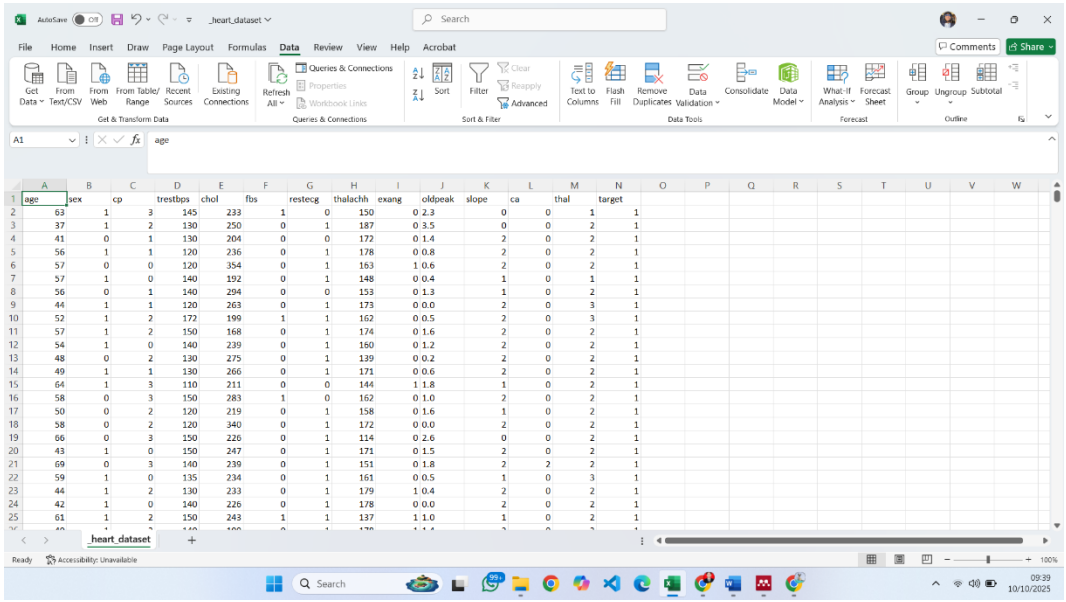
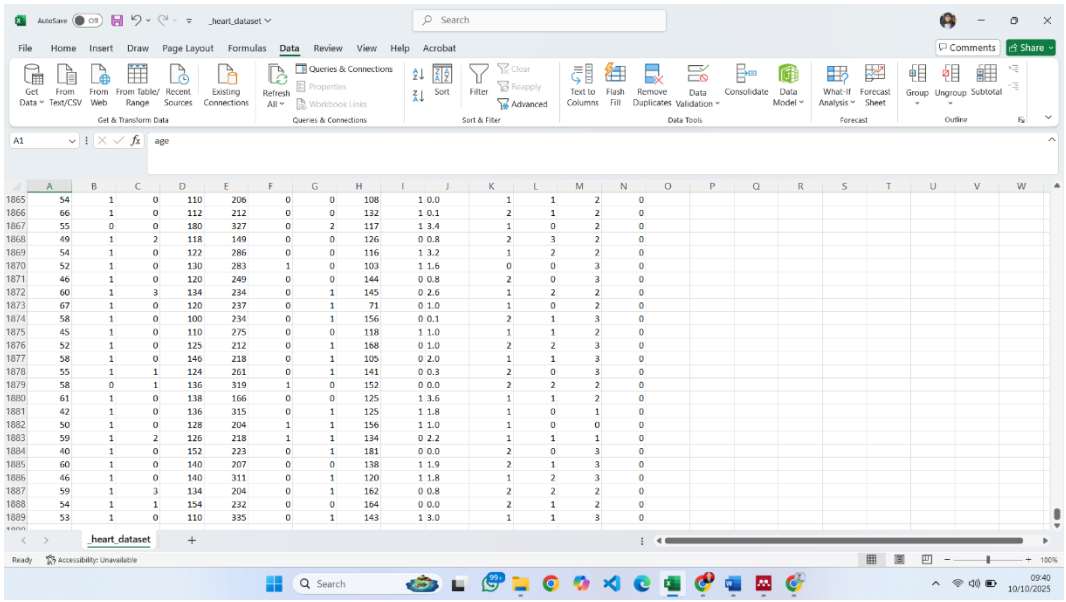
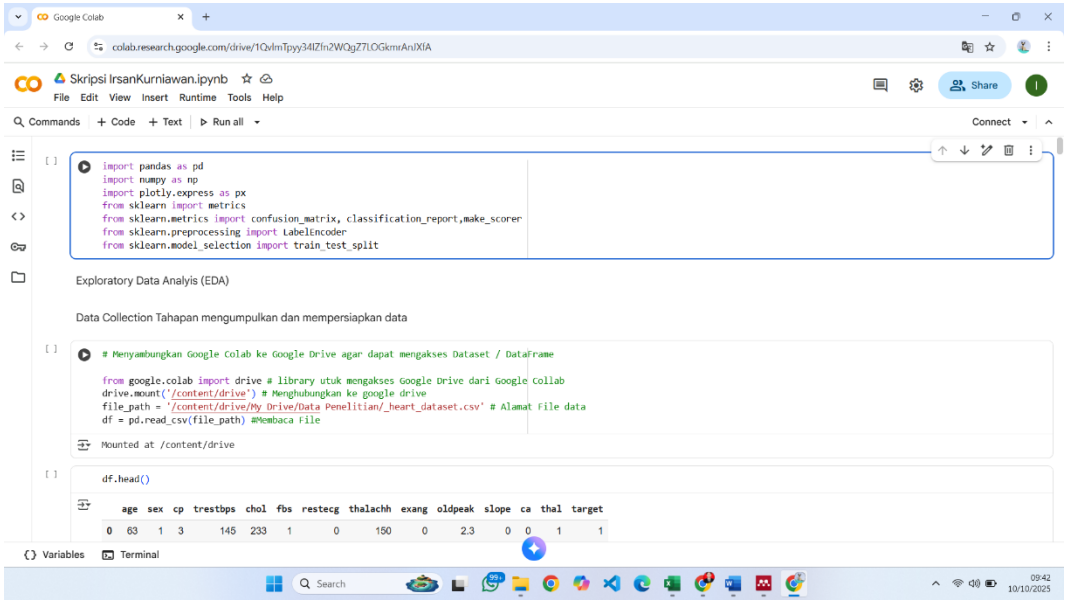
LAMPIRAN

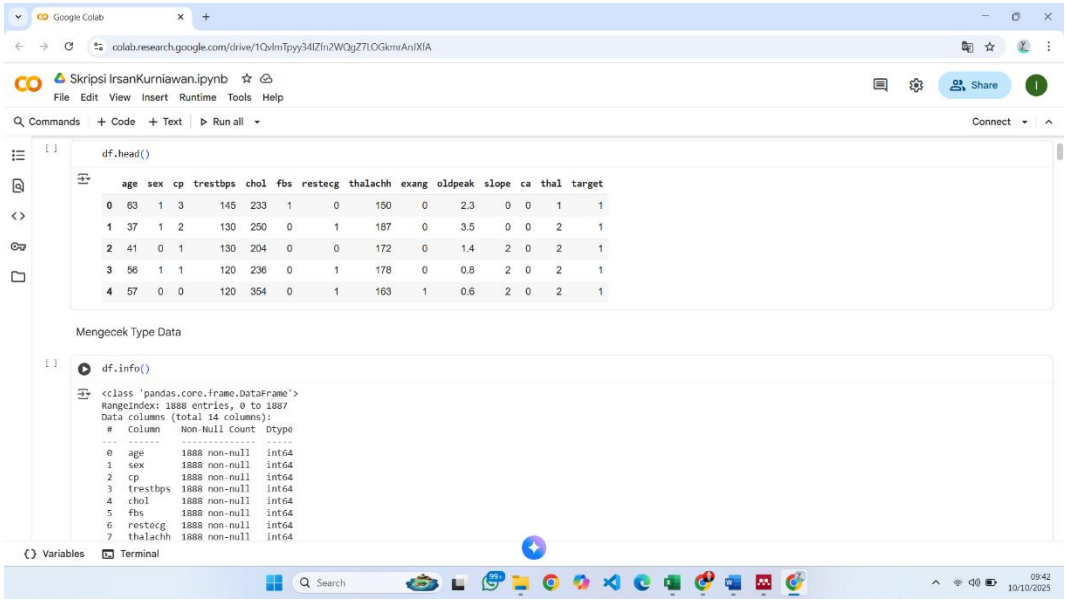


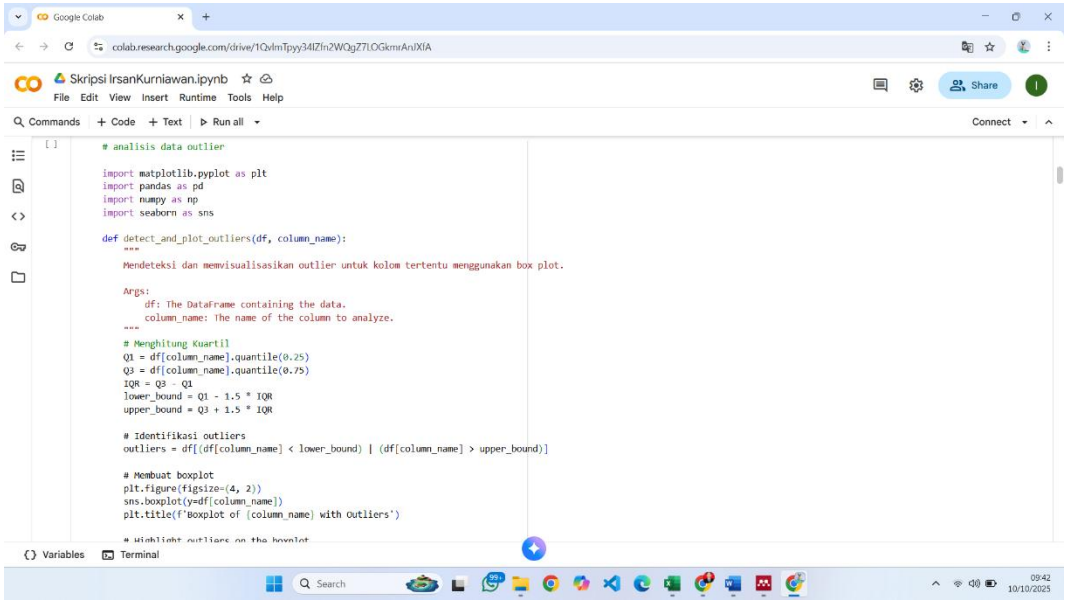
age	sex	cp	trestbps	chol	fbs	restecg	thalachh	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
58	0	3	150	283	1	0	162	0	1.0	2	0	2	1
50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
58	0	2	120	340	0	1	172	0	0.0	2	0	2	1
66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
69	0	3	140	239	0	1	151	0	1.8	2	2	2	1
59	1	0	135	234	0	1	161	0	0.5	1	0	3	1
44	1	2	130	233	0	1	179	1	0.4	2	0	2	1
42	1	0	140	226	0	1	178	0	0.0	2	0	2	1
61	1	2	150	243	1	1	137	1	1.0	1	0	2	1

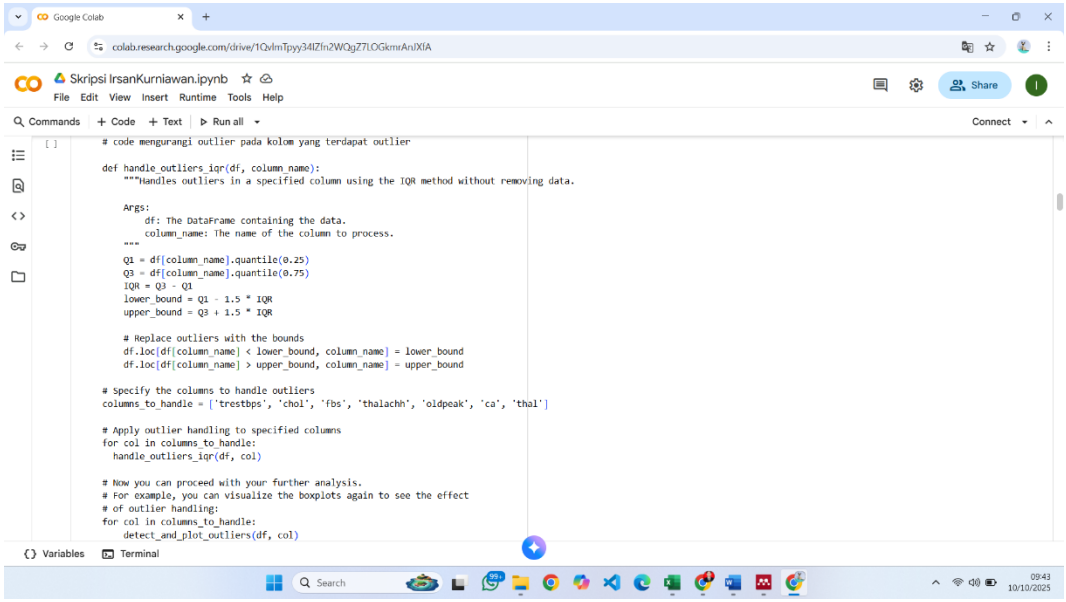


