Московский государственный технический университет им. Н.Э. Баумана

Микросервис для предсказания стоимости книги

26.12.2024

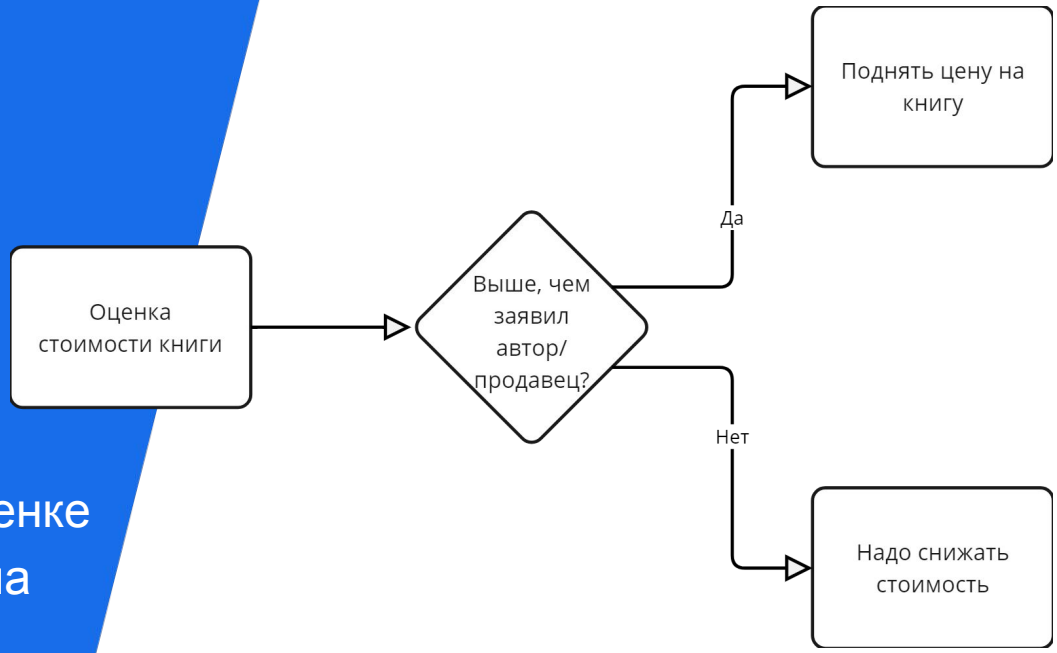
План презентации

1. Где использовать микросервис
2. Архитектура микросервиса
3. Данные для обучения
4. Модель для оценки текстов
5. Модель для работы с обложками
6. Модель для предсказания цены книги
7. Микросервис

Где использовать?

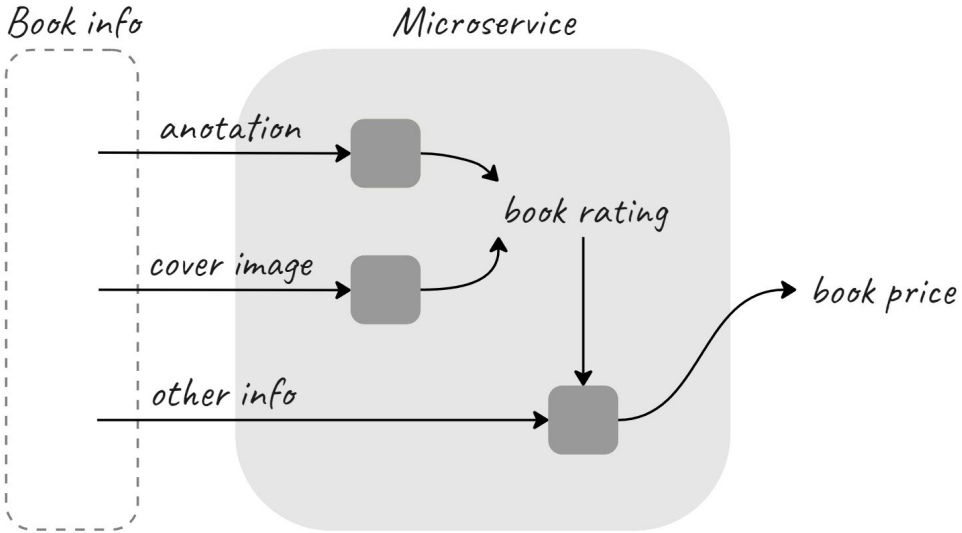
- В рекомендательных системах

- Помощь молодым авторам в оценке формата книги, её потенциала на рынке



Архитектура

- всего три модели
- 2 из них – предсказание рейтинга книги
- третья модель – предсказание цены книги



