Scott Breitbach
20-July-2022

Stroke Prediction

White Paper

Business Problem:

Stroke prediction using individual patient data.

Background / History:

According to the WHO stroke is the second leading cause of death worldwide. Of the approximately 55.4 million annual deaths, stroke accounts for around 11% of them, surpassed only by another cardiovascular-related issue, ischemic heart disease. Stroke is responsible for 1 out of every 6 deaths from cardiovascular disease.

Worldwide each year, around 15 million people will suffer a stroke. Of these roughly a third will die and about half of those who survive will be left disabled. In fact, stroke is a leading cause of long-term disability.

In the United States each year, almost 800,000 people suffer a stroke. That's roughly one person every forty seconds, with a stroke-related death occurring every three-and-a-half minutes on average. Of these, about one fourth have had a previous stroke.

There are some known factors that increase the likelihood of experiencing a stroke. As you might expect, this includes factors related to cardiovascular health such as high blood pressure and cholesterol, obesity and diabetes, even age. There are also some less obvious factors related to increased risk of stroke, such as race (being black doubles your risk relative to being white) and where you reside (living in the South increases your mortality risk).

To help prevent or, at minimum, potentially reduce the severity of a stroke, early detection is important. If we can identify individuals who are likely to experience a stroke, we can educate them both on what actions they can take to reduce their risk of having a stroke as well as raise awareness of what warning signs to look for and how to quickly respond in the event of a stroke. According to the CDC, "patients who arrive at the emergency room within 3 hours of their first symptoms often have less disability 3 months after a stroke than those who received delayed care."

Data Explanation:

About the dataset:

I found a couple of datasets on Kaggle that could be used for stroke prediction. Both had the same columns, but one consisted of over 40,000 entries, while the other only had around 5,000. While the smaller dataset had a higher proportion of positive stroke data (around 5% vs around 2% of the larger dataset), I went with the larger dataset because I figured that overall, more data was better and that I may end up removing a portion of the non-stroke data anyway.

At only 12 columns, two of which are the ‘id’ and target ‘stroke’ columns, there weren’t as many features as I had hoped. The target is binary 1/0 for whether the patient experienced a stroke. The 10 data columns consist of 5 numeric features, 2 of which are binary yes/no type encoded as 1/0. The other 5 are categorical, though one is binary yes/no, which I will encode to 1/0, and another that, while it isn’t yes/no, still only has two categories: Rural and Urban, which I will also encode to 1/0. The others have multiple categories but the third category in ‘gender’, listed as Other, only makes up 11 entries out of the 40,000. None of these 11 are listed as positive for stroke so I’m tempted to remove those rows so that I can make gender a single, binary row. Of course, this raises some ethical concerns.

With the huge imbalance in the target class, I will need to balance the data prior to modeling, but I also considered other options for removing some of the non-stroke data. With the risk of stroke increasing with age, I considered removing some of the younger participants in the study. I found that only 4 people under the age of 30 had a stroke and that group makes up around a third of the dataset. So, by removing that group I could improve my stroke to non-stroke ratio of the remaining data. While I did try this at first, I ended up leaving that data in because I got better results when modeling.

Distributions:

One of the first things I noticed when looking at histograms of the dataset was that the age distribution had huge spikes at the beginning and end. On the lower end I had initially assumed that there was a mistake because I don’t know why they would include children, especially under age 1, but I looked at some of the decimals provided and it looks like they correspond with fractions by month (i.e., twelfths of a year) and so are possibly intentional. I’m not sure why there is a spike at the upper age range (>80). I thought maybe entries over a certain age were rounded down to remove outliers or something, but it turns out that ages 80-82 are all overrepresented, though it is odd that there are no entries over age 82.

See Appendix, Table 1: Histograms

Considering possible outliers in blood sugar (‘avg_glucose_level’) and BMI, I looked up normal and possible ranges for humans and it looks like the distributions are within normal ranges. Both features appear to be skewed, however, and blood sugar is slightly bimodal.

One thing I found when looking at distributions was that it was difficult to see any information about the positive stroke results because the data is so imbalanced and so they appear almost as a flat line at the bottom. To better compare stroke vs non-stroke data, I performed a quick and dirty oversampling of the stroke data and created histograms again.

See Appendix, Table 2: Histograms (Balanced)

This was helpful in visualizing which factors may indicate a larger stroke risk. For example, while ‘gender’ or ‘Residence_type’ don’t show much difference between stroke and non-stroke groups, health-related variables, like ‘age’, ‘hypertension’, ‘heart_disease’, and ‘avg_glucose_level’ (associated with diabetes) appear to show some differences. Interestingly, while I didn’t expect ‘work_type’ to have any impact, being self-employed shows a clear increase in the stroke crowd (as well as being a child reducing risk) and being a former smoker, though being a current smoker doesn’t appear to have much impact.

Imputation of Null values:

Almost a third of the smoking variable consisted of Null values. Since this was already categorical, I simply created a class of ‘unknown’, which was then the second largest class in the group.

The only other variable containing Null values was ‘bmi’, of which only a little over 3% of the data was missing. I spent some time trying to figure out the best way to impute the missing values. Given the other available variables, I thought it seemed reasonable that I might be able to extrapolate approximate values using logistic regression. I compared results using these imputed values with data sets where imputations were made using mean and median. While it was a fun learning exercise for me, I ultimately landed on using median because it is much less complicated, and it provided me with the best results.

One-hot encoding:

After encoding ‘gender’, ‘ever_married’, and ‘Residence_type’ as binary 1s and 0s to keep the number of features down to a reasonable number, I one-hot encoded the categorical features ‘work_type’ and ‘smoking_status’.

Interestingly, when comparing stroke vs non-stroke distributions, this new ‘unknown’ class was exceedingly non-stroke. I considered reducing this variable to a smoking / non-smoking feature or even a one-vs-all former-smoker / other feature, but ultimately decided to leave it alone.

Similarly, I considered converting ‘work_status’ to a binary one-vs-all of self-employed / other, but decided to leave it as-is as well.

Splitting:

Once imputations were made and categorical variables were encoded, I split the data into training and testing sets. While it is technically preferred to split prior to imputing with median, which is at least in part derived from the test data, I felt that the impact of using the median was small enough that it would not be a significant issue.

Splitting was however done prior to transformations and scaling to prevent leaking of information that might otherwise occur from the test data into the training data.

Transformations:

I was able to perform box-cox transformations on the two skewed variables (glucose and BMI) as well as age. Because none of these were at or below zero, I did not have to rescale prior to the transformation.

See Appendix, Table 3: Box-Cox Transformation

Since box-cox converts the data to roughly a -2.5 to 2.5 range and the other binary variables are 0/1, I then scaled everything to a -1 to 1 range so that they were uniform, which is preferred in some models and shouldn’t hurt for other models.

Methods:

Several things considered in generating a useful predictive model from this dataset are the imbalanced target variable, which algorithms are appropriate for this type of dataset, and which metrics to use in comparing the models.

Feature selection:

Above I discussed several potential options for reducing the number of features, but effectively having only 10 features to begin with, I decided it was best to keep them in rather than risking loss of potentially useful information.

Balancing:

With positive stroke results making up only around 2% of the dataset, I needed to balance the data. I discovered this firsthand when I was initially trying to compare imputation methods and the results were consistently coming out at 98% accuracy. When balanced, the accuracy percent dropped to the mid-70s using a logistic regression model. Rather than sampling from the negative stroke data to reduce the size, I initially chose to upsample from the positive stroke data to match the size of the negative data.

Having limited success with resampling the positive stroke data, I tried upsampling using SMOTE. This showed some improvement, but almost doubled the data size without adding any useful information. One strategy I came across in my research was to upsample to 10% of the majority class using SMOTE, then sample the majority class down to double the size of the minority class. This made the data set a much more manageable size without any apparent negative impact to the results.

Finally, I went back to not balancing at all and instead utilizing the balanced class weight option available in many classification algorithms. I was satisfied with this method enough to proceed with model comparisons.

Algorithm selection:

In selecting the best algorithm, I tried out as many models as I could find that seemed appropriate for the task and did an initial comparison using accuracy, F1-score, precision, and recall as metrics. I selected the top five models, which showed accuracy of > 80%.

As I dug deeper into the model results, I experience several realizations. First was that accuracy is a poor metric for this data set because of the large imbalance (~98% negative for stroke). It occurred to me that it would be preferable to positively predict stroke and be wrong (false positive) than to predict no stroke and be wrong (false negative). To this end, I considered using the Recall metric, but this could lead to the opposite issue where a model would be biased toward predicting everything as positive. I ruled out the F1 Score because, while it provides a balance between recall and precision, it is impacted too much by the imbalanced data.

After some additional digging, I came across Matthews Correlation Coefficient, which is supposed to be one of the more reliable metrics for imbalanced binary data sets. Similarly, ROC AUC is a useful metric for comparing the number of true positives and true negatives.

See Appendix, Table 4: Metrics for Classification Models

Hyperparameter tuning:

I initially performed Hyperparameter tuning on the top five most accurate models to improve my results, primarily using GridSearchCV. I used RandomSearchCV for the MLP model because there were so many parameters it never would have finished in time. In fact, I ran RandomSearchCV multiple times and then performed GridSearchCV using only the best parameters from the results of these.

After adjusting my model evaluation metrics, I selected models with MCC > 0.15 and sorted by Recall to determine the top models, which were Logistic Regression, Support Vector Classifier (SVC) and Linear SVC. I later added Stochastic Gradient Decent (SGD) because, while it didn't score as well with MCC, it was comparable to the other models in AUC and Recall and I thought it could possibly be improved with some hyperparameter tuning.

