Post-block Assignment

Data Science 874

Data Understanding

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Figure 1: First 5 Rows of the dataset

Feature Description

Feature	Description	Data Type	Unique Values
id	Unique identifier	Numerical – Absolute	N/A
gender	Patient's gender	Categorical - Nominal	Male, Female, Other
age	Age of patient	Numerical – Absolute	N/A
hypertension	Does patient have hypertension?	Categorical – Nominal	0 (No), 1 (Yes)
heart_disease	Does patient have any heart diseases?	Categorical – Nominal	0 (No), 1 (Yes)
ever_married	Has patient been married?	Categorical – Nominal	0 (No), 1 (Yes)
work_type	Patient's work history	Categorical – Nominal	Private, Self-employed, Govt_job, children, Never_worked
residence_type	Patient's residence area type	Categorical – Nominal	Urban, Rural
avg_glucose_level	Average glucose level in blood	Numerical – Absolute	N/A
bmi	Body mass index	Numerical – Absolute	N/A
smoking_status	Current smoking status	Categorical – Nominal	formerly smoked, never smoked, snokes, Unknown
stroke	Has patient experienced stroke?	Categorical – Nominal	0 (No), 1 (Yes)

Table 1: Feature description

This dataset is used to predict whether a patient is likely to get a stroke. It contains 12 features including the target variable (stroke). The numerical variables are id, age, *avg_glucose_level* and *bmi*. All are absolute numerical values - age can be made into a ratio variable if it is converted to an integer type. Binary valued variables – such as stroke, *ever_married*, *heart_disease* - can be interpreted as categorical variables because 0 represents 'No' and 1 represents 'Yes'. All categorical variables are nominal. From the figure above we can see the features, their types, and the unique values they take on. It is important to note the following:

- *gender* has a value of 'Other', however the dataset contains only one instance of it
- *smoking_status* has a value of 'Unknown'. This indicates that the information about the patient is unavailable. We will evaluate whether this should be treated as missing value or not
- *age* is a float and contains values with non-zero decimal numbers
- *id* is a unique field used to identify a person and we expect it to not have any predictive capability

Data Quality Report

Data Quality Report for Numerical Features

	id	age	avg_glucose_level	bmi
count	5109.000000	5109.000000	5109.000000	5109.000000
mean	36513.985516	43.229986	106.140399	28.863300
std	21162.008804	22.613575	45.285004	7.699785
min	67.000000	0.080000	55.120000	10.300000
25%	17740.000000	25.000000	77.240000	23.800000
50%	36922.000000	45.000000	91.880000	28.100000
75%	54643.000000	61.000000	114.090000	32.800000
max	72940.000000	82.000000	271.740000	97.600000

Data Quality Report for Categorical Features

	gender	hypertension	heart_disease	ever_married	work_type	Residence_type	smoking_status	stroke
count	5109	5109	5109	5109	5109	5109	5109	5109
unique	2	2	2	2	5	2	3	2
top	Female	No	No	Yes	Private	Urban	never smoked	No
freq	2994	4611	4833	3353	2924	2596	3436	4860

Figure 2: Data Quality report before pre-processing

Target feature balance

From Figure 3, we can see that we're dealing with a highly unbalanced dataset. A common approach to dealing with this imbalance is the SMOTE technique. Another, approach is to be very selective with the metrics used for modelling. More specifically, it is important to use metrics such as F1-score and AUC as opposed to accuracy. The latter is the approach that will be taken moving forward into modelling.