О данных для обучения

- Код парсера можно посмотреть тут

- Датасет с описанием книг был собран на основе данных с сайта labirint.ru

Лабиринт

https://www.labirint.ru/books/951782/

О данных

Парсер работал 18 часов

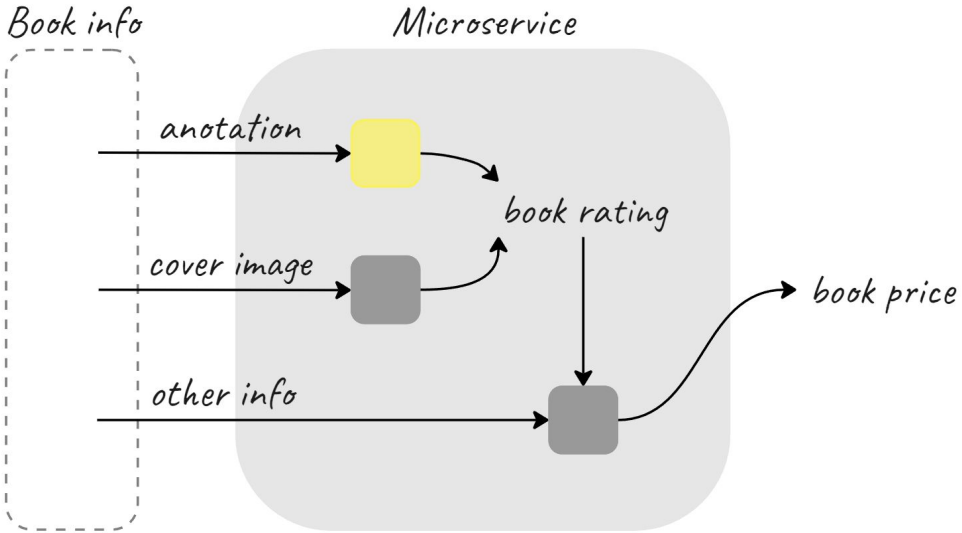
Почти 600 000 книг

1 Гб текстовых данных

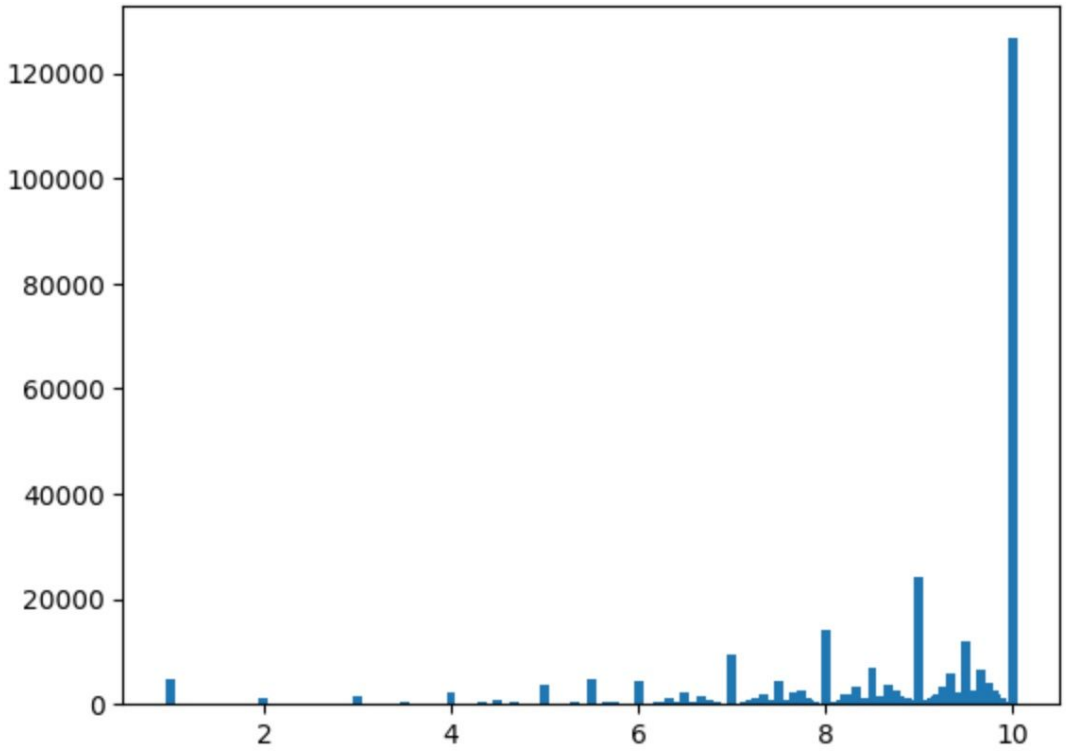
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
id	586077.0	NaN	NaN	NaN	555181.830352	266475.604056	33.0	331794.0	554479.0	782954.0	1020197.0
typeObject	583741	2	Книги	534210	NaN	NaN	NaN	NaN	NaN	NaN	NaN
groupOfType	583891	20	Нехудожественная литература	198064	NaN	NaN	NaN	NaN	NaN	NaN	NaN
underGroup	582969	102	Детская художественная литература	43916	NaN	NaN	NaN	NaN	NaN	NaN	NaN
genres	557745	404	Проза для детей	23423	NaN	NaN	NaN	NaN	NaN	NaN	NaN
bookName	586077	482189	Сказки	437	NaN	NaN	NaN	NaN	NaN	NaN	NaN
imgUrl	586077	586077	https://static10.labirint.ru/books/33/cover.jpg	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
age	194714.0	NaN	NaN	NaN	13.65947	3.733651	6.0	12.0	16.0	16.0	18.0
authors	488208	134270	Пушкин Александр Сергеевич	1011	NaN	NaN	NaN	NaN	NaN	NaN	NaN
publisher	585954	3798	АСТ	56341	NaN	NaN	NaN	NaN	NaN	NaN	NaN
datePublisher	585954.0	NaN	NaN	NaN	2015.663369	5.73397	1900.0	2011.0	2016.0	2020.0	2066.0
series	423675	39641	Учебники для вузов. Специальная литература	2294	NaN	NaN	NaN	NaN	NaN	NaN	NaN
bookGenres	158822	919	Современная отечественная проза	43493	NaN	NaN	NaN	NaN	NaN	NaN	NaN
allPrice	98271.0	NaN	NaN	NaN	1843.243958	2169.719353	29.0	488.0	1104.0	2603.0	80229.0
myPrice	98271.0	NaN	NaN	NaN	925.894048	1077.27878	17.0	246.0	558.0	1303.0	40115.0
sale	98271.0	NaN	NaN	NaN	50.646756	4.512729	25.0	50.0	50.0	50.0602	75.0
isbn	585343	584840	9781234567890	5	NaN	NaN	NaN	NaN	NaN	NaN	NaN
pages	571038.0	NaN	NaN	NaN	271.398262	4276.485748	1.0	96.0	240.0	363.0	3102023.0
pageType	552684	13	Офсет	333910	NaN	NaN	NaN	NaN	NaN	NaN	NaN
weight	577130.0	NaN	NaN	NaN	356.433663	354.12447	1.0	152.0	284.0	440.0	29134.0
da	577217.0	NaN	NaN	NaN	218.863445	45.148845	1.0	200.0	210.0	240.0	2947.0
db	577217.0	NaN	NaN	NaN	156.000688	41.254757	1.0	132.0	145.0	172.0	3297.0
dc	577217.0	NaN	NaN	NaN	16.832294	12.517661	1.0	8.0	15.0	23.0	895.0
covers	575085	19	обл - мягкий переплет (крепление скрепкой или...	259860	NaN	NaN	NaN	NaN	NaN	NaN	NaN
decoration	181160	1408	Частичная лакировка	70386	NaN	NaN	NaN	NaN	NaN	NaN	NaN
illustrations	554916	4	Без иллюстраций	259364	NaN	NaN	NaN	NaN	NaN	NaN	NaN
rate	390236.0	NaN	NaN	NaN	8.691699	1.729368	1.0	8.0	9.25	10.0	10.0
rateSize	586077.0	NaN	NaN	NaN	6.157269	23.754369	0.0	0.0	2.0	6.0	5725.0
annotation	586077	537621	Книжка-раскраска.Для детей младшего школьного ...	471	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Модель для оценки текста

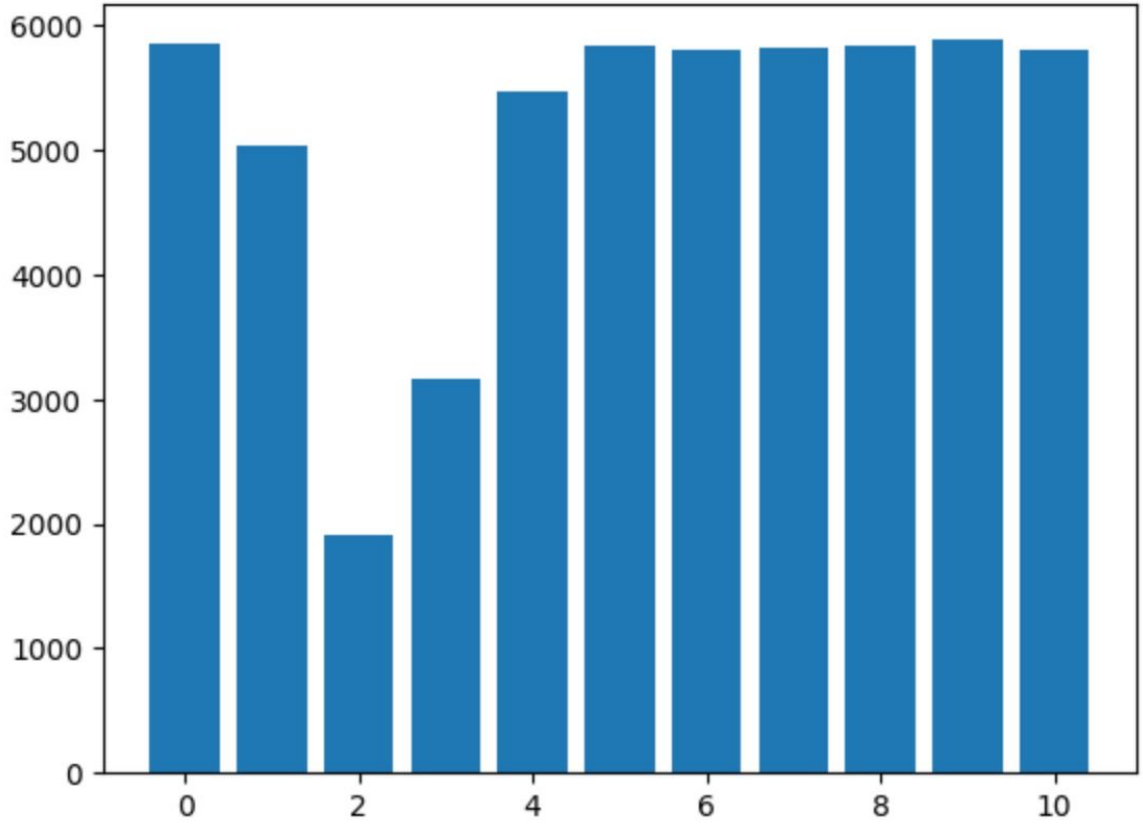
- BERT
- Классификация 11 классов
- Бинарная классификация