I later discovered that I could perform hyperparameter tuning using multiple metrics, so I added MCC, Recall, and AUC to the scoring and ran GridSearchCV with each of the selected top models. Logistic Regression showed basically no improvement, but both SVC methods improved slightly and SGD by quite a bit. I wish I had more time to continue playing around with this.

I also started learning about utilizing a VotingClassifier ensemble with weights based on scores from various metrics and saw some further improvement by grouping the models and weighting based on Recall, but again I didn't get to dive as deep as I wanted into this.

Analysis:

While several of the initial models showed greater than 95% accuracy, there was still an imbalance when looking at the confusion matrices. While they only make up a small number of the total observations (251 instances out of >14k records), only a relatively small proportion of the positive stroke instances are correctly predicted in the models with the highest accuracy.

After giving it some thought, I decided it would be better to err on the side of predicting stroke for individuals who don't have one (false positive) than predicting no stroke for someone who then has a stroke (false negative). Because of this, I modified the metrics for comparing model performance, landing on a combination of Matthews Correlation Coefficient, Area Under Curve, and Recall.

Conclusion:

Stroke prediction seems feasible, given enough information. This dataset could have benefited from an increased amount of positive stroke data as well as additional feature variables. With the given dataset, I ended up with more false negatives than I would like to see, but as with many things there is a balance to be struck. You could potentially predict almost all the true positives, but this would require incorrectly predicting the vast majority of the negative cases (false negatives), which doesn't make for a very useful model.

Based on the risk factors we already know, it might be interesting to take a different approach, collecting a lot of data that may or may not be related to stroke and performing feature selection to determine which of the feature variables are the best predictors of stroke. This information could be used to identify the highest risk factors and to help people mitigate their stroke risk.

Assumptions:

Apart from knowing it was available in Kaggle, I don't know anything about the actual source of the data and how it was generated. My (very big) assumption here is that the data was collected from a legitimate and representative source. Of the two similar stroke datasets I found, I selected this one because it was the larger dataset, with my assumption being that more is better. However, since there was no real information about the source of the data, it is possible that this larger dataset was generated from the smaller set and may not represent real-world data. There are some spikes in the histograms that suggest this data may already have been processed a bit, such as with the age feature, where there are spikes at the beginning and end, suggesting higher and lower values could have been rounded and one in the middle, which could be null values imputed with mean or median.

Limitations:

One limitation of the dataset is that it consists in large part of people who haven't had a stroke *yet*. Just because they haven't had a stroke at the time they were surveyed, doesn't mean they won't have one at some point in the future. Of course, it may be the case that the stroke column may indicate whether the individuals had a stroke in the following year (or some other timeframe), in which case the model results would indicate a short-term prediction.

While there weren't a lot of feature variables to use for prediction, some that were in the dataset are associated with known risk factors, like cardiovascular disease, high cholesterol, obesity, and diabetes. However, some known risk factors were not represented at all, like family history, location, race, and previous stroke.

Perhaps the biggest limitation was that positive cases make up a very small portion of the dataset.

Challenges:

One challenge I ran into was imputing Null values using logistic regression. It turned out not to be the best option for imputation with my dataset, but it was a learning experience nonetheless.

Balancing the dataset was a big challenge. Initially, I oversampled the small number of positive stroke observations to match the number of negative observations, but I believe that led to overfitting to those few positive examples. I later tried over-sampling using SMOTE as well as a combination of over- and under-sampling, but weighting the model appeared to work best to help manage the imbalance.

Future Uses / Additional Applications:

Stroke prediction, either on a case-by-case basis or in establishing relative risk factors, can be used to save lives around the world, especially if it can be simplified or lead to preventative measures.

Recommendations:

My first recommendation would be to include more feature variables in the dataset. Following this, I would like to see more positive stroke cases included.

Implementation Plan:

This predictive model could be useful for individuals to take some ownership of their personal healthcare. I would like to see an app where you are able to answer sets of questions and regularly provide and track your health information, such as your physical characteristics (weight, etc.), family history, results of blood screenings, blood pressure and the like. It could even connect to exercise and sleep apps to compile and screen your data and it could be a useful tool to share with your primary doctor at visits.

What I would like to see this app do, then, is to gamify your healthcare in such a way that it provides you with your personal health risks, like stroke or heart disease, and provide both personalized recommendations and feedback on actions you may take to improve your health. For example, if you start exercising more, it would show your cardiovascular health risks declining. Perhaps you could earn rewards, like badges or money that is good at a particular health food store or for fitness apparel.

Ethical Assessment:

Race was not included in the dataset, despite being a known risk factor. However, if it had been included, care would need to be taken to ensure that racial biases aren't resulting in any unintended consequences in the model results.

