

Прогнозирование рыночной цены мониторов с использованием методов машинного обучения в Google Colab

Бредихин Вадим Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описывается использование регрессионного анализа для прогнозирования рыночной стоимости мониторов. Для проведения исследования был использован облачный сервис Google Colab. В качестве исходных данных были собраны характеристики мониторов различных брендов по данным сайта www.dns-shop.ru. В результате исследования, путём создания моделей, удалось достичь точности прогнозирования стоимости покупки монитора до 98,1% по показателю R-квадрата.

Ключевые слова: регрессионный анализ, метод визуализации корреляций, Pandas, XGBoost, Python, Google Colab.

Predicting the market price of monitors using machine learning methods in Google Colab

Bredikhin Vadim Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the use of regression analysis to predict the market value of monitors. The Google Colab cloud service was used to conduct the study. As initial data, the characteristics of monitors of various brands were collected according to the website www.dns-shop.ru. As a result of the research, by creating models, it was possible to achieve the accuracy of predicting the cost of buying a monitor up to 98,1% in terms of the R-square indicator.

Keywords: Regression Analysis, a method for visualizing correlations, Pandas, XGBoost, Python, Google Colab.

1 Введение

1.1 Актуальность

Прогнозирование рыночной цены товаров является актуальной задачей для многих отраслей экономики, включая рынок электроники. Мониторы – один из наиболее востребованных продуктов, популярность которого обусловлена широким спектром применений от домашних пользователей до профессиональных дизайнеров и геймеров. Для производителей и продавцов

крайне важно иметь возможность точно предсказывать рыночные тенденции и формировать конкурентоспособные ценовые стратегии.

Использование методов машинного обучения позволяет значительно повысить точность прогнозирования стоимости продукции. Современные инструменты анализа больших объемов данных, такие как платформы Google Colab позволяют исследователям и аналитикам создавать модели, способные учитывать разнообразные факторы влияния на цену товара.

1.2 Обзор исследований

В работе Н.А. Труженникова, С.А. Сорокина и О.В. Косниковой рассматривается применение методов машинного обучения для анализа экономических тенденций и прогнозирования состояния рынков [1]. В исследовании С.С. Сенина проводится сравнительный анализ эффективности различных моделей машинного обучения [2]. В публикации Н.П. Перстеновой и Д.Д. Скрылевой рассматриваются основные проблемы и перспективы внедрения технологий машинного обучения в экономическом секторе. Акцент сделан на преимуществах и ограничениях каждого подхода [3]. В обзоре П.Г. Полторацкого изучаются способы эффективного прогнозирования потребительского спроса на товары в условиях электронной торговли [4]. В работе В.Д. Новикова и Р.М. Хамитов представляется исследование способов точного прогнозирования поведения потребителей на основе данных поведенческой аналитики [5].

1.3 Цель исследования

Целью исследования является построение моделей прогнозирования цены мониторов на основе регрессионного анализа.

2 Материалы и методы

Для решения поставленной цели отлично подойдет бесплатная облачная среда Google Colab [6], разработанная для написания программного кода с использованием различных библиотек машинного обучения.

В статье собраны данные за май 2025 г. Все данные были собраны с сайта www.dns-shop.ru. В работу были взяты модели мониторов последних лет выпуска. Данные о товарах собирались по следующим параметрам: цена, производитель, модель, частота процессора, дискретная видеокарта, оперативная память, SSD, экран. Вся информация была собрана в файл формата MS Excel. (рис.1).

№	Производитель	Модель	Частота Процессора ГГц	Дискретная видеокарта ГВ	Оперативная память ГВ	SSD ГВ	Экран ГЦ	Цена
1	Делл	GS15-6217	2.5	4	8	256	120	69999
2	Lenovo	IdeaPad Gaming 3 15IMH05	2.5	4	8	512	60	69999
3	Acer	Aspire 7 A715-75G-52NW	2.5	4	8	256	60	72999
4	MSI	GP76 Katana 115C-483RU	2.7	4	8	256	144	77999
5	MSI	GF63 Thin 11UC-353RU	2.7	4	8	256	144	79999
6	HP	Victus 16-e0119ur	3.3	4	8	512	144	79999
7	Acer	Nitro 5 AN517-52-55V9	2.5	4	8	512	120	79999
8	Dream Machines	RG3050-15RU20	2.7	4	16	512	144	80999
9	ASUS	TUF Gaming F15 FX506LH-HN036	2.5	4	16	512	144	81999
10	ASUS	VivoBook Pro 15 K3500PH-11136W	3.1	4	8	256	60	82999
11	Acer	Aspire 7 A715-42G-R4GR	2.1	4	8	512	144	82999
12	Делл	Inspiron 15 Plus 7510	2.7	4	8	512	60	82999
13	HP	VICTUS 16-e0099ur	3.3	4	16	512	144	83999
14	Делл	GS15-6901	3.3	4	8	512	120	84999
15	MSI	GP630 THIN 12UC-353RU	2.7	4	8	256	144	84999

