Analysis of Stroke Dataset and Building Logistic Regression and K Means Clustering Models

By Ameya Rajesh Kelaskar
School of Information Studies, Syracuse University

Course: IST 652: Scripting for Data Analysis

Presentation link: https://www.youtube.com/watch?v=bvffoAByfvM

Analysis of Stroke Dataset and Building Logistic Regression and K Means Clustering Models

By Ameya Rajesh Kelaskar
School of Information Studies, Syracuse University

1 Abstract

I have considered the problem of predicting the chances of a patient having a stroke, and for this, I have used healthcare dataset from Kaggle. On this dataset, I have first performed Preprocessing and Visualization, after which I have carried out feature selection. Furthermore, I have built a Logistic Regression model and have performed K-Means clustering. I conclude with the results of my model and insights I gained from performing my analysis.

2 Introduction

Stroke is a leading cause of death in the United States and is a major cause of serious disability for adults. About 795,000 people in the United States have a stroke each year. Therefore, I decided to analyze the stroke dataset and determine whether certain factors can be causes of stroke. This will help us determine if it is possible to predict and even prevent stroke. Therefore, it is a useful problem to analyze in the field of healthcare and will serve as a good reference for the general population as well as medical professionals to pinpoint the factors to look for while dealing with stroke.

3 Data

The data that I have used is from Kaggle, titled, ‘Stroke Prediction Dataset’. It has 11 clinical features (numerical as well as categorical) for predicting stroke events. The features are as follows:

1) id: unique identifier
2) gender: "Male", "Female" or "Other"
3) age: age of the patient
4) hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5) heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6) ever_married: "No" or "Yes"
7) work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
8) Residence_type: "Rural" or "Urban"
9) avg_glucose_level: average glucose level in blood
10) bmi: body mass index
11) smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*
12) stroke: 1 if the patient had a stroke or 0 if not

The dataset has 5110 rows and 11 columns.

4 Data Preprocessing

Data preprocessing is the first step that needs to be performed when as I have raw data at my disposal.

```
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 5110 non-null   int64
 1   gender             5110 non-null   object
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64
 4   heart_disease      5110 non-null   int64
 5   ever_married       5110 non-null   object
 6   work_type          5110 non-null   object
 7   Residence_type     5110 non-null   object
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object
 11  stroke             5110 non-null   int64
dtypes: float64(3), int64(4), object(5)
```

Figure 4.1: Structure of the data

Using the df.info() command in Python suggests that the dataset is fairly decent and does not require much preprocessing. However, there are still null values in the 'bmi' column, which need to be imputed using the mean. Thereafter, the outliers in the 'bmi' column need to be imputed using the median. This makes the data ready for some basic visualizations, which can be used to find some patterns in the data. Also, the 'object' datatypes need to be converted into the 'categorical' format. Finally, I have also checked the number of unique values per column.

5 Visualization

For visualization, I have used the 'matplotlib' and 'seaborn' libraries. Some basic plots on the numerical and categorical data reveal some insights which give an idea of how the data can be correlated. I used the pair plots function in seaborn to check for relationships in the numerical columns

