Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»


# Рубежный контроль №1

## по дисциплине «Методы машинного обучения»

Методы обработки данных

Вариант 1

ИСПОЛНИТЕЛЬ:

студент ИУ5-23М
Анцифров Н.С.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю. Е.

___ "_____" 2024 г.

Москва, 2024

## 1 Вариант и задачи

Задачи по варианту представлены в таблице 1.

Таблица 1 – Задачи по варианту

| Номер варианта | Номер задачи №1 | Номер задачи №2 |
|---|---|---|
| 1 | 1 | 21 |

*Задание для студентов группы ИУ5-23М:*

Для произвольной колонки данных построить график «Ящик с усами (boxplot)».

*Задача №1:*

Для набора данных проведите кодирование одного (произвольного) категориального признака с использованием метода «count (frequency) encoding».

*Задача №21:*

Для набора данных проведите масштабирование данных для одного (произвольного) числового признака с использованием масштабирования по медиане.

## 2 Описание набора данных

В качестве предметной области был выбран набор данных, содержащий данные об автомобилях, проданных за некоторый период на территории США. Данный набор доступен по адресу: https://www.kaggle.com/datasets/goyalshalini93/car-data.

Набор данных имеет следующие атрибуты:

- *car_ID* – порядковый номер строки;
- *symboling* – обозначение;
- *CarName* – марка + модель автомобиля;
- *fueltype* – тип топлива;

– *aspiration* – тип подачи воздуха в двигатель (атмосферный/турбированный);
– *doornumber* – число дверей;
– *carbody* – тип кузова;
– *drivewheel* – привод;
– *enginelocation* – расположение двигателя;
– *wheelbase* – длина колесной базы;
– *carlength* – длина автомобиля;
– *carwidth* – ширина автомобиля;
– *carheight* – высота автомобиля;
– *curbweight* – снаряженная масса;
– *enginetype* – тип двигателя;
– *cylindernumber* – число цилиндров;
– *enginesize* – объем двигателя;
– *fuelsystem* – тип топливной системы;
– *boreratio* – интерес для покупателя;
– *stroke* – поршни;
– *compressionratio* – компрессия;
– *horsepower* – лошадиные силы;
– *peakrpm* – обороты в минуты, при которых достигается максимальный момент;
– *citympg* – расход топлива по городу;
– *highwaympg* – расход по трассе;
– *price* – цена.

Для дальнейшей работы оставим столбцы *car_ID*, *CarName*, *fueltype*, *doornumber*, *carbody*, *drivewheel*, *horsepower*, *price*.

### 3 Вывода графика «Ящик с усами»

Выведем график «Ящик с усами» для столбца *horsepower*. Он показывает распределение параметра в диапазоне. Представим код и сам график на рисунке 1.

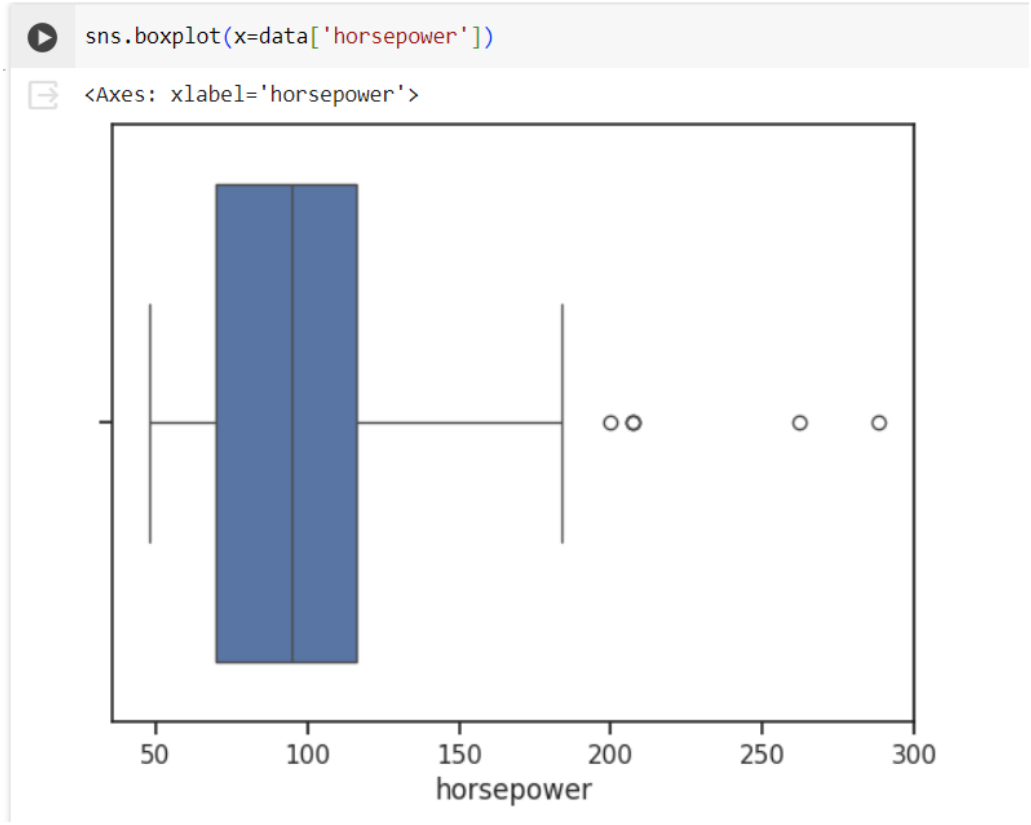



Рисунок 1 – Представление «Ящика с усами» для столбца *horsepower*

### 4 Решение задачи №1

В качестве категориального признака возьмём признак *carbody*, обозначающий тип кузова. Закодируем его с использованием метода «count / frequency encoding». Кодирование «count encoding» представим на рисунке 2, «frequency encoding» – на рисунке 3.

```
[7] from category_encoders.count import CountEncoder as ce_CountEncoder
```

```
[8] ce_CountEncoder1 = ce_CountEncoder()
    data_COUNT_ENC = ce_CountEncoder1.fit_transform(data['carbody'])
```

```
[9] data_COUNT_ENC
```

| | carbody |
|---|---|
| 0 | 6 |
| 1 | 6 |
| 2 | 70 |
| 3 | 96 |
| 4 | 96 |
| ... | ... |
| 200 | 96 |
| 201 | 96 |
| 202 | 96 |
| 203 | 96 |
| 204 | 96 |

205 rows × 1 columns

Next steps: View recommended plots

```
[10] data['carbody'].unique()
```

```
array(['convertible', 'hatchback', 'sedan', 'wagon', 'hardtop'],
      dtype=object)
```

```
[11] data_COUNT_ENC['carbody'].unique()
```

```
array([ 6, 70, 96, 25,  8])
```

Рисунок 2 – Кодирование count encoding

```
[12] ce_CountEncoder2 = ce_CountEncoder(normalize=True)
     data_FREQ_ENC = ce_CountEncoder2.fit_transform(data['carbody'])
     data_FREQ_ENC
```

| | carbody |
|---|---|
| 0 | 0.029268 |
| 1 | 0.029268 |
| 2 | 0.341463 |
| 3 | 0.468293 |
| 4 | 0.468293 |
| ... | ... |
| 200 | 0.468293 |
| 201 | 0.468293 |
| 202 | 0.468293 |
| 203 | 0.468293 |
| 204 | 0.468293 |

205 rows × 1 columns

Next steps: View recommended plots

```
[13] data['carbody'].unique()
```

```
array(['convertible', 'hatchback', 'sedan', 'wagon', 'hardtop'],
      dtype=object)
```

```
[15] data_FREQ_ENC['carbody'].unique()
```

```
array([0.02926829, 0.34146341, 0.46829268, 0.12195122, 0.03902439])
```

Рисунок 3 – Кодирование frequency encoding

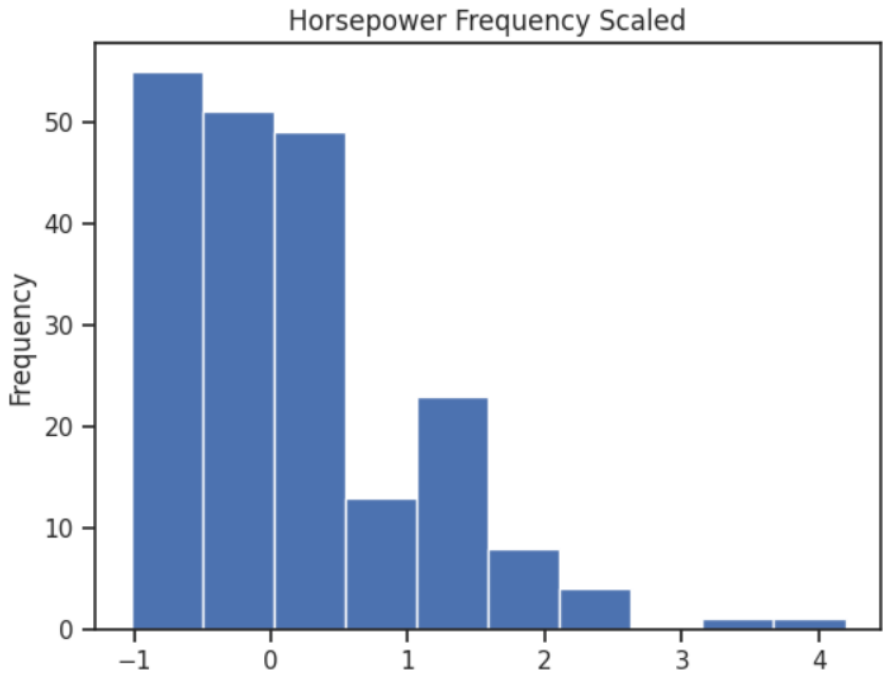
## 5 Решение задачи №21

Проведём масштабирование числового признака *horsepower* с использованием масштабирования по медиане. Представим процесс масштабирования с использованием RobustScaler на рисунке 4.

```
[16] from sklearn.preprocessing import RobustScaler
```

```
[20] sc = RobustScaler()
     data_sc = pd.Series(sc.fit_transform(data[['horsepower']]).reshape(1,-1)[0])
     data_sc.plot.hist(title='Horsepower Frequency Scaled')
```

```
<Axes: title={'center': 'Horsepower Frequency Scaled'}, ylabel='Frequency'>
```




```
[22] data_sc.median()
```

```
0.0
```

Рисунок 4 – Масштабирование по медиане