Nama: Ghaitsa Maulidina Shofa
NPM: 2006597014
Kelas PDB
Tugas 1

File adult.data berisikan data orang-orang dengan berbagai karakteristik. Dari karakteristik setiap orang, kemudian dianotasikan penghasilan mereka apakah di bawah 50 ribu dolar atau di atas 50 ribu dolar per tahunnya. Dari data ini kita bisa melatih mesin untuk memprediksi apakah seseorang memiliki penghasilan di bawah 50 ribu dolar atau di atas 50 ribu dolar dari karakteristik datanya. What you need to do are:

1. Understand the data. Tell us about the data. Visualize it. Describe it.
2. Lakukan penambangan data menggunakan SVM dan Neural Network. Bagaimana hasilnya. Mana kah yang memberikan performance yang lebih baik, apa sebabnya.

In [409]:
```
# Import
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as scp
from numpy.polynomial.polynomial import polyfit
from sklearn.model_selection import train_test_split
from sklearn import svm as svm
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score
from sklearn.preprocessing import StandardScaler
import math
```

1. Understand the data. Tell us about the data. Visualize it. Describe it.

Data yang digunakan adalah adult.csv. File tersebut berisikan data mengenai informasi diri banyak orang yang akan dicari hubungannya dengan penghasilan mereka dalam satu tahun dalam mata uang dollar. Informasi diri terdiri dari umur, kelas pekerjaan, fnlwgt value, edukasi, status pernikahan, pekerjaan, hubungan, ras, jenis kelamin, capital-gain, capital-loss, jam kerja per minggu, asal negara, dan threshold pemasukan per tahun (di bawah 50 ribu dolar atau di atasnya).

In [410]:
```
df = pd.read_csv("adult.csv",
                 sep=", ",
                 names=['age', 'workclass', 'fnlwgt', 'education', 'education-num', 'marital-status', 'occupation',
                        'relationship', 'race', 'sex', 'capital-gain', 'capital-loss', 'hours-per-week', 'native-country',
                        'income'],
                 engine='python')
print("The shape of this data is " + str(df.shape))
display(df.head())
```

The shape of this data is (32561, 15)

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

Membersihkan data categorical

Pada tugas kali ini, kita memiliki acuan value data yang benar yaitu melalui dokumen adult.names . Value-value yang didapat dari adult.csv masih belum sesuai dengan dokumen tersebut. Oleh karena itu kita perlu mencocokkan value yang ada dengan yang seharusnya ada.

Cara membersihkan datanya adalah dengan mencari tahu berapa jumlah tiap value categorical pada setiap kolom. Dengan kode yang sedemikian rupa dirangkai agar menampilkan hasil yang diinginkan, kita bisa melihat apakah ada value yang tidak seharusnya berada di sana. Misalnya, value yang salah ketik, tanda tanya (?), dan lain-lain.

Value-value tersebut tidak akan saya buang, melainkan saya akan mengganti valuenya dengan modus data. Di bawah ini adalah pengecekan data tidak sesuai dan pembersihannya.

Kolom 'workclass'

Value yang seharusnya: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
Value tidak sesuai: "?" dan "Prvate"
Cara membersihkan:

- "Prvate" akan diubah menjadi "Private" (asumsi salah ketik)
- "?" diubah menjadi modus dari data workclass, yaitu "Private"

Di bawah ini adalah before dan after hasil pembersihan datanya.

In [411]:
```
# ==== BEFORE =====
display(df.groupby('workclass').size())
```

```
workclass
?                    1836
Federal-gov           960
Local-gov            2093
Never-worked            7
Private             22695
Prvate                  1
Self-emp-inc         1116
Self-emp-not-inc     2541
State-gov            1298
Without-pay            14
dtype: int64
```

In [412]:
```
# ==== AFTER ====
df['workclass'] = df['workclass'].replace(["?", "Prvate"], "Private")
display(df.groupby('workclass').size())
```

```
workclass
Federal-gov           960
Local-gov            2093
Never-worked            7
Private             24532
Self-emp-inc         1116
Self-emp-not-inc     2541
State-gov            1298
Without-pay            14
dtype: int64
```

Kolom 'education'

Value yang seharusnya: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.