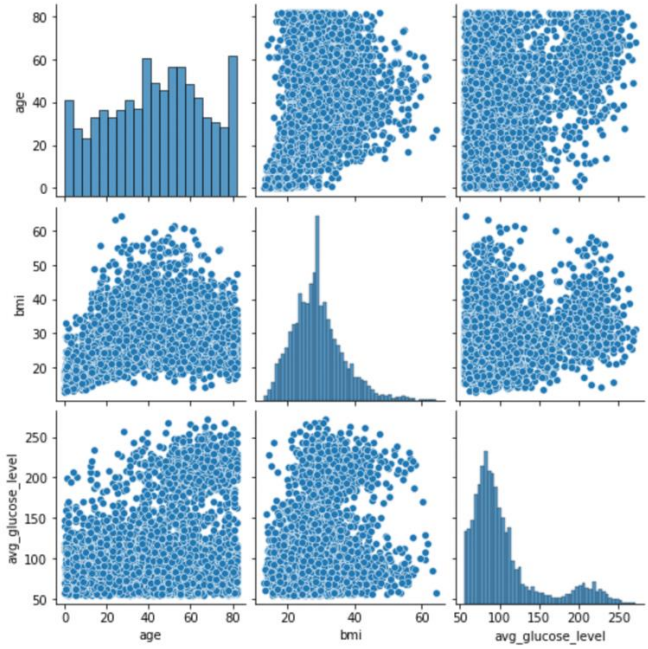

Figure 5.1: Seaborn pairplots

Thereafter, I performed some bivariate analysis by using boxplots:

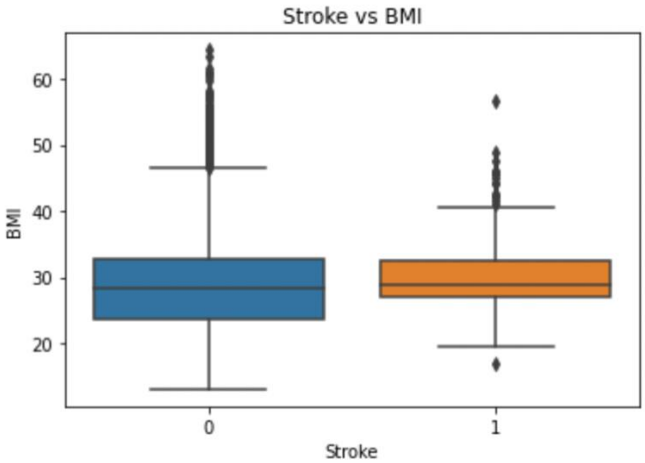

Figure 5.2: Stroke vs BMI

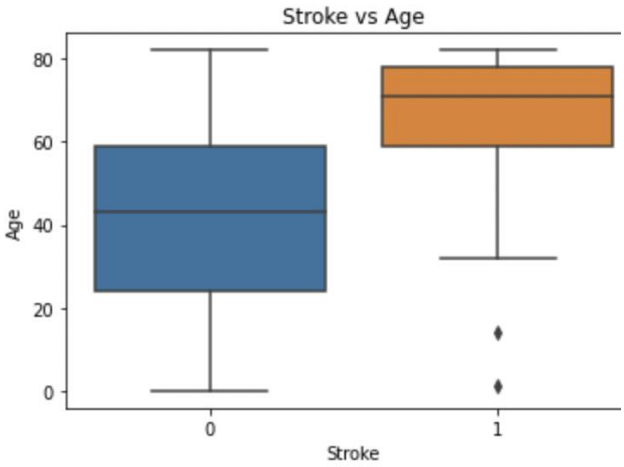

Figure 5.3: Stroke vs Age

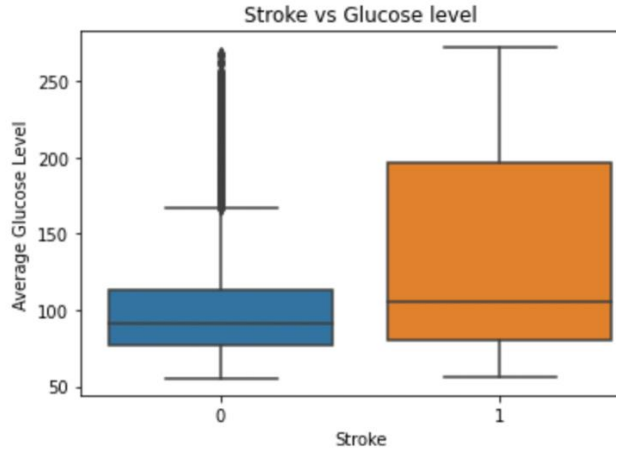

Figure 5.4: Stroke vs Average glucose level

The visualizations suggest that the numerical variables viz. ‘age’, ‘glucose’ and ‘bmi’ play an important role in determining the occurrence of stroke in patients. The patients who had a stroke tend to be older than the patients who did not. The average glucose levels of patients who did not have a stroke is significantly higher than the average glucose levels of the patients who did not. Lastly, the lower range of BMI of patients who had a stroke is greater than that of those who did not.