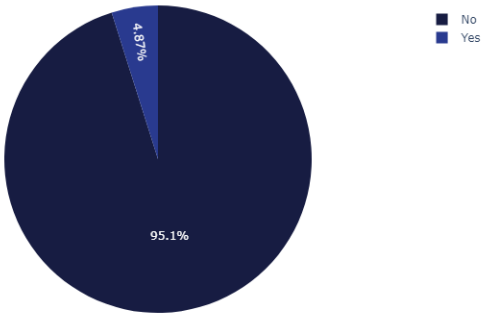

Figure 3: Positive to Negative Stroke ratio

Data Visualisation

Scatter Plot Matrix

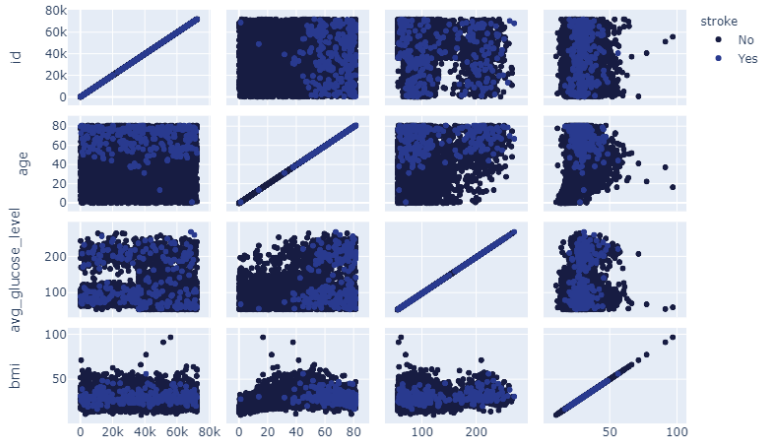

Figure 4: Scatter Plot Matrix for Numeric Features

Figure 4 shows the scatterplot matrix for numeric features. From the bottom right scatter plot, appears that the *bmi* has some outliers. It also appears that *avg_glucose_level* and *bmi* have some sort of relationship.

Numerical Feature Histograms

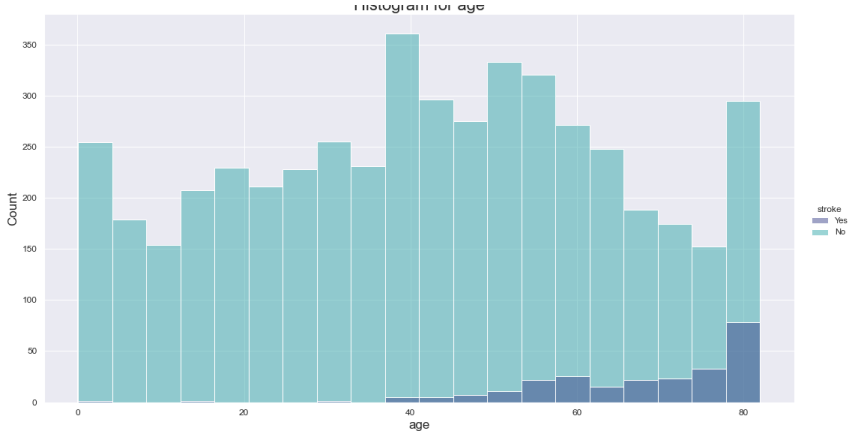

Figure 5: Histogram for age

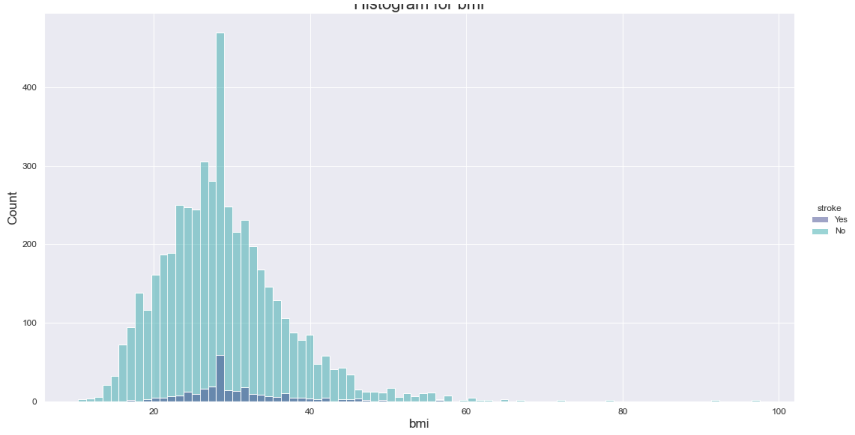

Figure 6: Histogram for bmi

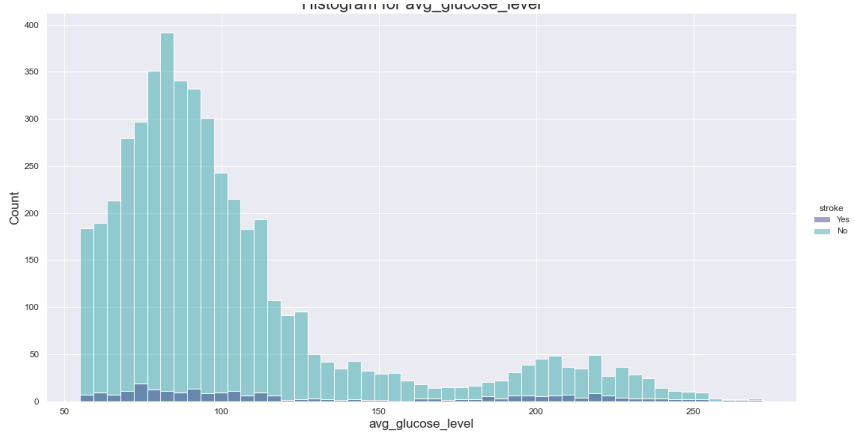

Figure 7: Histogram for avg_glucose_level

From Figure 5, we can see that *age* appears to be normally distributed with a mode of approximately 40 years old. Figure 6 shows the distribution for *bmi* and it appears to be a unimodal distribution that is skewed to the right. The distribution for *avg_glucose_level* appears to be a bimodal distribution that is skewed to the right as well.

