Laporan Tugas Besar Pembelajaran Mesin Lanjut
Klasifikasi Dataset “ Breast Cancer Wisconsin ” Menggunakan Automated Machine Learning

Disusun Oleh :

1. MUHAMMAD RIZKI NURFIQRI**(1301204009)**
2. KIKI DWI PRASETYO **(1301204027)**

IF-43-PIL-01

Program Studi Sarjana Informatika
Fakultas Informatika
Universitas Telkom
Bandung

2023

Daftar isi

\

BAB I
Formulasi Masalah

Pada bab ini, akan dijelaskan tentang masalah yang ingin diselesaikan melalui tugas besar pembelajaran mesin lanjut. Masalah yang akan dipecahkan adalah klasifikasi dataset "Breast Cancer Wisconsin" untuk memprediksi hasil dari label “Diagnosis” yang akan mempresentasikan yang menandakan bahwa mana yang kanker dengan inisial “M” lalu untuk yang bukan kanker berinisial “B”

BAB II
Eksplorasi dan Persiapan Data

A. Eksplorasi Data

Eksplorasi data (data exploration) adalah suatu proses dalam analisis data yang bertujuan untuk memahami karakteristik, pola, dan informasi yang terkandung dalam data. Eksplorasi data dilakukan sebelum analisis data yang lebih mendalam dilakukan, seperti pemodelan statistik atau pembuatan prediksi.

```
# Menampilkan ringkasan singkat dari data test
df_test.info()
```

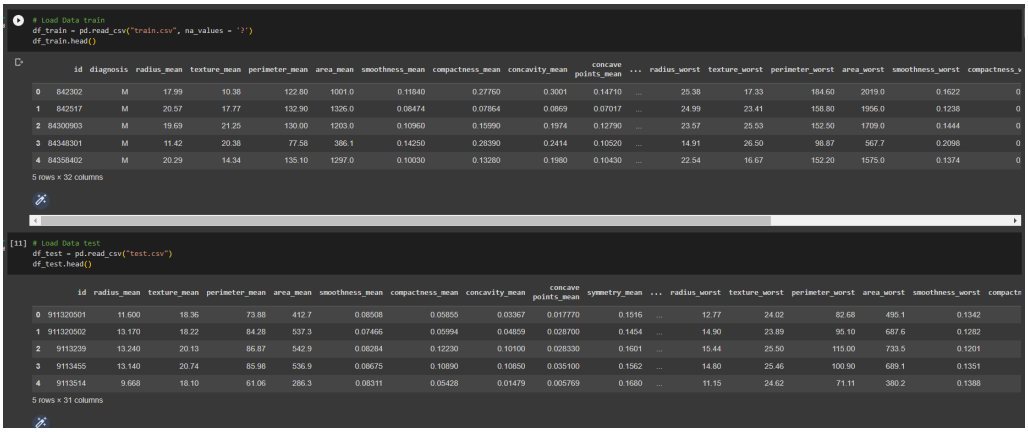
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 106 entries, 0 to 105
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   id                      106 non-null    int64  
 1   radius_mean             106 non-null    float64
 2   texture_mean            106 non-null    float64
 3   perimeter_mean          106 non-null    float64
 4   area_mean               106 non-null    float64
 5   smoothness_mean         106 non-null    float64
 6   compactness_mean        106 non-null    float64
 7   concavity_mean          106 non-null    float64
 8   concave points_mean     106 non-null    float64
 9   symmetry_mean           106 non-null    float64
 10  fractal_dimension_mean  106 non-null    float64
 11  radius_se               106 non-null    float64
 12  texture_se              106 non-null    float64
 13  perimeter_se            106 non-null    float64
 14  area_se                 106 non-null    float64
 15  smoothness_se           106 non-null    float64
 16  compactness_se          106 non-null    float64
 17  concavity_se            106 non-null    float64
 18  concave points_se       106 non-null    float64
 19  symmetry_se             106 non-null    float64
 20  fractal_dimension_se    106 non-null    float64
 21  radius_worst            106 non-null    float64
 22  texture_worst           106 non-null    float64
 23  perimeter_worst         106 non-null    float64
 24  area_worst              106 non-null    float64
 25  smoothness_worst        106 non-null    float64
 26  compactness_worst       106 non-null    float64
 27  concavity_worst         106 non-null    float64
 28  concave points_worst    106 non-null    float64
 29  symmetry_worst          106 non-null    float64
 30  fractal_dimension_worst  106 non-null    float64
dtypes: float64(30), int64(1)
memory usage: 25.8 KB
```

