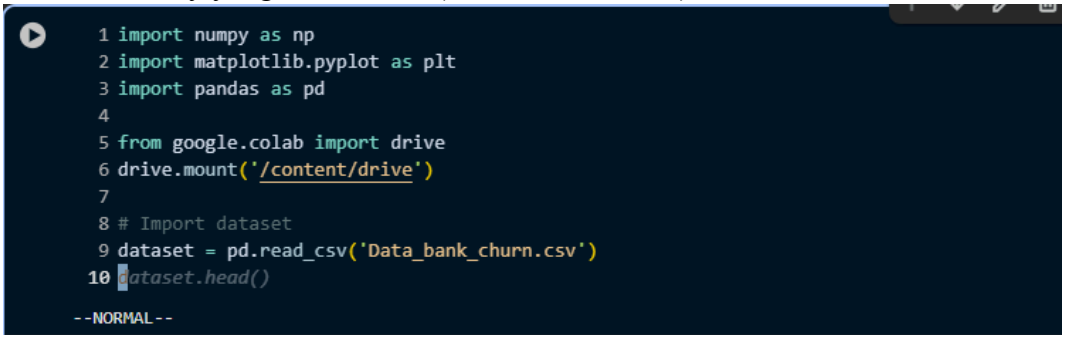
Michael Christian Handoko
J0404231117

Praktikum ANN dilakukan pada Google Colab yang secara otomatis sudah terinstall library yang dibutuhkan (Tensorflow, Keras).



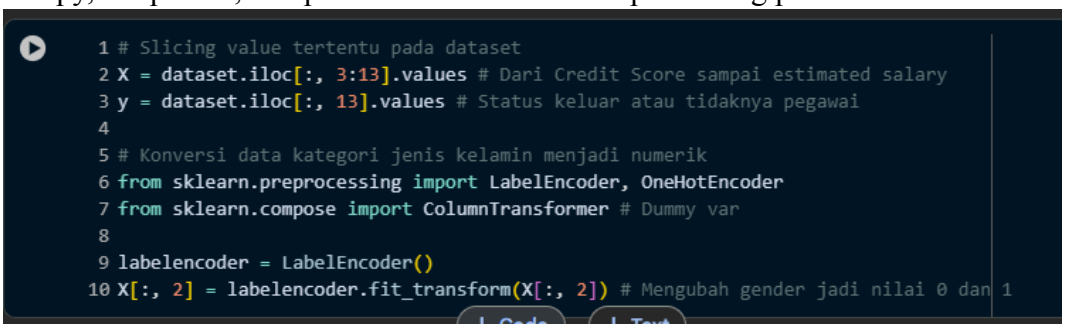
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from google.colab import drive
drive.mount('/content/drive')

# Import dataset
dataset = pd.read_csv('Data_bank_churn.csv')
dataset.head()
```

--NORMAL--

Cell kode pertama dilakukan impor dataset serta library dasar seperti numpy, matplotlib, dan pandas untuk melakukan processing pada dataset.

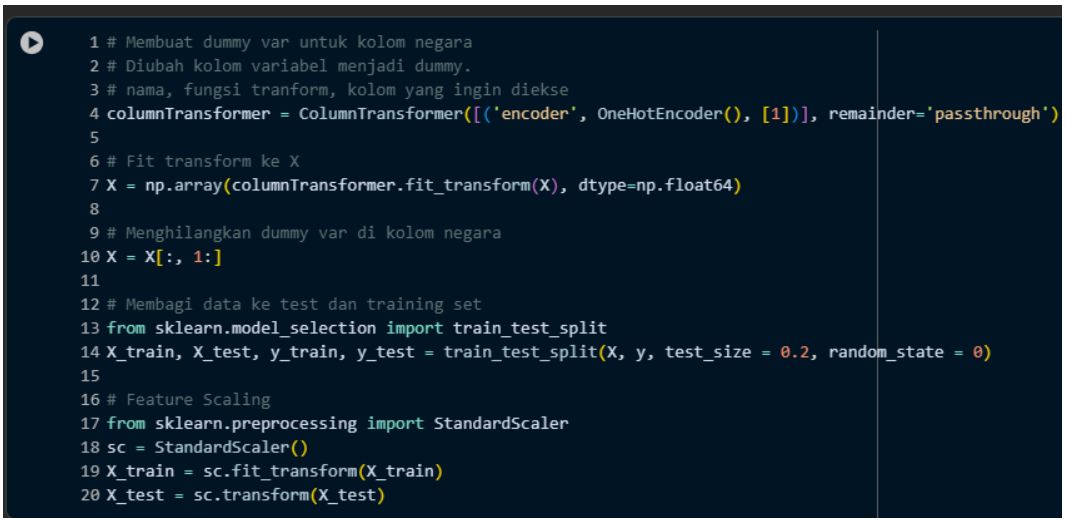


