

Прогнозирование рыночной цены телевизоров с использованием методов машинного обучения в Google Colab и Orange

Евдокименко Евгения Владимировна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается использование регрессионного анализа для прогнозирования рыночной стоимости телевизоров. Для проведения исследования использовались бесплатные инструменты: среда Orange, написанная на языке Python, и облачный сервис Google Colab. В качестве исходных данных были собраны характеристики телевизоров различных брендов по данным сайта www.dns-shop.ru. В результате исследования, путём создания моделей, удалось достичь точности прогнозирования стоимости покупки телевизора до 86,4% по показателю R-квадрата.

Ключевые слова: регрессионный анализ, метод ближайших соседей, линейная регрессия, Gradient Boosting, Orange, Python, Google Colab.

Predicting the market price of televisions using machine learning methods in Google Colab and Orange

Evdokimenko Evgeniya Vladimirovna

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the use of regression analysis to predict the market value of televisions. Free tools were used to conduct the research: the Orange environment, written in Python, and the Google Colab cloud service. As initial data, the characteristics of TVs 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 TV up to 86,4% in terms of the R-square indicator.

Keywords: Regression Analysis, nearest neighbor method, linear regression, Gradient Boosting, Orange, Python, Google Colab.

1 Введение

1.1 Актуальность

В последнее время большую популярность набирают методы машинного обучения. Одним их основных методом является использования данных технологий для прогнозирования цен на различные товары и услуги.

Данный метод позволяет оптимизировать решения участников рынка относительно покупок и продаж.

Для точного прогнозирования необходимо собрать большое количество данных, характеризующие товар, среди которых важнейшим является текущая рыночная цена. Основываясь на этих данных, можно построить различные модели, для дальнейшего прогнозирования цены. Благодаря регрессионному анализу, можно выявить и проанализировать зависимости между ценной и факторами.

1.2 Обзор исследований

В статье «Методы машинного обучения» автор Д. Сорокин рассматривает ключевые концепции и подходы в области машинного обучения. Основное внимание уделяется базовым понятиям. Кроме того, подробно описываются этапы подготовки данных [1].

В статье А.М. Гржибовского рассматривается применение линейного регрессионного анализа для ситуаций с одной зависимой и одной независимой переменной, с акцентом на условия применимости и общие принципы расчета значений [2].

Статья А. Вичугова посвящена принципам работы деревьев решений (Decision Tree) и их значимости в обработке больших данных (Big Data). Автор объясняет структуру и преимущества данного метода, а также обсуждает его роль в анализе больших объемов информации [3].

В статье А.М. Бабаева и М.А. Шемякиной рассматриваются традиционные подходы к машинному обучению, применяемые для классификации данных. Авторы анализируют такие методы, как логистическая регрессия, метод k-ближайших соседей и деревья решений, оценивая их эффективность и ограничения в различных сценариях [4].

Статья К.Е. Топухова, А.Ю. Романова и В.И. Лукьянова посвящена анализу различных методов машинного обучения, применяемых для автоматической классификации научных документов. Авторы рассматривают эффективность и особенности использования таких методов, как логистическая регрессия, метод опорных векторов (SVM), деревья решений и нейронные сети, в контексте задачи классификации текстов [5].

1.3 Цель исследования

Целью исследования является построение моделей прогнозирования цены телевизоров 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. А также бесплатная среда Orange, написанная на языке программирования Python.

В статье собраны данные за декабрь 2024 и январь 2025 гг. Все данные были собраны с сайта www.dns-shop.ru. В работу были взяты модели

телевизоров последних лет выпуска. Данные о товарах собирались по следующим параметрам: цена, год выпуска, диагональ экрана, мощность звука, частота обновления экрана, объем оперативной памяти, толщина без подставки, максимальная потребляемая мощность. Вся информация была собрана в файл формата MS Excel. (рис.1).

	Год	Диагональ экрана (дюйм)	Мощность звука (Вт)	Частота обновления экрана (Гц)	Объем оперативной памяти (Гб)	Толщина без подставки (мм)	Максимальная потребляемая мощность (Вт)	Цена (руб.)
2	2024	55	20	60	2	82	120	44999
3	2024	55	30	120	3	70	190	84999
4	2024	65	20	60	2	83	150	59999
5	2024	65	30	120	3	70	200	99999
6	2024	50	20	60	2	82	95	34999
7	2024	43	16	60	2	81	80	30999
8	2024	32	20	60	1,5	72	50	15999
9	2024	75	30	120	4	73	260	149999
10	2022	43	24	60	2	81,3	65	23999
11	2024	55	20	60	2	69,5	100	49999
12	2024	55	20	60	2	69,5	70	39999
13	2024	75	20	60	2	85	165	84999
14	2024	43	30	60	2	72	90	34999
15	2022	86	30	120	4	91,3	310	129999

Рисунок 1 – Фрагмент таблицы с данными MS Excel

Из таблицы рисунка 1 видно, что у многих моделей телевизоров одинаковый объем оперативной памяти, однако цена на них значительно варьируется. Для создания регрессионных моделей воспользуемся облачной средой Google Colab, а именно встроенными библиотеками pandas, seaborn, sklearn.

