LAPORAN AKHIR
PROYEK PEMBELAJARAN MESIN

STROKE PREDICTION
USING LOGISTIC REGRESSION

Disusun Oleh:

11419011	Edwin Immanuel Damanik
11419019	Hepniwer N A Purba
11419027	Santo Lamsar Harianja
11419058	Meyliza Veronica br Siregar
11419068	Geby W P Lumban Gaol

INSTITUT TEKNOLOGI DEL

SITOLUAMA - LAGUBOTI

TA 2021/2022

DAFTAR ISI

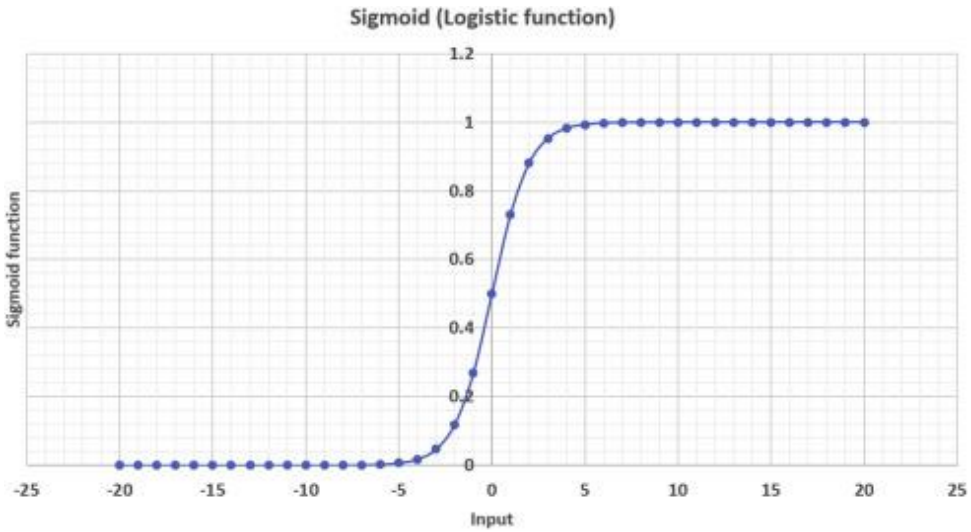
DAFTAR GAMBAR

BAB 1. PENDAHULUAN

1.1. Pengertian Regresi Logistik

Regresi Logistik adalah analisis regresi yang tepat untuk dilakukan ketika variabel dependen bersifat dikotomis (biner). Skala dikotomi yang dimaksud adalah skala data nominal dengan dua kategori, misalnya: Ya dan Tidak, Baik dan Buruk atau Tinggi dan Rendah. Seperti semua analisis regresi, regresi logistik adalah analisis prediktif. Regresi logistik digunakan untuk menggambarkan data dan untuk menjelaskan hubungan antara satu variabel biner dependen (apa yang ingin kita prediksi) dan satu atau lebih variabel independen (fitur) nominal, ordinal, interval atau rasio.

Fungsi sigmoid dan fungsi logit adalah beberapa variasi dari fungsi logistik. Fungsi logit adalah kebalikan dari fungsi logistik standar.



Gambar 1. Regresi logistik diterapkan pada kisaran 20 hingga 20

Sumber : https://www.sciencedirect.com/topics/computer-science/logistic-regression

Dalam persamaan fungsi logistik, *x* adalah variabel input. Jika nilai 20 hingga 20 dimasukkan ke dalam fungsi logistik, seperti yang diilustrasikan pada Gambar 1, maka input telah ditransfer ke antara 0 dan 1.

1.2. Tujuan

Tujuan dari pengolahan data ini adalah sebagai berikut:

- Melihat pengaruh variabel independen dari Stroke Prediction Dataset yaitu gender, age, hypertension, heart disease, ever married, work type, residence type, average glucose level, body mass index (BMI), dan smoking status terhadap variabel dependen yaitu stroke status.
- Memperoleh model regresi logistik untuk resiko terjangkit penyakit stroke.

1.3. Lingkungan Pekerjaan

Lingkungan pekerjaan yang dimaksudkan adalah spesifikasi software dan hardware ataupun tools yang digunakan dalam pengerjaan proyek ini. Dalam pengerjaan proyek ini, bahasa pemrograman yang digunakan adalah bahasa pemrograman python. Dibawah ini akan dijelaskan spesifikasi software dan hardware yang digunakan:

Spesifikasi hardware yang digunakan dalam pengerjaan proyek adalah sebagai berikut:

- Laptop : ASUS, LENOVO
- Processor : Intel Core i5
- RAM : 8GB

Spesifikasi software yang digunakan dalam pengerjaan proyek adalah sebagai berikut:

- Sistem Operasi : Windows 10
- Integrated Development Environment (IDE) : Jupyter Notebook

BAB 2. PREPROCESSING DATA

2.1. Missing Data

Missing data merupakan data atau informasi yang hilang atau tidak tersedia mengenai subjek penelitian pada variabel tertentu akibat faktor non sampling error. Missing data terjadi apabila ada salah satu data yang memiliki nilai kevalidan tidak tersedia atau hilang pada saat kita akan melakukan analisa.

```
In [4]: #PREPROCESSING DATA
        #1. Missing Data
        data.isnull().sum()

Out[4]: 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       0
        stroke               0
        dtype: int64
```

Gambar 2. Missing Data

Gambar 2 merupakan kode yang digunakan untuk melihat apakah terdapat missing data pada dataset *healthcare-dataset-stroke-data.csv*. Sedangkan untuk menghapus missing data yang terdapat pada dataset stroke prediction, digunakan code seperti yang terdapat pada Gambar 3.

```
In [5]: #Drop missing values
        modifiedDataset = data.dropna()

In [6]: modifiedDataset.isnull().sum()

Out[6]: 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
```

Gambar 3. Drop Missing Values

2.2. Data Formatting

Data formatting berfungsi untuk melakukan pengecekan terhadap tipe dan distribusi data pada dataset.

```
In [12]: #2. Data Formatting
         data.dtypes

Out[12]: 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
```

Gambar 4. Data Formatting

Gambar 4 merupakan kode yang digunakan untuk mengecek tipe dan distribusi dataset *healthcare-dataset-stroke-data.csv*.

