

Применение методов машинного обучения для построения модели прогнозирования цены подержанных ноутбуков серии MacbookAir в облачной среде GoogleColaboratory и в системе Orange

Языков Кирилл Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описывается применение регрессионного анализа для прогнозирования цен. Для проведения исследования использовалась бесплатная среда Orange, написанная на языке Python, а также бесплатная облачная среда GoogleColaboratory. Для примера собраны данные по подержанным ноутбукам серии MacbookAir с сайта avito.ru. В результате исследования, путем создания моделей, удалось спрогнозировать стоимость покупки либо продажи ноутбука с точность до 75% по R-квадрату.

Ключевые слова: регрессионный анализ, метод ближайших соседей, линейная регрессия, Orange, python, GoogleColaboratory.

Application of machine learning methods to build a model for predicting the price of used Macbook Air laptops in the Google Colaboratory cloud environment and in the Orange system

Yazykov Kirill Aleksandrovich

Sholom Aleichem Priamursky State University

student

Abstract

This article describes the use of regression analysis for price forecasting. To conduct the research, the free Orange environment written in Python was used, as well as the free Google Colaboratory cloud environment. For example, data on used Macbook Air series laptops are collected from the website avito.ru. As a result of the research, by creating models, it was possible to predict the cost of buying or selling a laptop with an accuracy of up to 75% by R-squared.

Keywords: regression analysis, nearest neighbor method, linear regression, Orange, python, Google Coolaboratory.

1 Введение

1.1 Актуальность

Последние несколько лет методы машинного обучения применяются во многих отраслях. Один из вариантов использования - прогнозирование цены. Это позволяет спрогнозировать цену покупки или продажи определенного

товара. Для прогноза цены требуется собрать данные по различным параметрам, одним из которых будет цена, предложенная в объявлении. На основе полученных данных можно построить различные модели, которые и помогут с прогнозированием. Среди основных методов прогнозирования выделяют регрессионный анализ.

1.2 Обзор исследований

В книге Н. Дрейпера и Г. Смита “Прикладной регрессионный анализ” описываются основы множественного регрессионного анализа, а также методы подбора и исследования линейных и нелинейных регрессионных моделей [1]. В учебном пособии Г. А. Соколова “Введение в регрессионный анализ и планирование регрессионных экспериментов в экономике” подробно описаны классическая регрессионная модель и методы ее создания [2]. А Н. Рослякова и Н. Горидько в своей монографии “Современный экономический рост: теория и регрессионный анализ” исследовали экономический рост регионов с помощью регрессионного анализа [3].

1.3 Цель исследования

Целью исследования является построения моделей прогнозирования цены подержанных ноутбуков серии MacbookAir на основе регрессионного анализа.

2 Материалы и методы

Для решения поставленной цели будут использованы бесплатная облачная среда GoogleCoollaboratory [4], разработанная для написания программного кода с использованием различных библиотек машинного обучения, а также бесплатная среда Orange, написанная на языке программирования Python.

3 Результаты исследования

Сбор данных осуществлялся по данным сайта www.avito.ru за декабрь 2022 года. Рассматривались объявления о продаже ноутбуков серии MacbookAir разных годов выпуска по всей России. Данные объявлений собирались по нескольким параметрам, а именно: год выпуска, диагональ экрана, плотность пикселей, частота процессора, количество ядер, объем оперативной памяти, цена. Все было собрано в файл MS Excel (рис. 1).

	A	B	C	D	E	F	G
1	Год	Диагональ экрана (дюйм)	Плотность пикселей (ppi)	Частота процессора (ГГц)	Количество ядер	Объем оперативной памяти (ГБ)	Цена
2	2020	13.3	227	3.2	4	8	70000
3	2015	13.3	227	1.6	2	8	28000
4	2020	13.3	227	3.2	4	8	67800
5	2022	13.6	224	3.5	8	8	110000
6	2020	13.3	227	3.2	4	8	69900
7	2019	13.3	227	1.6	2	8	40000
8	2021	13.3	227	3.2	2	8	74000
9	2013	11.6	135	1.3	2	4	21000
10	2022	13.6	224	3.5	8	8	137000
11	2015	13.3	128	1.6	2	8	45000
12	2019	13.3	227	1.6	2	8	35000
13	2020	13.3	227	3.2	4	8	99000
14	2018	13.3	227	1.6	2	8	49000
15	2018	13.3	227	1.6	2	8	65500

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 GoogleCoollaboratory, а именно встроенными библиотеками pandas, seaborn, sklearn.

Проведя исследование для поиска наиболее подходящих для анализа параметров, были выбраны «Год» и «Объем оперативной памяти». А в качестве основных методов были выбраны метод линейной регрессии (linearregression) и метод ближайших соседей (kNN).

