LAPORAN TUGAS CASE BASED 1
PEMBELAJARAN MESIN
Untuk memenuhi tugas mata kuliah Pembelajaran Mesin - DDR

Disusun oleh:

Vania Amadea

1301204365

IF4408

Program Studi S1 Informatika
Fakultas Informatika
Universitas Telkom
Bandung
2022

*Saya mengerjakan tugas ini dengan cara yang tidak melanggar aturan perkuliahan dan kode etik akademisi.

Daftar Isi

I. Library yang digunakan 3

II. Ikhtisar kumpulan data 3

III. Preprocessing data 5

IV. Visualisasi Data 7

V. Feature scaling 9

VI. Algoritma/metode yang diterapkan 9

VII. Evaluasi hasil 9

VIII. Link 10

IX. Reference Link 10

I. Library yang digunakan

```
[ ]  import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import StandardScaler
     from sklearn.neural_network import MLPClassifier
```

II. Ikhtisar kumpulan data

Data yang digunakan pada Case Based 1 merupakan Audit Data Dataset sesuai dengan ketentuan soal untuk mahasiswa yang memiliki NIM ganjil. Berdasarkan situs yang menyediakan dataset ini, dataset ini merupakan dataset non-rahasia satu tahun yang lengkap pada tahun 2015 hingga 2016 perusahaan dikumpulkan dari Kantor Auditor India untuk membangun prediktor untuk mengklasifikasikan perusahaan yang mencurigakan. Dataset ini digunakan untuk membangun model klasifikasi yang akan memprediksi dengan risiko yang ada.

- Isi data

Read dan menampilkan data csv

```
df_train = pd.read_csv('audit_risk.csv', index_col = None)
df_train
```

	Sector_score	LOCATION_ID	PARA_A	Score_A	Risk_A	PARA_B	Score_B	Risk_B	TOTAL	numbers	...	RiSk_E	History	Prob	Risk_F	Score	Inherent_Risk	CONTROL_RISK	Detection_Risk	Audit_Risk	Risk
0	3.89	23	4.18	0.6	2.508	2.50	0.2	0.500	6.68	5.0	...	0.4	0	0.2	0.0	2.4	8.574	0.4	0.5	1.7148	1
1	3.89	6	0.00	0.2	0.000	4.83	0.2	0.966	4.83	5.0	...	0.4	0	0.2	0.0	2.0	2.554	0.4	0.5	0.5108	0
2	3.89	6	0.51	0.2	0.102	0.23	0.2	0.046	0.74	5.0	...	0.4	0	0.2	0.0	2.0	1.548	0.4	0.5	0.3096	0
3	3.89	6	0.00	0.2	0.000	10.80	0.6	6.480	10.80	6.0	...	0.4	0	0.2	0.0	4.4	17.530	0.4	0.5	3.5060	1
4	3.89	6	0.00	0.2	0.000	0.08	0.2	0.016	0.08	5.0	...	0.4	0	0.2	0.0	2.0	1.416	0.4	0.5	0.2832	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
771	55.57	9	0.49	0.2	0.098	0.40	0.2	0.080	0.89	5.0	...	0.4	0	0.2	0.0	2.0	1.578	0.4	0.5	0.3156	0
772	55.57	16	0.47	0.2	0.094	0.37	0.2	0.074	0.84	5.0	...	0.4	0	0.2	0.0	2.0	1.568	0.4	0.5	0.3136	0
773	55.57	14	0.24	0.2	0.048	0.04	0.2	0.008	0.28	5.0	...	0.4	0	0.2	0.0	2.0	1.456	0.4	0.5	0.2912	0
774	55.57	18	0.20	0.2	0.040	0.00	0.2	0.000	0.20	5.0	...	0.4	0	0.2	0.0	2.0	1.440	0.4	0.5	0.2880	0
775	55.57	15	0.00	0.2	0.000	0.00	0.2	0.000	0.00	5.0	...	0.4	0	0.2	0.0	2.0	1.464	0.4	0.5	0.2928	0

776 rows × 27 columns

- Informasi data
  Dapat dilihat label dan tipe data setiap label.