Рисунок 1 – Фрагмент таблицы с данными MS Excel

В таблице рисунка 1 представлены мониторы разных производителей, однако это не главный параметр и практически не влияет на формирование стоимости. Для создания регрессионных моделей воспользуемся облачной средой Google Colab, а именно встроенными библиотеками pandas, seaborn, sklearn.

Проводя исследование для поиска наиболее подходящих для анализа параметров, были выбраны частота процессора, дискретная видеокарта, оперативная память и цена. А в качестве основного метода был выбран метод визуализации корреляций.

Все собранные данные представлены в документе MS excel и выгружены в сервис облачного хранения файлов URL: <https://disk.yandex.ru/i/9CMLAHL-u4dORQ>

3 Результаты и обсуждения

Для начала построим модель методом визуализации корреляций. Для этого импортируем библиотеку pandas, далее загружаем подготовленные данные (датасет). После загрузки просматриваем небольшую часть данных и проверяем работу кода на небольшом подмножестве данных с помощью функции df.head (5) (рис.2).



```
[ ] import pandas as pd

[ ] df=pd.read_excel("dataset (1).xlsx")

df.head(5)
```

	Производитель	Модель	Частота Процессора Гц	Дискретная видеокарта ГБ	Оперативная память ГБ	SSD ГБ	ЭкранГц	Цена
0	Dell	G515-6217	2.5	4.0	8.0	256.0	120.0	65999.0
1	Lenovo	IdeaPad Gaming 3 15IMH05	2.5	4.0	8.0	512.0	60.0	69999.0
2	Acer	Aspire 7 A715-75G-52NW	2.5	4.0	8.0	256.0	60.0	72999.0
3	MSI	GF76 Katana 11SC-483XRU	2.7	4.0	8.0	256.0	144.0	77999.0
4	MSI	GF63 Thin 11UC-255XRU	2.7	4.0	8.0	256.0	144.0	79999.0

Рисунок 2 – Загрузка и проверка данных из датасета

Для построения эффективных моделей требуется провести детальный анализ структуры исходных данных. Этот этап включает исследование типов данных, оценку наличия пропусков, обнаружение возможных выбросов и выявление корреляционных связей между признаками. На основе проведенного анализа определяются переменные цена, производитель, модель, частота процессора, дискретная видеокарта, оперативная память, SSD, экран, которые будут использованы как входные параметры для модели, при этом целью является прогнозирование значения цены. Подобный тщательный предварительный анализ позволяет выбрать оптимальные алгоритмы машинного обучения и настроить модель так, чтобы обеспечить максимальную точность решения задачи прогнозирования (рис.3).

