1. Executive Summary

This project attempts to predict if a patient will have a stroke event in the near future. I was not given an exact definition of 'near future', and I was not provided a background for the dataset used, including but not limited to: how the data was collected, how old the data is, how accurate the data is, and the population it is meant to represent.

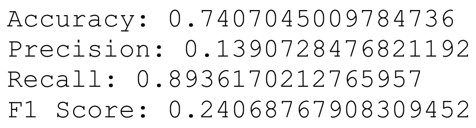
My features, after dropping the arbitrary 'id', included gender, age, hypertension, heart_disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, and smoking_status. I augmented the features to include a cross term between (avg_glucose_level and bmi) and (hypertension and heart_disease).

As we will be predicting strokes in the near future, it is imperative to focus on the ratio of correctly predicted stroke versus all stroke labels. When predicting diagnosis, it is better to predict more labeled strokes as strokes at the cost of diagnosing more false labels as true. To do this, I evaluate my models with an emphasis on the recall score: TP/(TP+FN).

I began by examining basic characteristics of the dataset. I found a large imbalance between stroke labels and no stroke labels. In the dataset, there were 4,861 no stroke labels compared to only 249 stroke labels. I created a data pipeline and used SMOTE to balance the training dataset. Later, I used Principal Component Analysis to reduce the feature space from 16 to 6. I used this updated dataset for some, but not all models. In general, PCA did not increase my recall score for most models. Details will be provided in the methodology and result section.

I created and tested several models, including unbalanced logistic regression, balanced logistic regression, bagging with decision tree classifiers(DTC), balanced bagging with DTC, and a multi-layer perceptron(MLP) model with semi-optimized parameters. When examining recall in specific, my best model was an optimized MLP neural-network using a balanced, oversampled, training dataset and PCA reduced feature space.

My metrics for this model are the following:



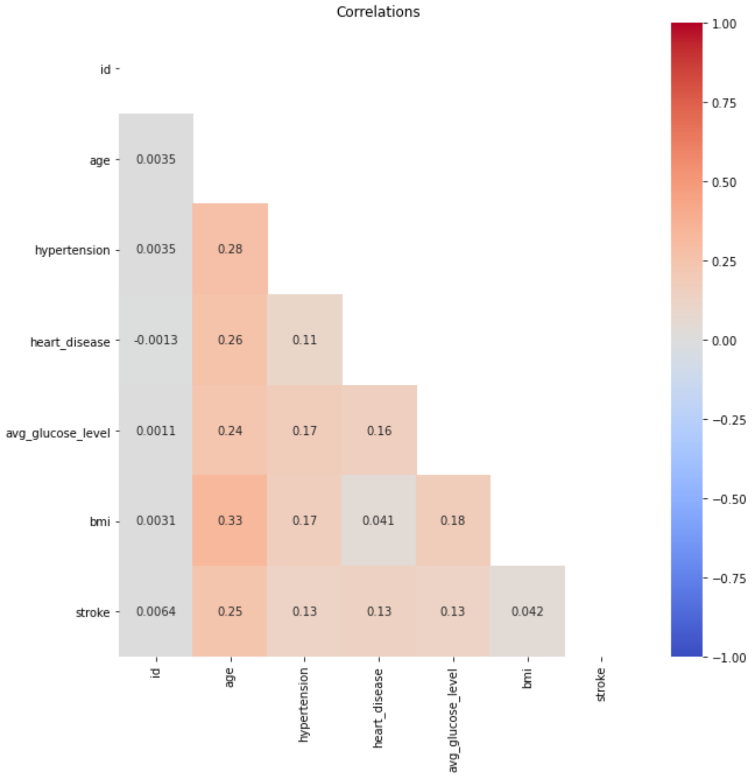
```
Accuracy: 0.7407045009784736
Precision: 0.1390728476821192
Recall: 0.8936170212765957
F1 Score: 0.24068767908309452
```

One can see from the previous metrics, that my model massively overclassified stroke labels. A benefit of this, was a high recall score, indicating that the model is effective in predicting those that will have a stroke in the near future to have a stroke soon. It does, however, predict many more people who will not have a stroke soon to have a stroke in the near future. I chose to go with this overclassified result, as it is better for doctors and patients to be safe than sorry when determining if the patient will have a stroke soon.