After this, I plotted a heatmap of the features to determine the correlation between the features, so that I get an idea of how much one feature influences the others. I used the heatmap feature in seaborn for the same.

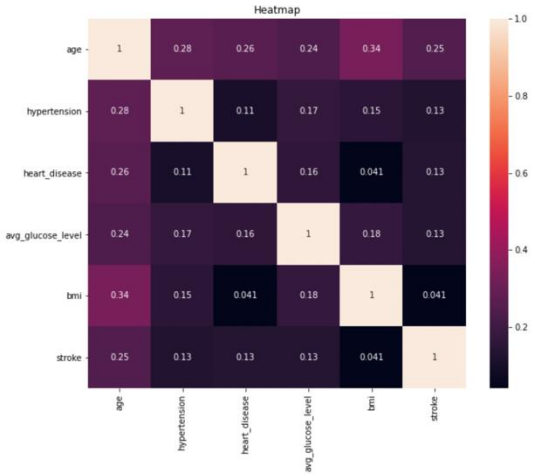

Figure 5.5: Heatmap

After this I looped through the categorical features to determine whether there exists a correlation between them and the occurrence of stroke.

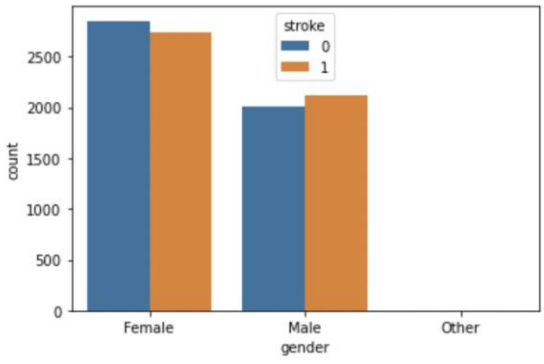

Figure 5.6: Gender count and stroke

This plot suggests that females have had higher number of strokes. However more percentage of males had a stroke.

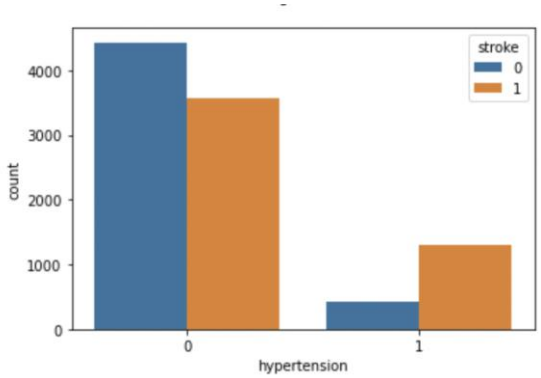

Figure 5.7: Hypertension count and stroke

This plot suggests that hypertension is closely related to stroke. Patients who had hypertension have a significantly greater chance of having a stroke.

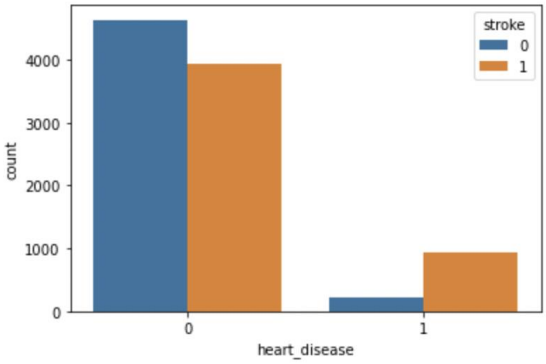

Figure 5.8: Hypertension count and stroke

This plot suggest that heart disease is strongly correlated with stroke. People who have heart disease have more chances of suffering from stroke.

