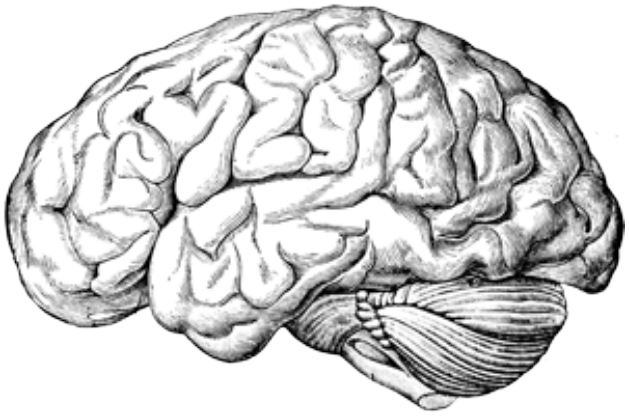
Stroke Prediction



Caleb Riese

May 2021

Contents

1 Abstract

This project was a culmination of all the things we've learned this semester. We combined different types of neural networks and other artificial intelligence methods to try and predict something. My topic was Stroke prediction, which takes many different input labels and tries to determine if that patient is prone to having a stroke or not. During this semester long project I tried many different methods to improve my final accuracy for the predictions. Some of the techniques we used in this course are neural networks, callbacks, feature removal, normalization, and much more.

2 Why I Chose This Topic

Strokes are one of the leading causes of death in the United States, with more than 800,000 people having strokes every year. Being able to predict the chance of a stroke would be very beneficial in reducing the number of deaths. I found this project very interesting since this is a type of problem actual AI researches are doing.

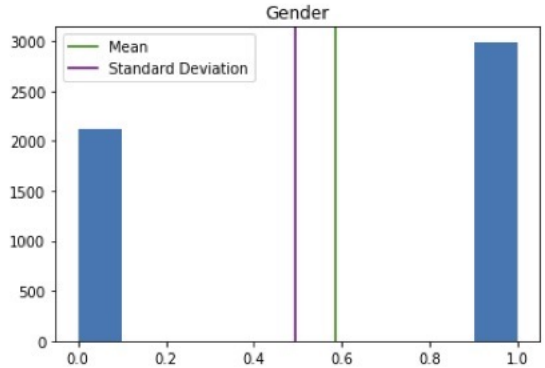
3 The Dataset

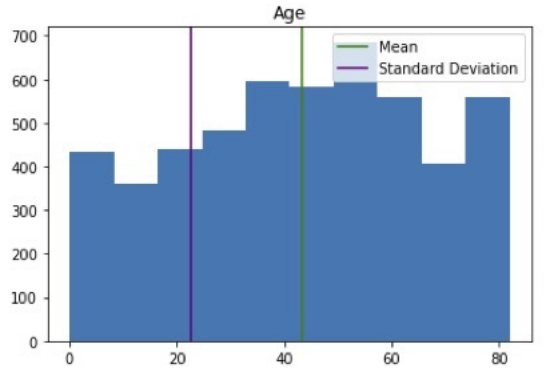
I found the dataset on Kaggle, it's called "Stroke Prediction Dataset". It has over five thousand rows of data to use for prediction. After going through all the phases in this project I've come to the conclusion that this is not the best dataset for a beginner AI project. It is heavily imbalanced and even after trying multiple ways of normalizing it I found the results to be pretty bad. However I think choosing this imbalanced set forced me to learn a lot of the techniques better since it required more effort.

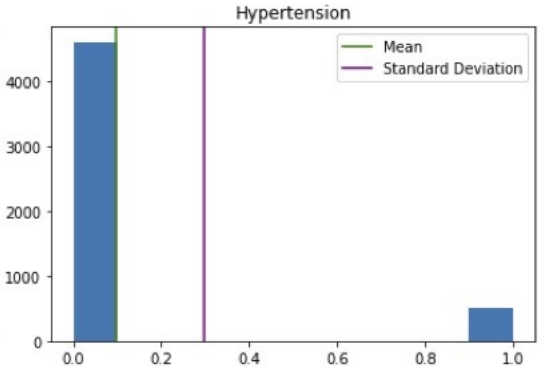
4 Input Features

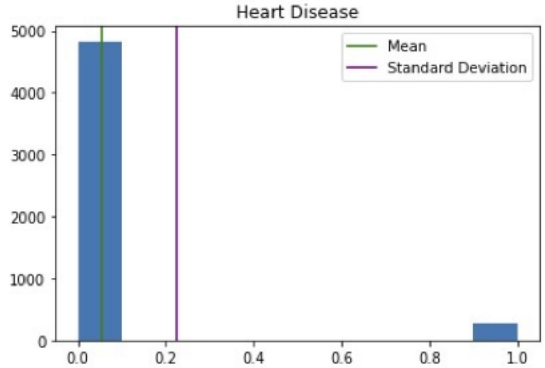
There are 11 input features for this dataset, they are

