Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Отчет

по дисциплине «Технология Машинного обучения»

Выполнил:
студент группы ИУ5-62
Миронов Святослав
подпись, дата

2019 г.

Задание:

1. Выбрать набор данных (датасет), содержащий категориальные признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.)
2. Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи:

- обработку пропусков в данных;
- кодирование категориальных признаков;
- масштабирование данных.

Lab3

June 3, 2019

```
In [1]: import numpy as np
        import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt
        %matplotlib inline
        sns.set(style="ticks")

In [2]: data = pd.read_csv('BlackFriday.csv', sep=",")
```

0.1 Description

The dataset here is a sample of the transactions made in a retail store. The store wants to know better the customer purchase behaviour against different products. Specifically, here the problem is a regression problem where we are trying to predict the dependent variable (the amount of purchase) with the help of the information contained in the other variables.

Classification problem can also be settled in this dataset since several variables are categorical, and some other approaches could be "Predicting the age of the consumer" or even "Predict the category of goods bought". This dataset is also particularly convenient for clustering and maybe find different clusters of consumers within it.

```
In [3]: data.shape

Out[3]: (537577, 12)

In [4]: data.dtypes

Out[4]: User_ID                       int64
        Product_ID                   object
        Gender                       object
        Age                          object
        Occupation                    int64
        City_Category                object
        Stay_In_Current_City_Years   object
        Marital_Status                int64
        Product_Category_1            int64
        Product_Category_2          float64
        Product_Category_3          float64
        Purchase                      int64
        dtype: object
```

```
In [5]: data.isnull().sum()
```

```
Out[5]: User_ID                          0
        Product_ID                       0
        Gender                           0
        Age                              0
        Occupation                       0
        City_Category                    0
        Stay_In_Current_City_Years       0
        Marital_Status                   0
        Product_Category_1               0
        Product_Category_2          166986
        Product_Category_3          373299
        Purchase                         0
        dtype: int64
```

```
In [6]: data.head()
```

```
Out[6]:    User_ID Product_ID Gender   Age  Occupation City_Category  \
        0  1000001  P00069042      F  0-17          10             A
        1  1000001  P00248942      F  0-17          10             A
        2  1000001  P00087842      F  0-17          10             A
        3  1000001  P00085442      F  0-17          10             A
        4  1000002  P00285442      M   55+          16             C

          Stay_In_Current_City_Years  Marital_Status  Product_Category_1  \
        0                          2               0                   3
        1                          2               0                   1
        2                          2               0                  12
        3                          2               0                  12
        4                         4+               0                   8

           Product_Category_2  Product_Category_3  Purchase
        0                 NaN                 NaN      8370
        1                 6.0                14.0     15200
        2                 NaN                 NaN      1422
        3                14.0                 NaN      1057
        4                 NaN                 NaN      7969
```

```
In [7]: #  ,
        data_new_1 = data.dropna(axis=0, how='any')
        (data.shape, data_new_1.shape)
```

```
Out[7]: ((537577, 12), (164278, 12))
```

0.2 Product_Category_2 Product_Category_3 , ,

```
In [8]: data = data.fillna(0)
        data.head()
```

Out[8]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	0.0	0.0	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	6.0	14.0	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	0.0	0.0	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	14.0	0.0	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	0.0	0.0	7969

```
In [9]: print (data['Gender'].unique())
        print (data['Age'].unique())
        print (data['City_Category'].unique())
        print (data['Stay_In_Current_City_Years'].unique())
```

```
['F' 'M']
['0-17' '55+' '26-35' '46-50' '51-55' '36-45' '18-25']
['A' 'C' 'B']
['2' '4+' '3' '1' '0']
```

0.3 ,

```
In [10]: from sklearn.preprocessing import LabelEncoder
```

```
In [11]: le = LabelEncoder()
         gender_int = le.fit_transform(data['Gender'])
         data['Gender_int']=gender_int
         del data['Gender']

         Age_int = le.fit_transform(data['Age'])
         data['Age_int']=Age_int
         del data['Age']

         City_Category_int = le.fit_transform(data['City_Category'])
         data['City_Category_int']=City_Category_int
         del data['City_Category']
```

```
Stay_In_Current_City_Years_int = le.fit_transform(data['Stay_In_Current_City_Years'])
data['Stay_In_Current_City_Years_int']=Stay_In_Current_City_Years_int
del data['Stay_In_Current_City_Years']

data.head()
```