2. Background/Introduction

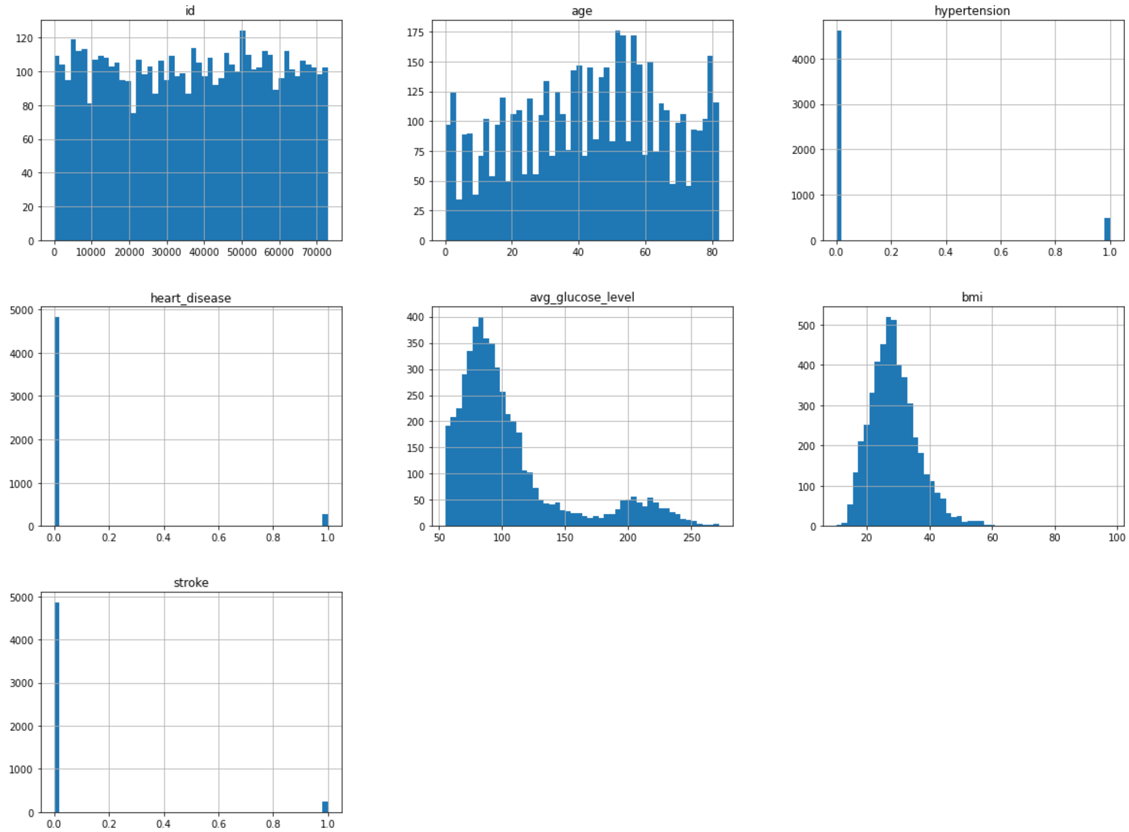
According to the CDC, more than 795,000 people in the U.S. have a stroke each year. This results in someone in the U.S. having a stroke every 40 seconds. Stroke rates rapidly differ between different ethnic and social populations. For example, the risk of a stroke is double for African Americans when compared to Americans for western European descent.

According to the WHO, high blood pressure, high cholesterol, smoking, obesity, and diabetes are heavily correlated with stroke risk. The below correlation matrix between the initial numerical features validates this claim. In the figure blow, age, hypertension, heart disease, average glucose(sugar) level, and BMI were positively correlated with stroke rate in that decreasing order.



3. Methodology

I initially dropped the id feature, as it was arbitrarily decided at the time of data collection and has no benefit for my models. I checked for missing values and found that only body mass index(bmi) had missing values, with only ~4% of its values being nan. After creating histograms for my numerical features (included below), I found that bmi was approximately normally distributed.



The mean was 28.86 and the median was 28.1, as such, I decided there was no large benefit between a mean and median imputation strategy. This was confirmed when I used both to create separate KNN models with highly similar outcomes. I then scrapped this testing and stuck with median imputation.

My overall strategy for the data pipeline was to impute bmi, then augment two new features, then label encode my categorical features with an implicit ordering, then one hot encode the work_type, as there were many non-ordinal values, and use a standard scaler for the numerical features. As this was a complicated data pipeline, I chose to gradually pipeline the data rather than use one large column transformer with sub-pipelines for everything.