2.3. Normalisasi Data

Normalisasi data adalah proses membuat beberapa variabel memiliki rentang nilai yang sama, tidak ada yang terlalu besar maupun terlalu kecil. Tujuan normalisasi adalah untuk mengubah nilai kolom numerik dalam kumpulan data ke skala yang sama, tanpa mengganggu perbedaan dalam rentang nilai.

Metode normalisasi data yang digunakan dalam dataset *healthcare-dataset-stroke-data.csv* adalah metode Min-Max. Metode Min-Max bekerja dengan cara mengurangi setiap nilai suatu fitur dengan nilai minimum fitur tersebut dan membaginya dengan range nilai atau nilai maksimum dikurangi nilai minimum fitur tersebut.
Penerapan normalisasi data pada program python yang telah dilakukan :

```
In [13]: #3. Normalisasi Data
from sklearn import preprocessing
import pandas as pd
col_list = ["age", "hypertension", "heart_disease", "avg_glucose_level", "bmi", "stroke"]
stroke = pd.read_csv("healthcare-dataset-stroke-data.csv", usecols=col_list)
scaler = preprocessing.MinMaxScaler()
names = stroke.columns
d = scaler.fit_transform(stroke)
scaled_df = pd.DataFrame(d, columns = names)
scaled_df.head()
```

Out[13]:

	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
0	0.816895	0.0	1.0	0.801265	0.301260	1.0
1	0.975586	0.0	1.0	0.234512	0.254296	1.0
2	0.597168	0.0	0.0	0.536008	0.276060	1.0
3	0.963379	1.0	0.0	0.549349	0.156930	1.0
4	0.987793	0.0	0.0	0.605161	0.214204	1.0

Gambar 5. Normalisasi Data

2.4. Outlier

Outlier adalah kasus atau data yang memiliki karakteristik unik yang terlihat sangat berbeda jauh dari observasi-observasi lainnya dan muncul dalam bentuk nilai ekstrim baik untuk sebuah variabel tunggal atau variabel kombinasi.

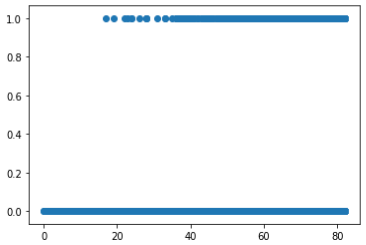
Untuk menentukan outlier pada dataset Stroke Prediction digunakan dengan membuat visualisasi data kedalam bentuk scatter plot seperti code pada gambar di bawah.

```
In [40]: #Outlier
         import matplotlib.pyplot as plt
         import seaborn as sns
         %matplotlib inline

In [41]: feature_columns = ['age', 'hypertension', 'heart_disease', 'avg_glucose_level', 'bmi']
         X = data[feature_columns].values
         y = data['stroke'].values

In [42]: X

Out[42]: array([[ 67.  ,   0.  ,   1.  , 228.69,  36.6 ],
                [ 80.  ,   0.  ,   1.  , 105.92,  32.5 ],
                [ 49.  ,   0.  ,   0.  , 171.23,  34.4 ],
                ...,
                [ 35.  ,   0.  ,   0.  ,  82.99,  30.6 ],
                [ 51.  ,   0.  ,   0.  , 166.29,  25.6 ],
                [ 44.  ,   0.  ,   0.  ,  85.28,  26.2 ]])
```

Gambar 6. Outlier (1)

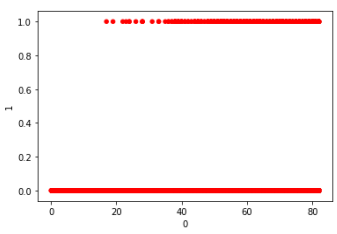
Tampilan outlier pada program adalah sebagai berikut :

```
In [44]: from numpy import where
         from matplotlib import pyplot
         pyplot.scatter(X[:,0],X[:,1])
         pyplot.show()
```



Gambar 7. Outlier (2)

```
In [48]: df['key1'] = (y)
         colors = np.where(df["key1"]==0, 'r', 'g')
         print(colors)
         df.plot.scatter(x = 0, y = 1, c = colors)
         plt.show()

['g' 'g' 'g' ... 'r' 'r' 'r']
```



Gambar 8. Outlier (3)

BAB 3. EKSPLORASI DATA

Menurut Organisasi Kesehatan Dunia (WHO) stroke adalah penyebab kematian ke-2 secara global, dan merupakan penyebab kematian sebanyak 11% dari total keseluruhan kematian.

Dataset *healthcare-dataset-stroke-data.csv* digunakan untuk memprediksi kemungkinan pasien terkena stroke berdasarkan parameter input seperti jenis kelamin, usia, berbagai penyakit, dan status merokok. Setiap baris dalam data memberikan informasi yang relevan tentang pasien.

