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, искусственный интеллект и системы управления

КАФЕДРА Системы обработки информации и управления

Методические указания к лабораторным работам по курсу «Машинное обучение»

«Обработка признаков (часть 1)»

Выполнил Поташников М.Д. (ИУ5-24М)

Москва, 2023 г.

ЗАДАНИЕ

1. 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.
2. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:
    1. устранение пропусков в данных;
    2. кодирование категориальных признаков;
    3. нормализация числовых признаков.

Сформировать отчет и разместить его в своем репозитории на github.

```
import sklearn
import pandas as pd
import seaborn as sns
from sklearn.datasets import load_diabetes
import matplotlib.pyplot as plt
import numpy as np

data = pd.DataFrame(pd.read_csv('/content/mission_launches.csv'))
```

Возьмем датасет с информацией о космических программах со всего мира.

```
data.drop(labels = ['Unnamed: 0.1', 'Unnamed: 0'], inplace = True,
axis= 1)
data
```

```
data.isnull().sum()
```

```
Unnamed: 0.1         0
Unnamed: 0           0
Organisation         0
Location             0
Date                 0
Detail               0
Rocket_Status        0
Price             3360
Mission_Status       0
dtype: int64
```

Немного преобразуем числовой признак, чтобы конвертировать в Float:

```
for i, dat in enumerate(data['Price']):
  if type(dat) == str:
    if dat.find(',')>=0:
      dat = dat.replace(',', '')
      data['Price'][i] = dat
```

Для каждой космической компании считаем среднее значение стоимости космической программы:

```
nan_arr = []
for org in data['Organisation'].unique():
  nan_arr.append([org, data[data['Organisation'] == org]
['Price'].dropna().astype('float').mean()])
```

Заменяем пропущенные значения на среднюю стоимость космических программ соответствующих компаний:

```
for i, price in enumerate(data['Price']):
  if pd.isnull(price):
    for j in nan_arr:
      if j[0] == data['Organisation'][i]:
```

```
        price = j[1]
        break
    data['Price'][i] = price
```

Пропущенных значений хоть и стало намного меньше, но все еще остались. Видимо для некоторых компаний вообще нет информации о стоимости. Тогда просто удалим такие строчки.

```
data.isnull().sum()
```

```
Organisation        0
Location            0
Date                0
Detail              0
Rocket_Status       0
Price             458
Mission_Status      0
dtype: int64
```

```
data.dropna(inplace = True)
```

```
from sklearn.preprocessing import LabelEncoder
```

```
data['Mission_Status'].unique()
```

```
array(['Success', 'Failure', 'Partial Failure', 'Prelaunch Failure'],
      dtype=object)
```

```
le = LabelEncoder()
data['Mission_Status'] = le.fit_transform(data['Mission_Status'])
```

```
data['Mission_Status'].unique()
```

```
array([3, 0, 1, 2])
```

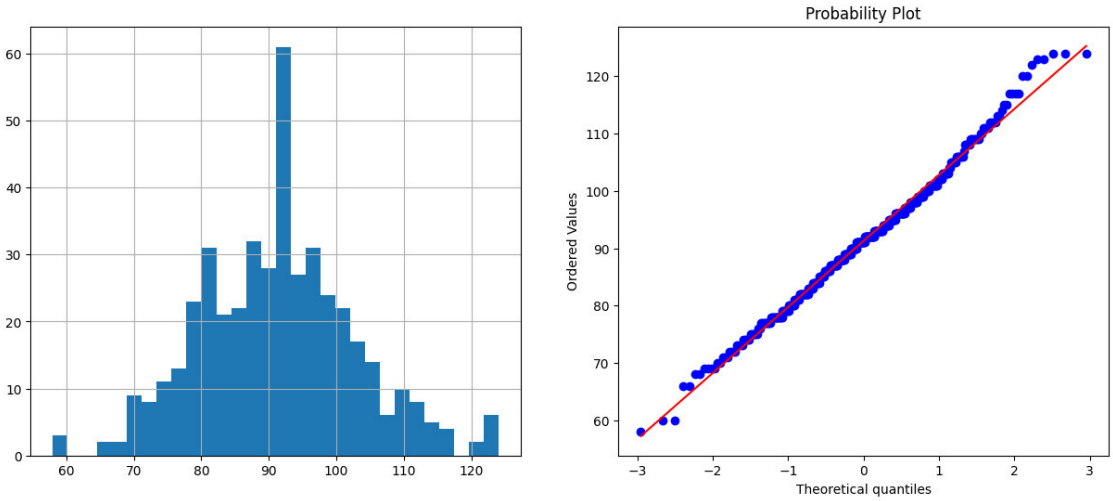
```
import scipy.stats as stats
from sklearn.datasets import load_diabetes

def diagnostic_plots(df, variable):
    plt.figure(figsize=(15,6))
    # гистограмма
    plt.subplot(1, 2, 1)
    df[variable].hist(bins=30)
    ## Q-Q plot
    plt.subplot(1, 2, 2)
    stats.probplot(df[variable], dist="norm", plot=plt)
    plt.show()

df = load_diabetes(as_frame=True, scaled=False)

data = df['data']
target = df['target']
```

```
diagnostic_plots(data, 's6')
```




```
data['s6_boxcox'], param = stats.boxcox(data['s6'])
print('Оптимальное значение λ = {}'.format(param))
diagnostic_plots(data, 's6_boxcox')
```

```
Оптимальное значение λ = 0.5104884693621882
```