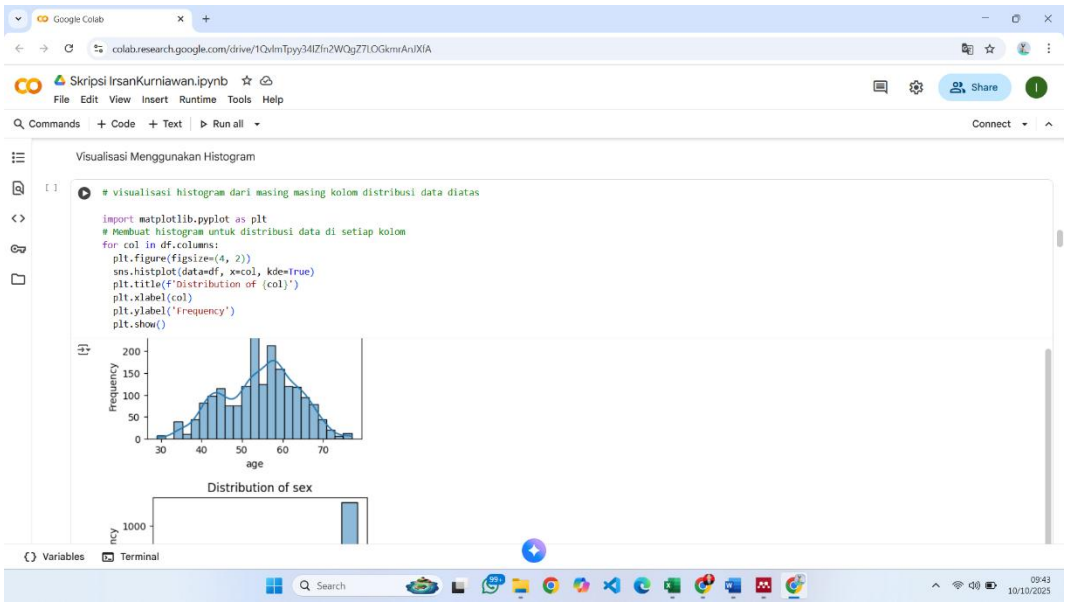
1865	54	1	0	110	206	0	0	108	1	0.0	1	1	2	0
1866	66	1	0	112	212	0	0	132	1	0.1	2	1	2	0
1867	55	0	0	180	327	0	2	117	1	3.4	1	0	2	0
1868	49	1	2	118	149	0	0	126	0	0.8	2	3	2	0
1869	54	1	0	122	286	0	0	116	1	3.2	1	2	2	0
1870	52	1	0	130	283	1	0	103	1	1.6	0	0	3	0
1871	46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
1872	60	1	3	134	234	0	1	145	0	2.6	1	2	2	0
1873	67	1	0	120	237	0	1	71	0	1.0	1	0	2	0
1874	58	1	0	100	234	0	1	156	0	0.1	2	1	3	0
1875	45	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1876	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1877	58	1	0	146	218	0	1	105	0	2.0	1	1	3	0
1878	55	1	1	124	261	0	1	141	0	0.3	2	0	3	0
1879	58	0	1	136	319	1	0	152	0	0.0	2	2	2	0
1880	61	1	0	138	166	0	0	125	1	3.6	1	1	2	0
1881	42	1	0	136	315	0	1	125	1	1.8	1	0	1	0
1882	50	1	0	128	204	1	1	156	1	1.0	1	0	0	0
1883	59	1	2	126	218	1	1	134	0	2.2	1	1	1	0
1884	40	1	0	152	223	0	1	181	0	0.0	2	0	3	0
1885	60	1	0	140	207	0	0	138	1	1.9	2	1	3	0
1886	46	1	0	140	311	0	1	120	1	1.8	1	2	3	0
1887	59	1	3	134	204	0	1	162	0	0.8	2	2	2	0
1888	54	1	1	154	232	0	0	164	0	0.0	2	1	2	0
1889	53	1	0	110	335	0	1	143	1	3.0	1	1	3	0

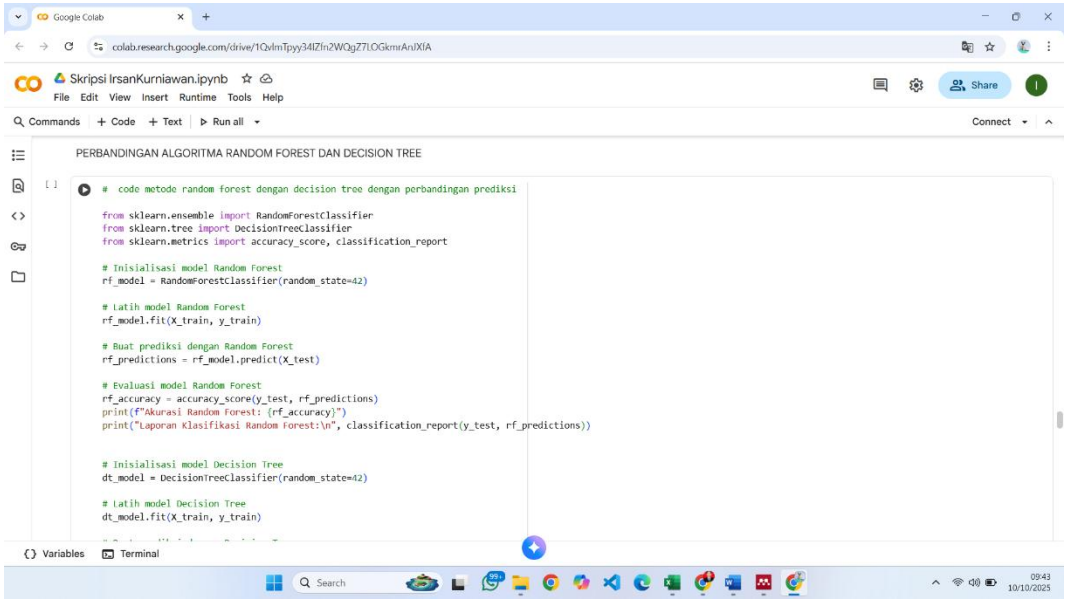

Google Colab
colab.research.google.com/drive/1QvlmTpyy34IZfn2WOgZ7LOGkmrAnJXIA
Skripsi IrsanKurniawan.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
Connect
import pandas as pd
import numpy as np
import plotly.express as px
from sklearn import metrics