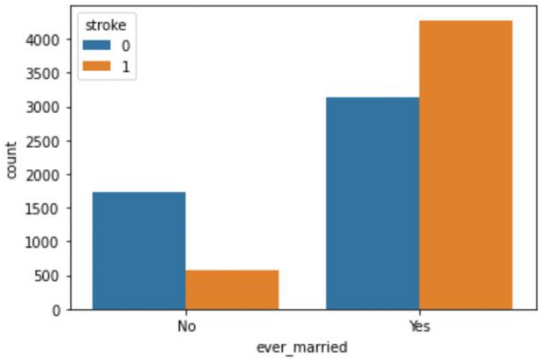

Figure 5.9: Marital status and stroke

This plot suggests that marital status is correlated to stroke. The stroke rate is much higher among married individuals.

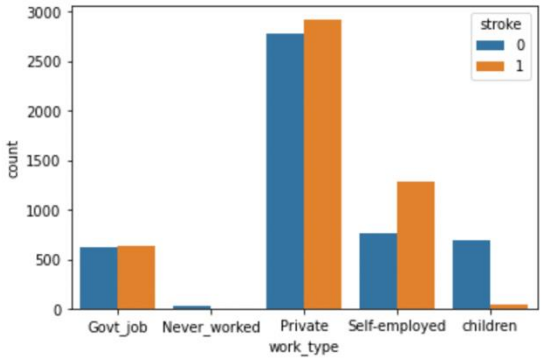

Figure 5.10: Work type and stroke

This plot suggests that occupation is correlated with stroke. Private and self-employed individuals have higher rates of stroke.

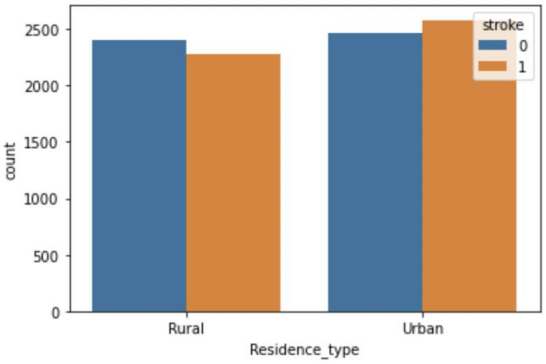

Figure 5.11: Residence type and stroke

Residence type has a slight correlation with stroke possibility, as the rate of stroke among the urban population is slightly higher than that of the rural population.

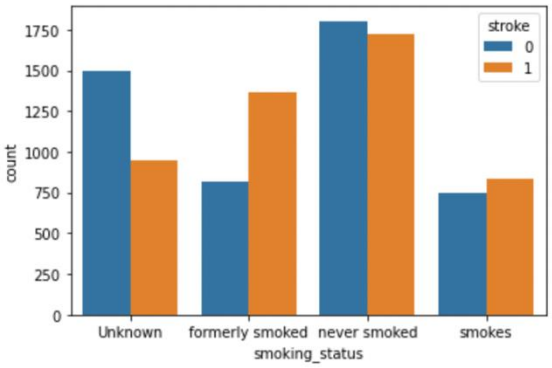

Figure 5.12: Smoking status and stroke

Smoking status has a major impact on stroke occurrence, as people who have previously smoked and the people who are regular smokers have a higher rate of stroke occurrence than those who did not and those whose status is unknown.

6 Imputing outliers in BMI using median

To impute the outliers, I used the percentile method and imputed all the values above 99.9th percentile and below 0.1st percentile.

7 Checking if the 'stroke' variable is balanced

I visualized a count plot to determine the balance of the stroke variable.

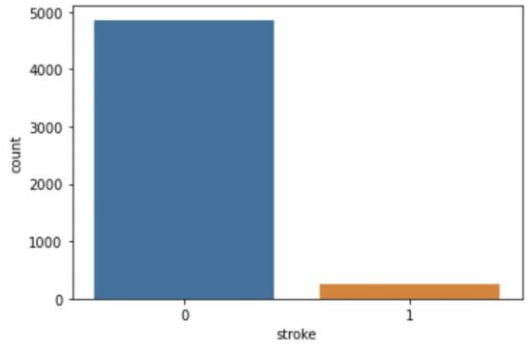

Figure 7.1: Before balancing the stroke variable

Clearly, the data is highly imbalanced. Therefore, I had to balance this, for which I used the over-sampling technique. In over-sampling, I create more random records which will have stroke=1. This will increase the accuracy during prediction. The balanced variable looks like the following:

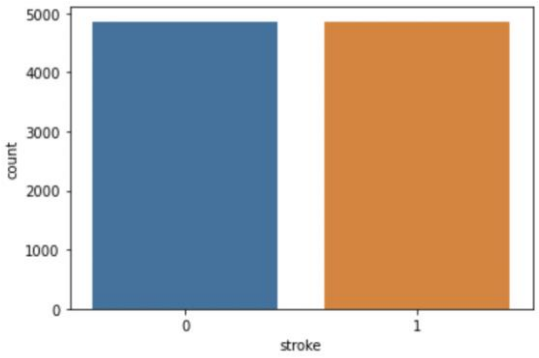

Figure 7.2: After balancing the stroke variable

Now, the variable has been balanced, as there are equal instances of stroke=1 and stroke=0. This will ensure high metrics in the Logistic Regression model.

8 Feature Selection

Feature selection is a necessary step to determine which columns will help us in modelling and which columns will be of no use in the process. Before running the code for feature importance, I had to convert the categorical features into numerical values. For this, I used the 'get_dummies' function in pandas.

After converting all features into numeric values, I used a Decision Tree classifier to calculate the feature importance of all the features and sorted them in descending order.

	index	Feature	Feature Importance
0	0	age	0.479466
1	1	avg_glucose_level	0.218601
2	2	bmi	0.193520
3	4	gender_Male	0.019891
4	21	smoking_status_never smoked	0.015620
5	19	smoking_status_Unknown	0.012753
6	12	work_type_Govt_job	0.011070
7	14	work_type_Private	0.008495
8	11	ever_married_Yes	0.007801
9	20	smoking_status_formerly smoked	0.006987
10	9	heart_disease_1	0.006627
11	15	work_type_Self-employed	0.005558
12	6	hypertension_0	0.005045
13	22	smoking_status_smokes	0.003699
14	16	work_type_children	0.002925
15	18	Residence_type_Urban	0.001151
16	3	gender_Female	0.000791
17	10	ever_married_No	0.000000
18	8	heart_disease_0	0.000000
19	7	hypertension_1	0.000000
20	13	work_type_Never_worked	0.000000
21	17	Residence_type_Rural	0.000000
22	5	gender_Other	0.000000

Figure 8.1: Feature Importance

I decided to eliminate all features that have zero importance, as they will not be of any use in the modelling process.

9 Separating the data into X and y, creating a train-test split and a Hold-out sample

I separated the data into X (for independent variables) and y (for the dependent variable). For creating the training-test split, I used the 'train_test_split' feature in sklearn. My training data was 80%, testing data was 20% and the Hold-out sample was 10%. This hold-out sample is data that will be unseen by the model, which we will use to determine the model performance at the end.

10 Creating the Logistic Regression Model and checking the metrics for evaluation

The variable that I am predicting is binary, therefore I needed to use a classification algorithm, therefore I selected the Logistic regression model, as it helps to classify binary data. I used sklearn to implement this model.

I used accuracy for the training and test data and the classification report as metrics to evaluate my model.

```
The Training Accuracy is:  0.778396913844835
The test Accuracy is:  0.7722365038560411
              precision    recall  f1-score   support

           0       0.80      0.74      0.77      3504
           1       0.76      0.82      0.79      3495

    accuracy                           0.78      6999
   macro avg       0.78      0.78      0.78      6999
weighted avg       0.78      0.78      0.78      6999
```

Figure 10.1: Performance metrics

All the metrics are decent, which indicate that it is a good model. After this, I plotted a confusion matrix for further evaluation.

