value.
5. Arti. (2022, March 8). *Top 10 deep learning models that will help make advanced AI*. Analytics Insight. Retrieved January 10, 2023, from https://www.analyticsinsight.net/top-10-deep-learning-models-that-will-help-make-advanced-ai/
6. *Auto adjust font size in seaborn heatmap*. Stack Overflow. (1962, October 1). Retrieved January 10, 2023, from https://stackoverflow.com/questions/33104322/auto-adjust-font-size-in-seaborn-heatmap
7. Biswal, A. (2022, November 30). *Recurrent neural network (RNN) tutorial: Types and examples [updated]: Simplilearn*. Simplilearn.com. Retrieved January 10, 2023, from https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn
8. Bonthu, H. (2021, July 11). *An introduction to logistic regression*. Analytics Vidhya. Retrieved January 9, 2023, from https://www.analyticsvidhya.com/blog/2021/07/an-introduction-to-logistic-regression/
9. Brownlee, J. (2020, August 20). *A gentle introduction to the rectified linear unit (ReLU)*. MachineLearningMastery.com. Retrieved January 9, 2023, from https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/
10. Brownlee, J. (2022, August 16). *Your first deep learning project in python with keras step-by-step*. MachineLearningMastery.com. Retrieved January 10, 2023, from https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/
11. *Build your first artificial neural network with python from scratch*. MLTut. (2022, December 16). Retrieved January 10, 2023, from https://www.mltut.com/implementation-of-artificial-neural-network-in-python/
12. *Classifier comparison*. Scikit. Retrieved January 10, 2023, from https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html
13. *Confusion matrix in machine learning*. Confusion Matrix in Machine Learning | Machine Learning. Retrieved January 10, 2023, from

https://python-course.eu/machine-learning/confusion-matrix-in-machine-learning.php

14. *Create a confusion matrix for neural network predictions*. DeepLizard. Retrieved January 10, 2023, from https://deeplizard.com/learn/video/km7pxKy4UHU
15. Data to Fish. Retrieved January 10, 2023, from https://datatofish.com/numpy-array-to-pandas-dataframe/
16. *Decision tree algorithm, explained*. KDnuggets. Retrieved January 9, 2023, from https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html
17. *Deep learning assists with Stroke Evaluation and Management*. RSNA. Retrieved January 9, 2023, from https://www.rsna.org/news/2022/april/AI-For-Stroke-Evaluation
18. *Deep Learning Networks: 7 awesome types of deep learning networks*. EDUCBA. (2021, November 15). Retrieved January 10, 2023, from https://www.educba.com/deep-learning-networks/
19. *Deep Learning vs. machine learning – what's the difference?* Levity. Retrieved January 9, 2023, from https://levity.ai/blog/difference-machine-learning-deep-learning#:~:text=Machine%20Learning%20means%20computers%20learning,documents%2C%20images%2C%20and%20text.
20. Dixit, S. (2021, April 16). *Building your own artificial neural network from scratch*. Medium. Retrieved January 10, 2023, from https://towardsdatascience.com/building-your-own-artificial-neural-network-from-scratch-on-churn-modeling-dataset-using-keras-in-690782f7d051
21. Fedele, S. (1963, May 1). *How to replace an entire column on pandas.dataframe*. Stack Overflow. Retrieved January 9, 2023, from https://stackoverflow.com/questions/36846060/how-to-replace-an-entire-column-on-pandas-dataframe
22. Gad, A. F. (2021, April 9). *Accuracy, precision, and recall in deep learning*. Paperspace Blog. Retrieved January 9, 2023, from https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/#:~:text=The%20recall%20is%20calculated%20as,the%20more%20positive%20samples%20detected.
23. Google. *Descending into ML: Training and loss | machine learning | google developers*. Google. Retrieved January 9, 2023, from https://developers.google.com/machine-learning/crash-course/descending-into-ml/training-and-loss#:~:text=Loss%20is%20the%20penalty%20for,otherwise%2C%20the%20loss%20is%20greater.
24. Gupta, A. (2022, December 2). *A comprehensive guide on Deep learning optimizers*. Analytics Vidhya. Retrieved January 9, 2023, from https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-l

