Laporan Case-Based 1
Machine Learning

Dosen Pembimbing:
DEDE ROHIDIN, DRS.,MT
CII3C3-IF-44-08

Disusun oleh:
1301200421 Herjanto Janawisuta

S1 Informatika
Universitas Telkom
Bandung, Jawa Barat 2022

Kata Pengantar

Dengan mengucapkan puji dan rasa syukur kepada Allah SWT yang telah memberikan rahmat sehingga kita dapat menyelesaikan tugas dari mata kuliah Pembelajaran Mesin dengan tema “Implementasi AAN/MLP/RNN/LSTM/CNN” dengan benar dan tepat waktu.

Untuk memenuhi nilai tugas pada mata kuliah Pembelajaran Mesin, maka dibuatkan tugas yang dapat saya selesaikan. Tidak hanya itu, tujuan dari pembuatan laporan dan pengerjaan tugas ini adalah untuk menambah wawasan tentang pembahasan Implementasi ANN bagi kita semua.

Saya mengucapkan terima kasih kepada semua pihak mulai dari Pak DEDE ROHIDIN, DRS.,MT selaku dosen pembimbing yang telah memberikan saya tugas besar untuk membuat Algoritma ANN.

Saya sangat menyadari laporan yang saya susun masih jauh dari kata sempurna. Tetapi saya akan terus berusaha untuk selalu menjadi lebih baik untuk kedepannya.

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

Bandung, 7 November 2022

Herjanto Janawisuta

BAB I
PENDAHULUAN

Case-Based 1

Scenario

Mengikuti keberhasilan tugas sebelumnya, Anda diberi kesempatan lebih lanjut untuk mengesankan atasan Anda mengenai kemampuan Anda untuk menganalisis data. Anda diminta untuk melakukan beberapa analisis dan menghasilkan seperangkat aturan yang berguna menggunakan dataset berikut:

Kumpulan data berikut tersedia online, tautan ke kumpulan data adalah sebagai berikut:
[untuk NIM akhir GENAP gunakan data ini]
https://archive.ics.uci.edu/ml/datasets/Arrhythmia
[untuk NIM akhir GANJIL gunakan data ini]
https://archive.ics.uci.edu/ml/datasets/Audit+Data

Anda perlu mempelajari data dengan hati-hati. Kemudian pilih teknik pra-pemrosesan data apa yang akan dilakukan untuk meningkatkan kualitas data tersebut. Akan ada banyak hal yang harus diuraikan dan kemudian Anda harus mengumpulkan case-based ini sebagai karya individu.

Hint: Anda bebas memilih satu dari tiga alat analisis data yaitu Weka, R, atau Python untuk membantu
Anda menganalisis data dan menunjukkan pra-pemrosesan data yang diperlukan.

Tugas Anda

Tujuan dari tugas ini yaitu Anda diharapkan mampu menjelaskan, mengimplementasikan, menganalisis, dan mendesain teknik pembelajaran mesin supervised learning yaitu AAN/MLP/RNN/LSTM/CNN. Pertama, selidiki masalah kualitas data yang telah diberikan di atas. Jelaskan keputusan Anda mengenai pendekatan pra-pemrosesan data. Jelajahi kumpulan data dengan meringkas data menggunakan statistik dan mengidentifikasi masalah kualitas data apa pun. Tidak ada batasan jumlah ringkasan yang akan dilaporkan tetapi Anda diharapkan hanya melaporkan yang paling relevan. Kedua, pilih salah satu dari metode unsupervised learning yang telah dipelajari yaitu AAN/MLP/RNN/LSTM/CNN. Anda hanya perlu memilih satu metode untuk diterapkan. Gunakan algoritma tersebut untuk memberikan beberapa output/outcome dengan menggunakan variasi hyperparameter, kemudian menganalisis hasilnya.

BAB II
PEMBAHASAN

2.1 Ikhtisar Kumpulan Data yang Dipilih

Pada bagian ini saya mendapatkan data ganjil dengan nim yang diakhiri dengan "1" maka dari itu saya akan memakai data audit dan trial sebagai acuan pengerjaan soal.

audit_risk.csv

```
# Read Dataset audit_risk

url_1 = 'https://raw.githubusercontent.com/Herjantoj/audit_data/main/audit_risk.csv'
audit_risk = pd.read_csv(url_1)
# audit_risk.describe()
audit_risk
```

	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

trial.csv

```
# Read Dataset trial

url_2 = 'https://raw.githubusercontent.com/Herjantoj/audit_data/main/trial.csv'
trial = pd.read_csv(url_2)
# trial.describe()
trial
```

