Using various Data Mining techniques on Stroke Prediction Dataset

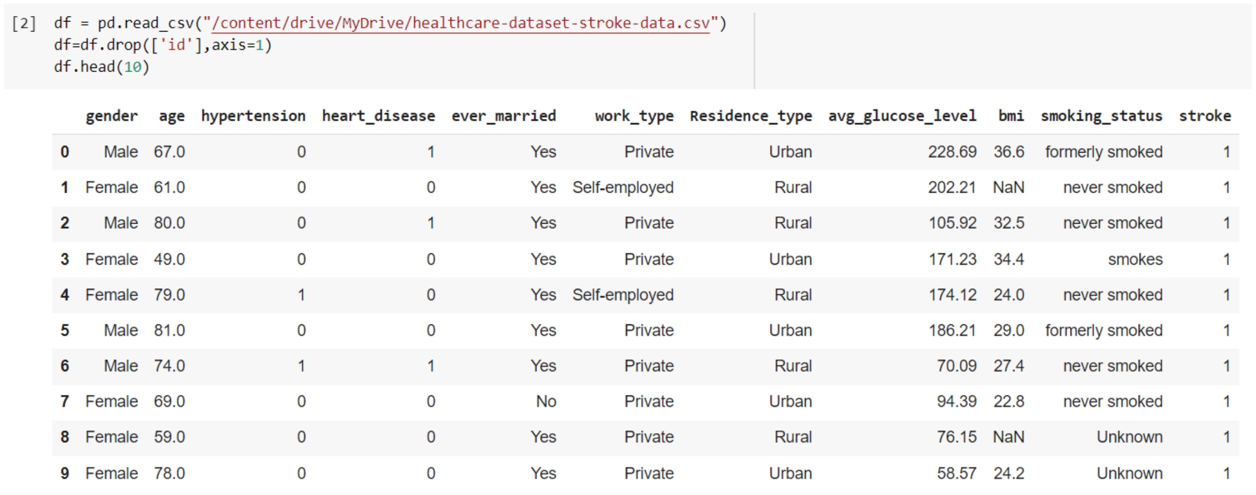
By Nenad Kajgana

About the project

In this project I will use the techniques used in the Data Mining/Science field(Preprocessing, EDA, creating models, evaluating) on a dataset containing medical data. I will use multiple classification, regression and clustering models with explanation on the pros and cons of those models and what models would be the most suitable for this dataset. This projects main focus will be classification, but techniques of regression and clustering will also be used as a demonstration.

About the dataset

The stroke prediction dataset contains numerous columns about a person, mainly health data and an end column for whether the person had a stroke or not. According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. The columns which the dataset contains are id, age, hypertension, heart_disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, smoking_status and out Y column for which we will predict stroke. For the regression part of the project the bmi column will be used. All of the steps will be centered around the stroke column and how the other columns depend on it. The dataset contains 5110 rows, only 249 of them are instances which have had a stroke.



```
[2] df = pd.read_csv("/content/drive/MyDrive/healthcare-dataset-stroke-data.csv")
    df=df.drop(['id'],axis=1)
    df.head(10)
```

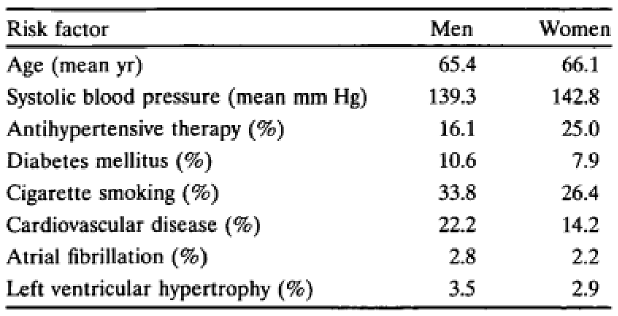
	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1
5	Male	81.0	0	0	Yes	Private	Urban	186.21	29.0	formerly smoked	1
6	Male	74.0	1	1	Yes	Private	Rural	70.09	27.4	never smoked	1
7	Female	69.0	0	0	No	Private	Urban	94.39	22.8	never smoked	1
8	Female	59.0	0	0	Yes	Private	Rural	76.15	NaN	Unknown	1
9	Female	78.0	0	0	Yes	Private	Urban	58.57	24.2	Unknown	1

10 instances of the dataset(Initial look)

Initial thoughts in the dataset

By doing some basic research for the area I have found the following information:

- People are most likely to have a stroke between the ages of 55 and 85
- Men have a higher risk of having a stroke than women
- Hypertension is the most potent risk factor for a stroke, high blood pressure is the main factor
- Cigarette smoking increases the chances two-fold, smoking also increases blood pressure.
- Heart disease is a factor
- High bmi leads to high blood pressure and that leads to having a stroke as mentioned before
- Glucose level >6.0 mmol/L (108 mg/dL), has been observed in two thirds of all ischemic stroke subtypes on admission and in at least 50% in each subtype including lacunar strokes.

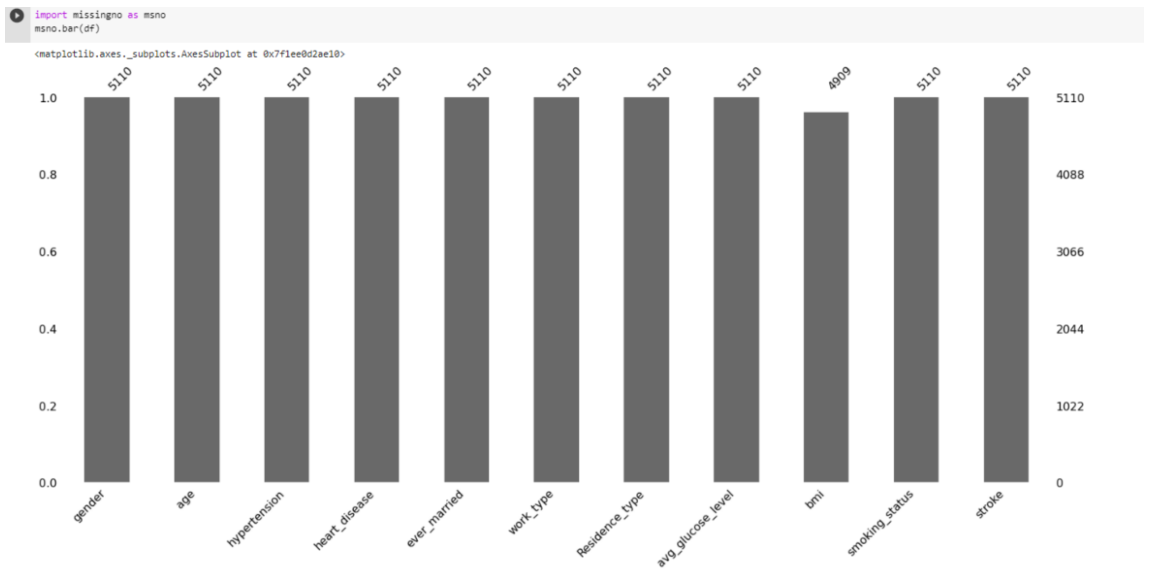


Risk factor	Men	Women
Age (mean yr)	65.4	66.1
Systolic blood pressure (mean mm Hg)	139.3	142.8
Antihypertensive therapy (%)	16.1	25.0
Diabetes mellitus (%)	10.6	7.9
Cigarette smoking (%)	33.8	26.4
Cardiovascular disease (%)	22.2	14.2
Atrial fibrillation (%)	2.8	2.2
Left ventricular hypertrophy (%)	3.5	2.9

By doing analysis on our dataset we will see if the data matches the overall analysis.

1. Missing values

We will use the missingno library for this part. By initial command we can see that only the column bmi has missing values.

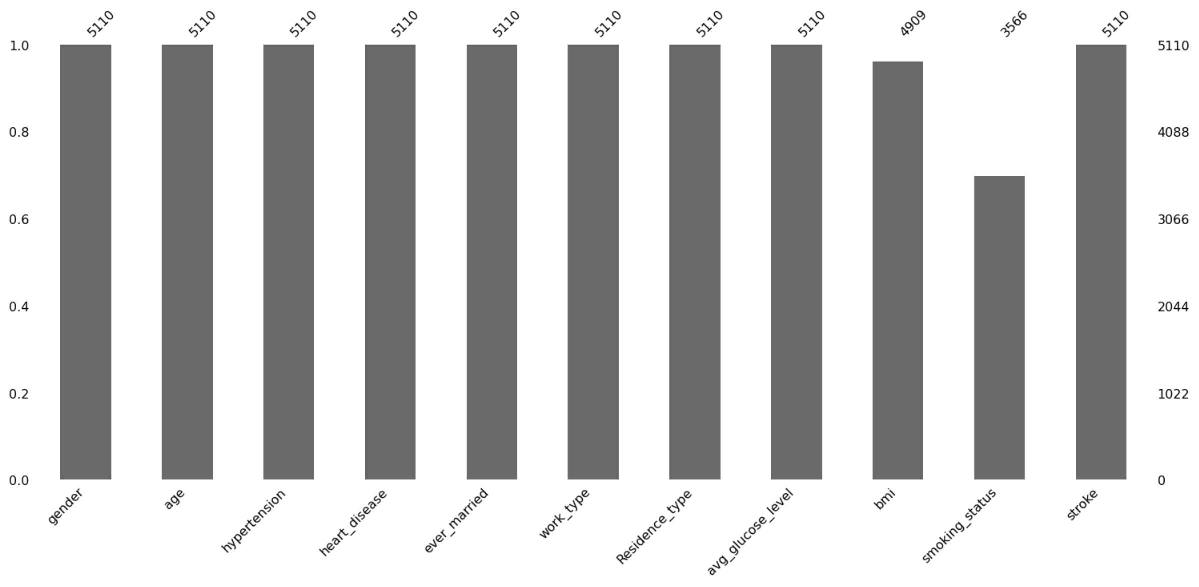


```
import missingno as msno
msno.bar(df)
```

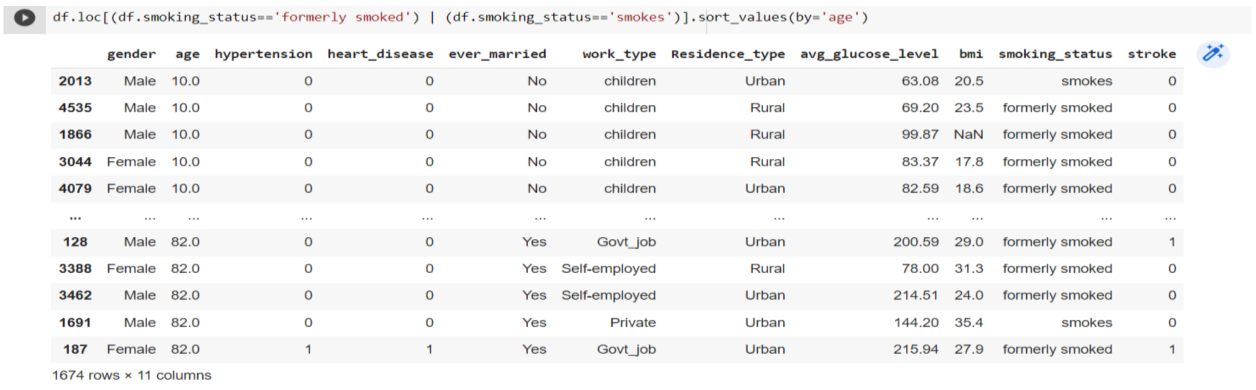
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1ee0d2ae10>
```

The column only has 201 missing values, which is 0.04% of the dataset. I have 2 approaches for this. One approach would be to drop these values when undersampling the dataset(because of the imbalance of the instances that have a stroke) and the other way of solving this problem would be to use a regression model for imputing these values. We will use both techniques for this project and all the regression techniques will be used on those 201 bmi missing values.

There is another problem. That's the smoking_status column. It contains values which are 'Unknown'. By replacing the Unknown with the None we get this bar graph for missing values.



30% of the values in this dataset have the ‘Unknown’ value. While we could just not use these values when undersampling, a small analysis will be performed on the instances with that value. By making a temporary dataframe I was able to get this results.

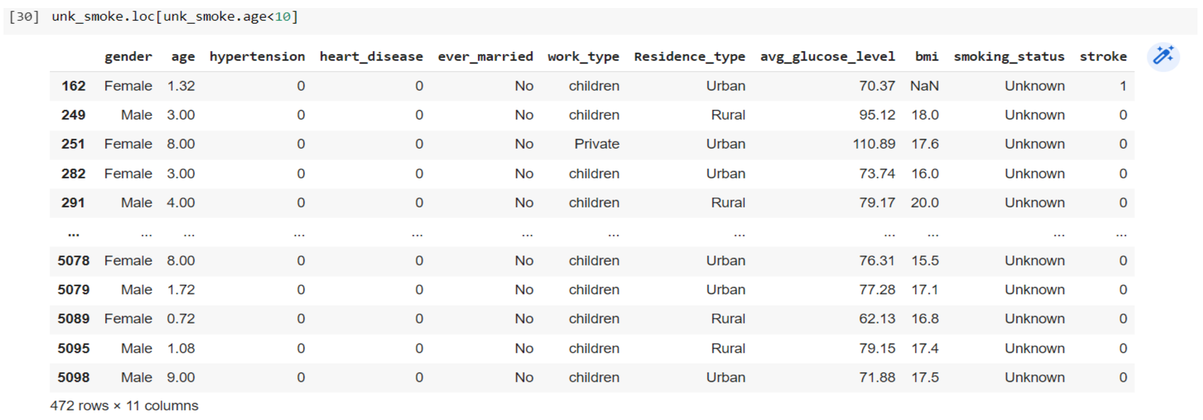


```
df.loc[(df.smoking_status=='formerly smoked') | (df.smoking_status=='smokes')].sort_values(by='age')
```

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
2013	Male	10.0	0	0	No	children	Urban	63.08	20.5	smokes	0
4535	Male	10.0	0	0	No	children	Rural	69.20	23.5	formerly smoked	0
1866	Male	10.0	0	0	No	children	Rural	99.87	NaN	formerly smoked	0
3044	Female	10.0	0	0	No	children	Rural	83.37	17.8	formerly smoked	0
4079	Female	10.0	0	0	No	children	Urban	82.59	18.6	formerly smoked	0
...	...	...	...	...	...	...	...	...	...	...	...
128	Male	82.0	0	0	Yes	Govt_job	Urban	200.59	29.0	formerly smoked	1
3388	Female	82.0	0	0	Yes	Self-employed	Rural	78.00	31.3	formerly smoked	0
3462	Male	82.0	0	0	Yes	Self-employed	Urban	214.51	24.0	formerly smoked	0
1691	Male	82.0	0	0	Yes	Private	Urban	144.20	35.4	smokes	0
187	Female	82.0	1	1	Yes	Govt_job	Urban	215.94	27.9	formerly smoked	1

1674 rows × 11 columns

By looking in the initial dataframe we can see that no person under the age of 10 has smoked or smokes.

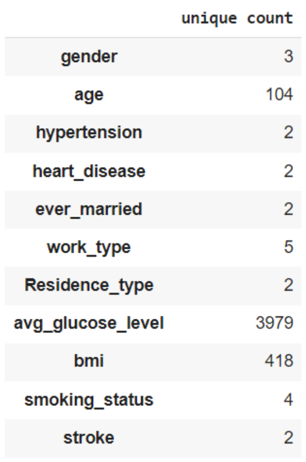