Informasi Atribut :

1) id: identifier unik.
2) gender: "Male", "Female" atau "Other".
3) age: usia pasien.
4) hypertension: 0 jika pasien tidak memiliki hipertensi, dan 1 jika pasien memiliki hipertensi.
5) heart_disease: 0 jika pasien tidak memiliki penyakit jantung, dan 1 jika pasien memiliki penyakit jantung.
6) ever_married: "No" atau "Yes".
7) work_type: "children", "Govt_jov", "Never_worked", "Private" atau "Self-employed".
8) Residence_type: "Rural" atau "Urban".
9) avg_glucose_level: tingkat glukosa rata-rata dalam darah.
10) bmi: indeks massa tubuh.
11) smoking_status: "formerly smoked", "never smoked", "smokes" atau "Unknown"*.

    Note: "Unknown" pada smoking_status berarti informasi tersebut tidak tersedia untuk pasien ini.

12) stroke: 0 jika pasien tidak mengalami stroke, dan 1 jika pasien mengalami stroke.

Jumlah data pada dataset *healthcare-dataset-stroke-data.csv* adalah sebanyak 5110 data dan terdapat 12 atribut di dalamnya Dari 5110 data tersebut, yang digunakan adalah 4909 data karena sudah dilakukan preprocessing data pada missing data. 4909 data dibagi menjadi subset pelatihan dan pengujian. 80% dari instance yaitu sebanyak 3927 data akan ditugaskan untuk pelatihan, dan 20% yaitu sebanyak 982 data akan digunakan untuk pengujian.

3.1. Confusion Matrix

Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah yang merupakan representasi hasil proses klasifikasi pada confusion matrix yaitu *True Positif, True Negatif, False Positif, dan False Negatif.*

Gambar 9 merupakan code dan hasil dari confusion matrix pada dataset *healthcare-dataset-stroke-data.csv.*

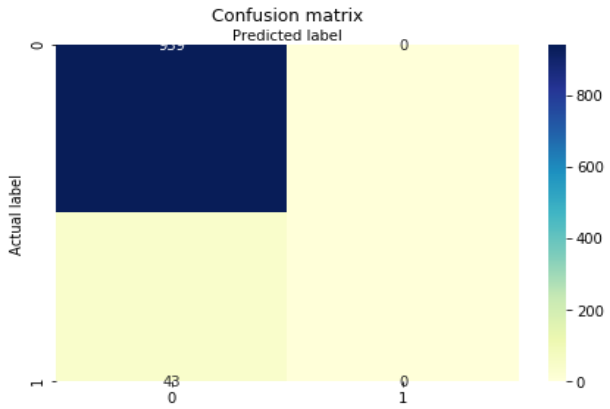
```
In [54]: #import the metrics class
         from sklearn import metrics
         cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
         cnf_matrix

Out[54]: array([[939,   0],
                [ 43,   0]], dtype=int64)
```

Gambar 9. Confusion Matrix

Dilakukan visualisasi pada model yang telah dihasilkan sehingga memudahkan dalam mengevaluasi. Hasil visualisasi yang diperoleh berdasarkan model dapat dilihat pada Gambar 11.

```
In [56]: #Confussion Matrix
         class_names=[0,1]
         fig, ax = plt.subplots()
         tick_marks = np.arange(len(class_names))
         plt.xticks(tick_marks, class_names)
         plt.yticks(tick_marks, class_names)

         sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
         ax.xaxis.set_label_position("top")
         plt.tight_layout()
         plt.title('Confusion matrix', y=1.1)
         plt.ylabel('Actual label')
         plt.xlabel('Predicted label')
```

Gambar 10. Code Visualisasi Confusion Matrix



Gambar 11. Visualisasi Confusion Matrix

Berdasarkan confusion matrix dan visualisasi dari model yang telah diperoleh dalam hasil pengujian, maka hasil yang diperoleh:

- TP (True Positive)
  Terdapat 939 hasil yang menunjukkan data positif dan diprediksi benar.
- FP (False Positive)

  Terdapat 0 hasil yang menunjukkan data negatif dan diprediksi sebagai data positive.
- FN (False Negative)

  Terdapat 43 hasil yang menunjukkan data positif dan diprediksi sebagai data negative.
- TN (True Negative)

  Terdapat 0 yang menunjukkan data negatif dan diprediksi benar.

Dari confusion matrix yang telah diperoleh, dapat diketahui keakuratan dari model yang telah dibuat dengan performance matrix seperti: accuracy, recall, dan precision.

	precision	recall	f1-score	support
0	0.96	1.00	0.98	939
1	0.00	0.00	0.00	43
accuracy			0.96	982
macro avg	0.48	0.50	0.49	982
weighted avg	0.91	0.96	0.93	982

