Laporan Tugas 1

Penambangan Data dan Intelegensia Bisnis

Zaki Raihan

1606878505

Penambangan Data dan Intelegensia Bisnis - B

Program Studi Sistem Informasi

Fakultas Ilmu Komputer

Universitas Indonesia

Depok

2018

1. Pendahuluan

Laporan ini terkait dengan langkah-langkah yang saya lakukan ketika bertemu dengan suatu data. Seperti yang kita ketahui bahawasannya data merupakan salah satu komoditi yang cukup penting di zaman sekarang ini. Banyak sekali perusahaan yang menggunakan data sebagai acuan untuk menentukan arah jalannya bisnis yang mereka miliki. Namun, sering kali data yang dimiliki atau didapatkan tidak dalam kondisi yang baik. Terkadang data yang kita dapatkan masih memiliki nilai yang kosong, terdapat banyak bias, kesalahan penulisan, sampai adanya *outlier* yang dapat mengganggu kualitas data yang akan di proses. Sesuai dengan yang diajarkan di kelas, maka ketika kita mendapatkan suatu data maka kita diharusnkan untuk mempersiapkan data tersebut sebelum kita olah sesuai dengan kebutuhan kita. Dengan mempersiapkan data, maka kita dapat meningkatkan kualitas data sehingga data akan lebih representatif dengan keadaan di dunia nyata dan memiliki sedikit bias.

2. Deskripsi Data

Data yang digunakan adalah data *home equity load*, yaitu sebuah pinjaman yang mana peminjam menggunakan *equity* dari rumahnya sebagai agunan dasarnya.
Data ini memiliki beberapa variabel didalamnya yaitu sebanyak 13 variabel:
. BAD: 1 = peminjam terkena gagal pinjaman atau mengalami tunggakan serius; 0 = peminjam membayar hutang
. LOAN: banyaknya pinjaman yang ingin diminta
. MORTDUE: Amount due on existing mortgage
. VALUE: nilai dari properti saat ini
. REASON: alasan melakukan peminjaman, DebtCon = debt consolidation; HomeImp = home improvement
. JOB: jenis pekerjaan
. YOJ: lama sudah bekerja di pekerjaan saat ini
. DEROG: banyaknya riwayat laporan buruk
. DELINQ: banyaknya kredit macet
. CLAGE: usia batas kredit tertua dalam beberapa bulan
. NINQ: jumlah permintaan kredit baru-baru ini
. CLNO: jumlah batas kredit
. DEBTINC: ratio debt-to-income

3. Data Preparation

Untuk melakukan persiapan data, saya menggunakan bahasa pemrograman python dengan bantuan beberapa modul seperti numpy dan pandas. Pandas sendiri sudah cukup sering digunakan di dunia pengolahan data. Dengan menggunakan modul panda ini, saya berusaha untuk mempersiapkan data dengan tahapan sebagai berikut:

- Untuk kolom yang berlebihan dari kolom standar yang dimiliki data maka akan saya hapus (drop column)
- Untuk bersifat katagorikal maka akan saya ubah menjadi numerik. Hal ini untuk memudahkan apabila ingin melihat korelasi antar variabel
- Jika data memiliki outlier, maka outlier tersebut akan saya buang (diubah menjadi kosong nilainya - NaN)
- Untuk data yang terdapat kesalahan penulisan yang masih bisa dilihat tulisan aslinya, maka nilainya saya sesuaikan dengan nilai aslinya

- Untuk data yang memiliki value kosong, maka akan memungkinkan untuk saya ganti dengan beberapa aturan:
    - Untuk diawal pemrosesan data setiap row yang memiliki nilai kosong akan saya hapus
    - Jika setelah data saya proses dan masih terdapat nilai kosong (akibat penghapusan outlier ataupun ada data yang salah tulis) maka akan saya berlakukan hal tersebut:
        - Untuk data bersifat numerik, maka nilai yang kosong akan saya gantikan dengan nilai mean dari kolom yang bersangkutan.
        - Untuk data yang (awalnya) bersifat katagorikal, maka saya menentukan nilai default untuk kolom yang bersangkutan, yaitu:
            - Untuk kolom "JOB" maka nilai defaultnya adalah "Other" (atau setelah di mapping ke numerik menjadi 0)
            - Untuk kolom "REASON" maka nilai defaultnya adalah "DebtCon" (atau setelah di mapping ke numerik menjadi 1)
- Terakhir, data saya lakukan normalisasi dengan metode Min-Max sehingga range dari data hanya berkisar 0 s.d. 1