Value tidak sesuai: "?" dan "Some-colleges"
Cara membersihkan: >"Some-colleges" akan diubah menjadi "Some-college" (asumsi salah ketik)
>"?" diubah menjadi modusnya, yaitu "HS-grad"
Di bawah ini adalah before dan after hasil pembersihan datanya.

In [413]:
```
# BEFORE
df.groupby('education').size()
```

Out[413]:
```
education
10th              933
11th             1175
12th              433
1st-4th           168
5th-6th           333
7th-8th           646
9th               514
?                   1
Assoc-acdm       1067
Assoc-voc        1382
Bachelors        5354
Doctorate         413
HS-grad         10501
Masters          1723
Preschool          51
Prof-school       576
Some-college     7290
Some-colleges       1
dtype: int64
```

In [414]:
```
# ====AFTER====
df['education'] = df['education'].replace(["?"], "HS-grad")
df['education'] = df['education'].replace(["Some-colleges"], "Some-college")
df.groupby('education').size()
```

Out[414]:
```
education
10th              933
11th             1175
12th              433
1st-4th           168
5th-6th           333
7th-8th           646
9th               514
Assoc-acdm       1067
Assoc-voc        1382
Bachelors        5354
Doctorate         413
HS-grad         10502
Masters          1723
Preschool          51
Prof-school       576
Some-college     7291
dtype: int64
```

Kolom 'marital-status'

Value yang seharusnya: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
Value tidak sesuai: "Married-cvi-spouse" dan "Never-mind"
Cara membersihkan:

- "Married-cvi-spouse" akan diubah menjadi "Married-civ-spouse" (asumsi salah ketik)
- "Never-mind" diubah menjadi modus dari data workclass, yaitu "Private"

Di bawah ini adalah before dan after hasil pembersihan datanya.

In [415]:
```
# ====BEFORE====
df.groupby('marital-status').size()
```

Out[415]:
```
marital-status
Divorced                  4443
Married-AF-spouse           23
Married-civ-spouse       14975
Married-cvi-spouse           2
Married-spouse-absent      418
Never-married            10682
Never-mind                   1
Separated                 1025
Widowed                    993
dtype: int64
```

In [416]:
```
# ====AFTER====
df['marital-status'] = df['marital-status'].replace(["Married-cvi-spouse", "Never-mind"], "Married-civ-spouse")
df.groupby('marital-status').size()
```

Out[416]:
```
marital-status
Divorced                  4443
Married-AF-spouse           23
Married-civ-spouse       14977
Married-spouse-absent      418
Never-married            10682
Separated                 1025
Widowed                    993
dtype: int64
```

Kolom 'occupation'

Value yang seharusnya: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
Value tidak sesuai: "?" dan "? "
Cara membersihkan:

- "?" diubah menjadi modus dari data workclass, yaitu "Craft-repair"

Di bawah ini adalah before dan after hasil pembersihan datanya.

In [417]:
```
# BEFORE
df.groupby('occupation').size()
```

Out[417]:
```
occupation
?                    1843
?                       1
Adm-clerical         3769
Armed-Forces            9
Craft-repair         4099
Exec-managerial      4066
Farming-fishing       994
Handlers-cleaners    1370
Machine-op-inspct    2002
Other-service        3295
Priv-house-serv       149
Prof-specialty       4140
Protective-serv       649
Sales                3650
Tech-support          928
Transport-moving     1597
dtype: int64
```

In [418]:
```
# AFTER
df['occupation'] = df['occupation'].replace(["?", "? "], "Craft-repair")
df.groupby('occupation').size()
```

Out[418]:
```
occupation
Adm-clerical         3769
Armed-Forces            9
Craft-repair         5943
Exec-managerial      4066
Farming-fishing       994
Handlers-cleaners    1370
Machine-op-inspct    2002
Other-service        3295
Priv-house-serv       149
Prof-specialty       4140
Protective-serv       649
Sales                3650
Tech-support          928
Transport-moving     1597
dtype: int64
```

Kolom 'relationship'

Value yang seharusnya: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
Value tidak sesuai: -
Cara membersihkan:

- Tidak ada yang perlu dibersihkan

In [419]:
```
df.groupby('relationship').size()
```

Out[419]:
```
relationship
Husband           13193
Not-in-family      8305
Other-relative      981
Own-child          5068
Unmarried          3446
Wife               1568
dtype: int64
```

Kolom 'race'

Value yang seharusnya: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
Value tidak sesuai: "Yellow"
Cara membersihkan:

- "Yellow" akan diubah menjadi "Other" (asumsi maksud dari yellow tidak diketahui dan bukan termasuk ras modus)

Di bawah ini adalah before dan after hasil pembersihan datanya.

In [420]:
```
# BEFORE
df.groupby('race').size()
```

Out[420]:
```
race
Amer-Indian-Eskimo      311
Asian-Pac-Islander     1039
Black                  3124
Other                   271
White                 27815
Yellow                    1
dtype: int64
```

In [421]:
```
# AFTER
df['race'] = df['race'].replace(["Yellow"], "Other")
df.groupby('race').size()
```

Out[421]:
```
race
Amer-Indian-Eskimo      311
Asian-Pac-Islander     1039
Black                  3124
Other                   272
White                 27815
dtype: int64
```

Kolom 'sex'

Value yang seharusnya: Female, Male
Value tidak sesuai: "?" dan "Male/Female"
Cara membersihkan:

- "Male/Female" diubah menjadi modus dari data workclass, yaitu "Male"
- "?" diubah menjadi modus dari data workclass, yaitu "Male"

Di bawah ini adalah before dan after hasil pembersihan datanya.

In [422]:
```
# BEFORE
df.groupby('sex').size()
```

Out[422]:
```
sex
?                1
Female       10771
Male         21788
Male/Female      1
dtype: int64
```

In [423]:
```
# AFTER
df['sex'] = df['sex'].replace(["?", "Male/Female"], "Male")
df.groupby('sex').size()
```

Out[423]:
```
sex
Female    10771
Male      21790
dtype: int64
```

Kolom 'native-country'

Value yang seharusnya: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands
Value tidak sesuai: "United-States-America" dan "?"
Cara membersihkan:

- "United-States-America" akan diubah menjadi "United-States" (asumsi salah ketik)
- "?" diubah menjadi modus dari data workclass, yaitu "United-States"

Di bawah ini adalah before dan after hasil pembersihan datanya.

In [424]:
```
# BEFORE
df.groupby('native-country').size()
```

Out[424]:
```
native-country
?                              583
Cambodia                        19
Canada                         121
China                           75
Columbia                        59
Cuba                            95
Dominican-Republic              70
Ecuador                         28
El-Salvador                    106
England                         90
France                          29
Germany                        137
Greece                          29
Guatemala                       64
Haiti                           44
Holand-Netherlands               1
Honduras                        13
Hong                            20
Hungary                         13
India                          100
Iran                            43
Ireland                         24
Italy                           73
Jamaica                         81
Japan                           62
Laos                            18
Mexico                         643
Nicaragua                       34
Outlying-US(Guam-USVI-etc)      14
Peru                            31
Philippines                    198
Poland                          60
Portugal                        37
Puerto-Rico                    114
Scotland                        12
South                           80
Taiwan                          51
Thailand                        18
Trinadad&Tobago                 19
United-States                29169
United-States-America            1
Vietnam                         67
Yugoslavia                      16
dtype: int64
```

In [425]:
```
# AFTER
df['native-country'] = df['native-country'].replace(["United-States-America", "?"], "United-States")
df.groupby('native-country').size()
```