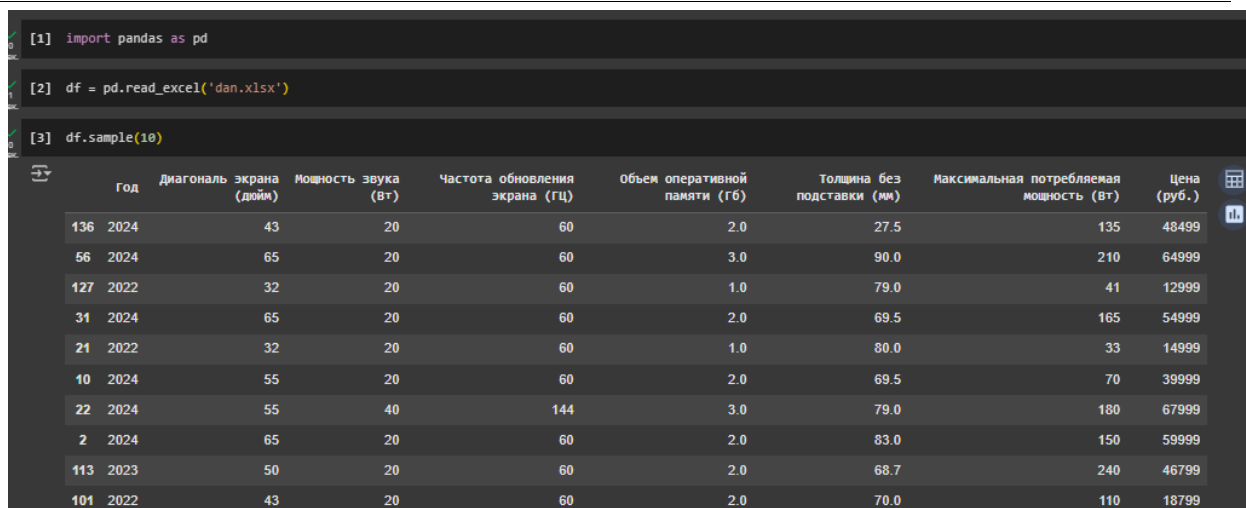
Проводя исследование для поиска наиболее подходящих для анализа параметров, были выбраны максимальная потребляемая мощность, мощность звука, цена. А в качестве основного метода был выбран метод линейной регрессии (linear regression).

Все собранные данные представлены в документе MS excel и выгружены в сервис облачного хранения файлов.

URL: <https://disk.yandex.ru/i/dn4j4ACA0YN0oQ>

3 Результаты и обсуждения

Для начала построим модель методом линейной регрессии. 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.sample (10) (рис.2).

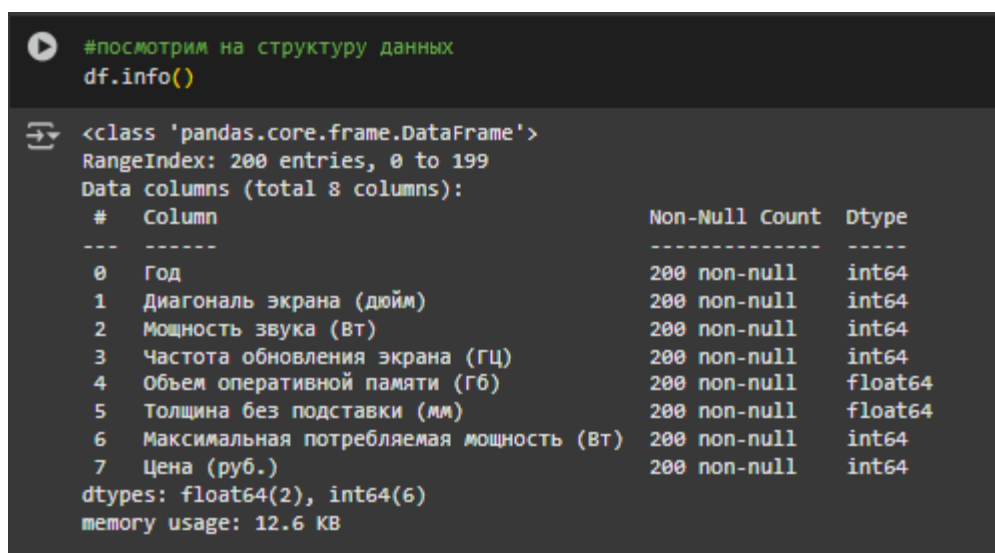


```
[1] import pandas as pd
[2] df = pd.read_excel('dan.xlsx')
[3] df.sample(10)
```

	Год	диагональ экрана (дюйм)	Мощность звука (Вт)	Частота обновления экрана (Гц)	Объем оперативной памяти (Гб)	Толщина без подставки (мм)	Максимальная потребляемая мощность (Вт)	Цена (руб.)
136	2024	43	20	60	2.0	27.5	135	48499
56	2024	65	20	60	3.0	90.0	210	64999
127	2022	32	20	60	1.0	79.0	41	12999
31	2024	65	20	60	2.0	69.5	165	54999
21	2022	32	20	60	1.0	80.0	33	14999
10	2024	55	20	60	2.0	69.5	70	39999
22	2024	55	40	144	3.0	79.0	180	67999
2	2024	65	20	60	2.0	83.0	150	59999
113	2023	50	20	60	2.0	68.7	240	46799
101	2022	43	20	60	2.0	70.0	110	18799

