**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»**

Курс «Технологии машинного обучения»

Лабораторная работа №2

Выполнил:

студент группы ИУ5-63Б

Комаров Д. С.

Проверил:

Гапанюк Ю. Е.

2024 г.

## **Задание**

*Выбрать набор данных (датасет), содержащий категориальные и числовые 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.) Просьба не использовать датасет, на котором данная задача решалась в лекции. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи: устранение пропусков в данных; кодирование категориальных признаков; нормализация числовых признаков.*

**Ход работы:**

**import** numpy **as** np  
**import** pandas **as** pd  
**import** seaborn **as** sns  
**import** matplotlib.pyplot **as** plt  
**from** sklearn.impute **import** SimpleImputer  
%matplotlib inline  
sns.set(style="ticks")

data\_loaded = pd.read\_csv('sample\_data/crimes.csv', sep=",")

*# размер набора данных*  
data\_loaded.shape

(25648, 12)

data\_loaded.head()

{"summary":"{\n \"name\": \"data\_loaded\",\n \"rows\": 25648,\n \"fields\": [\n {\n \"column\": \"CrimeDate\",\n \"properties\": {\n \"dtype\": \"object\",\n \"num\_unique\_values\": 186,\n \"samples\": [\n \"07/26/2016\",\n \"06/05/2016\",\n \"05/31/2016\"\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"CrimeTime\",\n \"properties\": {\n \"dtype\": \"category\",\n \"num\_unique\_values\": 1904,\n \"samples\": [\n \"06:29:00\",\n \"21:10:00\",\n \"04:16:00\"\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"CrimeCode\",\n \"properties\": {\n \"dtype\": \"category\",\n \"num\_unique\_values\": 74,\n \"samples\": [\n \"4E\",\n \"1O\",\n \"5B\"\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"Location\",\n \"properties\": {\n \"dtype\": \"category\",\n \"num\_unique\_values\": 9446,\n \"samples\": [\n \"1100 CAMBRIA ST\",\n \"2800 GREENMOUNT AV\",\n \"400 E BELVEDERE AVE\"\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"Description\",\n \"properties\": {\n \"dtype\": \"category\",\n \"num\_unique\_values\": 15,\n \"samples\": [\n \"ASSAULT BY THREAT\",\n \"HOMICIDE\",\n \"ROBBERY - STREET\"\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"Inside/Outside\",\n \"properties\": {\n \"dtype\": \"category\",\n \"num\_unique\_values\": 4,\n \"samples\": [\n \"I\",\n \"Inside\",\n \"O\"\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"Weapon\",\n \"properties\": {\n \"dtype\": \"category\",\n \"num\_unique\_values\": 4,\n \"samples\": [\n \"HANDS\",\n \"KNIFE\",\n \"FIREARM\"\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"Post\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 268.0528927926924,\n \"min\": 0.0,\n \"max\": 944.0,\n \"num\_unique\_values\": 147,\n \"samples\": [\n 0.0,\n 113.0,\n 314.0\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"District\",\n \"properties\": {\n \"dtype\": \"category\",\n \"num\_unique\_values\": 12,\n \"samples\": [\n \"SOUTHESTERN\",\n \"NORTHESTERN\",\n \"CENTRAL\"\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"Neighborhood\",\n \"properties\": {\n \"dtype\": \"category\",\n \"num\_unique\_values\": 277,\n \"samples\": [\n \"Howard Park\",\n \"Cylburn\",\n \"Wyman Park\"\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"Location 1\",\n \"properties\": {\n \"dtype\": \"string\",\n \"num\_unique\_values\": 17126,\n \"samples\": [\n \"(39.3582600000, -76.6016000000)\",\n \"(39.3424100000, -76.6827200000)\",\n \"(39.2867600000, -76.5633500000)\"\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n },\n {\n \"column\": \"Total Incidents\",\n \"properties\": {\n \"dtype\": \"number\",\n \"std\": 0.0,\n \"min\": 1.0,\n \"max\": 1.0,\n \"num\_unique\_values\": 1,\n \"samples\": [\n 1.0\n ],\n \"semantic\_type\": \"\",\n \"description\": \"\"\n }\n }\n ]\n}","type":"dataframe","variable\_name":"data\_loaded"}

*# Создаем список, содержащий признаки, их типы данных и количество пропусков*  
data\_features = list(zip(  
 *# Признаки*  
 [i **for** i **in** data\_loaded.columns],  
 zip(  
 *# Типы колонок*  
 [str(i) **for** i **in** data\_loaded.dtypes],  
 *# Проверим, есть ли пропущенные значения, и вычислим их процентное соотношение к общему количеству данных*  
 [i **for** i **in** data\_loaded.isnull().sum()],  
 *# Процентное соотношение пропущенных значений к общему количеству данных*  
 [(i / len(data\_loaded)) \* 100 **for** i **in** data\_loaded.isnull().sum()]  
 )  
))  
*# Выводим признаки с типом данных, количеством пропусков и их процентным соотношением к общему количеству данных*  
data\_features