Gambar 12. Performance Matrix

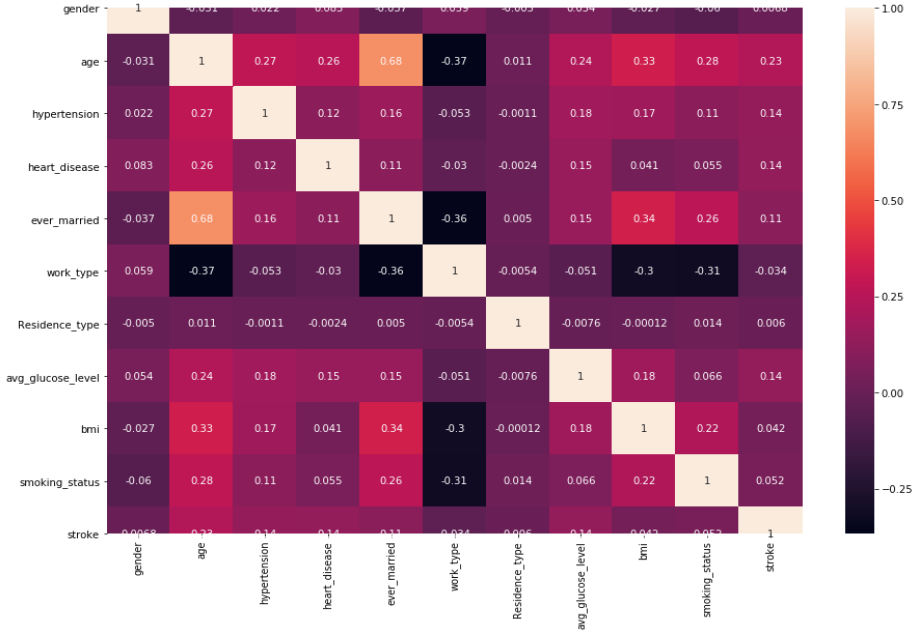
3.2. Correlation

Matrix korelasi adalah sebuah matrix yang digunakan untuk mengetahui ketergantungan antara variabel yang berkelipatan pada saat yang bersamaan. Hasil dari tabel tersebut mengandung koefisien korelasi di antara para variabel lainnya.

Pada code program Gambar 13, digunakan argumen *annot=**True*** untuk menampilkan korelasi antar atribut. Jika nilai korelasi mendekati 1 maka hubungan antar atribut semakin tinggi.

```
In [37]: #Correlation
         plt.figure(figsize=(15,10))
         sns.heatmap(data.corr(method='pearson'), annot=True)
```

Gambar 13. Code Matrix Correlation



Gambar 14. Matrix Correlation

Berdasarkan matrix correlation yang dihasilkan pada Gambar 14, beberapa variabel yang menunjukkan korelasi yang efektif adalah: age, hypertension, heart_disease, ever_married, avg_glucose_level.

Orang yang berusia lebih dari 60 tahun cenderung mengalami stroke. Beberapa outlier juga dapat dilihat bahwa orang yang berusia di bawah 20 tahun mengalami stroke. Pengamatan lain adalah orang yang tidak mengalami stroke juga terdiri dari orang yang berusia > 60 tahun.

BAB 4. KODE PROGRAM

4.1. Import Library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

4.2. Import Dataset

```
data = pd.read_csv('healthcare-dataset-stroke-data.csv')
```

4.3. Missing Data

```
#1. Missing Data
data.isnull().sum()

#Drop missing values
modifiedDataset = data.dropna()

modifiedDataset.isnull().sum()

modifiedDataset.to_csv('healthcare-dataset-stroke-
data.csv',index=False)

data=data.dropna()

data.dropna(axis = 1)

# Dropping ID Column
data=data.drop(['id'],axis=1)
```

4.4. Data Formatting

```
#2. Data Formatting
data.dtypes
```

4.5. Normalisasi Data

```
#3. Normalisasi Data
from sklearn import preprocessing
import pandas as pd
col_list = ["age", "hypertension", "heart_disease",
"avg_glucose_level", "bmi", "stroke"]
stroke = pd.read_csv("healthcare-dataset-stroke-data.csv",
usecols=col_list)
scaler = preprocessing.MinMaxScaler()
names = stroke.columns
d = scaler.fit_transform(stroke)
scaled_df = pd.DataFrame(d, columns = names)
scaled_df.head()
```

4.6. Jumlah Penderita Stroke

```
#Jumlah Penderita Stroke
print('Unique Value\n',data['stroke'].unique())
print('Value Counts\n',data['stroke'].value_counts())

sns.countplot(data=data,x='stroke')
```

4.7. Value pada Gender dan Jumlahnya

```
#Value pada gender dan jumlahnya
print('Unique values\n',data['gender'].unique())
print('Value Counts\n',data['gender'].value_counts())

sns.countplot(data=data,x='gender')
```

4.8. Jumlah Penderita Stroke berdasarkan Gender

```
#Jumlah Penderita Stroke berdasarkan gender
sns.countplot(data=data,x='gender',hue='stroke')
```

4.9. Label Encoding

```
#4. Label Encoding
from sklearn.preprocessing import LabelEncoder
cols=data.select_dtypes(include=['object']).columns
print(cols)
# This code will fetech columns whose data type is object.
le=LabelEncoder()
# Initializing our Label Encoder object
data[cols]=data[cols].apply(le.fit_transform)
# Transfering categorical data into numeric
print(data.head(10))
```

4.10. Correlation

```
#Correlation
plt.figure(figsize=(15,10))
sns.heatmap(data.corr(method='pearson'), annot=True)
```

4.11. Outlier

```
#Outlier
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

feature_columns = ['age', 'hypertension', 'heart_disease',
'avg_glucose_level', 'bmi']
X = data[feature_columns].values
y = data['stroke'].values

from numpy import where
```

```
from matplotlib import pyplot
pyplot.scatter(X[:,0],X[:,1])
pyplot.show()
```

4.12. Splitting Data

```
#Splitting Data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.2, random_state = 0)
```

4.13. Logistic Regression

```
#Logistic Regression
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)
```

4.14. Import The Metric Class

```
#import the metrics class
from sklearn import metrics
cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
cnf_matrix
```

4.15. Accuracy

```
from sklearn.metrics import confusion_matrix, accuracy_score
accuracy = accuracy_score(y_test, y_pred) * 100
print('Accuracy of our model is equal ' +
str(round(accuracy, 2)) + ' %. ')
```

4.16. Classification Report

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

4.17. ROC

```
from sklearn.metrics import roc_auc_score, auc, roc_curve,
recall_score

#ROC
predicted_probab_log = classifier.predict_proba(X_test)
predicted_probab_log = predicted_probab_log[:, 1]
fpr, tpr, _ = roc_curve(y_test, predicted_probab_log)
```

4.18. AUC

```
auc = roc_auc_score(y_test, predicted_probab_log)
print('AUC: %.2f' % auc)
```

```
from matplotlib import pyplot
pyplot.plot(fpr, tpr, marker='.', label='Logistic
Regression')
pyplot.xlabel('False Positive Rate')
pyplot.ylabel('True Positive Rate')
pyplot.legend()
pyplot.show()
```

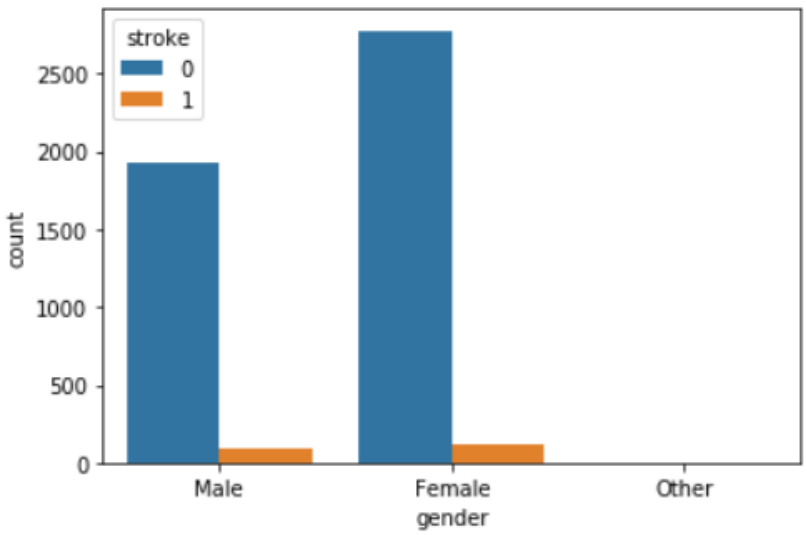
BAB 5. ANALISIS OUTPUT DAN HASIL

Dataset healthcare-dataset-stroke-data memiliki jumlah data sebanyak 5110 data dan terdapat 12 atribut di dalamnya. Dari 5110 data tersebut, yang digunakan adalah 4909 data karena sudah dilakukan preprocessing data pada missing data. Data tersebut memiliki label Stroke Status yang memungkinkan klasifikasi data dengan *unique value* yaitu "0" menandakan pasien stroke dan "1" menandakan pasien tidak stroke. Dari jumlah 4909 data pasien diketahui sebanyak 209 pasien merupakan penderita stroke dan sebanyak 4700 pasien tidak menderita stroke.

Hasil analisis pada data pasien berdasarkan tiap faktor resiko adalah sebagai berikut:

5.1. Berdasarkan Gender

Jumlah pasien jika dikelompokkan berdasarkan jenis kelamin maka diketahui sebanyak 2897 pasien berjenis kelamin perempuan dan sebanyak 2011 pasien berjenis kelamin laki-laki. Ketika dianalisis jumlah pasien penderita stroke berjenis kelamin wanita lebih banyak dibandingkan jumlah pasien penderita stroke berjenis kelamin laki-laki. Hal ini dapat kita lihat pada grafik berikut ini:

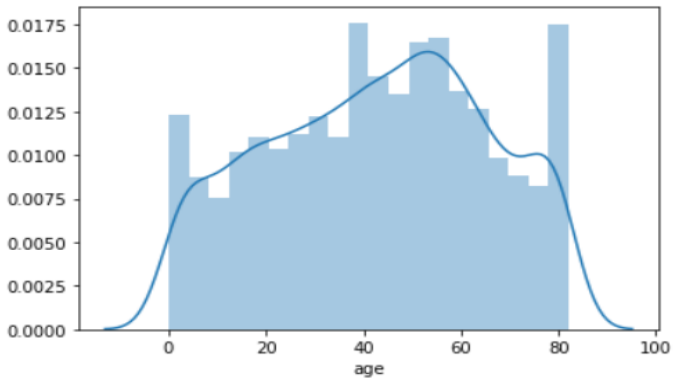


Gambar 15. Penyakit Stroke Berdasarkan Gender

Pada grafik di atas, bagan berwarna biru merupakan jumlah pasien tidak menderita stroke dan bagan berwarna orange adalah jumlah pasien yang menderita stroke. Dari grafik tersebut dapat dilihat bahwa bagan berwarna orange (jumlah penderita stroke) pada wanita lebih tinggi jika dibandingkan dengan bagan berwarna orange pada pria.

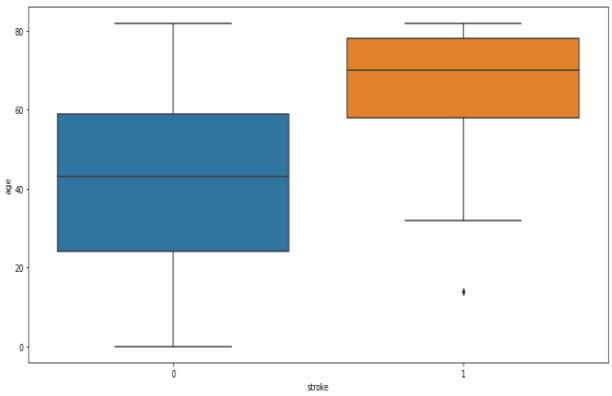
5.2. Berdasarkan Age

Pasien dikelompokkan dari berbagai kalangan usia. Mulai dari kalangan usia 0 tahun sampai dengan usia 80 tahun. Ketika dilakukan analisis, pasien dengan kalangan usia 40 tahun ke atas lebih banyak atau lebih rentan terkena stroke. Hal tersebut dapat dilihat dari grafik di bawah ini:



Gambar 16. Plot Distribusi Age

Pada grafik tersebut, dapat dilihat bahwa pasien berusia 40 sampai 80 tahun, mencapai titik 0,0175. Dimana jumlah penderita stroke lebih banyak di usia 40 tahun ke atas dibandingkan dengan 40 tahun ke bawah.

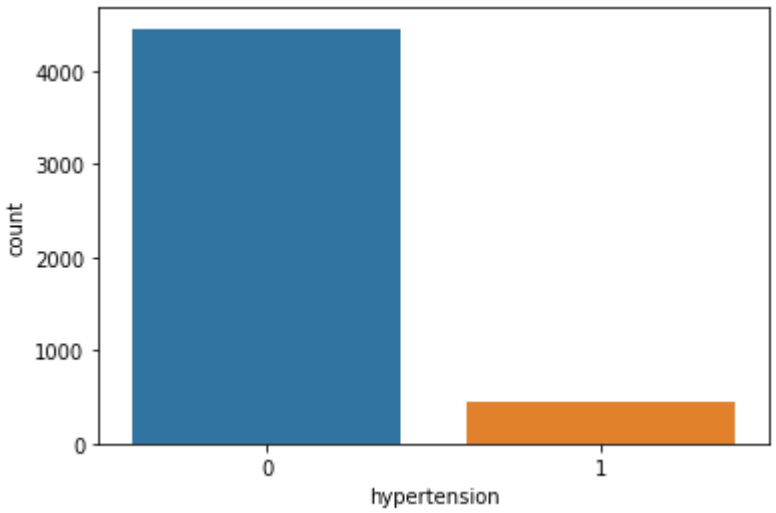


Gambar 17. Boxplot Penyakit Stroke Berdasarkan Age

Berdasarkan plot yang dihasilkan, dapat dilihat bahwa orang yang berusia lebih dari 60 tahun cenderung mengalami stroke. Dapat dilihat juga bahwa beberapa orang yang berusia di bawah 20 tahun mengalami stroke. Pengamatan lain adalah orang yang tidak mengalami stroke juga terdiri dari orang yang berusia > 60 tahun.

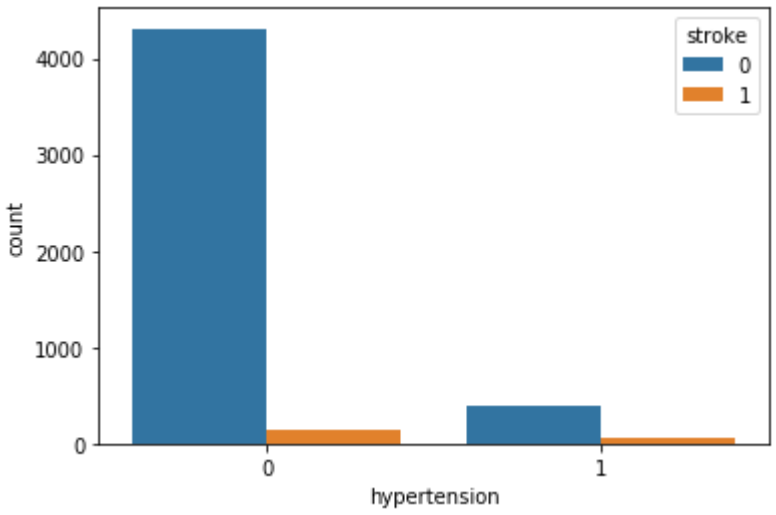
5.3. Berdasarkan Hypertension

Pasien dapat diklasifikasikan berdasarkan hypertension. Jumlah Pasien berdasarkan hypertension yaitu 4458 pasien untuk pasien yang tidak memiliki hipertensi, dan 451 pasien untuk pasien yang memiliki hipertensi.



Gambar 18. Jumlah Pasien dengan Hypertension

Hipertensi adalah suatu kondisi ketika seseorang memiliki tekanan darah tinggi.

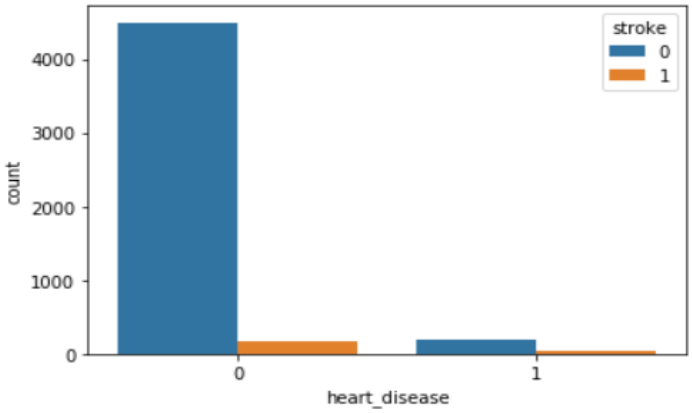


Gambar 19. Penyakit Stroke Berdasarkan Hypertension

Berdasarkan grafik pada Gambar, hipertensi juga dapat menyebabkan stroke. Pada data yang diambil, stroke lebih banyak dialami oleh pasien yang tidak memiliki hipertensi daripada pasien yang memiliki hipertensi.