Ringkasan lengkap dari dataframe

```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 776 entries, 0 to 775
Data columns (total 27 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Sector_score    776 non-null    float64
 1   LOCATION_ID     776 non-null    object 
 2   PARA_A          776 non-null    float64
 3   Score_A         776 non-null    float64
 4   Risk_A          776 non-null    float64
 5   PARA_B          776 non-null    float64
 6   Score_B         776 non-null    float64
 7   Risk_B          776 non-null    float64
 8   TOTAL           776 non-null    float64
 9   numbers         776 non-null    float64
 10  Score_B.1       776 non-null    float64
 11  Risk_C          776 non-null    float64
 12  Money_Value     775 non-null    float64
 13  Score_MV        776 non-null    float64
 14  Risk_D          776 non-null    float64
 15  District_Loss   776 non-null    int64  
 16  PROB            776 non-null    float64
 17  RiSk_E          776 non-null    float64
 18  History         776 non-null    int64  
 19  Prob            776 non-null    float64
 20  Risk_F          776 non-null    float64
 21  Score           776 non-null    float64
 22  Inherent_Risk   776 non-null    float64
 23  CONTROL_RISK    776 non-null    float64
 24  Detection_Risk  776 non-null    float64
 25  Audit_Risk      776 non-null    float64
 26  Risk            776 non-null    int64  
dtypes: float64(23), int64(3), object(1)
memory usage: 163.8+ KB
```

Berdasarkan data yang ada, harus dilakukan tahap *preprocessing* data untuk memastikan kualitas data baik sebelum digunakan ke dalam model yang dipilih. Tahap ini dapat dilakukan dengan beberapa hal sederhana yaitu mencari missing value, mengisi missing value, membuang bagian yang tidak diperlukan, dan memeriksanya kembali.

III. Preprocessing data

- Dilakukan pencarian missing value
  Terdapat missing value pada Money_Value sebanyak 1

Memeriksa adanya missing value pada dataframe

```
df_train.isnull().sum()
```

```
Sector_score      0
LOCATION_ID       0
PARA_A            0
Score_A           0
Risk_A            0
PARA_B            0
Score_B           0
Risk_B            0
TOTAL             0
numbers           0
Score_B.1         0
Risk_C            0
Money_Value       1
Score_MV          0
Risk_D            0
District_Loss     0
PROB              0
RiSk_E            0
History           0
Prob              0
Risk_F            0
Score             0
Inherent_Risk     0
CONTROL_RISK      0
Detection_Risk    0
Audit_Risk        0
Risk              0
dtype: int64
```

- Isi missing value dengan mean Money_Value

Isi null dengan nilai mean

```
[ ]  df_train['Money_Value'].fillna((df_train['Money_Value'].mean()), inplace = True)
```

- Membuang kolom LOCATION_ID dan TOTAL karena tidak dibutuhkan

Drop kolom LOCATION_ID dan TOTAL

```
df_train.drop(['LOCATION_ID', 'TOTAL'], axis = 1, inplace = True)
```

- Memeriksa kembali data

Memeriksa adanya missing value pada dataframe

```
df_train.isnull().sum()
```

```
Sector_score      0
LOCATION_ID       0
PARA_A            0
Score_A           0
Risk_A            0
PARA_B            0
Score_B           0
Risk_B            0
TOTAL             0
numbers           0
Score_B.1         0
Risk_C            0
Money_Value       0
Score_MV          0
Risk_D            0
District_Loss     0
PROB              0
RiSk_E            0
History           0
Prob              0
Risk_F            0
Score             0
Inherent_Risk     0
CONTROL_RISK      0
Detection_Risk    0
Audit_Risk        0
Risk              0
dtype: int64
```