My first augmented feature is: weight_glucose = avg_glucose_level * bmi. As diabetes was found by the WHO to correlate to strokes, I believed a cross term of glucose blood level and body mass index could be indicative of diabetes or similar issues. Likewise, my second augmented feature is: hyper_tension_heart_disease = hypertension * heart_disease. As both hypertension and heart_disease were Boolean values, hyper_tension_heart_disease is also a Boolean value that is true if the patient has both hypertension and a heart disease.

I then label encoded the other binary features, which did nothing as I wanted it too. I also decided to label encode gender, as there was only one ‘other’ gender value in the dataset, and women were found to be more likely to have a stroke according to the

CDC, so there is an implicit ordering in gender in relation to stroke risk. Similarly, those who were married had a lower risk of stroke, so I kept the implicit ordering in ever_married. The same logic applies to why I label encoded smoking_status. I then used a standard scaler for the numerical features, so features like avg_glucose_level would not dominate my models. After one hot encoding work_type as discussed above, my unbalanced pipeline dataset was ready with a feature space with a dimensionality of 16. As this was a relatively large feature space, I chose to use PCA to reduce this down to 6. Multiple values were tested in the range of 3 to 13, but 6 seemed to maximize my recall scores across the models.

It should be noted that prior to using PCA, I first created a basic logistic regression model to classify strokes. After initially using the unbalanced pipelined data, I found that my model obtained 95% accuracy by simply predicting all entries to not have a stroke in the near future. To counteract this, I had to balance my model. When balancing my dataset, I ensured that I only balanced my training data and left the test data as is, so I could have a better representation of how my model would perform on the population. I chose not to under sample my no stroke data, as I would be throwing away most of my data and could remove important patterns in the process. Rather than under sample no strokes, I chose to oversample the entries that were labeled to have a stroke in the near future. I used SMOTE from imblearn to do this. After balancing my dataset this way, my recall and precision scores were no longer 0.

My initial ensemble method was a bagging classifier using decision trees. I used my balanced training dataset. I also trained a balanced bagging classifier that created sub-datasets by randomly picking from the existing balanced training set with replacements.

I then trained a neural net classifier using a multi-layer perceptron network. I played around with different values of the width and depth of the network along with largely differing alpha values to semi-optimize my parameters.

I then performed K-fold cross validation on my ensemble and neural net classifiers to obtain a better representation of how they would perform on population data.

Following this, I created more comparisons of the MLP model to generate my best model, with a large focus on recall rather than precision or accuracy for the reasons states in the executive summary section.

4. Results

My initial unbalanced logistic regression model gave the following metrics:

```
Accuracy: 0.9422700587084148
Precision: 0.0
Recall: 0.0
F1 Score: 0.0
```