Out[11]:

	User_ID	Product_ID	Occupation	Marital_Status	Product_Category_1	\
0	1000001	P00069042	10	0	3	
1	1000001	P00248942	10	0	1	
2	1000001	P00087842	10	0	12	
3	1000001	P00085442	10	0	12	
4	1000002	P00285442	16	0	8	

	Product_Category_2	Product_Category_3	Purchase	Gender_int	Age_int	\
0	0.0	0.0	8370	0	0	
1	6.0	14.0	15200	0	0	
2	0.0	0.0	1422	0	0	
3	14.0	0.0	1057	0	0	
4	0.0	0.0	7969	1	6	

	City_Category_int	Stay_In_Current_City_Years_int
0	0	2
1	0	2
2	0	2
3	0	2
4	2	4

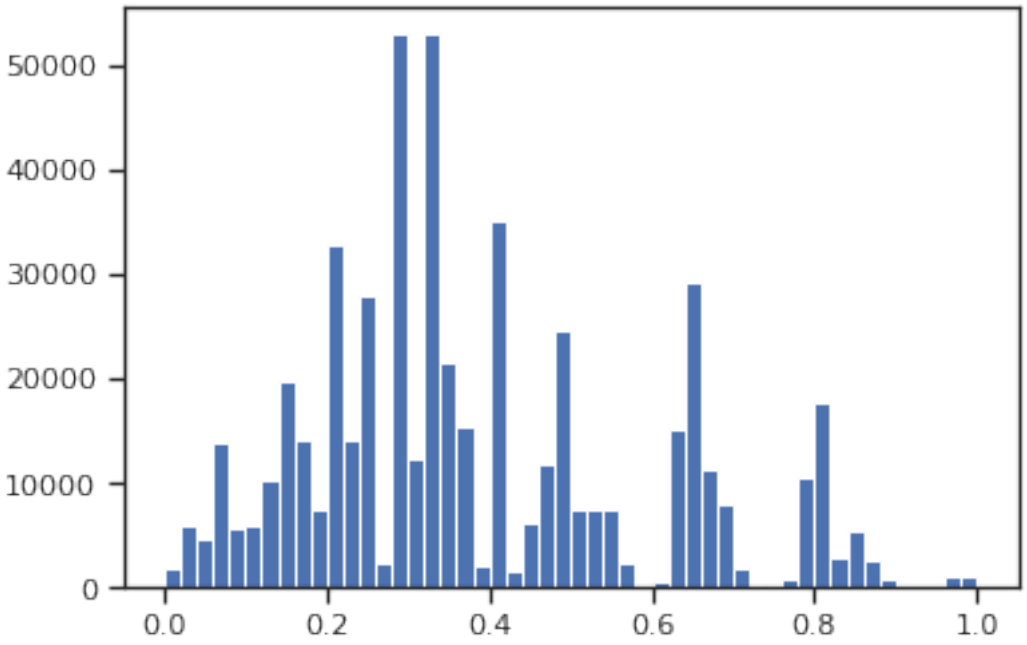
In [15]:
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

In [23]:
```
sc = MinMaxScaler()
sc_data = sc.fit_transform(data[['Purchase']])

plt.hist(sc_data, 50)
plt.show()
```

```
/srv/conda/lib/python3.6/site-packages/sklearn/preprocessing/data.py:323: DataConversionWarning
  return self.partial_fit(X, y)
```

Out[23]:

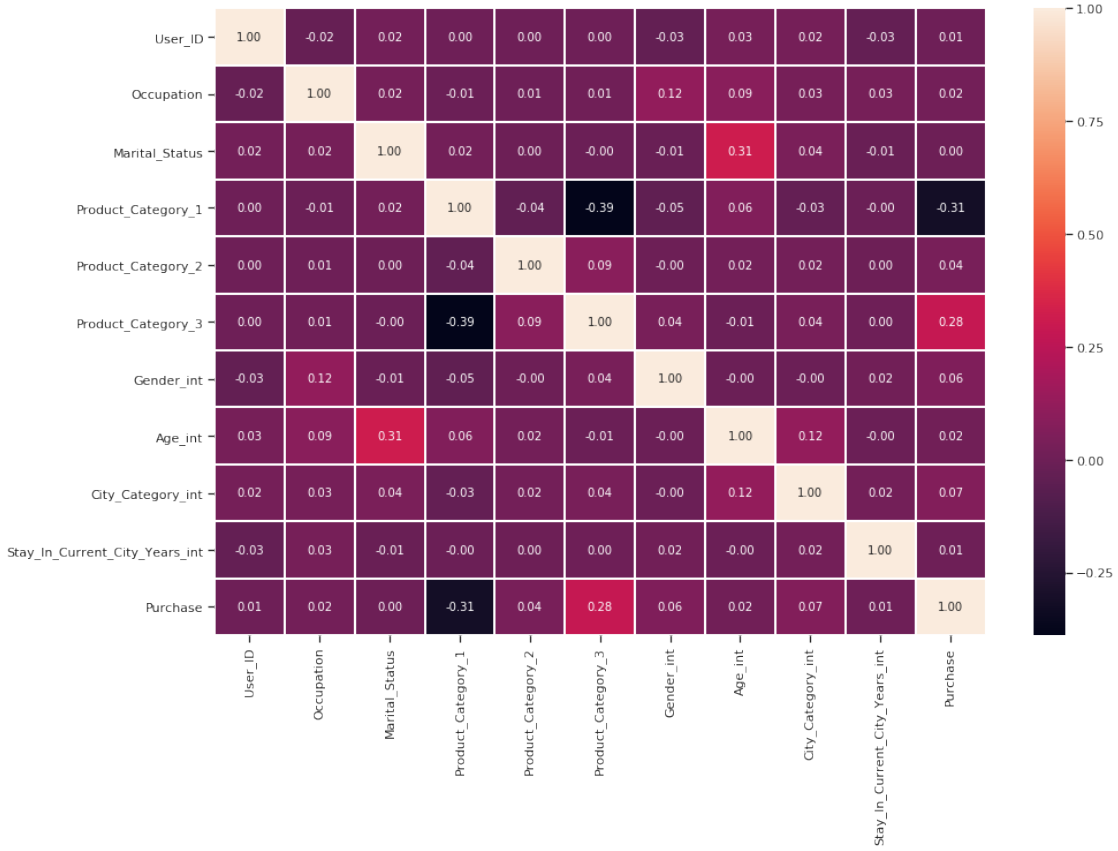



```
In [25]: data_new=data.drop(['Purchase'],axis='columns')
         data_new['Purchase']=sc_data

In [26]: plt.figure(figsize=(15, 10))
         sns.heatmap(data_new.corr(),annot=True, fmt='.2f', linewidths=1)

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0fba110630>

Out[26]:
```


User_ID 1.00 -0.02 0.02 0.00 0.00 0.00 -0.03 0.03 0.02 -0.03 0.01
Occupation -0.02 1.00 0.02 -0.01 0.01 0.01 0.12 0.09 0.03 0.03 0.02
Marital_Status 0.02 0.02 1.00 0.02 0.00 -0.00 -0.01 0.31 0.04 -0.01 0.00
Product_Category_1 0.00 -0.01 0.02 1.00 -0.04 -0.39 -0.05 0.06 -0.03 -0.00 -0.31
Product_Category_2 0.00 0.01 0.00 -0.04 1.00 0.09 -0.00 0.02 0.02 0.00 0.04
Product_Category_3 0.00 0.01 -0.00 -0.39 0.09 1.00 0.04 -0.01 0.04 0.00 0.28
Gender_int -0.03 0.12 -0.01 -0.05 -0.00 0.04 1.00 -0.00 -0.00 0.02 0.06
Age_int 0.03 0.09 0.31 0.06 0.02 -0.01 -0.00 1.00 0.12 -0.00 0.02
City_Category_int 0.02 0.03 0.04 -0.03 0.02 0.04 -0.00 0.12 1.00 0.02 0.07
Stay_In_Current_City_Years_int -0.03 0.03 -0.01 -0.00 0.00 0.00 0.02 -0.00 0.02 1.00 0.01
Purchase 0.01 0.02 0.00 -0.31 0.04 0.28 0.06 0.02 0.07 0.01 1.00
User_ID
Occupation
Marital_Status
Product_Category_1
Product_Category_2
Product_Category_3
Gender_int
Age_int
City_Category_int
Stay_In_Current_City_Years_int
Purchase
1.00
0.75
0.50
0.25
0.00
−0.25

In [0]: