Kode program yang akan diajarkan di sini bisa diunduh di tautan [berikut ini](https://colab.research.google.com/drive/1DhpiG1KdEQZPjDk_rAcqO_rM9btPOrJG?usp=sharing). Silahkan buat salinan dari kode ini dan jalankan di Google Colab Anda.

**Tujuan**

Pada latihan ini kita akan menggunakan logistic regression untuk memprediksi apakah seseorang akan membeli setelah melihat iklan sebuah produk.

**Tahapan Latihan**

Tahapan yang dilalui dalam latihan kali ini adalah sebagai berikut:

1. Ubah dataset menjadi Dataframe.
2. Hapus kolom 'User ID'.
3. Pisahkan atribut dan label.
4. Latih model Logistic Regression.
5. Evaluasi akurasi model.

**Codelab**

Dataset untuk latihan bisa Anda unduh pada [tautan](https://www.kaggle.com/dragonheir/logistic-regression) berikut. Seperti biasa, unggah dataset Social\_Network\_Ads pada session storage Google Colab.

Setelah kita mengunggah berkas data pada Colab kita akan mengubah dataset menjadi dataframe Pandas. Jangan lupa untuk mengimpor library dasar dan menyesuaikan path/lokasi datanya ya.

1. import pandas as pd
3. # membaca dataset dan mengubahnya menjadi dataframe
4. df = pd.read\_csv('Social\_Network\_Ads.csv')

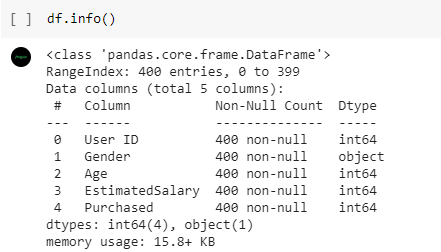
Pada cell selanjutnya gunakan fungsi head() pada dataframe untuk melihat 5 baris pertama dari dataset.

1. df.head()

Hasil dari fungsi df.head() seperti di bawah ini.

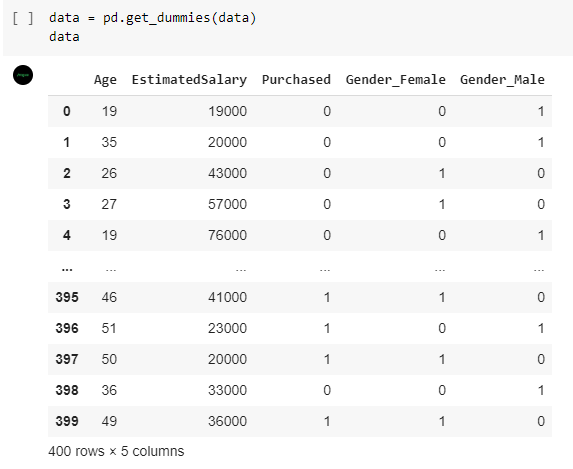
Kita juga perlu melihat apakah ada nilai yang kosong pada setiap atribut dengan menggunakan fungsi info(). Dapat dilihat bahwa nilai pada semua kolom sudah lengkap.

1. df.info()

Sedangkan tampilan hasil dari fungsi df.info() sebagai berikut.  


Pada dataset terdapat kolom ‘User ID’. Kolom tersebut merupakan atribut yang tidak penting untuk dipelajari oleh model sehingga perlu dihilangkan. Untuk menghilangkan kolom dari dataframe, gunakan fungsi drop. Jangan lupa panggil fungsi get\_dummies() untuk melakukan proses O*ne-Hot Encoding* karena label pada dataset kita merupakan data kategorikal.

1. # drop kolom yang tidak diperlukan
2. data = df.drop(columns=['User ID'])
4. # jalankan proses *one-hot encoding*dengan pd.get\_dummies()
5. data = pd.get\_dummies(data)
6. data

Ketika kode di atas dijalankan hasilnya seperti di bawah ini.  


Kemudian kita pisahkan antara atribut dan label.

1. # pisahkan atribut dan label
2. predictions = ['Age' , 'EstimatedSalary' , 'Gender\_Female' , 'Gender\_Male']
3. X = data[predictions]
4. y = data['Purchased']

Sebelum kita membagi data menjadi train dan test set, kita perlu melakukan standardisasi data seperti yang sudah dijelaskan di modul [sebelumnya](https://www.dicoding.com/academies/184/tutorials/8352?from=8347).

1. # lakukan normalisasi terhadap data yang kita miliki
2. from sklearn.preprocessing import StandardScaler
3. scaler = StandardScaler()
4. scaler.fit(X)
5. scaled\_data = scaler.transform(X)
6. scaled\_data = pd.DataFrame(scaled\_data, columns= X.columns)
7. scaled\_data.head()

Sekarang, kita akan membagi data menjadi train dan test set dengan fungsi train\_test\_split yang disediakan SKLearn.

1. from sklearn.model\_selection import train\_test\_split
3. # bagi data menjadi train dan test untuk setiap atribut dan label
4. X\_train, X\_test, y\_train, y\_test = train\_test\_split(scaled\_data, y, test\_size=0.2, random\_state=1)

Setelah membagi data, kita buat model dengan membuat sebuah objek logistic regression. Setelah model dibuat, kita bisa melatih model kita dengan train set menggunakan fungsi fit().

1. from sklearn import linear\_model
3. # latih model dengan fungsi fit
4. model = linear\_model.LogisticRegression()
5. model.fit(X\_train, y\_train)

Setelah model dilatih, kita bisa menguji akurasi model pada test set dengan memanggil fungsi score() pada objek model.

1. # uji akurasi model
2. model.score(X\_test, y\_test)

Sehingga hasilnya sebagai berikut.