4. Hasil/Temuan

   Saya menemukan bahwa data masih memiliki banyak kesalahan dan bias, berikut adalah hal-hal yang saya temukan baik sebelum ataupun sesudah data saya siapkan:
    a. Ada data yang memiliki kolom lebih dari 13, sehingga kolom-kolom yang melebihi dari 13 akan saya hapus.
    b. Pada kolom DebtInc banyak baris data yang kosong, yaitu sekitar 2000-an baris, untuk itu di awal 2000-an baris tersebut saya hapus (drop)
    c. Adanya kesalahan penulisan pada variabel "JOB", dimana ada data yang tertulis "Othe" yang seharusnya adalah "Other" untuk itu saya ubah menjadi Other (secara numerik menjadi angka 0)
    d. Adanya kesalahan penulisan pada variabel "REASON" dimana data yang tertulis adalah "Debtcons" yang seharusnya adalah "Debtcon" untuk itu saya ubah menjadi Debtcon (secara numerik menjadi angka 1)
    e. Terdapat cukup banyak outlier pada data (pada kolom tertentu) sehingga saya mengubah outlier tersebut menjadi nilai mean dari kolom data yang bersangkutan.
    f. Nilai dari data pada tiap kolom cukup bervariasi mulai dari satuan, puluhan, bahkan ada variabel yang mengandung data dengan nilai ratusan ribu. Untuk itu saya melakukan normalisasi dengan metode minmax sehingga rentang nilai di setiap variabel data sama, yaitu 0 s.d. 1.
    g. Terdapat korelasi yang cukup baik antar data dimana:
        i. Data variabel "BAD" memiliki korelasi yang cukup baik secara negatif dengan variabel "LOAN", "VALUE" dan "CLAGE" serta korelasi yang cukup baik secara positif dengan variabel "DEROG", "DELINQ", dan "DEBTINC".
        ii. Data variabel "VALUE" memiliki korelasi yang cukup baik dengan beberapa variabel lain seperti "LOAN", "MORTDUE", "JOB", dan "CLNO"

Untuk korelasi antar variabel secara keseluruhan bisa dilihat dari gambar dibawah ini:

	BAD	LOAN	MORTDUE	VALUE	REASON	JOB	YOJ	DEROG	DELINQ	CLAGE	NINQ	CLNO	DEBTINC
BAD	1.0	-0.082	-0.043	-0.078	-0.0039	-0.055	-0.063	0.17	0.28	-0.13	0.044	-0.031	0.13
LOAN	-0.082	1.0	0.15	0.33	0.33	0.1	0.099	-0.0072	-0.091	0.073	0.0003	0.11	0.15
MORTDUE	-0.043	0.15	1.0	0.84	0.069	0.24	-0.12	-0.041	-0.058	0.044	-0.026	0.32	0.24
VALUE	-0.078	0.33	0.84	1.0	0.051	0.32	-0.02	-0.062	-0.066	0.17	-0.021	0.29	0.17
REASON	-0.0039	0.33	0.069	0.051	1.0	-0.036	-0.12	-0.0048	-0.057	-0.032	0.057	0.086	0.069
JOB	-0.055	0.1	0.24	0.32	-0.036	1.0	-0.022	-0.049	-0.022	0.15	-0.031	0.18	-0.11
YOJ	-0.063	0.099	-0.12	-0.02	-0.12	-0.022	1.0	-0.037	0.056	0.27	-0.012	0.034	-0.035
DEROG	0.17	-0.0072	-0.041	-0.062	-0.0048	-0.049	-0.037	1.0	0.11	-0.07	0.58	-0.011	0.0083
DELINQ	0.28	-0.091	-0.058	-0.066	-0.057	-0.022	0.056	0.11	1.0	0.019	-0.017	0.12	0.0026
CLAGE	-0.13	0.073	0.044	0.17	-0.032	0.15	0.27	-0.07	0.019	1.0	-0.085	0.18	-0.076
NINQ	0.044	0.0003	-0.026	-0.021	0.057	-0.031	-0.012	0.58	-0.017	-0.085	1.0	-0.034	0.069
CLNO	-0.031	0.11	0.32	0.29	0.086	0.18	0.034	-0.011	0.12	0.18	-0.034	1.0	0.13
DEBTINC	0.13	0.15	0.24	0.17	0.069	-0.11	-0.035	0.0083	0.0026	-0.076	0.069	0.13	1.0

