Lab 9: Classification using Kernal Machines (SVM)

Created by: Preksha Shah | 2348446

Date : 10-04-2024

```
# Importing necessary libraries for data manipulation and visualization
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Importing necessary functions for machine learning tasks
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.impute import SimpleImputer


# Load the dataset into your Python environment
df = pd.read_csv("/content/healthcare-dataset-stroke-data.csv")
```

Display basic information about the dataset

```
# Number of samples and features
num_samples, num_features = df.shape
print("Number of samples:", num_samples)
print("Number of features:", num_features)
```

```
Number of samples: 5110
Number of features: 12
```

Inferences:

1. You have a moderately sized dataset, providing sufficient data for analysis and modeling.
2. The dataset contains 12 features, suggesting a variety of attributes or characteristics being measured.
3. The dataset size and feature count indicate potential complexity and richness in the data.
4. Exploratory data analysis (EDA) and feature selection techniques can help uncover insights and patterns within the data.
5. With careful preprocessing and analysis, you can leverage this dataset for various machine learning tasks, such as classification, regression, or clustering.

```
# Data types of features
print("\nData types of features:")
print(df.dtypes)
```

```
Data types of features:
id                     int64
gender                object
age                  float64
hypertension           int64
heart_disease          int64
ever_married          object
work_type             object
Residence_type        object
avg_glucose_level    float64
bmi                  float64
smoking_status        object
stroke                 int64
dtype: object
```

1. The dataset includes both numerical (int64 and float64) and categorical (object) data types.
2. Numerical features include 'id', 'age', 'hypertension', 'heart_disease', 'avg_glucose_level', 'bmi', and 'stroke'.
3. Categorical features include 'gender', 'ever_married', 'work_type', 'Residence_type', and 'smoking_status'.
4. 'stroke' appears to be the target variable, indicating a potential classification task.
5. Data types provide insight into the nature of features and guide preprocessing steps for analysis and modeling.

```
# Additional basic information about the dataset
print("\nAdditional information:")
print(df.info())
```

```
Additional information:
<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
None
```

1. The dataset consists of 5110 entries (rows) and 12 columns (features).
2. There are no missing values for most columns, except for the 'bmi' column, which has some missing values (201 missing values).
3. Memory usage for the DataFrame is approximately 479.2 KB.
4. Data types include float64 (3 columns), int64 (4 columns), and object (5 columns).
5. Features such as 'gender', 'ever_married', 'work_type', 'Residence_type', and 'smoking_status' are categorical, while others are numerical.
6. The 'stroke' column indicates potential class labels for a classification task, with values of 0 (no stroke) and 1 (stroke).

```
# Display the first few rows
print("\nFirst few rows of the dataset:")
print(df.head())
```

```
First few rows of the dataset:
      id  gender   age  hypertension  heart_disease ever_married  \
0   9046    Male  67.0             0              1          Yes   
1  51676  Female  61.0             0              0          Yes   
2  31112    Male  80.0             0              1          Yes   
3  60182  Female  49.0             0              0          Yes   
4   1665  Female  79.0             1              0          Yes   

       work_type Residence_type  avg_glucose_level   bmi   smoking_status  \
0        Private          Urban             228.69  36.6  formerly smoked   
1  Self-employed          Rural             202.21   NaN     never smoked   
2        Private          Rural             105.92  32.5     never smoked   
3        Private          Urban             171.23  34.4           smokes   
4  Self-employed          Rural             174.12  24.0     never smoked   

   stroke  
0       1  
1       1  
2       1  
3       1  
4       1  
```

1. The dataset contains information about individuals' demographics, health status, and lifestyle factors.
2. Each row represents a unique individual, identified by the 'id' column.
3. Features include gender, age, hypertension, heart disease, marital status, occupation, residence type, average glucose level, body mass index (BMI), smoking status, and stroke occurrence.
4. Missing values are present in the 'bmi' column.
5. Categorical variables include gender, ever_married, work_type, residence_type, and smoking_status.
6. The target variable 'stroke' indicates whether an individual has had a stroke (1) or not (0).
7. Further analysis and preprocessing are necessary to handle missing values and prepare the data for modeling.