```
[30] unk_smoke.loc[unk_smoke.age<10]
```

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
162	Female	1.32	0	0	No	children	Urban	70.37	NaN	Unknown	1
249	Male	3.00	0	0	No	children	Rural	95.12	18.0	Unknown	0
251	Female	8.00	0	0	No	Private	Urban	110.89	17.6	Unknown	0
282	Female	3.00	0	0	No	children	Urban	73.74	16.0	Unknown	0
291	Male	4.00	0	0	No	children	Rural	79.17	20.0	Unknown	0
...	...	...	...	...	...	...	...	...	...	...	...
5078	Female	8.00	0	0	No	children	Urban	76.31	15.5	Unknown	0
5079	Male	1.72	0	0	No	children	Urban	77.28	17.1	Unknown	0
5089	Female	0.72	0	0	No	children	Rural	62.13	16.8	Unknown	0
5095	Male	1.08	0	0	No	children	Rural	79.15	17.4	Unknown	0
5098	Male	9.00	0	0	No	children	Urban	71.88	17.5	Unknown	0

472 rows × 11 columns

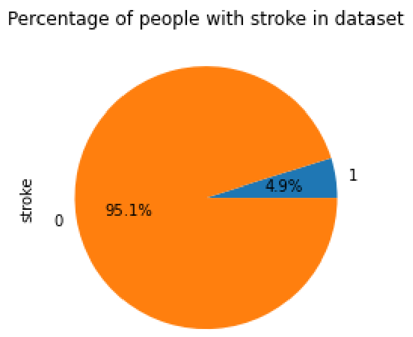
In the dataframe containing the instances of people that have the ‘Unknown’ there are 497 rows that have an age under 10. We can safely impute these values to never smoked and that would leave us with 1047 rows. This lowers the missing values by 10%. The other ‘Unknown’ values we will leave them as they are. They will be not used in the modeling phase, but if we were to use a model for them on the whole dataset, we could count them as a categorical value for the column.



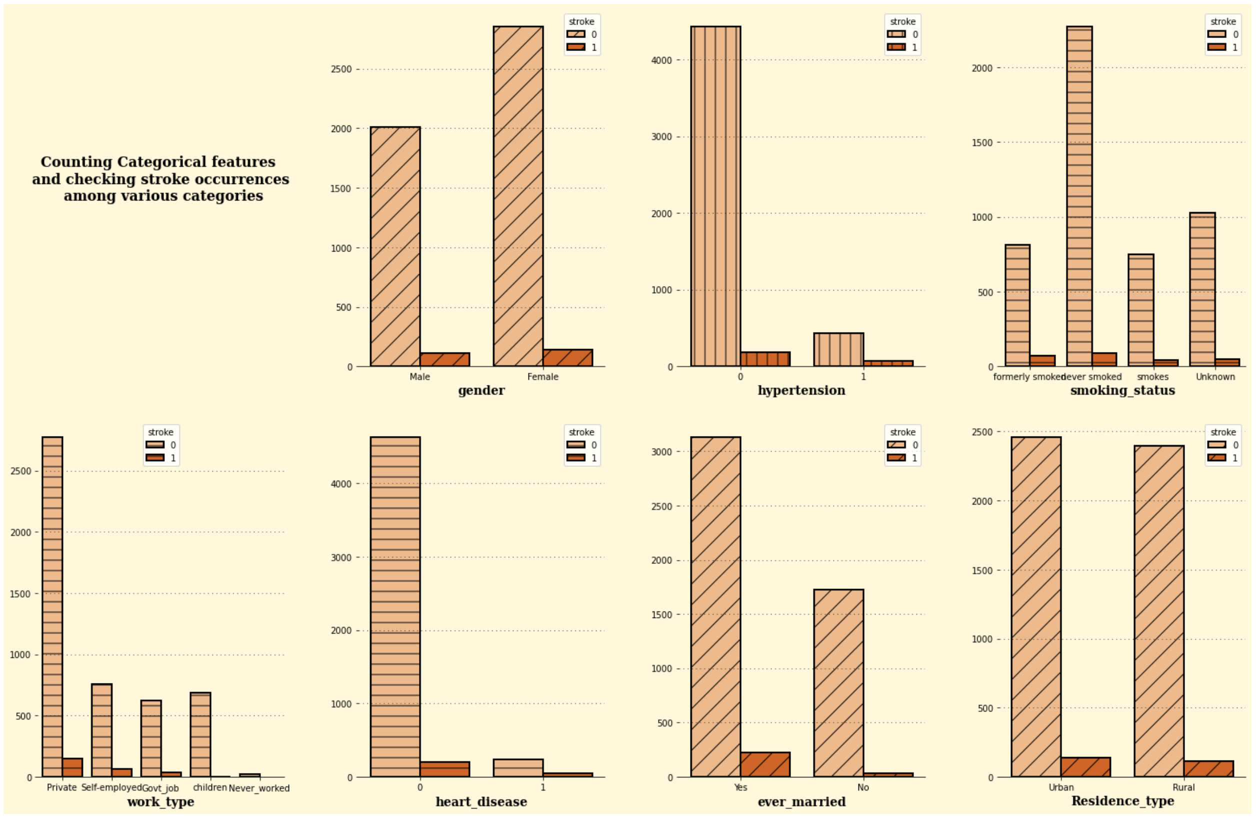
	unique count
gender	3
age	104
hypertension	2
heart_disease	2
ever_married	2
work_type	5
Residence_type	2
avg_glucose_level	3979
bmi	418
smoking_status	4
stroke	2

By checking out the unique values in the dataset we can see that all are okay except for the gender column. It has 1 instance with value ‘Other’. As it is one we can simply remove it. Also the age column will be floored to an integer.

2. EDA

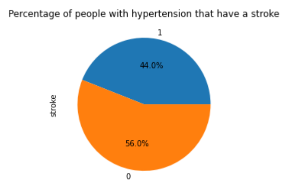


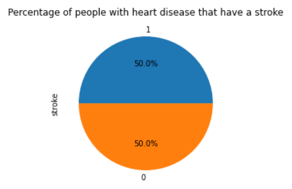
As stated less than 5% of the instances have the value 1 in the stroke column. By the world stroke organization 1 in 4 people have experienced a stroke in their lifetime.

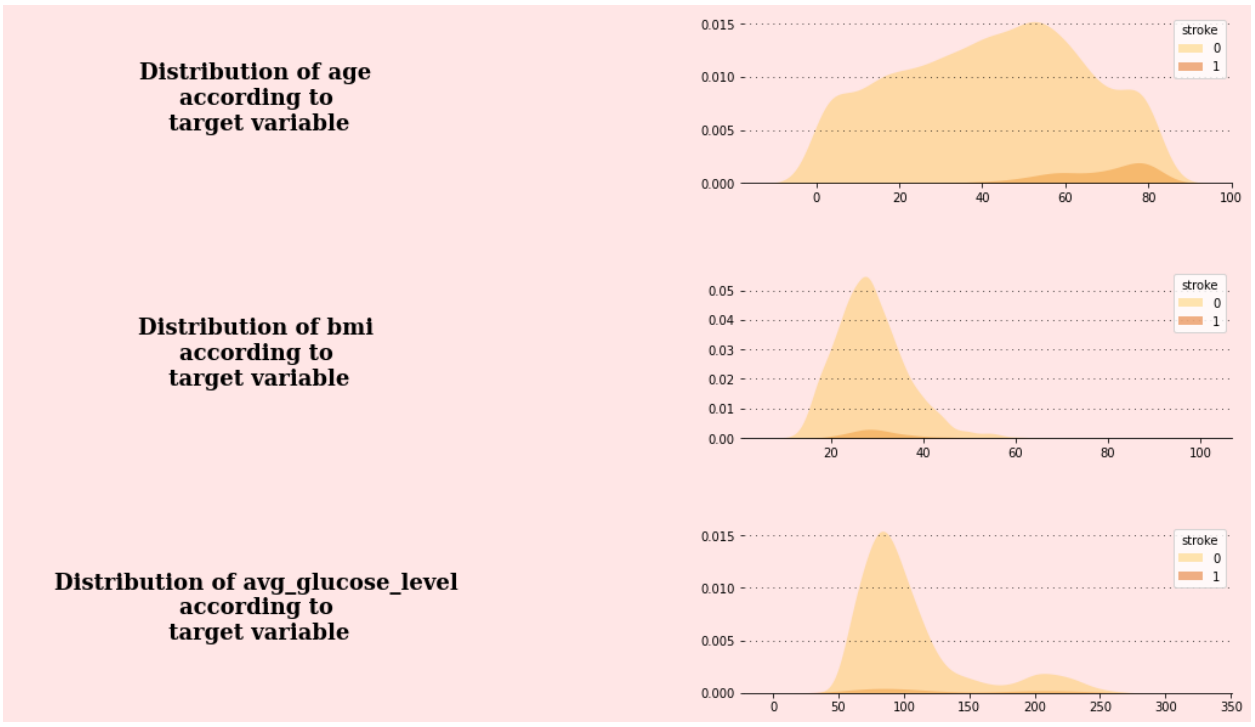


As we can see by these graphs we can draw the following conclusions:

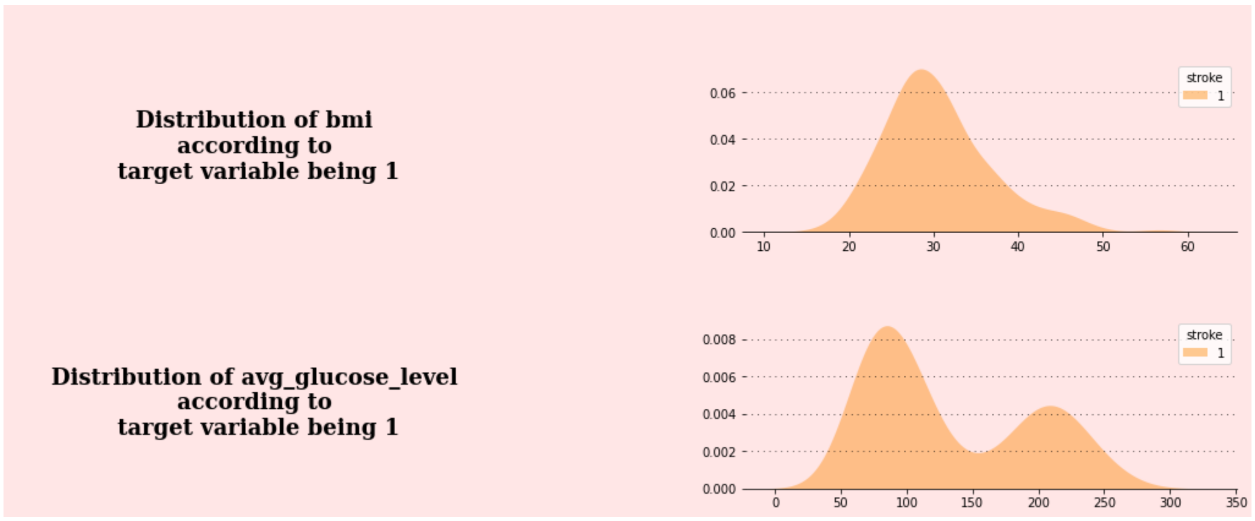
- Around 20% of both female and male instances have had a stroke
- Hypertension usually leads to a stroke as per research, this is true as in our dataset around 44% of the people that have hypertension also have experienced a stroke. Same with heart disease.
- People who are married are more likely to had a stroke in our dataset
- Private jobs have the most stroke occurrences and smoking does not have any factor here, although smoking is a big factor per research.



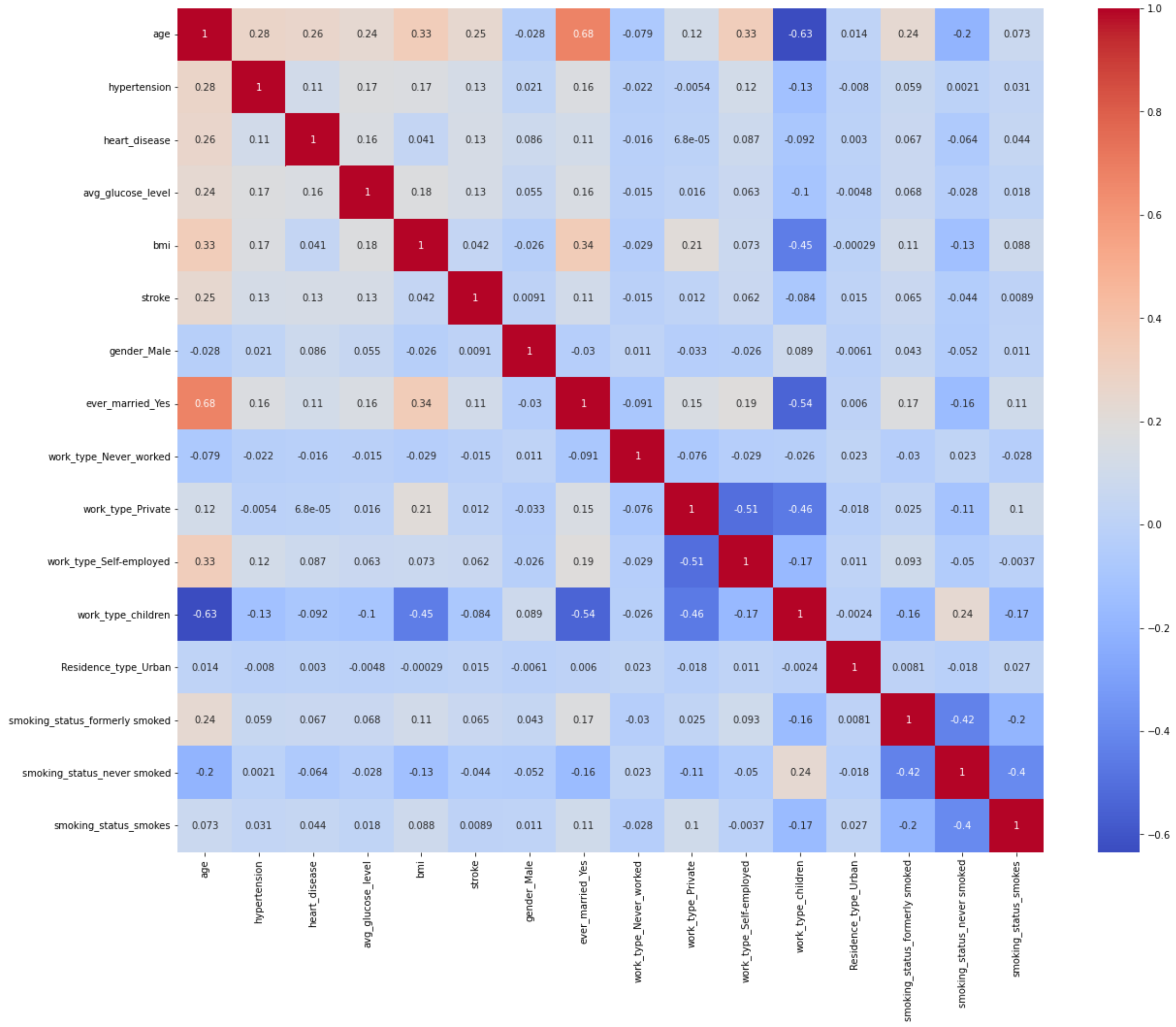




As we can see the distribution of people that had a stroke is the same as the initial assumptions.

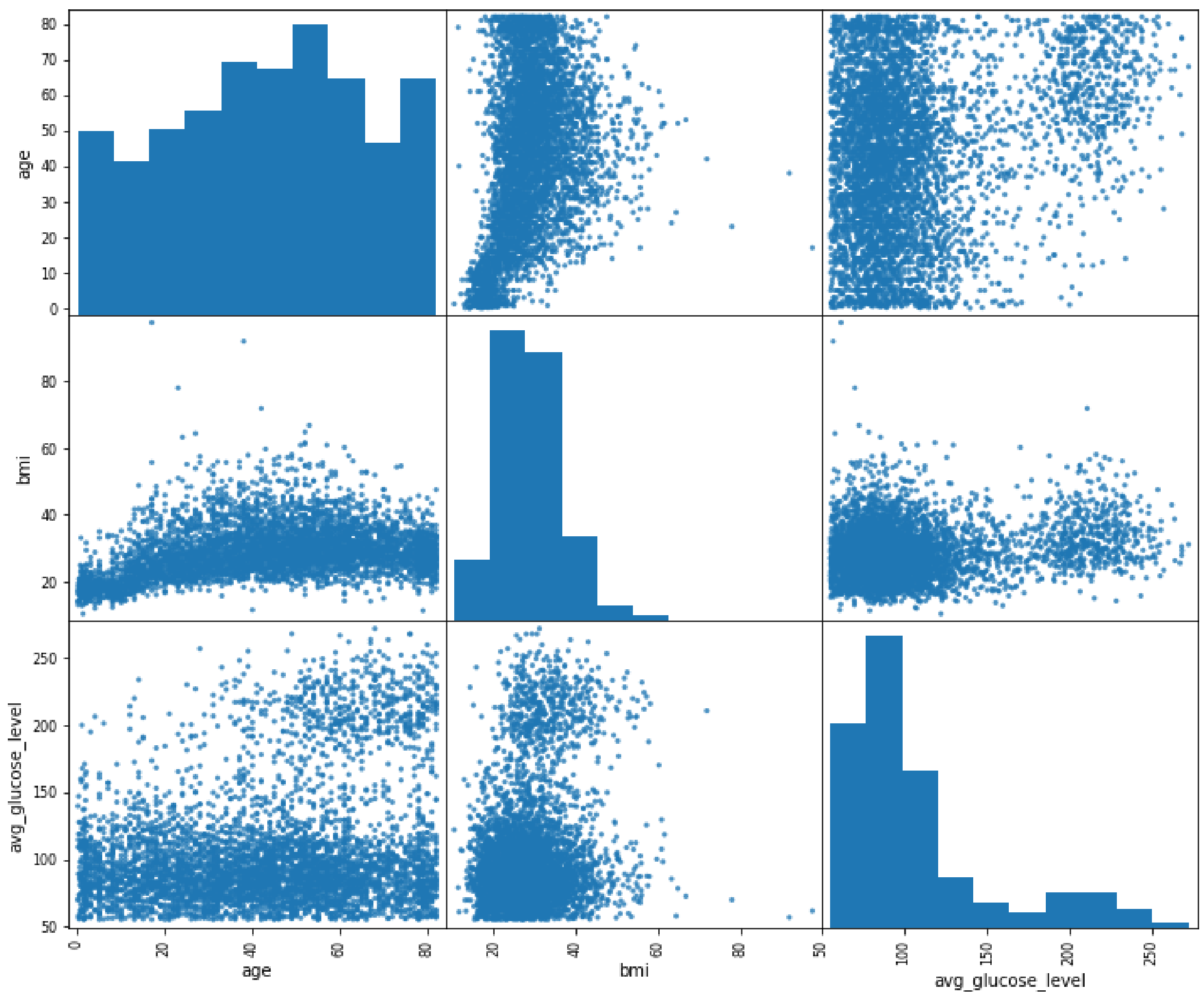


Here in these we can get a better view of the distribution of the instances where there had been a stroke. Obesity is common with stroke, but here the data distribution is normal and by these 2 images we can see that there isn't much significant peaks or information that we could draw conclusions from.

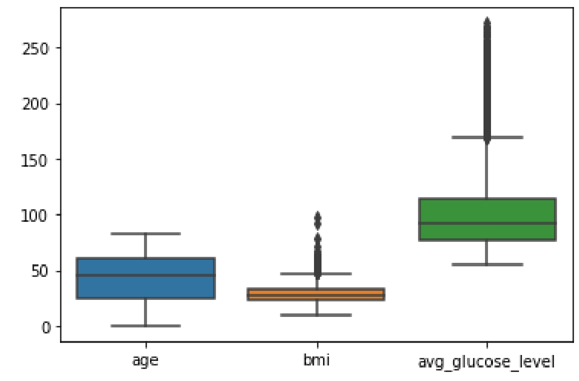


	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke	gender_Male	ever_married_Yes	work_type_Never_worked	work_type_Private	work_type_Self-employed	work_type_children	Residence_type_Urban	smoking_status_formerly smoked	smoking_status_never smoked	smoking_status_smokes
age	1	0.28	0.26	0.24	0.33	0.25	-0.028	0.68	-0.079	0.12	0.33	-0.63	0.014	0.24	-0.2	0.073
hypertension	0.28	1	0.11	0.17	0.17	0.13	0.021	0.16	-0.022	-0.0054	0.12	-0.13	-0.008	0.059	0.0021	0.031
heart_disease	0.26	0.11	1	0.16	0.041	0.13	0.086	0.11	-0.016	6.8e-05	0.087	-0.092	0.003	0.067	-0.064	0.044
avg_glucose_level	0.24	0.17	0.16	1	0.18	0.13	0.055	0.16	-0.015	0.016	0.063	-0.1	-0.0048	0.068	-0.028	0.018
bmi	0.33	0.17	0.041	0.18	1	0.042	-0.026	0.34	-0.029	0.21	0.073	-0.45	-0.00029	0.11	-0.13	0.088
stroke	0.25	0.13	0.13	0.13	0.042	1	0.0091	0.11	-0.015	0.012	0.062	-0.084	0.015	0.065	-0.044	0.0089
gender_Male	-0.028	0.021	0.086	0.055	-0.026	0.0091	1	-0.03	0.011	-0.033	-0.026	0.089	-0.0061	0.043	-0.052	0.011
ever_married_Yes	0.68	0.16	0.11	0.16	0.34	0.11	-0.03	1	-0.091	0.15	0.19	-0.54	0.006	0.17	-0.16	0.11
work_type_Never_worked	-0.079	-0.022	-0.016	-0.015	-0.029	-0.015	0.011	-0.091	1	-0.076	-0.029	-0.026	0.023	-0.03	0.023	-0.028