Categorical Feature Histograms

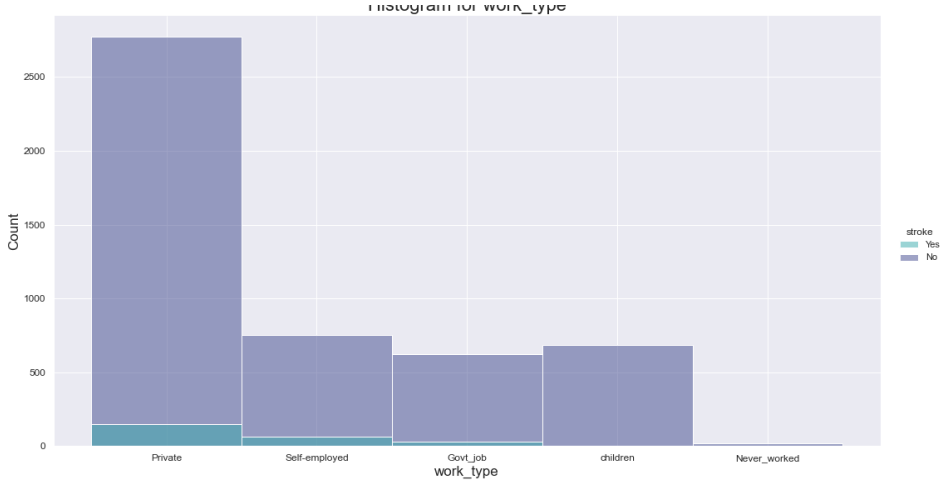

Figure 8: Histogram for work_type

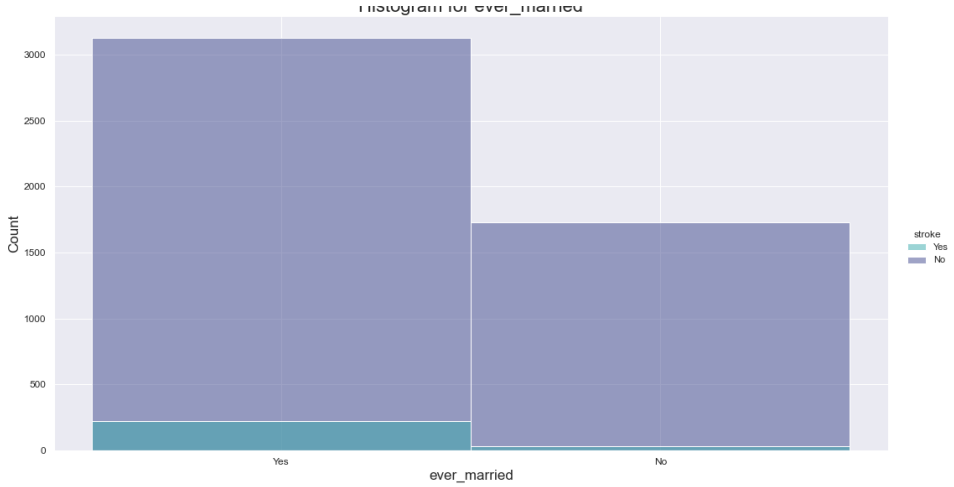

Figure 9: Histogram for ever_married

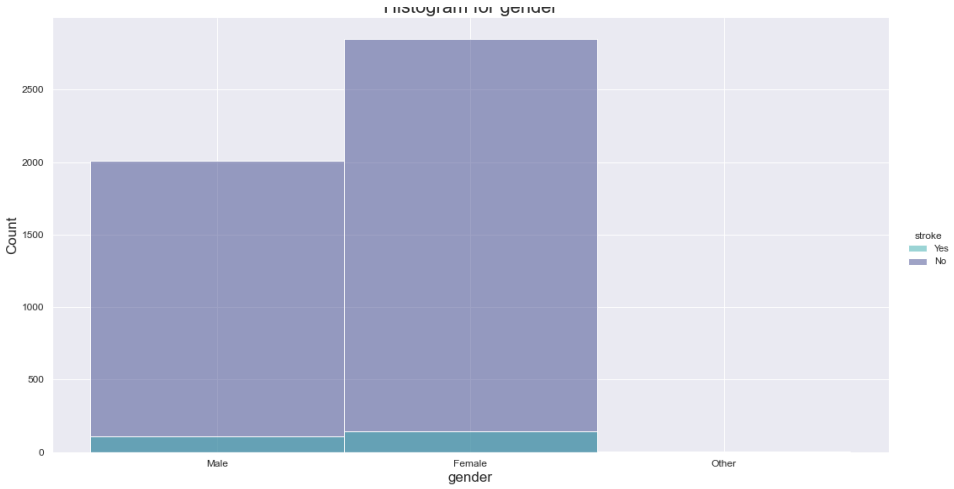

Figure 10: Histogram for gender

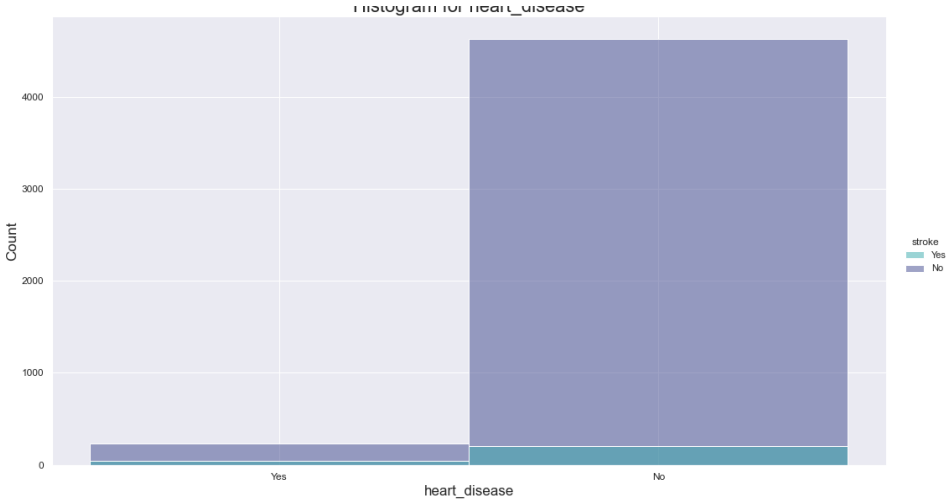

Figure 11: Histogram for heart_disease

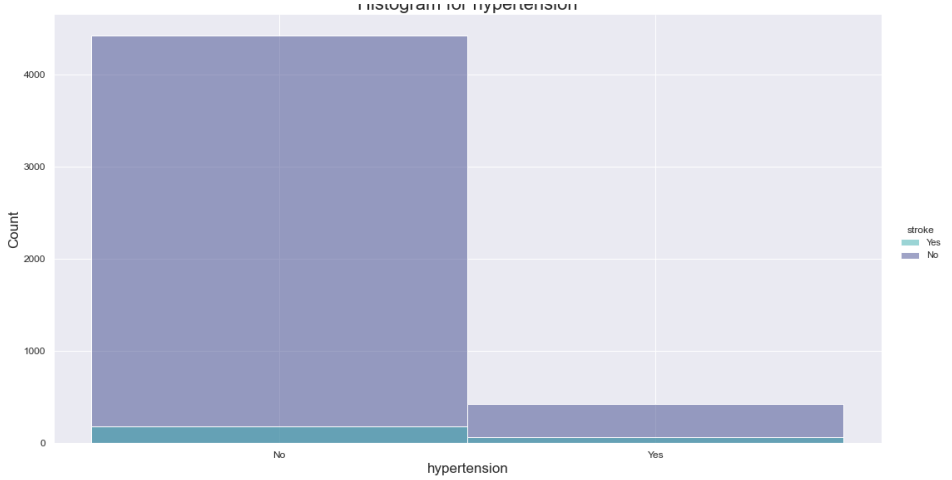

Figure 12: Histogram for hypertension