earning-optimizers/#:~:text=An%20optimizer%20is%20a%20function,loss%20and%20improve%20the%20accuracy.
25. Gupta, P. (2017, November 16). *Regularization in machine learning*. Medium. Retrieved January 9, 2023, from https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a
26. Gupta, R. (2019, October 11). *6 deep learning models - when should you use them?* Medium. Retrieved January 10, 2023, from https://towardsdatascience.com/6-deep-learning-models-10d20afec175
27. Holtz, Y. *Heatmap*. The Python Graph Gallery. Retrieved January 10, 2023, from https://www.python-graph-gallery.com/heatmap/
28. Holtz, Y. *Heatmaps with various input formats*. The Python Graph Gallery. Retrieved January 10, 2023, from https://www.python-graph-gallery.com/90-heatmaps-with-various-input-format/
29. Hovden, I. T. (1970, January 1). *Optimizing artificial neural network hyperparameters and architecture*. [PDF] Optimizing Artificial Neural Network Hyperparameters and Architecture | Semantic Scholar. Retrieved January 9, 2023, from https://www.semanticscholar.org/paper/Optimizing-Artificial-Neural-Network-and-Hovden/743cccba1436e29131bb93cd0a462c26a0a689b5
30. *How do I convert a pandas dataframe to a PYTORCH tensor?* Stack Overflow. (1965, May 1). Retrieved January 10, 2023, from https://stackoverflow.com/questions/50307707/how-do-i-convert-a-pandas-dataframe-to-a-pytorch-tensor
31. *How do I determine the best predictor in a linear regression model?* Cross Validated. (1963, October 1). Retrieved January 10, 2023, from https://stats.stackexchange.com/questions/238501/how-do-i-determine-the-best-predictor-in-a-linear-regression-model
32. *How to convert pandas dataframe to numpy array - with examples*. Stack Vidhya. (2022, December 13). Retrieved January 10, 2023, from https://www.stackvidhya.com/how-to-convert-pandas-dataframe-to-numpy-array/
33. *How to count occurrences of specific value in pandas column?* GeeksforGeeks. (2021, December 23). Retrieved January 10, 2023, from https://www.geeksforgeeks.org/how-to-count-occurrences-of-specific-value-in-pandas-column/#:~:text=We%20can%20count%20by%20using,values%20in%20a%20particular%20column
34. *How to easily create heatmaps in python*. Statology. (2021, July 16). Retrieved January 10, 2023, from https://www.statology.org/heatmap-python/