	Sector_score	LOCATION_ID	PARA_A	SCORE_A	PARA_B	SCORE_B	TOTAL	numbers	Marks	Money_Value	MONEY_Marks	District	Loss	LOSS_SCORE	History	History_score	Score	Risk
0	3.89	23	4.18	6	2.50	2	6.68	5.0	2	3.38	2	2	0	2	0	2	2.4	1
1	3.89	6	0.00	2	4.83	2	4.83	5.0	2	0.94	2	2	0	2	0	2	2.0	0
2	3.89	6	0.51	2	0.23	2	0.74	5.0	2	0.00	2	2	0	2	0	2	2.0	0
3	3.89	6	0.00	2	10.80	6	10.80	6.0	6	11.75	6	2	0	2	0	2	4.4	1
4	3.89	6	0.00	2	0.08	2	0.08	5.0	2	0.00	2	2	0	2	0	2	2.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
771	55.57	9	0.49	2	0.40	2	0.89	5.0	2	0.00	2	2	0	2	0	2	2.0	0
772	55.57	16	0.47	2	0.37	2	0.84	5.0	2	0.00	2	2	0	2	0	2	2.0	0
773	55.57	14	0.24	2	0.04	2	0.28	5.0	2	0.00	2	2	0	2	0	2	2.0	0
774	55.57	18	0.20	2	0.00	2	0.20	5.0	2	0.00	2	2	0	2	0	2	2.0	0
775	55.57	15	0.00	2	0.00	2	0.00	5.0	2	0.32	2	2	0	2	0	2	2.0	0

776 rows × 18 columns

Data ini merupakan data *history risk factor* dimana kita diminta untuk melakukan research mengenai dataset ini untuk bisa memperkirakan perusahaan mana saja yang bisa diklasifikasikan sebagai perusahaan yang *Fraud* berdasarkan history fakta sekarang yang ada.

2.2 Pre-Processing Data

Me-rename kolom pada dataset trial untuk proses merger atau penggabungan.

```
[678] #Rename atribut pada dataset trial

      trial.rename(columns = {'SCORE_A':'Score_A'}, inplace = True)
      trial.rename(columns = {'SCORE_B':'Score_B'}, inplace = True)
      trial.rename(columns = {'Risk':'Risk_Trial'}, inplace = True)
```

Membagi value pada beberapa atribut trial untuk menyamakannya dengan value pada dataset audit_risk.

```
#Membagi atribut Score_A dan Score_B dengan 10

trial['Score_A'] = trial['Score_A']/10
trial['Score_B'] = trial['Score_B']/10
```

Menggabung kedua dataset menjadi satu untuk mempermudah mengolah dataset.

```
[681] #Menggabung dua dataset

      df = pd.merge(audit_risk, trial, how='outer', on = ['History', 'LOCATION_ID', 'Money_Value', 'PARA_A', 'PARA_B',
            'Score', 'Score_A', 'Score_B', 'Sector_score', 'TOTAL', 'numbers'])
      df.columns

      Index(['Sector_score', 'LOCATION_ID', 'PARA_A', 'Score_A', 'Risk_A', 'PARA_B',
             'Score_B', 'Risk_B', 'TOTAL', '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',
             'Detection_Risk', 'Audit_Risk', 'Risk', 'Marks', 'MONEY_Marks',
             'District', 'Loss', 'LOSS_SCORE', 'History_score', 'Risk_Trial'],
            dtype='object')
```

Cek missing value pada dataset

```
[682] # Cek Missing Value

      df.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
      Marks             0
      MONEY_Marks       0
      District          0
      Loss              0
      LOSS_SCORE        0
      History_score     0
      Risk_Trial        0
      dtype: int64
```

Dapat dilihat terdapat 1 buah missing value pada atribut Money_Value, maka dari itu kita dapat mengganti nya dengan nilai median pada atribut tersebut.

```
[683] # Mengganti missing value dengan nilai rata2 pada atribut Monet_Value

      df['Money_Value'] = df['Money_Value'].fillna(df['Money_Value'].median())
      df.isnull().sum()
```

Melakukan drop kepada dua atribut ini, karena mereka memiliki value yang sama sepanjang baris.

```
[684] #Atribut 'Detection_Risk' dan 'Risk_F' memiliki value yang sama sepanjang baris, sehingga bisa di drop saja

      df = df.drop(['Detection_Risk', 'Risk_F'], axis = 1)
      df.info()
```

Mengubah Atribut “LOCATION_ID” menjadi *unique*.

```
[685] # Membuat value "LOCATION_ID" menjadi unique

      df["LOCATION_ID"].unique()

      array(['23', '6', '7', '8', '13', '37', '24', '3', '4', '14', '5', '20',
             '19', '21', '22', '9', '11', '12', '29', '30', '38', '31', '2',
             '32', '16', '33', '15', '36', '34', '18', '25', '39', '27', '35',
             '40', '41', '42', '1', '28', 'LOHARU', 'NUH', 'SAFIDON', '43',
             '44', '17'], dtype=object)
```