from sklearn.metrics import confusion_matrix, classification_report,make_scorer
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
Exploratory Data Analyis (EDA)
Data Collection Tahapan mengumpulkan dan mempersiapkan data
# Menyambungkan Google Colab ke Google Drive agar dapat mengakses Dataset / DataFrame
from google.colab import drive # library utuk mengakses Google Drive dari Google Collab
drive.mount('/content/drive') # Menghubungkan ke google drive
file_path = '/content/drive/My Drive/Data Penelitian/_heart_dataset.csv' # Alamat File data
df = pd.read_csv(file_path) #Membaca File
Mounted at /content/drive
df.head()
age sex cp trestbps chol fbs restecg thalachh exang oldpeak slope ca thal target
0 63 1 3 145 233 1 0 150 0 2.3 0 0 1 1
Variables
Terminal
Search
09:42
10/10/2025

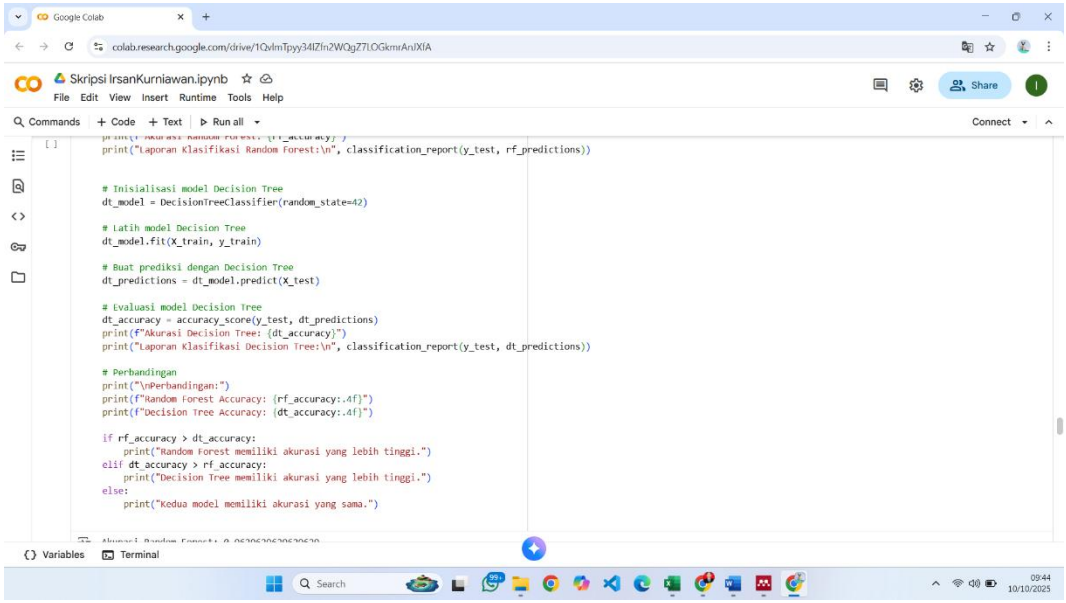

Google Colab
colab.research.google.com/drive/1QvlmTpyy34IZfn2WOgZ7LOGkmrAnJXIA
Skripsi IrsanKurniawan.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
Connect
df.head()
age sex cp trestbps chol fbs restecg thalachh exang oldpeak slope ca thal target
0 63 1 3 145 233 1 0 150 0 2.3 0 0 1 1
1 37 1 2 130 250 0 1 187 0 3.5 0 0 2 1
2 41 0 1 130 204 0 0 172 0 1.4 2 0 2 1
3 56 1 1 120 236 0 1 178 0 0.8 2 0 2 1
4 57 0 0 120 354 0 1 163 1 0.6 2 0 2 1
Mengecek Type Data
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1888 entries, 0 to 1887
Data columns (total 14 columns):
Column Non-Null Count Dtype
0 age 1888 non-null int64
1 sex 1888 non-null int64
2 cp 1888 non-null int64
3 trestbps 1888 non-null int64
4 chol 1888 non-null int64
5 fbs 1888 non-null int64
6 restecg 1888 non-null int64
7 thalachh 1888 non-null int64
Variables
Terminal
Search
09:42
10/10/2025


Google Colab
colab.research.google.com/drive/1QvlmTpyy34IZfn2WQgZ7LOGkmrAnJXIA
Skripsi IrsanKurniawan.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
Connect
[ ]
# analisis data outlier
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
def detect_and_plot_outliers(df, column_name):
"""
Mendeteksi dan memvisualisasikan outlier untuk kolom tertentu menggunakan box plot.
Args:
df: The DataFrame containing the data.
column_name: The name of the column to analyze.
