UNTAR
Universitas Tarumanagara

Terakreditasi BAN PT A (Unggul)

UNTAR untuk INDONESIA

Analisis Data Googleplaystore

Big Data Praktikum

Studi kasus analisis data Googleplaystore

- Menghapus data duplikat
- Membersihkan data
- Menangani data yang hilang (missing data)

T1) Memanggil data Google Play Store menggunakan Pandas

```
import pandas as pd
import numpy as np


file = "googleplaystore.csv"
df = pd.read_csv(file)
```

T2: Membersihkan dan menganalisis kumpulan data

T2i) Cetak 10 baris pertama dari dataframe

```
df.head(10)
# df[0:10]
# df.iloc[0:10]
```

T2ii) Cetak 3 baris terakhir dari dataframe

```
# df.tail(3)
# df[-3:]
df.iloc[-3:]
```

T3) Dapatkan informasi umum tentang kumpulan data

```
df.info()
```

T4) Jelajahi kumpulan data dan coba pahami arti setiap kolom. Untuk setiap kolom, tuliskan makna dan tipe datanya

- App: Nama aplikasi (Text)
- Category: Kategori tempat aplikasi tersebut berada (Categorical - nominal)
- Rating: Peringkat pengguna secara keseluruhan (Numerical - continuous)
- Reviews: Jumlah ulasan pengguna (Numerical - continuous)
- Size: Ukuran aplikasi dalam MB (Numerical - continuous)
- Installs: Jumlah pemasangan pengguna (Numerical - discrete)
- Type: Gratis/Berbayar (Categorical - nominal)
- Price: Harga aplikasi (Numerical - continuous)
- Content: Peringkat Kelompok usia yang dapat menggunakan aplikasi (Categorical - ordinal)
- Genres: Genre yang dimiliki aplikasi tersebut (Categorical - nominal)
- Last Updated: Kapan terakhir kali diperbarui (Text; or integer jika Anda mengubah tanggal menjadi waktu Unix)
- Current Ver: Versi saat ini di Play Store (Text)
- Android Ver: Minimal diperlukan versi Android(Text)

T5) Dapatkan ukuran dataframe

```
df.shape
```

T6) Dapatkan nama kolom dari dataframe

```
df.columns
```

T7) Apakah menurut Anda kolom (variabel) harus lebih mudah dibaca? Jika ya, ganti namanya (misal, huruf besar yang sesuai, hilangkan garis bawah, kata lengkap)

```
df.columns
```

```
df.rename(index=str, columns={"Content
Rating":"ContentRating", "Last
Updated":"LastUpdated", "Current
Ver":"CurrentVersion", "Android
Ver":"AndroidVersion"}, inplace=True)
```

```
df.columns
```

Sebelum	Sesudah
App	App
Category	Category
Rating	Rating
Reviews	Reviews
Size	Size
Installs	Installs
Type	Type
Price	Price
Content Rating	ContentRating
Genres	Genres
Last Updated	LastUpdated
Current Ver	CurrentVersion
Android Ver	AndroidVersion

T8) Berapa banyak baris yang memiliki duplikat (data yang sama lebih dari satu)?

```
duplicatesNum = df.duplicated().sum()
print("There are %d duplicate records"%
(duplicatesNum))
```

T9i) Berapa banyak baris yang terduplikasi merujuk pada 'blood pressure'?

```
duplicates = df[(df.duplicated())]
duplicatedWithBloodPressure =
duplicates[duplicates['App'].str.contains('Blood pressure',
case=False)]


print("%d duplicated records have 'blood pressure' in their name" %
(duplicatedWithBloodPressure.shape[0]))


print("%d duplicated records have 'blood pressure' in their name" %
(len(duplicatedWithBloodPressure)))
```

T9ii) Cetak records (baris) pada T9i

```
# Mencetak baris yang berisi duplikasi yang berisi 'blood pressure'
duplicatedWithBloodPressure
```

T10) Mari kita lihat lebih dekat kumpulan datanya. Apakah ada lebih banyak duplikat?

Hint: Periksa fungsi duplikat Pandas untuk memahami cara kerja fungsi tersebut. Apakah Anda perlu menentukan informasi apa yang perlu dipertimbangkan saat memeriksa duplikat?