Bisa dilihat pada gambar diatas terdapat ID yang bukan numerik, karena itu kita bisa drop baris yang terdapat ID non-numerik tersebut.

```
# Menghapus baris yang terdapat value non-numerik

df = df[(df.LOCATION_ID != 'LOHARU')]
df = df[(df.LOCATION_ID != 'NUH')]
df = df[(df.LOCATION_ID != 'SAFIDON')]
df = df.astype(float)
print("Jumlah baris : ",len(df))

Jumlah baris :  807
```

Mencari dan menghapus value yang duplikat pada dataset.

```
[688] # Menghapus value yang duplikat

      df = df.drop_duplicates(keep = 'first')
      print("Jumlah baris : ",len(df))

      Jumlah baris :  760
```

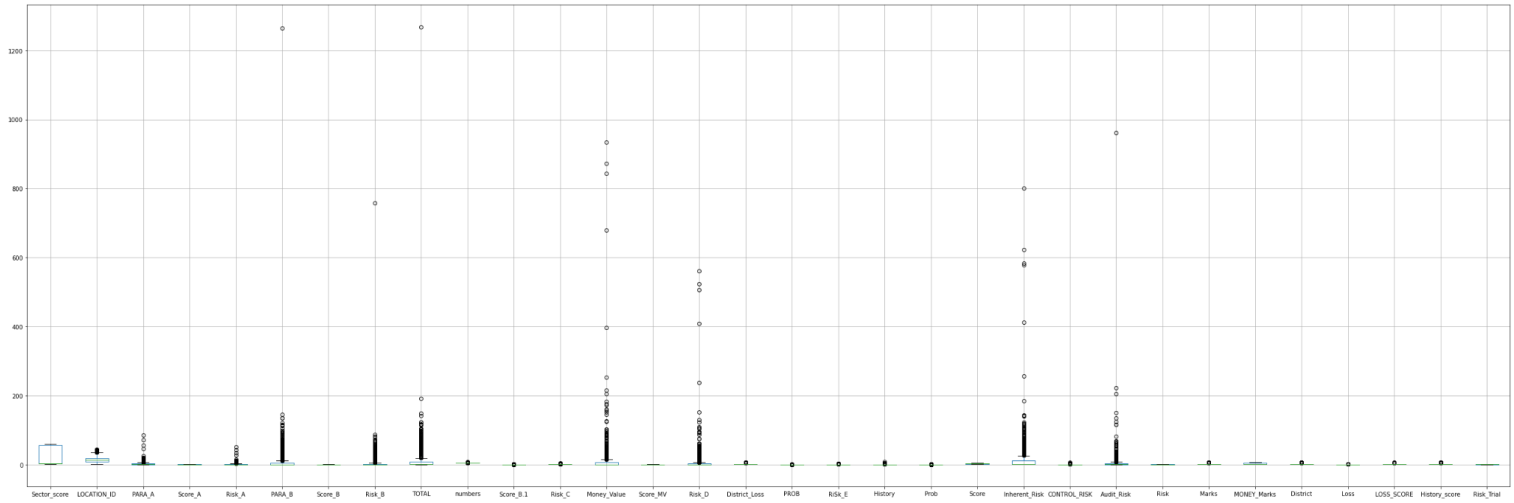
Menghitung nilai outlier pada setiap atribut atau kolom dataset “df”.

```
[689] # Menghitung outlier

      Q1 = df.quantile(0.25)
      Q3 = df.quantile(0.75)
      IQR = Q3 - Q1
      ((df < (Q1 - 1.5 * df)) | (df > (Q3 + 1.5 * IQR))).sum()
```

```
Sector_score       0
LOCATION_ID       94
PARA_A           245
Score_A            0
Risk_A           244
PARA_B           132
Score_B            0
Risk_B           139
TOTAL            251
numbers           70
Score_B.1         70
Risk_C            70
Money_Value      114
Score_MV           0
Risk_D           149
District_Loss    122
PROB              22
RiSk_E           138
History           50
Prob              50
Score              0
Inherent_Risk    132
CONTROL_RISK     173
Audit_Risk       126
Risk               0
Marks             70
MONEY_Marks        0
District         122
Loss              21
LOSS_SCORE        22
History_score     50
Risk_Trial         0
dtype: int64
```