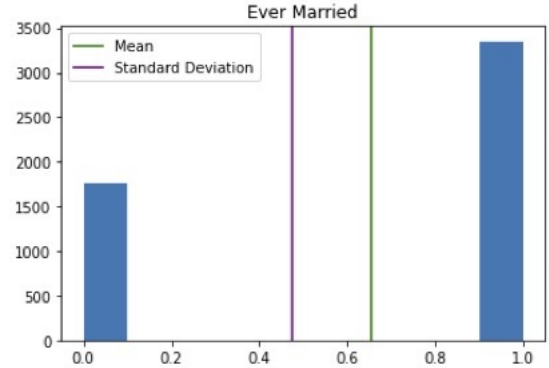
1. Gender: Male or Female
2. Age: age of the patient
3. Hypertension: patient doesn't have hypertension, patient has hypertension
4. Heart Disease: patient doesn't have any heart diseases, patient does have a heart disease
5. Ever Married: No or Yes
6. Employed: employed or unemployed
7. Residence Type: Rural or Urban
8. Average Glucose Level: average glucose level in blood
9. BMI: body mass index
10. Smoking Status: never smoked, formerly smoked, or smokes
11. Stroke: patient had a stroke or patient didn't have a stroke

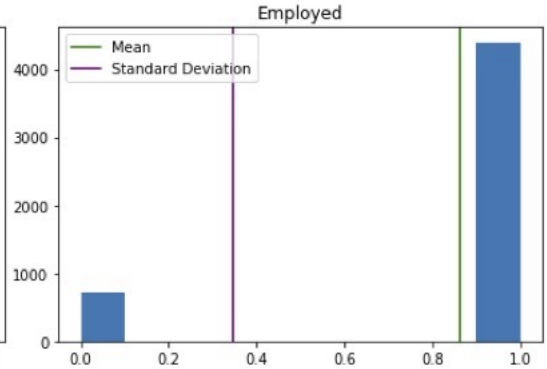

Gender
Mean
Standard Deviation

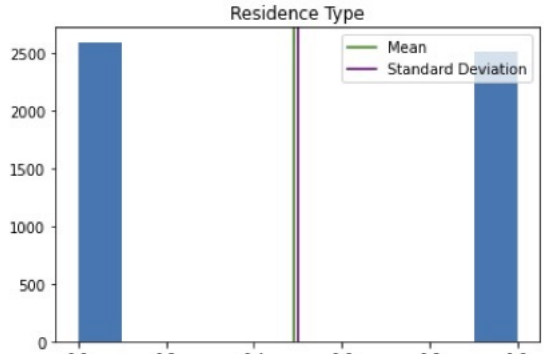

Age
Mean
Standard Deviation

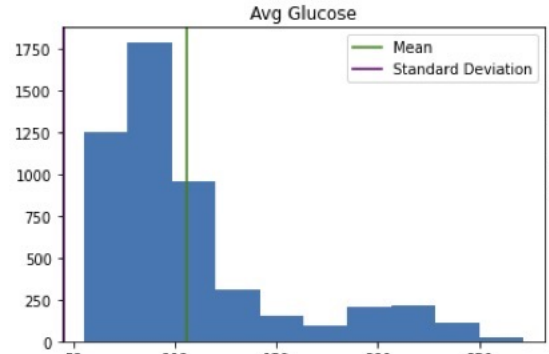

Hypertension
Mean
Standard Deviation

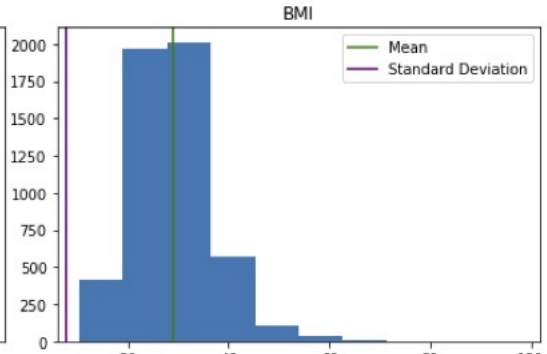

Heart Disease
Mean
Standard Deviation

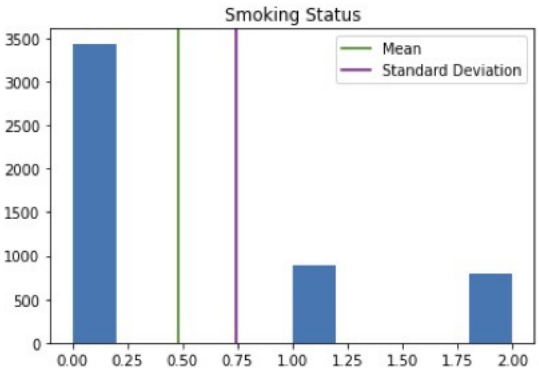

Ever Married
Mean
Standard Deviation

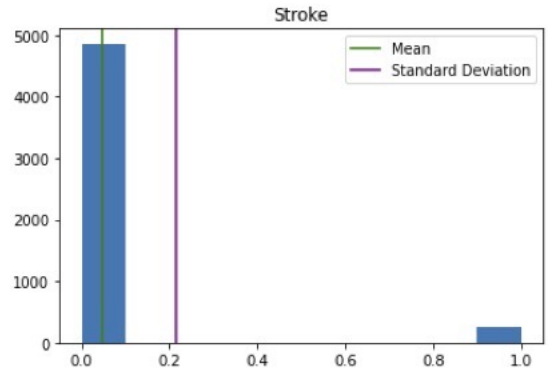

Employed
Mean
Standard Deviation


Residence Type
Mean
Standard Deviation


Avg Glucose
Mean
Standard Deviation


BMI
Mean
Standard Deviation


Smoking Status
Mean
Standard Deviation


