

Применение методов машинного обучения для построения модели прогнозирования цены планшетов в облачной среде Google Colaboratory и в системе Orange

Андреева Альбина Андреевна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается применение методов машинного обучения для прогнозирования цен планшетных компьютеров. В рамках исследования использованы платформы Google Colaboratory и Orange, предоставляющие бесплатные инструменты для анализа данных. Исходные данные собраны с сайта DNS.ru и включали цену, диагональ, расширение экрана и другие. Применение методов линейной регрессии и метода ближайших соседей позволило достичь точности прогнозирования до 99,9% (по R-квадрату). Результаты исследования акцентируют внимание на важности внедрения современных технологий для автоматизации оценки цен, а также на возможностях их применения в электронной коммерции и бизнес-аналитике.

Ключевые слова: регрессионный анализ, метод ближайших соседей, линейная регрессия, Orange, Python, Google Colaboratory, прогнозирование цен.

Application of machine learning methods to build a model for predicting the price of tablet computers in the Google Colaboratory cloud environment and in the Orange system

Andreeva Albina Andreevna

Sholom-Aleichem Priamursky State University

Student

Abstract

The article examines the application of machine learning methods for predicting the prices of tablet computers. The study used Google Colaboratory and Orange platforms, which provide free tools for data analysis. The initial dataset was collected from the DNS.ru website and included such parameters as price, diagonal, screen resolution, and others. The use of linear regression and k-nearest neighbors method allowed us to achieve a prediction accuracy of up to 99.9% (R-square). The results of the study highlight the importance of implementing modern technologies to automate pricing assessments and their potential applications in e-commerce and business analytics.

Keywords: regression analysis, nearest neighbor method, linear regression, Orange, Python, Google Colaboratory, price forecasting.

1. Введение

1.1 Актуальность

Машинное обучение становится все более важным инструментом в анализе данных и прогнозировании цен. Основная значимость данной статьи заключается в ее практическом применении: использование регрессионного анализа и методов машинного обучения позволяет автоматизировать процесс оценки цен, что может быть особенно полезно в сфере розничной торговли и электронной коммерции. Примеры использования платформ Google Colaboratory и Orange демонстрируют, что существующие инструменты способны обеспечить необходимую точность прогнозирования для коммерческого использования.

1.2 Обзор исследований

В процессе подготовки статьи были проанализированы работы, посвящённые различным аспектам машинного обучения, статистического анализа и методов прогнозирования. Так, П.А. Абаньшин провёл сравнительный анализ методов автоматизированного прогнозирования цен, что позволило учесть сильные и слабые стороны различных подходов при построении собственных моделей [1].

Работы А.С. Боброва и Р.И. Баженова показывают успешное применение методов машинного обучения в прогнозировании цен автомобилей, что подтверждает актуальность подобных исследований для различных сфер, включая аналитику бытовой техники [2].

В работе "Использование методов машинного обучения для решения задач прогнозирования суммы и вероятности покупки на основе данных электронной коммерции" О.А. Мамиева, Н.А. Финогенова и Сологуба Г.Б. рассматриваются методы классификации и регрессии для прогнозирования покупок, что может быть адаптировано для цен на планшеты в облачных средах [6].

В работе "Применение методов машинного обучения для задач прогнозирования вероятности невыкупа" Ю.Л. Змеева рассматриваются различные алгоритмы, включая логистическую регрессию и случайный лес, с акцентом на их точность и эффективность. Также проводятся анализ и отбор признаков, влияющих на точность моделей, с целью оптимизации прогнозирования невыкупа заказов [4].

В исследовании "Искусственный интеллект в прогнозировании фондового рынка" рассматриваются различные алгоритмы машинного обучения, которые могут быть применены для анализа ценовых трендов [5].

Наконец, в учебном пособии Ю.Е. Воскобойникова демонстрирует практическое применение регрессионного анализа для обработки экспериментальных данных. Этот материал помог уточнить выбор параметров и методов для построения моделей [3].

1.3 Цель исследования

Целью исследования является построения моделей прогнозирования цены компьютерных планшетов марок Xiaomi, Lenovo, HUAWEI, Samsung, Apple и другие на основе регрессионного анализа.