h. Untuk source code metode pengolahan data dapat dilihat dalam lampiran

5. Kesimpulan

Data Preparation merupakan proses yang cukup penting dalam pengolahan suatu data. Hal tersebut dikarenakan sering kali data yang kita peroleh terdapat banyak kekurangan seperti adanya data yang hilang, kesalahan penulisan, sampai adanya kolom yang berlebih. Selain itu Data preparation juga penting untuk mengurangi bias yang dimiliki oleh data yang kita miliki. Sering kali ada saja data yang nilainya jauh berbeda dari data yang lain (Outlier) sehingga dapat mengganggu kita pada saat melakukan pengelolaan data. Jika Data Preparation dilakukan dengan baik, maka hasil yang kita dapatkan pada saat pengelolaan data nantinya juga akan baik pula.

Untuk dokumentasi source code dari data preparation yang saya lakukan dapat dilihat pada:

https://github.com/zakiraihan/PDIB-ZakiRaihan-1606878505/tree/master/Tugas%201

In [1]:

```
%matplotlib inline
import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import random
import math

# Read hmeq.csv file
hmeq = pd.read_csv("hmeq.csv")

# Removing any unnecessary column, because the data were just 13 column,
# so i assume that column more than 13 were unnecessary
hmeq_prepared = hmeq.drop(columns=['Unnamed: 13', 'Unnamed: 14', 'Unnamed: 15', 'Unnamed: 1

# Drop Row that has NaN value
hmeq_prepared = hmeq_prepared.dropna()

# Seeing the first 5 row of data
hmeq_prepared.head()
```

Out[1]:

	BAD	LOAN	MORTDUE	VALUE	REASON	JOB	YOJ	DEROG	DELINQ	CLAGE	I
5	1	1700	30548.0	40320.0	HomeImp	Other	9	0.0	0.0	101.466002	
7	1	1800	28502.0	43034.0	HomeImp	Other	11	0.0	0.0	88.766030	
18	1	2300	28192.0	40150.0	HomeImp	Mgr	Other	4.5	0.0	0.000000	
19	0	2300	102370.0	120953.0	HomeImp	Office	2	0.0	0.0	90.992533	
25	1	2400	34863.0	47471.0	HomeImp	Mgr	12	0.0	0.0	70.491080	

In [2]:

```
# Check Unique Value for some catagorical column
for column in ["BAD", "REASON", "JOB"]:
    print (hmeq_prepared[column].unique())
```

```
[1 0 3]
['HomeImp' 'DebtCon' 'DebtCons']
['Other' 'Mgr' 'Office' 'ProfExe' 'Sales' 'Self' 'Othe']
```

In [3]:

```
# Preparing data, changing catagorycal data into numeric data

# Mapping catagorical into numeric
hmeq_prepared['JOB'] = hmeq_prepared['JOB'].map({'Other': 0, 'Office':1, 'Sales':2, 'Mgr':3
# Filling any missing data in "JOB" variable using 0 ("i assume it will be other")
hmeq_prepared['JOB'] = hmeq_prepared['JOB'].fillna(0)

# Mapping catagorical into numeric
hmeq_prepared['REASON'] = hmeq_prepared['REASON'].map({'HomeImp': 0, 'DebtCon': 1})
# Filling any missing data in "REASON" variable using 0 ("i assume it will be HomeImp")
hmeq_prepared['REASON'] = hmeq_prepared['REASON'].fillna(1)

# Clean typo in "BAD" variable (there is data that typed 10 instead of 1)
hmeq_prepared['BAD'] = hmeq_prepared['BAD'].replace(10, 1)
# Clean typo in "BAD" variable (there is data that typed 3 instead of 1,
# i assume they type 3 because the loan were bad)
hmeq_prepared['BAD'] = hmeq_prepared['BAD'].replace(3, 1)

# Seeing first 5 data
hmeq_prepared.head()
```