"""
# Menghitung Kuartil
Q1 = df[column_name].quantile(0.25)
Q3 = df[column_name].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
# Identifikasi outliers
outliers = df[(df[column_name] < lower_bound) | (df[column_name] > upper_bound)]
# Membuat boxplot
plt.figure(figsize=(4, 2))
sns.boxplot(y=df[column_name])
plt.title(f'Boxplot of {column_name} with Outliers')
Variables
Terminal
Search
09:42
10/10/2025

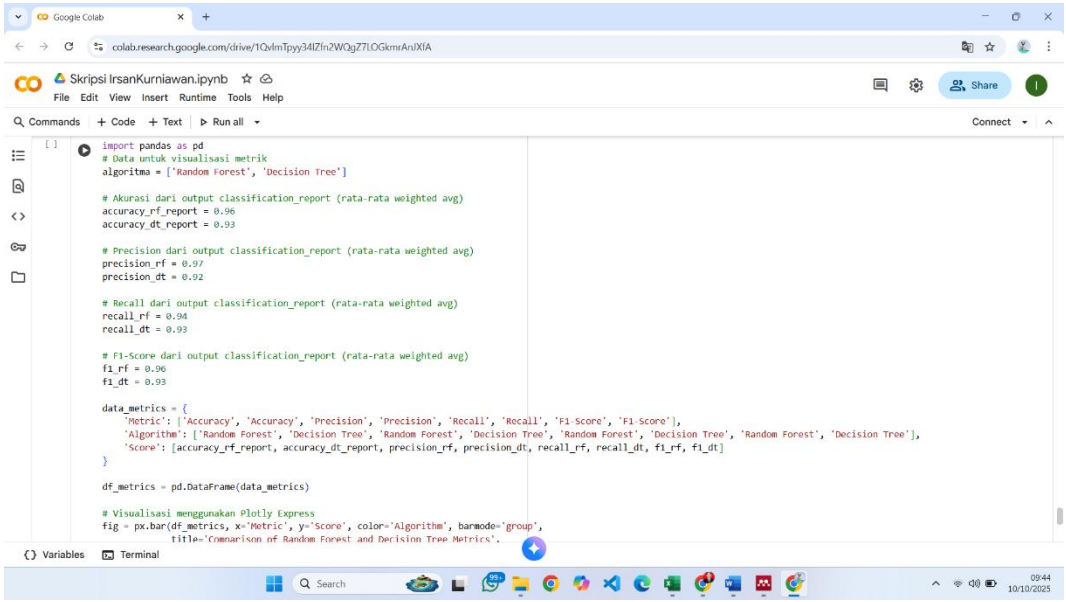

Google Colab
colab.research.google.com/drive/1QvlmTpyy34IZfn2WQgZ7LOGkmrAnJXIA
Skripsi IrsanKurniawan.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
Connect
[ ]
# code mengurangi outlier pada kolom yang terdapat outlier
def handle_outliers_iqr(df, column_name):
"""Handles outliers in a specified column using the IQR method without removing data.
Args:
df: The DataFrame containing the data.
column_name: The name of the column to process.
"""
Q1 = df[column_name].quantile(0.25)
Q3 = df[column_name].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
# Replace outliers with the bounds
df.loc[df[column_name] < lower_bound, column_name] = lower_bound
df.loc[df[column_name] > upper_bound, column_name] = upper_bound
# Specify the columns to handle outliers
columns_to_handle = ['trestbps', 'chol', 'fbs', 'thalachh', 'oldpeak', 'ca', 'thal']
# Apply outlier handling to specified columns
for col in columns_to_handle:
handle_outliers_iqr(df, col)