```
# Menampilkan ringkasan singkat dari data train
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 463 entries, 0 to 462
Data columns (total 32 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   id                      463 non-null    int64  
 1   diagnosis               463 non-null    object 
 2   radius_mean             463 non-null    float64
 3   texture_mean            463 non-null    float64
 4   perimeter_mean          463 non-null    float64
 5   area_mean               463 non-null    float64
 6   smoothness_mean         463 non-null    float64
 7   compactness_mean        463 non-null    float64
 8   concavity_mean          456 non-null    float64
 9   concave points_mean     456 non-null    float64
 10  symmetry_mean           463 non-null    float64
 11  fractal_dimension_mean  463 non-null    float64
 12  radius_se               463 non-null    float64
 13  texture_se              463 non-null    float64
 14  perimeter_se            463 non-null    float64
 15  area_se                 463 non-null    float64
 16  smoothness_se           463 non-null    float64
 17  compactness_se          463 non-null    float64
 18  concavity_se            456 non-null    float64
 19  concave points_se       456 non-null    float64
 20  symmetry_se             463 non-null    float64
 21  fractal_dimension_se    463 non-null    float64
 22  radius_worst            463 non-null    float64
 23  texture_worst           463 non-null    float64
 24  perimeter_worst         463 non-null    float64
 25  area_worst              463 non-null    float64
 26  smoothness_worst        463 non-null    float64
 27  compactness_worst       463 non-null    float64
 28  concavity_worst         456 non-null    float64
 29  concave points_worst    456 non-null    float64
 30  symmetry_worst          463 non-null    float64
 31  fractal_dimension_worst  463 non-null    float64
dtypes: float64(30), int64(1), object(1)
memory usage: 115.9+ KB
```

B. Load Dataset

Load dataset adalah sebuah proses untuk membaca atau memuat data dalam suatu format tertentu ke dalam suatu program atau aplikasi untuk tujuan analisis atau pemrosesan data lebih lanjut. Dataset dapat berupa file teks, file CSV, file Excel, file SQL, atau format lain yang sesuai dengan program yang digunakan.



```
# Load Data train
df_train = pd.read_csv("train.csv", na_values = '?')
df_train.head()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactness_w
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	25.38	17.33	184.60	2019.0	0.1622	0
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	24.99	23.41	158.80	1956.0	0.1238	0
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	23.57	25.53	152.50	1709.0	0.1444	0
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	14.91	26.50	98.87	567.7	0.2098	0
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	22.54	16.67	152.20	1575.0	0.1374	0