Out[425]:
```
native-country
Cambodia                        19
Canada                         121
China                           75
Columbia                        59
Cuba                            95
Dominican-Republic              70
Ecuador                         28
El-Salvador                    106
England                         90
France                          29
Germany                        137
Greece                          29
Guatemala                       64
Haiti                           44
Holand-Netherlands               1
Honduras                        13
Hong                            20
Hungary                         13
India                          100
Iran                            43
Ireland                         24
Italy                           73
Jamaica                         81
Japan                           62
Laos                            18
Mexico                         643
Nicaragua                       34
Outlying-US(Guam-USVI-etc)      14
Peru                            31
Philippines                    198
Poland                          60
Portugal                        37
Puerto-Rico                    114
Scotland                        12
South                           80
Taiwan                          51
Thailand                        18
Trinadad&Tobago                 19
United-States                29753
Vietnam                         67
Yugoslavia                      16
dtype: int64
```

In [426]:
```
df.groupby('income').size()
```

Out[426]:
```
income
<=50K    24720
>50K      7841
dtype: int64
```

Membersihkan data continuous

Masih merujuk kepada dokumen data.names , kita bisa mencari tahu kolom mana saja yang tipe datanya kontinu (berupa integer). Rencana saya dalam membersihkan kolom dengan value kontinu adalah dengan menghapus kolom yang valuenya tidak relevan dengan tujuan dari tugas ini.

Sebelum data dibersihkan, cocokan terlebih dahulu tipe datanya dengan tipe data yang seharusnya (mengacu ke dokumen data.names). Ternyata, kolom 'capital-gain' masih bertipe data object yang mana seharusnya bertipe int64 (lihat data di bawah). Setelah diubah, operasi pembersihan data kontinu baru bisa dimulai.

In [427]:
```
display(df.dtypes)
```

```
age                int64
workclass         object
fnlwgt             int64
education         object
education-num      int64
marital-status    object
occupation        object
relationship      object
race              object
sex               object
capital-gain      object
capital-loss       int64
hours-per-week     int64
native-country    object
income            object
dtype: object
```

Kolom 'capital-gain'

Ternyata yang membuat capital-gain menjadi tipe data object adalah adanya value '?'. Oleh karena itu, kita ganti terlebih dahulu value '?' dengan modus dari data. Akan tetapi, setelah dicek size dari setiap value, '?' hanya terdapat di satu baris data yang mana efeknya tidak signifikan pada keseluruhan data.

In [428]:
```
# Menunjukkan jumlah setiap value pada capital-gain
display(df.groupby('capital-gain').size())
```

```
capital-gain
0        29847
10520       43
1055        25
10566        6
10605       12
         ...
9386        22
9562         4
991          5
99999      159
?            1
Length: 121, dtype: int64
```

Kemudian, berdasarkan observasi saya, baris bervalue '0' yang berjumlah 29847 data memenuhi ~92% dari baris tabel. Karena angka '0' terlalu banyak dan dirasa tidak akan menghasilkan sesuatu yang bermanfaat yang berkaitan dengan soal tugas, saya memutuskan untuk menghapus kolom capital-gain. Di bawah ini merupakan bentul tabel setelah capital-gain dihapus

In [429]:
```
# Menghapus kolom capital-gain
df.drop(['capital-gain'], axis=1, inplace=True)
```