2. Материалы и методы

Для достижения поставленной цели идеально подходит бесплатная облачная платформа Google Colab, которая специально создана для написания программного кода с использованием многочисленных библиотек машинного обучения. Кроме того, можно воспользоваться бесплатной средой Orange, разработанной на языке программирования Python. Весь процесс будет основан на готовых наборах данных, содержащих информацию о ценах и характеристиках планшетов: URL: https://docs.google.com/spreadsheets/d/1IDiXDZYbwBbTNuh25Ge7Tciu9z_DS72a/edit?usp=drive_link&ouid=111118899553537318197&rtpof=true&sd=true

3. Результаты и обсуждения

Сбор данных осуществлялся по данным сайта www.DNS.ru. Рассматривались объявления о продаже компьютерных планшетов марок Xiaomi, Lenovo, HUAWEI, Samsung, Apple и другие. Данные объявлений собирались по нескольким параметрам, а именно: цена, диагональ, разрешение экрана, плотность пикселей, частота обновления экрана, количество ядер, объём оперативной памяти, объём встроенной памяти, гарантийный срок, ёмкость аккумулятора, $\log(\text{цена})$. Все было собрано в файл MS Excel (см.рис. 1).

	A	B	C	D	E	F	G	H	I	J	K	L
1	Модель	Цена,руб	Диагональ	Разрешение экрана	Плотность	Частота о	Количество	Объем оп	Объем вс	Гарантия	Емкость а	LOG10(Цена)
2	KENSHI E38	5999	8	1280x800	188	60	4	2	32	12	4000	3,778079
3	KENSHI E28 LTE	7499	8	1280x800	188	60	4	3	64	12	4000	3,875003
4	KENSHI Kids H20 LTE	7499	8	1280x800	238	60	4	3	64	12	4000	3,875003
5	KENSHI Kids H19 LTE	7499	8	1280x800	213	60	4	3	64	12	4000	3,875003
6	Digma CITI 1314C LTE	7599	10,1	1280x800	149	60	8	3	32	12	6000	3,880756
7	DEXP Ursus K41 LTE	7799	10,1	1920x1200	224	60	8	2	32	12	5000	3,892039
8	KENSHI Kids H44 LTE	7999	8	1280x800	213	60	8	4	64	12	5000	3,903036
9	KENSHI Kids H34 LTE	7999	8	1280x800	213	60	8	4	64	12	5000	3,903036
10	KENSHI Kids E18 LTE	8499	10,1	1280x800	149	60	4	3	64	12	5000	3,929368
11	KENSHI Kids E17 LTE	8499	10,1	1280x800	149	60	4	3	64	12	5000	3,929368
12	KENSHI H38 LTE	8999	8	1920x1200	283	60	4	3	64	12	4000	3,954194
13	KENSHI H10 LTE	9499	10,1	1280x800	149	60	8	4	64	12	6000	3,977678
14	Digma CITI 1313C LTE	9499	10,1	1280x800	149	60	8	3	32	12	6000	3,977678
15	KENSHI K10	10499	10,36	2000x1200	225	50	8	4	64	12	6000	4,021148
16	KENSHI H10 LTE	9499	10,1	1280x800	149	60	8	4	64	12	6000	3,977678
17	KENSHI KP10 LTE	10499	10,36	2000x1200	225	60	8	4	64	12	6000	4,021148
18	KENSHI H11 LTE	11499	10,1	1280x800	149	60	8	4	64	12	6000	4,06066
19	Xiaomi Redmi Pad SE 8.7	11999	8,7	1340x800	179	90	8	4	64	12	6650	4,079145
20	INOI inoIPad Pro LTE	12299	10,1	1920x1080	224	60	8	4	128	12	6000	4,08987
21	DEXP Ursus D21	12399	10,36	2000x1200	224	60	8	4	128	12	6000	4,093387
22	KENSHI KP11 LTE	12999	10,36	2000x1200	225	60	8	6	128	12	6000	4,11391
23	OPPO Pad Air Wi-Fi	12999	10,36	2000x1200	225	60	8	4	64	12	7100	4,11391
24	Lenovo Tab M10 HD (2nd G	13199	10,1	1280x800	149	60	8	4	64	12	5000	4,120541
25	Teclast P40HD LTE	13499	10,1	1920x1200	224	60	8	6	128	12	6000	4,130302
26	Xiaomi Redmi Pad SE 8.7 LT	13999	8,7	1340x800	179	90	8	4	64	12	6650	4,146097
27	KENSHI E10 LTE	13999	10,36	2000x1200	215	60	8	6	128	12	6200	4,146097
28	Digma Pro HIT 18 LTE	14399	10,51	1920x1200	215	60	8	8	128	12	7000	4,158332
29	HUAWEI MatePad SE 11 Wi-	14499	11	1920x1200	207	60	8	4	128	12	7700	4,161338
30	BlackView Tab 60 LTE	14499	8,68	1340x800	180	60	8	6	128	12	6050	4,161338
31	Lenovo Tab M10 Plus (3rd G	14599	10,6	2000x1200	220	60	8	4	64	12	7500	4,164323
32	KENSHI Armor O8 LTE	14699	8	1280x800	188	60	8	4	64	12	4860	4,167288
33	Digma Pro HIT 16 LTE	14899	10,4	2000x1200	224	60	8	8	256	12	7000	4,173157
34	KENSHI E11 LTE	14999	10,95	2000x1200	213	60	8	8	128	12	7200	4,176062
35	Samsung Galaxy Tab A9 Wi-	14999	8,7	1340x800	179	60	8	4	64	12	5100	4,176062
36	Teclast T40 Pro 2023 LTE	14999	10,4	2000x1200	224	60	8	8	128	12	7000	4,176062
37	Blackview Tab 12 LTE	14999	10,1	1920x1200	224	60	8	4	64	12	6580	4,176062
38	EdPad Wi-Fi	14999	11	1920x1200	207	90	8	4	64	12	7040	4,176062
39	HONOR Pad X8a Wi-Fi	14999	11	1920x1200	207	90	8	4	64	12	8300	4,176062
40	Xiaomi Redmi Pad SE 8.7 LT	14999	8,7	1340x800	179	90	8	4	128	12	6650	4,176062
41	Blackview Tab 11 SE LTE	15299	10,36	2000x1200	225	60	8	8	128	12	7680	4,184663