One ethical quandary I ran into was what to do with the 'Other' class in the 'gender' category. I removed it to simplify the model (and because they only accounted for 11 observations), but in the real world, there will be people who check the 'Other' box and will need to be included. There didn't appear to be a large difference in gender so it may not matter much, but it something to keep in mind.

A consideration with apps that collect personal information is securing that personal data, especially when it is as personal as an individual's private health information. Any company that stores this kind of information needs to take extra care to prevent bad actors from accessing the data and they definitely shouldn't share the data with a third party without explicit and clear consent.

References:

Amal, Lirilkumar. "Heart Stroke." Kaggle, 26 Oct. 2020, https://www.kaggle.com/datasets/lirilkumaramal/heart-stroke.

Brownlee, Jason. "How to Develop a Weighted Average Ensemble with Python." Machine Learning Mastery, 7 May 2021, https://machinelearningmastery.com/weighted-average-ensemble-with-python/.

Brownlee, Jason. "Train-Test Split for Evaluating Machine Learning Algorithms." Machine Learning Mastery, 26 Aug. 2020, https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/.

Chicco, Davide, and Giuseppe Jurman. "The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation." BioMed Central, BMC Genomics, 2 Jan. 2020, https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-019-6413-7.

Eddie_4072. "Dealing With Missing Values in Python." Analytics Vidhya, 20 June 2022, https://www.analyticsvidhya.com/blog/2021/05/dealing-with-missing-values-in-python-a-complete-guide/.

Emergency Nutrition Network. "The Limits of Human Starvation." Field Exchange 15, 4 Jan. 2002, https://www.ennonline.net/fex/15/limits.

Emon, Minhaz Uddin, et al. “Performance Analysis of Machine Learning Approaches in Stroke Prediction.” 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2020, https://doi.org/10.1109/iceca49313.2020.9297525.

Know Your Risk forStroke. Centers for Disease Control and Prevention, 12 Apr. 2022, https://www.cdc.gov/stroke/risk_factors.htm.

Kudva, Yogish C. “Diabetes.” Mayo Clinic, Mayo Foundation for Medical Education and Research, 30 Oct. 2020, https://www.mayoclinic.org/diseases-conditions/diabetes/diagnosis-treatment/drc-20371451.

Nik. “One-Hot Encoding in Scikit-Learn with Onehotencoder.” Datagy, 23 Feb. 2022, https://datagy.io/sklearn-one-hot-encode/.

Stroke Facts. Centers for Disease Control and Prevention, 5 Apr. 2022, https://www.cdc.gov/stroke/facts.htm.

Stroke, Cerebrovascular Accident. World Health Organization, http://www.emro.who.int/health-topics/stroke-cerebrovascular-accident/index.html.

The Top 10 Causes of Death. World Health Organization, 9 Dec. 2020, https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

“What Is the Body Mass Index (BMI)?” NHS Choices, NHS, 15 July 2019, https://www.nhs.uk/common-health-questions/lifestyle/what-is-the-body-mass-index-bmi/.

Zach. “How to Calculate AUC (Area under Curve) in Python.” Statology, 9 Sept. 2021, https://www.statology.org/auc-in-python/.

Appendix:

Table 1: Histograms

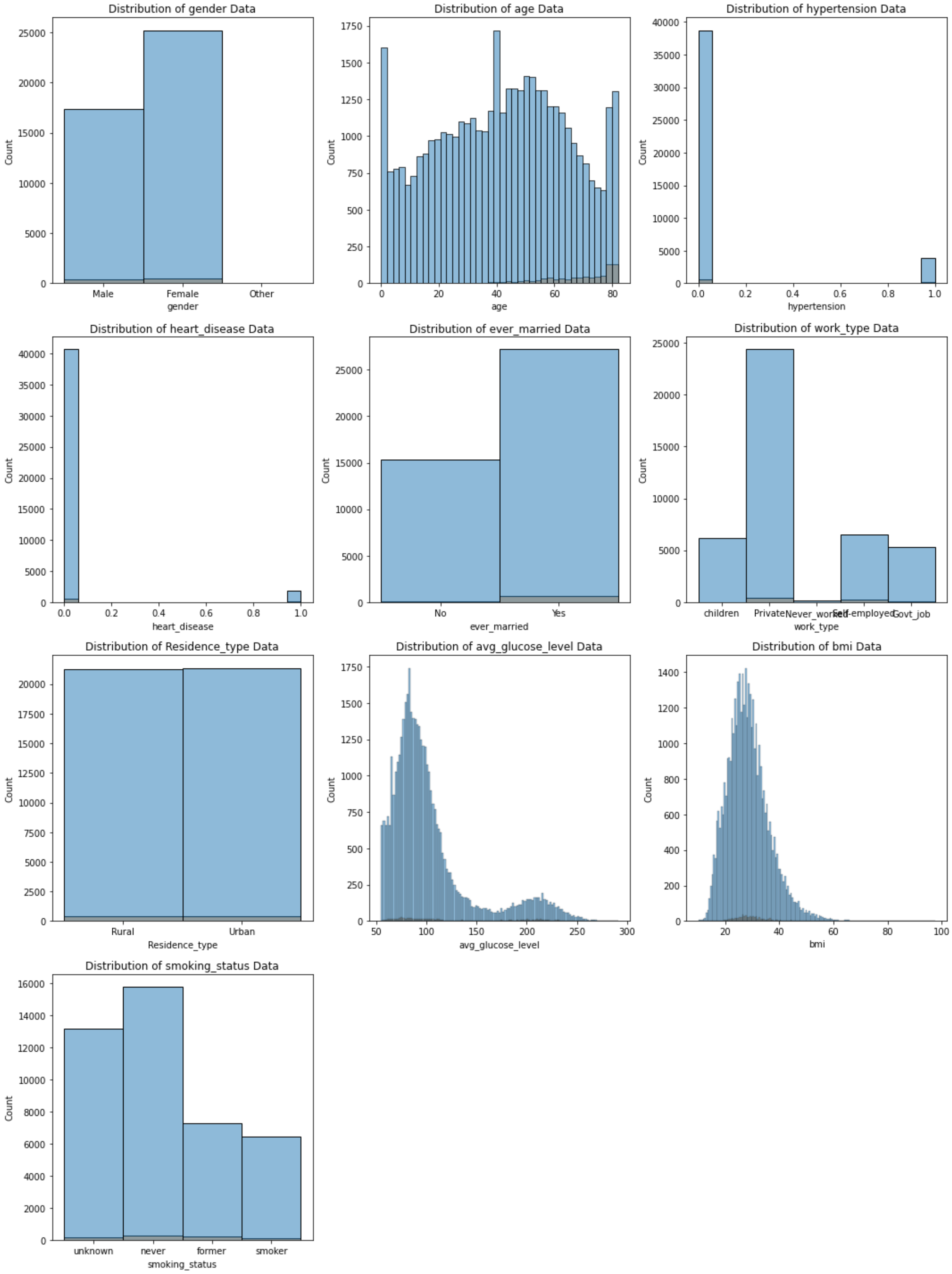


Table 2: Histograms (Balanced)

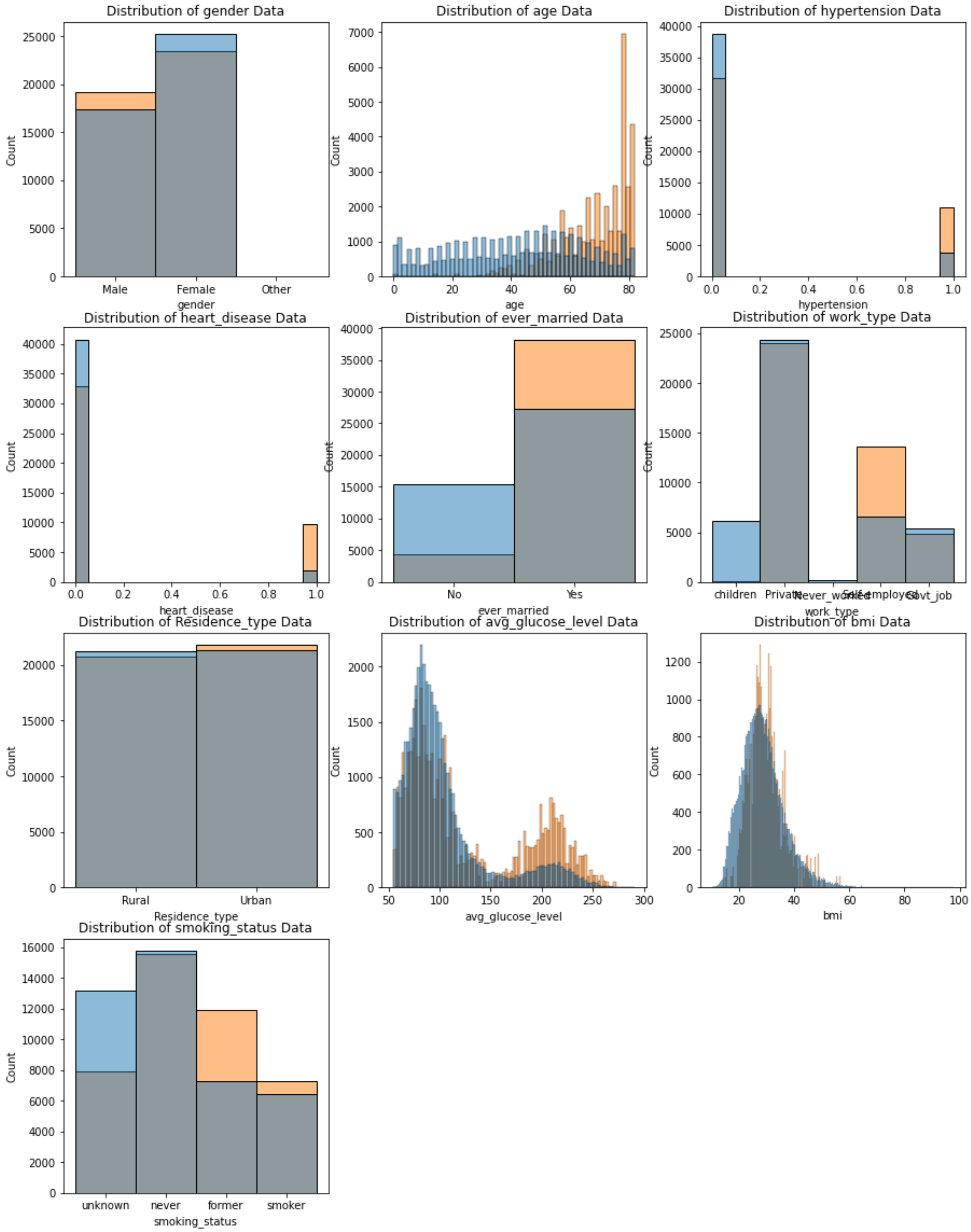
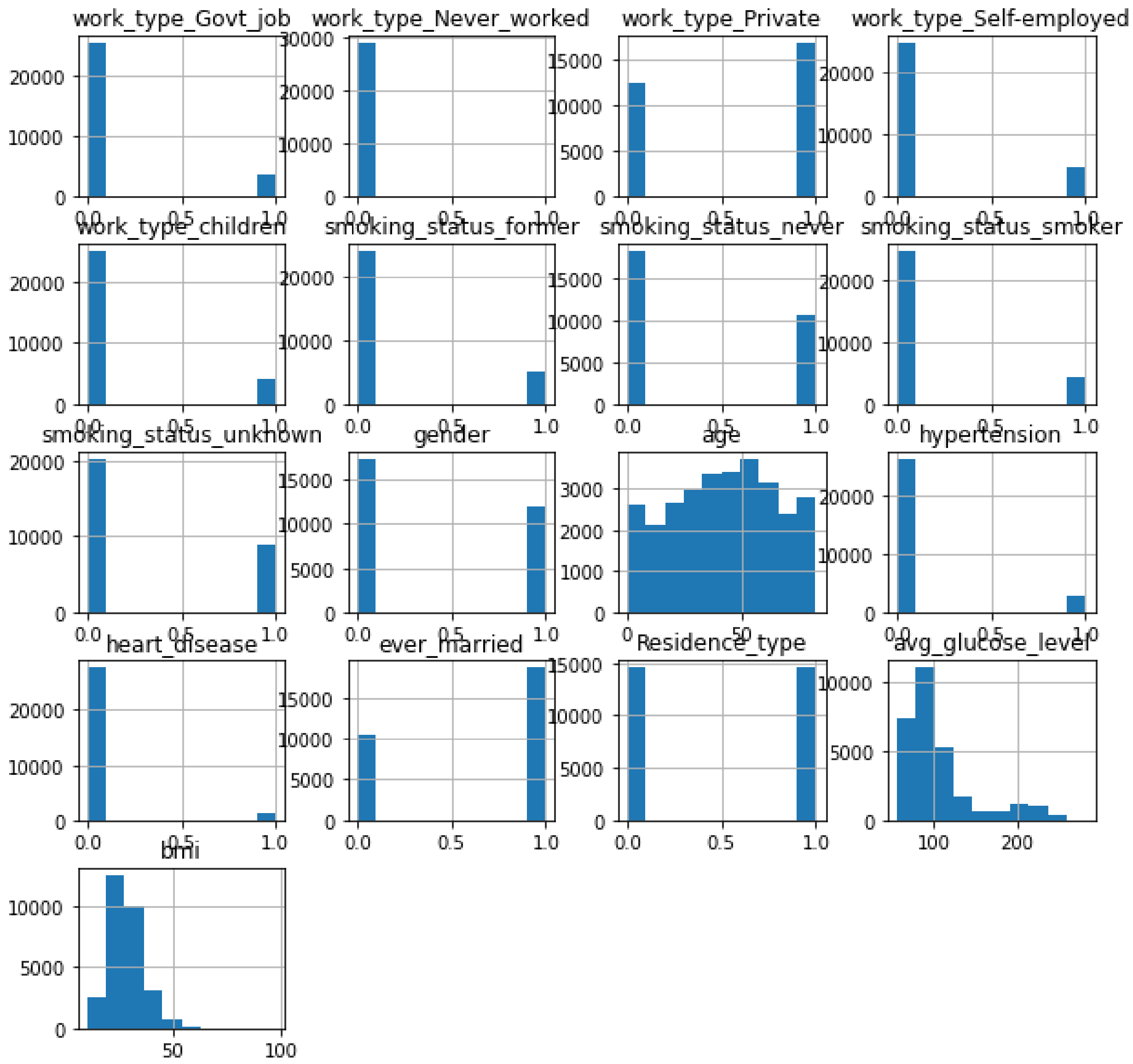