Stroke
Mean
Standard Deviation

	gender	age	hypertension	heartDisease	everMarried	employed	residenceType	avgGlucoseLevel	bmi	smokingStatus	stroke
count	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000
mean	0.585910	43.226614	0.097456	0.054012	0.656164	0.861252	0.491977	106.147677	28.893110	0.481996	0.048728
std	0.492612	22.612647	0.296607	0.226063	0.475034	0.345717	0.499985	45.283560	7.698018	0.747390	0.215320
min	0.000000	0.080000	0.000000	0.000000	0.000000	0.000000	0.000000	55.120000	10.300000	0.000000	0.000000
25%	0.000000	25.000000	0.000000	0.000000	0.000000	1.000000	0.000000	77.245000	23.800000	0.000000	0.000000
50%	1.000000	45.000000	0.000000	0.000000	1.000000	1.000000	0.000000	91.885000	28.400000	0.000000	0.000000
75%	1.000000	61.000000	0.000000	0.000000	1.000000	1.000000	1.000000	114.090000	32.800000	1.000000	0.000000
max	1.000000	82.000000	1.000000	1.000000	1.000000	1.000000	1.000000	271.740000	97.600000	2.000000	1.000000

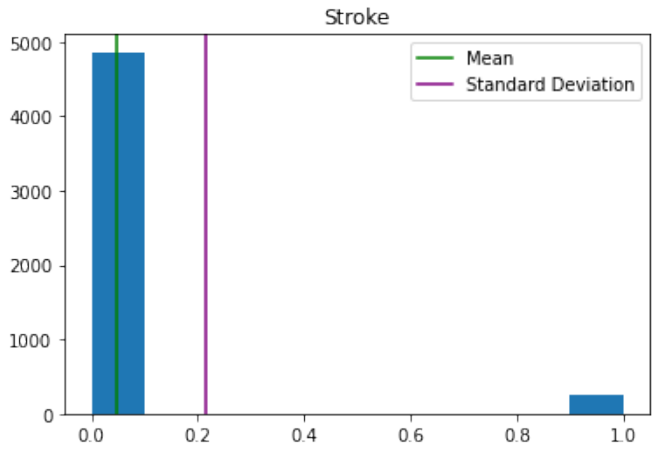
5 Cleaning The Dataset

Cleaning the dataset I had to clean the dataset for five columns: Gender, Ever Married, Employed, Residence Type, and Smoking Status. The corresponding keys are:

1. True = 1
2. False = 0
3. Male = 0
4. Female = 1
5. Urban = 0
6. Rural = 1
7. Never Smoked = 0
8. Formerly Smoked = 1
9. Smokes = 2

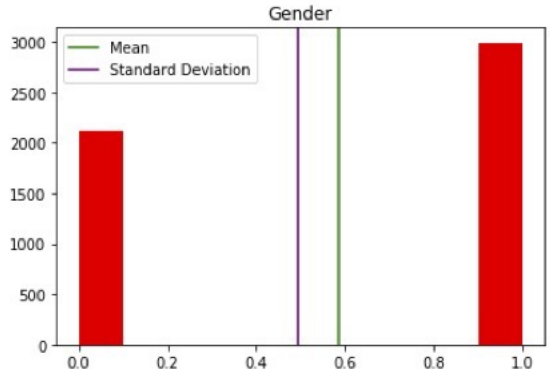
6 Distribution Of The Output Labels

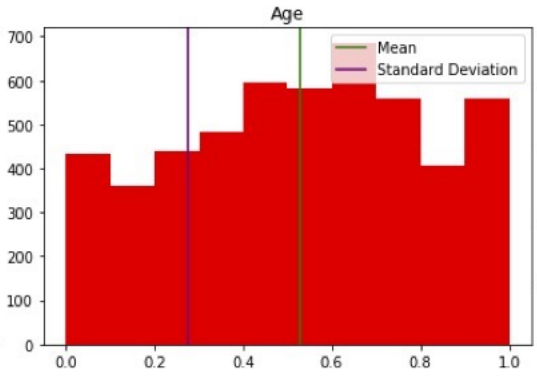
The distribution of the output labels isn't the best, about 3% are strokes and 97% of them are not strokes. This can cause problems for the predictions but it can be accounted for. I was able to improve on this a little bit but overall the imbalance affected the predictions a lot.

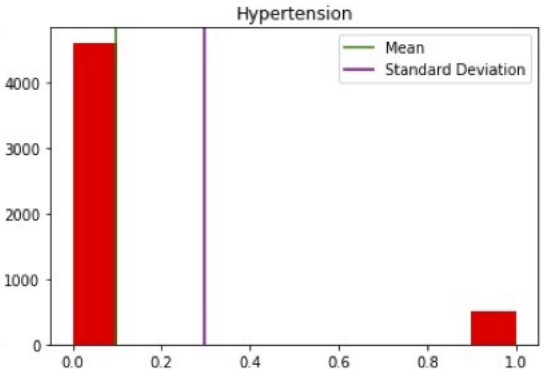


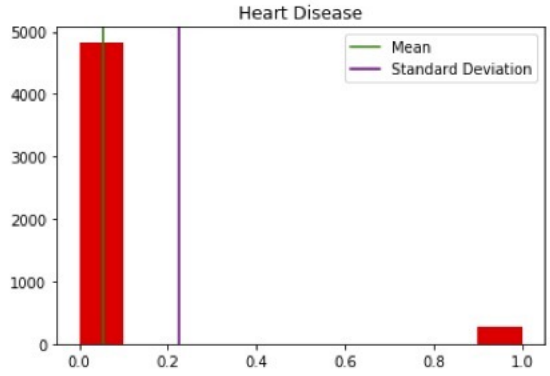
7 How I Normalized The Data

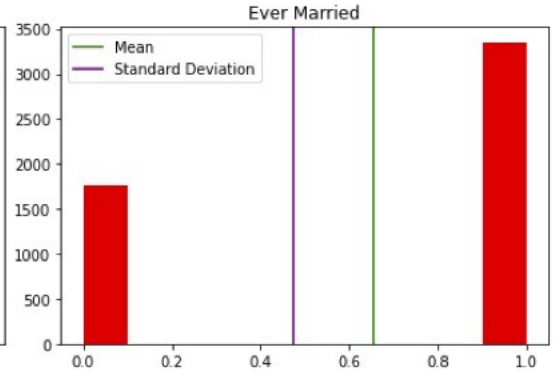
I first normalized my data by using the rescaling method, this is where you average the input feature by its maximum value. I had to do this on four of the input features: Age, Glucose Level, BMI, and Smoking Status. I also attempted the standardization method and Min-Max method but didn't find them to produce any better results.