Рисунок 1 – Фрагмент таблицы с данными MS Excel

Из таблицы видно, что все модели имеют довольно схожие характеристики, но разброс цены достаточно велик. Для создания регрессионных моделей воспользуемся облачной средой Google Coollaboratory, а именно встроенными библиотеками pandas openpyxl scikit-learn matplotlib.

Проведя исследование для поиска наиболее подходящих для анализа параметров, были выбраны диагональ экрана и количество ядер процессора. А в качестве основных методов были выбраны метод линейной регрессии (linear regression) и метод ближайших соседей (kNN).

Для начала будет построена модель методом линейной регрессии. Так как у нас лишь один массив данных, а для проверки работы модели нужно два, то с помощью библиотек pandas, scikit-learn, matplotlib и модуля train_test_split создадим две выборки: обучающую и тестовую. И уже после этого с помощью модуля LinearRegression построим модель линейной регрессии.

Полученный график и есть модель линейной регрессии. На графике прекрасно видно, что точки тестовой выборки (зеленые), то есть спрогнозированные цены, находятся наиболее близко к линии.

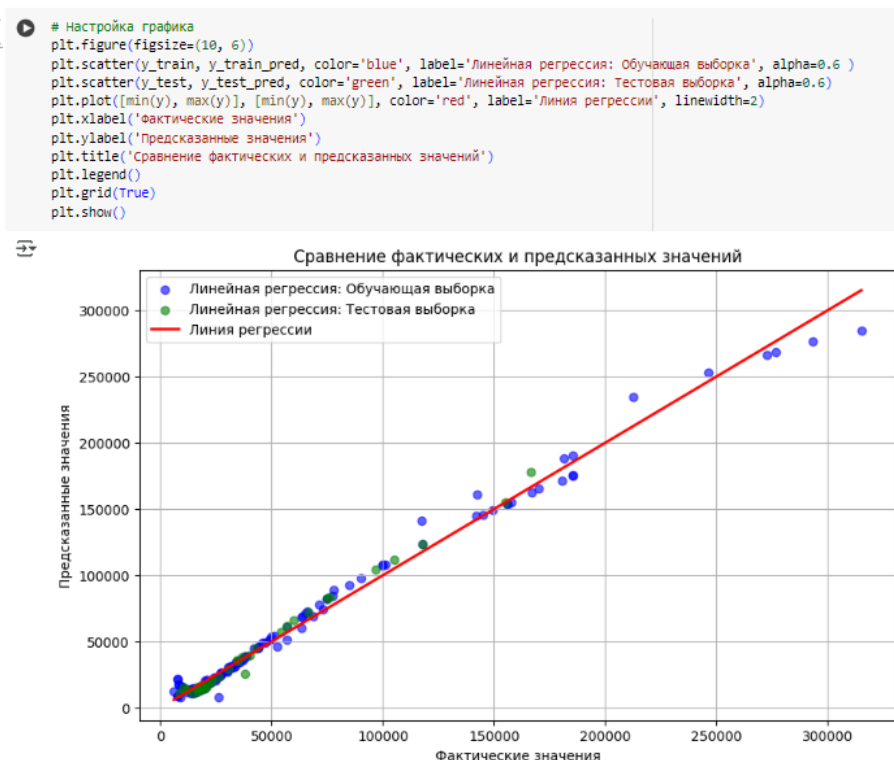


Рисунок 2 – Линейная регрессия