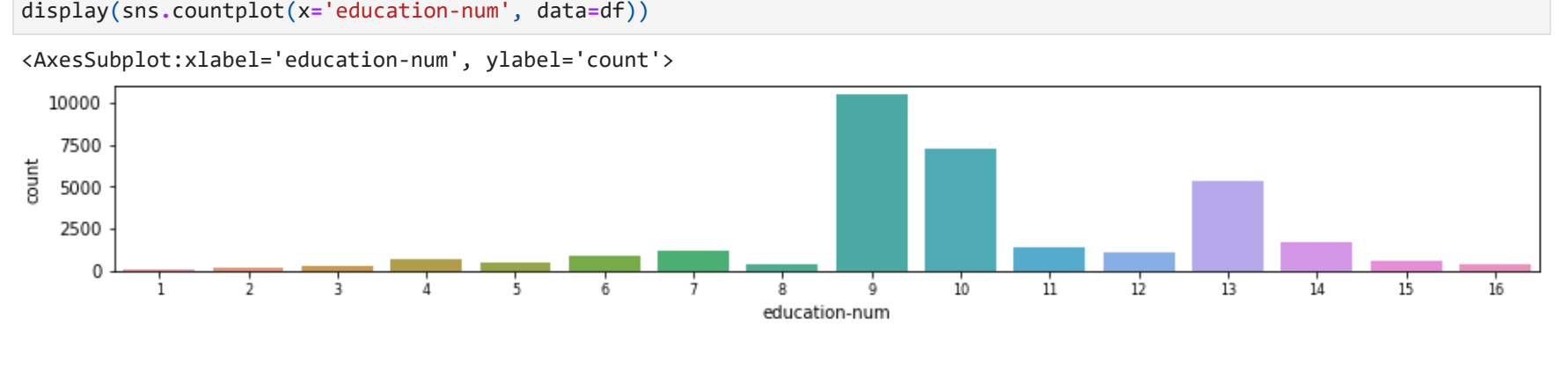
Setelah permasalahan ketidakcocokan tipe data selesai, saya akan mengecek kolom-kolom dengan data kontinu lainnya.

Kolom 'age'

Berdasarkan data di bawah, karena data terlihat merata, maka 'age' bisa langsung kita pakai.

In [430]:
```
plt.figure(figsize=(15,2))
plt.xticks(size=8)
plt.yticks(size=10)
sns.countplot(x='age', data=df)
```

Out[430]: <AxesSubplot:xlabel='age', ylabel='count'>



Kolom 'fnlwgt'

Remove kolom 'fnlwgt' karena datanya sangat tersebar sehingga akan sulit menarik informasi dari data ini.

In [431]:
```
display(df.groupby('fnlwgt').size())
# Remove 'fnwlgt'
df.drop(['fnlwgt'], axis=1, inplace=True)
```

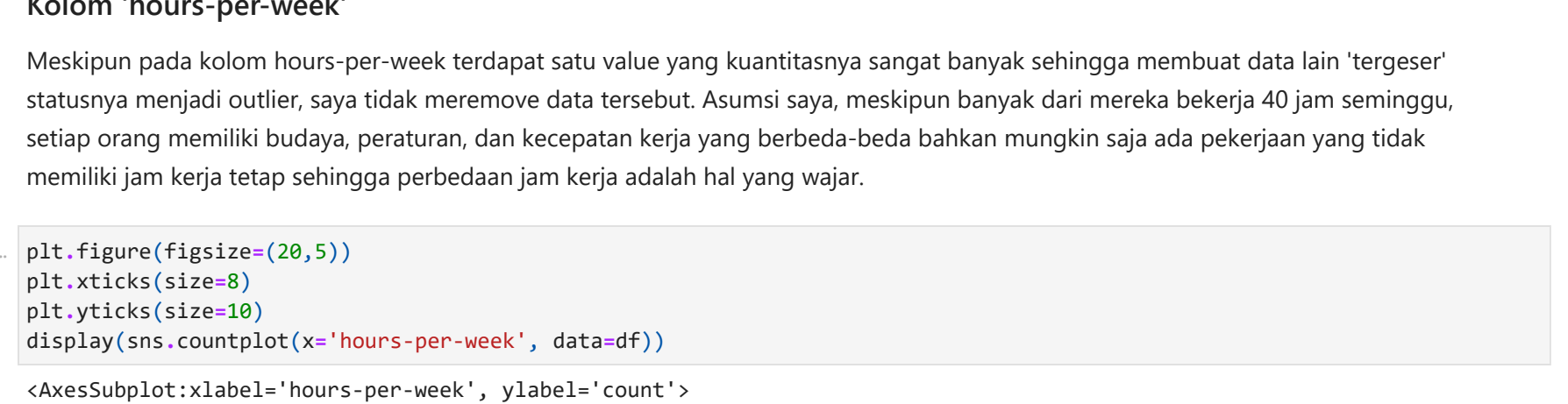
```
fnlwgt
12285       1
13769       1
14878       1
18827       1
19214       1
           ..
1268339     1
1366120     1
1455435     1
1484705     1
22296450    1
Length: 21648, dtype: int64
```