work_type_Private	0.12	-0.0054	6.8e-05	0.016	0.21	0.012	-0.033	0.15	-0.076	1	-0.51	-0.46	-0.018	0.025	-0.11	0.1
work_type_Self-employed	0.33	0.12	0.087	0.063	0.073	0.062	-0.026	0.19	-0.029	-0.51	1	-0.17	0.011	0.093	-0.05	-0.0037
work_type_children	-0.63	-0.13	-0.092	-0.1	-0.45	-0.084	0.089	-0.54	-0.026	-0.46	-0.17	1	-0.0024	-0.16	0.24	-0.17
Residence_type_Urban	0.014	-0.008	0.003	-0.0048	-0.00029	0.015	-0.0061	0.006	0.023	-0.018	0.011	-0.0024	1	0.0081	-0.018	0.027
smoking_status_formerly smoked	0.24	0.059	0.067	0.068	0.11	0.065	0.043	0.17	-0.03	0.025	0.093	-0.16	0.0081	1	-0.42	-0.2
smoking_status_never smoked	-0.2	0.0021	-0.064	-0.028	-0.13	-0.044	-0.052	-0.16	0.023	-0.11	-0.05	0.24	-0.018	-0.42	1	-0.4
smoking_status_smokes	0.073	0.031	0.044	0.018	0.088	0.0089	0.011	0.11	-0.028	0.1	-0.0037	-0.17	0.027	-0.2	-0.4	1

A simple pairplot using pearson coefficient with all categorical types. Methods like spearman and kendall gave similar results. No unusual correlations here. High results here are completely normal as they make sense. This is a good sign for our data and is a good pointer it is genuine.



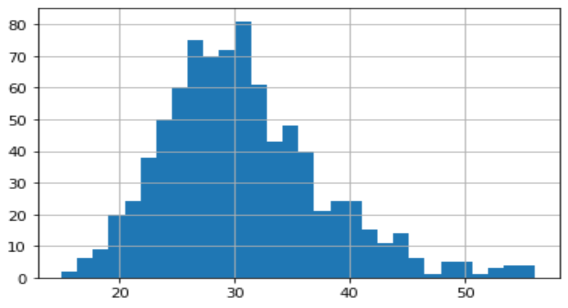
Here is a scatter matrix plot of the continuous variables in our dataset. BMI has somewhat a linear correlation with age and that is evident because BMI does increase with age and decrease in the latter years, there are a few points that could be outliers, but I think those are just instances where the person has a very high BMI. The other thing we could talk about is the correlation between age and glucose level. There are 2 clusters which can be observed, one of them is normal and the other appears to have the majority of points in the 55-80 age mark.

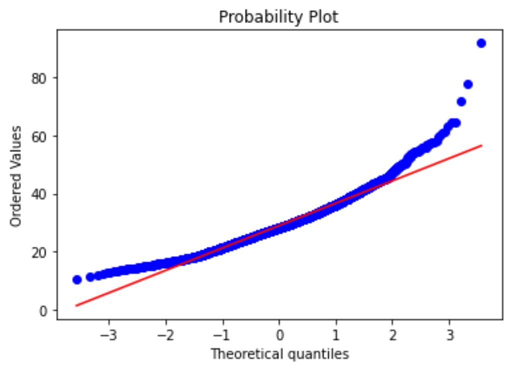


In this graph on the right we can see the outliers which are totally normal given the properties of the nature of the variables. Such instances with these high values are present in people that are obese, suffer from diabetes and have other related diseases.

3. Linear Regression model for filling out BMI missing values

Before we start, here is information about the distribution of the BMI column.





In these two graphs we can see that BMI follows a Gaussian distribution. This is a good sign as we can get a good model.

For feature selection we will use all the columns, preprocess the "Unknown" values of smoking_status as before. And will try out a few models on a dataset containing BMI that doesn't have null values. Then later we will use the best model, trained on the whole dataset where BMI is not null and fill out the missing data on the other dataset with the null values, then combine both for the main classification problem.

Other noticeable steps in the preprocessing includes the encoding and scaling part. Trying out encoders like Ordinal, OneHot and Label I've had similar results and ended up going with the basic LabelEncoder from sklearn. Because most of the columns have binary outputs, I will talk about only 2 columns in terms of the encoding. There hasn't been much research done on type of work associated with BMI so the label encoder will encode them by default. The smoking status is an interesting one, by doing research I have found out that people who gave up smoking usually have a higher BMI, smokers usually have a lower BMI and people who have never smoked come in the middle.

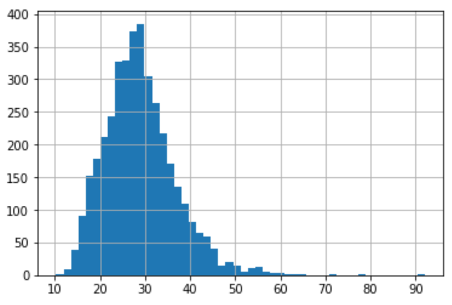


Figure 1:never smoked

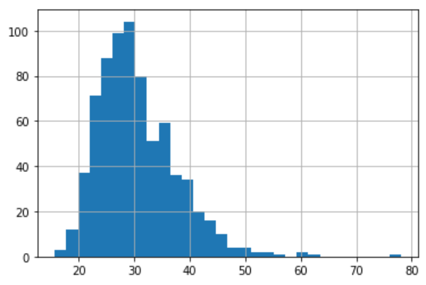


Figure 3:smokes

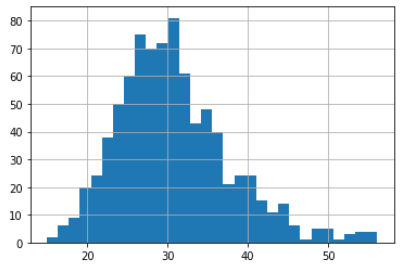
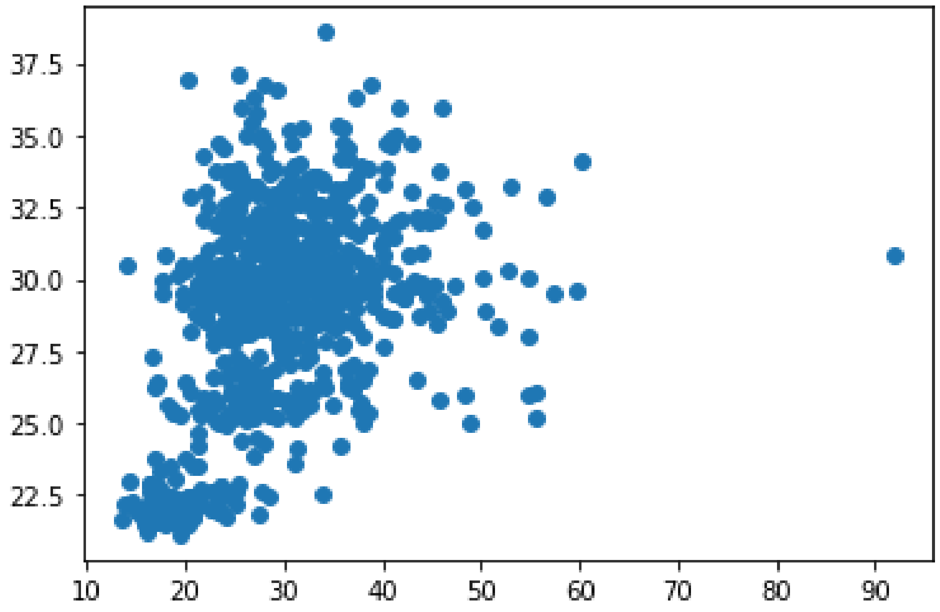
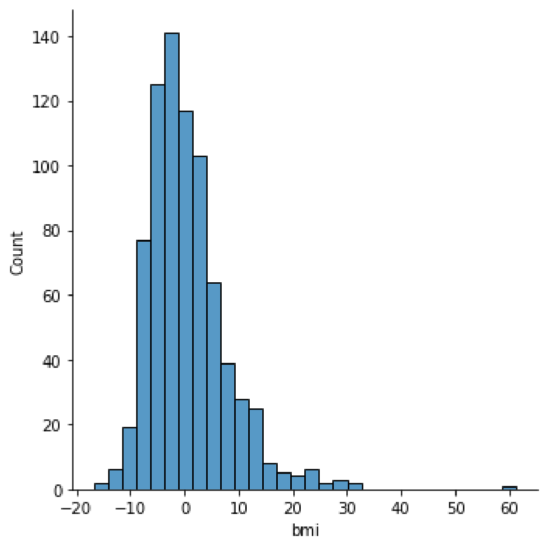


Figure 2:formerly smoked

By looking at these histograms, we can see that the majority of instances are in the never smoked category and have the highest number of BMI, which goes against our hypothesis. Lowest number of instances are in the formerly smoked category and have the least amount of outliers. We will leave the encoding as it is, ['formerly smoked' 'never smoked' 'smokes']=[0,1,2] .

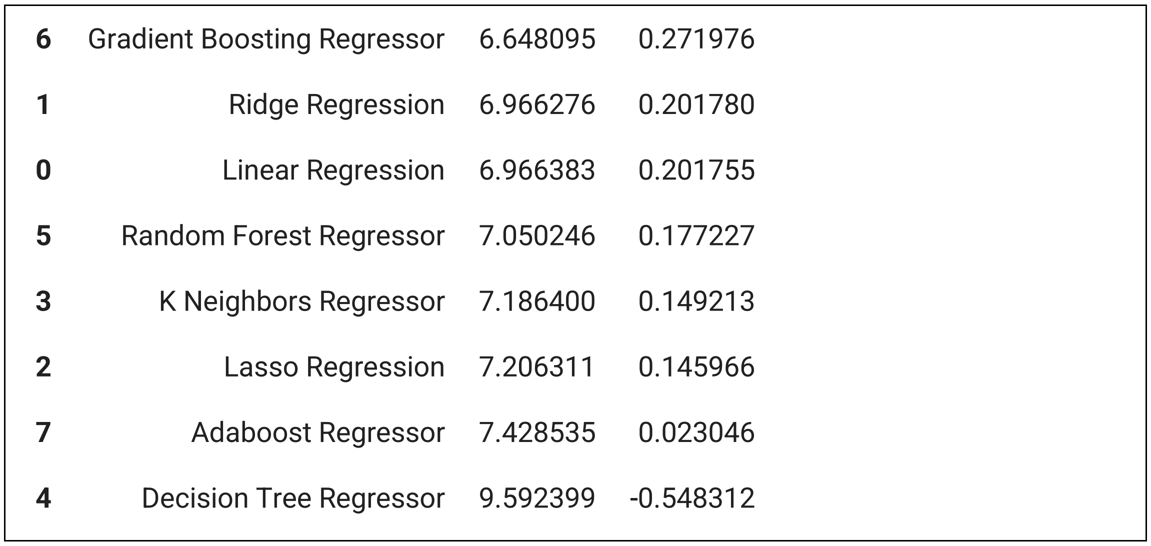
We will scale our continous data using the Standard Scaler from sklearn and as a benchmark base model, the Linear Regression model from sklearn will be used.





Bellow are the results for our following model. As we can see by the graph and the results our benchmark model is not that good, It has bad predictions for the values that are on the edges of the Gaussian Distribution.

