**Laporan Case-Based 1**

**Mata Kuliah Pembelajaran Mesin**

**Artificial Neural Network (ANN)**

**Wandi Yusuf Kurniawan – 1301218601 – IFX-45-GAB – IKN**

****

**Program Studi Sarjana Informatika**

**Fakultas Informatika**

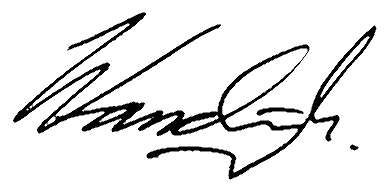
**Universitas Telkom**

**Bandung**

**2022**

**PERNYATAAN KODE ETIK AKADEMIK UNIVERSITAS TELKOM**

Dengan ini saya bersaksi bahwa hasil kerja ini **saya kerjakan sendiri** dan tidak mencontek hasil kerja dari mahasiswa lain dan tidak melakukan kecurangan dalam pengerjaan tugas ini. Jika kesaksian saya ini tidak benar, maka **saya bersedia menerima sanksi diberi minimum nilai E untuk mata kuliah ini dan/atau maksimum untuk semua mata kuliah pada semester ini.**



Wandi Yusuf Kurniawan

Pertama-tama, penulis melakukan import library yang dibutuhkan untuk mengerjakan tugas ini pada Jupyter Notebook menggunakan bahasa Python, diantaranya adalah:

Text

Description automatically generated

**IKHTISAR KUMPULAN DATA YANG DIPILIH**

Kemudian inisialisasi variable data untuk membaca file .csv yang diambil dari <https://archive.ics.uci.edu/ml/datasets/Audit+Data> dan lakukan .head() untuk memastikan bahwa variable tersebut sudah berisi dataset yang nantinya akan dijadikan pembentukkan model neural network:

Table

Description automatically generated

Table

Description automatically generated

Cek .info() pada data untuk mengetahui informasi kolom, berdasarkan table di bawah terdapat 776 baris dan 27 kolom, salah satu kolom, yaitu Money\_Value, memiliki satu baris kosong atau null.

Chart

Description automatically generated

**RINGKASAN PRA-PEMROSESAN DATA YANG DIUSULKAN**

Penulis menghapus dua kolom yang tidak diperlukan, yaitu 'LOCATION\_ID' karena bukan bertipe int/float dan 'TOTAL' karena penjumlahan dari kolom ’Risk\_A’ dan ’Risk\_B’, serta satu baris pada kolom Money\_Value yang harus diimputasi agar tidak ada nilai N/A sehingga jumlah kolomnya berkurang menjadi 25.

Table

Description automatically generated with medium confidence

Penulis melakukan .describe() untuk melihat statistic deskriptif apakah kolom tersebut sangat mempengaruhi label, yaitu pada kolom ’Risk’, dengan mengecek simpangan baku atau std.

Table

Description automatically generated

Text

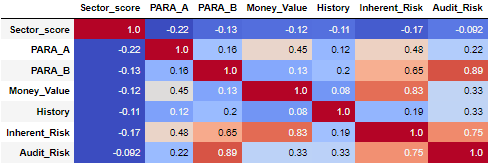
Description automatically generated with medium confidenceKarena jumlah kolomnya masih sangat banyak, maka dilakukanlah reduksi dimensi pada dataset. Pertama-tama, lakukan pemisahan dataset dengan variable X (untuk kolom fitur) dan y (untuk kolom target). Kemudian hitung nilai normalized\_X = X / X.mean() dan ambil nilai .var() yang lebih dari satu.

Jika digambarkan dalam kolerasi antar kolom, maka hasilnya sebagai berikut:

A screenshot of a computer

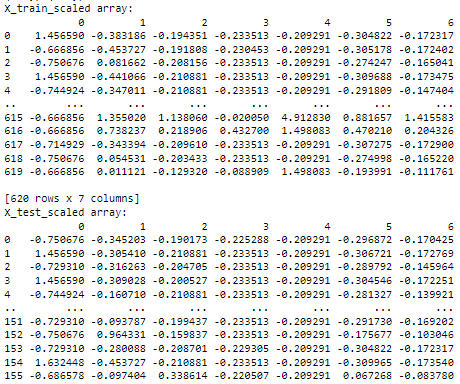
Description automatically generated with medium confidence

Terdapat empat pasang kolom yang korelasinya hampir satu (yang berwarna merah pekat), ini menandakan bahwa dataset tersebut masih memiliki redudansi fitur dan akan sangat berpengaruh pada performansi model yang akan dibuat. Penulis menghapus salah satu daripada keempat pasangan fitur, yaitu pada kolom ‘Risk\_A’, ‘Risk\_B’, ‘Risk\_D’, dan ‘Risk\_F’.