Kolom 'education-num'

Berdasarkan data di bawah, karena data terlihat merata, maka data bisa langsung kita pakai.

In [432]:
```
plt.figure(figsize=(15,2))
plt.xticks(size=8)
plt.yticks(size=10)
display(sns.countplot(x='education-num', data=df))
```

<AxesSubplot:xlabel='education-num', ylabel='count'>



Kolom 'capital-loss'

Berdasarkan observasi saya, baris bervalue '0' yang berjumlah 31041 data memenuhi ~95% dari baris tabel. Karena angka '0'-nya terlalu banyak dan dirasa tidak menghasilkan sesuatu yang bermanfaat yang berkaitan dengan soal tugas, saya memutuskan untuk menghapus kolom capital-loss.

In [433]:
```
# Terdapat 31041 value '0' pada capital-loss
display(df.groupby('capital-loss').size())
```

```
capital-loss
0       31041
155         1
213         4
323         3
419         3
        ...
3004        2
3683        2
3770        2
3900        2
4356        3
Length: 93, dtype: int64
```

Di bawah ini merupakan bentuk tabel setelah capital-loss dihapus

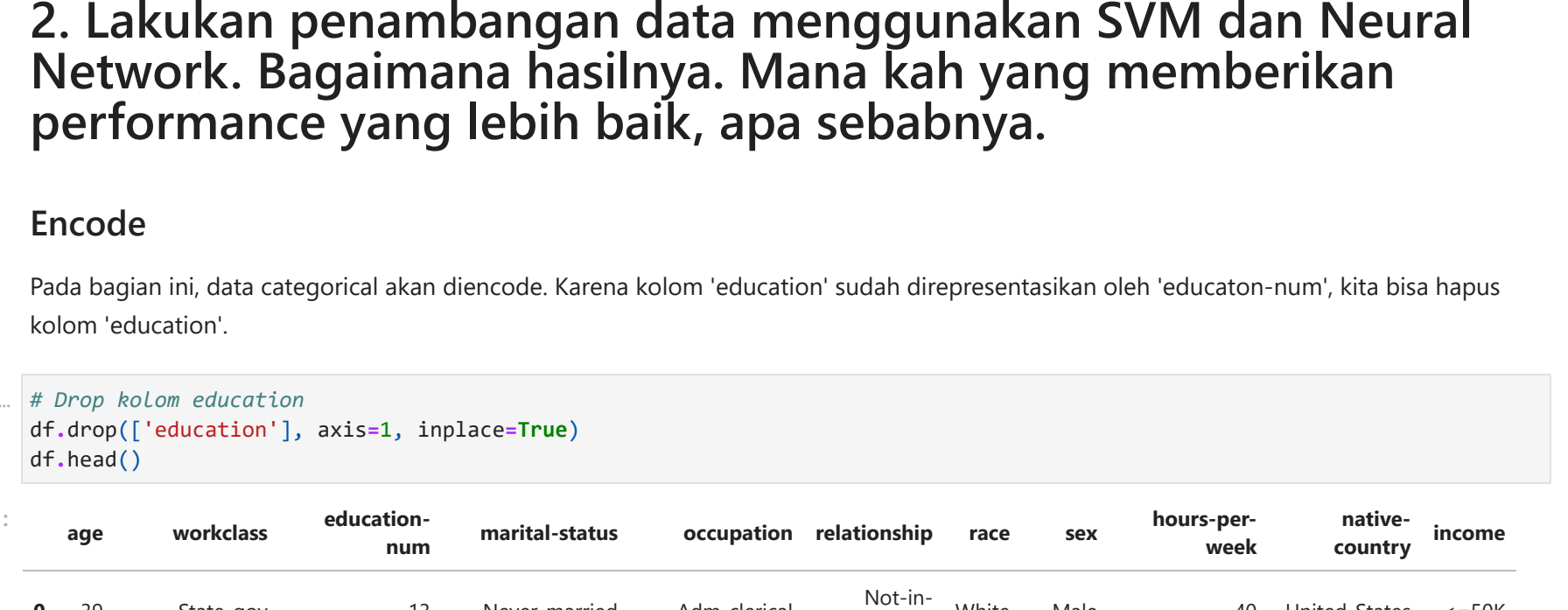
In [434]:
```
# Menghapus kolom capital-loss
df.drop(['capital-loss'], axis=1, inplace=True)
```