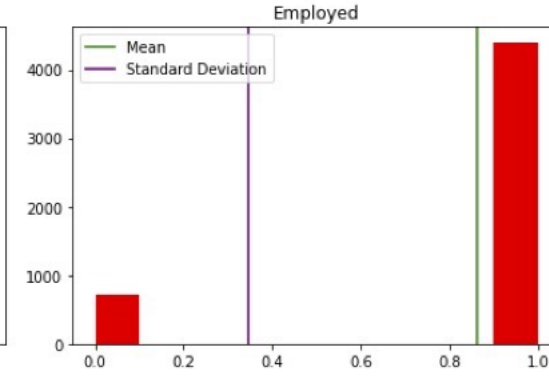

Gender
Mean
Standard Deviation

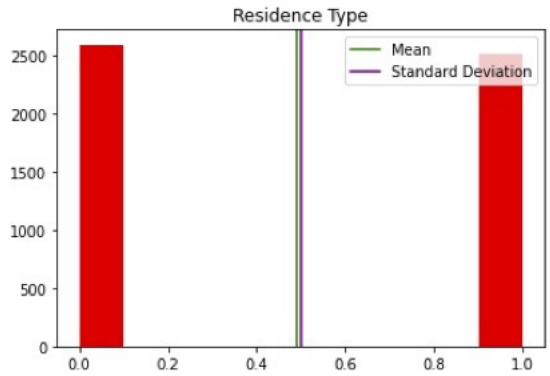

Age
Mean
Standard Deviation

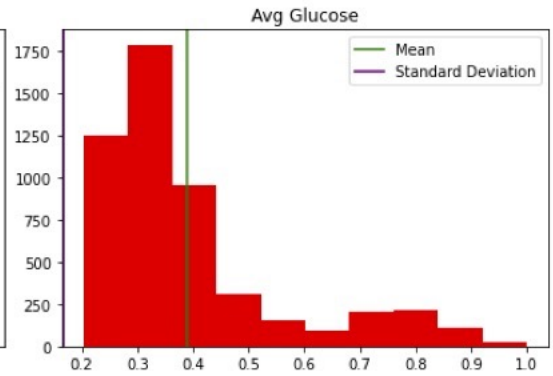

Hypertension
Mean
Standard Deviation

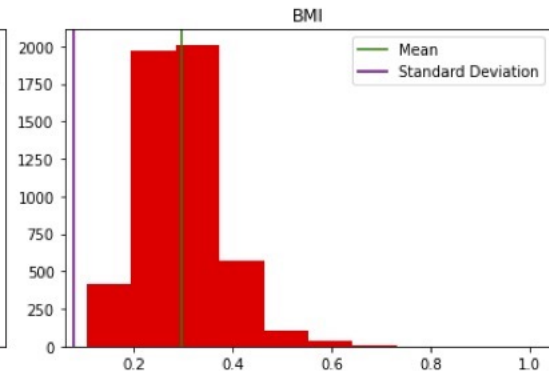

Heart Disease
Mean
Standard Deviation

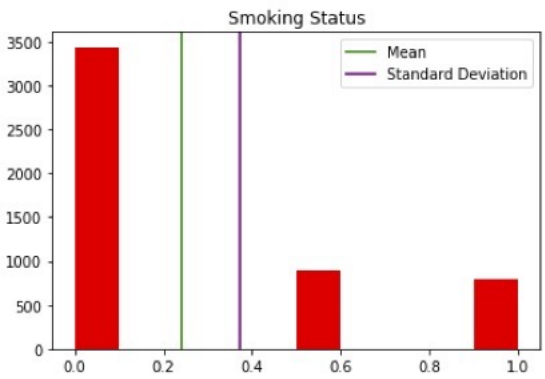

Ever Married
Mean
Standard Deviation

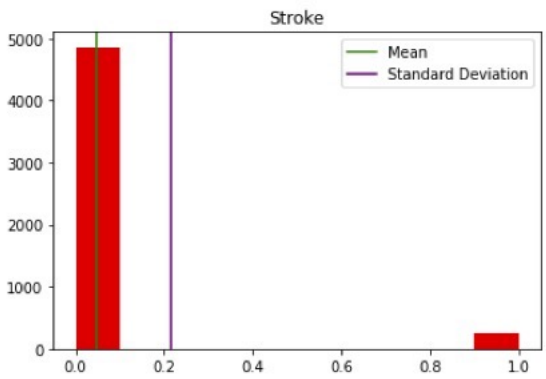

Employed
Mean
Standard Deviation


Residence Type
Mean
Standard Deviation


Avg Glucose
Mean
Standard Deviation


BMI
Mean
Standard Deviation


Smoking Status
Mean
Standard Deviation


Stroke
Mean
Standard Deviation

8 Data

I split the data into the Validation and the Training sets randomly, by shuffling the data before splitting it. With the training set consisting of 70% and the validation set only consisting of 30%. This is what we did in the example and I found it worked pretty well with those ratios. I also testing 80/20 and 90/10 but found 70/30 to produce the best results.

9 My Model

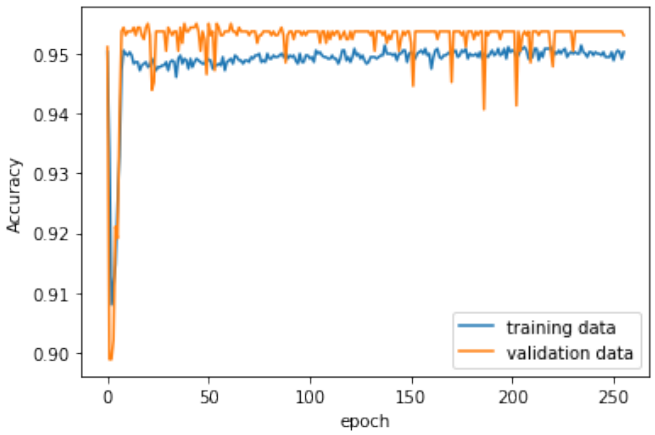
The models in this course are using what is called "feed forward" networks. I experimented with different layers and found having a single hidden layer worked the best.

The baseline model is a logistic regression model since I have a binary classification problem. For the baseline I did 512 epochs and the resulting accuracy was about 95% and the loss was 15%.