5 rows × 32 columns

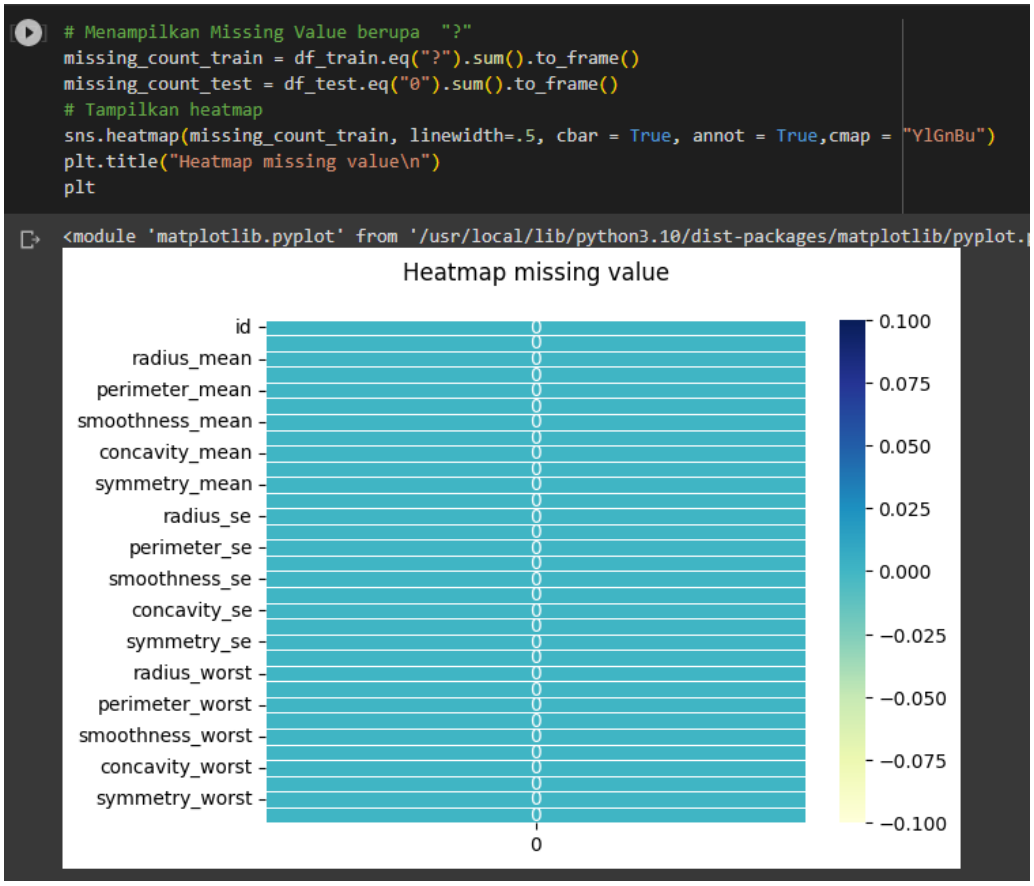
```
[11] # Load Data test
df_test = pd.read_csv("test.csv")
df_test.head()
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	...	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactn
0	911320501	11.600	18.36	73.88	412.7	0.08508	0.05855	0.03367	0.017770	0.1516	...	12.77	24.02	82.68	495.1	0.1342	
1	911320502	13.170	18.22	84.28	537.3	0.07466	0.05994	0.04859	0.028700	0.1454	...	14.90	23.89	95.10	687.6	0.1282	
2	9113239	13.240	20.13	86.87	542.9	0.08284	0.12230	0.10100	0.028330	0.1601	...	15.44	25.50	115.00	733.5	0.1201	
3	9113455	13.140	20.74	85.98	536.9	0.08675	0.10890	0.10850	0.035100	0.1562	...	14.80	25.46	100.90	689.1	0.1351	
4	9113514	9.668	18.10	61.06	286.3	0.08311	0.05428	0.01479	0.005769	0.1680	...	11.15	24.62	71.11	380.2	0.1388	

5 rows × 31 columns

C. **Mengganti nilai Missing Value**

Beberapa Dataset ada beberapa yang perlu di bersihkan, yaitu pada dataset ini meimiliki baris yang berisikan nilai “?” dan ada beberapa atribut seperti “id” irelevan dalam proses training,



```
# Menampilkan Missing Value berupa  "?"
missing_count_train = df_train.eq("?").sum().to_frame()
missing_count_test = df_test.eq("0").sum().to_frame()
# Tampilkan heatmap
sns.heatmap(missing_count_train, linewidth=.5, cbar = True, annot = True,cmap = "YlGnBu")
plt.title("Heatmap missing value\n")
plt
```

```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.10/dist-packages/matplotlib/pyplot.
```


Lalu kami mengganti nilai tersebut dengan nilai mean

```
#Ganti missing value dengan nilai mean
df_train = df_train.fillna(df_train.mean())
df_test = df_test.fillna(df_test.mean())
```

```
<ipython-input-19-e04d38d2c11e>:2: FutureWarning: The default value of r
  df_train = df_train.fillna(df_train.mean())
```

```
[20] missing_values_train = df_train.isnull().sum()
     missing_values_test = df_test.isnull().sum()
     print("Jumlah missing value di train.csv: \n", missing_values_train)
     print("Jumlah missing value di test.csv: \n", missing_values_test)
```

D. Spliting feature Data

Splitting feature data adalah suatu proses dalam analisis data yang bertujuan untuk memisahkan variabel atau fitur dalam dataset menjadi dua atau lebih kelompok yang saling terpisah. Tujuannya adalah untuk memudahkan analisis dan pemrosesan data selanjutnya, serta memastikan bahwa data yang digunakan untuk pelatihan model atau pengujian model tidak tercampur atau terjadi kebocoran informasi.

```
#mendefinisikan fitur-fitur dan label
features = df_train[['radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst']]
label = df_train[['diagnosis']]
```

```
x_train, x_test, y_train, y_test = train_test_split(features, label, test_size = 0.2)
```

BAB III
Pemodelan

A. Metode Machine Learning menggunakan Automation Tools (TpOT).

TpOT (Tree-based Pipeline Optimization Tool) adalah sebuah perangkat lunak yang digunakan untuk otomatisasi dan optimasi pembuatan pipeline (rantai alur kerja) dalam pembelajaran mesin. TpOT dikembangkan dengan menggunakan algoritma genetika dan pohon keputusan untuk mencari dan mengoptimalkan kombinasi terbaik dari preprocessing data, model, dan hiperparameter untuk mendapatkan model yang optimal.

Pada Tugas Besar ini kali ini kami menggunakan TpOT karena dapat membuat memberikan informasi yang benar dalam melakukan pemodelan dari dataset yang kami miliki. Dengan ini dapat diharaspkan mampu memberikan akurasi yang terbaik dari dataset ini. Dan kami menggunakan TPOTClassifier dengan atribut generations=5, population_size=20, cv=5, random_state=42, verbosity=2

```
#menginstall tpot
!pip install tpot
import tpot
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting tpot
  Downloading TPOT-0.12.0-py3-none-any.whl (87 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 87.4/87.4 kB 8.2 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.16.3 in /usr/local/lib/python3.10/dist-packages (from tpot) (1.22.4)
Requirement already satisfied: scipy>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from tpot) (1.10.1)
Requirement already satisfied: scikit-learn>=0.22.0 in /usr/local/lib/python3.10/dist-packages (from tpot) (1.2.2)
Collecting deap>=1.2 (from tpot)
  Downloading deap-1.3.3-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (139 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 139.9/139.9 kB 14.5 MB/s eta 0:00:00
Collecting update-checker>=0.16 (from tpot)
  Downloading update_checker-0.18.0-py3-none-any.whl (7.0 kB)
Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.10/dist-packages (from tpot) (4.65.0)
Collecting stopit>=1.1.1 (from tpot)
  Downloading stopit-1.1.2.tar.gz (18 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: pandas>=0.24.2 in /usr/local/lib/python3.10/dist-packages (from tpot) (1.5.3)
Requirement already satisfied: joblib>=0.13.2 in /usr/local/lib/python3.10/dist-packages (from tpot) (1.2.0)
Requirement already satisfied: xgboost>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tpot) (1.7.5)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24.2->tpot) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24.2->tpot) (2022.7.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.22.0->tpot) (3.1.0)
Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from update-checker>=0.16->tpot) (2.27.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas>=0.24.2->tpot) (1.16.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.3.0->update-checker>=0.16->tpot) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.3.0->update-checker>=0.16->tpot) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests>=2.3.0->update-checker>=0.16->tpot) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.3.0->update-checker>=0.16->tpot) (3.4)
Building wheels for collected packages: stopit
  Building wheel for stopit (setup.py) ... done
  Created wheel for stopit: filename=stopit-1.1.2-py3-none-any.whl size=11938 sha256=ef73280d56ddf87d0d3ac63750ee9f3d6de1ffc79c1c78383251227849926bb4
  Stored in directory: /root/.cache/pip/wheels/af/f9/87/bf5b3d565c2a007b4dae9d8142dccc85a9f164e517062dd519
Successfully built stopit
Installing collected packages: stopit, deap, update-checker, tpot
Successfully installed deap-1.3.3 stopit-1.1.2 tpot-0.12.0 update-checker-0.18.0
```

```
[ ] from tpot.tpot import TPOTClassifier

    tpot =  TPOTClassifier(generations=5, population_size=20, cv=5,
                                    random_state=42, verbosity=2)
```

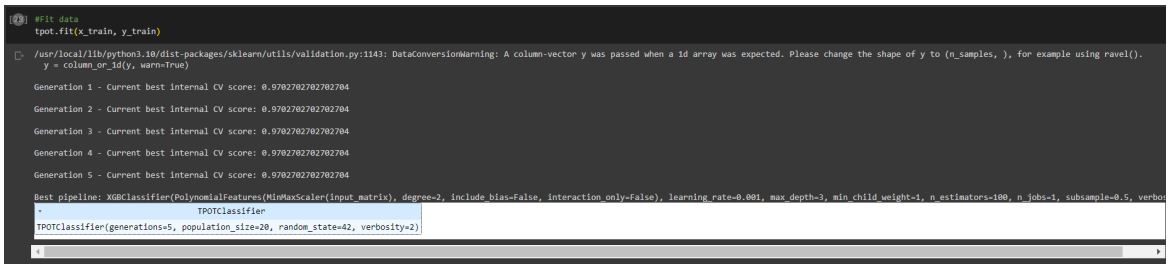
B. Fit Data

"Fit data" adalah istilah yang sering digunakan dalam konteks pemodelan atau pembelajaran mesin. Fungsi "fit" pada dasarnya digunakan untuk melatih model atau mengatur parameter model agar cocok dengan data yang diberikan. Dalam konteks ini, "fit data" merujuk pada proses melatih model dengan data yang telah dikumpulkan atau disediakan.

Fungsi "fit" biasanya diimplementasikan dalam kerangka kerja atau pustaka pemodelan tertentu, seperti scikit-learn dalam Python. Biasanya, fungsi ini mengambil dua argumen utama: data masukan (fit data) dan target yang sesuai (label atau hasil yang diharapkan). Proses melatih model dengan data masukan tersebut melibatkan

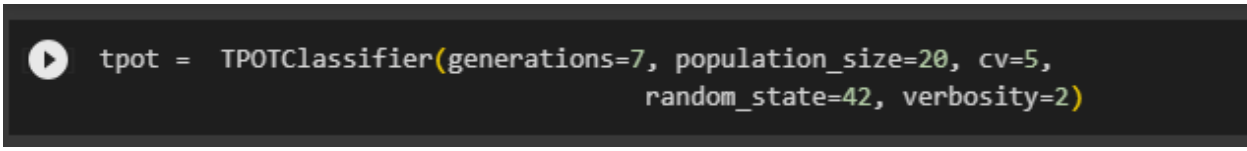
penyesuaian parameter internal model untuk meminimalkan kesalahan atau perbedaan antara output yang dihasilkan oleh model dan target yang diharapkan.

dibawah ini adalah model fit data dari hasil TPOTClassifier dengan atribut generations=5

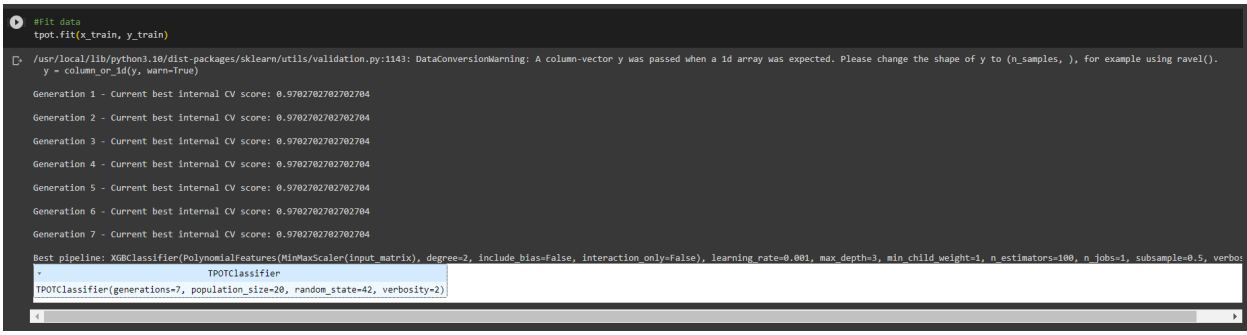


C. Eksperimen

1. Pemodelan
   Berikut adalah eksperimen kami dengan mengubah nilai TPOTClassifier dengan atribut generations manjadi 7



2. Fit Data
   Dan dibawah ini adalah hasil dari Fit data dengan perubahan TPOTClassifier dengan atribut generations manjadi 7

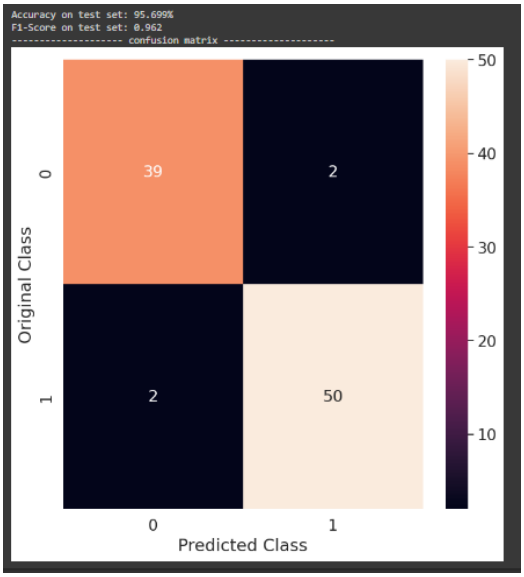


BAB IV
Evaluasi

Dalam tugas besar ini, kami akan mengukur performa model TpOT dalam klasifikasi pada dataset Breast Cancer Wisconsin (Diagnostic) menggunakan akurasi, F1-score, dan confusion matrix. Metode ini dipilih untuk mengevaluasi kemampuan model dalam membedakan kanker dengan non-kanker. Akurasi memberikan gambaran persentase prediksi yang tepat, sedangkan F1-score menggabungkan presisi dan recall untuk memberikan gambaran lebih lengkap. Confusion matrix memberikan informasi tentang prediksi yang benar dan salah untuk setiap kelas klasifikasi. Evaluasi ini akan membantu kami dalam menilai kualitas model TpOT dan mengidentifikasi area yang perlu diperbaiki atau dikembangkan lebih lanjut.

```
#EVALUASI
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score

y_pred=tpot.predict(x_test)
print("Accuracy on test set: %0.3f%%"%(accuracy_score(y_test, y_pred)*100))
print("F1-Score on test set: %0.3f"%(f1_score(y_test, y_pred)))
print("-"*20, "confusion matrix", "-"*20)
plt.figure(figsize=(8,8))
df_cm = pd.DataFrame(confusion_matrix(y_test, y_pred), range(2),range(2))
sns.set(font_scale=1.4)#for label size
sns.heatmap(df_cm, annot=True,annot_kws={"size": 16}, fmt='g')
plt.xlabel('Predicted Class')
plt.ylabel('Original Class')
plt.show()
```



Hasil evaluasi dari performa model TpOT dalam klasifikasi pada dataset Breast Cancer Wisconsin (Diagnostic) menunjukkan tingkat akurasi sebesar 95,699%, F1-score sebesar 0,962, dan confusion matrix dengan false positive sebanyak 39, false negative sebanyak 2, dan true negative sebanyak 50. Dengan hasil ini, dapat disimpulkan bahwa model TpOT memiliki performa yang sangat baik dalam mengklasifikasikan kanker dan non-kanker pada dataset ini.

BAB V
Penutup

A. Penutup

Dalam tugas besar ini, kami berhasil menerapkan Automated Machine Learning menggunakan model TPOT dan mengevaluasi performa model TPOT dalam klasifikasi pada dataset Breast Cancer Wisconsin (Diagnostic). Kami menggunakan metrik evaluasi akurasi, F1-score, dan confusion matrix untuk menilai kinerja model TPOT dalam membedakan antara kanker dan non-kanker.

Namun, untuk meningkatkan performa dan keandalan model, diperlukan upaya perbaikan dan pengembangan lebih lanjut. Analisis yang lebih mendalam terhadap kasus false positive dan false negative, penerapan teknik pra-pemrosesan data, dan penalaan parameter.

Dengan upaya perbaikan dan pengembangan yang tepat, model TPOT memiliki potensi untuk menjadi alat yang sangat berguna dalam membantu identifikasi kanker pada dataset Breast Cancer Wisconsin (Diagnostic). Harapan kami adalah bahwa hasil dari tugas besar ini dapat memberikan kontribusi positif dan membuka peluang untuk pengembangan lebih lanjut di bidang klasifikasi kanker.

Kami berharap penelitian dan pengembangan dalam bidang klasifikasi kanker terus berkembang untuk memberikan manfaat yang besar bagi dunia medis dan kesehatan. Dengan demikian, kami mengakhiri tugas besar ini dengan harapan yang tinggi terhadap potensi dan kemajuan di masa depan.

B. Link Source Code

https://drive.google.com/drive/folders/1d0fJAzBRmXyw9WiSlltJr0tPu9SphniE?usp=sharing

REFERENCE

- Tukey, J. W. (1977). Exploratory data analysis. Addison-Wesley.
- Wickham, H. (2016). ggplot2: elegant graphics for data analysis. Springer.
- Tufte, E. R. (2001). The visual display of quantitative information. Graphics press.
- Kandel, S., Parikh, R., Paepcke, A., Hellerstein, J. M., & Heer, J. (2011). Profiler: Integrated statistical analysis and visualization for data quality assessment. ACM Transactions on Interactive Intelligent Systems (TiiS), 1(3), 1-37.
- Dasu, T., & Johnson, T. (2003). Exploratory data mining and data cleaning. John Wiley & Sons.
- McKinney, W. (2010). Data structures for statistical computing in python. Proceedings of the 9th Python in Science Conference, 51-56.
- Wickham, H. (2011). The split-apply-combine strategy for data analysis. Journal of Statistical Software, 40(1), 1-29.
- Ince, D. C., Hatton, L., & Graham-Cumming, J. (2012). The case for open computer programs. Nature, 482(7386), 485-488.
- Peng, R. D. (2011). Reproducible research in computational science. Science, 334(6060), 1226-1227.