Полученный график и есть модель линейной регрессии. На графике видно, что точки тестовой выборки (зеленые), то есть спрогнозированные цены, находятся наиболее близко к линии.

Чтобы проверить хорошая ли модель получилась, снова воспользуемся модулями библиотеки pandas, scikit-learn, matplotlib. 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 99 %, что говорит о том, что модель хорошая (рис.3).

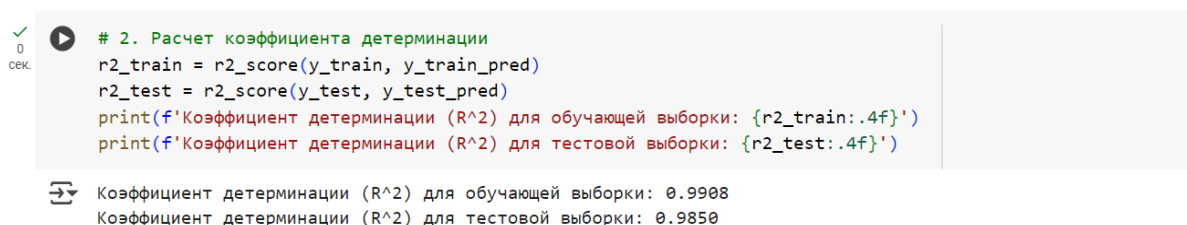
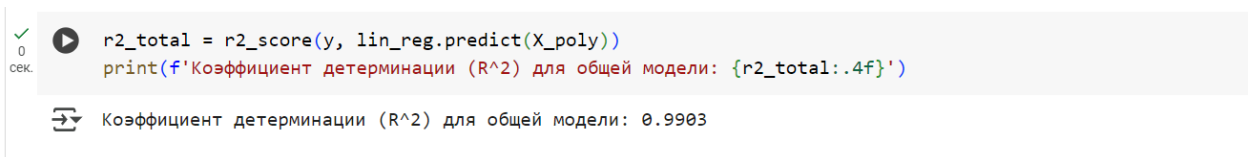


Рисунок 3 – Расчет коэффициента детерминации

После попробуем создать модель, используя все параметры: цена, диагональ, разрешение экрана, плотность пикселей, частота обновления экрана, количество ядер, объём оперативной памяти, объём встроенной памяти, гарантийный срок, ёмкость аккумулятора, $\log(\text{цена})$. (рис.4).



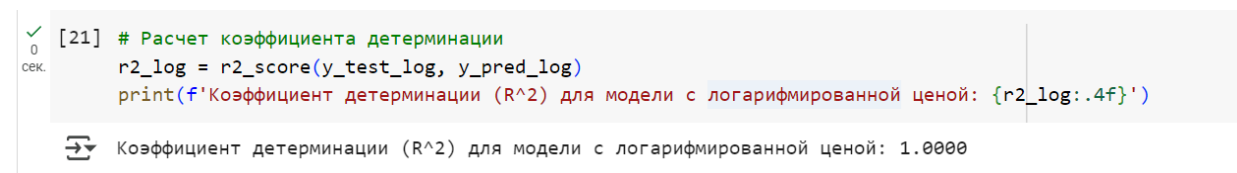
```
0 сек. r2_total = r2_score(y, lin_reg.predict(X_poly))
print(f'Коэффициент детерминации (R^2) для общей модели: {r2_total:.4f}')
```

⇒ Коэффициент детерминации (R^2) для общей модели: 0.9903

Рисунок 4 – расчет коэффициента детерминации общей модели линейной регрессии

На рисунке (рис.4) видно, что в данном случае модель осталась равной прошлому значению, так как коэффициента детерминации общей модели линейной регрессии равен 99 %.