 # Dua record dikatakan sama jika berisi informasi untuk aplikasi yang sama

 # Jadi, kita perlu membatasi pengecekan pada kolom nama aplikasi

```
duplicatedAppsSum = df.duplicated(subset='App').sum()


print("In fact, there are %d duplicate records"%
(duplicatedAppsSum))
```

T11) Temukan aplikasi dengan jumlah rekaman terduplikasi terbanyak dan jumlah rekaman terduplikasi untuk aplikasi ini

Hint: Bagaimana Anda bisa menghitung records pada dataframe?

```
duplicatedApps = df[df.duplicated(subset='App')]

duplicatedAppsSummary =  duplicatedApps['App'].value_counts()

print("The app %s has %d duplicated records" %
(duplicatedAppsSummary.index[0], duplicatedAppsSummary[0]))
```

#Solution 2 menggunakan fungsi Python

```
duplicatedApps = df[df.duplicated(subset='App')]
dupsPerApp = []

duplicatedAppsNames = duplicatedApps['App'].unique()
for u in duplicatedAppsNames:
  dupsPerApp.append(len(duplicatedApps[duplicatedApps['App']==u]))

dupsPerAppArray = np.array(dupsPerApp)

maxDupRecords     = np.max(dupsPerAppArray)

maxDupRecordsApp  = duplicatedAppsNames[dupsPerAppArray.argmax()]

print("The app %s has %d duplicated records" % (maxDupRecordsApp,
maxDupRecords))
```

T12) Selidiki record untuk aplikasi yang Anda temukan di tugas T10. Apa yang bisa Anda amati? Bisakah Anda mengidentifikasi alasan mengapa ada duplikat record?

```
duplicatedApps[duplicatedApps['App']==maxDupRecordsApp]
```

Recored duplikat ada karena::

dua atau lebih record identik (untuk semua variabel); misalnya, baris 2976, 3007, 3015, 3020

```
d1 = df[2976:2977].append(df[3007:3008])
d2 = df[3015:3016].append(df[3020:3021])
d1.append(d2)
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	ContentRating	Genres	LastUpdated	CurrentVersion	AndroidVersion
2976	CBS Sports App - Scores, News, Stats & Watch Live	SPORTS	4.3	91031	Varies with device	5,000,000+	Free	0	Everyone	Sports	August 4, 2018	Varies with device	5.0 and up
3007	CBS Sports App - Scores, News, Stats & Watch Live	SPORTS	4.3	91031	Varies with device	5,000,000+	Free	0	Everyone	Sports	August 4, 2018	Varies with device	5.0 and up
3015	CBS Sports App - Scores, News, Stats & Watch Live	SPORTS	4.3	91031	Varies with device	5,000,000+	Free	0	Everyone	Sports	August 4, 2018	Varies with device	5.0 and up
3020	CBS Sports App - Scores, News, Stats & Watch Live	SPORTS	4.3	91031	Varies with device	5,000,000+	Free	0	Everyone	Sports	August 4, 2018	Varies with device	5.0 and up

Dua atau lebih record berhubungan dengan aplikasi yang sama. Namun, jumlah ulasannya mungkin berbeda; misalnya baris 3020 dan 3056. Hal ini menunjukkan bahwa pengumpulan informasi terjadi pada titik waktu yang berbeda sehingga menghasilkan kumpulan data dengan informasi yang tidak konsisten

```
df[3020:3021].append(df[3056:3057])
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	ContentRating	Genres	LastUpdated	CurrentVersion	AndroidVersion
3020	CBS Sports App - Scores, News, Stats & Watch Live	SPORTS	4.3	91031	Varies with device	5,000,000+	Free	0	Everyone	Sports	August 4, 2018	Varies with device	5.0 and up
3056	CBS Sports App - Scores, News, Stats & Watch Live	SPORTS	4.3	91033	Varies with device	5,000,000+	Free	0	Everyone	Sports	August 4, 2018	Varies with device	5.0 and up

T13) Dengan asumsi bahwa data telah diambil secara berurutan, hapus semua duplikat dengan menentukan record mana yang akan disimpan

Hint: Anda mungkin ingin memeriksa dokumentasi fungsi drop_duplikat

#Karena data telah di-scrap secara berurutan, artinya duplikat terbaru adalah yang terbaru. Oleh karena itu, kita perlu menyimpan duplikat terakhir

```
dfClean = df.drop_duplicates(subset='App', keep='last')
dfClean.shape
```