[('CrimeDate', ('object', 0, 0.0)),  
 ('CrimeTime', ('object', 0, 0.0)),  
 ('CrimeCode', ('object', 1, 0.003898939488459139)),  
 ('Location', ('object', 104, 0.40548970679975044)),  
 ('Description', ('object', 1, 0.003898939488459139)),  
 ('Inside/Outside', ('object', 93, 0.36260137242669993)),  
 ('Weapon', ('object', 16485, 64.27401746724891)),  
 ('Post', ('float64', 8, 0.031191515907673113)),  
 ('District', ('object', 8, 0.031191515907673113)),  
 ('Neighborhood', ('object', 113, 0.44058016219588275)),  
 ('Location 1', ('object', 101, 0.3937928883343731)),  
 ('Total Incidents', ('float64', 1, 0.003898939488459139))]

*# проверим есть ли пропущенные значения*  
data\_loaded.isnull().sum()

CrimeDate 0  
CrimeTime 0  
CrimeCode 1  
Location 104  
Description 1  
Inside/Outside 93  
Weapon 16485  
Post 8  
District 8  
Neighborhood 113  
Location 1 101  
Total Incidents 1  
dtype: int64

data = data\_loaded  
  
*# удалим значения*  
data.dropna(subset=['CrimeCode', 'District', 'Location'], inplace=True)

imp = SimpleImputer(missing\_values=np.nan, strategy='most\_frequent')  
data\_imp\_df = pd.DataFrame(data=data\_imp, columns=['Weapon'])  
  
data['Weapon'] = data\_imp\_df['Weapon']

*# Создаем список, содержащий признаки, их типы данных и количество пропусков*  
data\_features = list(zip(  
 *# Признаки*  
 [i **for** i **in** data\_loaded.columns],  
 zip(  
 *# Типы колонок*  
 [str(i) **for** i **in** data\_loaded.dtypes],  
 *# Проверим, есть ли пропущенные значения, и вычислим их процентное соотношение к общему количеству данных*  
 [i **for** i **in** data\_loaded.isnull().sum()],  
 *# Процентное соотношение пропущенных значений к общему количеству данных*  
 [(i / len(data\_loaded)) \* 100 **for** i **in** data\_loaded.isnull().sum()]  
 )  
))  
*# Выводим признаки с типом данных, количеством пропусков и их процентным соотношением к общему количеству данных*  
data\_features

[('CrimeDate', ('object', 0, 0.0)),  
 ('CrimeTime', ('object', 0, 0.0)),  
 ('CrimeCode', ('object', 0, 0.0)),  
 ('Location', ('object', 0, 0.0)),  
 ('Description', ('object', 0, 0.0)),  
 ('Inside/Outside', ('object', 91, 0.35624804259317255)),  
 ('Weapon', ('object', 103, 0.4032258064516129)),  
 ('Post', ('float64', 0, 0.0)),  
 ('District', ('object', 0, 0.0)),  
 ('Neighborhood', ('object', 12, 0.04697776385844034)),  
 ('Location 1', ('object', 0, 0.0)),  
 ('Total Incidents', ('float64', 0, 0.0))]

# **кодирование категориальных признаков;**

# Выбран кодовый формат

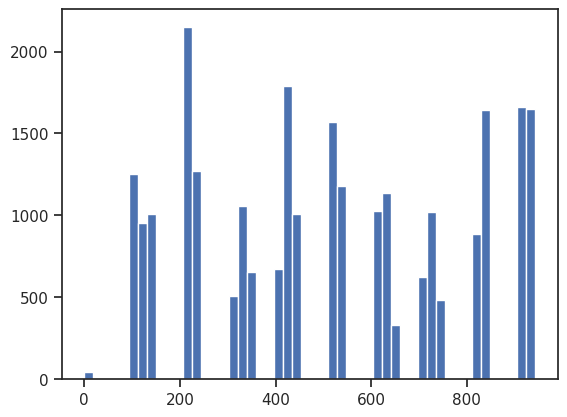
One-hot encoding предполагает, что значение категории заменяется на отдельную колонку, которая содержит бинарные значения.

dummies = pd.get\_dummies(data[['Weapon']])  
  
dummies = dummies.astype(int)  
  
print(dummies.head())

Weapon\_FIREARM Weapon\_HANDS Weapon\_KNIFE Weapon\_OTHER  
0 0 1 0 0  
1 1 0 0 0  
2 0 1 0 0  
3 0 1 0 0  
4 0 1 0 0

# **Масштабирование;**

**from** sklearn.preprocessing **import** MinMaxScaler, StandardScaler, Normalizer  
sc1 = MinMaxScaler()  
sc1\_data = sc1.fit\_transform(data[['Post']])  
plt.hist(data['Post'], 50)  
plt.show()



plt.hist(sc1\_data, 50)  
plt.show()