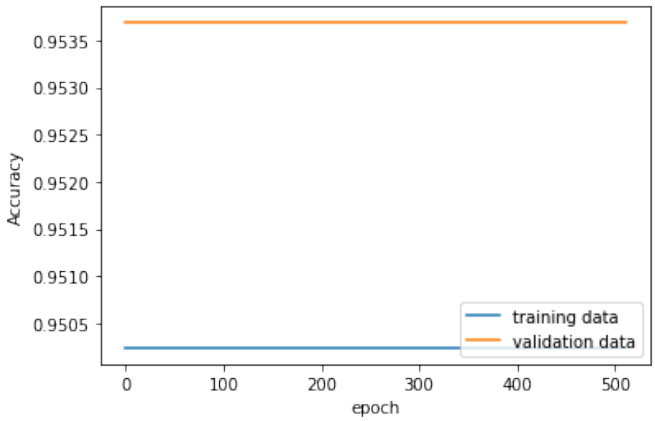
For the neural network model, accuracy and loss was very similar staying at about 95% for the accuracy and 15% for the loss. I kept the epochs at 256 and the accuracy didn't seem to increase with more epochs.

Model	Activation	Layers	Neurons	Accuracy	Loss
1	Sigmoid	2	5	95.13	15.31
2	Sigmoid/Relu	3	7	95.39	15.64
3	Sigmoid/Relu	4	9	95.23	15.01
4	Sigmoid/Relu	5	11	95.13	15.22
5	Sigmoid/Relu	6	13	95.33	14.71
6	Linear	3	13	94.65	81

10 Learning Curves



You can see that overall the baseline model didn't increase with further training.



I'm not sure if something went wrong with my learning curve on the hidden layer model, but it seemed that my accuracy was constant for the entire training.

11 Overfitting

I found that the data became overfitted very quick. With only 256 epochs the accuracy seemed to stay consistent throughout all of them, with it trending downward towards the end.

12 Callbacks

The callbacks were very helpful with the early stopping and other features. They helped me evaluate my models better and choose the hyperparameters that were best for my model.

13 Predictions

Overall the predictions were very accurate with an average of 95%. Even after testing different types of models with different layers and hyperparameters, I couldn't seem to get past this. I think this is because my data is very heavily imbalanced with 97% being not strokes.