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 мощность звука, максимальная потребляемая мощность и цена, 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).



```
#посмотрим на структуру данных
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 8 columns):
 #   Column                                     Non-Null Count  Dtype
---  -
 0   Год                                       200 non-null    int64
 1   Диагональ экрана (дюйм)                 200 non-null    int64
 2   Мощность звука (Вт)                     200 non-null    int64
 3   Частота обновления экрана (Гц)          200 non-null    int64
 4   Объем оперативной памяти (Гб)           200 non-null    float64
 5   Толщина без подставки (мм)              200 non-null    float64
 6   Максимальная потребляемая мощность (Вт) 200 non-null    int64
 7   Цена (руб.)                             200 non-null    int64
dtypes: float64(2), int64(6)
memory usage: 12.6 KB
```

Рисунок 3 – Структура данных

Далее были составлены гистограммы рассеивания данных показывают взаимосвязь между двумя переменными: количественными и каким-либо другим параметром: мощность звука, максимальная потребляемая мощность и цена (рис.4, рис.5, рис.6).

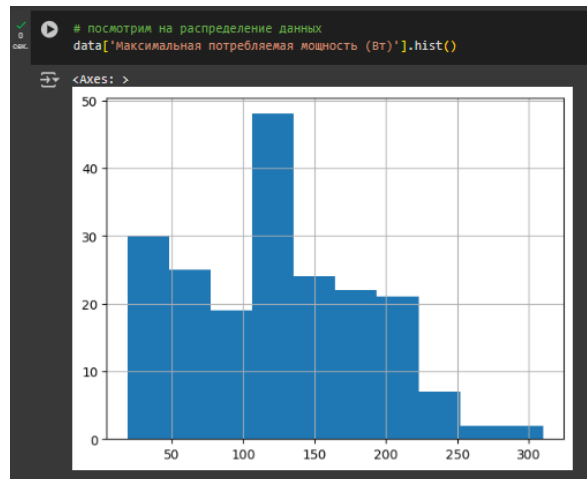


Рисунок 4 – Гистограмма рассеивания

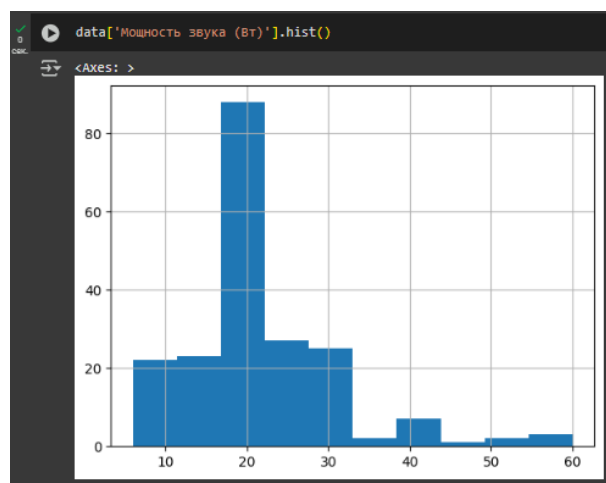


Рисунок 5 – Гистограмма рассеивания

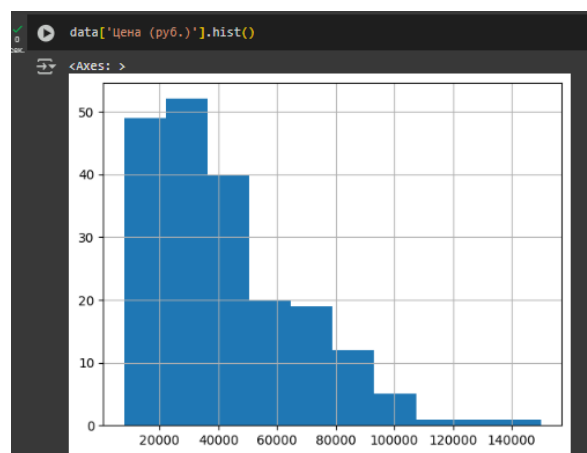


Рисунок 6 – Гистограмма рассеивания

Для начала построим модель методом линейной регрессии. График исходных данных и построенной модели, показывает зависимость максимальной потребляемой мощности 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 Linear Regression построим модель линейной регрессии по параметрам. В качестве переменной «х» – максимальная потребляемая мощность, переменной «у» – цена (рис.7).