Menampilkan data dengan boxplot untuk melihat outliers dengan visual.

```
#Boxplot

plt.figure(figsize=(45,15))
df.boxplot(column=['Sector_score', 'LOCATION_ID', 'PARA_A', 'Score_A', 'Risk_A', 'PARA_B',
       'Score_B', 'Risk_B', 'TOTAL', 'numbers', 'Score_B.1', 'Risk_C',
       'Money_Value', 'Score_MV', 'Risk_D', 'District_Loss', 'PROB', 'RiSk_E',
       'History', 'Prob', 'Score', 'Inherent_Risk', 'CONTROL_RISK',
       'Audit_Risk', 'Risk', 'Marks', 'MONEY_Marks','District',
       'Loss', 'LOSS_SCORE', 'History_score', 'Risk_Trial'])
plt.show()
```



Karena terdapat banyak sekali outliers yang terdapat pada dataset, maka kita tidak bisa melakukan drop pada atribut2 tersebut. karena kalau kita drop jadi bias, dan kalo mengubah outlier jadi central tendency maka output bisa berbeda dari yang diinginkan.

2.3 Klasifikasi

Menghilangkan atribut risk yang nantinya akan dipakai sebagai acuan testing

```
[691] # Menghapus atribut risk pada dataset sebagai bahan training dan test

      classification_X = df.drop(["Risk"], axis = 1)
      classification_y = df["Risk"]
```

Membagi data menjadi data train dan data testing

```
[808] # Train-Test Split

    X_train_org, X_test_org, y_train, y_test = train_test_split(classification_X, classification_y, test_size = 0.25, random_state = 0)
```

Karena kita tidak melakukan drop pada outlier-outlier yang ada pada dataset, maka solusinya adalah dengan *scaling*.

```
[809] # Feature Scaling

      scaler = MinMaxScaler()

      X_train = scaler.fit_transform(X_train_org)
      X_test  = scaler.transform(X_test_org)
```

2.4 Algoritma ANN

Metode Artificial Neural Network (ANN) sendiri merupakan suatu pendekatan model kecerdasan yang didasari akan cara bekerja struktur otak manusia dan kemudian diimplementasikan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses learning berlangsung.

Dengan menggunakan ini diharapkan mendapatkan nilai training dan nilai validasi yang sesuai dengan harapan dan tidak terjadi overfitting model atau bad model.

Disini saya menggunakan 2 *hidden-layer*, dimana keduanya menggunakan activation *'tanh'*, sementara pada output menggunakan 'relu'

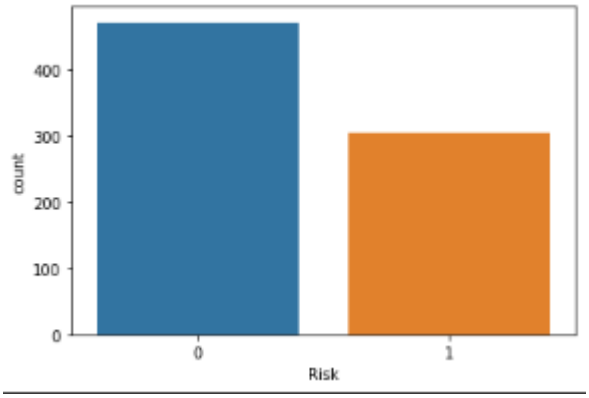
```
[810] # activation function

      model = tf.keras.Sequential([
          tf.keras.layers.Dense(units=31, input_dim=31),
          tf.keras.layers.Dense(32, activation='tanh'),
          tf.keras.layers.Dense(32, activation='tanh'),
          tf.keras.layers.Dense(1, activation='relu')
      ])
      model.summary()
```

Testing dataset menghasilkan accuracy : 0.9737 atau 97%

```
print("Evaluasi data ")
results = model.evaluate(X_test, y_test, batch_size=128)
```

```
Evaluasi data
2/2 [==============================] - 0s 4ms/step - loss: 0.0666 - accuracy: 0.9737
```

Dapat disimpulkan bahwa data yang kita dapat merupakan data dengan kualitas yang tinggi. Dimana dataset hanya memiliki 1 missing value, sedikit value yang duplikat, dan tidak perlu melakukan *undersampling* ataupun *oversampling* karena data sudah balance. Kekurangannya ada pada banyaknya *outliers* pada dataset, tapi hal itu dapat teratasi dengan melakukan scaling. Kualitas data terbukti setelah mencoba berbagai kombinasi activation function pada hidden-layer, hampir rata-rata hasil akurasi > 90%.



Link Colab dan Youtube:
https://colab.research.google.com/drive/1MwGOqRtE69hNOwj9qzX5C0b2O0gyeww3?usp=sharing
https://youtu.be/c3MUzpehHug