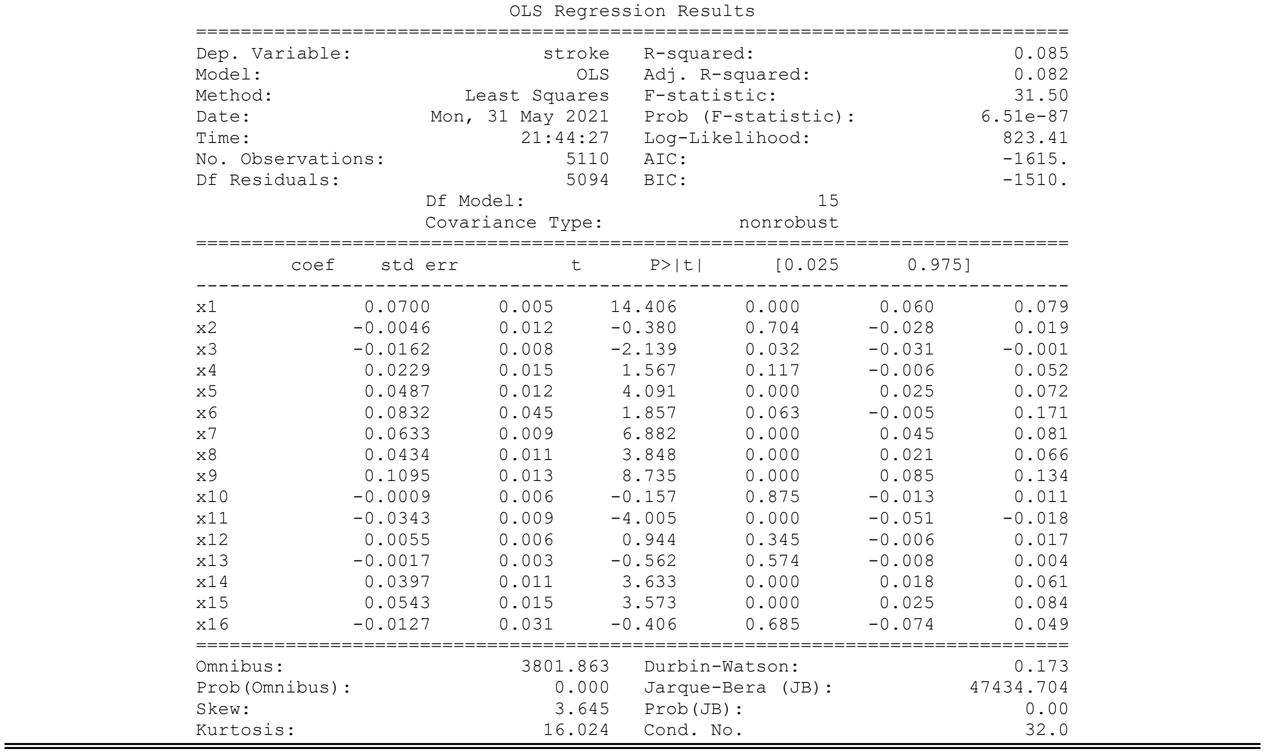
The model obtained a high precision by classifying all test entries as no stroke. This demonstrates how the unbalanced data led to a model that was highly biased towards the majority class.

After balancing only my training data and repeating the same basic logistic regression model, I obtained the following metrics:

```
Accuracy: 0.738747553816047
Precision: 0.15562913907284767
Recall: 0.7966101694915254
F1 Score: 0.260387811634349
```

My recall was relatively high, at the expense of a low precision. Overall, all my metrics improved, with an emphasis on recall, after balancing my model.

I then performed statical modeling for feature selection, using ordinary least squares.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 stroke   R-squared:                       0.085
Model:                            OLS   Adj. R-squared:                  0.082
Method:                 Least Squares   F-statistic:                     31.50
Date:                Mon, 31 May 2021   Prob (F-statistic):           6.51e-87
Time:                        21:44:27   Log-Likelihood:                 823.41
No. Observations:                5110   AIC:                            -1615.
Df Residuals:                    5094   BIC:                            -1510.
                     Df Model:                          15
                     Covariance Type:            nonrobust
==============================================================================
        coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.0700      0.005     14.406      0.000       0.060       0.079
x2            -0.0046      0.012     -0.380      0.704      -0.028       0.019
x3            -0.0162      0.008     -2.139      0.032      -0.031      -0.001
x4             0.0229      0.015      1.567      0.117      -0.006       0.052
x5             0.0487      0.012      4.091      0.000       0.025       0.072
x6             0.0832      0.045      1.857      0.063      -0.005       0.171
x7             0.0633      0.009      6.882      0.000       0.045       0.081
x8             0.0434      0.011      3.848      0.000       0.021       0.066
x9             0.1095      0.013      8.735      0.000       0.085       0.134
x10           -0.0009      0.006     -0.157      0.875      -0.013       0.011
x11           -0.0343      0.009     -4.005      0.000      -0.051      -0.018
x12            0.0055      0.006      0.944      0.345      -0.006       0.017
x13           -0.0017      0.003     -0.562      0.574      -0.008       0.004
x14            0.0397      0.011      3.633      0.000       0.018       0.061
x15            0.0543      0.015      3.573      0.000       0.025       0.084
x16           -0.0127      0.031     -0.406      0.685      -0.074       0.049
==============================================================================
Omnibus:                     3801.863   Durbin-Watson:                   0.173
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            47434.704
Skew:                           3.645   Prob(JB):                         0.00
Kurtosis:                      16.024   Cond. No.                         32.0
```

My low R-squared value indicated that I had extraneous features and should perform PCA to eliminate the features that provide little insight into stroke rate. Using the p-value and t-test indicated above, I believed only 8 features were truly correlated with stroke likelihood in the near feature. Upon further examination of different PCA reductions, 6 ended up being my ideal feature space size.

Following this PCA, I then used a basic bagging classifier model that used simple decision trees . My metrics for this model are the following:

```
Accuracy: 0.8874755381604696
Precision: 0.1375
Recall: 0.19298245614035087
```

```
F1 Score: 0.16058394160583941
```

This performed worse than my balanced logistic regression model with regards to recall but had a higher accuracy score. As my emphasis was on recall, I chose to rerun this model using a balanced bagging classifier instead. My metrics were the following:

```
Accuracy: 0.8796477495107632
Precision: 0.13333333333333333
Recall: 0.21052631578947367
F1 Score: 0.163265306122449
```

My recall increased negligibly, indicating that this model was not the one to use for predicting strokes. I then trained several neural network classifiers using a multi-layer perceptron network. I played around with different hidden layer sizes and alpha values. I found my best overall result to have an alpha=1e-9 and the hidden layer to be 2x8 nodes. The metrics were the following:

```
Accuracy: 0.6868884540117417
Precision: 0.13774104683195593
Recall: 0.8771929824561403
F1 Score: 0.2380952380952381
```

This was the highest recall score I have had and seemed like the best model so far for this study.

I then performed K-fold cross validation on my initial ensemble and neural network models. My bagging model returned an average accuracy of .7984 and my MLP neural network model returned an accuracy of .951 but had a low recall score.

I then further optimized my neural network MLP classifier by keeping alpha at 1e9 but increasing the depth of the neural network to 9. I fit it with my oversampled balanced dataset and returned the following metrics:

```
Accuracy: 0.7407045009784736
Precision: 0.1390728476821192
Recall: 0.8936170212765957
F1 Score: 0.24068767908309452
```

5. Discussion

By no means should my best model be used as a diagnostic test to determine if a patient will have a stroke in the near future. The UCLA hospital can use this model as part of their reasoning to conduct further tests and inquiries into a patient’s health to determine if they will have a stroke in the near future. Although the recall score is high, indicating that over 89% of patients who will have a stroke are flagged by the model, that still means that about 11% of patients who will have a stroke are not picked up by the model. On top of this, the low precision score of

~14% indicates that most patients flagged for a stroke by the model, will not in fact have a stroke in the near future. As such, this model is not a diagnostic test, but can be used as an aid to push for further testing of a patient. This is my recommendation.

For UCLA health to generate a better model, I would advise for an increase in data collection. This data collection would be better if taken at a larger population level, such as national or state, with an increase in the number of initial features. I would advise UCLA health to directly include features such as diabetes level, stroke history in the extended family, general fitness activity, use of other stimulants, and other known correlators of a stroke as determined by organization such as the CDC or WHO. I would also advise for a strict time period on what the near future indicates.

6. Conclusion

In this project, I was initially given a large, unbalanced dataset with a mix of numerical and categorical data. After imputing the missing BMI data entries, I label encoded the categorical features with an implicit ordering and one hot encoded the rest. I then performed an initial logistic regression and determined the unbalanced dataset led to a heavy bias towards predicting the majority class of ‘no stroke’. To counteract this, I used an oversampling of the minority label class of ‘yes stroke’ to generate a balanced training dataset.

This balanced training set was then used to train a series of logistic regression, ensemble, and neural network models. After the logistic regression model, but before training the other models, I used principal component analysis to reduce the feature space from 16 to 6. This reduced feature space was used to train the other models. Each of the models was evaluated with accuracy, precision, recall, and an F1 score. As discussed in the introduction, an emphasis on a high recall score was given priority.

My optimal model regarding recall was a multi-layer perceptron neural network with a hidden layer of 2x9 nodes and an alpha value of 1e9. Although this model had a high recall score (~89), it had a very low precision and average accuracy score. As such, the model serves as one of many tools that should be used to determine further testing for risk of a stroke in the near future. To reiterate, it should not be used as a diagnostic test for determining if a patient will have a stroke. Much more research and long-term data collection is necessary to produce a stroke classification model that could be used as a diagnostic test.

References

https://utswmed.org/medblog/stroke-symptoms-women-risk/

Key information: Women have higher stroke rate than men.

https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb

Key information: Tutorial on balancing the dataset.

https://www.cdc.gov/stroke/facts.htm

Key information: Stroke correlators and background information.