```
 MAE: 5.325788007525139
 R_2: 0.1827850217445769
MAPE: 0.1810778357020085
 MSE: 54.06559933877375
RMSE: 2.3077668876047985
```



6	Gradient Boosting Regressor	6.648095	0.271976
1	Ridge Regression	6.966276	0.201780
0	Linear Regression	6.966383	0.201755
5	Random Forest Regressor	7.050246	0.177227
3	K Neighbors Regressor	7.186400	0.149213
2	Lasso Regression	7.206311	0.145966
7	Adaboost Regressor	7.428535	0.023046
4	Decision Tree Regressor	9.592399	-0.548312

In the table above are the list of regression models I have used to get better results. I got the following results using techniques such as GridSearch cross validation and ensemble modeling, they are sorted by highest R^2 score and lowest RMSE error. As we can see the Gradient Boosting Regressor has the best scores, even though they are not ideal we will use them in the further steps of this project.

```
from sklearn.model_selection import GridSearchCV
param_grid = {'n_estimators':range(20,1001,10),
              'max_depth':range(5,16,2),
              'min_samples_split':range(200,1001,200),
              'learning_rate':[0.01]}
clf = GridSearchCV(GradientBoostingRegressor(random_state=1),
                   param_grid = param_grid, scoring='r2',
                   cv=cv).fit(X_train_scaled, y_train)
print(clf.best_estimator_)
print("R Squared:",clf.best_score_)
```

By using the code above and various hyperparameters we were able to get the Gradient Boosting Regressor model with the following parameters, which were seen to give the best results.

```
GradientBoostingRegressor(learning_rate=0.2, max_depth=10,
                          min_samples_split=100, n_estimators=20,
                          random_state=1)
```

4. Classification of Dataset

As things stand we have a dataset of 4037 rows. The smoking_status column has all the 'Unknown' values removed. BMI will be imputed using the following model discussed above, with training on 3884 and imputing 153 values. We will use 2 techniques, the first one will be undersampling, which we will get a more 'real' model with original values from the dataset(apart from the bmi imputed values). The other method will be oversampling which will train the model better, but at a cost because we will use an OverSampler to even out the stroke instances and because some of the data will be artificial we can't really be sure of the results for that model.

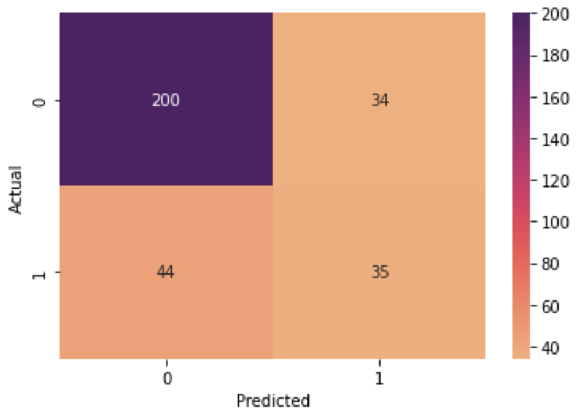
I used the RandomUnderSampler from the library imblearn.