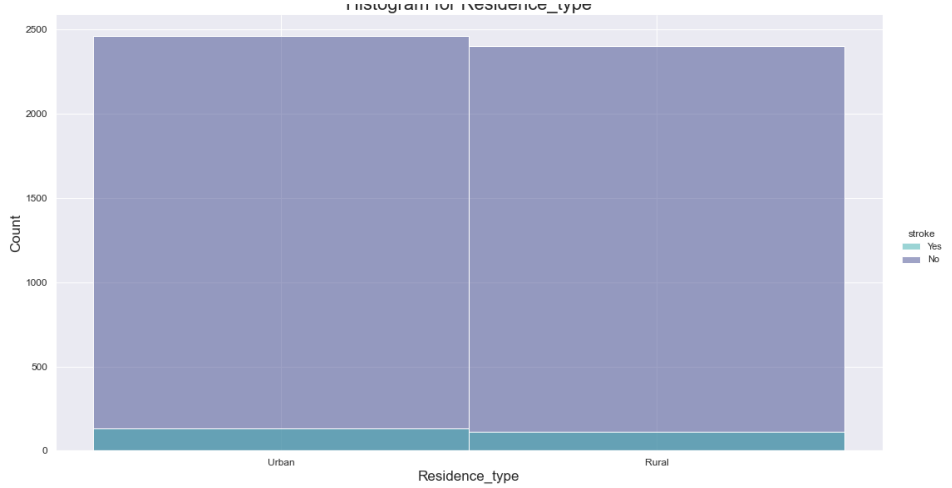

Figure 13: Histogram for residence_type

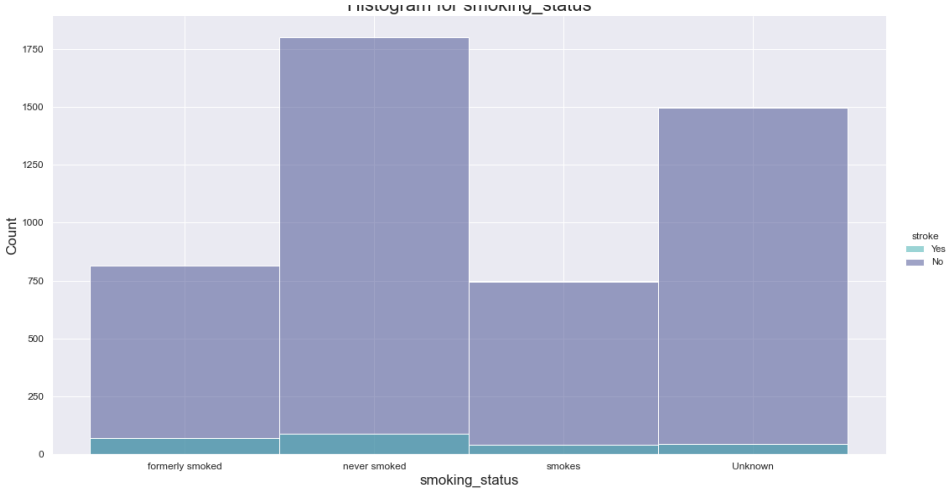

Figure 14: Histogram for smoking_status

As we can see from the bar plots, some features have highly imbalanced categories. It is expected that features with imbalanced categories will be weaker indicators for our target variable, this can be seen further down in the feature selection section.