```
# Slicing value tertentu pada dataset
X = dataset.iloc[:, 3:13].values # Dari Credit Score sampai estimated salary
y = dataset.iloc[:, 13].values # Status keluar atau tidaknya pegawai

# Konversi data kategori jenis kelamin menjadi numerik
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer # Dummy var

labelencoder = LabelEncoder()
X[:, 2] = labelencoder.fit_transform(X[:, 2]) # Mengubah gender jadi nilai 0 dan 1
```

- Cell kode kedua dilakukan *slicing* untuk variabel independen X pada line kedua. Diperlukan kolom ke 3-12 (Dari credit score sampai estimated salary).
- Dilakukan juga *slicing* pada untuk variabel dependen y, yaitu kolom “exited” yang menandakan status keluar/tidaknya pegawai.
- Setelah itu, diubah kolom ‘Gender’ yang saat ini merupakan jenis data object menjadi nilai numerik menggunakan fungsi LabelEncoder.



```
# Membuat dummy var untuk kolom negara
# Diubah kolom variabel menjadi dummy.
# nama, fungsi tranform, kolom yang ingin diekse
columnTransformer = ColumnTransformer([('encoder', OneHotEncoder(), [1])], remainder='passthrough')

# Fit transform ke X
X = np.array(columnTransformer.fit_transform(X), dtype=np.float64)

# Menghilangkan dummy var di kolom negara
X = X[:, 1:]

# Membagi data ke test dan training set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

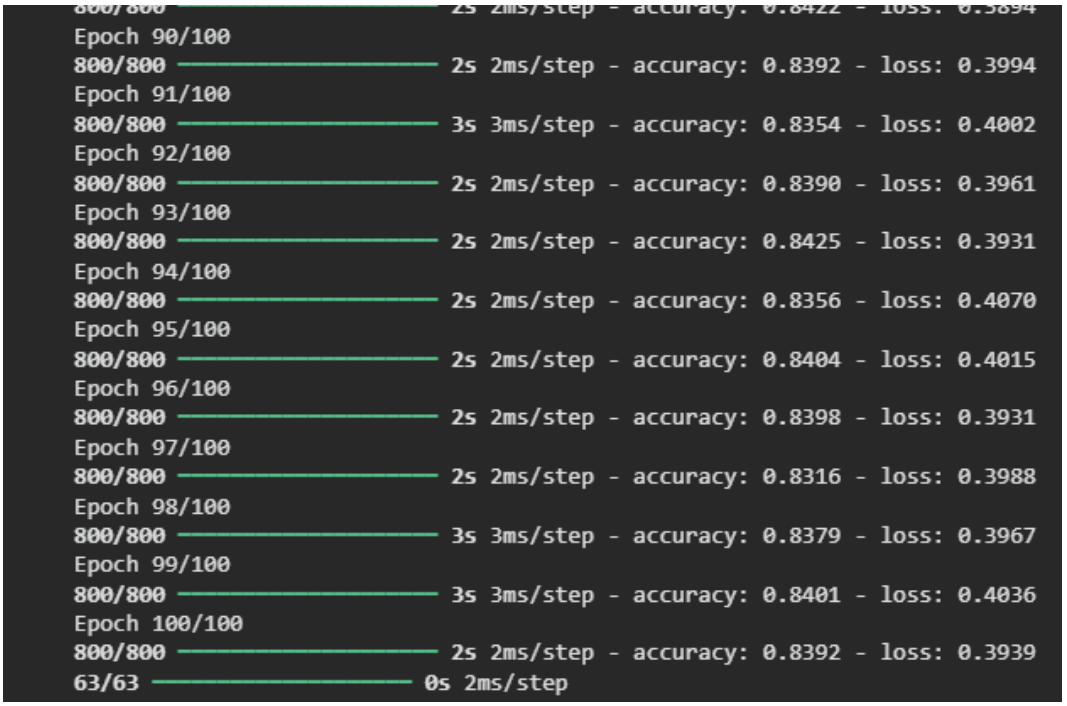
- Cell kode ketiga dilakukan pembuatan *dummy variabel* untuk kolom Geography. Fungsinya untuk mengubah urutan tiap nilai geografi suatu kostumer menjadi angka.
- Nilai tersebut kemudian dipecah menjadi tiga kolom dengan masing-masing nilai nol dan satu pada line 7. Hal ini bertujuan untuk mengidentifikasi suatu kostumer hanya dengan angka pada kolom yang sudah disiapkan dan diidentifikasi.
    - pada modul, ditulis dtype=np.float(). Ini akan memunculkan error dimana fungsi float sudah di-*deprecate*. Ganti menjadi dtype=np.float64().
- Pada line 10 dilakukan penghapusan salah satu kolom di variabel X untuk menghindari *dummy variable trap* (relasi yang sempurna antar variabel sehingga tidak memiliki keunikan pada data). Kolom yang dihilangkan adalah kolom di index 1.
- Line 13-14 dilakukan train-test split dimana ukuran test sebesar 20%.
- Line 17-20 dilakukan feature scaling menggunakan StandardScaler() untuk menghindari disproporsi pada data.

```
1 # Import Lib Keras dkk
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import Dense
4
5 # Inisialisasi ANN
6 classifier = Sequential()
7
8 # Menambahkan input layer dan hidden layer pertama
9 classifier.add(Dense(units = 6, kernel_initializer = 'uniform', activation = 'relu', input_dim = 11))
10
11 # Hidden layer kedua
12 classifier.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'relu'))
13
14 # Menambah output layer
15 classifier.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))
16
17 # Run ANN
18 classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
19
20 # Fitting ANN ke training set
21 classifier.fit(X_train, y_train, batch_size = 10, epochs = 100)
22
23 # Memprediksi hasil test set
24 y_pred = classifier.predict(X_test)
25 y_pred = (y_pred > 0.5)
26
27 # Membuat confusion matrix
28 from sklearn.metrics import confusion_matrix
29 cm = confusion_matrix(y_test, y_pred)
30
31
```