Memeriksa dataset dengan menampilkan 5 teratas

```
df_train.head()
```

	Sector_score	PARA_A	Score_A	Risk_A	PARA_B	Score_B	Risk_B	numbers	Score_B.1	Risk_C	...	RiSk_E	History	Prob	Risk_F	Score	Inherent_Risk	CONTROL_RISK	Detection_Risk	Audit_Risk	Risk
0	3.89	4.18	0.6	2.508	2.50	0.2	0.500	5.0	0.2	1.0	...	0.4	0	0.2	0.0	2.4	8.574	0.4	0.5	1.7148	1
1	3.89	0.00	0.2	0.000	4.83	0.2	0.966	5.0	0.2	1.0	...	0.4	0	0.2	0.0	2.0	2.554	0.4	0.5	0.5108	0
2	3.89	0.51	0.2	0.102	0.23	0.2	0.046	5.0	0.2	1.0	...	0.4	0	0.2	0.0	2.0	1.548	0.4	0.5	0.3096	0
3	3.89	0.00	0.2	0.000	10.80	0.6	6.480	6.0	0.6	3.6	...	0.4	0	0.2	0.0	4.4	17.530	0.4	0.5	3.5060	1
4	3.89	0.00	0.2	0.000	0.08	0.2	0.016	5.0	0.2	1.0	...	0.4	0	0.2	0.0	2.0	1.416	0.4	0.5	0.2832	0

5 rows × 25 columns

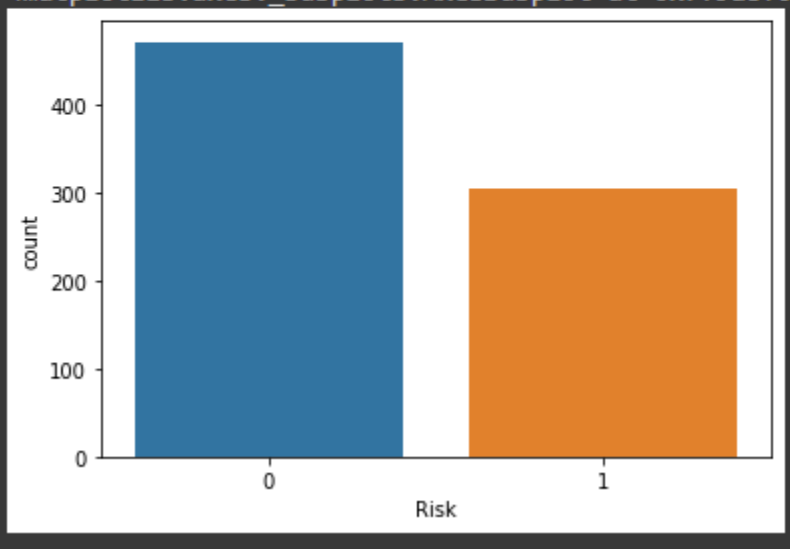
IV. Visualisasi Data

- Melihat banyaknya data 0 dan 1 pada target yaitu kolom Risk

Melihat banyaknya data 0 dan 1 pada kolom Risk (Target)

```
sns.countplot(df_train['Risk'], label='Count')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_deco
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f9a8f02
```



- Membuat variabel x dan y, x berisi nilai kecuali Risk, y berisi nilai Risk saja

```
x.head()
```