Numerical Features Box Plots

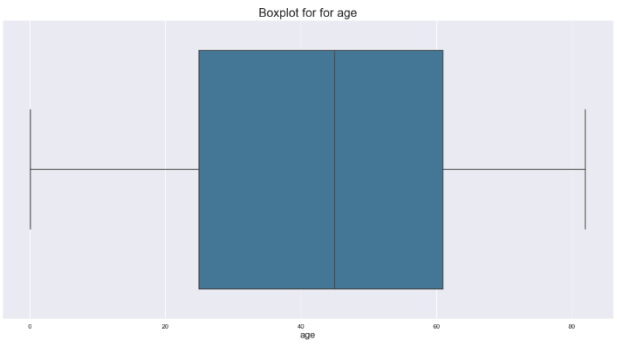

Figure 15: Box plot for Age

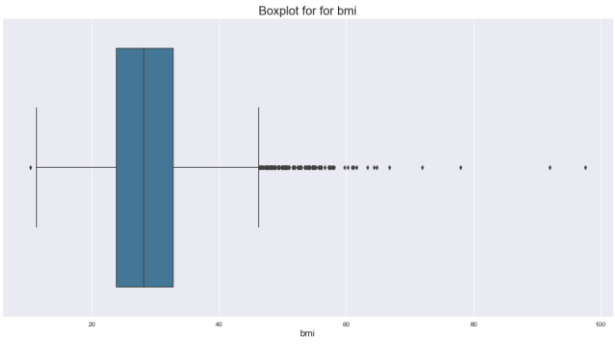

Figure 16: Box plot for bmi

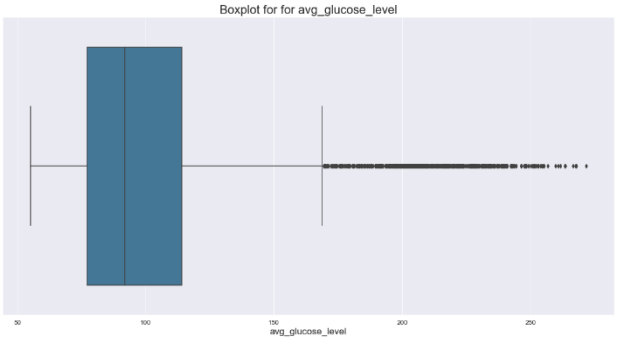

Figure 17: Box plot for avg_glucose_level

From the box plots for bmi and avg_glucose_level we can see that there seems to be outliers – mostly after the upper quartile.