- Pada cell kode keempat, line 2-3 diimpor library Sequential dan Dense yang diperlukan untuk keperluan neural network model dan layer.
    - Pada praktikum ini, Dense merupakan salah satu layer jaringan neural yang dipasang 6 layer.
    - Sequential merupakan salah satu model ANN yang merupakan tumpukan lapisan secara linear.
- Line 6, 9, 12, dan 15 diinisiasi model Sequential dan ditambahkan input layer, hidden layer, dan output layer.

    - Hidden layer berfungsi sebagai lapisan pemroses antara input dan output layer. Hidden layer berguna untuk belajar pola representasi yang kompleks dari data input dan melakukan ekstrasi fitur dari data mentah
    - Dalam jaringan saraf tiruan, kernel merujuk pada bobot (weights) dari layer. Kernel Initializer uniform berarti setiap nilai dalam rentang tertentu memiliki peluang yang sama untuk dipilih
    - Pada hidden layer diinisiasi activation relu (Rectified Linear Unit) yang mengeluarkan output 0 jika input kurang dari 0, dan mengeluarkan input itu sendiri jika input lebih besar dari atau sama dengan 0.
    - Pada layer output, activation sigmoid dapat mengeluarkan output nilai antara 0 dan 1, yang dapat diinterpretasikan sebagai probabilitas. Aktivasi jenis ini berguna dalam kasus ini dimana keluarannya merupakan probabilitas.
- Pada line 17 dilakukan kompilasi menggunakan optimizer adam, fungsi loss binary_crossentropy, dan accuracy sebagai metrik evaluasi.
    - Optimizer adam (Adaptive Moment Estimation) mengadaptasi nilai learning untuk tiap parameter dan menggunakan momentum untuk mempercepat proses pelatihan.
    - binary_crossentropy cocok untuk klasifikasi biner
- Line 21 merupakan Fitting ANN selama 100 epoch dengan ukuran batch 10.
    - batch 10 membagi data menjadi batch yang lebih kecil. Dalam kasus ini dibagi menjadi 10. Semakin kecil ukurannya maka pembaruannya lebih sering dan dapat menghasilkan data yang noisy.
    - epoch mewakili satu kali putaran penuh melalui seluruh dataset pelatihan. Dalam kasus ini diberikan 8000 contoh data pelatihan dan batch size 10, maka satu epoch akan terdiri dari 8000/10 = 800 iterasi. Melatih banyak epoch memungkinkan model untuk belajar lebih banyak dari data test.
- Line 24 dipakai kembali untuk memprediksi output untuk set X_test. Prediksi kemudian diubah menjadi nilai biner 0 atau 1 dan diprediksi lebih besar dari 0.5
- Confusion Matrix dibuat untuk mengevaluasi kinerja pengklasifikasi dengan membandingkan nilai sebenarnya (y_test) dengan nilai prediksi (y_pred).

Hasil:



- Akurasi training stabil di kisaran 83–85% setelah sekitar 20–30 epoch.
- Loss juga stabil di kisaran 0.38–0.40, artinya model sudah menemukan pola yang cukup baik.