Table 3: Box-Cox Transformation

3a: Feature distributions before Box-Cox Transformation



3b: Feature distributions after Box-Cox Transformation

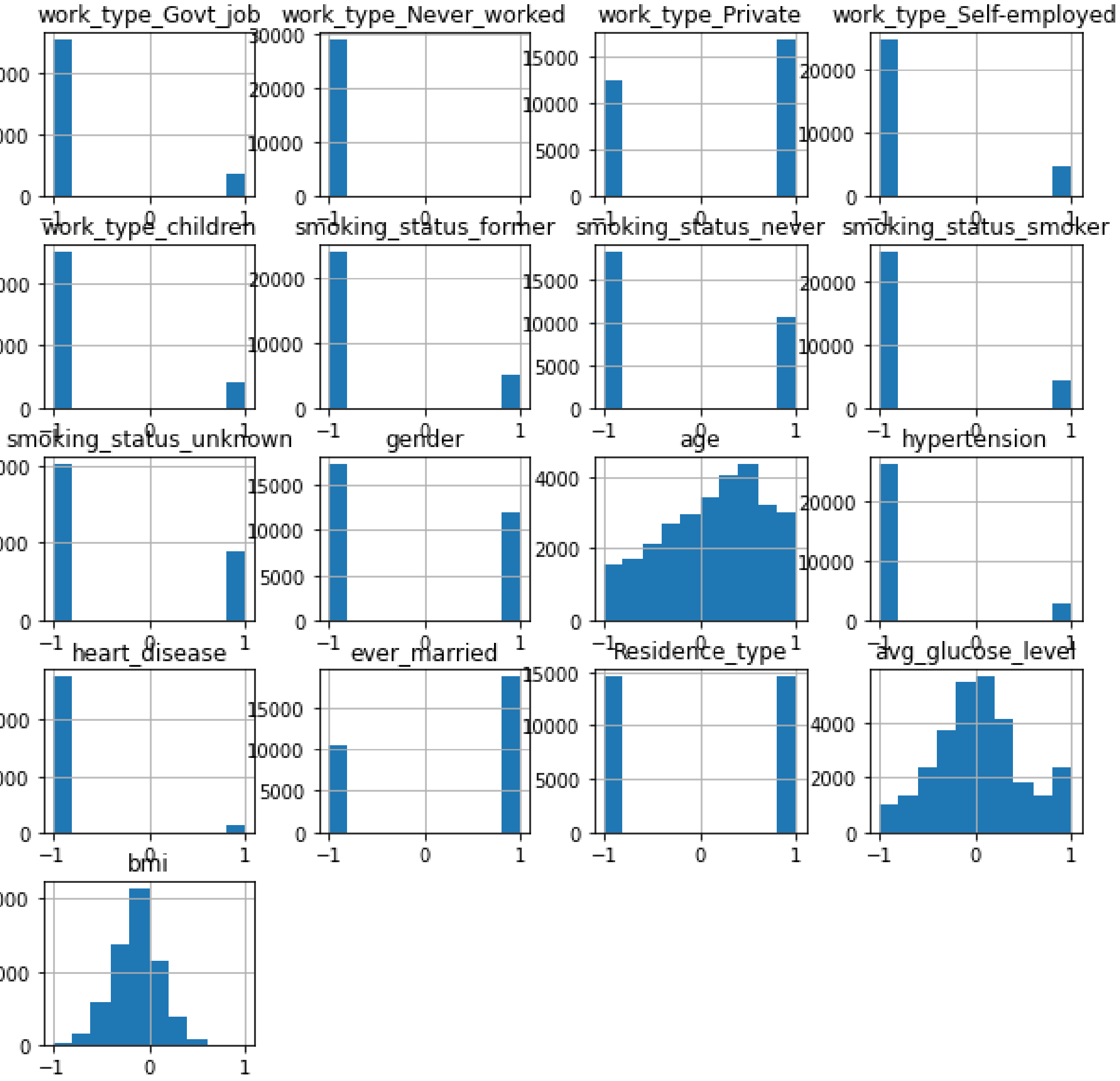
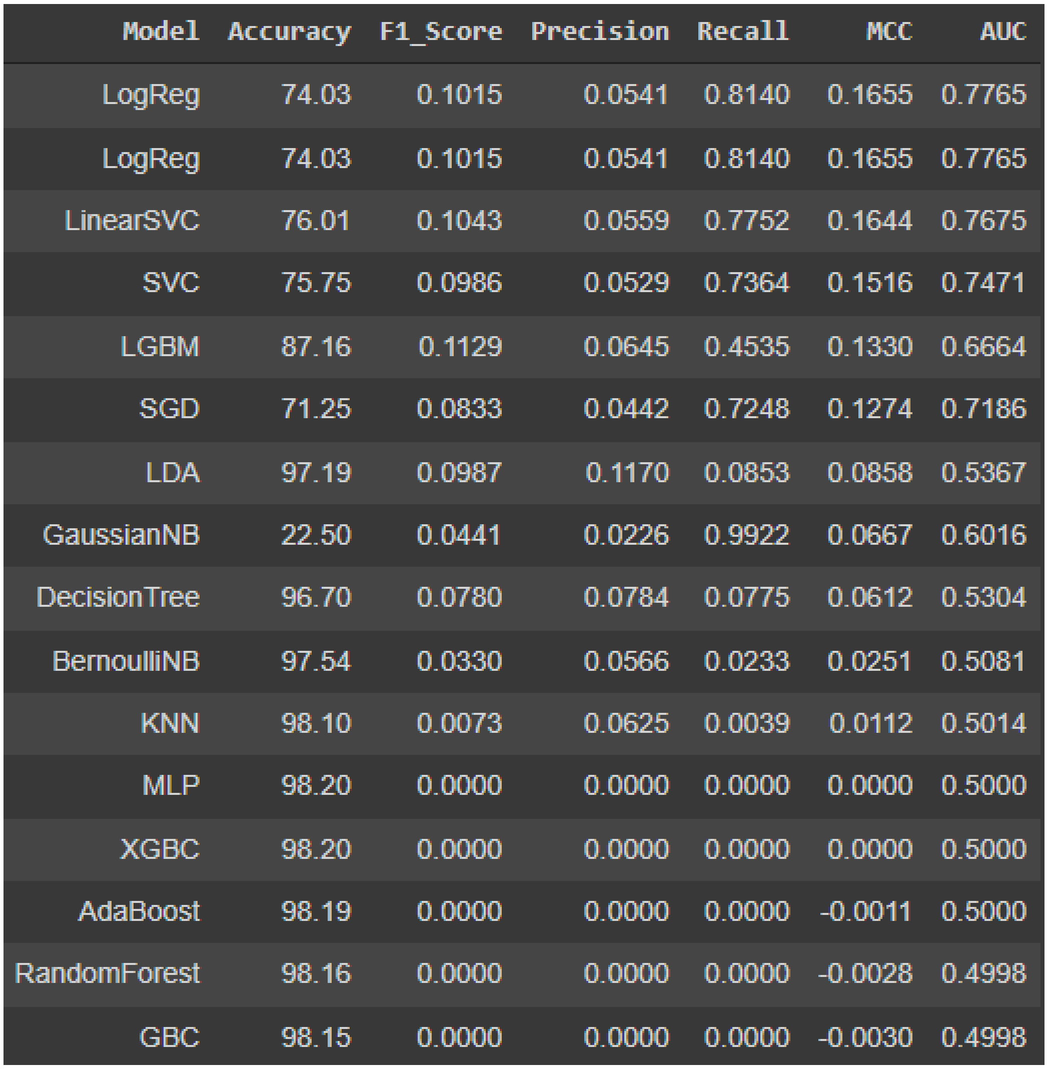