Дисбаланс классов

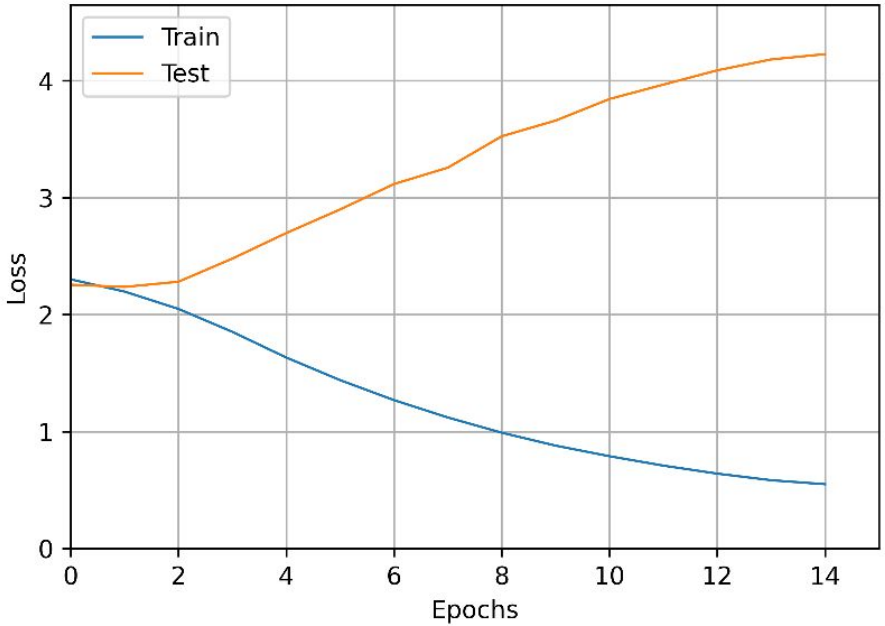


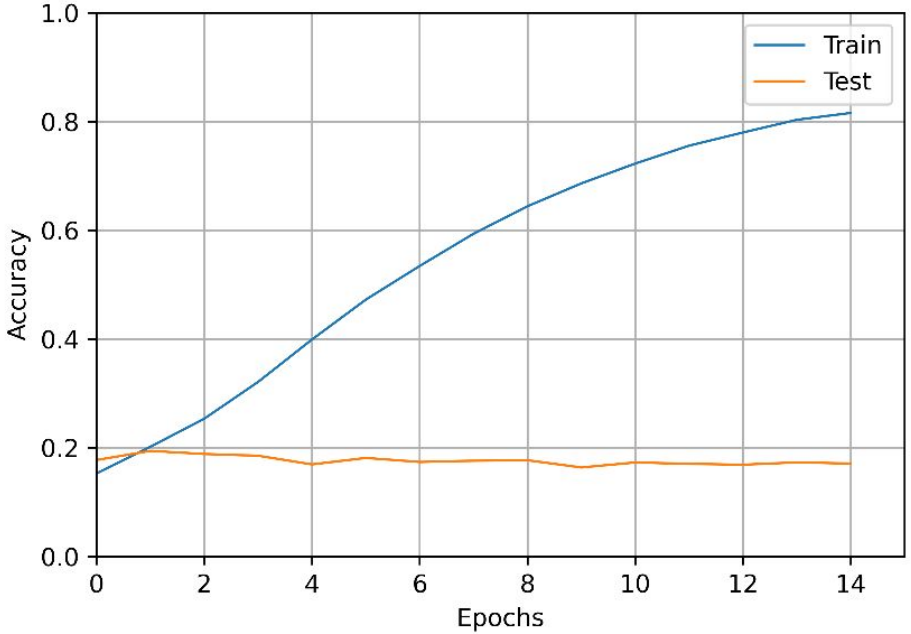
Классов одинаковое количество



11 классов – рейтинг книги от 0 до 10

Модель: DeepPavlov





Бинарная классификация

разделили на 2 класса

0 – оценка < 9.25

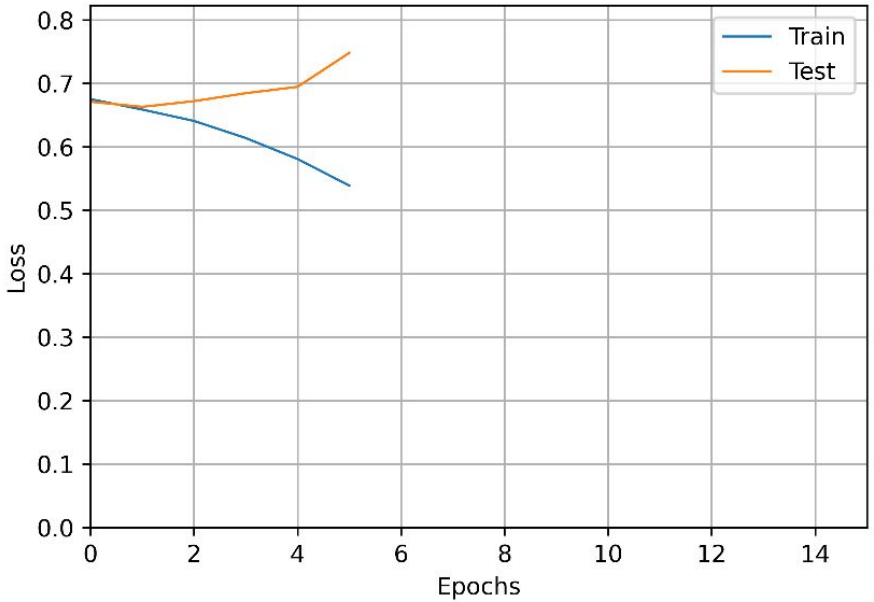
1 – оценка >= 9.25

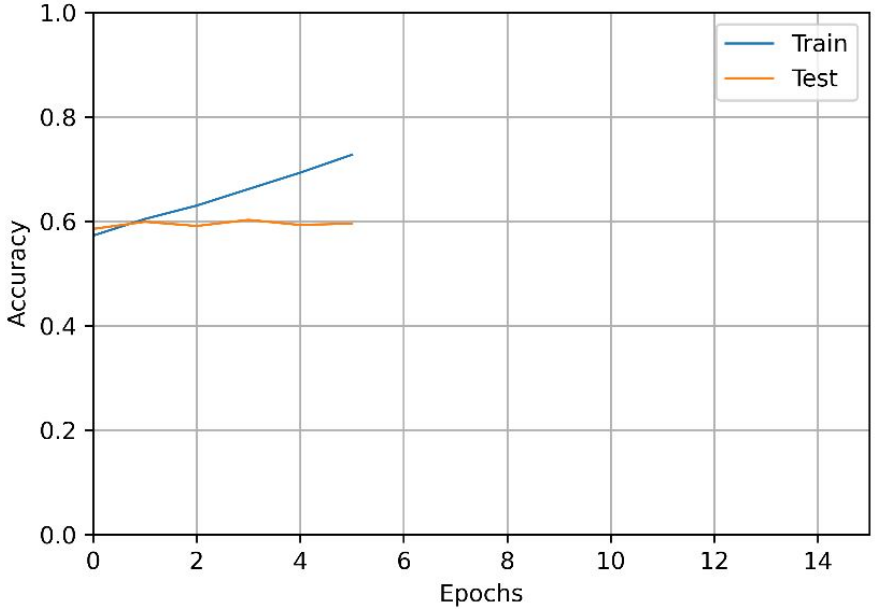
соотношение между классами:

195537 194699

LaBSE

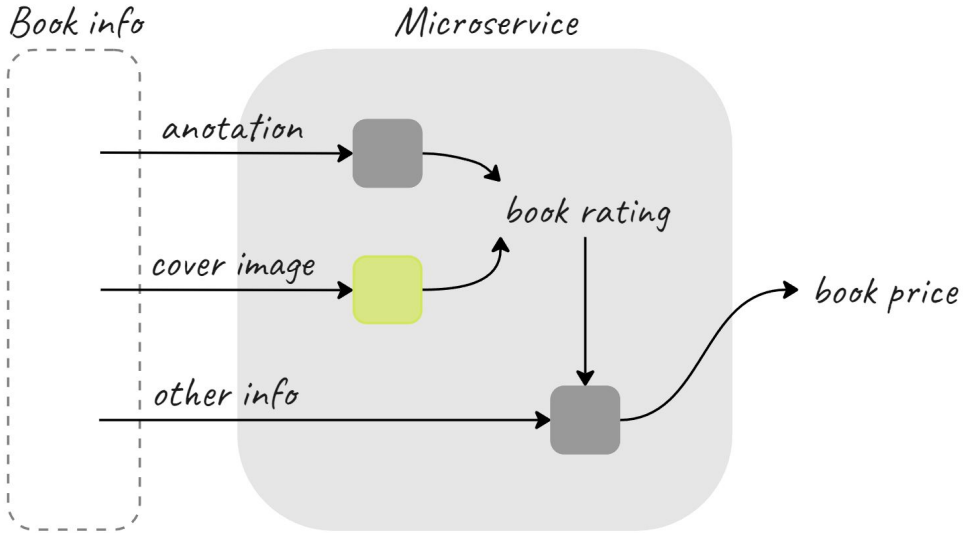
Best accuracy(Train/Test): 0.71/0.6





Модель для работы с обложками книг

- CNN
- Классификация 11 классов
- Бинарная классификация

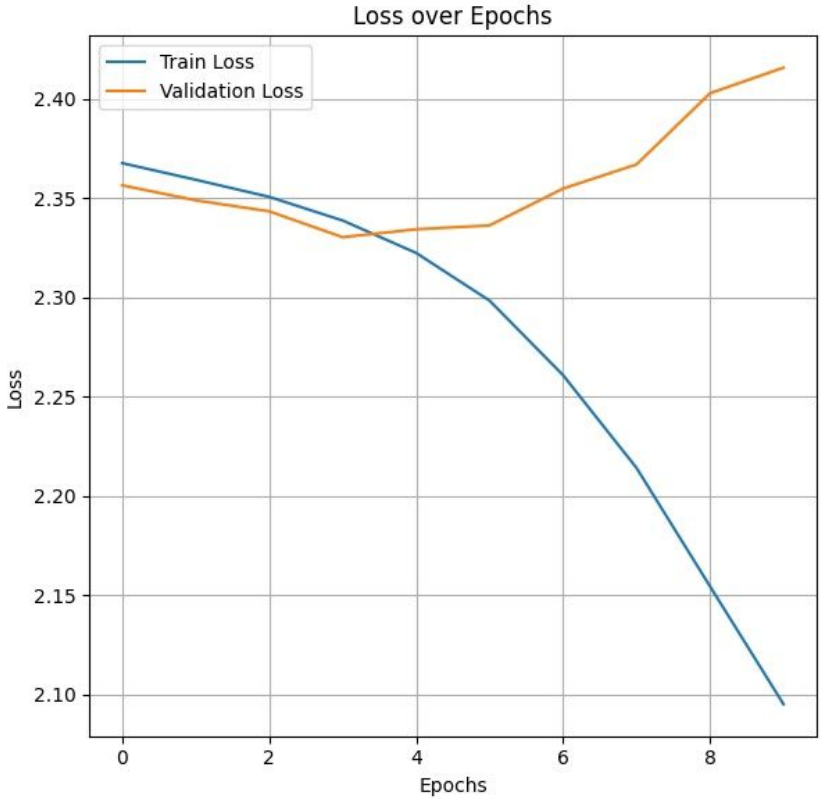


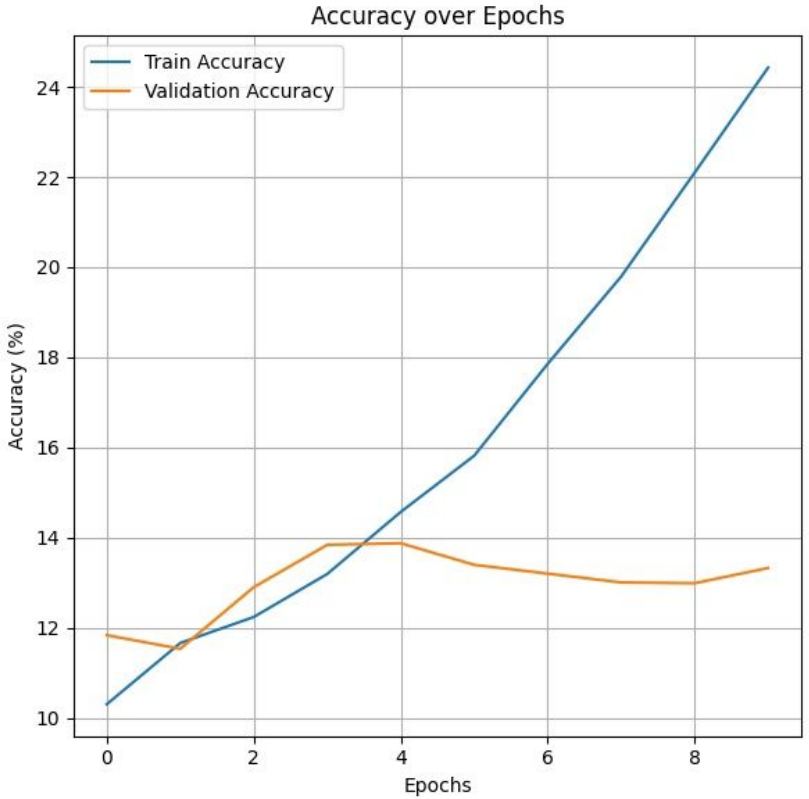
Классификация 11 классов

```
class CNN(nn.Module):
    def __init__(self, num_classes):
        super(CNN, self).__init__()
        self.conv_layers = nn.Sequential(
            nn.Conv2d(3, 32, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2),
            nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2),
            nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )
        self.fc_layers = nn.Sequential(
            nn.Flatten(),
            nn.Linear(128 * (IMG_SIZE // 8) * (IMG_SIZE // 8),
128),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(128, num_classes)
        )
```

- loss – CrossEntropyLoss()
- optimizer – Оптимизатор: Adam

Обучение


Loss over Epochs
Train Loss
Validation Loss
Loss
Epochs
2.40
2.35
2.30
2.25
2.20
2.15
2.10
0
2
4
6
8


Accuracy over Epochs
Train Accuracy
Validation Accuracy
Accuracy (%)
Epochs
24
22
20
18
16
14
12
10
0
2
4
6
8

