**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по лабораторной работе №3

Выполнил:  
Студент группы ИУ5-62Б   
Андреев Илья

Подпись и дата:

Проверил:  
преподаватель каф. ИУ5   
Гапанюк Ю.Е.

Подпись и дата:

г. Москва, 2020 г.

# Лабораторная работа 3

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

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

### 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](https://github.com/ugapanyuk/ml_course_2020/blob/master/common/notebooks/missing/handling_missing_norm.ipynb) решить следующие задачи:

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

**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

**import** **seaborn** **as** **sns**

**import** **matplotlib.pyplot** **as** **plt**

%**matplotlib** inline

**from** **sklearn.impute** **import** SimpleImputer

## Загрузка и первичный анализ

In [2]:

data = pd.read\_csv('train.csv')

In [3]:

data.head()

Out[3]:

|  | **PassengerId** | **Survived** | **Pclass** | **Name** | **Sex** | **Age** | **SibSp** | **Parch** | **Ticket** | **Fare** | **Cabin** | **Embarked** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | NaN | S |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NaN | S |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.1000 | C123 | S |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 | 8.0500 | NaN | S |

In [4]:

data.shape

Out[4]:

(891, 12)

In [5]:

data.dtypes

Out[5]:

PassengerId int64

Survived int64

Pclass int64

Name object

Sex object

Age float64

SibSp int64

Parch int64

Ticket object

Fare float64

Cabin object

Embarked object

dtype: object

In [6]:

data.isnull().sum()

Out[6]:

PassengerId 0

Survived 0

Pclass 0

Name 0

Sex 0

Age 177

SibSp 0

Parch 0

Ticket 0

Fare 0

Cabin 687

Embarked 2

dtype: int64

In [7]:

data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 891 entries, 0 to 890

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 PassengerId 891 non-null int64

1 Survived 891 non-null int64

2 Pclass 891 non-null int64

3 Name 891 non-null object

4 Sex 891 non-null object

5 Age 714 non-null float64

6 SibSp 891 non-null int64

7 Parch 891 non-null int64

8 Ticket 891 non-null object

9 Fare 891 non-null float64

10 Cabin 204 non-null object

11 Embarked 889 non-null object

dtypes: float64(2), int64(5), object(5)

memory usage: 83.7+ KB

## Обработка пропусков

In [8]:

*# Удаляем столбцы, которые не несут значимой информации*

data.drop(['Cabin','Name','Ticket'], axis = 1, inplace = **True**)

In [9]:

data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 891 entries, 0 to 890

Data columns (total 9 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 PassengerId 891 non-null int64

1 Survived 891 non-null int64

2 Pclass 891 non-null int64

3 Sex 891 non-null object

4 Age 714 non-null float64

5 SibSp 891 non-null int64

6 Parch 891 non-null int64

7 Fare 891 non-null float64

8 Embarked 889 non-null object

dtypes: float64(2), int64(5), object(2)

memory usage: 62.8+ KB

In [10]:

*# Заполняем отсутствующие значения возраста средним возрастом*

data['Age'] = data['Age'].replace(0,np.nan)

data['Age'] = data['Age'].fillna(data['Age'].mean())

In [11]:

data.isnull().sum()

Out[11]:

PassengerId 0

Survived 0

Pclass 0

Sex 0

Age 0

SibSp 0

Parch 0

Fare 0

Embarked 2

dtype: int64

## Преобразование категориальных признаков в числовые

In [12]:

data['Embarked'].value\_counts()

Out[12]:

S 644

C 168

Q 77

Name: Embarked, dtype: int64

In [13]:

*# Кодируем признаки Pclass и Embarked в отдельные столбцы*

data = pd.get\_dummies(data, columns=['Pclass','Embarked'])

In [14]:

*# Пол кодируем в 1/0*

data['IsMale']=data.Sex.replace({'female':0,'male':1})

data.drop('Sex', axis = 1, inplace = **True**)

In [15]:

data.head()

Out[15]:

|  | **PassengerId** | **Survived** | **Age** | **SibSp** | **Parch** | **Fare** | **Pclass\_1** | **Pclass\_2** | **Pclass\_3** | **Embarked\_C** | **Embarked\_Q** | **Embarked\_S** | **IsMale** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | 0 | 22.0 | 1 | 0 | 7.2500 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| **1** | 2 | 1 | 38.0 | 1 | 0 | 71.2833 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| **2** | 3 | 1 | 26.0 | 0 | 0 | 7.9250 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| **3** | 4 | 1 | 35.0 | 1 | 0 | 53.1000 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| **4** | 5 | 0 | 35.0 | 0 | 0 | 8.0500 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |

## Масштабирование значений

In [16]:

**from** **sklearn.preprocessing** **import** StandardScaler

In [17]:

sc2 = StandardScaler()

sc2.fit(data)

sc2\_data = sc2.transform(data)

In [18]:

data = pd.DataFrame(sc2\_data, columns = data.columns)

In [19]:

data.head()

Out[19]:

|  | **PassengerId** | **Survived** | **Age** | **SibSp** | **Parch** | **Fare** | **Pclass\_1** | **Pclass\_2** | **Pclass\_3** | **Embarked\_C** | **Embarked\_Q** | **Embarked\_S** | **IsMale** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | -1.730108 | -0.789272 | -0.592481 | 0.432793 | -0.473674 | -0.502445 | -0.565685 | -0.510152 | 0.902587 | -0.482043 | -0.307562 | 0.619306 | 0.737695 |
| **1** | -1.726220 | 1.266990 | 0.638789 | 0.432793 | -0.473674 | 0.786845 | 1.767767 | -0.510152 | -1.107926 | 2.074505 | -0.307562 | -1.614710 | -1.355574 |
| **2** | -1.722332 | 1.266990 | -0.284663 | -0.474545 | -0.473674 | -0.488854 | -0.565685 | -0.510152 | 0.902587 | -0.482043 | -0.307562 | 0.619306 | -1.355574 |
| **3** | -1.718444 | 1.266990 | 0.407926 | 0.432793 | -0.473674 | 0.420730 | 1.767767 | -0.510152 | -1.107926 | -0.482043 | -0.307562 | 0.619306 | -1.355574 |
| **4** | -1.714556 | -0.789272 | 0.407926 | -0.474545 | -0.473674 | -0.486337 | -0.565685 | -0.510152 | 0.902587 | -0.482043 | -0.307562 | 0.619306 | 0.737695 |