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 8 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Производитель                        205 non-null    object
1   Модель                              205 non-null    object
2   Частота Процессора ГЦ                205 non-null    float64
3   Дискретная видеокарта ГБ            206 non-null    float64
4   Оперативная память ГБ              205 non-null    float64
5   SSD ГБ                              205 non-null    float64
6   ЭкранГЦ                             205 non-null    float64
7   Цена                                205 non-null    float64
dtypes: float64(6), object(2)
memory usage: 64.0+ MB
```

Рисунок 3 – Структура данных

Далее были составлены гистограммы распределения данных показывают взаимосвязь между двумя переменными: цена, производитель, модель, частота процессора, дискретная видеокарта, оперативная память, SSD, экран (рис.4).



Рисунок 4 – Гистограммы для всех числовых столбцов

Для начала построим модель методом визуализации корреляций. График исходных данных и построенной модели, показывает зависимость частоты процессора к цене. Так как у нас лишь один массив данных, а для проверки работы модели нам нужно два, то с помощью библиотеки Scikit-learn (sklearn) и модуля `train_test_split` создадим две выборки: обучающую и тестовую. И уже после этого с помощью модуля `plt.title` построим модель диаграммы рассеивания по параметрам. В качестве переменной «х» – частота процессора, переменной «у» – цена (рис.5). Во втором случае за переменную «х» – оперативная память, переменной «у» – цена (рис.6). И в третьем случае в качестве «х» – дискретная видеокарта, переменной «у» – цена (рис.7).

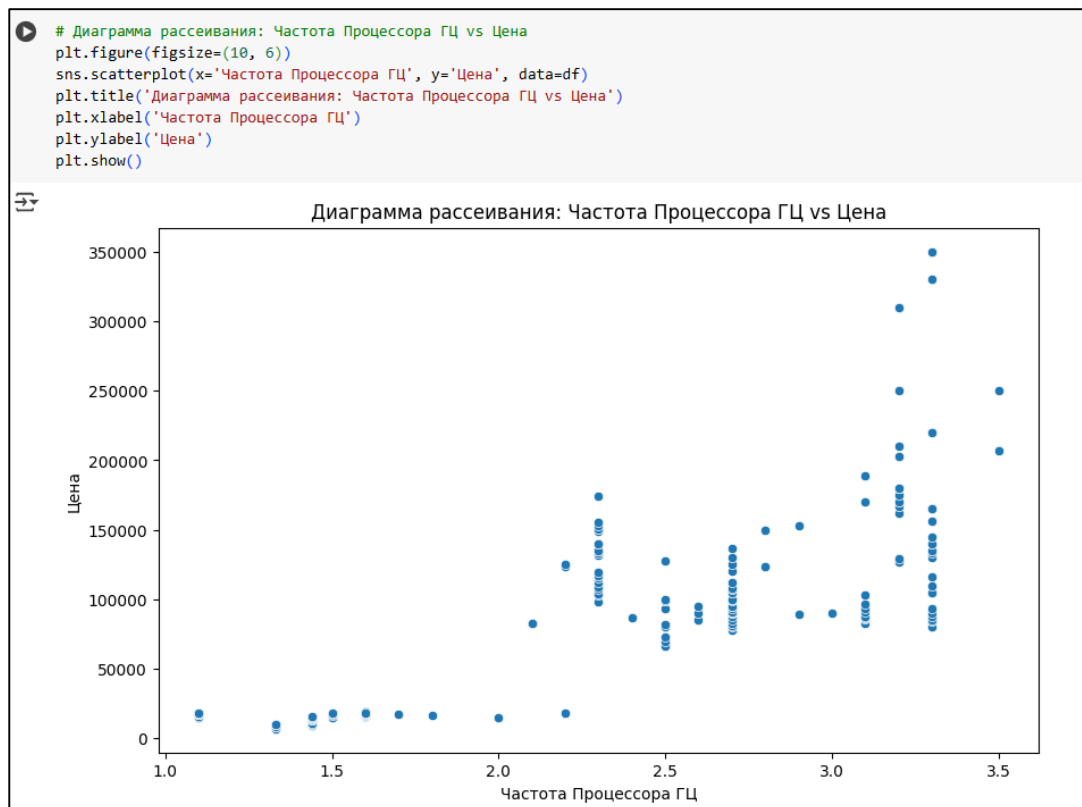


Рисунок 5 – Диаграмма рассеивания: Цена от Частоты процессора

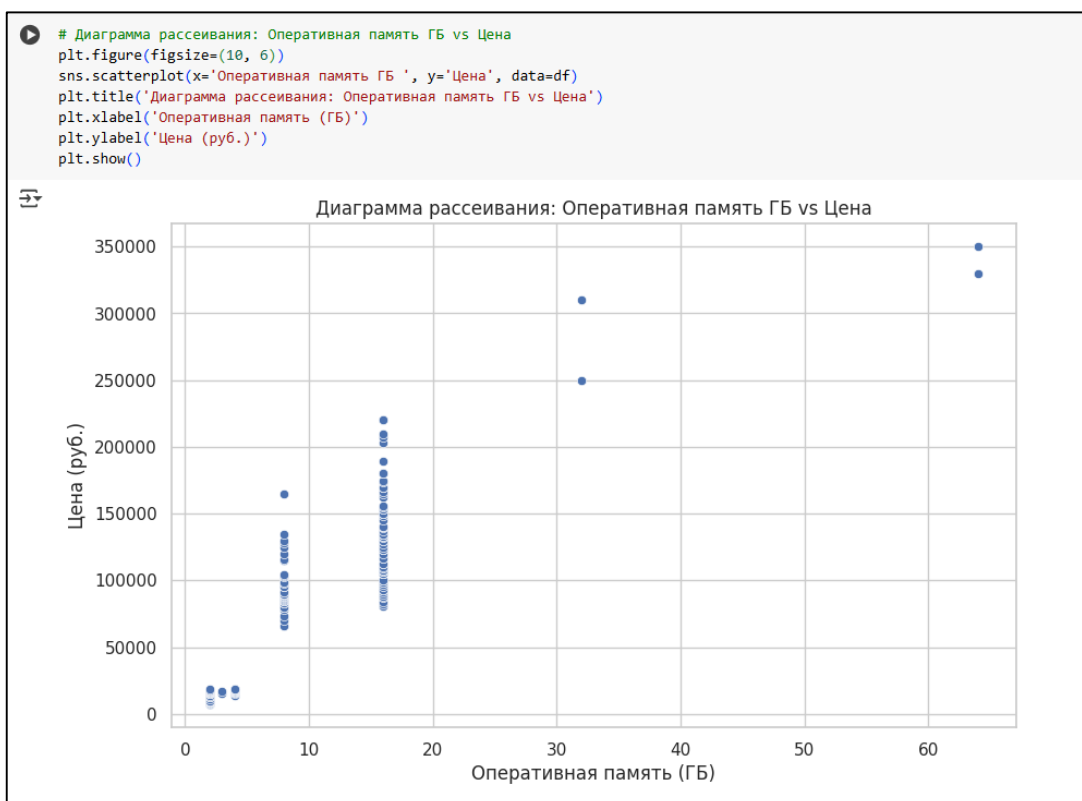


Рисунок 6 – Диаграмма рассеивания: Цена от Оперативной памяти



Рисунок 7 – Диаграмма рассеивания: Цена от Оперативной памяти

Полученные графики и есть модель визуализации корреляций. На рисунках прекрасно видно, что точки тестовой выборки (синие), то есть спрогнозированные цены, имеют положительный наклон. Это означает, что существует прямая зависимость, чем больше переменная «х», тем больше цена.

В качестве переменной «х» представлены характеристики мониторов: частота процессора, оперативная память и дискретная видеокарта. В качестве переменной «у» – цена.

Создадим список `numeric_columns`, содержащий названия столбцов, представляющих числовые данные. Данные столбцы могут содержать информацию о таких характеристиках как частота процессора, объем видеокарты, оперативной памяти, SSD и цена. Создадим новый датасет, который включает только те строки, где нет пропущенных значений в указанных числовых столбцах. Метод `dropna()` удаляет все строки, в которых отсутствует хотя бы одно значение. Определим целевую переменную «у» и признаки «х». «х» содержит все столбцы, кроме столбца 'Цена', который будет использоваться как целевая переменная. «у» содержит только значения из столбца 'Цена', который будет предсказываться на основе других признаков (рис. 8).