Рассмотрим коэффициент детерминации модели с логарифмированной ценой. Возьмем вместо параметра Цена десятичный логарифм от этого параметра и проведем повторное исследование с зависимостью всех параметров от логарифма от цены.



```
0 сек. [21] # Расчет коэффициента детерминации
r2_log = r2_score(y_test_log, y_pred_log)
print(f'Коэффициент детерминации (R^2) для модели с логарифмированной ценой: {r2_log:.4f}')
```

⇒ Коэффициент детерминации (R^2) для модели с логарифмированной ценой: 1.0000

Рисунок 5 – Коэффициент детерминации модели с логарифмированной ценой

Модель получилась увеличилась на один процент, коэффициент детерминации теперь равен 100 %. (рис.5).

Также для прогнозирования цены используется модель созданная методом ближайших соседей (kNN). Для создания модели нам понадобятся параметры разрешение экрана, плотность пикселей, частота обновления экрана, количество ядер, объем оперативной памяти и цена.

С помощью библиотеки pandas, scikit-learn, matplotlib и модуля KNeighborsRegressor, а также используя все те же обучающую и тестовую выборки, создаем нужную модель, выводя как итог процент верно предсказанных вариантов.

```

✓ [22] from sklearn.neighbors import KNeighborsRegressor
0 сек.

✓ [30] knn_3 = KNeighborsRegressor(n_neighbors=3)
0 сек.
      knn_3.fit(X_train, y_train)
      y_pred_knn_3 = knn_3.predict(X_test)

✓ [24] # 6 соседей
0 сек.
      knn_6 = KNeighborsRegressor(n_neighbors=6)
      knn_6.fit(X_train, y_train)
      y_pred_knn_6 = knn_6.predict(X_test)

✓ [25] # Расчет R^2 для метода ближайших соседей
0 сек.
      r2_knn_3 = r2_score(y_test, y_pred_knn_3)
      r2_knn_6 = r2_score(y_test, y_pred_knn_6)

✓ [26] r2_knn_6 = r2_score(y_test, y_pred_knn_6)
0 сек.

✓ [27] print(f'Коэффициент детерминации (R^2) для метода ближайших соседей (3 соседа): {r2_knn_3:.4f}')
0 сек.
      print(f'Коэффициент детерминации (R^2) для метода ближайших соседей (6 соседей): {r2_knn_6:.4f}')

⇒ Коэффициент детерминации (R^2) для метода ближайших соседей (3 соседа): 0.9809

```

Рисунок 6 – Метод ближайших соседей (3 соседа)

На рисунке видно, что процент верно предсказанных вариантов составляет 98,09 что является отличным показателем. Сделаем анализ со всеми параметрами, как и в прошлом методе (рис.6).

```

✓ [28] # 5. Метод ближайших соседей с логарифмированной ценой
0 сек.
      knn_6_log = KNeighborsRegressor(n_neighbors=6)
      knn_6_log.fit(X_train, y_train_log)
      y_pred_knn_6_log = knn_6_log.predict(X_test)

✓ [29] # Расчет R^2
0 сек.
      r2_knn_6_log = r2_score(y_test_log, y_pred_knn_6_log)
      print(f'Коэффициент детерминации (R^2) для метода ближайших соседей (6 соседей, логарифм цены): {r2_knn_6_log:.4f}')

⇒ Коэффициент детерминации (R^2) для метода ближайших соседей (6 соседей, логарифм цены): 0.9820

```

Рисунок 7 – Метод ближайших соседей (6 соседей)

В данном случае модель осталась практически такой же. Добавим снова десятичный логарифм от цены, чтобы сравнить результат (рис.7).

```

[ ] # 5. Метод ближайших соседей с логарифмированной ценой
      knn_6_log = KNeighborsRegressor(n_neighbors=6)
      knn_6_log.fit(X_train, y_train_log)
      y_pred_knn_6_log = knn_6_log.predict(X_test)

```

Рисунок 8 - Метод ближайших соседей (6 соседей)

Модель достигла результата 98,2%, что свидетельствует о её надёжности. Вероятно, это стало возможным благодаря оптимальному выбору методов обработки данных, таких как логарифмическое преобразование, которое идеально подошло для данного типа данных.

Похожий анализ также был проведен с помощью среды Orange, но с использованием большим количеством регрессионных моделей.

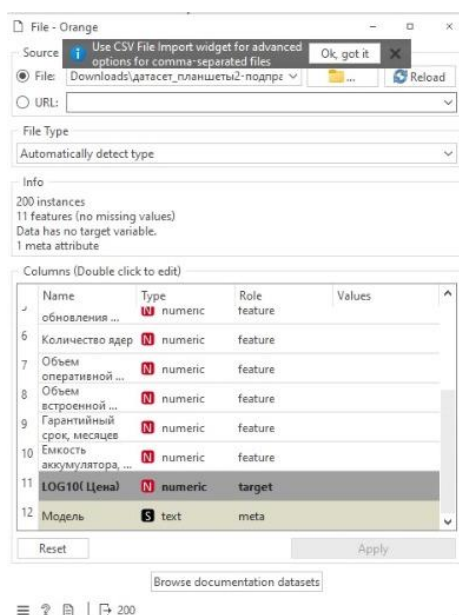


Рисунок 9 – Виджет файл

Внеся собранные данные в Orange, применив кроссвалидацию с 5 фолдами, а также добавив ещё несколько объектов: Distributions, Correlations, Scatter Plot. Свяжем их с Test and Score (рис. 10). Уже от Test and Score проведем к нужным моделям, выставив верные параметры. В качестве моделей для прогнозирования выбраны Random Forest, Gradient Boosting, kNN, AdaBoost и Linear Regression.

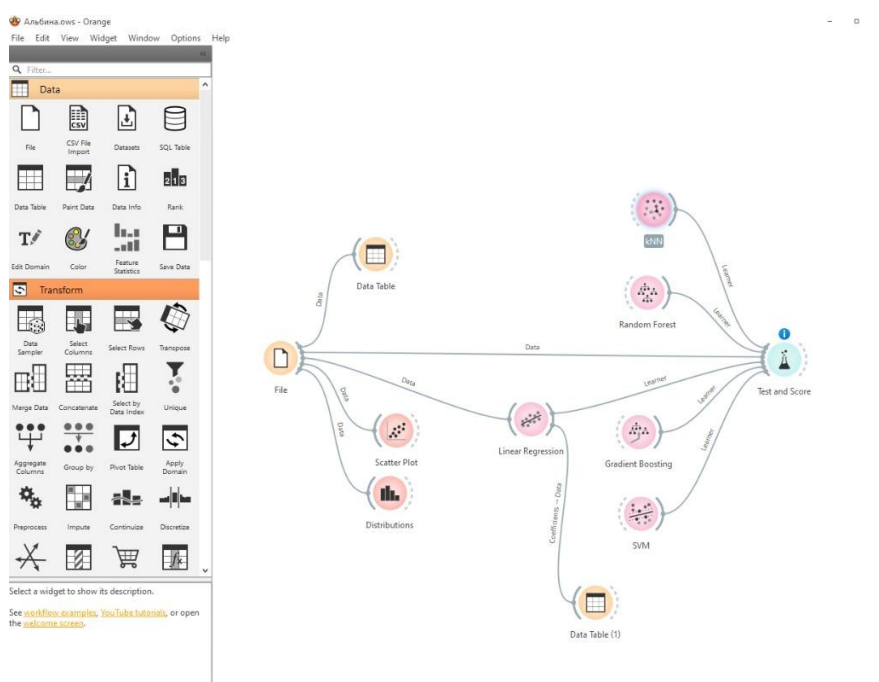


Рисунок 10 - Полная модель

Программная среда Orange самостоятельно проводит анализ, на основе внесенных данных и выставленных параметров. Результат проведенного анализа выводится в Test and Score в виде коэффициента детерминации, из которого уже можно сделать вывод о качестве каждой модели (рис.10).

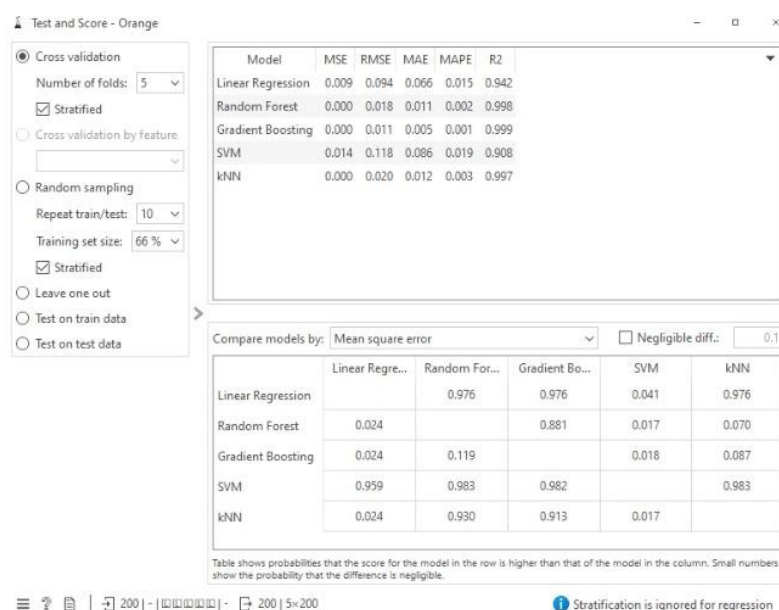


Рисунок 11 - Результат в Test and Score

Самой достоверной и хорошей из моделей получилась модель градиентного бустинга, значение которого 0,999.

4 Выводы

Таким образом, модели прогнозирования разработанные в Google Colaboratory и Orange, могут использоваться для оценки стоимости планшетных компьютеров разных брендов, подтверждая свою практическую значимость. Достигнутые результаты обладают достаточной точностью для использования этих моделей в сфере электронной коммерции и бизнес-анализа. Кроме того, проведенное исследование представляет ценность в образовательном контексте для освоения методов машинного обучения и их применения при работе с данными.

Библиографический список

1. Абаньшин П. А. Сравнительный анализ методов автоматизированного прогнозирования изменения цен акций на фондовом рынке // Вестник науки и образования. 2020. №12. С. 55–63.
2. Бобров А. С., Баженов Р. И. Применение методов машинного обучения для построения модели прогнозирования цены подержанных автомобилей Audi Q7 в системе Orange // Сборник трудов XIII Всероссийской школы-семинара молодых ученых, аспирантов, студентов и школьников. Ростов-на-Дону – Таганрог, 2022. С. 112–119.
3. Воскобойникова Ю. Е. Регрессионный анализ данных в пакете MATHCAD

- + CD: учебное пособие. СПб.: Лань, 2021. 224 с.
4. Змеева Ю. Л.. Применение методов машинного обучения для задач прогнозирования вероятности невыкупа. URL: <https://na-journal.ru/6-2022-informacionnye-tehnologii/3941-primenenie-metodov-mashinnogo-obucheniya-dlya-zadach-prognozirovaniya-veroyatnosti-nevykupa> 2.
 5. Искусственный интеллект в прогнозировании фондового рынка. URL: https://www.cesurg.ru/ru/jarticles_cesurg/677.html?SSr=210134b36816ffffff27c__07e70a1010152e-5541 8.
 6. Мамиев О. А., Финогенов Н. А., Сологуб Г. Б. Использование методов машинного обучения для решения задач прогнозирования суммы и вероятности покупки на основе данных электронной коммерции //Моделирование и анализ данных. 2020. Т. 10. №. 4. С. 31-40.
 7. Google Colaboratory. - URL: https://colab.research.google.com/drive/1jHWcbBRDFjkWKzNzMAaUbP_Rrt2FnxXK?usp=sharing
 8. Данные. - URL: https://docs.google.com/spreadsheets/d/1IDiXDZYbwBbTNuh25Ge7Tciu9z_DS72a/edit?usp=drive_link&ouid=111118899553537318197&rtpof=true&sd=true
 9. Orange. - URL: https://drive.google.com/file/d/1eaafWLPMJ7_AA49dRoiklfijUHIgeAdW/view?usp=sharing