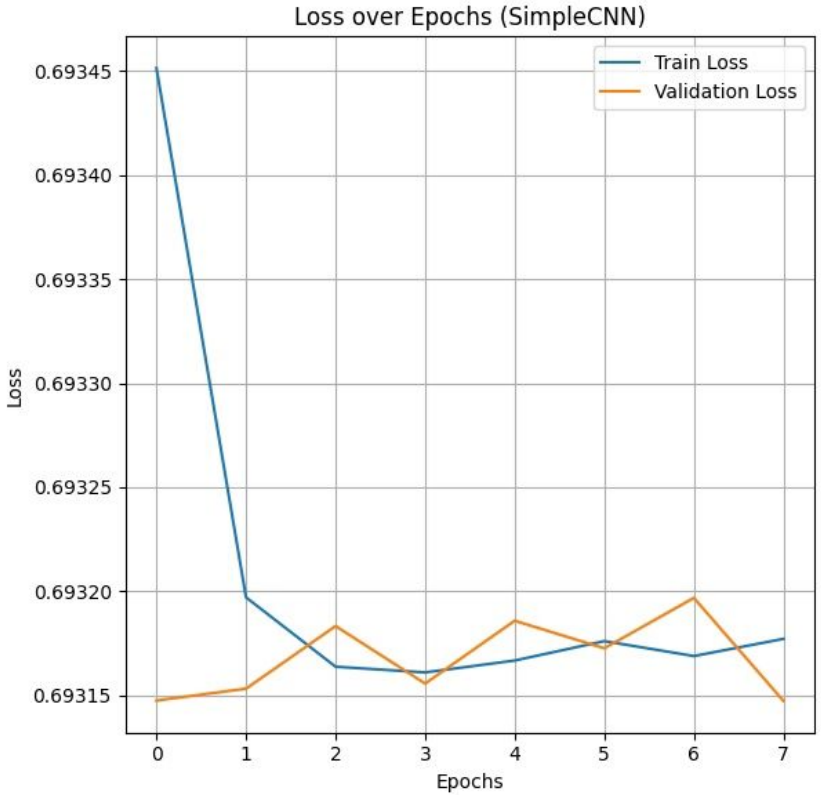
Бинарная классификация. CNN #1

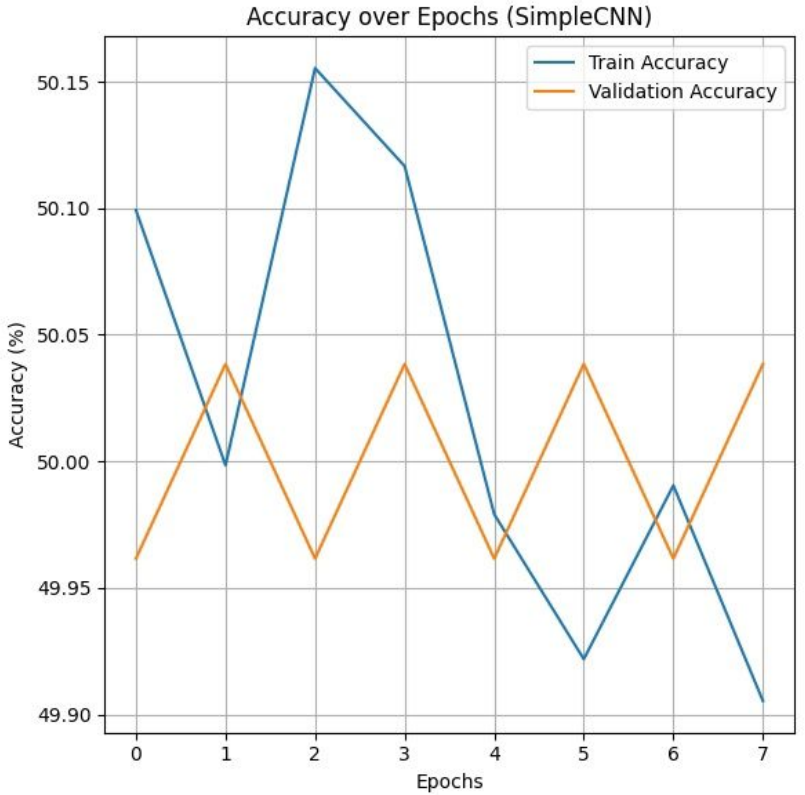
```
class SimpleCNN(nn.Module):
    def __init__(self, img_size=128):
        super(SimpleCNN, self).__init__()
        self.conv_layers = nn.Sequential(
            nn.Conv2d(3, 32, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2),
            nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )
        self.fc_layers = nn.Sequential(
            nn.Flatten(),
            nn.Linear(64 * (img_size // 4) * (img_size // 4), 128),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(128, 1),
            nn.Sigmoid()
        )

    def forward(self, x):
        return self.fc_layers(self.conv_layers(x))
```

loss – BCELoss()