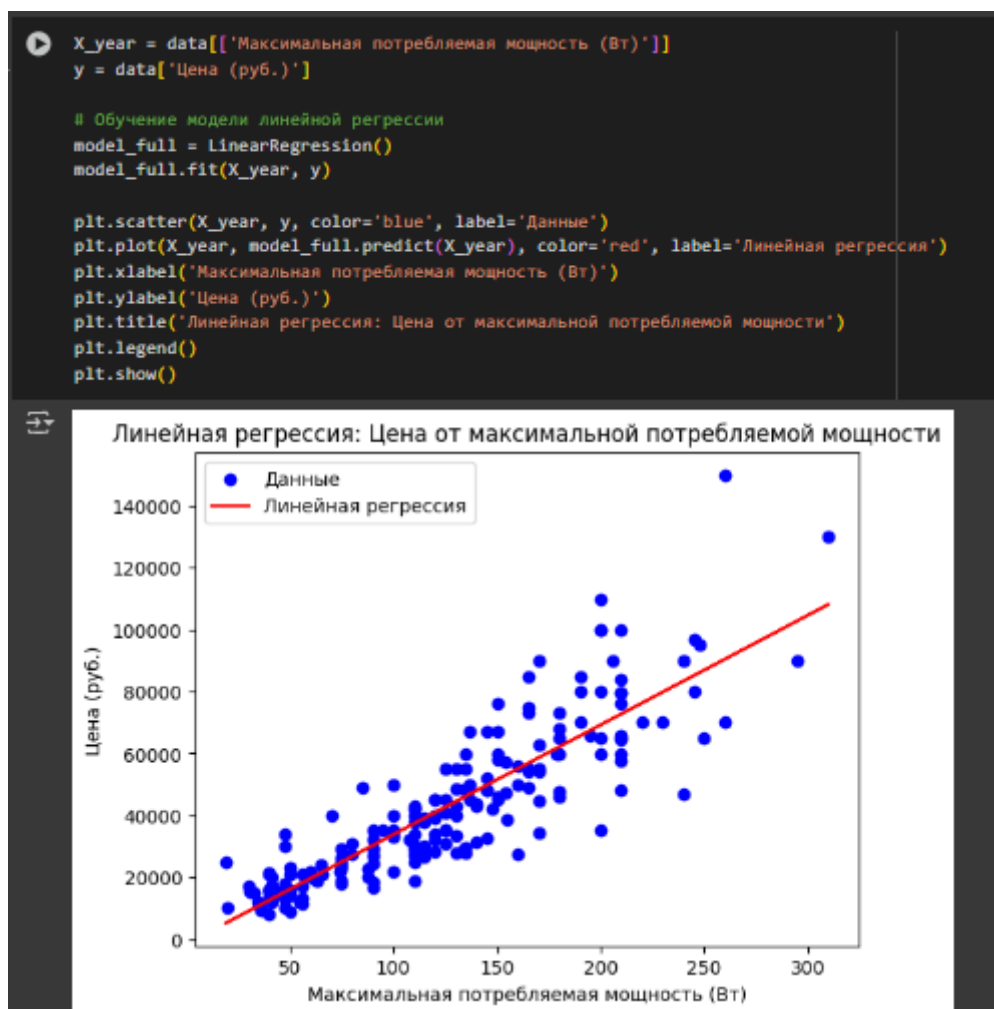


Рисунок 7 – Линейная регрессия: Цена от максимальной потребляемой мощности

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

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 74,7%, что указывает на хорошую предсказательную способность модели (рис.8).

```
from sklearn.metrics import mean_squared_error

# Предположим, что у вас есть обученная модель (например, линейная регрессия)
# и данные: X_train, X_test, y_train, y_test

# Пример: создание модели и обучение
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)

# Предсказания для тренировочного и тестового наборов
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# Вычисление MSE для тренировочного и тестового наборов
print('MSE train: {:.2f}, test: {:.2f}'.format(
    mean_squared_error(y_train, y_train_pred),
    mean_squared_error(y_test, y_test_pred)
))

print('R^2 train: {:.3f}, test: {:.3f}'.format(
    r2_score(y_train, y_train_pred),
    r2_score(y_test, y_test_pred)))
```

MSE train: 160739445.50, test: 95139299.39
R^2 train: 0.747, test: 0.835

Рисунок 8 – Расчет коэффициента детерминации зависимости цены от максимальной потребляемой мощности

Теперь попробуем произвести расчет коэффициент детерминации общей модели (рис.9).

```
model_full = LinearRegression()
model_full.fit(X, y)
y_pred_full = model_full.predict(X)
r2_full = r2_score(y, y_pred_full)
print(f'Коэффициент детерминации общей модели: {r2_full}')
```

Коэффициент детерминации общей модели: 0.7638633515066355

Рисунок 9 – Расчет коэффициента детерминации общей модели

Видно, что модель не поменяла свое значение, и точность вычисления составляет 76,3%.

Теперь возьмем вместо параметра цена десятичный логарифм от этого параметра и проведем повторное исследование с зависимостью всех параметров от логарифма от цены (рис.10).

```
[24] y_pred_log = model_log.predict(X_test_log)
r2_log = r2_score(y_test_log, y_pred_log)
print(f'Коэффициент детерминации (логарифмированная цена): {r2_log}')
```

Коэффициент детерминации (логарифмированная цена): 0.7826566828006637

Рисунок 10 – Коэффициент детерминации модели с логарифмированной ценой

Ссылка на работу в Google Colab приведена ниже.

Google Colab – URL:
https://colab.research.google.com/drive/1j2ZNHISrB4XDpFLWNk1nP0QpooKuEK_n?usp=sharing

Также был проведен анализ с помощью среды Orange, но с использованием большего количества регрессионных моделей.

Внеся собранные данные с помощью виджета File (рис.11) в Orange и разделив данные на тестовые и тренировочные с помощью объекта Cross validation, в котором отметим количество фолдов в размере 5, а также добавив ещё несколько объектов: Distributions и Scatter Plot. Свяжем их с Test and Score (рис.12). Уже от Test and Score проведем к нужным моделям, выставив верные параметры. В качестве моделей для прогнозирования выбраны kNN, Random Forest, Tree, SVM, Gradient Boosting.

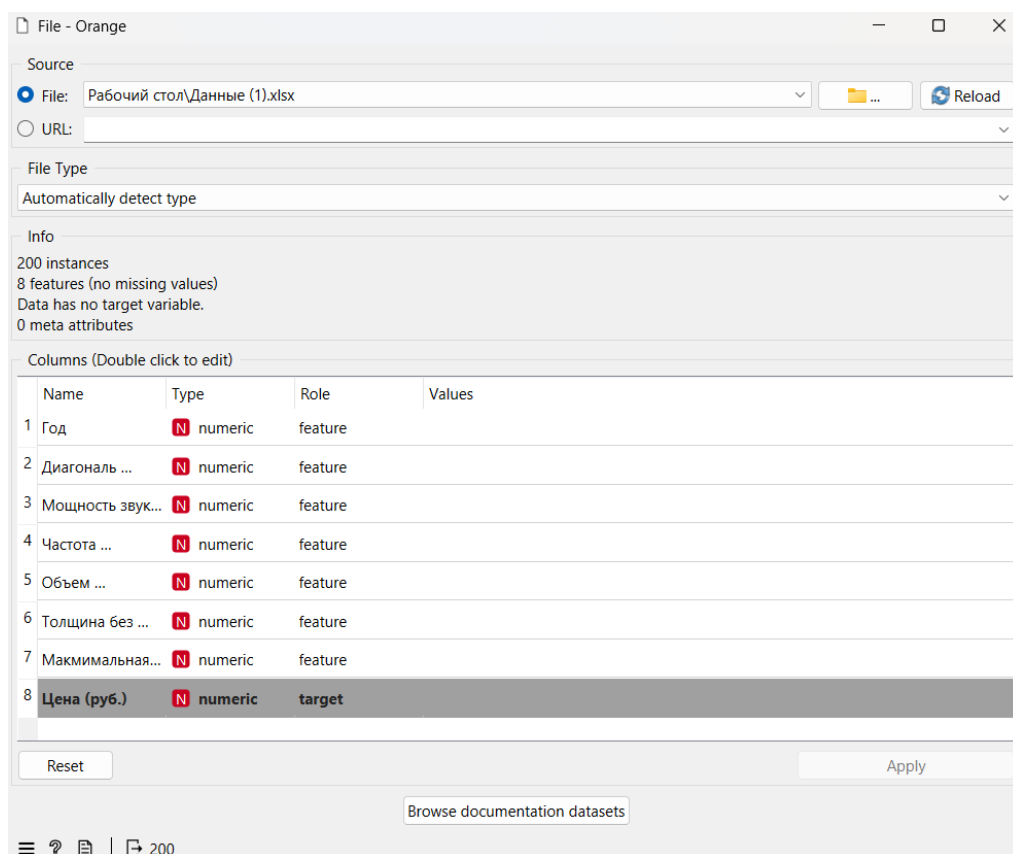


Рисунок 11 – Работа виджета File

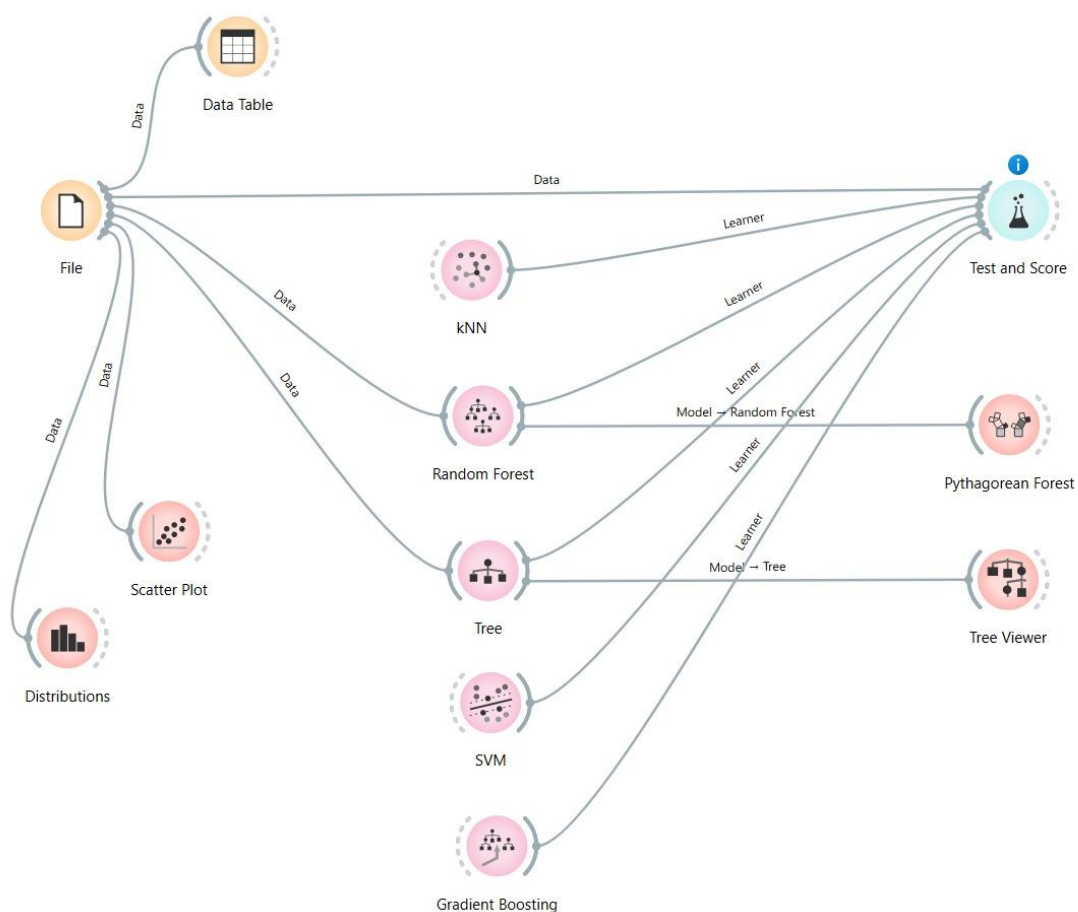


Рисунок 12 – Полная модель в среде Orange

После модели Random Forest и Tree проведем к Pythagorean Forest и Tree Viewer соответственно, и получим следующие графические результаты (рис.13, рис.14).

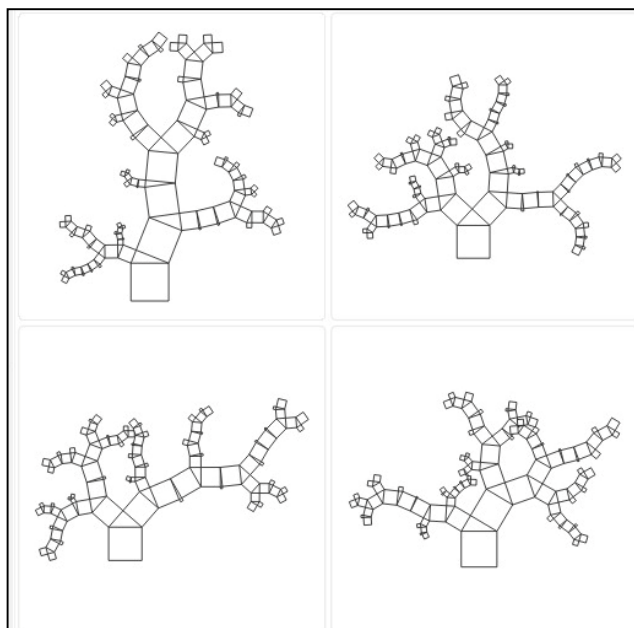


Рисунок 13 – Фрагмент примера работы модели PythagoreanForest

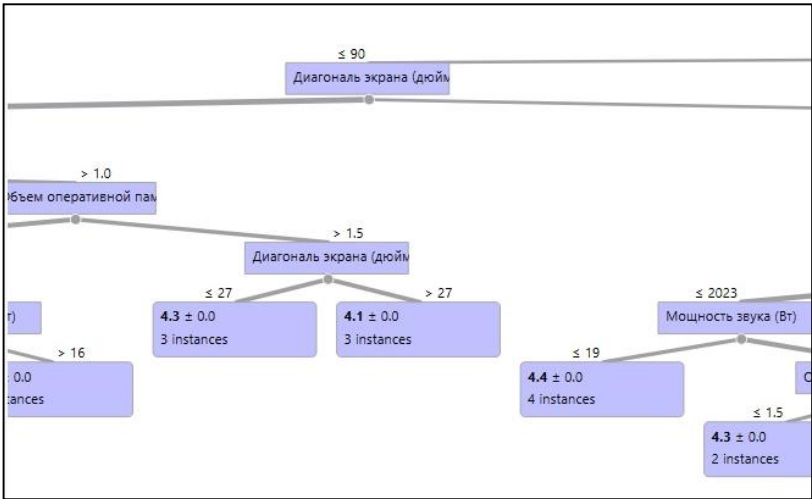


Рисунок 14 – Фрагмент примера работы модели Tree Viewer

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.15).