T14i) Periksa jenis variabel yang menunjukkan kapan terakhir kali aplikasi diperbarui dan ubah ke dalam format yang sesuai

```
#dfClean.loc[:, 'LastUpdated']
```

```
dfClean.loc[:, 'LastUpdated'] =
pd.to_datetime(dfClean['LastUpdated'], format='%B %d, %Y')
```

```
#dfClean.loc[:, 'LastUpdated']
```

T14ii) Aplikasi manakah yang terakhir diperbarui dan kapan hal ini terjadi?

```
sortedLastUpdated = dfClean['LastUpdated'].sort_values()


print("%s is the least recently updated application and was updated
on %s"% (dfClean.loc[sortedLastUpdated.index[0]]['App'],
str(sortedLastUpdated[0])))
```

T15) Berapa banyak nilai unik yang ada pada variabel Price? Cetak lima harga unik pertama.

```
uniquePricesNum = dfClean['Price'].nunique()
uniquePrices = dfClean['Price'].unique()


print("There are %d unique price values. The first five are %s" %
(uniquePricesNum, uniquePrices[:5]))


print("There are %d unique price values. The first five are %s" %
(len(uniquePrices), uniquePrices[:5]))
```

T16i) Tulis fungsi moneyWithoutCurrencySymbol yang mengambil string dan mengembalikan string sebagai float tanpa simbol mata uang (jika ada)

Ini adalah fungsi umum yang mengambil string dengan simbol $ dan mengembalikan float setelah menghapus simbol $ dan mengubah sisa string menjadi float

```
def moneyWithoutCurrencySymbol(v):
  if type(v) is not float:
    return float(v.replace("$", ""))
  else:
    return v
```

T16ii) Lakukan tugas pemrosesan yang diperlukan untuk mengubah variabel Price menjadi float

```
dfClean[dfClean['App']=="Moco+ - Chat, Meet People"]
```

```
# Solution 1
dfClean.loc[:, 'Price'] =
dfClean['Price'].apply(moneyWithoutCurrencySymbol)
```

```
#dfClean[dfClean['App']=="Moco+ - Chat, Meet People"]
```

T16ii) Lakukan tugas pemrosesan yang diperlukan untuk mengubah variabel Price menjadi float

#Solution 2

```
dfClean["Price"] = pd.to_numeric(dfClean["Price"].str.strip("$"))
```

T17i) Berapa banyak data yang hilang untuk setiap variabel? Apakah kumpulan data ini berantakan atau bersih?

```
dfClean.isna().sum()
```

```
dfClean.isnull().sum()
```

T17ii) Hitung kolom kategorikal dengan nilai mode

Ganti data yang hilang (NaN) di kolom kategori " Type", " Content Rating ", " Android Version " dan " Current Version" dengan mode setiap kolom

```
#dfClean[dfClean['Type'].isna()]
```

```
dfClean['Type'] = dfClean['Type'].fillna(dfClean['Type'].mode()[0])
```

```
#dfClean[dfClean['App'] == 'Command & Conquer: Rivals']
```

```
dfClean['AndroidVersion'] =
dfClean['AndroidVersion'].fillna(dfClean['AndroidVersion'].mode()[0])


dfClean['CurrentVersion'] =
dfClean['CurrentVersion'].fillna(dfClean['CurrentVersion'].mode()[0])
```

#Solution 2

```
dfClean[dfClean['AndroidVersion'].isna()]
```

```
dfClean['AndroidVersion'].mode()
```

```
dfClean.loc[dfClean['AndroidVersion'].isna(), 'AndroidVersion'] =
dfClean['AndroidVersion'].mode()[0]
```

```
#dfClean[dfClean['App'] == '[substratum] Vacuum: P']
#dfClean[dfClean['App'] == 'Pi Dark [substratum]']
```

#Solution2

```
dfClean.loc[dfClean['Type'].isna(), 'Type'] = dfClean['Type'].mode()[0]


dfClean.loc[dfClean['AndroidVersion'].isna(), 'AndroidVersion'] =
dfClean['AndroidVersion'].mode()[0]


dfClean.loc[dfClean['CurrentVersion'].isna(), 'CurrentVersion'] =
dfClean['CurrentVersion'].mode()[0]
```