Обучение CNN #1



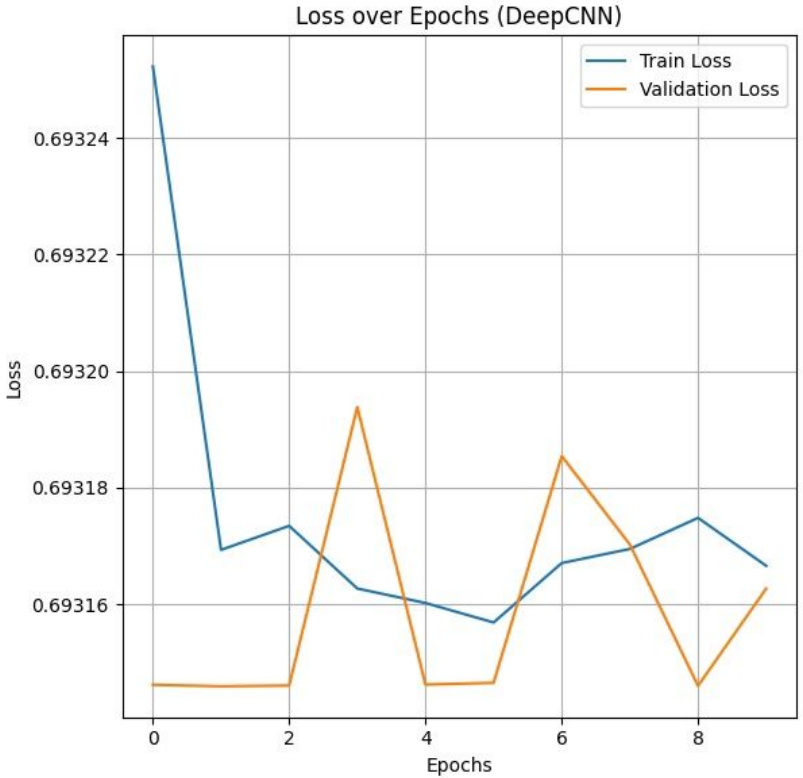


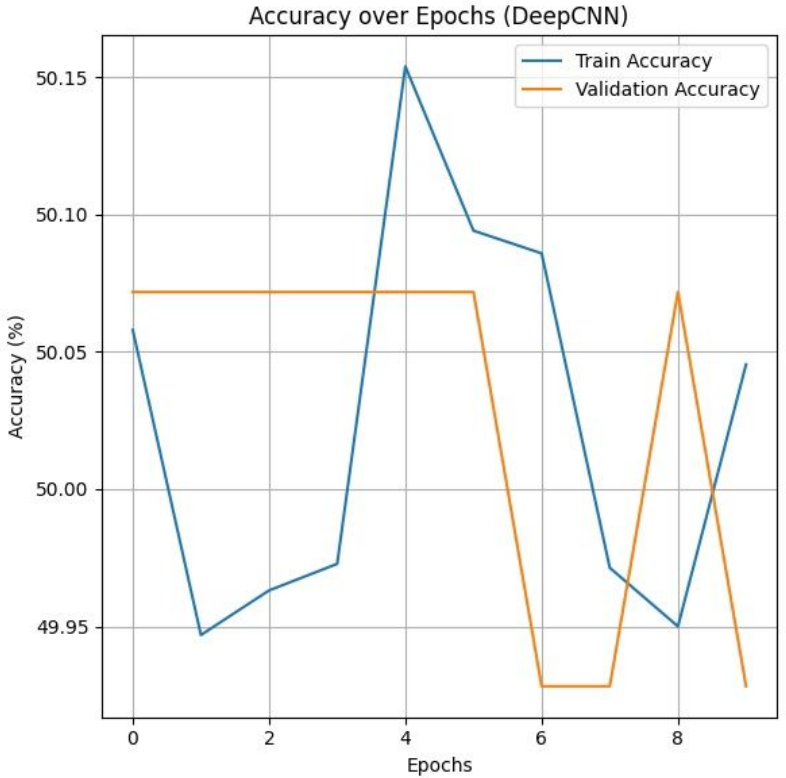
Бинарная классификация. CNN #2

```
class DeepCNN(nn.Module):
    def __init__(self, img_size=128):
        super(DeepCNN, self).__init__()
        self.conv_layers = nn.Sequential(
            nn.Conv2d(3, 32, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2),
            nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.Conv2d(128, 256, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )
        self.fc_layers = nn.Sequential(
            nn.Flatten(),
            nn.Linear(256 * (img_size // 4) * (img_size // 4), 256),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(256, 1),
            nn.Sigmoid()
        )
```

loss – BCELoss()

Обучение CNN #2



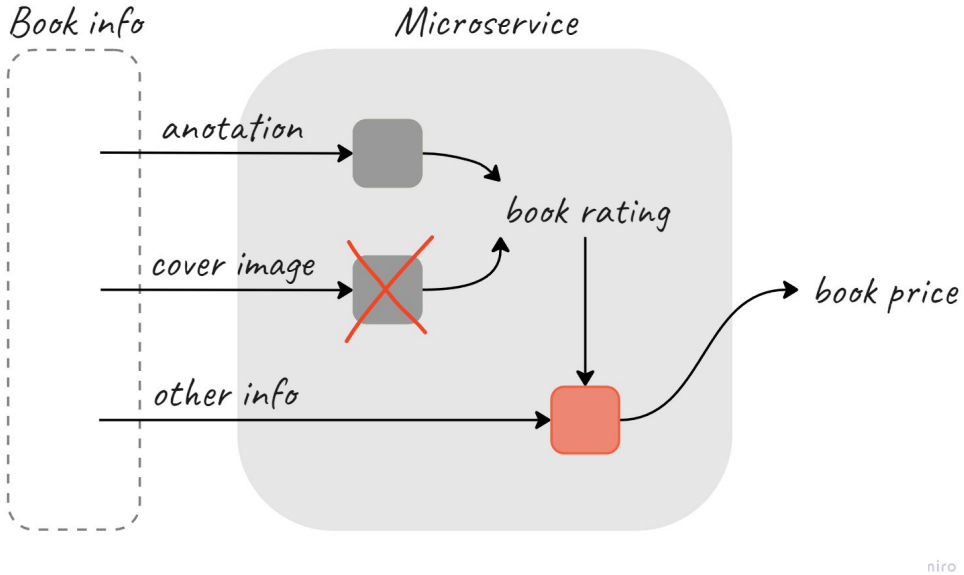


Вывод

Нельзя судить о книге по обложке!

Модель для предсказания цены

1. Предобработка данных
   1.1. Промежуточная
   1.2. С помощью тепловой карты корреляций
   1.3. Отбор значимых признаков
2. Обучение нейронной сети
3. Оценка модели



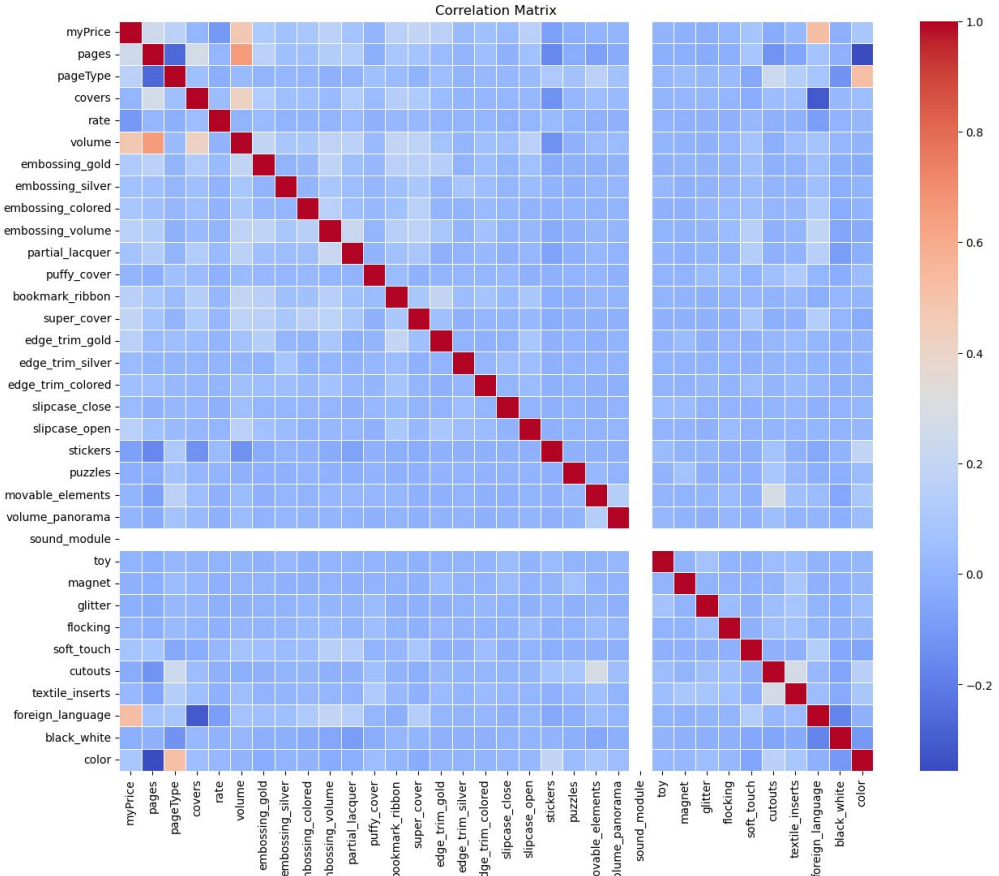
Предобработка данных

['Unnamed: 0', 'id', 'typeObject', 'groupOfType', 'underGroup', 'genres', 'bookName', 'imgUrl', 'age', 'authors', 'publisher', 'datePublisher', 'series', 'bookGenres', 'allPrice', 'myPrice', 'sale', 'isbn', 'pages', 'pageType', 'weight', 'da', 'db', 'dc', 'covers', 'decoration', 'illustrations', 'rate', 'rateSize', 'annotation']

	pages	volume	covers	pageType	foreign_language	color	black_white	partial_lacquer	slipcase_open	bookmark_ribbon	rate	embossing_gold	embossing_volume
1	448.0	482.6	9	1	0	0	1	0	0	1	0	0	0
2	320.0	991.4	11	4	0	1	0	0	0	0	0	0	0
3	95.0	307.3	0	1	0	1	0	0	0	0	0	0	0
4	112.0	312.0	0	1	0	1	1	0	0	0	0	0	0
5	176.0	291.9	0	1	0	0	0	0	0	0	0	0	0
6	576.0	1093.0	11	0	0	0	0	0	0	0	1	0	0
7	560.0	995.7	11	0	0	0	1	0	0	0	1	0	0
8	464.0	586.3	11	0	0	0	0	0	0	0	1	0	0
9	512.0	642.7	11	0	0	0	0	0	0	0	1	0	0
10	512.0	642.7	11	0	0	0	1	0	0	0	1	0	0
11	464.0	623.7	11	0	0	0	0	0	0	0	1	0	0
12	608.0	790.7	11	0	0	0	0	0	0	0	1	0	0
13	156.0	819.0	11	1	0	0	1	0	0	0	0	0	0
14	8.0	139.5	7	5	0	1	0	0	0	0	0	0	0

