Оценка нескольких моделей

(в среде Colab)

Цель этого примера - сравнить производительность моделей в ноутбуке Jupyter.

**1 Введение**

Цель этого примера - сравнить производительность моделей.

В этом примере будет использоваться набор данных для жилья в Бостоне. В наборе данных имеется 506 строк. Целевая переменная - средняя цена на жилье. Существует 13 переменных-предикторов, включая среднее количество комнат на жилье, уровень преступности по городу и т. Д. Более подробную информацию об этом наборе данных можно найти в [UCI](https://archive.ics.uci.edu/ml/datasets/Housing) .

Набор данных разбит на два для этого анализа: один для обучения (70% образцов) и один для тестирования (30% образцов). После разработки модели на основе данных обучения ее производительность будет проверяться на данных тестирования. Три модели будут сравниваться после их установки:

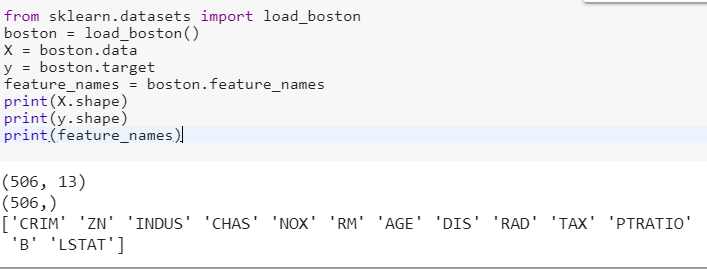
1. a linear model using all variables

2. a linear model after variable selection

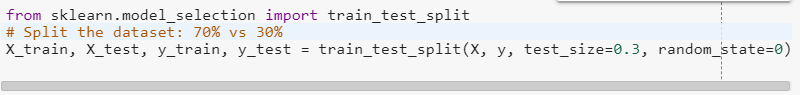
3. a Gradient Boosting Machine (GBM) model

**2 Подготовка данных**

Мы начнем с загрузки данных.



Затем мы разделили набор данных на два: один с 70% образцов, а другой 30%.

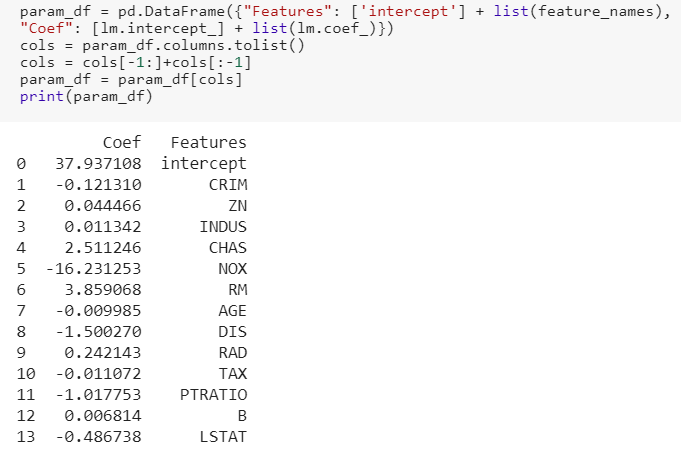


**3 Обучение моделей**

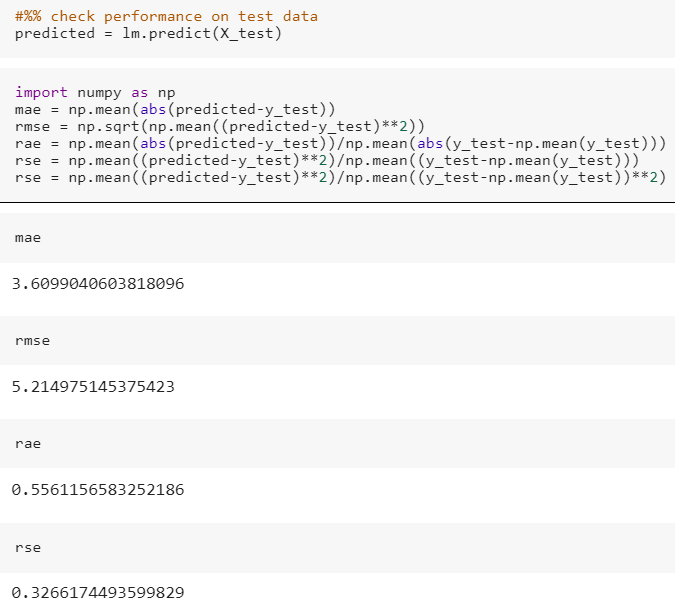
**3.1 Линейная модель**

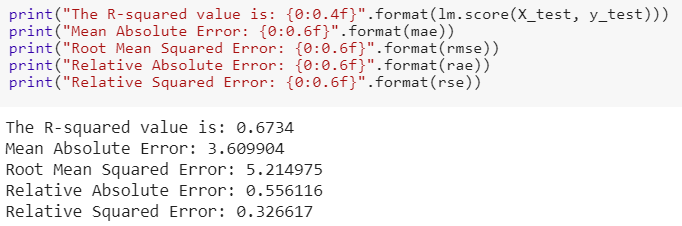
Следующие строки кода соответствуют линейной модели, использующей все функции.





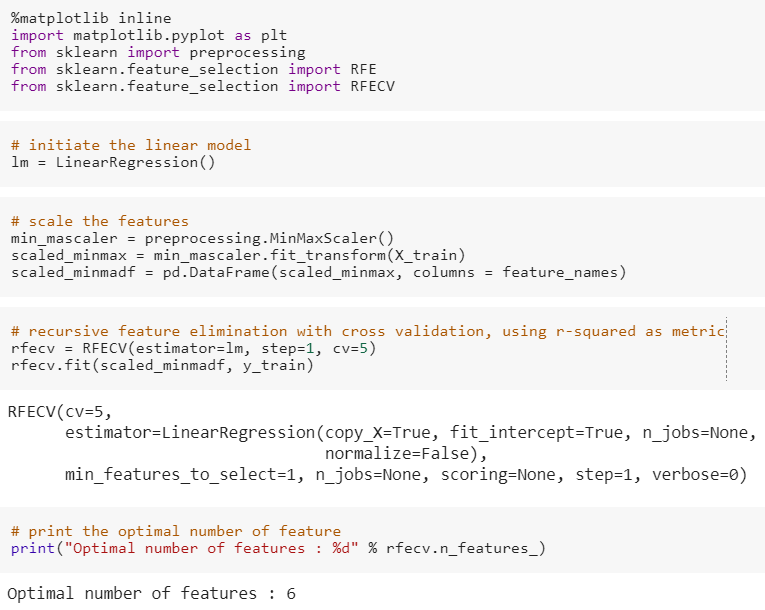
Затем мы проверяем производительность модели на тестовом наборе данных.



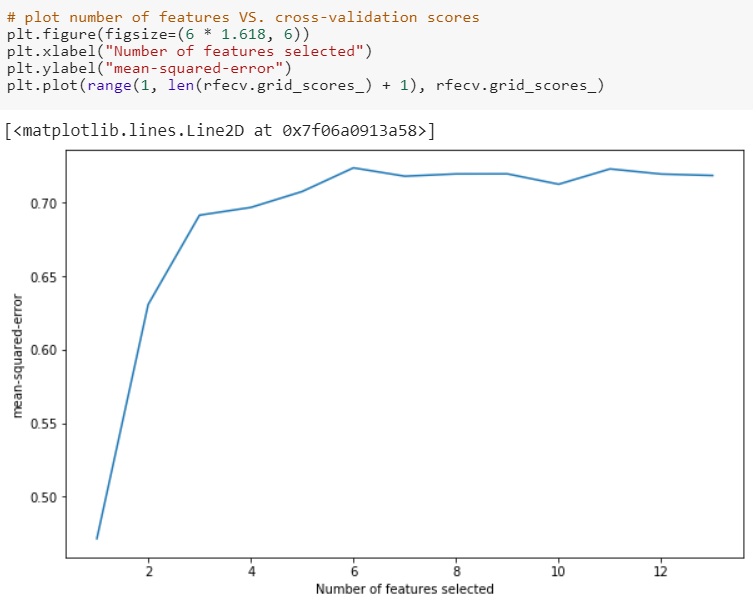


**3.2 Линейная модель с выбором функций**

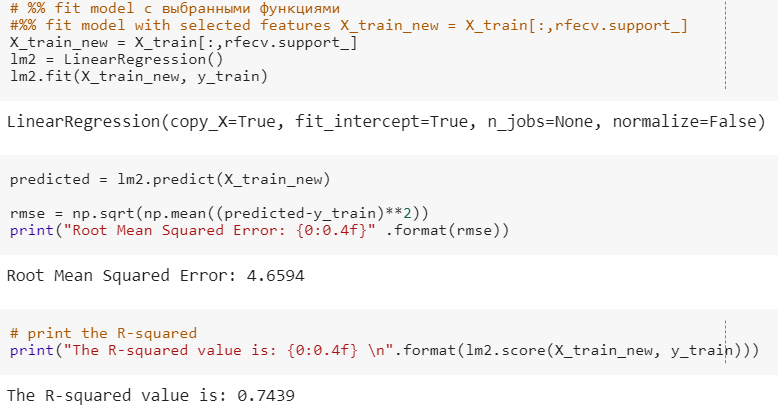
Следующие строки кода выбирают переменные, используя рекурсивное устранение признаков.



*# save the selected features*



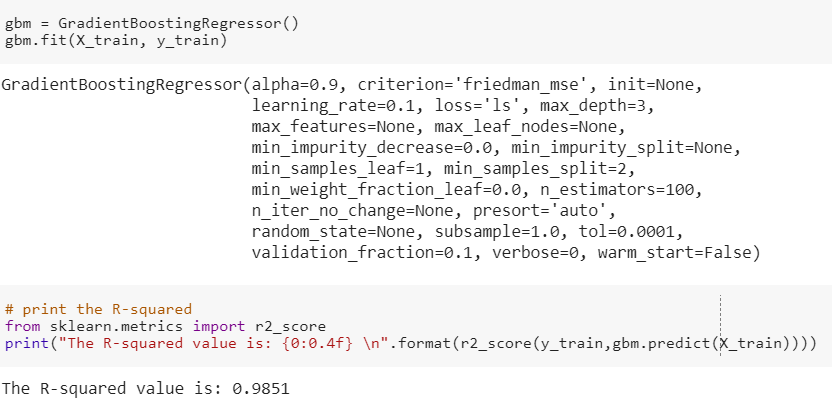
Используя выбранные функции, мы можем использовать их в линейной модели.

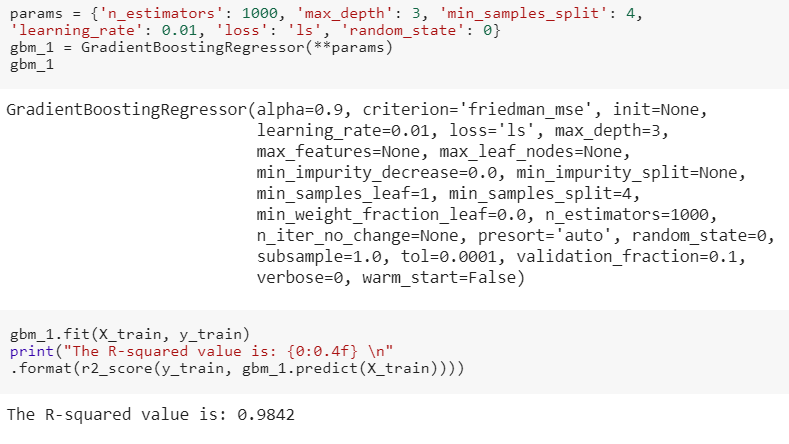




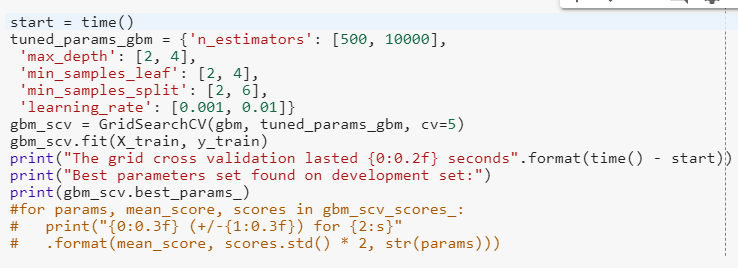
Затем оценивается производительность модели на тестовом наборе данных.

### 3.3 GBM





параметров модели GBM, используя кросс-валидацию вместе с поиском сетки.

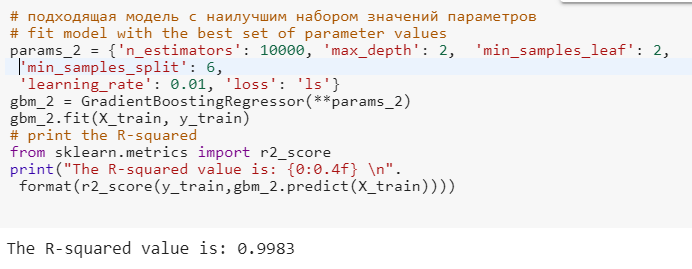


he grid cross validation lasted 380.76 seconds

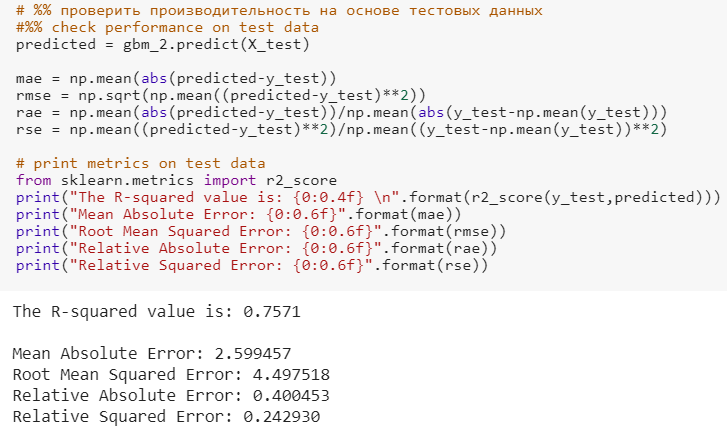
Best parameters set found on development set:

{'learning\_rate': 0.01, 'max\_depth': 2, 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 10000}

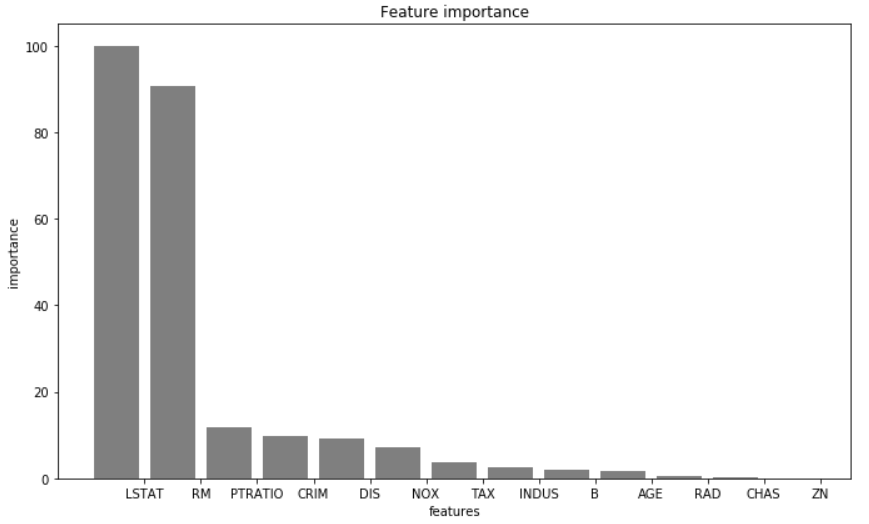
С параметрами, которые дают лучшую производительность, мы поместим модель GBM.

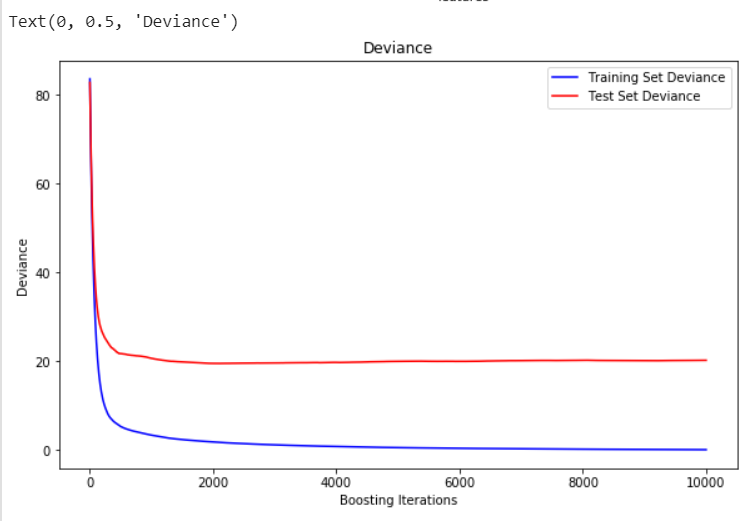


Затем мы можем проверить производительность модели на тестовых данных.



Построение графиков проверки важности признаком (входов) и зависимости величины ошибок (Loss)от числа итераций на обучающем и тестовом выборках.



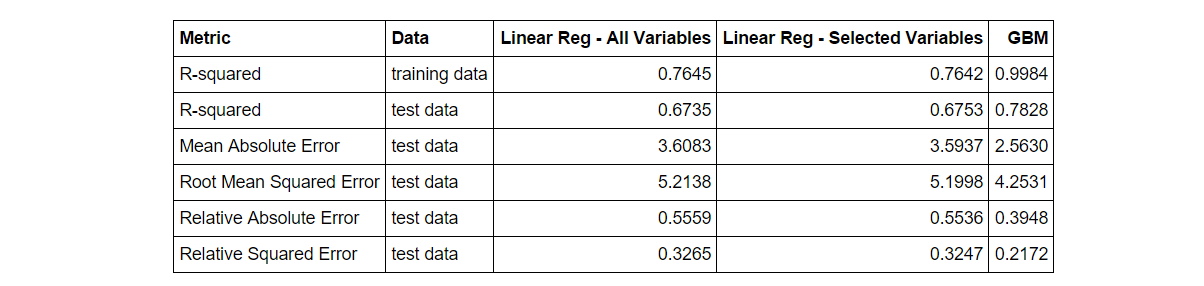


**4 Заключение**

Следующая таблица сравнивает производительность трех моделей. Как и ожидалось, значение R-квадрата для тестовых данных ниже, чем у данных обучения для всех моделей.

Линейная модель после выбора переменной лучше работает на тестовых данных, чем модель со всеми переменными. Например, RMSE для уменьшенной модели составляет 5.1998 по сравнению с 5.2138 для полной модели. Таким образом, более простая модель (с двумя меньшими переменными) работает лучше, чем более сложная. Это иллюстрирует преимущества проведения выбора переменных.

GBM лучше всего подходит для всех моделей по всем критериям.

[](https://az712634.vo.msecnd.net/notebooks/98d632c2-c9cc-11e5-920e-22a0981b53ab.PNG)

Создан сотрудником Microsoft.   
Авторское право (C) Microsoft. Все права защищены.