Outliers

To detect outliers, the upper quartile (Q1), the lower quartile (Q3) and inter-quartile range (IQR) were computed for each numeric feature. A commonly used rule says that a data point is an outlier if it is more than (Q3 + 1.5 · IQR), or less than (Q1 - 1.5 · IQR). These are referred to as 'upper' and 'lower' respectively in Figure 18. This is a fair number of outliers (roughly 15%) so instead of removing the instances, we will apply a clamp transformation.

```
#Outliers for avg_glucose_level: 627. Upper: 169.36. Lower: 22
#Outliers for age: 0. Upper: 115. Lower: -29
#Outliers for bmi: 126. Upper: 46. Lower: 10
```

Figure 18: Outliers for numeric features

Missing Values

It is evident from Figure 1 that the dataset has missing values. We can see where these missing values come from in Figure 19.

```
Missing values per feature

id                     0
gender                 0
age                    0
hypertension           0
heart_disease          0
ever_married           0
work_type              0
Residence_type         0
avg_glucose_level      0
bmi                  201
smoking_status      1544
stroke                 0
dtype: int64

% values missing for bmi: 3.93%
% values missing for smoking_status: 30.22%
Number of rows with at least one missing value:  1684
Percentage of rows with at least one missing value: 32.96 %
```

Figure 19: Missing values

It would be negligent to just drop these values without first evaluating the best approach for this given dataset.

```
Stroke Values for missing BMI observations
No     161
Yes     40
Name: stroke, dtype: int64

Stroke Values for missing smoking_status observations
No     1497
Yes      47
Name: stroke, dtype: int64
```

Figure 20: Target feature counts for missing values

From Figure 20, we can see that the missing *bmi* account for a very small portion of the missing values and as such the missing values were imputed with the median value – median as opposed to mean because the dataset contains outliers. With regards to the *smoking_status* missing values, the approaches that are worth considering are to either drop the missing values, drop the column, imputation or leaving it as is. Because these missing values account for 30.22% of the dataset's values, the first approach is unfeasible. To drop the column without first evaluating how important this feature is would also be negligent.

From Figure 22 we can see that *smoking_status* appears the be the 4th most important feature and thus it would not make sense to drop the column. We can also see from Figure 15 that 'Unknown' is the second most recorded value for smoking_status. Furthermore, from the comparison between Figure 21 and 22, it is evident that filling 'Unknown' with the mode reduces the feature's strength by approximately 26%. This could be due to the feature becoming increasingly imbalanced. Therefore, the feature will be left as is.

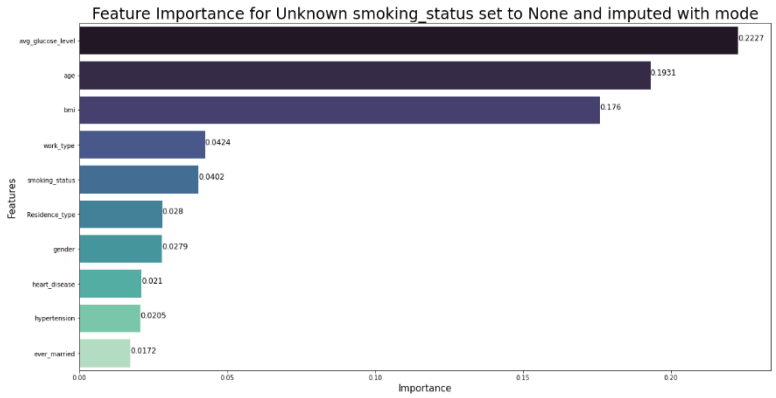

Figure 21: Feature importance's according to Random Forest Classifier with Unknown filled with mode

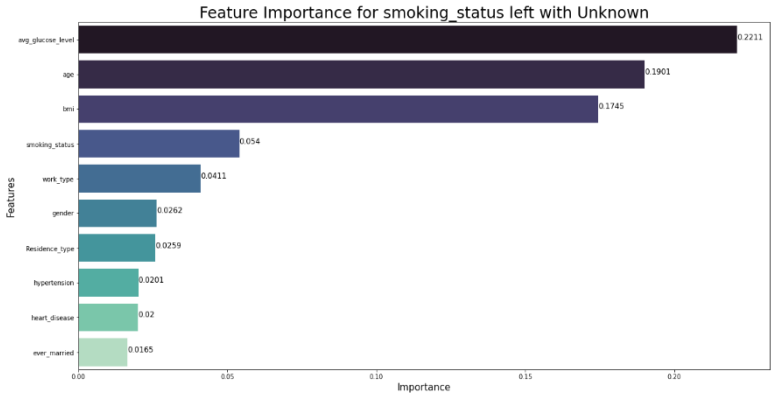

Figure 22: Feature importance's according to Random Forest Classifier with Unknown left as is

Data Quality Issues

From the steps above, a summary of the data quality issues and strategies to deal with them are shown in Table 1.

