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 «Радиотехнический»

Кафедра ИУ5 «Системы обработки информации и управления»

Лабораторная работа №2

по дисциплине «Технологии машинного обучения»

Выполнил:
студент группы РТ5-61Б
М.А. Ходосов

2022 г.

Задание лабораторной работы:

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.

Лабораторная работа №2

Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных.

Цель рабораторной работы: изучение способов предварительной обработки данных для дальнейшего формирования моделей.

Для работы возьмем набор данных из РК1, задача 3, датасет 6 (Human Resources)

Рассмотрим, что в нём находится:

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('../datasets/HRDataset_v14.csv')
data.shape
```

Out[2]:
```
(311, 36)
```

Рассмотрим, какие колонки есть и сколько пустых значений:

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 311 entries, 0 to 310
Data columns (total 36 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   Employee_Name               311 non-null    object 
 1   EmpID                       311 non-null    int64  
 2   MarriedID                   311 non-null    int64  
 3   MaritalStatusID             311 non-null    int64  
 4   GenderID                    311 non-null    int64  
 5   EmpStatusID                 311 non-null    int64  
 6   DeptID                      311 non-null    int64  
 7   PerfScoreID                 311 non-null    int64  
 8   FromDiversityJobFairID      311 non-null    int64  
 9   Salary                      311 non-null    int64  
 10  Termd                       311 non-null    int64  
 11  PositionID                  311 non-null    int64  
 12  Position                    311 non-null    object 
 13  State                       311 non-null    object 
 14  Zip                         311 non-null    int64  
 15  DOB                         311 non-null    object 
 16  Sex                         311 non-null    object 
 17  MaritalDesc                 311 non-null    object 
 18  CitizenDesc                 311 non-null    object 
 19  HispanicLatino              311 non-null    object 
 20  RaceDesc                    311 non-null    object 
 21  DateofHire                  311 non-null    object 
 22  DateofTermination           104 non-null    object 
 23  TermReason                  311 non-null    object 
 24  EmploymentStatus            311 non-null    object 
 25  Department                  311 non-null    object 
 26  ManagerName                 311 non-null    object 
 27  ManagerID                   303 non-null    float64
 28  RecruitmentSource           311 non-null    object 
 29  PerformanceScore            311 non-null    object 
 30  EngagementSurvey            311 non-null    float64
 31  EmpSatisfaction             311 non-null    int64  
 32  SpecialProjectsCount        311 non-null    int64  
 33  LastPerformanceReview_Date  311 non-null    object 
 34  DaysLateLast30              311 non-null    int64  
 35  Absences                    311 non-null    int64  
dtypes: float64(2), int64(16), object(18)
memory usage: 87.6+ KB
```

In [4]:
```
data.isnull().sum()
```

Out[4]:
```
Employee_Name                 0
EmpID                         0
MarriedID                     0
MaritalStatusID               0
GenderID                      0
EmpStatusID                   0
DeptID                        0
PerfScoreID                   0
FromDiversityJobFairID        0
Salary                        0
Termd                         0
PositionID                    0
Position                      0
State                         0
Zip                           0
DOB                           0
Sex                           0
MaritalDesc                   0
CitizenDesc                   0
HispanicLatino                0
RaceDesc                      0
DateofHire                    0
DateofTermination           207
TermReason                    0
EmploymentStatus              0
Department                    0
ManagerName                   0
ManagerID                     8
RecruitmentSource             0
PerformanceScore              0
EngagementSurvey              0
EmpSatisfaction               0
SpecialProjectsCount          0
LastPerformanceReview_Date    0
DaysLateLast30                0
Absences                      0
dtype: int64
```

Пустые значения есть в столбцах DateOfTermination и ManagerID.

Обработка пропусков в данных

Давайте просто удалим эти столбцы чтобы было проще работать.

Получим data_wec (without empty cols)

In [5]:
```
data_wec = data.dropna(axis = 1, how = 'any')
(data.shape, data_wec.shape)
```

Out[5]:
```
((311, 36), (311, 34))
```

Кодирование категориальных признаков

Закодируем колонку 'SEX' с помощью one-hot encoding

In [6]:
```
from sklearn.preprocessing import OneHotEncoder

ohe = OneHotEncoder()
data_tmp = data[['Sex']]
data_enc = ohe.fit_transform(data_tmp)
data_enc
```

Out[6]:
```
<311x2 sparse matrix of type '<class 'numpy.float64'>'
        with 311 stored elements in Compressed Sparse Row format>
```

In [7]:
```
data_enc.todense()[0:10]
```

Out[7]:
```
matrix([[0., 1.],
        [0., 1.],
        [1., 0.],
        [1., 0.],
        [1., 0.],
        [1., 0.],
        [1., 0.],
        [0., 1.],
        [1., 0.],
        [0., 1.]])
```

Таким образом, получили все возможные варианты пола человека (да, их два).

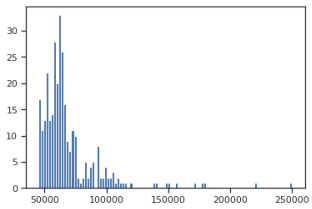
Масштабирование данных

Для масштабирования возьмем колонку Salary (зарплату).

In [8]:
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer

scaler = MinMaxScaler()
scale_data = scaler.fit_transform(data[['Salary']])

plt.hist(data['Salary'], 100)
plt.show()
```



In [9]:
```
plt.hist(scale_data, 100)
plt.show()
```