35. *How to exclude columns in pandas?* GeeksforGeeks. (2022, September 15). Retrieved January 10, 2023, from https://www.geeksforgeeks.org/how-to-exclude-columns-in-pandas/
36. *How to interpret negative correlations on a heatmap (or in general)?* Stack Overflow. (1968, January 1). Retrieved January 10, 2023, from https://stackoverflow.com/questions/65657627/how-to-interpret-negative-correlations-on-a-heatmap-or-in-general
37. *How to make predictions using Keras Model?* ProjectPro. Retrieved January 10, 2023, from https://www.projectpro.io/recipes/make-predictions-keras-model
38. *How to select rows by index in a pandas DataFrame*. Statology. (2021, November 9). Retrieved January 10, 2023, from https://www.statology.org/pandas-select-rows-by-index/
39. *Indexing and selecting data with pandas*. GeeksforGeeks. (2022, April 13). Retrieved January 10, 2023, from https://www.geeksforgeeks.org/indexing-and-selecting-data-with-pandas/
40. Jose, G. V. (2019, October 3). *Useful plots to diagnose your neural network*. Medium. Retrieved January 10, 2023, from https://towardsdatascience.com/useful-plots-to-diagnose-your-neural-network-521907fa2f45
41. *Keras - plot training, validation and test set accuracy*. Stack Overflow. (1964, February 1). Retrieved January 10, 2023, from https://stackoverflow.com/questions/41908379/keras-plot-training-validation-and-test-set-accuracy
42. *Keras Model Plot Loss Code Examples & Solutions for this technical problem cluster*. Code Examples & Solutions For This Technical Problem Cluster. Retrieved January 10, 2023, from https://www.codegrepper.com/code-examples/python/plot+to+see+the+accuracy+and+loss+keras
43. Kumar, A. (2022, April 16). *Correlation Concepts, Matrix & Heatmap using Seaborn*. Data Analytics. Retrieved January 9, 2023, from https://vitalflux.com/correlation-heatmap-with-seaborn-pandas/#:~:text=The%20values%20in%20the%20cells,the%20strength%20of%20these%20relationships.
44. *Load a pandas dataframe*. TensorFlow. Retrieved January 10, 2023, from https://www.tensorflow.org/tutorials/load_data/pandas_dataframe
45. *Machine learning - Confusion Matrix*. Python Machine Learning - Confusion Matrix. Retrieved January 10, 2023, from https://www.w3schools.com/python/python_ml_confusion_matrix.asp
46. Madhavan, S., & Sturdevant, M. *https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html*. IBM developer. Retrieved January 10, 2023, from

https://developer.ibm.com/tutorials/learn-classification-algorithms-using-python-and-scikit-learn/

47. Mayo Foundation for Medical Education and Research. (2022, January 20). *Stroke*. Mayo Clinic. Retrieved January 9, 2023, from https://www.mayoclinic.org/diseases-conditions/stroke/symptoms-causes/syc-20350113
48. Meel, V. (2022, June 7). *Ann and CNN: Analyzing differences and similarities*. viso.ai. Retrieved January 9, 2023, from https://viso.ai/deep-learning/ann-and-cnn-analyzing-differences-and-similarities/#:~:text=In%20general%2C%20CNN%20tends%20to,image%20inputs%20are%20not%20necessary.
49. Mohadarkar, S. (2022, July 26). *Implementing Artificial Neural Network in Python from Scratch*. Analytics Vidhya. Retrieved January 10, 2023, from https://www.analyticsvidhya.com/blog/2021/10/implementing-artificial-neural-networkclassification-in-python-from-scratch/
50. Myrianthous, G. (2021, April 17). *How to split a dataset into training and testing sets with python*. Medium. Retrieved January 10, 2023, from https://towardsdatascience.com/how-to-split-a-dataset-into-training-and-testing-sets-b146b1649830
51. Narkhede, S. (2021, June 15). *Understanding AUC - ROC Curve*. Medium. Retrieved January 10, 2023, from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
52. Nelson, D. (2022, July 21). *Overview of classification methods in python with scikit-learn*. Stack Abuse. Retrieved January 10, 2023, from https://stackabuse.com/overview-of-classification-methods-in-python-with-scikit-learn/
53. *Neurosciences - Stroke Facts*. UTMBHealthCOM. Retrieved January 9, 2023, from https://www.utmbhealth.com/services/neurology/procedures-conditions/stroke/stroke-facts#:~:text=Stroke%20is%20the%20fifth%20leading,cause%20of%20death%20for%20women.
54. Nicholls, J. K., Ince, J., Minhas, J. S., & Chung, E. M. L. (2021, December 13). *Emerging detection techniques for large vessel occlusion stroke: A scoping review*. Frontiers. Retrieved January 9, 2023, from https://www.frontiersin.org/articles/10.3389/fneur.2021.780324/full
55. *Pandas - find specific value in entire dataframe*. Stack Overflow. (1965, December 1). Retrieved January 10, 2023, from https://stackoverflow.com/questions/53587315/pandas-find-specific-value-in-entire-dataframe