Feature	Data Quality Issue	Strategy
stroke	• Irregular Cardinality (binary variable)	• Convert to categorical (0 for No and 1 for Yes)
hypertension	• Irregular Cardinality (binary variable)	• Convert to categorical (0 for No and 1 for Yes)
heart_disease	• Irregular Cardinality (binary variable)	• Convert to categorical (0 for No and 1 for Yes)
gender	• Invalid Outlier ('Other')	• Drop observation
bmi	• Missing values • Outliers	• Impute with median (28.1) • Clamp Transformation (upper: 46 and lower: 10)
id	• High cardinality (Irrelevant)	• Drop feature
age	• Datatype issue (float has decimals)	• Convert into int
avg_glucose_level	• Outliers	• Clamp Transformation (upper: 169.36 and lower: 22)

Table 2: Data quality issues

Data Preparation

Outliers

After applying a clamping transformation, we can see from Figure 23 that all outliers that were previously identified have been removed.

```
#Outliers for avg_glucose_level: 0. Upper: 169.36. Lower: 22
#Outliers for age: 0. Upper: 115. Lower: -29
#Outliers for bmi: 0. Upper: 46. Lower: 10
```

Figure 23: Outliers after clamping transformation

Missing Values

```
Missing values per feature

id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                  0
smoking_status       0
stroke               0
dtype: int64

% values missing for bmi: 0.0%
% values missing for smoking_status: 0.0%
Number of rows with at least one missing value:  0
Percentage of rows with at least one missing value: 0.0 %
```

Figure 24: Missing Values after cleaning

Features after cleaning

Feature	Data Type	Unique Values
gender	Categorical - Nominal	Male, Female
age	Numerical – Ratio	N/A
hypertension	Categorical – Nominal	0 (No), 1 (Yes)
heart_disease	Categorical – Nominal	0 (No), 1 (Yes)
ever_married	Categorical – Nominal	0 (No), 1 (Yes)
work_type	Categorical – Nominal	Private, Self-employed, Govt_job, children, Never_worked
residence_type	Categorical – Nominal	Urban, Rural
avg_glucose_level	Numerical – Absolute	N/A
bmi	Numerical – Absolute	N/A
smoking_status	Categorical – Nominal	formerly smoked, never smoked, smokes, Unknown
stroke	Categorical – Nominal	0 (No), 1 (Yes)

Table 3: Feature description after cleaning

Data Quality Report after cleaning

Data Quality Report for Numerical Features

	age	avg_glucose_level	bmi
count	5109.000000	5109.000000	5109.000000
mean	43.218634	100.987918	28.691642
std	22.634799	33.212706	7.121011
min	0.000000	55.120000	10.300000
25%	25.000000	77.240000	23.800000
50%	45.000000	91.880000	28.100000
75%	61.000000	114.090000	32.800000
max	82.000000	169.357500	46.300000

Data Quality Report for Categorical Features

	gender	hypertension	heart_disease	ever_married	work_type	Residence_type	smoking_status	stroke
count	5109	5109	5109	5109	5109	5109	5109	5109
unique	2	2	2	2	5	2	4	2
top	Female	No	No	Yes	Private	Urban	never smoked	No
freq	2994	4611	4833	3353	2924	2596	1892	4860