Univariate Analysis:

For numerical variables:

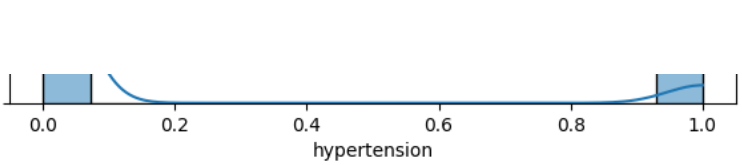
```
# Calculate basic descriptive statistics
print("\nDescriptive statistics for numerical variables:")
print(df.describe())
```

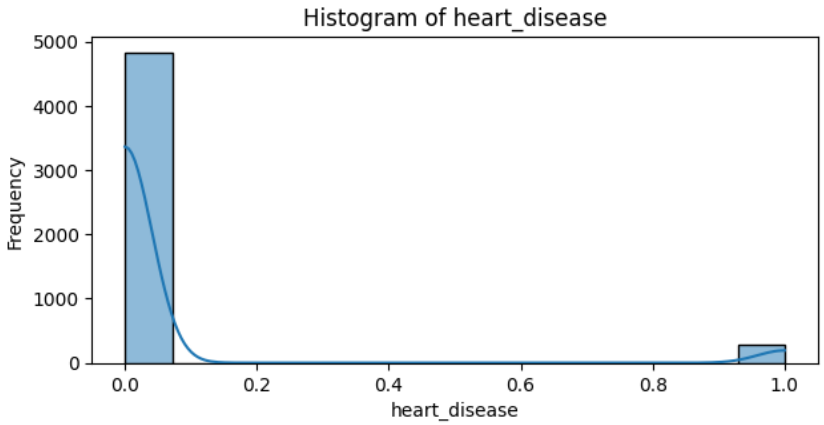
```
Descriptive statistics for numerical variables:
                 id          age  hypertension  heart_disease  \
count   5110.000000  5110.000000   5110.000000    5110.000000
mean   36517.829354    43.226614      0.097456       0.054012
std    21161.721625    22.612647      0.296607       0.226063
min       67.000000     0.080000      0.000000       0.000000
25%    17741.250000    25.000000      0.000000       0.000000
50%    36932.000000    45.000000      0.000000       0.000000
75%    54682.000000    61.000000      0.000000       0.000000
max    72940.000000    82.000000      1.000000       1.000000

       avg_glucose_level          bmi       stroke
count        5110.000000  4909.000000  5110.000000
mean          106.147677    28.893237     0.048728
std            45.283560     7.854067     0.215320
min            55.120000    10.300000     0.000000
25%            77.245000    23.500000     0.000000
50%            91.885000    28.100000     0.000000
75%           114.090000    33.100000     0.000000
max           271.740000    97.600000     1.000000
```

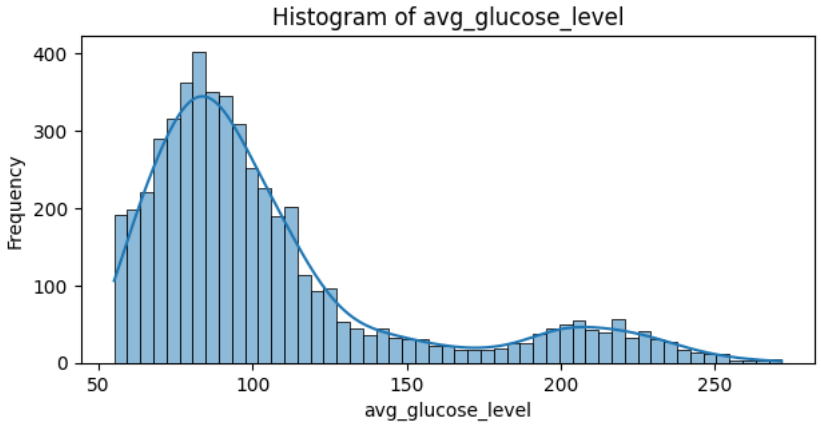
The dataset's numerical variables show:

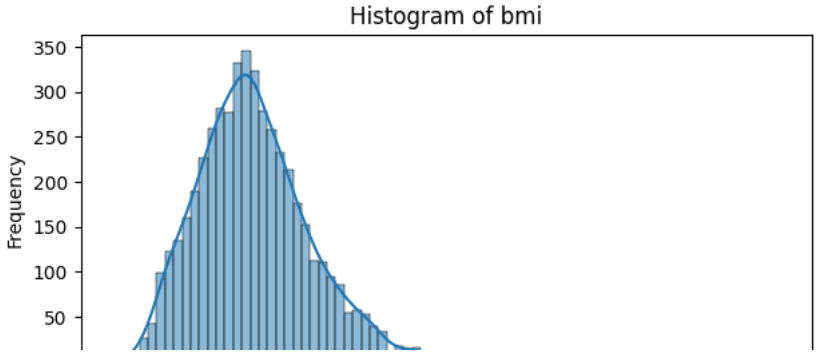
- 'age': Average age is approximately 43, ranging from 0.08 to 82.
- 'hypertension' and 'heart_disease': Both are binary variables.
- 'avg_glucose_level': Average level is about 106.15, ranging from 55.12 to 271.74.
- 'bmi': Average BMI is around 28.89, ranging from 10.3 to 97.6.
- 'stroke': Approximately 4.87% of individuals have had a stroke.

```
# Visualize the distribution using histograms, kernel density plots, and box plots
numerical_cols = df.select_dtypes(include=['int64', 'float64']).columns


for col in numerical_cols:
    plt.figure(figsize=(12,6))

    # Histogram
    plt.subplot(2, 2, 1)
    sns.histplot(df[col], kde=True)
    plt.title(f'Histogram of {col}')
    plt.xlabel(col)
    plt.ylabel('Frequency')

    plt.tight_layout()
    plt.show()
```


0
0.0
0.2
0.4
0.6
0.8
1.0
hypertension


Histogram of heart_disease
5000
4000
3000
2000
1000
0
Frequency
0.0
0.2
0.4
0.6
0.8
1.0
heart_disease


Histogram of avg_glucose_level
400
300
200
100
0
Frequency
50
100
150
200
250
avg_glucose_level


Histogram of bmi
350
300
250
200
150
100
50
Frequency

Inferences:

1. **Age Distribution**:
    - The distribution of ages appears roughly uniform, with a slight peak around the middle age range.
    - Most individuals in the dataset are between 20 and 80 years old.
2. **Average Glucose Level Distribution**:
    - The distribution of average glucose levels is slightly right-skewed, with a peak around the lower end of the scale.
    - There is a wide range of average glucose levels, indicating variability among individuals.
3. **BMI Distribution**:
    - The distribution of BMI (Body Mass Index) is slightly right-skewed, with a peak around the middle of the scale.
    - Most individuals have a BMI between 20 and 40, with fewer outliers at higher values.

For categorical variables:

```
#  Display frequency tables showing counts and percentages
print("\nFrequency tables for categorical variables:")
categorical_cols = df.select_dtypes(include=['object']).columns

for col in categorical_cols:
    print(f"\nFrequency table for {col}:")
    print(df[col].value_counts(normalize=True))
```

```
Frequency tables for categorical variables:

Frequency table for gender:
gender
Female    0.585910
Male      0.413894
Other     0.000196
Name: proportion, dtype: float64

Frequency table for ever_married:
```

```
ever_married
Yes    0.656164
No     0.343836
Name: proportion, dtype: float64

Frequency table for work_type:
work_type
Private          0.572407
Self-employed    0.160274
children         0.134442
Govt_job         0.128571
Never_worked     0.004305
Name: proportion, dtype: float64

Frequency table for Residence_type:
Residence_type
Urban    0.508023
Rural    0.491977
Name: proportion, dtype: float64

Frequency table for smoking_status:
smoking_status
never smoked       0.370254
Unknown            0.302153
formerly smoked    0.173190
smokes             0.154403
Name: proportion, dtype: float64
```

Inferences:

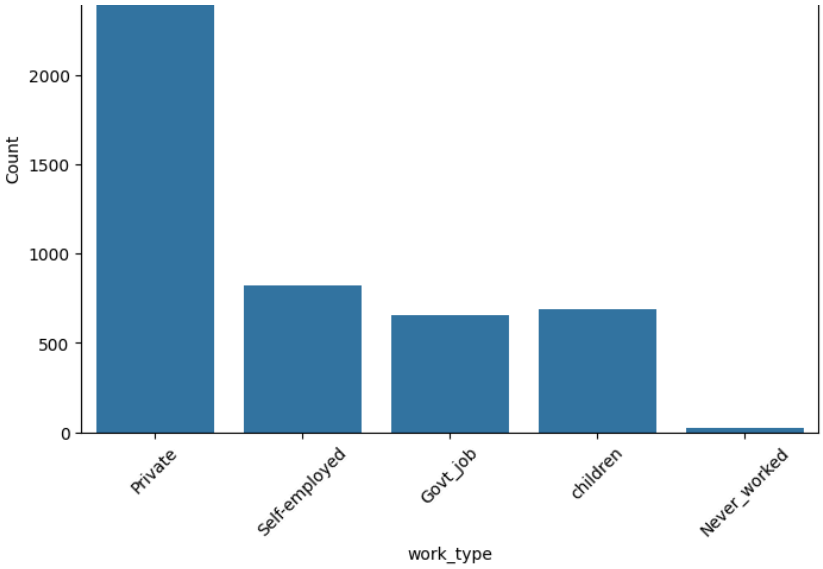
1. **Gender**:
    - The bar plot shows the distribution of genders among individuals in the dataset.
    - It indicates whether the dataset is balanced or skewed towards a particular gender.
2. **Ever Married**:
    - The bar plot illustrates the proportion of individuals who are married versus unmarried.
    - It helps understand the marital status distribution in the dataset.
3. **Work Type**:
    - The bar plot displays the distribution of individuals across different types of work.
    - It provides insights into the employment status and occupation diversity among individuals.
4. **Residence Type**:
    - The bar plot compares the number of individuals living in urban versus rural areas.
    - It helps understand the geographic distribution of individuals in the dataset.
5. **Smoking Status**:
    - The bar plot depicts the distribution of individuals based on their smoking status.
    - It shows the prevalence of different smoking habits among individuals. From the bar plots of categorical variables in the Stroke Prediction Dataset:
6. **Gender**:
    - The bar plot shows the distribution of genders among individuals in the dataset.
    - It indicates whether the dataset is balanced or skewed towards a particular gender.
7. **Ever Married**:
    - The bar plot illustrates the proportion of individuals who are married versus unmarried.
    - It helps understand the marital status distribution in the dataset.
8. **Work Type**:
    - The bar plot displays the distribution of individuals across different types of work.
    - It provides insights into the employment status and occupation diversity among individuals.
9. **Residence Type**:
    - The bar plot compares the number of individuals living in urban versus rural areas.
    - It helps understand the geographic distribution of individuals in the dataset.
10. **Smoking Status**:
    - The bar plot depicts the distribution of individuals based on their smoking status.
    - It shows the prevalence of different smoking habits among individuals.

