Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

Кулик Андрей

Прогноз спроса по характеристикам и цене товар

Практическая работа

Научный руководитель:
к.ф.-м.н.
О.В.Сенько

Москва, 2020

Содержание

1 Введение

В данной работе рассматривается задача регрессии. Целью является прогнозирование спроса на товар по его цене и характеристикам. Были использованы разные методы машинного обучения для данной цели. Для сравнения каждого из подходов используются метрики и кросс-валидация.

2 Исследуемые данные

В качестве данных взят датасет, содержащий 1241 товар. Каждому товару соответсвует его спрос(числовой признак - целевая переменная), 517 бинарных признаков, а также цена. Так как всего один небинарный признак, посмотрем на его распеределение(Рис. 1). Как видно у нас есть один выброс, так что уберем этот объект из выборки. Выборка разделена на обучающую и тестовую(размер тестовой - 30%)

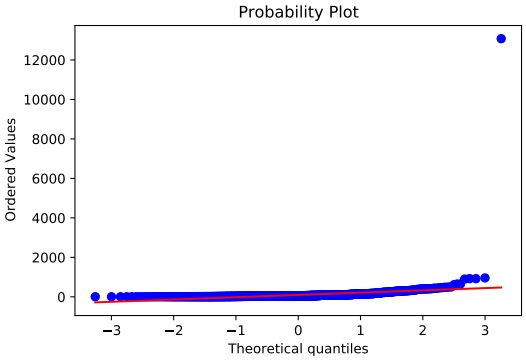


Рис. 1: Распеределение стоимости

3 Модели для обучения

Для данной задачи были выбраны 3 модели обучения: Elastic Net, Random forest, Gradient boosting. Для проверки моделей используется кросс-валидация.

3.1 Elastic Net

Elastic net - линейная модель регрессии, которая объеденяет L1 и L2 регуляризаторы. Известно, классическая задача регрессии ствится так:

$$|y - Xw|^2 \to \min_w .$$

С целью построения наилучшей модели эластическая сеть вводит 2 регуляризатора L1 и L2:

$$|y - Xw|_2^2 + \lambda_1 |w|_1 + \lambda_2 |w|_2^2 \to \min_w$$

Для контроля вклада L1 и L2 введем α и l_1:

$$\alpha = \lambda_1 + \lambda_2; l_1 = \frac{\lambda_1}{\lambda_1 + \lambda_2}$$

Тогда конечная форма примет вид:

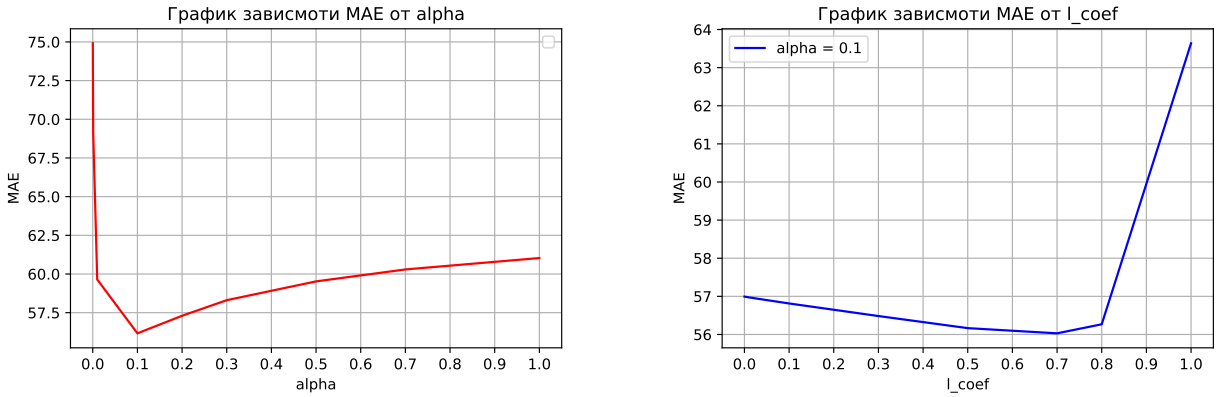
$$|y - Xw|_2^2 + \alpha * l_1 * ||w||_1 + 0.5 * \alpha * (1 - l_1) * ||w||_2^2$$

Здесь, если $l_1 = 0$, то вклад только от L2 - гребневая регрессия, если $l_1 = 1$, то это модель Лассо(вклад только от L1). $\alpha = 0$ эквивалентно линейной регрессии.

Найдем лучшие гиперпараметры alpha и l_1 для нашей задачи. Модель ElasticNet

сымпортирована из библиотеки sklearn. Качество модели при соответсвующих параметрах оцениваем с помощью MAE(mean absolute error)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|a(x_i) - y_i|$$



Лучшим параметром alpha является 0.1, а l_1 - 0.7. Также заметим, что сначала значение ошибки быстро падает, а после 0.1 возрастает. Это можно объяснить тем, что при очень маленьких значениях оба регулизатора штрафуют незначительно,а при больших, наоборот слишком сильно, и следовательно MAE высокий. С l_1 же сначала идет плавный спад погрешности до 0.7, а затем резкий скачов вверх. Это можно объяснить тем, что изначально вклад в основном идет от L2 регуляризатора, он штрафует большие значения весов. Далее повышается штраф L1 регуляризатора, он решает проблему мультиколлинеарности, обнуляя веса некоторых коэффициентов модели. При высоких значениях l_1 уменьшается вклад L2 и ошибка увеличивается.

Посчитаем MAE на лучших параметрах и проверим модель на кросс-валидации с 5-ю фолдами. MAE у модели показал результат 56.03, по кросс-валидации: 59.4, что означает, модель не переобучена. Как видно на графике(Рис. 3) модель трудно предсказывает высокие значения спроса, скорее всего таких данных очень мало в исходной выборке, а ElasticNet не устойчива к таким выбросам из-за линейности функции.

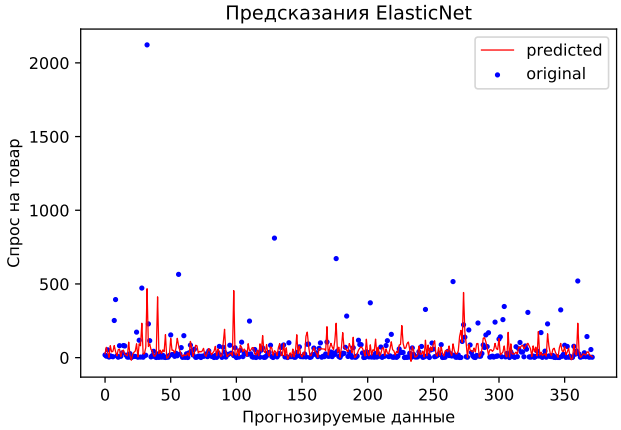


Рис.3 Ошибки в предсказании Elastic Net

3.2 Градиентный бустинг

Бустинг в задаче регрессии: Рассмотрим задачу минимизации квадратичной функции потерь:

$$\frac{1}{2}\sum_{i=1}^{l}(a(x_i)-y_i)^2 \to \min_w$$

Будем искать алгоритм в виде суммы базовых:

$$a(x)=\sum_{n}^{N} b_n(x)$$

где базовые алгоритмы b_n принадлежат некоторому семейству A. Построим первый базовый алгоритм:

$$b_1(x) := \underset{b\in A}{argmin}\frac{1}{2}\sum_{i=1}^{l}(b(x_i)-y_i)^2$$

Решение такой задачи не представляет трудностей для многих семейств алгоритмов. Теперь мы можем посчитать остатки на каждом объекте — расстояния от ответа нашего алгоритма до истинного ответа:

$$z_i^{(1)} = y_i - b_1(x_i)$$

Если прибавить эти остатки к ответам построенного алгоритма, то он не будет допускать ошибок на обучающей выборке. Значит, будет разумным построить второй алгоритм так, чтобы его ответы были как можно ближе к остаткам:

$$b_2(x) := \underset{b\in A}{argmin}\frac{1}{2}\sum_{i=1}^{l}(b(x_i)-z_i^{(1)})^2$$

Каждый следующий алгоритм тоже будем настраивать на остатки предыдущих:

$$z_i^{(N-1)} = y_i - \sum_{n=1}^{N-1} b_n(x_i) = y_i - a_{N-1}(x_i)$$

$$b_N(x) := \underset{b\in A}{argmin}\frac{1}{2}\sum_{i=1}^{l}(b(x_i)-z_i^{(N-1)})^2$$

Заметим, что остатки могут быть найдены как антиградиент функции потерь по ответу модели, посчитанный в точке ответа уже построенной композиции:

$$z_i^{(N-1)} = y_i - a_{N-1}(x_i) = -\frac{\partial}{\partial a}\frac{1}{2}\sum_{i=1}^{l}(a-y_i)^2\Big|_{a=a_{N-1}(x_i)}$$

Это наблюдение наводит нас на мысль, что аналогичным образом можно было бы строить композиции, оптимизирующие и другие функции потерь — достаточно заменить остатки z(N) i на градиент нужного функционала. Именно так и работает градиентный бустинг. Идея заключается в том, что мы будем строить алгоритм минимизации с помощью градиентного спуска.

Эксперименты Для данной задачи в качестве функции потерь, которую оптимизирует градиентный бустинг, я взял Lead absolute deviation(lad).

$$S=\sum_{i=1}^{n}|y_i-f(x_i)|.$$

, где задача минимизации состоит в нахождении

$$f(x_i) \approx y_i.$$

.