Figure 25: Data Quality Report after cleaning

Categorical Encoding

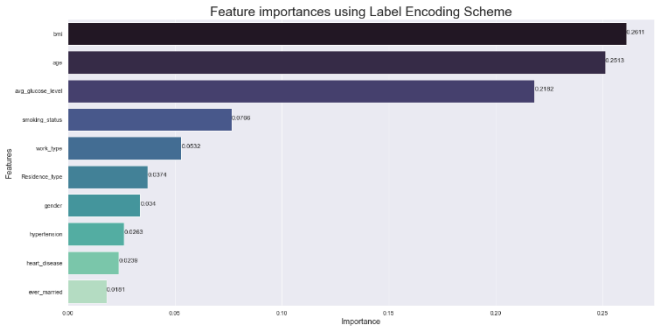

Figure 26: Feature importance using Label Encoding

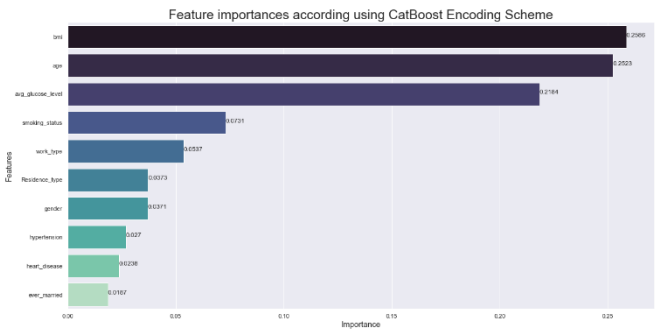

Figure 27:Feature importance using CatBoost Encoding

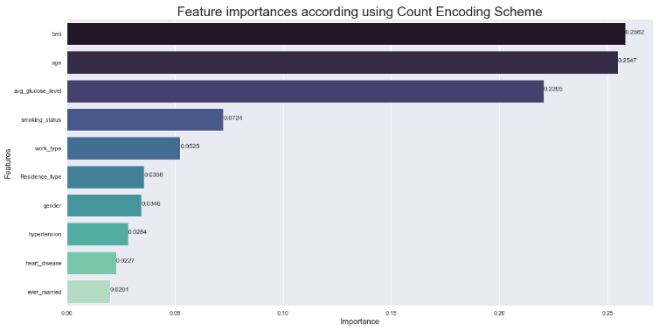

Figure 28: Feature Importance using Count Encoding

Three different categorical encoding schemes were evaluated for the final dataset to be used for modelling – Label, CatBoost and Count encoding. From Figures 26, 27 and 28 we can see the feature importance for each scheme. It appears that Count Encoding yields the best results, and this is what was selected for modelling.

Feature Selection

Inter-feature correlation

From Figure 29 we can see that neither of the 3 numeric features have any significant collinearities that they warrant one of the features being dropped. It is important to note that categorical variables were excluded from the matrix because none of the categorical variables are ordinal so it would not be a fair assessment of any collinearity

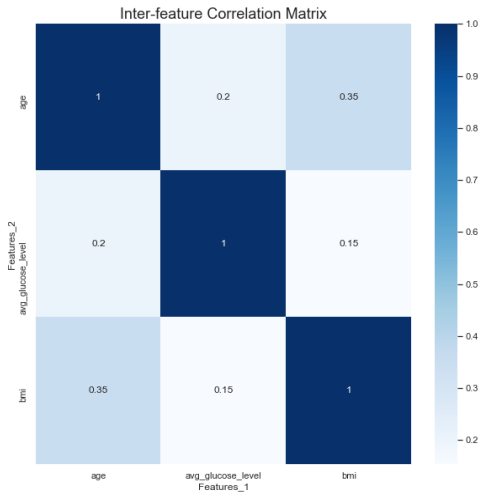

Figure 29: Inter-feature correlation matrix for numerical features

In the modelling process, the effect – on model performance - of dropping features in order of weakest will be investigated to assess the ideal number of features

Data

	Residence_type	age	avg_glucose_level	bmi	ever_married	gender	heart_disease	hypertension	smoking_status	work_type	stroke
0	2596	67	169.3575	36.6	3353	2115	276	4611	884	2924	1
1	2513	61	169.3575	28.1	3353	2994	4833	4611	1892	819	1
2	2513	80	105.9200	32.5	3353	2115	276	4611	1892	2924	1
3	2596	49	169.3575	34.4	3353	2994	4833	4611	789	2924	1
4	2513	79	169.3575	24.0	3353	2994	4833	498	1892	819	1

Figure 30: First five rows of data after cleaning

The final dataset was partitioned into a 60% training set 20% for validation and 20% test set.

Modelling Technique

After completing the EDA, I would typically evaluate the performance of various types of models for a particular problem: such as tree-based models, neural networks, memory-based classifiers. More recently there has been a rising popularity with tree-based models in the space of supervised classification and regression. **Tree-based boosting models** such as XGBoost and LightGBM have become popular for achieving extremely good performances in Kaggle competitions – high accuracy and fast training times. Models such as these have several benefits such as:

- Less sensitive to outliers
- Does not require normalisation or scaling - equal importance given to features regardless of scale. Does not make any assumptions of the data's shape. It is also quite effective in modelling non-linear relationships
- More interpretable and explainable results than some models such as Neural Network models
- Helpful in EDA
- Requires less effort in preprocessing