```
the number of classes before fit Counter({0: 4860, 1: 249})
the number of classes after fit Counter({0: 249, 1: 249})
```

Now we have dataset of 498 instances. The baseline model for this problem will be the LogisticRegression from sklearn as it is a binary classification problem.

```
[[31 17]
 [ 7 45]]
0.7894736842105263
              precision    recall  f1-score   support

           0       0.82      0.65      0.72        48
           1       0.73      0.87      0.79        52

    accuracy                           0.76       100
   macro avg       0.77      0.76      0.76       100
weighted avg       0.77      0.76      0.76       100
```

As we can see our first model has good results. Judging by the lack of data for the model, we can say that we obtained a good F1_SCORE. Precision and recall for the classes which instances have had a stroke appear to be good but recall is not that good for instances that have not had a stroke. These are the initial results and we will change up the dataset for better results and will use more techniques.



The final dataset, through extensive research with the shape of the dataset and how we will scale, encode and impute the missing values, will have a following dataset with these characteristics:

- The BMI values will be imputed using age. Age will be converted to classes of Kid, Teen, Adult and Senior.
- BMI will be imputed with the median of each class.
- Unknown values of smoking_status will be kept and be treated as a category rather than a None value
- We will use the np.log function for each numerical value and the we will scale them

The results will vary from around 0-10% depending on the random sample that will be taken with the UnderSampler.

1. LogisticRegression

```
[[33 15]
 [ 6 46]]

F1: 0.81
              precision    recall  f1-score   support

           0       0.85      0.69      0.76        48
           1       0.75      0.88      0.81        52

    accuracy                           0.79       100
   macro avg       0.80      0.79      0.79       100
weighted avg       0.80      0.79      0.79       100
```

2. RandomForrestClassifier with criterion entropy

```
[[33 15]
 [10 42]]

F1: 0.77
              precision    recall  f1-score   support

           0       0.77      0.69      0.73        48
           1       0.74      0.81      0.77        52

    accuracy                           0.75       100
   macro avg       0.75      0.75      0.75       100
weighted avg       0.75      0.75      0.75       100
```

3. **<u>GaussianNB</u>**

```
[[32 16]
 [ 6 46]]

F1: 0.8070175438596492
              precision    recall  f1-score   support

           0       0.84      0.67      0.74        48
           1       0.74      0.88      0.81        52

    accuracy                           0.78       100
   macro avg       0.79      0.78      0.78       100
weighted avg       0.79      0.78      0.78       100
```

4. **<u>AdaboostClassifier using KFold with 5 splits</u>**

```
Mean of F1 Score: 0.8015466015466016
```

5. **<u>AdaboostClassifier using GridSearch</u>**

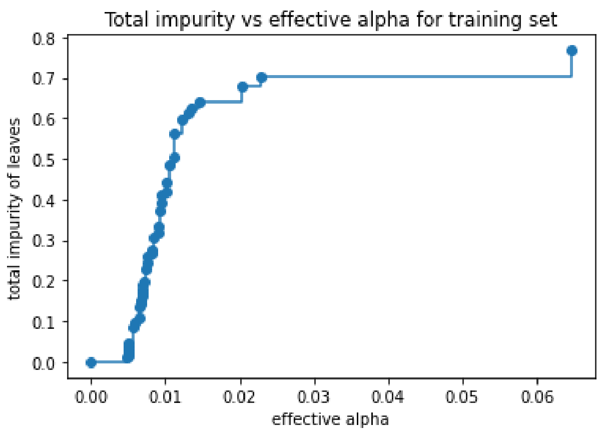
```
[[29 19]
 [ 3 49]]

F1: 0.8166666666666668
              precision    recall  f1-score   support

           0       0.91      0.60      0.72        48
           1       0.72      0.94      0.82        52

    accuracy                           0.78       100
   macro avg       0.81      0.77      0.77       100
weighted avg       0.81      0.78      0.77       100
```

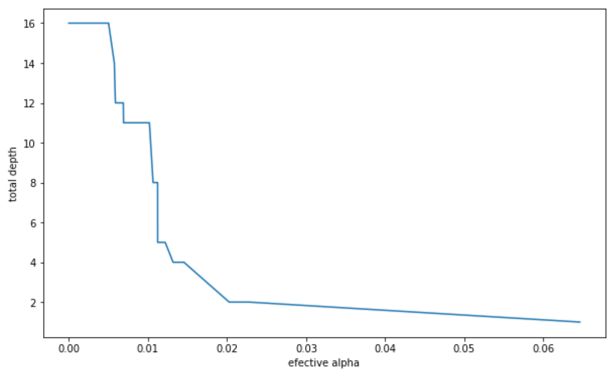
6. **<u>Voting classifier using 3 best classifiers from above</u>**

```
[[30 18]
 [ 3 49]]

F1: 0.8235294117647058
              precision    recall  f1-score   support

           0       0.91      0.62      0.74        48
           1       0.73      0.94      0.82        52