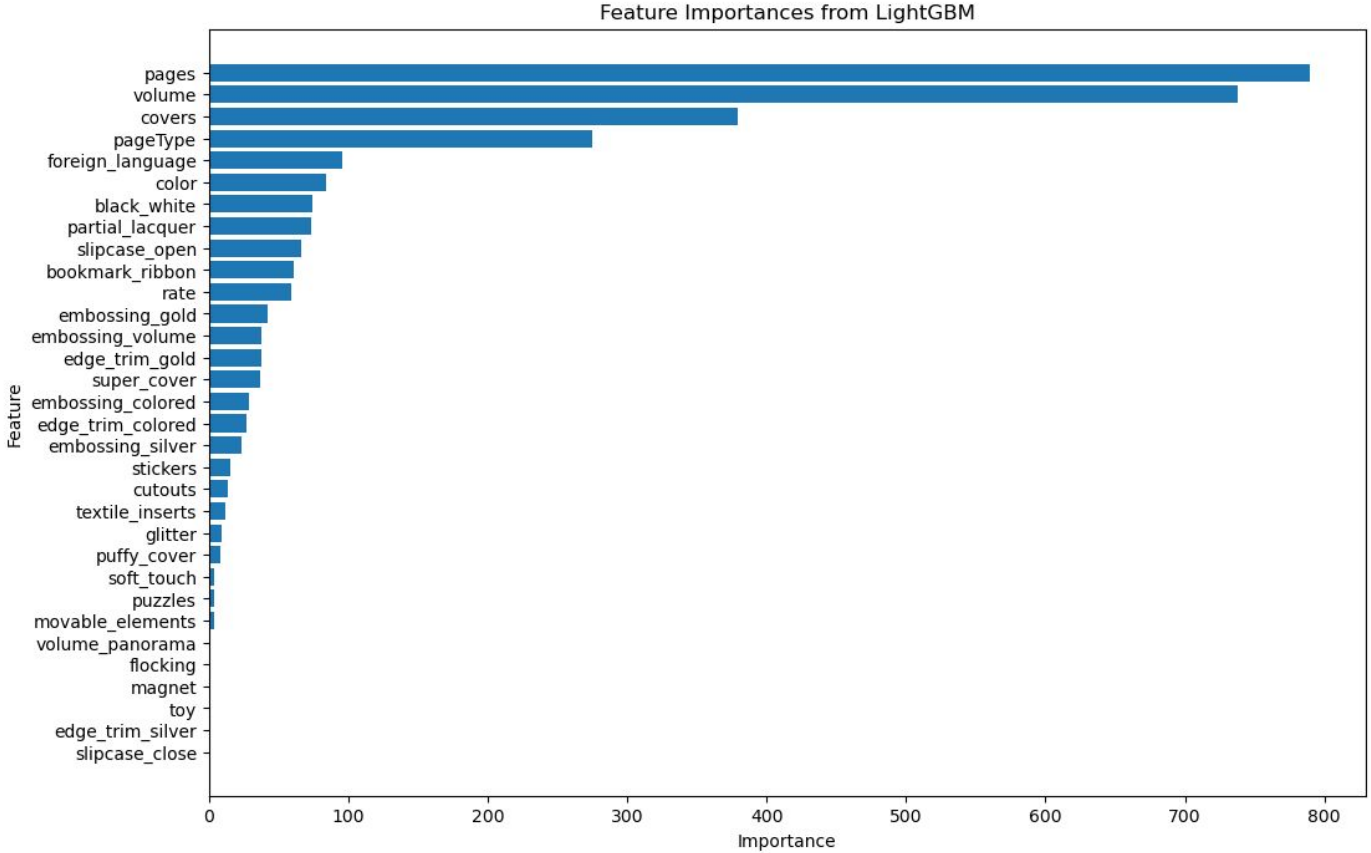
Признаки после промежуточной обработки

Предобработка данных. Тепловая карта корреляций



	Variable1	Variable2	Correlation
39	pages	volume	0.654479
31	myPrice	foreign_language	0.522521
101	pageType	color	0.517418
5	myPrice	volume	0.471289
107	covers	volume	0.415888
743	movable_elements	cutouts	0.287476
1016	cutouts	textile_inserts	0.275956
37	pages	covers	0.272274
1	myPrice	pages	0.252241
97	pageType	cutouts	0.246228

Предобработка данных. Анализ важности признаков



	Feature	Importance
0	pages	790
4	volume	738
2	covers	379
1	pageType	275
29	foreign_language	96
31	color	84
30	black_white	74
9	partial_lacquer	73
17	slipcase_open	66
11	bookmark_ribbon	61
3	rate	59
5	embossing_gold	42
8	embossing_volume	38
13	edge_trim_gold	38
12	super_cover	37
7	embossing_colored	29
15	edge_trim_colored	27
6	embossing_silver	23