56. Prasad, A. (2020, September 13). *Pytorch for deep learning - binary classification (logistic regression)*. Medium. Retrieved January 10, 2023, from https://medium.com/analytics-vidhya/pytorch-for-deep-learning-binary-classification-logistic-regression-382abd97fb43
57. Prosise, J. (2022, October 23). *Binary classification with Neural Networks*. Atmosera. Retrieved January 9, 2023, from https://www.atmosera.com/blog/binary-classification-with-neural-networks/
58. Rai, S. (2021, August 26). *Traditional vs. deep learning classification models*. Analytics Vidhya. Retrieved January 10, 2023, from https://www.analyticsvidhya.com/blog/2021/05/emperical-study-of-traditional-vs-deep-learning-classification-models/
59. Real Python. (2021, February 2). *Pandas sort: Your guide to sorting data in python*. Real Python. Retrieved January 10, 2023, from https://realpython.com/pandas-sort-python/
60. Rendyk. (2022, December 2). *Tuning the hyperparameters and layers of neural network deep learning*. Analytics Vidhya. Retrieved January 9, 2023, from https://www.analyticsvidhya.com/blog/2021/05/tuning-the-hyperparameters-and-layers-of-neural-network-deep-learning/#:~:text=The%20hyperparameters%20to%20tune%20are,tune%20the%20number%20of%20layers.
61. Roy, A. (2020, November 6). *A dive into decision trees*. Medium. Retrieved January 9, 2023, from https://towardsdatascience.com/a-dive-into-decision-trees-a128923c9298
62. Saxena, S. (2018, May 13). *Precision vs recall*. Medium. Retrieved January 9, 2023, from https://medium.com/@shrutisaxena0617/precision-vs-recall-386cf9f89488
63. Saxena, S. (2022, December 5). *Binary cross entropy/log loss for binary classification*. Analytics Vidhya. Retrieved January 9, 2023, from https://www.analyticsvidhya.com/blog/2021/03/binary-cross-entropy-log-loss-for-binary-classification/#:~:text=Binary%20cross%20entropy%20compares%20each,far%20from%20the%20actual%20value.
64. *Seaborn How To's*. Delft Stack. Retrieved January 10, 2023, from https://www.delftstack.com/howto/seaborn/
65. Singh, H. (2021, March 12). *Neural network: Introduction to neural network: Neural network for DL*. Analytics Vidhya. Retrieved January 9, 2023, from https://www.analyticsvidhya.com/blog/2021/03/basics-of-neural-network/
66. *Sklearn.metrics.f1_score*. scikit. Retrieved January 9, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html#sklearn.metrics.f1_score

67. *Sklearn.model_selection.train_test_split*. Scikit. Retrieved January 10, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
68. *Supervised vs. unsupervised learning; which is best?* Alteryx. Retrieved January 9, 2023, from https://www.alteryx.com/glossary/supervised-vs-unsupervised-learning#:~:text=Supervised%20and%20unsupervised%20learning%20have,tagged%20with%20the%20right%20answer.&text=A%20classification%20problem%20uses%20algorithms%20to%20classify%20data%20into%20particular%20segments.
69. *Top 10 deep learning models that will help make advanced AI*. Analytics Insight. (2022, March 8). Retrieved January 10, 2023, from https://www.analyticsinsight.net/top-10-deep-learning-models-that-will-help-make-advanced-ai/
70. V, K. (2021, April 14). *Artificial Neural Network: How does artificial neural network work*. Analytics Vidhya. Retrieved January 9, 2023, from https://www.analyticsvidhya.com/blog/2021/04/artificial-neural-network-its-inspiration-and-the-working-mechanism/
71. Viswarupan, N. (2021, November 24). *Drawing loss curves for deep neural network training in Pytorch*. Medium. Retrieved January 10, 2023, from https://niruhan.medium.com/drawing-loss-curves-for-deep-neural-network-training-in-pytorch-ac617b24c388
72. Zach. (2021, November 9). *How to select rows by index in a pandas dataframe*. Statology. Retrieved January 10, 2023, from https://www.statology.org/pandas-select-rows-by-index/