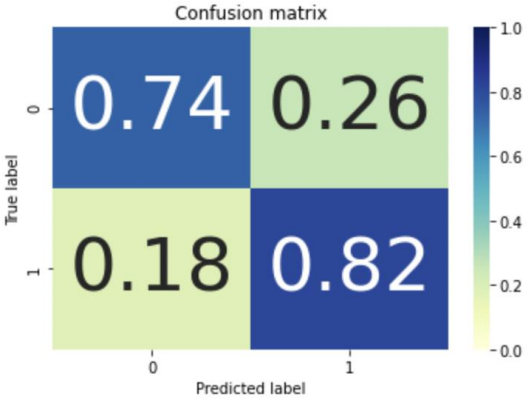

Figure 10.2: Confusion matrix

The confusion matrix indicates that the model predicted 0's correctly with 74% accuracy and predicted 1's correctly with 82% accuracy.

Thereafter, I calculated the true positive rate, precision, false positive rate, false negative rate and the average of these rates.

```
The True Positive Rate is: [0.73829909 0.818598  ]
The Precision is: [0.80316672 0.75727898]
The False positive rate is: [0.181402   0.26170091]
The False Negative Rate is:  [0.26170091 0.181402  ]

The average TPR is: 0.7784485419483802
The average Precision is: 0.7802228509998161
The average False positive rate is: 0.22155145805161972
The average False Negative Rate is: 0.22155145805161972
```

Figure 10.3: Various rates

After this, I used a dummy classifier to check whether the model performs better than it. The dummy classifier gave an accuracy of 50.12%, which indicates that

our model greatly outperforms the dummy classifier. Finally, I tested the model on a hold-out sample and got an accuracy of 78.4%.

11 K-means Clustering

K-means is an unsupervised machine learning algorithm. In this algorithm, there is no variable to be predicted, rather this model is used to form clusters when raw data is given as input. An important metric used to evaluate the model performance and to determine the ideal number of clusters is 'inertia', which is the variance per the number of Ks.

After creating the training data, I calculated the variance for the number of Ks in the range (2,20).

```
The inertia for 2 Clusters is:  9612998.80058299
The inertia for 3 Clusters is:  6714886.228957671
The inertia for 4 Clusters is:  5029679.228089355
The inertia for 5 Clusters is:  4148915.21153538
The inertia for 6 Clusters is:  3514391.647002897
The inertia for 7 Clusters is:  3055719.7761602225
The inertia for 8 Clusters is:  2696843.7758025406
The inertia for 9 Clusters is:  2452667.762972195
The inertia for 10 Clusters is:  2246538.787130049
The inertia for 11 Clusters is:  2095059.9462378705
The inertia for 12 Clusters is:  1957661.684295487
The inertia for 13 Clusters is:  1794048.0119114995
The inertia for 14 Clusters is:  1692792.051500854
The inertia for 15 Clusters is:  1585686.9797906275
The inertia for 16 Clusters is:  1524820.0626054637
The inertia for 17 Clusters is:  1454541.083568749
The inertia for 18 Clusters is:  1384879.865778488
The inertia for 19 Clusters is:  1339076.0178947242
```

Figure 11.1: Inertia values

After this, I plotted the inertia per K to determine the ideal K value using the elbow technique.

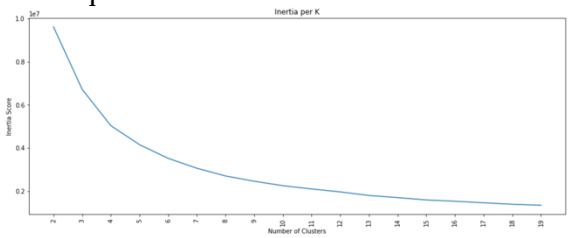

Figure 11.2: Elbow method

I decided to form 5 clusters after inspecting the above plot. I stored the cluster counts in a new dataframe. Thus, the five clusters were created as follows.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
0	2282	1246	3054	1149	1991

Figure 11.3: Clusters

I could not determine how to interpret these clusters, and it was not possible to visualize them because the number of features was too high. Therefore, I tried the process again with only two features to check whether I get any insights from the resulting visualizations First, I used 'age' and 'bmi' to create clusters. Using the elbow method, the ideal K value was 4.