Для начала построим модель методом линейной регрессии. Так как у нас лишь один массив данных, а для проверки работы модели 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 LinearRegression построим модель линейной регрессии по параметрам Год и Цена (рис.2).

```

model = linear_model.LinearRegression()
model.fit(x_train, y_train)
y_train_pred = model.predict(x_train)
y_test_pred = model.predict(x_test)

[ ] y_pred=model.predict(x_test.reshape(-1,1))

[ ] plt.figure(figsize=(10, 5))
# plt.plot(x_train, y_real, label='real')
plt.scatter(x_train, y_train, label='train')
plt.scatter(x_test, y_test, label='test')
plt.plot(x_test,y_pred, label='predicted')
plt.legend()
plt.show()

```

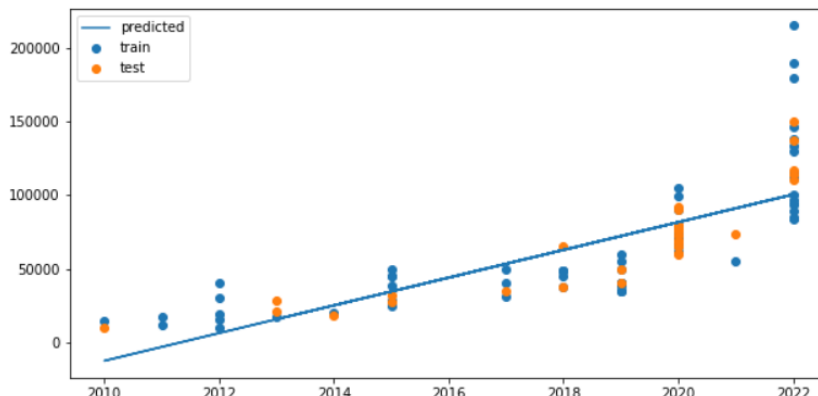


Рисунок 2 – Линейная регрессия (зависимость цены от года выпуска)

Полученный график и есть модель линейной регрессии. На рисунке прекрасно видно, что точки тестовой выборки (оранжевые), 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 71,7%, что говорит о том, что модель неплоха (рис.3).

```

[ ] from sklearn.metrics import mean_absolute_error, mean_squared_error, median_absolute_error, r2_score

print('MSE train: {:.3f}, test: {:.3f}'.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: 775523503.466, test: 330112573.411
R^2 train: 0.567, test: 0.717

```

Рисунок 3 – Расчет коэффициента детерминации зависимости цены от года

После попробуем создать модель, используя все параметры: «Год», «Диагональ экрана», «Плотность пикселей», «Частота процессора», «Количество ядер», «Объем оперативной памяти», «Цена» (рис.4).

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, median_absolute_error, r2_score

print('MSE train: {:.3f}, test: {:.3f}'.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: 433209366.627, test: 125409777.789
R^2 train: 0.758, test: 0.893

Рисунок 4 – расчет коэффициента детерминации общей модели линейной регрессии

На рисунке видно, что в данном случае модель вышла лучше, так как коэффициент детерминации равен уже 89,3%. Но чтобы сделать распределение интенсивностей более симметричными, возьмем вместо параметра Цена десятичный логарифм от этого параметра и проведем повторное исследование с зависимостью всех параметров от логарифма от цены.

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, median_absolute_error, r2_score

print('MSE train: {:.3f}, test: {:.3f}'.format(
    mean_squared_error(y2_train, y2_train_pred),
    mean_squared_error(y2_test, y2_test_pred)))
print('R^2 train: {:.3f}, test: {:.3f}'.format(
    r2_score(y2_train, y2_train_pred),
    r2_score(y2_test, y2_test_pred)))
```

MSE train: 0.011, test: 0.006
R^2 train: 0.870, test: 0.914

Рисунок 5 – Коэффициент детерминации модели с логарифмированной ценой

И сразу заметен результата. Модель получилась достаточно достоверной, а коэффициент детерминации теперь равен 91,4%

Также для прогнозирования цены используется модель, созданная методом ближайших соседей (kNN). Для создания модели понадобятся параметры «Год», «Объем оперативной памяти» и «Цена».

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

```
[ ] from sklearn.neighbors import KNeighborsRegressor

[ ] from sklearn.model_selection import train_test_split

[ ] x=df[['Год']].values
    y=df[['Цена']].values

[ ] x_train, x_test, y_train, y_test = train_test_split(
    ... x, y, test_size=0.3, random_state=0)

[ ] knr3 = KNeighborsRegressor(n_neighbors=3)
    knr3.fit(x_train.reshape(-1,1), y_train.reshape(-1,1))
    knr3.score(x_train.reshape(-1,1), y_train.reshape(-1,1))

0.74559230571645
```

Рисунок 6 – Метод ближайших соседей (3 соседа)

На рисунке видно, что процент верно предсказанных вариантов составляет 74,6%, что является неплохим показателем. Сделаем анализ со всеми параметрами, как и в прошлом методе (рис.6).

```
[ ] from sklearn.neighbors import KNeighborsRegressor

[ ] knr6 = KNeighborsRegressor(n_neighbors=6)

▶ knr6.fit(x1_train, y1_train.reshape(-1,1))
  knr6.score(x1_train, y1_train.reshape(-1,1))

🔗 0.7427877393414755
```

Рисунок 6 – Метод ближайших соседей (6 соседей)

В данном случае модель стала немного хуже. Добавим снова десятичный логарифм от цены, чтобы сравнить результат (рис.7).

```
[68] knr6 = KNeighborsRegressor(n_neighbors=6)

[70] knr6.fit(x2_train, y2_train.reshape(-1,1))
      knr6.score(x2_train, y2_train.reshape(-1,1))

0.8533586427173947
```

Рисунок 7 - Метод ближайших соседей (6 соседей)

Модель стала лучше, процент успешно предсказанной цены составляет 85,3%, что говорит о том, что модель достаточно достоверна.

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

Внеся собранные данные в Orange и разделив данные на тестовые и тренировочные в пропорции 70/30 с помощью объекта DateSampler, а также добавив ещё несколько объектов: Distributions, Correlations, ScatterPlot. Свяжем их с TestandScore (рис. 8). Уже от TestandScore проведем к нужным моделям, выставив верные параметры. В качестве моделей для прогнозирования выбраны RandomForest, Tree, kNN и LinearRegression.

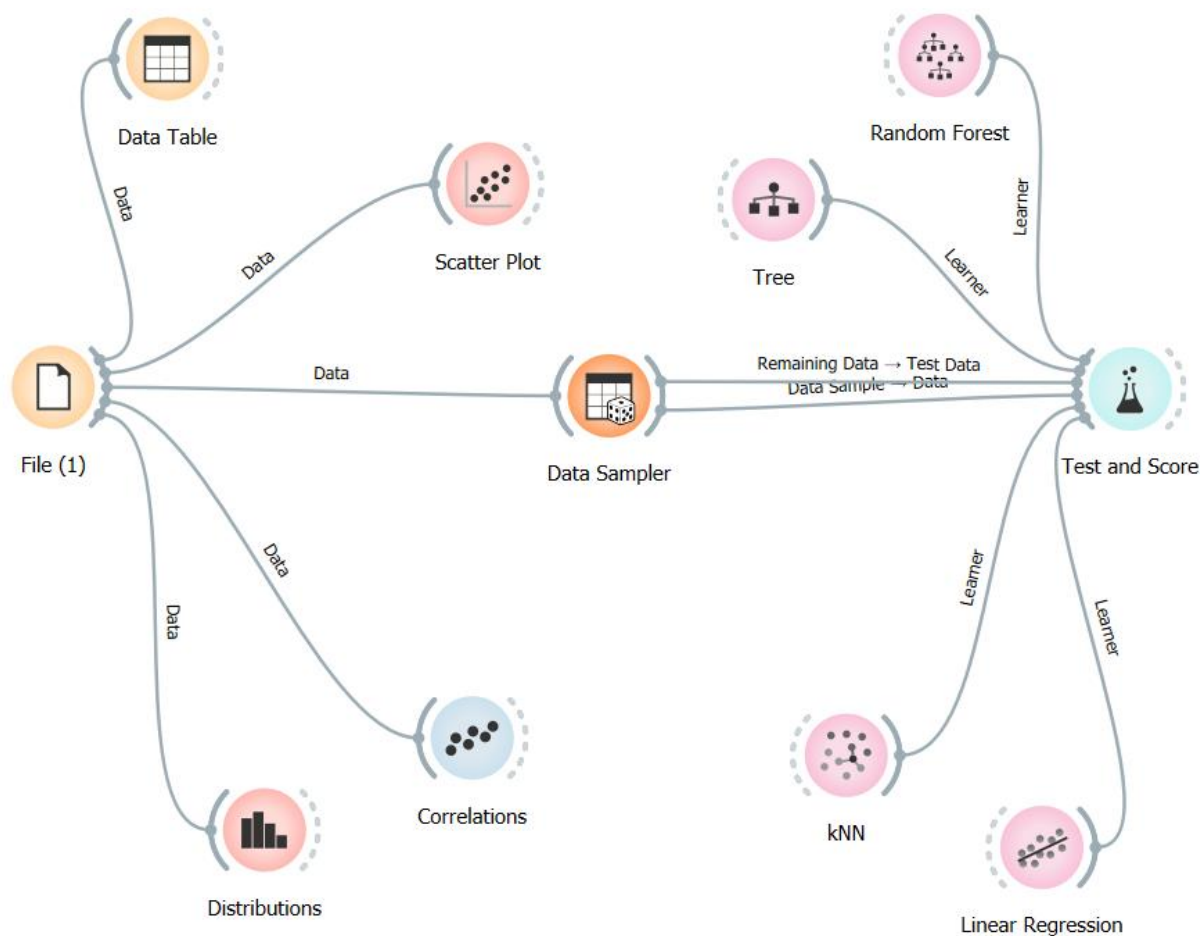


Рисунок 8 - Полная модель

Программная среда Orange самостоятельно проводит анализ, на основе внесенных данных и выставленных параметров. Результат проведенного анализа выводится в TestandScore в виде коэффициента детерминации, из которого уже можно сделать вывод о качестве каждой модели (рис.9).

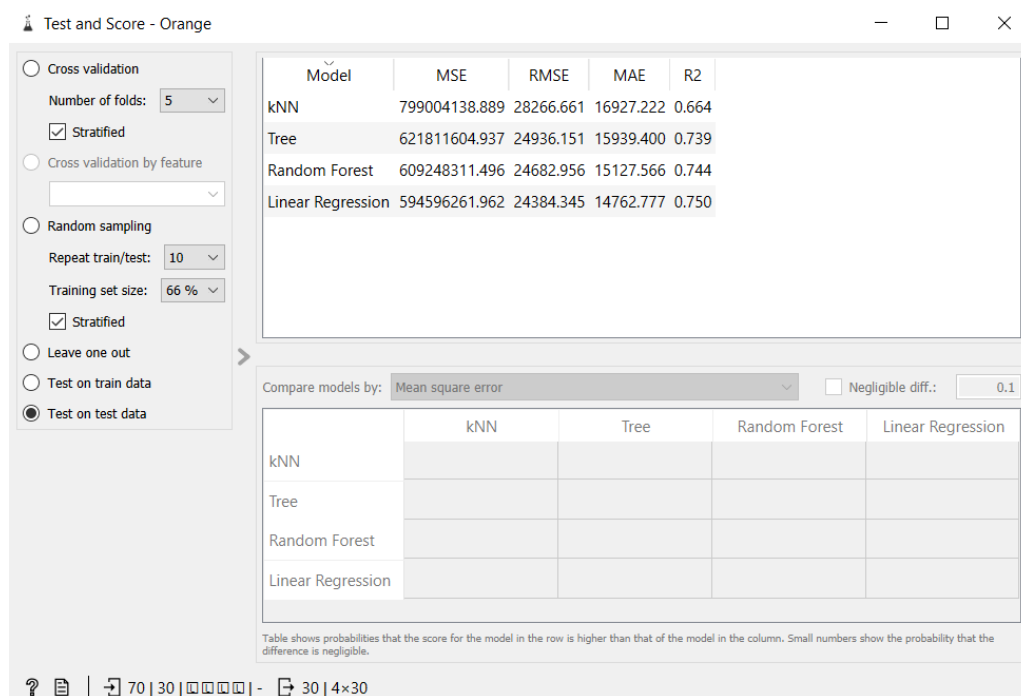


Рисунок 9 - Результат в TestandScore

Самой достоверной и хорошей из моделей получилась модель линейной регрессии, коэффициент детерминации которой равен 75%.

4 Выводы

Таким образом, полученные модели прогнозирования, созданные в GoogleCoollaboratory и Orange можно использовать для покупки и продажи ноутбуков серии MacbookAir разных годов выпуска, а данные при прогнозе будут достаточно достоверными. Проведенное исследование может быть использовано при обучении методам машинного обучения в различных направлениях.

Библиографический список

1. Дрейпер Н. Прикладной регрессионный анализ. Диалектика, 2016. 912 с.
2. Соколов Г. А. Введение в регрессионный анализ и планирование регрессионных экспериментов в экономике. М.: НИЦ ИНФРА-М, 2022. 202 с.
3. Рослякова Н., Горидько Н. Экономический рост регионов и транспортная инфраструктура: регрессионный анализ // РИСК: ресурсы, информация, снабжение, конкуренция. 2013. №. 1. С. 157-161.
4. Сервис для проведения исследования Colaboratory URL: <https://colab.research.google.com/>
5. Сенникова А. Е., Ворокова Н. Х. Регрессионный анализ влияния объема и структуры основных фондов на эффективность сельскохозяйственного производства // Вестник Алтайской академии экономики и права. 2020. №. 11. С. 329-332.