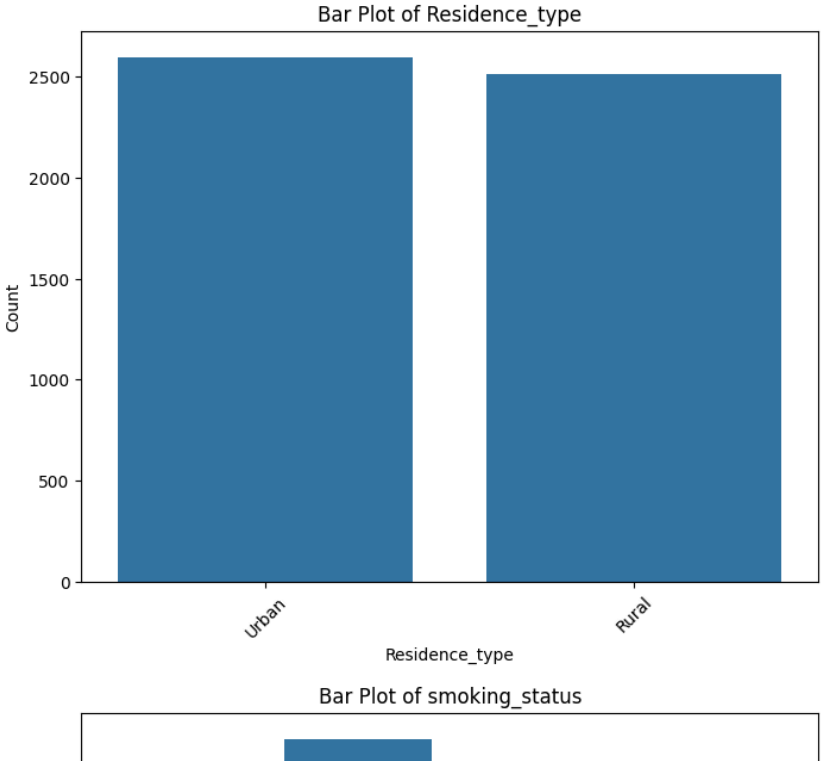
These bar plots provide a visual representation of the categorical variables in the dataset, helping identify patterns, imbalances, or trends in the data distribution.

Inferece:

1. A slight majority of individuals are female (58.59%), with a significant proportion being male (41.39%).
2. Most individuals are married (65.62%).
3. The majority work in the private sector (57.24%), followed by self-employment (16.03%).
4. Residence type is almost evenly split between urban (50.80%) and rural (49.20%).
5. The smoking status is diverse, with a significant proportion having never smoked (37.03%), followed by unknown (30.22%), formerly smoked (17.32%), and currently smoking (15.44%).

```
# Visualize using bar plots
for col in categorical_cols:
    plt.figure(figsize=(8, 6))
    sns.countplot(x=col, data=df)
    plt.title(f'Bar Plot of {col}')
    plt.xlabel(col)
    plt.ylabel('Count')
    plt.xticks(rotation=45)
    plt.show()
```


2000
1500
1000
500
0
Count
Private
Self-employed
Govt_job
children
Never_worked
work_type


Bar Plot of Residence_type
2500
2000
1500
1000
500
0
Count
Urban
Rural
Residence_type

Bar Plot of smoking_status

From the bar plots of categorical variables in the Stroke Prediction Dataset:

1. **Gender Distribution**:
    - There is a slight majority of females compared to males, with a small proportion categorized as 'Other'.
    - This indicates that the dataset is relatively balanced in terms of gender representation.
2. **Marital Status Distribution**:

- A larger proportion of individuals in the dataset are married ('Yes') compared to those who are not married ('No').
- This suggests that marital status could be an important factor in analyzing stroke risk.