Table 4: Metrics for Classification Models



Model	Accuracy	F1_Score	Precision	Recall	MCC	AUC
LogReg	74.03	0.1015	0.0541	0.8140	0.1655	0.7765
LogReg	74.03	0.1015	0.0541	0.8140	0.1655	0.7765
LinearSVC	76.01	0.1043	0.0559	0.7752	0.1644	0.7675
SVC	75.75	0.0986	0.0529	0.7364	0.1516	0.7471
LGBM	87.16	0.1129	0.0645	0.4535	0.1330	0.6664
SGD	71.25	0.0833	0.0442	0.7248	0.1274	0.7186
LDA	97.19	0.0987	0.1170	0.0853	0.0858	0.5367
GaussianNB	22.50	0.0441	0.0226	0.9922	0.0667	0.6016
DecisionTree	96.70	0.0780	0.0784	0.0775	0.0612	0.5304
BernoulliNB	97.54	0.0330	0.0566	0.0233	0.0251	0.5081
KNN	98.10	0.0073	0.0625	0.0039	0.0112	0.5014
MLP	98.20	0.0000	0.0000	0.0000	0.0000	0.5000
XGBC	98.20	0.0000	0.0000	0.0000	0.0000	0.5000
AdaBoost	98.19	0.0000	0.0000	0.0000	-0.0011	0.5000
RandomForest	98.16	0.0000	0.0000	0.0000	-0.0028	0.4998
GBC	98.15	0.0000	0.0000	0.0000	-0.0030	0.4998