Out[3]:

	BAD	LOAN	MORTDUE	VALUE	REASON	JOB	YOJ	DEROG	DELINQ	CLAGE	NI
5	1	1700	30548.0	40320.0	0.0	0.0	9	0.0	0.0	101.466002	
7	1	1800	28502.0	43034.0	0.0	0.0	11	0.0	0.0	88.766030	
18	1	2300	28192.0	40150.0	0.0	3.0	Other	4.5	0.0	0.000000	5
19	0	2300	102370.0	120953.0	0.0	1.0	2	0.0	0.0	90.992533	
25	1	2400	34863.0	47471.0	0.0	3.0	12	0.0	0.0	70.491080	

In [4]:

```
# Fuction for cleaning outlier from data, it will fill the outlier using median of data
def cleanOutlier(dataSet, listColumn):
    for column in listColumn:
        Q1 = dataSet[column].quantile(0.25)
        Q3 = dataSet[column].quantile(0.75)
        IQR = Q3-Q1
        outlier = ((dataSet[column] < (Q1 - 1.5 * IQR)) | (dataSet[column] > (Q3 + 1.5 * IQ
        dataSet[outlier] = np.nan
        dataSet[column] = dataSet[column].fillna(dataSet.mean())

# Normalize the data using MinMaxScaler, so the value of data will around 0-1
def normalizeDataset(dataset, listColumn=[]):
    if (len(listColumn) > 0):
        for column in listColumn:
            max_value = dataset[column].max()
            min_value = dataset[column].min()
            dataset[column] = (dataset[column] - min_value) / (max_value - min_value)
    else:
        dataset = (dataset - dataset.mean())/dataset.std()
        dataset = dataset.round(4)
```

In [5]:

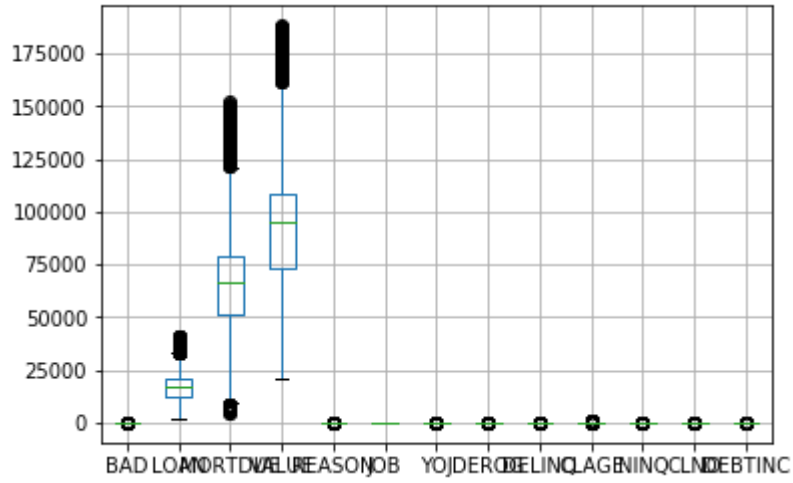
```
# Panda read the YOJ variable as "object" instead of number, so i changing it into number t
hmeq_prepared['YOJ'] = pd.to_numeric(hmeq_prepared['YOJ'], errors='coerce')
# Fill NaN value of YOJ using mean of YOJ
hmeq_prepared['YOJ'] = hmeq_prepared['YOJ'].fillna(hmeq_prepared['YOJ'].mean())
# Try to clean the data from outlier in any variable
cleanOutlier(hmeq_prepared, ['LOAN','MORTDUE','VALUE','CLNO','DEBTINC'])

# Filling any missing data in dataset using their represntative column
hmeq_prepared = hmeq_prepared.fillna(hmeq_prepared.mean())

# Seeing the boxplot result, to check if the outlier still exist
hmeq_prepared.boxplot()
```

Out[5]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1fba3fb38d0>
```



In [6]:

```
# Try to normazise some variable in the dataset
normalizeDataset(hmeq_prepared, ['LOAN', 'JOB', 'REASON', 'MORTDUE','VALUE','YOJ','DEROG','
```

In [7]:

```
hmeq_prepared.corr(method='pearson').style.format("{:.2}").background_gradient(cmap=plt.get
```

Out[7]:

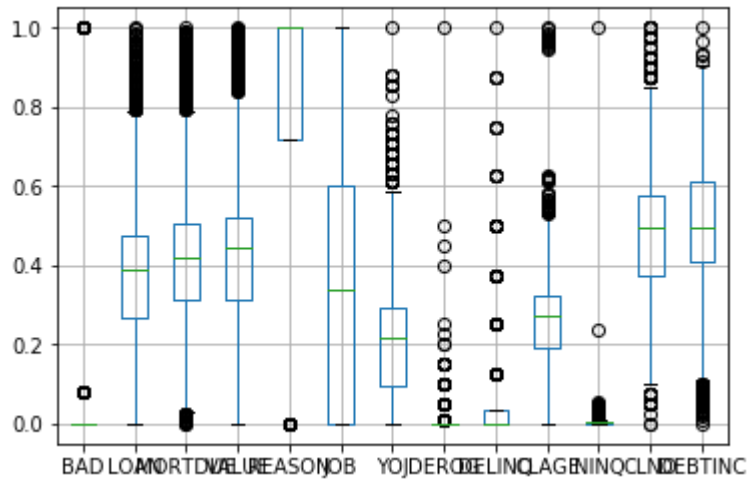
	BAD	LOAN	MORTDUE	VALUE	REASON	JOB	YOJ	DEROG	DELINQ	CL
BAD	1.0	-0.082	-0.043	-0.078	-0.0039	-0.055	-0.063	0.17	0.28	
LOAN	-0.082	1.0	0.15	0.33	0.33	0.1	0.099	-0.0072	-0.091	(
MORTDUE	-0.043	0.15	1.0	0.84	0.069	0.24	-0.12	-0.041	-0.058	(
VALUE	-0.078	0.33	0.84	1.0	0.051	0.32	-0.02	-0.062	-0.066	
REASON	-0.0039	0.33	0.069	0.051	1.0	-0.036	-0.12	-0.0048	-0.057	-(
JOB	-0.055	0.1	0.24	0.32	-0.036	1.0	-0.022	-0.049	-0.022	
YOJ	-0.063	0.099	-0.12	-0.02	-0.12	-0.022	1.0	-0.037	0.056	
DEROG	0.17	-0.0072	-0.041	-0.062	-0.0048	-0.049	-0.037	1.0	0.11	
DELINQ	0.28	-0.091	-0.058	-0.066	-0.057	-0.022	0.056	0.11	1.0	(
CLAGE	-0.13	0.073	0.044	0.17	-0.032	0.15	0.27	-0.07	0.019	
NINQ	0.044	0.0003	-0.026	-0.021	0.057	-0.031	-0.012	0.58	-0.017	-(
CLNO	-0.031	0.11	0.32	0.29	0.086	0.18	0.034	-0.011	0.12	
DEBTINC	0.13	0.15	0.24	0.17	0.069	-0.11	-0.035	0.0083	0.0026	-(

In [8]:

```
hmeq_prepared.boxplot()
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1fba55f3be0>
```



In [9]:

```
# Saving the prepared hmeq to new csv file called hmeq_prepared.csv
hmeq_prepared.to_csv('hmeq_prepared.csv', sep=',', encoding='utf-8',index=False)
```

In [10]:

```
hmeq_prepared.head()
```

Out[10]:

	BAD	LOAN	MORTDUE	VALUE	REASON	JOB	YOJ	DEROG	DELINQ	CLAGE
5	1.0	0.000000	0.173285	0.114819	0.0	0.0	0.219512	0.000	0.0	0.156162
7	1.0	0.002538	0.159366	0.131070	0.0	0.0	0.268293	0.000	0.0	0.136616
18	1.0	0.015228	0.157257	0.113802	0.0	0.6	0.222108	0.225	0.0	0.000000
19	0.0	0.015228	0.661886	0.597623	0.0	0.2	0.048780	0.000	0.0	0.140043
25	1.0	0.017766	0.202640	0.157637	0.0	0.6	0.292683	0.000	0.0	0.108490

In [ ]:

```

```

In [ ]:

```

```