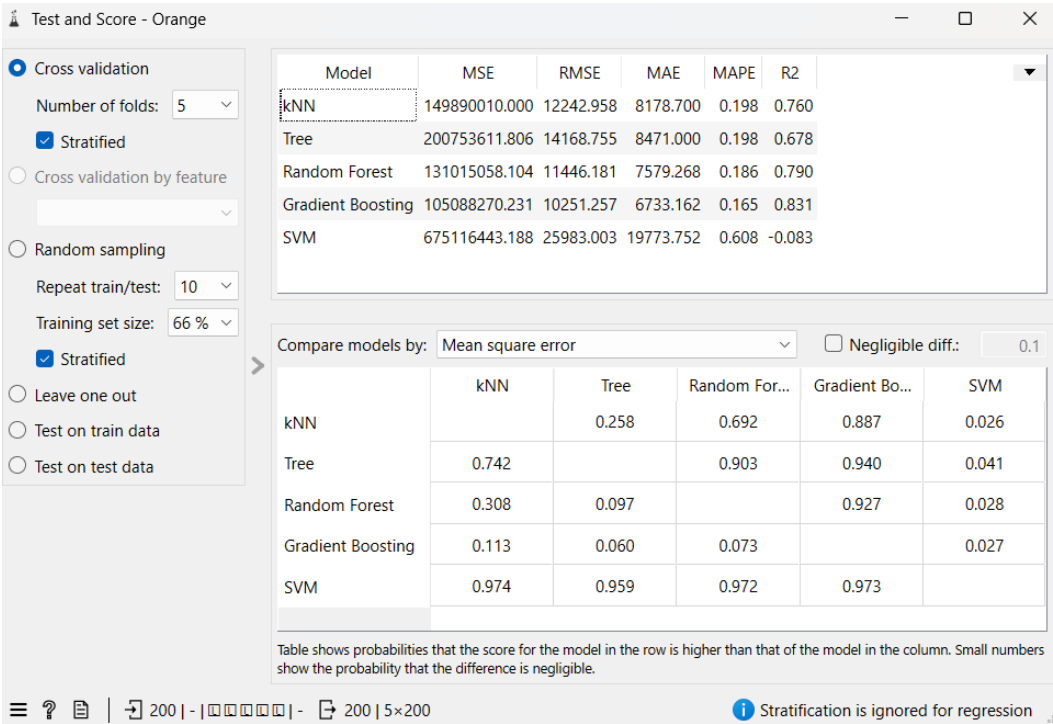


Рисунок 15 – Результат в Test and Score

Самой достоверной и хорошей из моделей получилась модель Gradient Boosting, коэффициент детерминации которой равен 83,1%.
Далее произведем такой же анализ, но при этом добавив в данные параметр, десятичный логарифм от цены (рис.16).

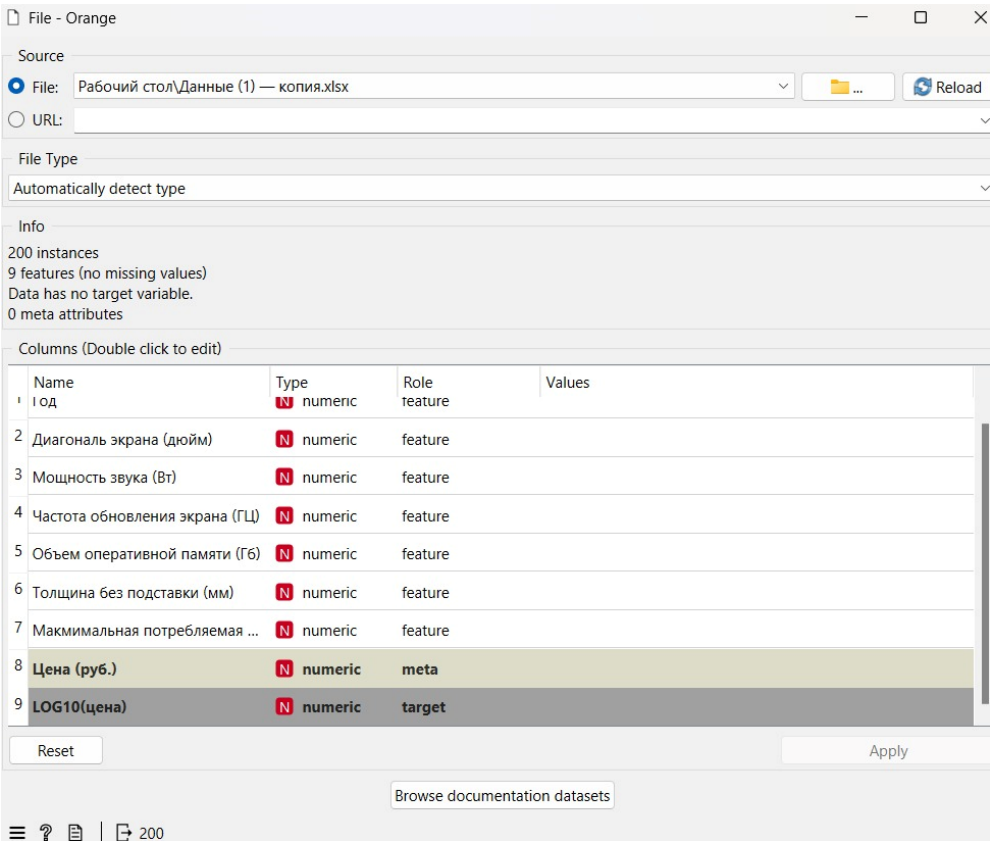


Рисунок 16 – Работа виджета File

После проведенного анализа программная среда Orange выдает результат в Test and Score в виде коэффициента детерминации (рис.17).

