MVP

LAMYA ALARWAN

Handle missing values:

```
#showing information about dataset
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   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)
memory usage: 479.2+ KB
```

```
# missing values
imp_KNN = KNNImputer(n_neighbors = 4)
d['bmi'] = imp_KNN.fit_transform(d[['bmi']])
```

```
#showing information about dataset
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   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                5110 non-null   float64
 10  smoking_status     5110 non-null   object
 11  stroke             5110 non-null   int64
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

Handle columns:

```
#showing information about dataset
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   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                5110 non-null   float64
 10  smoking_status     5110 non-null   object
 11  stroke             5110 non-null   int64
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

```
#Handle smoking_status column
smoking = pd.get_dummies(d[['smoking_status']], drop_first= True)
smoking.head(2)
```

	smoking_status_formerly smoked	smoking_status_never smoked	smoking_status_smokes
0	1	0	0
1	0	1	0

```
#Handle gender column
gender = pd.get_dummies(d[['gender']], drop_first= True)
gender.head(2)
```

	gender_Male	gender_Other
0	1	0
1	0	0

```
#Handle ever_married column
married = pd.get_dummies(d[['ever_married']], drop_first= True)
married.head(2)
```

	ever_married_Yes
0	1
1	1

```
#Handle Residence_type column
residence = pd.get_dummies(d[['Residence_type']], drop_first= True)
residence.head(2)
```

	Residence_type_Urban
0	1
1	0

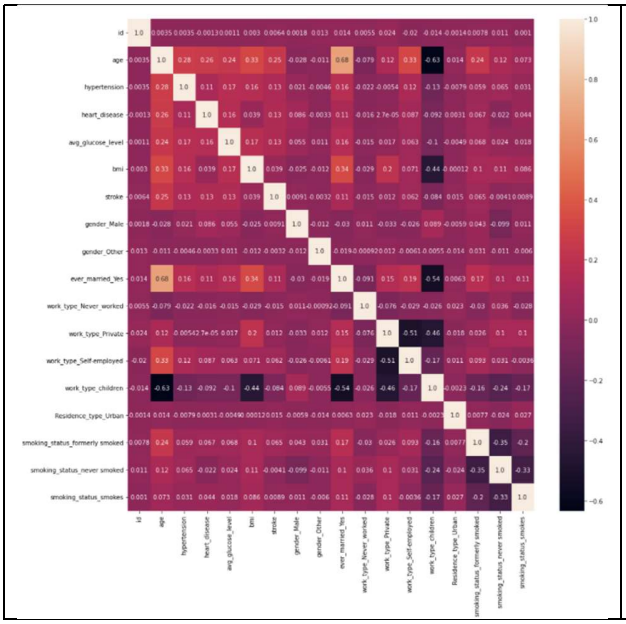
```
#Handle work_type column
work = pd.get_dummies(d[['work_type']], drop_first= True)
work.head(2)
```

	work_type_Never_worked	work_type_Private	work_type_Self-employed	work_type_children
0	0	1	0	0
1	0	0	1	0

```
#showing information about dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 18 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   id                             5110 non-null   int64
 1   age                            5110 non-null   float64
 2   hypertension                   5110 non-null   int64
 3   heart_disease                  5110 non-null   int64
 4   avg_glucose_level              5110 non-null   float64
 5   bmi                            5110 non-null   float64
 6   stroke                         5110 non-null   int64
 7   gender_Male                    5110 non-null   uint8
 8   gender_Other                   5110 non-null   uint8
 9   ever_married_Yes               5110 non-null   uint8
 10  work_type_Never_worked         5110 non-null   uint8
 11  work_type_Private              5110 non-null   uint8
 12  work_type_Self-employed        5110 non-null   uint8
 13  work_type_children             5110 non-null   uint8
 14  Residence_type_Urban           5110 non-null   uint8
 15  smoking_status_formerly smoked 5110 non-null   uint8
 16  smoking_status_never smoked    5110 non-null   uint8
 17  smoking_status_smokes          5110 non-null   uint8
dtypes: float64(3), int64(4), uint8(11)
memory usage: 334.5 KB
```

Correlation:

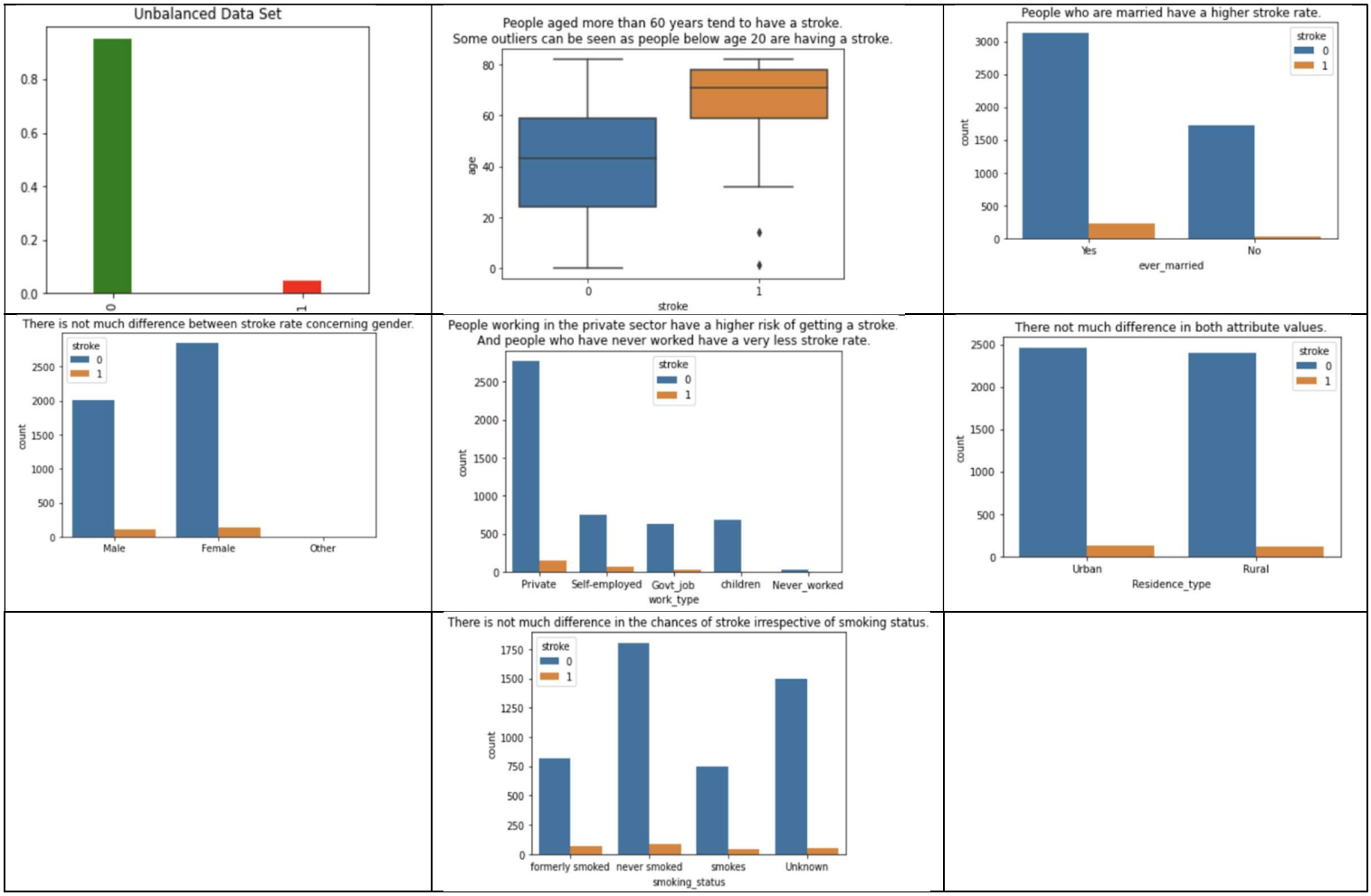


1- The [age] has (0.25) high correlation with [stroke].

2- [Hypertension]- [heart _disease] and [avg_glucose_ level] have (0.13) correlation with [stroke].

3- The [ever_ married_ yes] has (0.11) correlation with [stroke].

4- The [age] has (0.68) correlation with [ever_ married_ yes].



Modeling:

```
LG = LogisticRegression()
LG.fit(X_train, y_train)
```

→

```
LG.score(X_test,y_test)
```

```
0.9445531637312459
```

```
K= KNeighborsClassifier()
K.fit(X_train, y_train)
```

→

```
K.score(X_test,y_test)
```

```
0.945857795172863 6
```

```
S= SVC()
S.fit(X_train, y_train)
```

→

```
S.score(X_test,y_test)
```

```
0.945857795172863 6
```