LAD - является robust функцией, тоесть устойчивой к выбросам, что актуально в нашей задаче.

Реализацию возьмем из библиотеки sclearn, в качестве параметра loss - возьмем 'lad', что говорит о том, что модель модель использовать LAD в качетсве функции потерь. Сделаем предсказания и исследуем ошибки.(Рис. 4) MAE = 46.7, что значительно меньше, чем у Elastic Net. Это связано с тем, что модель меньше реагировала на выбросы, тем самым точнее предсказала результаты. Проверим на кросс-валидации с 5-ю фолдами. MAE = 47.35, а значит модель не переобучилась.

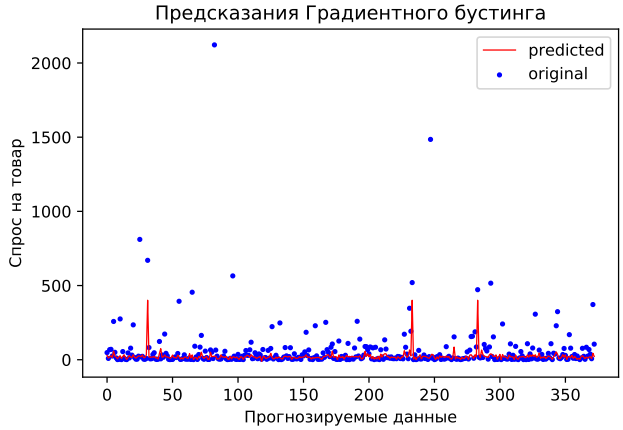


Рис.4 Ошибки в предсказании Градиентный бустинг

Градиентный спуск позволил понизить MAE благодаря тому, что использовал в своей модели LAD функцию, которая является robust, для наших данных это важный критерий при выборе функции потерь модели.

3.3 Random forest regression

Главное проблемой деревья решений является переобучение, ведь можно дойти до слишком шлубокого уровня дерева,аппроксимируя таким образом нюансы конкретных данных вместо общих характеристик распределений, из которых они получены. Решением данной задачи называется *баггинг*.Багинг использует ансамбль параллельно работающих переобучаемых оценивателей и (для задачи регрессии) *усредняет* полученных результатов для получения оптимальной классификации. Ансамбль случайных деревьев принятия решений называется **random forest**.

Для более эффективной рандомизации деревьев принятия решений обеспечивается определенная стохастичность процесса выбора разбиений. При этом всякий раз в обучении участвуют все данные, но результаты обучения все равно сохраняют требуемую случайность.

Применим данный метод для нашей задачи. Регрессор сымпортирован из библиотеки sklearn. Будем использовать стандартные гиперпараметры, тогда MAE = 50.77, этот результат лучше, чем Elastic Net, но хуже Градиентного спуска. По кросс-валидации MAE = 50.93, это значит, что модель не переобучилась. Посмотрим на график ошибок.(Рис.5)

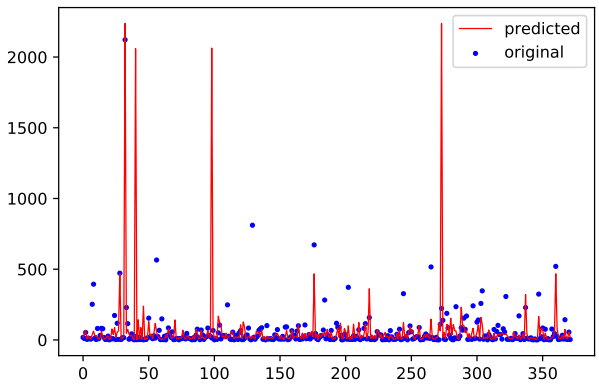


Рис.5 Ошибки в предсказании Случайные леса

Можно наблюдать, что модели удалось предсказать очень высокие значения. Но тем самым это привело к очень высоким ошибкам в тех случаях, когда модель предсказывала 'высокое значение', а исходное было 'низким'.

4 Вывод

Были реализованна задача предсказания спроса товара по его характеристикам и цене. Для этой реализиции были применены 3 модели машиннного обучения для выявления лучшей из них. Опыты показали, что для данной задачи лучшей моделью та, которая устойчива к выбросам ('высоким' значениям данных). Для этого используются robust функции потерь.В данном случае градиентный бустинг позволил нам использовать такую функцию(LAD в данном случае), так как этот метод может использовать любые дифференцируемые функции. Таким образом удалось достигнуть результата при MAE = 56.

Список литературы

[1] LAD - https://en.wikipedia.org/wiki/Least_absolute_deviations

[2] GradientBoostingRegression -
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

[3] Градиентный бустниг -
http://www.machinelearning.ru/wiki/images/7/7e/Sem03_ensembles_2014.pdf

[4] Elastic Net -
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html