```
[ ] # Выбор только числовых столбцов
numeric_columns = [
    'Частота Процессора ГЦ',
    'Дискретная видеокарта ГБ',
    'Оперативная память ГБ ',
    'SSD ГБ',
    'ЭкранГЦ',
    'Цена'
]

# Удаление строк с пропущенными значениями
data = df[numeric_columns].dropna()

# Определение признаков и целевой переменной
X = data.drop('Цена', axis=1)
y = data['Цена']
```

Рисунок 8 – Выбор числовых столбцов

Чтобы проверить хорошая ли модель получилась, снова воспользуемся модулями библиотеки sklearn. Наиболее важным является `r2_score`, так как именно этот модуль поможет узнать коэффициент детерминации модели (R^2). Коэффициент детерминации для модели при правильном расчете принимает значения от 0 до 1. Для приемлемых моделей предполагается, что коэффициент детерминации должен быть хотя бы не меньше 50%. У полученной модели линейной регрессии коэффициент детерминации составляет 84,7%, что указывает на хорошую предсказательную способность модели (рис.9).

```
# Модель линейной регрессии
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
y_pred_lin = lin_reg.predict(X_test)

print("Линейная регрессия:")
print("MSE:", mean_squared_error(y_test, y_pred_lin))
print("R^2:", r2_score(y_test, y_pred_lin))
```

⇒ Линейная регрессия:
MSE: 432983491.07455456
R^2: 0.8471443502122131

Рисунок 9 – Расчет коэффициента детерминации

Теперь попробуем произвести расчет коэффициент детерминации для следующих моделей: линейная регрессия, дерево решений, случайный лес, KNN, AdaBoost, XGBoost (рис.10).