    accuracy                           0.79       100
   macro avg       0.82      0.78      0.78       100
weighted avg       0.82      0.79      0.78       100
```

7. DecisionTree and using pruning techniques

Now we will try and create a better model by minimalizing the depth of the tree and tuning effective alpha parameters. The alpha parameter determines how clean the split of the tree is. This technique is used for making a more general model and is used to prevent overfitting of our model. Here we will use a DecisionTree model from sklearn with criterion entropy as it has shown it gives the best results.

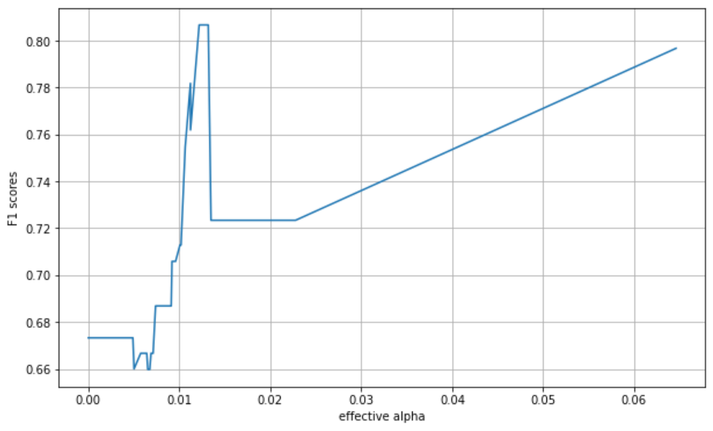




These two graphs show how our effective alpha varies by the size of the tree. In the second graph we can see how it changes by the size, a tree with 16 leaves has a minimal effective alpha, the tree is too complex and could result in an overfit model.

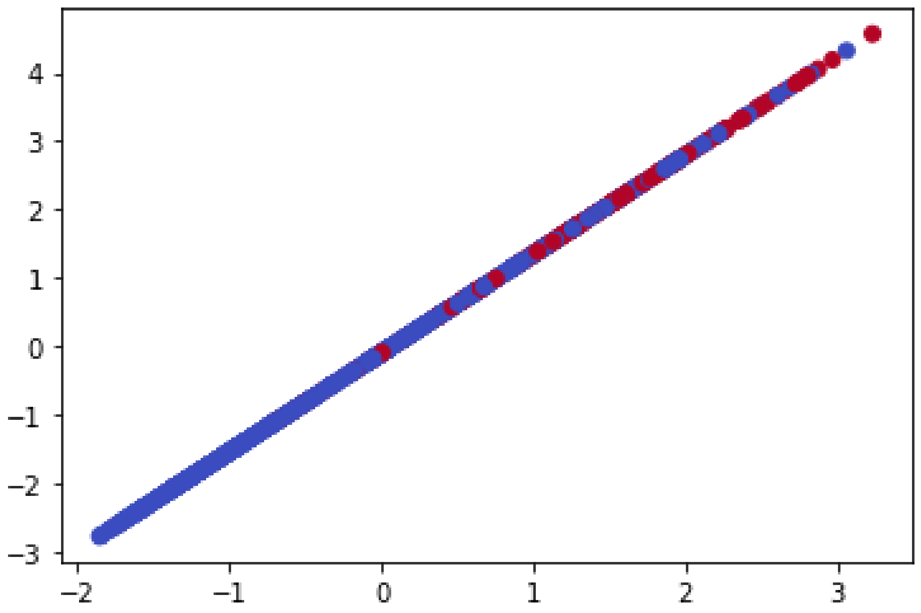
By using these alpha parameters we can see how our scores would change with the changing of the parameter.

Alpha: 0.0121835663078483, F1: 0.8067226890756303



As we can see with a alpha value of zero we have a weaker model which is overfit. By changing the value of alpha we can see the model is becoming better and has a f1 score of around 80%. By having a less complex tree, smaller one, we can see that the model performs far better than a very complex one. We can choose the tradeoff however we like.

5. LDA classification

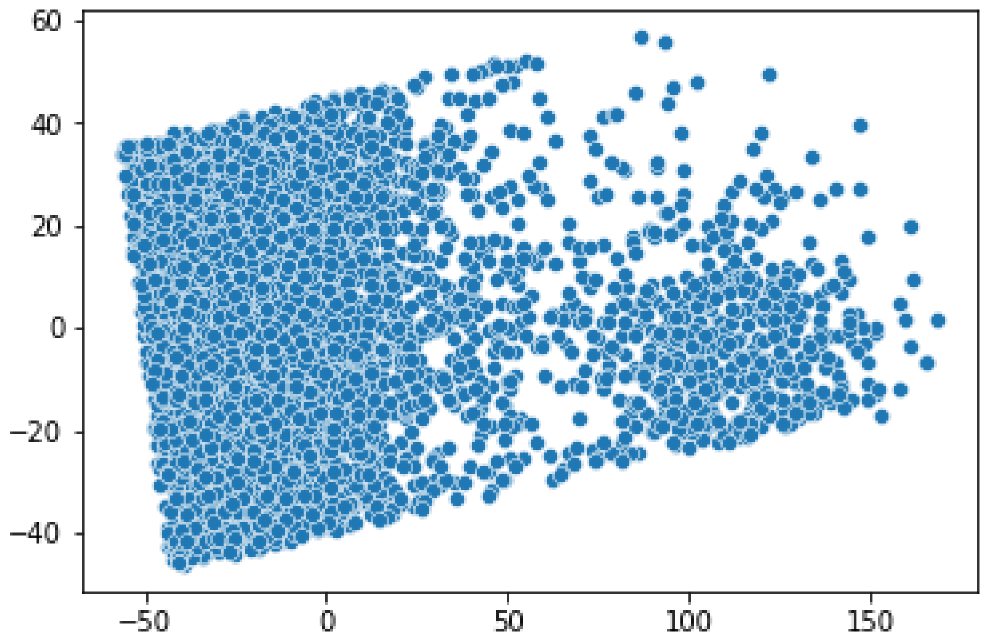


We use LDA to lower the dimensionality of our dataset. This graph shows how our data would be classified by lowering all the dimensions. LDA is used to lower the use of resources in high dimensional data, in our case the data does not have many dimensions so LDA would be unneeded.

By using LDA with a RepeatedStratifiedKFold with 10 splits on 3 repeats and Grid Search for the best distance metric we get a F1 score of 0.6997121397121399 with eigen distance.

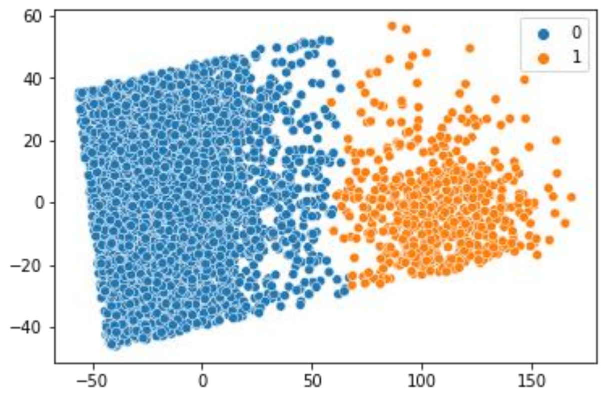
6. Clustering

In this part we will use various clustering techniques on our dataset. Before we do the clustering we would like to visually see how our data would be clustered. This would be a challenge as our data has 12 dimensions and we as humans cannot visually comprehend that many dimensions. We will solve this problem using PCA. PCA is a technique used to lower the number of dimensions of data. It is very similar to LDA but the main difference is LDA is centered around the target variable, where PCA takes the whole data.

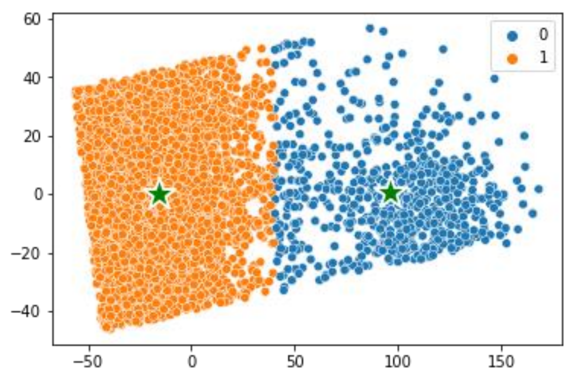


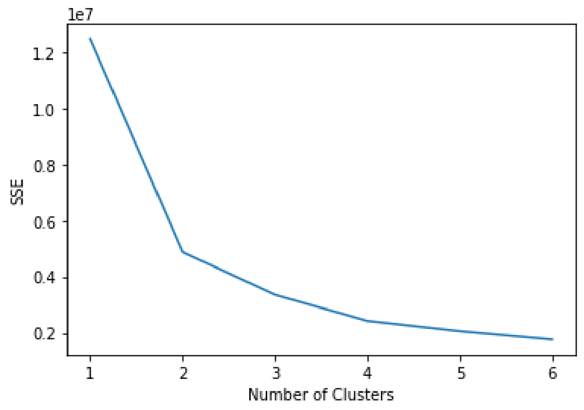
4:Data reduced to 2 dimensions

As we can see the scatter plot, we can see that there are 2 clusters visible by the eye. Between these 2 clusters there are points that are not so simple to append to a cluster. We will use Agglomerative clustering and kmeans clustering to see the results of how the algorithms will classify these points.



Here we can see that agglomerative clustering gives us a very good result in terms of our 2 dimensional data. This algorithm works best because it chooses the N closest points and continues building on those points. It works better than k-means because k-means clustering chooses a random point in the space and builds upon it. There could be inaccurate clusters depending on the point k-means randomly chooses.





These are our clusters with the k-means algorithm. We determined the best k by using the Knee method. As we can see the optimal number of clusters here is 2. As mentioned above k-means can give various results depending on the point in the dimension space it assigns in the first part.

Two other examples of clustering are shown in the notepad for clustering. There are not optimal for our two dimensional data. Density based clustering would be better for multidimensional data, by increasing the number of components for our PCA we could get better clustering results for our data.

7. References:

- https://www.nhs.uk/conditions/stroke/causes/
- https://www.nhs.uk/conditions/high-blood-pressure-hypertension/
- https://www.nhs.uk/conditions/obesity/
- https://www.nhs.uk/conditions/high-cholesterol
- https://www.nhs.uk/conditions/diabetes/
- https://www.ahajournals.org/doi/10.1161/CIRCRESAHA.121.319915
- https://colab.research.google.com/drive/1m5K8CYooUM7hXWOAXdnSVCkNTLzEyeNR?usp=sharing
- https://colab.research.google.com/drive/1ZkUsIzuF_TtSTvKHn9bOh6EpTKIUXyN3?usp=sharing
- https://colab.research.google.com/drive/1BTSnLzdEAdwkFxatEuI7YkPSI761kDIW?usp=sharing
- https://colab.research.google.com/drive/1gh7hr9OZoiLVBZCOhJdTv7ZaN5NDJgXX?usp=sharing