14 Feature Importance and Reduction

One of the main parts of Phase three is investigating which features are the most significant to the model and trying to reduce the non-informative features. We do this by iteratively removing input features to discover which are most effective. Here are my results after doing this.

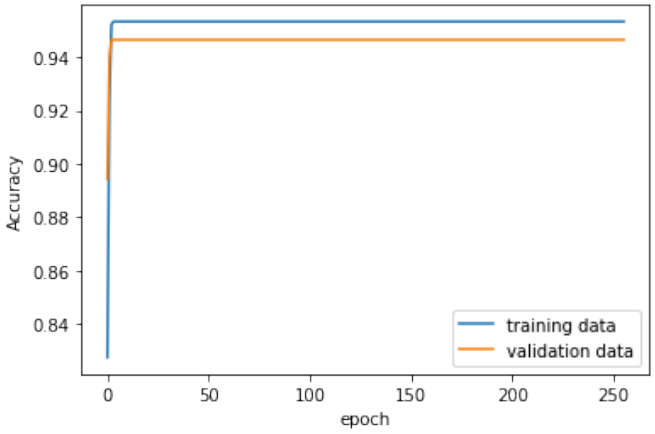
15 Individual Feature Importance

I have ten input features for my data set. I chose to study all ten input features to find out which were the most important. These features are gender, age, heart disease, Body Mass Index(BMI), smoking status, marriage status, employment status, residence type, hypertension, and glucose level.

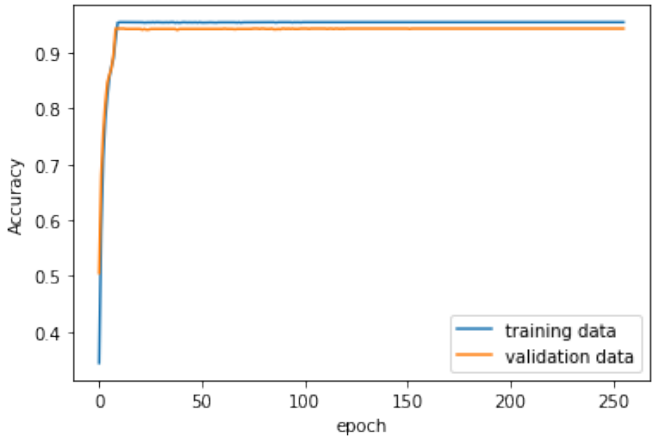
Input Feature	Accuracy
Gender	94.82
Age	95.09
Hypertension	95.34
Heart Disease	94.98
Married	94.72
Employed	95.56
Residence	94.94
Glucose Level	95.01
BMI	95.26
Smoking Status	95.42

16 Removing Non-Informative Features

After discovering which features were most and least informative I removed the features that had the lowest accuracy scores. These features were Gender, Heart Disease, Marriage, Residence, and Glucose Level.Then I compared this "feature reduced model" with my original model including all of the features.



Original



Reduced

The original model with all input features had an accuracy of 94.65%. My feature reduced model had an accuracy of 94.32%. These numbers are still incredibly close, so overall I'm thinking that each input feature isn't that helpful in deciding the accuracy of the model.

17 Problems I Had

Trying to find the non-informative input features took a long time. I tried many different accuracy methods but kept getting the same result for all input features. Finally I found a method that got different results for each feature, however they are still very similar. I think since my data is so heavily imbalanced each input feature is getting the same accuracy after many epochs.

18 Conclusion

My conclusion for this project is that my dataset was bad to start with but with a lot of this strategies such as removing non-informative features and choosing different hyperparameters I got better results. I would have liked to further increase the accuracy of my models but I'm happy with what I achieved. I found this project very educational and really enjoyed applying the lessons we learned in the lectures. My knowledge on artificial intelligence improved a lot this semester and I hope to keep learning more about it in the future.

19 References

1. My Notebook

- https://colab.research.google.com/drive/1jh2u3xFJti1QvW2UVvv$_h dEdb_j dvrrs?usp = sharing$

2. My Overleaf

    - https://www.overleaf.com/read/cfvdkbwtfgcy

3. My Github

    - https://github.com/calebriese/CS4300-Project

4. Stroke Prediction Dataset

    - https://www.kaggle.com/fedesoriano/stroke-prediction-dataset