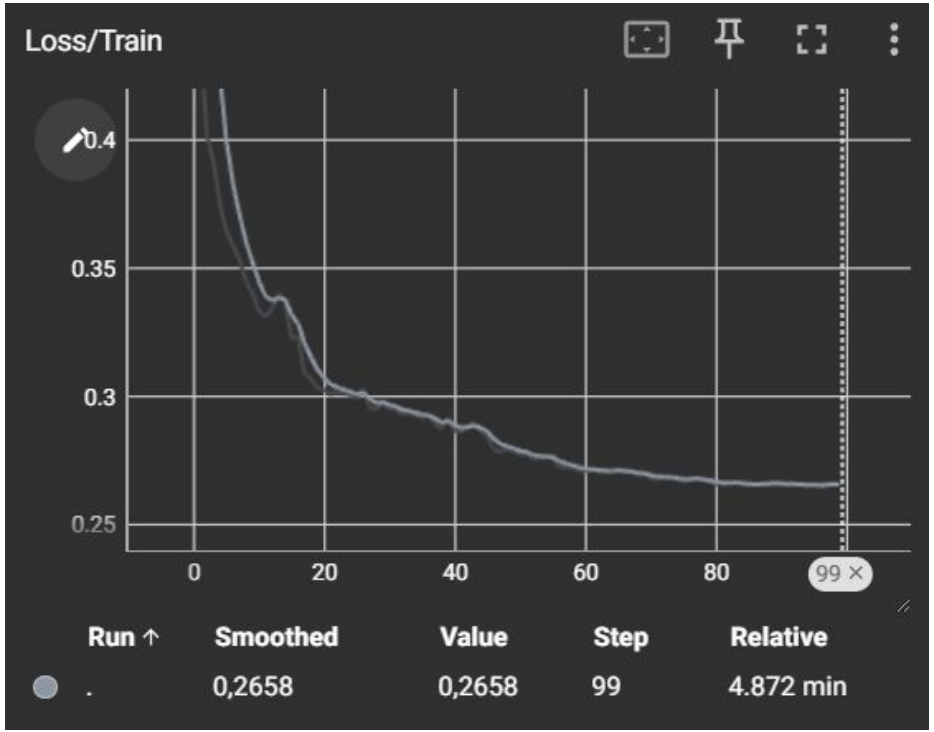
Архитектура модели

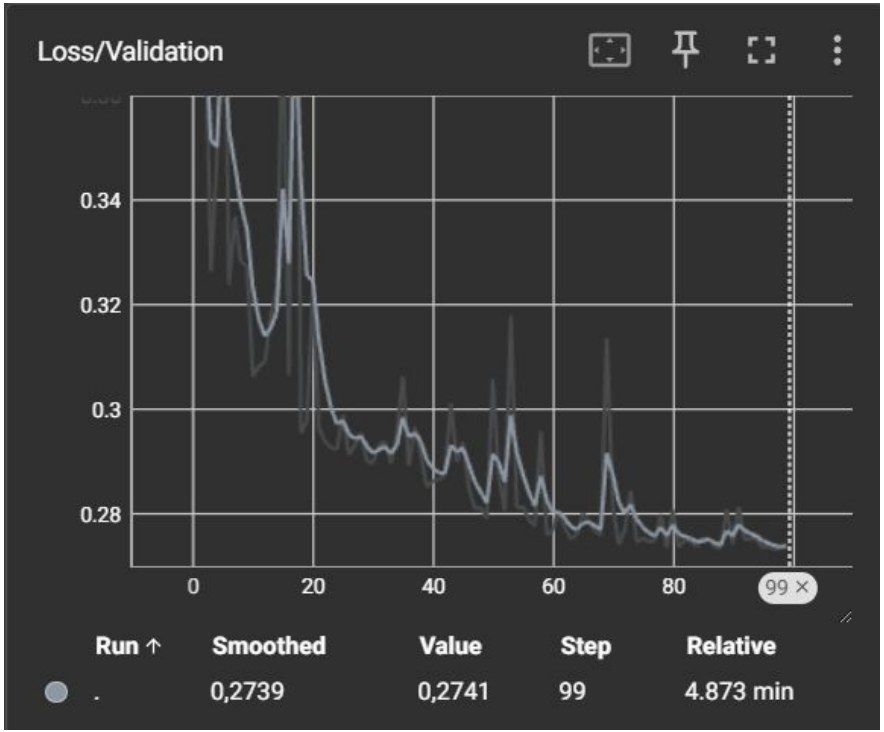
```
AdvancedPriceRegressor(
  (fc1): Linear(in_features=18, out_features=512, bias=True)
  (bn1): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU()
  (dropout1): Dropout(p=0.2, inplace=False)
  (fc2): Linear(in_features=512, out_features=256, bias=True)
  (bn2): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU()
  (dropout2): Dropout(p=0.2, inplace=False)
  (fc3): Linear(in_features=256, out_features=128, bias=True)
  (relu3): ReLU()
  (fc_out): Linear(in_features=128, out_features=1, bias=True)
  (residual): Linear(in_features=18, out_features=1, bias=True)
)
```

Обучение модели

- Функция потерь: MSELoss
- Оптимизатор: AdamW
- Планировщик обучения: ReduceLROnPlateau
- Ранняя остановка: отсутствие прогресса на протяжении 10 эпох

График потерь для Train и Validation





Оценка модели

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_1|}{n}$$

$$MSE = \frac{1}{N}\sum_{i=0}^{N} (\hat{y}_i - y_i)^2$$

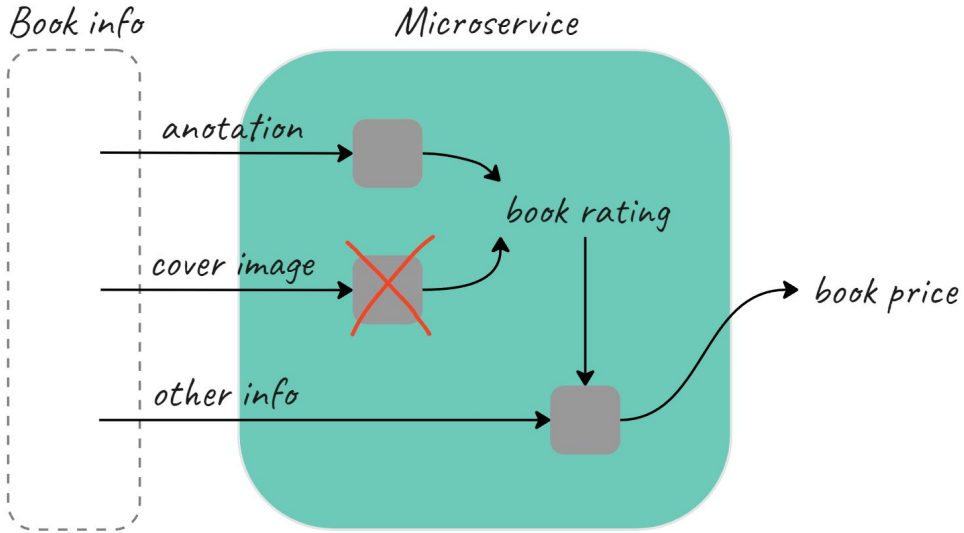
$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

Test MAE: 0.3991, MSE: 0.2929, R^2: 0.7366

Микросервис

Микросервис содержит 2 модели

от модели для работы с обложками книг пришлось отказаться



```
class MicroService:
    def __init__(self, cleaner, annotationclassifier, bookregressor, device):
        ...

    def _str_transform(self, string, size):
        ...

    def __call__(self, dataframe: pd.DataFrame):
        ...
```

Пример работы

Джефф Хокинс: 1000 мозгов. Новая теория интеллекта ✓ На складе

A thousand brains. A new theory of intelligence

Автор: Хокинс Джефф
Переводчик: Черников Сергей
Издательство: Портал, 2024 г.
Серия: Мастерская мозга

16+

Цена для всех ~~1328~~
Ваша цена (-50%) 664 ₽

Добавить в корзину

Добавить в отложенные
+ к сравнению

ID товара: 918432
ISBN: 978-5-907473-58-4
Страниц: 368 (Офсет) — прочитаете за 8 дней
Оформление
Масса: 504 г
Размеры: 213x148x24 мм
Содержание Полистать книгу

Рейтинг 8.59

Оценить (оценило: 17) ★★★★★★★★★☆☆

Аннотация к книге "1000 мозгов. Новая теория интеллекта"

Несмотря на все достижения нейробиологии, мы мало продвинулись в решении ее главного вопроса: какова биологическая природа интеллекта? Автор предлагает ответ, построенный на сенсационном научном открытии. В книге он рассказывает о том, для чего нейроны коры головного мозга объединились в странные сообщества под названием "кортикальные колонки". В неокортексе их сто пятьдесят тысяч. Джефф Хокинс и его команда установили, что каждая колонка, составляющая кору головного мозга, создает собственную модель мира. В итоге образуется не одна модель, а тысячи, и наше восприятие - это коллективное решение кортикальных колонок, принятое ими путем голосования.
Новый взгляд на деятельность головного мозга авторы идеи назвали теорией "тысячи мозгов". Они утверждают, что не искусственные нейронные сети, а открытые ими законы работы неокортекса лягут в основу развития искусственного интеллекта в будущем.
"То, что здесь описано, так волнующе, так возбуждающе, что превратит ваш разум в...

Читать полностью

ПОКУПАЮТ ВМЕСТЕ

Пять причин купить

1 Джефф Хокинс — один из самых успешных предпринимателей в Кремниевой

```
y = ms(pd.DataFrame(test))
np.exp(float(y[0][0]))
```

100%

100%

100%

496.4209040863347

"Данные — это новая нефть, но их переработка — это ключ к настоящей ценности."
(c) Клайв Хамби (Clive Humby)