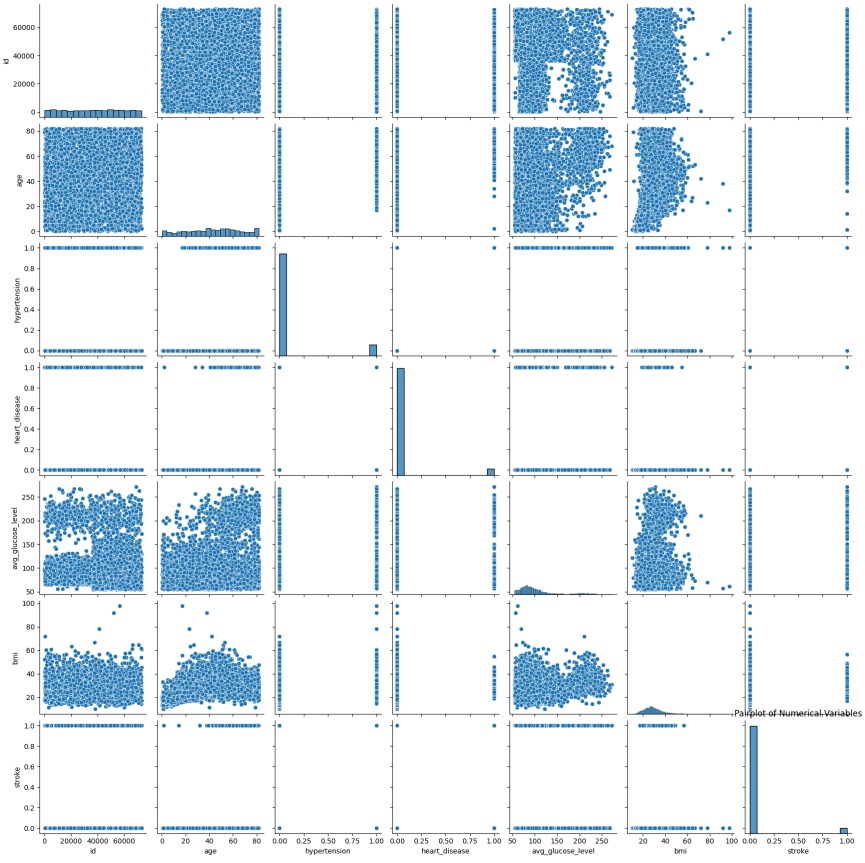
3. **Work Type Distribution**:
   - The most common work type is 'Private', followed by 'Self-employed', 'Children', and 'Govt_job'. Few individuals fall into the 'Never_worked' category.
   - This distribution provides insights into the employment status of individuals in the dataset.
4. **Residence Type Distribution**:
   - The dataset has a nearly equal distribution of individuals living in urban and rural areas.
   - This indicates that there is representation from both urban and rural populations in the dataset.
5. **Smoking Status Distribution**:
   - The majority of individuals either 'Never smoked' or have 'Unknown' smoking status.
   - Former and current smokers represent a smaller proportion of the dataset.
   - This distribution highlights the prevalence of non-smokers and individuals with unknown smoking status in the dataset.

These bar plots provide visual insights into the distribution of categorical variables, offering valuable information for further analysis and modeling related to stroke prediction.

Bivariate Analysis

Explore relationships between pairs of numerical variables using scatter plots or pair plots

```
# Pairplot for exploring pairwise relationships between numerical variables
numerical_cols = df.select_dtypes(include=['int64', 'float64']).columns
sns.pairplot(df[numerical_cols])
plt.title('Pairplot of Numerical Variables')
plt.show()
```



The pairplot allows us to visualize pairwise relationships between numerical variables in the dataset. Here are some inferences we can draw from the pairplot:

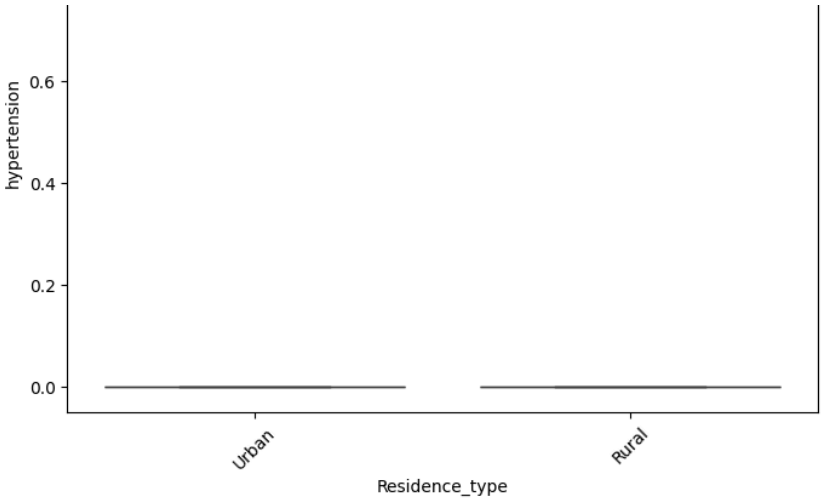
1. **Age vs. Average Glucose Level**:
    - There doesn't seem to be a strong linear relationship between age and average glucose level. However, there may be some clustering or patterns within certain age groups.
2. **Age vs. BMI**:
    - Similar to age vs. average glucose level, there doesn't appear to be a clear linear relationship between age and BMI. However, there may be clusters or trends within specific age ranges.
3. **Average Glucose Level vs. BMI**:
    - There might be a slight positive correlation between average glucose level and BMI, indicating that individuals with higher glucos levels tend to have higher BMIs. However, the relationship is not extremely strong.