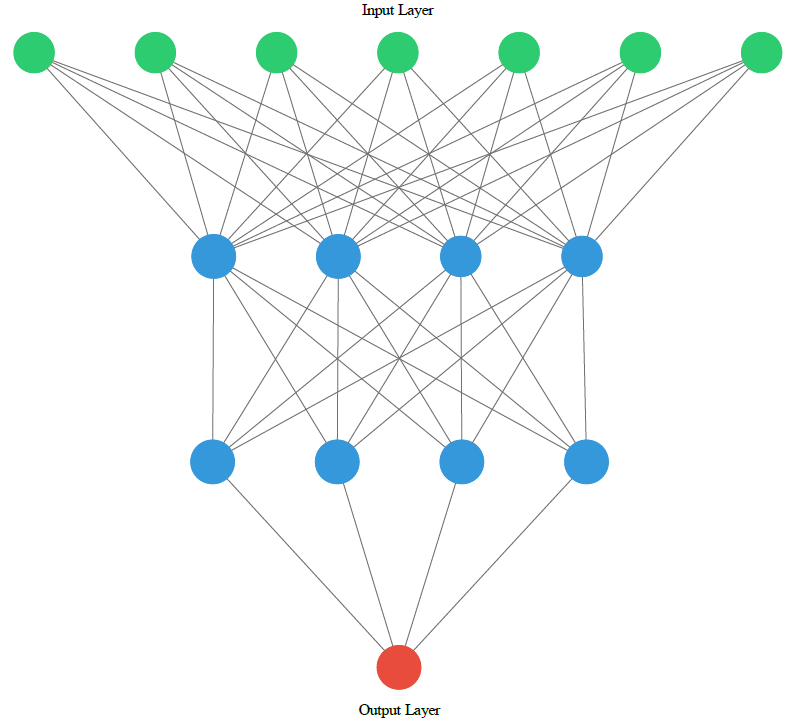
Setelah selesai reduksi fitur dengan menyisakan tujuh fitur yang siap dijadikan sebagai input model, penulis melakukan pemisahan dataset latih dan uji dengan menggunakan train\_test\_split dengan ukuran dataset uji sebesar 20%. Selanjutnya, variable X\_train dan X\_test dinormalisasi dengan menggunakan StandardScaler()agar memudahkan model dalam penyesuaian bobot antar-node dan antar-layer yang akan dibahas pada bagian selanjutnya.

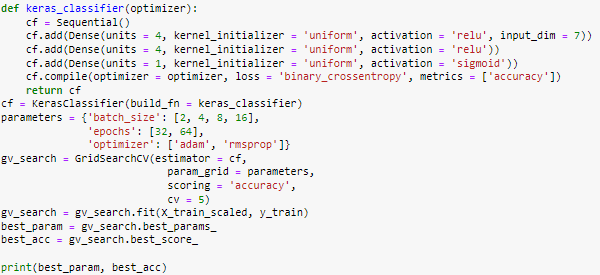


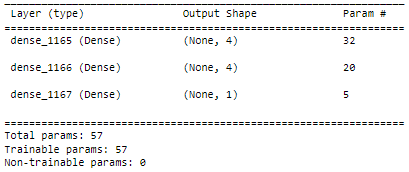


**MENERAPKAN ALGORITMA YANG DI PILIH**

Algoritma yang saya rancang adalah artificial neural network (ANN) dengan dua hidden layer karena datasetnya hanya berupa fitur dan label saja, bukan merupakan data yang tidak terstruktur seperti gambar atau audio.







{'batch\_size': 2, 'epochs': 64, 'optimizer': 'adam'} 0.9290322580645161

ANN ini memiliki 57 parameter yang terdiri dari input sebanyak tujuh kolom fitur hasil reduksi, empat node di setiap hidden layer menggunakan fungsi aktivasi relu, dan satu output berupa kolom label menggunakan fungsi aktivasi sigmoid, kemudian di-compile untuk menghitung nilai loss menggunakan binary\_crossentropy dan metrik akurasi yang biasanya akan meningkat pada setiap iterasi.