Kolom 'hours-per-week'

Meskipun pada kolom hours-per-week terdapat satu value yang kuantitasnya sangat banyak sehingga membuat data lain 'tergeser' statusnya menjadi outlier, saya tidak meremove data tersebut. Asumsi saya, meskipun banyak dari mereka bekerja 40 jam seminggu, setiap orang memiliki budaya, peraturan, dan kecepatan kerja yang berbeda-beda bahkan mungkin saja ada pekerjaan yang tidak memiliki jam kerja tetap sehingga perbedaan jam kerja adalah hal yang wajar.

In [435]:
```
plt.figure(figsize=(20,5))
plt.xticks(size=8)
plt.yticks(size=10)
display(sns.countplot(x='hours-per-week', data=df))
```

<AxesSubplot:xlabel='hours-per-week', ylabel='count'>



Hasil pembersihan data continuous

Berikut adalah hasil akhir tabel setelah kolom fnlwgt, capital-gain, capital-loss, kolom dihapus.

In [436]:
```
df.head()
```

Out[436]:

	age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	hours-per-week	native-country	income
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	40	United-States	<=50K
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13	United-States	<=50K
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40	United-States	<=50K
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	40	United-States	<=50K
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	40	Cuba	<=50K

2. Lakukan penambangan data menggunakan SVM dan Neural Network. Bagaimana hasilnya. Mana kah yang memberikan performance yang lebih baik, apa sebabnya.

Encode

Pada bagian ini, data categorical akan diencode. Karena kolom 'education' sudah direpresentasikan oleh 'educaton-num', kita bisa hapus kolom 'education'.

In [437]:
```
# Drop kolom education
df.drop(['education'], axis=1, inplace=True)
df.head()
```

Out[437]:

	age	workclass	education-num	marital-status	occupation	relationship	race	sex	hours-per-week	native-country	income
0	39	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40	United-States	<=50K
1	50	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13	United-States	<=50K
2	38	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40	United-States	<=50K
3	53	Private	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	40	United-States	<=50K
4	28	Private	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	40	Cuba	<=50K

Encode kolom income

Kita akan encode income dengan label encoding yaitu 0 untuk <=50K dan 1 untuk >50K

In [438]:
```
df_encoded = df.copy()
df_encoded['income'] = df_encoded['income'].replace('<=50K', 0).replace('>50K', 1)
```

Encode kategorikal tidak ordinal

Encode data kategorikal yang tidak ordinal (tidak ada urutannya) menggunakan one-hot-encoding

In [439]:
```
df_encoded = pd.get_dummies(df_encoded)
display(df_encoded.head())
```

	age	education-num	hours-per-week	income	workclass_Federal-gov	workclass_Local-gov	workclass_Never-worked	workclass_Private	workclass_Self-emp-inc	workclass_Self-emp-not-inc
0	39	13	40	0	0	0	0	0	0	0
1	50	13	13	0	0	0	0	0	0	1
2	38	9	40	0	0	0	0	1	0	0
3	53	7	40	0	0	0	0	1	0	0
4	28	13	40	0	0	0	0	1	0	0