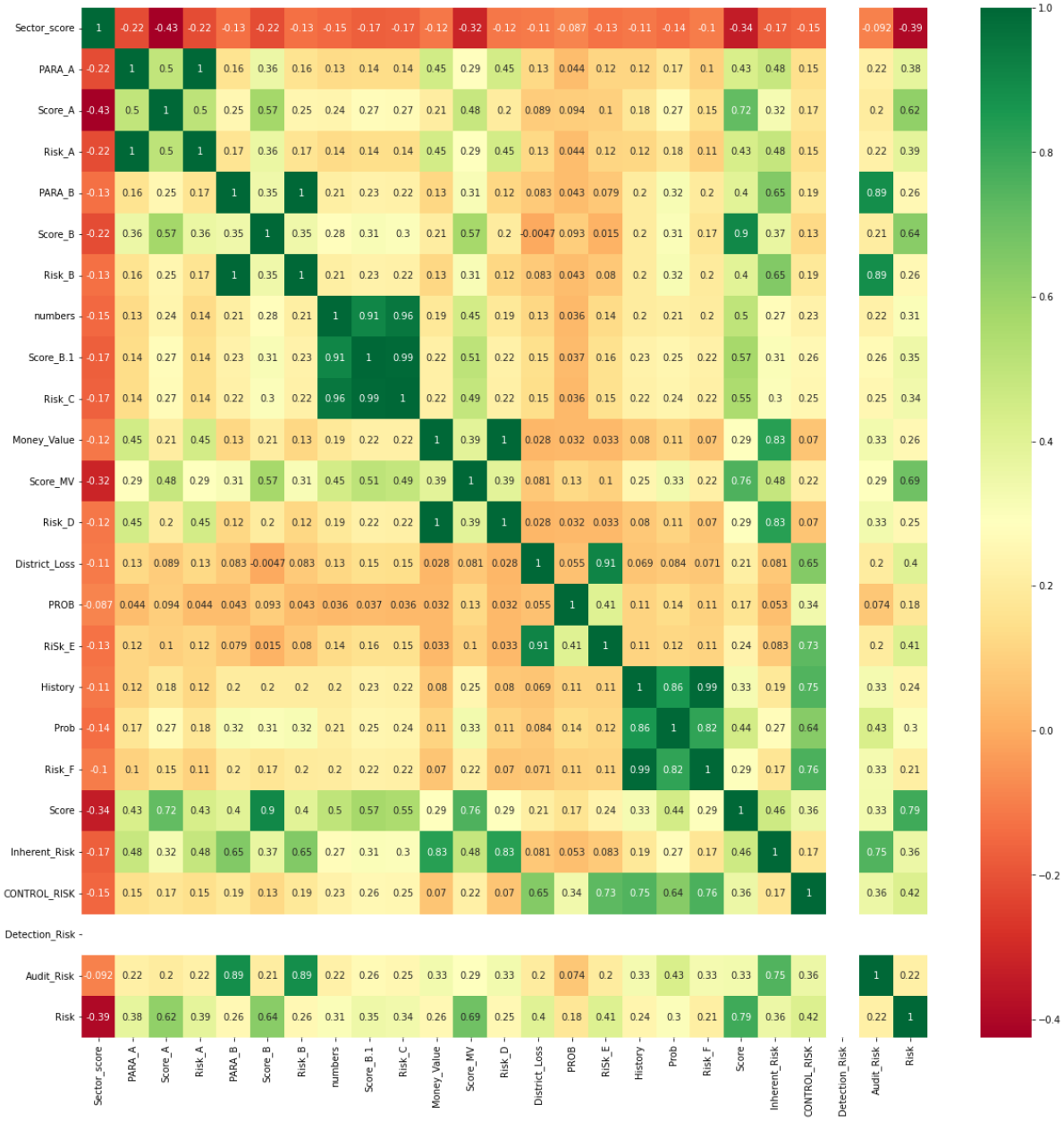
	Sector_score	PARA_A	Score_A	Risk_A	PARA_B	Score_B	Risk_B	numbers	Score_B.1	Risk_C	...	PROB	RiSk_E	History	Prob	Risk_F	Score	Inherent_Risk	CONTROL_RISK	Detection_Risk	Audit_Risk
0	3.89	4.18	0.6	2.508	2.50	0.2	0.500	5.0	0.2	1.0	...	0.2	0.4	0	0.2	0.0	2.4	8.574	0.4	0.5	1.7148
1	3.89	0.00	0.2	0.000	4.83	0.2	0.966	5.0	0.2	1.0	...	0.2	0.4	0	0.2	0.0	2.0	2.554	0.4	0.5	0.5108
2	3.89	0.51	0.2	0.102	0.23	0.2	0.046	5.0	0.2	1.0	...	0.2	0.4	0	0.2	0.0	2.0	1.548	0.4	0.5	0.3096
3	3.89	0.00	0.2	0.000	10.80	0.6	6.480	6.0	0.6	3.6	...	0.2	0.4	0	0.2	0.0	4.4	17.530	0.4	0.5	3.5060
4	3.89	0.00	0.2	0.000	0.08	0.2	0.016	5.0	0.2	1.0	...	0.2	0.4	0	0.2	0.0	2.0	1.416	0.4	0.5	0.2832

5 rows × 24 columns

```
[ ] y.head()
```

```
0    1
1    0
2    0
3    1
4    0
Name: Risk, dtype: int64
```

- Melihat korelasi data
  Di sini kita dapat melihat adanya celah di Detection_Risk, maka harus dibuang atau didrop.




```
[ ]  x = x.drop('Detection_Risk', axis = 1)
     x.columns
```

```
Index(['Sector_score', 'PARA_A', 'Score_A', 'Risk_A', 'PARA_B', 'Score_B',
       'Risk_B', 'numbers', 'Score_B.1', 'Risk_C', 'Money_Value', 'Score_MV',
       'Risk_D', 'District_Loss', 'PROB', 'RiSk_E', 'History', 'Prob',
       'Risk_F', 'Score', 'Inherent_Risk', 'CONTROL_RISK', 'Audit_Risk'],
      dtype='object')
```

V. Feature scaling

Pada bagian ini, dilakukan train test split yaitu membagi dataset menjadi dua bagian yakni bagian yang digunakan untuk training data dan untuk testing data dengan proporsi 80% untuk training dan 20% untuk test. Setelah melakukan train test split, dilakukan feature scaling data pada dataset memiliki rentang nilai (scale) yang sama yaitu dari 0 sampai 1.

```
x_train,x_test, y_train, y_test = train_test_split(x,y , test_size=0.2, stratify=y,random_state=2)

[ ]  sc_x = StandardScaler()
     x_train_scaled = pd.DataFrame(sc_x.fit_transform(x_train))
     x_test_scaled = pd.DataFrame(sc_x.transform(x_test))
```

VI. Algoritma/metode yang diterapkan

Metode *supervised learning* yang digunakan adalah MLP. Multilayer perceptron (MLP) adalah jaringan saraf tiruan feedforward yang menghasilkan serangkaian output dari serangkaian input dan MLP memiliki tiga lapisan (input layer, hidden layer, dan output layer) sebagai cirinya. MLP merupakan metode yang tepat untuk kasus ini dikarenakan dataset merupakan data tabular dan memiliki records yang tidak terlalu banyak meskipun dimensinya cukup besar.

Salah satu cara menggunakan MLP adalah dengan library yang sudah tersedia.

```
[ ]  mlp = MLPClassifier()
     mlp.fit(x_train, y_train)

     /usr/local/lib/python3.7/dist-packages/skl
       ConvergenceWarning,
     MLPClassifier()
```

VII. Evaluasi hasil

Evaluasi hasil dilakukan dengan menggunakan fungsi score dan ternyata didapatkan 1.0 yang artinya 100%. Oleh karena itu, tidak perlu diadakannya resample data.

```
[ ]  mlp.score(x_test, y_test)

     1.0
```

VIII. Link

Presentation slide link :
https://www.canva.com/design/DAFRc9cuXEg/Fn4t6q1F6cJJJ4LTr9f8kw/view?utm_content=DAFRc9cuXEg&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton
Presentation video, docs, and colab link :
CASE BASED 1 - VANIA AMADEA (1301204365)

IX. Reference Link

https://hyanuun.com/apa-itu-mlp-multi-layer-perceptron-mari-kita-berkenalan-dengan-salah-satu-algoritma-kecerdasan-buatan-ai/
https://algorit.ma/blog/library-python/
https://www.tensorflow.org/api_docs/python/tf/keras/Sequential
https://garudacyber.co.id/artikel/1461-algoritma-multi-layer-perceptron