T17iii) Berapa nilai yang pantas untuk diperhitungkan pada variabel lain? Terapkan imputasinya.

```
# Kita berasumsi bahwa aplikasi dengan peringkat NaN adalah aplikasi baru dengan hanya sedikit ulasan atau aplikasi yang tidak memiliki ulasan apa pun, mungkin karena tidak dirancang dan dibangun dengan baik. Jadi strategi terbaiknya adalah dengan memperhitungkan record Rating NaN tersebut dengan nilai minimum, yaitu 0
#dfClean[dfClean['Rating'].isna()]
#Solution1
dfClean['Rating'] = dfClean['Rating'].fillna(0)
#dfClean[dfClean['Rating']==0]
#Solution2
#dfClean.loc[dfClean['Rating'].isna(), 'Rating'] = 0
```

T18i) Manakah nilai yang mungkin untuk variabel Installs?

```
# Mari kita periksa nilai unik dari Installs
dfClean['Installs'].unique()
```

T18ii) Berdasarkan hasil tugas T18i, apakah Anda melihat ada nilai ganjil? Apakah menurut Anda pembersihan diperlukan?

#Tampaknya ada entri dengan "0+" dan "0". Mari kita homogenkan dan ubah "0" menjadi "0+"

```
dfClean.loc[dfClean["Installs"]=="0","Installs"] = "0+"
```

T18iii) Jika Anda melakukan pembersihan apa pun di T18ii, periksa apakah pembersihan telah dilakukan

```
dfClean[(dfClean['Installs']=="0")]
```

T19i) Lakukan tugas pemrosesan yang diperlukan untuk mengubah variabel Size menjadi float menggunakan metrik umum (yaitu, yang memungkinkan untuk membandingkan record antara satu sama lain menggunakan nilai ukuran). Tetapkan catatan yang nilainya " Varies with device " ke 1

#Karena ukuran beberapa aplikasi dalam Megabyte sedangkan aplikasi lainnya dalam Kilobyte,

#kita perlu mengubahnya menjadi metrik umum. Di sini, kita mengubah semuanya menjadi Megabyte

```
def sizeToFloat (v):
  if "M" in v:
    return float(v.strip("M"))
  elif v[-1]=="k":
    return float(v.strip("k"))/1024
  else:
    return 1.0

dfClean['Size'] = dfClean['Size'].apply(sizeToFloat)
```

T19ii) Aplikasi manakah yang terkecil dan berapa ukurannya?

```
sortedSize = dfClean['Size'].sort_values()


# sortedSize[-1]→ the last index
# sortedSize[0] → the first index


print("%s is the smallest application and its size is %.4fM" %
(dfClean.loc[sortedSize.index[0]]['App'], sortedSize[0]))
```

T19iii) Kategori aplikasi terbesar manakah yang paling umum dan aplikasi manakah yang termasuk dalam Kategori ini?

```
largestApps = dfClean.loc[sortedSize[sortedSize==sortedSize[-1]].index]


largestAppsCategory = largestApps['Category'].mode()[0]


largestAppsCategoryNames =
largestApps[largestApps['Category']==largestAppsCategory]['App']


print("The most common category of the largest applications is  ",
largestAppsCategory)


print("The applications that belong to this category are\n",
largestAppsCategoryNames.values)
```

T20) Apakah ada kesalahan logis dalam variabel Type dan Price? Apakah ada aplikasi yang tipenya Gratis dan harus berbayar? Apakah ada aplikasi dengan tipe Berbayar yang bisa Anda dapatkan secara gratis? Periksa integritas kumpulan data menggunakan kondisi ini

```
free = dfClean[(dfClean['Price']>0) & (dfClean['Type']=="Free") ]
paid = dfClean[(dfClean['Price']<=0) & (dfClean['Type']=="Paid") ]


print ("The are %d Free applications for which you have to pay" %
(len(free)))
print ("The are %d Paid applications which are given for free" %
(len(paid)))
```

T21) Peninjau suatu aplikasi mungkin menilaiatau tidak menilai ulasan tersebut. Periksa konsistensi dataset mengenai hal ini

#Di sini kita perlu memeriksa bahwa tidak ada aplikasi tanpa ulasan apa pun yang memiliki peringkat di atas 0

```
dfClean[(dfClean['Reviews']==0) & (dfClean['Rating']>0)]
```

T22) Dapatkan beberapa statistik deskriptif untuk kumpulan data

```
dfClean.describe()
```

T23) Buat kolom baru yang menunjukkan perbedaan antara kolom variabel Last Updated dan tanggal sekarang.

```
from datetime import datetime,date


dfClean["NotUpdatedFor"]  =
pd.to_datetime(datetime.today().strftime("%m-%d-%Y")) -
dfClean["LastUpdated"]


dfClean.head()
```