5 rows × 87 columns

Korelasinya dengan 'income'

Pada bagian ini, kita akan mencari tahu berapa nilai korelasi antara column-column di atas dengan income. Menurut https://journals.lww.com/anesthesia-analgesia/fulltext/2018/05000/correlation_coefficients__appropriate_use_and.50.aspx, data yang bisa diabaikan adalah ketika korelasinya dengan target bernilai < 0.1

In [440]:
```
all_corr = df_encoded.corr()['income']
all_corr
```

Out[440]:
```
age                             0.234037
education-num                   0.335154
hours-per-week                  0.229597
income                          1.000000
workclass_Federal-gov           0.059372
                                  ...
native-country_Thailand        -0.004078
native-country_Trinadad&Tobago -0.007660
native-country_United-States    0.038942
native-country_Vietnam         -0.017649
native-country_Yugoslavia       0.006959
Name: income, Length: 87, dtype: float64
```

Filter data dengan batas nilai korelasi yang sudah ditentukan.

In [441]:
```
filtered_corr = all_corr[(all_corr <= -0.1) | (all_corr >= 0.1)]
df_filtered = df_encoded[filtered_corr.index]
# Berikut adalah atribut yang sudah terfilter
filtered_corr
```

Out[441]:
```
age                                  0.234037
education-num                        0.335154
hours-per-week                       0.229597
income                               1.000000
workclass_Private                   -0.125573
workclass_Self-emp-inc               0.139467
marital-status_Divorced             -0.126995
marital-status_Married-civ-spouse    0.444659
marital-status_Never-married        -0.318411
occupation_Exec-managerial           0.214861
occupation_Other-service            -0.156348
occupation_Prof-specialty            0.185866
relationship_Husband                 0.401035
relationship_Not-in-family          -0.188497
relationship_Own-child              -0.228532
relationship_Unmarried              -0.142857
relationship_Wife                    0.123264
sex_Female                          -0.215980
sex_Male                             0.215980
Name: income, dtype: float64
```

In [472]:
```
x_train, x_test, y_train, y_test = train_test_split(df_filtered.drop('income',  axis=1, inplace=False), df_filtered['incom
```

SVM

In [475]:
```
# SVM
svc = svm.SVC(gamma='auto')
svc.fit(x_test, y_test)
y_pred = svc.predict(x_test)

print("accuracy: ", accuracy_score(y_test, y_pred))
print("precision score: ", precision_score(y_test, y_pred))
print("recall: ", recall_score(y_test, y_pred))
print("f1: ", f1_score(y_test, y_pred))
```

```
precision  0.8947368421052632
recall  0.9444444444444444
f1_score  0.9189189189189189
TP  17
TN  16
FP  2
FN  1
```

Neural Network

```
# Build model
nn = NeuralNetwork(epochs=5, l1_sizes=[input_size,12], h_l=[sigmoid], nn_out=1000)
nn.fit(X_train, y_train)
y_pred = nn.predict(X_test)
# Evaluasi
print("Akurasi:", accuracy_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred))
print("Recall:", recall_score(y_test, y_pred))
print("F1 :", f1_score(y_test, y_pred))
```

```
Akurasi: 0.9473684210526315
Precision: 0.9473684210526315
F1 : 0.9473684210526315
```

Penutup

Yang membedakan performa model lebih baik adalah Neural Network 1000. Neural Network membutuhkan input data yang banyak dan dilakukannya dengan GPU. Semakin banyak data yang dimasukkan ke dalam network, maka akan semakin baik pembelajaran yang dilakukan oleh model. Sedangkan dengan metode prediksi manual dan model machine learning yang lain tidak cukup banyak, oleh karena itu lebih cocok digunakan.