5.4. Berdasarkan Heart Disease

Jumlah penderita yang terkena stroke akibat dari penyakit jantung yang dimiliki sebelumnya lebih sedikit dibandingkan dengan yang tidak memiliki riwayat penyakit jantung. Hal tersebut dapat dilihat berdasarkan grafik di bawah ini:

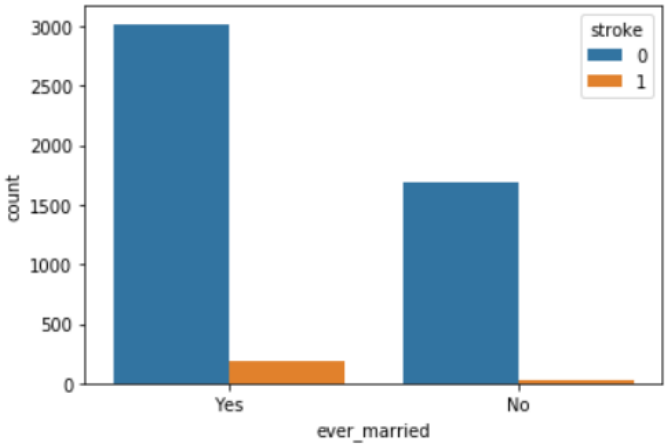


Gambar 20. Penyakit Stroke Berdasarkan Heart Disease

Pada grafik digambarkan bahwa untuk diagram berwarna biru merupakan jumlah pasien yang tidak terkena stroke karena penyakit jantung. Dapat dilihat jumlahnya mencapai lebih dari 4000. Sementara untuk pasien yang terkena stroke karena penyakit jantung yang sudah dimiliki sebelumnya sangatlah sedikit.

5.5. Berdasarkan Ever Married

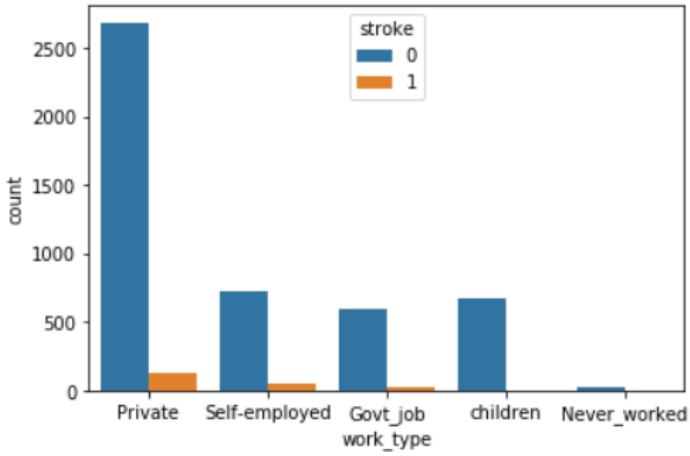
Jumlah pasien yang terkena stroke saat sudah menikah lebih banyak dibandingkan dengan pasien yang belum menikah. Kemudian jumlah pasien yang tidak terkena stroke pada saat sudah menikah lebih banyak daripada pasien yang belum menikah. Hal tersebut dapat dilihat pada grafik di bawah ini:



Gambar 21. Penyakit Stroke Berdasarkan Ever Married

5.6. Berdasarkan Work Type

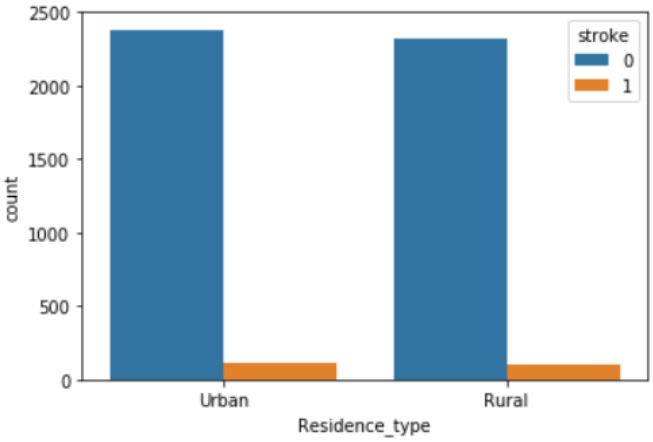
Untuk kategori pekerja private, penderita stroke sangat sedikit daripada yang tidak terkena stroke. Jumlah pasien yang tidak terkena stroke mencapai lebih dari 2500 pasien. Sementara untuk yang terkena stroke berkisar antara 10 sampai 50 pasien. Kemudian untuk kategori pekerja self employed, jumlah pasien yang tidak terkena stroke lebih banyak daripada yang terkena stroke. Sama halnya dengan kategori Govt Job. Kemudian untuk kategori children dan yang tidak pernah bekerja, tidak ada sama sekali pasien yang terkena stroke. Hal tersebut dapat dilihat pada grafik di bawah ini:



Gambar 22. Penyakit Stroke Berdasarkan Work Type

5.7. Berdasarkan Residence Type

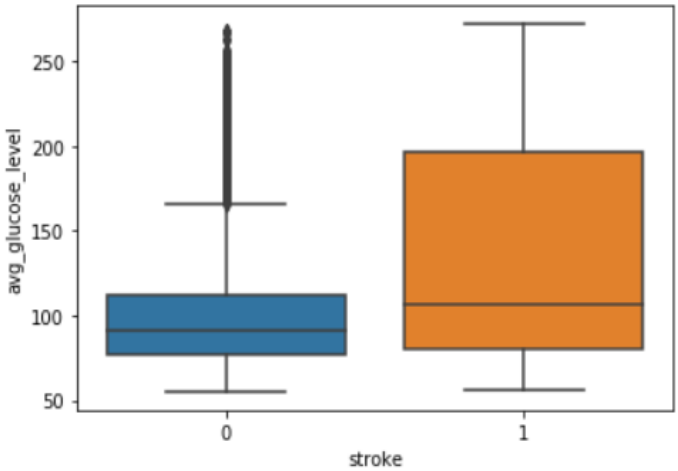
Jumlah pasien penderita stroke di perkotaan (urban) lebih tinggi dibandingkan jumlah pasien penderita stroke di pedesaan (rural). Hal ini dapat kita ketahui lebih jelas dengan melihat grafik di bawah ini:



Gambar 23. Penyakit Stroke Berdasarkan Residence Type

5.8. Berdasarkan Average Glucose Level

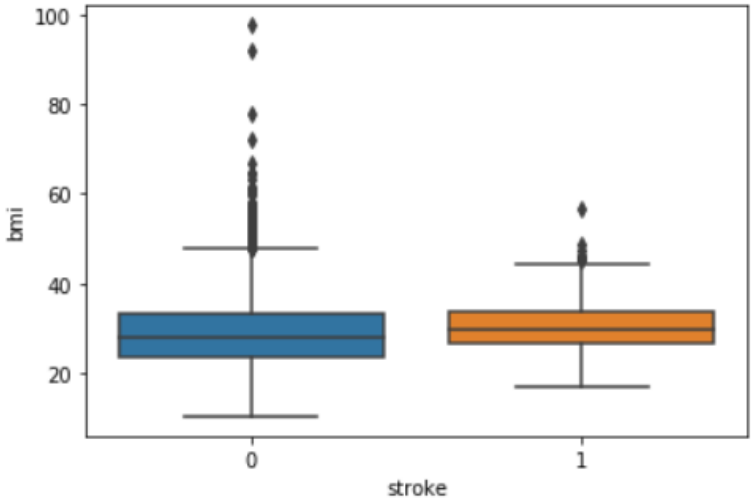
Jumlah pasien penderita stroke berdasarkan average glucose levelnya diketahui bahwa pasien yang mengalami stroke memiliki kadar glukosa rata-rata lebih dari 100. Dapat dilihat pada diagram di bawah ini:



Gambar 24. Boxplot Penyakit Stroke Berdasarkan Average Glucose Level

5.9. Berdasarkan Body Mass Index

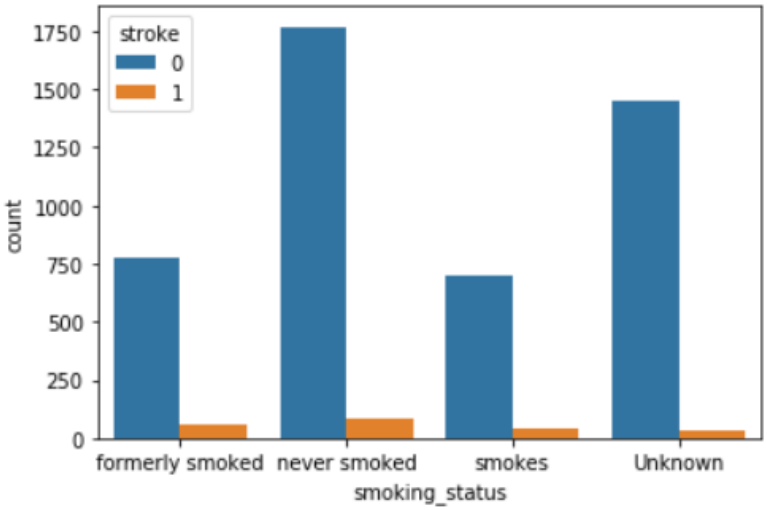
Dari data diketahui bahwa jumlah pasien penderita stroke paling banyak ada pada rentang usia 20 sampai 40 tahun. Dapat kita lihat pada diagram berikut ini:



Gambar 25. Penyakit Stroke Berdasarkan BMI

5.10. Berdasarkan Smoking Status

Jumlah pasien yang merokok ada 737 pasien, belum pernah merokok ada 1852 pasien, yang tidak lagi merokok ada sebanyak 837 pasien, dan yang tidak diketahui ada 1483 pasien. Setelah dianalisis pasien yang belum pernah merokok sekalipun juga dapat menderita penyakit stroke. Hal ini dapat kita lihat pada grafik dibawah ini:



Gambar 26. Penyakit Stroke Berdasarkan Smoking Status

Dari model regresi logistik yang diperoleh diketahui bahwa akurasi model adalah 95.62%. Hal ini dapat lebih jelas diketahui dari gambar dibawah ini:

```
from sklearn.metrics import confusion_matrix, accuracy_score
accuracy = accuracy_score(y_test, y_pred) * 100
print('Accuracy of our model is equal ' + str(round(accuracy, 2)) + ' %. ')
```

```
Accuracy of our model is equal 95.62 %.
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98       939
           1       0.00      0.00      0.00        43

    accuracy                           0.96       982
   macro avg       0.48      0.50      0.49       982
weighted avg       0.91      0.96      0.93       982
```

Gambar 27. Metrik kinerja dari klasifikasi "Regresi Logistik"

PEMBAGIAN KERJA

NIM	Nama	Pembagian Kerja
11419011	Edwin Immanuel Damanik	- Membuat kode program. - Membuat slide presentasi.
11419019	Hepniwer N A Purba	- Membuat kode program. - Membuat laporan akhir proyek.
11419027	Santo Lamsar Harianja	- Membuat kode program. - Membuat slide presentasi.
11419058	Meyliza Veronica Siregar	- Membuat kode program. - Membuat laporan akhir proyek.
11419068	Geby W P Lumban Gaol	- Membuat kode program. - Membuat laporan akhir proyek.

REFERENSI

Logistic regression. Logistic Regression - an overview | ScienceDirect Topics. (n.d.). Retrieved December 29, 2021, from https://www.sciencedirect.com/topics/computer-science/logistic-regression

Fedesoriano. (2021, January 26). *Stroke prediction dataset*. Kaggle. Retrieved December 29, 2021, from https://www.kaggle.com/fedesoriano/stroke-prediction-dataset