Hyperparameter yang digunakan ada tiga, yaitu batch\_size (membagi dataset train menjadi beberapa bagian), epoch atau jumlah iterasi, dan optimizer untuk proses backpropagation atau penyesuaian bobot antar node atau antar layer.

GridSearchCV (cross-validation) digunakan untuk menentukan parameter terbaik dalam menghasilkan akurasi yang maksimal dengan membagi data latih menjadi 5 bagian, salah satunya digunakan untuk validasi.

Dari 16 kombinasi parameter, 4 dari parameter batch\_size dan 2 dari masing-masing parameter epochs dan optimizer, dihasilkan parameter terbaik, yaitu batch\_size = 2, epochs = 64, dan optimizer = ‘adam’, dengan akurasi data latih-validasi sebesar 92,9%.

**EVALUASI HASIL**

Pengukuran performansi model yang penulis lakukan ada empat, yaitu akurasi, presisi, recall, dan F1-score, dengan masing-masing rumus sebagai berikut:

Di mana TP adalah true positive (nilai prediksi dan actual sama-sama 1), FN adalah false negative (nilai prediksi 0, tetapi nilai actual 1), FP adalah false positive (nilai prediksi 1, tetapi nilai actual 0), dan mana TN adalah true negative (nilai prediksi dan actual sama-sama 0).

Chart, treemap chart

Description automatically generated

Setelah memasukkan data uji ke dalam model yang sudah dibuat dengan fungsi .predict(), hasil confusion matrix menunjukkan bahwa terdapat 90 data TN, 6 data FP, 13 data FN, dan 47 data TP.

Table

Description automatically generated with medium confidence

Dengan akurasi data uji sebesar 87,82% menunjukkan bahwa model tersebut sudah cukup baik dalam memprediksi nilai ‘Risk’ dari 7 fitur, walaupun ada sedikit overfitting karena akurasi data latih-validasi sekitar 5,08% lebih baik daripada data uji.

**Link Source Code, Laporan, dan Slide Presentasi :** <https://github.com/onedeetelyu/tugas_machine_learning>

**Link Video Presentasi :** <https://youtu.be/MuWaO5dUc6o>

**LAMPIRAN**

Fungsi nilai untuk StandardScaler()

rata-rata dari semua nilai , standar deviasi dari semua nilai

Fungsi RelU

Fungsi Sigmoid

Binary Crossentropy Loss

jumlah output, nilai output aktual, nilai output prediksi.

Optimizer untuk hyperparameter.

|  |  |
| --- | --- |
| tf.keras.optimizers.Adam(  learning\_rate=0.001,  beta\_1=0.9,  beta\_2=0.999,  epsilon=1e-07,  amsgrad=False,  name="Adam",  \*\*kwargs  ) | tf.keras.optimizers.RMSprop(  learning\_rate=0.001,  rho=0.9,  momentum=0.0,  epsilon=1e-07,  centered=False,  name="RMSprop",  \*\*kwargs  ) |

learning\_rate: Tingkat pembelajaran untuk penyesuaian bobot antar-node dan antar-layer. Nilai defaultnya **0,001**.

beta\_1: Laju peluruhan eksponensial untuk estimasi momen pertama. Nilai defaultnya **0.9**.

beta\_2: Laju peluruhan eksponensial untuk estimasi momen kedua. Nilai defaultnya **0,999**.

epsilon: Sebuah konstanta kecil untuk stabilitas numerik. Epsilon ini adalah *"topi epsilon"* di paper *“Kingma dan Ba”* ​​(di rumus sebelum bagian 2.1), bukan epsilon di algoritma 1. Nilai defaultnya **1e-7 (0,0000001)**.

amsgrad: Apakah akan menerapkan varian AMsGrad dari algoritma ini dari paper *"On the Convergence of Adam and Beyond"*. Nilai defaultnya **False**.

rho: Faktor pemotongan untuk gradien sebelumnya dan yang akan datang. Nilai defaultnya **0.9**.

momentum: Nilai skalar atau skalar Tensor. Nilai defaultnya **0,0**.

centered: Boolean. Jika bernilai True, gradien dinormalisasi dengan estimasi varians gradien; jika bernilai Salah, pada momen kedua yang tidak terpusat. Menyetel ini ke True dapat membantu pelatihan, tetapi sedikit memakan waktu komputasi dan memori. Nilai defaultnya **False**.

Sumber: <https://keras.io/api/optimizers/adam/>, <https://keras.io/api/optimizers/rmsprop/>.