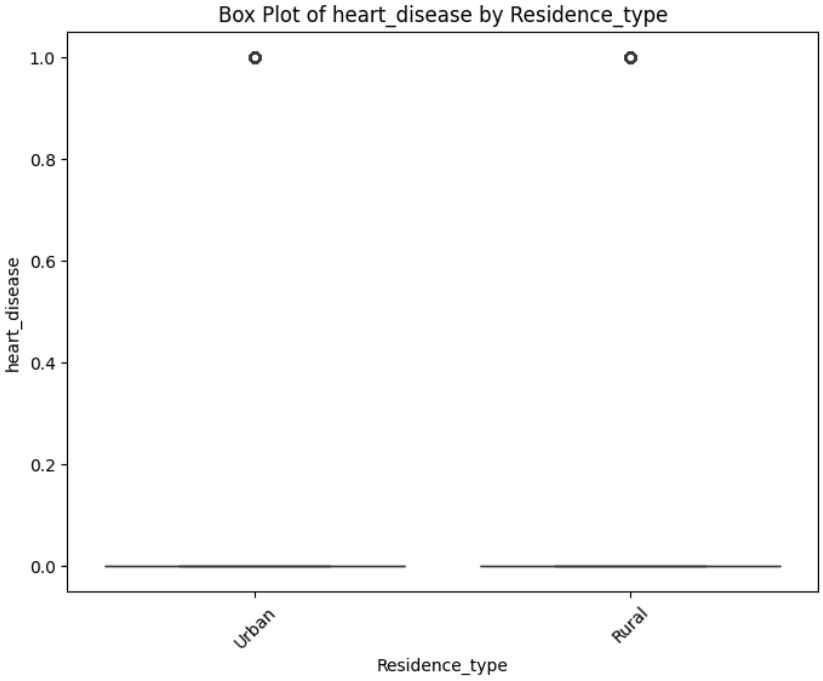
4. **Age vs. Stroke**:
    - We might observe differences in age distributions between individuals who have had a stroke and those who haven't. For example, there may be a higher concentration of older individuals among those who have had a stroke.
5. **Average Glucose Level vs. Stroke** and **BMI vs. Stroke**:
    - Similar to age, there may be differences in the distributions of average glucose level and BMI between individuals who have had a stroke and those who haven't. These differences could indicate potential risk factors associated with stroke.

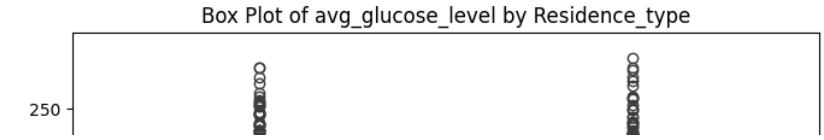
Overall, the pairplot provides a visual overview of the relationships between numerical variables in the dataset, helping identify potential patterns, trends, and correlations. Further analysis, including correlation coefficients and statistical tests, can provide more insights into the strength and significance of these relationships.

Explore relationships between numerical and categorical variables using box plots or violin plots

```
# Box plots to visualize the relationship between numerical and categorical variables
categorical_cols = df.select_dtypes(include=['object']).columns

for cat_col in categorical_cols:
    for num_col in numerical_cols:
        plt.figure(figsize=(8, 6))
        sns.boxplot(x=cat_col, y=num_col, data=df)
        plt.title(f'Box Plot of {num_col} by {cat_col}')
        plt.xlabel(cat_col)
        plt.ylabel(num_col)
        plt.xticks(rotation=45)
        plt.show()
```


hypertension
0.6
0.4
0.2
0.0
Urban
Rural
Residence_type


Box Plot of heart_disease by Residence_type
heart_disease
1.0
0.8
0.6
0.4
0.2
0.0
Urban
Rural
Residence_type


Box Plot of avg_glucose_level by Residence_type
250