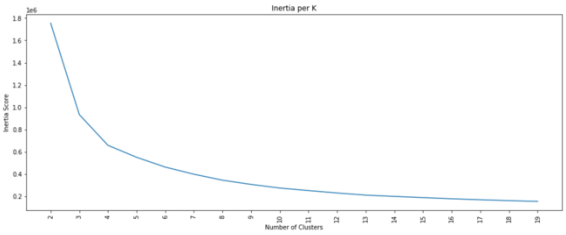

Figure 11.4: Elbow method

The visualization of these clusters against the age and bmi columns was as follows:

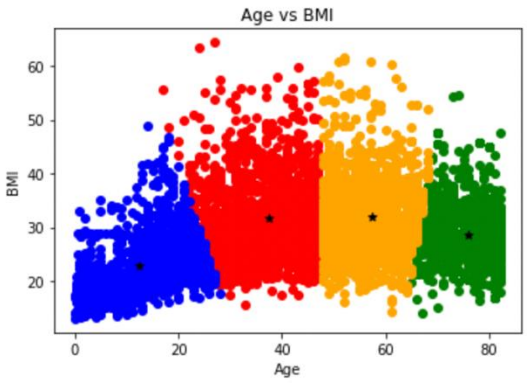

Fig 11.5: Age vs BMI

The above plot indicates that people with high BMI levels are concentrated in the age range of approximately 30-60 years.

I followed the same process for age and average glucose levels, and got the following cluster visualization:

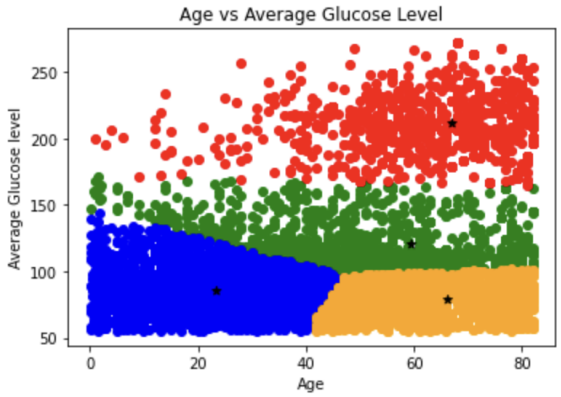

Fig 11.6: Age vs average glucose level

The above plot indicates that the population with high glucose levels is largely concentrated among the higher age group of 40 years and above, while the younger population (under 40) generally has low to moderate glucose levels. I performed the same calculation using BMI and average glucose level:

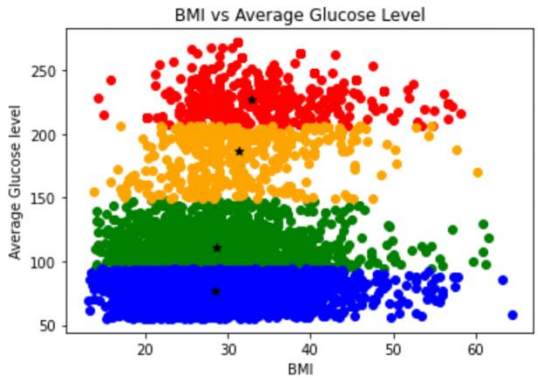

Fig 11.7: BMI vs average glucose level

The plot suggests that the people who have a high glucose level also tend to have a higher BMI.

12 Conclusion

From the logistic regression model and K-means clustering, I was able to confirm the link between the features, determine the importance of each feature with regards to determining the possibility of stroke. The quality of the logistic regression model was good, as indicated by the performance metrics and the cluster sizes for the K-means regression algorithm were ideal as determined by the elbow method. In conclusion, it is safe to ascertain that stroke can be predicted if we are given the features assessed above. Moreover, stroke can also be prevented by controlling factors like average glucose levels and BMI in patients.

13 References

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

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

https://numpy.org/doc/

https://pandas.pydata.org/pandas-docs/stable/

https://matplotlib.org/stable/contents.html

https://seaborn.pydata.org

https://scikit-learn.org/stable/index.html

https://www.youtube.com/watch?v=bvffoAByfvM