```
# Вывод таблицы с результатами
print(results_df)
```



	Модель	MSE	R^2
0	Линейная регрессия	4.329835e+08	0.847144
1	Дерево решений	2.765375e+08	0.902374
2	Случайный лес	3.105191e+08	0.890378
3	KNN	4.190077e+08	0.852078
4	AdaBoost	5.301114e+08	0.812855
5	XGBoost	2.431514e+08	0.914161

Рисунок 10 – Расчет коэффициента детерминации

Самой достоверной и хорошей из моделей получилась модель XGBoost, коэффициент детерминации которой равен 91,4%.

Теперь возьмем вместо параметра цена десятичный логарифм от этого параметра и проведем повторное исследование с зависимостью всех параметров от логарифма от цены (рис.11).

```
# Вывод таблицы с результатами
print(results_df)
```



	Модель	MSE	R^2
0	Линейная регрессия	4.329835e+08	0.847144
1	Дерево решений	4.014000e-03	0.979594
2	Случайный лес	4.484142e-03	0.977204
3	KNN	8.318069e-03	0.957714
4	AdaBoost	7.936271e-03	0.959654
5	XGBoost	3.643640e-03	0.981477

Рисунок 11 – Коэффициент детерминации модели с логарифмированной ценой

В данной модели самой достоверной и хорошей из моделей также получилась модель XGBoost, коэффициент детерминации которой равен 98,1%.

Представим результаты исследований в сравнительной таблице (табл. 1).

Таблица 1 – Сравнительная таблица расчета коэффициента детерминации

Модель	Коэффициента детерминации	Коэффициент детерминации модели с логарифмированной ценой
Линейная регрессия	0,847144	0,847144
Дерево решений	0,902374	0,979594
Случайный лес	0,890378	0,977204
KNN	0,852078	0,957714
AdaBoost	0,812855	0,959654
XGBoost	0,914161	0,981477

Исходя, из данных выше можно заметить, что после добавления десятичного логарифма от цены результаты параметра коэффициента детерминации изменились в лучшую сторону. Однако можно заметить, что модель линейной регрессии не изменилась.

Ссылка на работу в Google Colab: URL:
<https://colab.research.google.com/drive/1n-TfIK0SaaMWHr8pQ2UVq7F66nH5TqUF?usp=sharing>

4 Выводы

Таким образом, разработанные модели прогнозирования, реализованные в Google Colab, могут применяться для принятия решений о покупке и продаже мониторов, обеспечивая достаточную достоверность прогнозируемых данных. Результаты данного исследования могут найти применение в образовательных программах по методам машинного обучения в различных областях.

Библиографический список

1. Труженников Н.А., Сорокин С.А., Косникова О.В. Методы машинного обучения в анализе экономической конъюнктуры // Экономика и управление. 2023. № 3. С. 123–135. URL: <https://cyberleninka.ru/article/n/metody-mashinnogo-obucheniya-v-analize-ekonomicheskoy-konyunktury> (дата обращения: 27.05.2025).
2. Сенин С.С. Сравнение нескольких моделей методов машинного обучения // Компьютерные и информационные науки. 2023. № 2. С. 45–56. URL: <https://cyberleninka.ru/article/n/sravnenie-neskolkih-modeley-metodov-mashinnogo-obucheniya> (дата обращения: 27.05.2025).
3. Сенин С.С. Актуальные вопросы применения методов машинного обучения в экономике // Компьютерные и информационные науки. 2023. № 2. С. 45–56. URL: <https://cyberleninka.ru/article/n/aktualnye-voprosy-primeneniya-metodov-mashinnogo-obucheniya-v-ekonomike> (дата обращения: 27.05.2025).
4. Полторацкий П.Г. Прогнозирование спроса в электронной коммерции методами машинного обучения // Компьютерные и информационные науки. 2023. № 5. С. 89–100. URL: <https://cyberleninka.ru/article/n/prognozirovanie-sprosa-v-elektronnoy-kommertsii-metodami-mashinnogo-obucheniya> (дата обращения: 27.05.2025).
5. Новиков В.Д., Хамитов Р.М. Прогнозирование потребительской активности с использованием методов машинного обучения // Компьютерные и информационные науки. 2023. № 6. С. 101–112. URL: <https://cyberleninka.ru/article/n/prognozirovanie-potrebitelskoy-aktivnosti-s-ispolzovaniem-metodov-mashinnogo-obucheniya> (дата обращения: 27.05.2025).
6. Google Colaboratory / URL: <https://colab.research.google.com> (дата обращения: 24.05.2025).