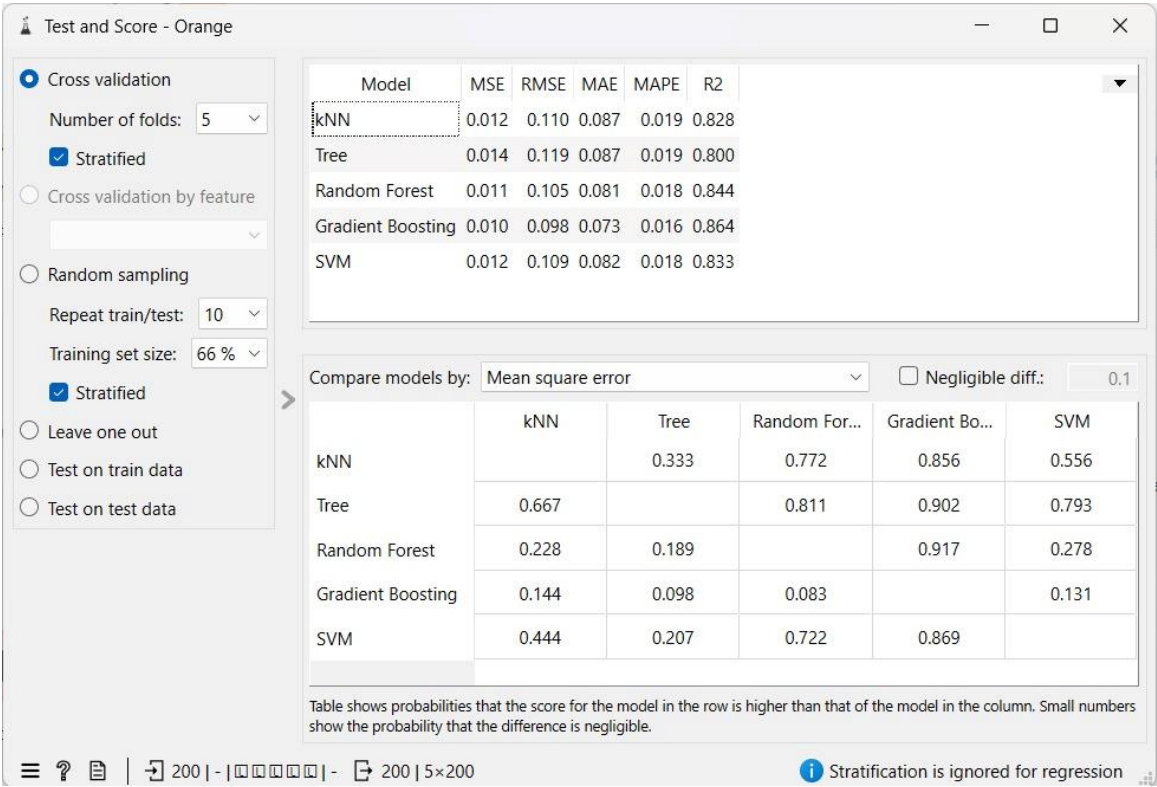


Рисунок 17 – Результат в Test and Score

В данной модели самой достоверной и хорошей из моделей также получилась модель Gradient Boosting, коэффициент детерминации которой равен 86,4%.

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

4 Выводы

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

Библиографический список

1. Сорокин Д. Методы машинного обучения // ITGLOBAL.COM. 31.10.2023. URL:<https://itglobal.com/ru-ru/company/blog/metody-mashinnogo-obucheniya/>(дата обращения: 23.01.2025)
2. Гржибовский А. М. Однофакторный линейный регрессионный анализ // Экология человека. 2008. №10. URL: <https://cyberleninka.ru/article/n/odnofaktornyy-lineynyy-regressionnyy-analiz> (дата обращения: 23.12.2024).
3. Вичугова А. Как увидеть лес за деревьями: что такое DecisionTree и зачем это нужно в Big Data // Цифровая трансформация. 18.03.2020. URL: <https://bigdataschool.ru/blog/decision-tree.html> (дата обращения: 23.01.2025)
4. Бабаев А.М., Шемякина М.А. Обзор классических методов машинного обучения в контексте решения задач классификации // Форум молодых ученых. 2018. №. 11-1 (27). С. 137-143 URL:<https://cyberleninka.ru/article/n/obzor-klassicheskikh-metodov-mashinnogo-obucheniya-v-kontekste-resheniya-zadach-klassifikatsii>(дата обращения: 23.01.2025)
5. Топухов К.Е., Романов А.Ю., Лукьянов В.И. Исследование методов машинного обучения для классификации научных текстов // Электронный сетевой политематический журнал «Научные труды КубГТУ». 2023. №6. С. 78–89. URL: <https://ntk.kubstu.ru/data/mc/0103/4729.pdf> (дата обращения: 23.01.2025)
6. Google Colaboratory/ URL: <https://colab.research.google.com/>(дата обращения: 24.01.2025)