# Now you can proceed with your further analysis.
# For example, you can visualize the boxplots again to see the effect
# of outlier handling:
for col in columns_to_handle:
detect_and_plot_outliers(df, col)
Variables
Terminal
Search
09:43
10/10/2025


Google Colab
colab.research.google.com/drive/1QvlmTpyy34IZfn2WOgZ7LOGkmrAnJXIA
Skripsi IrsanKurniawan.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
Connect
Visualisasi Menggunakan Histogram
# visualisasi histogram dari masing masing kolom distribusi data diatas
import matplotlib.pyplot as plt
# Membuat histogram untuk distribusi data di setiap kolom
for col in df.columns:
plt.figure(figsize=(4, 2))
sns.histplot(data=df, x=col, kde=True)
plt.title(f'Distribution of {col}')
plt.xlabel(col)
plt.ylabel('Frequency')
plt.show()
Frequency
age
Distribution of sex
Variables
Terminal
Search
09:43
10/10/2025


Google Colab
colab.research.google.com/drive/1QvlmTpyy34IZfn2WOgZ7LOGkmrAnJXIA
Skripsi IrsanKurniawan.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
Connect
PERBANDINGAN ALGORITMA RANDOM FOREST DAN DECISION TREE
# code metode random forest dengan decision tree dengan perbandingan prediksi
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
# Inisialisasi model Random Forest
rf_model = RandomForestClassifier(random_state=42)
# Latih model Random Forest
rf_model.fit(X_train, y_train)
# Buat prediksi dengan Random Forest
rf_predictions = rf_model.predict(X_test)
# Evaluasi model Random Forest
rf_accuracy = accuracy_score(y_test, rf_predictions)
print(f"Akurasi Random Forest: {rf_accuracy}")
print("Laporan Klasifikasi Random Forest:\n", classification_report(y_test, rf_predictions))
# Inisialisasi model Decision Tree
dt_model = DecisionTreeClassifier(random_state=42)
# Latih model Decision Tree
dt_model.fit(X_train, y_train)
Variables
Terminal
Search
09:43
10/10/2025



```
print("Laporan Klasifikasi Random Forest:\n", classification_report(y_test, rf_predictions))


# Inisialisasi model Decision Tree
dt_model = DecisionTreeClassifier(random_state=42)

# Latih model Decision Tree
dt_model.fit(X_train, y_train)

# Buat prediksi dengan Decision Tree
dt_predictions = dt_model.predict(X_test)

# Evaluasi model Decision Tree
dt_accuracy = accuracy_score(y_test, dt_predictions)
print(f"Akurasi Decision Tree: {dt_accuracy}")
print("Laporan Klasifikasi Decision Tree:\n", classification_report(y_test, dt_predictions))

# Perbandingan
print("\nPerbandingan:")
print(f"Random Forest Accuracy: {rf_accuracy:.4f}")
print(f"Decision Tree Accuracy: {dt_accuracy:.4f}")

if rf_accuracy > dt_accuracy:
    print("Random Forest memiliki akurasi yang lebih tinggi.")
elif dt_accuracy > rf_accuracy:
    print("Decision Tree memiliki akurasi yang lebih tinggi.")
else:
    print("Kedua model memiliki akurasi yang sama.")
```



```
import pandas as pd
# Data untuk visualisasi metrik
algoritma = ['Random Forest', 'Decision Tree']

# Akurasi dari output classification_report (rata-rata weighted avg)
accuracy_rf_report = 0.96
accuracy_dt_report = 0.93

# Precision dari output classification_report (rata-rata weighted avg)
precision_rf = 0.97
precision_dt = 0.92

# Recall dari output classification_report (rata-rata weighted avg)
recall_rf = 0.94
recall_dt = 0.93

# F1-Score dari output classification_report (rata-rata weighted avg)
f1_rf = 0.96
f1_dt = 0.93

data_metrics = {
    'Metric': ['Accuracy', 'Accuracy', 'Precision', 'Precision', 'Recall', 'Recall', 'F1-Score', 'F1-Score'],
    'Algorithm': ['Random Forest', 'Decision Tree', 'Random Forest', 'Decision Tree', 'Random Forest', 'Decision Tree', 'Random Forest', 'Decision Tree'],
    'Score': [accuracy_rf_report, accuracy_dt_report, precision_rf, precision_dt, recall_rf, recall_dt, f1_rf, f1_dt]
}

df_metrics = pd.DataFrame(data_metrics)

# Visualisasi menggunakan Plotly Express
fig = px.bar(df_metrics, x='Metric', y='Score', color='Algorithm', barmode='group',
             title='Comparison of Random Forest and Decision Tree Metrics',
```


Google Colab
colab.research.google.com/drive/1QvlmTpyy34IlZln2WQgZ7LOGkmrAnIXIA
Skripsi IrsanKurniawan.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
Connect
Comparison of Random Forest and Decision Tree Metrics
Value
Metric
Accuracy
Precision
Recall
F1-Score
0.96
0.93
0.97
0.92
0.94
0.93
0.96
0.93
Algorithm
Random Forest
Decision Tree
Variables
Terminal
Search
09:44
10/10/2025