Санкт-Петербургский государственный университет
Кафедра компьютерного моделирования и многопроцессорных систем

Воробьева Любовь Александровна

Выпускная квалификационная работа бакалавра

Прогнозирование просмотров объявлений по автомобилям с пробегом на классифайде Авито

Направление: 01.03.02 «Прикладная математика и информатика»
ООП: Прикладная математика, фундаментальная информатика и программирование

Научный руководитель:
кандидат физ.-мат. наук,
Панкратова Ярославна Борисовна

Рецензент:
кандидат физ.-мат. наук,
Гончарова Анастасия Борисовна

Санкт-Петербург
2021

Содержание

Введение

В настоящее время большую популярность приобретает динамическое ценообразование с использованием алгоритмов машинного обучения. Такой подход позволяет увеличить прибыль и объем продаж за счет гибкой подстройки цен под изменения рынка. Динамическое ценообразование широко используется в туристической и транспортной отраслях, однако на рынке автомобилей такой способ формирования цены появился относительно недавно.

Так как дилеры сами устанавливают цену автомобиля, им необходим инструмент оптимальной оценки машин, предлагающий не только цену вывода в продажу, но и стратегию последующей переоценки. При размещении рекламы на классифайде одним из ключевых факторов такого алгоритма являются просмотры объявления, так как их количество непосредственно связано с числом звонков по автомобилю, которые в свою очередь приводят к продаже. Таким образом, управление ценой автомобиля связано с управлением просмотрами. При этом важно учитывать не только возможность переоценки, но и применения услуг продвижения на классифайде.

Задачу создания модели динамического ценообразования для автомобилей можно разделить на несколько подзадач:

1. Построение модели для прогнозирования количества просмотров по автомобилю в каждый день с начала продажи без применения продвижения или переоценки.
2. Моделирование коэффициента конверсии просмотров в звонки.
3. Построение модели для прогнозирования просмотров с учетом применения продвижения и переоценки.
4. Построение модели динамического ценообразования.

В данной работе представлено решение первой подзадачи.

Постановка задачи

Целью данной работы является исследование связи характеристик автомобиля с количеством просмотров по нему, построение модели машинного обучения для прогнозирования просмотров, которые получит конкретный автомобиль в каждый день с начала продаж, а также интерпретация результатов модели как рекомендации к переоценке. Для выполнения поставленных целей необходимо решить следующие задачи:

1. Ознакомиться с контекстом задачи
2. Изучить модель случайного леса
3. Собрать необходимые для работы данные
4. Обучить и настроить модель на собранных данных
5. Оценить качество полученной модели, интерпретировать ее

Обзор литературы

При написании работы основными источниками информации являлись учебно-методическая литература, публикации из научных изданий, а также интернет-источники.

Теоретические основы принципов машинного обучения с учителем, в частности построения модели случайного леса, были взяты из книг [1], [2] и [3]. Практическая часть работы основывалась на прослушанном курсе лекций [4] и книге [5]. Кроме того, дополнительными источниками информации являлись публикации [6]-[10].

На основе представленной литературы была построена модель случайного леса, проведена ее оценка и представлена интерпретация результатов.

Глава 1. Основные понятия и определения

1.1 Дилеры и размещение на классифайдах

В данной работе рассматривается работа дилеров на рынке автомобилей с пробегом. Введем несколько определений.

Определение 1.1. Классифайд — ресурс с объявлениями от физических и юридических лиц, сгруппированными по темам.

Определение 1.2. Целевой срок оборачиваемости — целевой срок оборачиваемости склада для дилера составляет 30 дней. Под оборачиваемостью имеется в виду количество дней, в течение которых автомобиль находится в продаже.

Определение 1.3. Токсичный склад — токсичным складом считаются автомобили, которые остаются непроданными по истечении двух целевых сроков оборачиваемости, то есть дольше 60 дней.

Всех дилеров на рынке условно можно разделить на следующие категории:

1. Около 70% рынка занимают дилеры, прибыль которых находится под угрозой. Основной характеристикой данной категории является токсичный склад, составляющий более 10% от общего числа машин. У таких дилеров наблюдаются высокие продажи в первые дни, которые, как правило, обусловлены выводом автомобилей по заниженной ценой. При этом оставшиеся машины продаются крайне медленно и впоследствии образуют токсичный склад.
2. Дилеры, продажи которых в первые 10 дней составляют более 30%, а токсичный склад менее 10%.
3. Дилеры, оборачиваемость которых составляет не более 30 дней, а токсичный склад практически отсутствует. В отличие от первых

двух категорий, данный дилер пользуется оценкой автомобилей, а также услугами продвижения на классифайде.

Независимо от категории, основной целью каждого автодилера является увеличение прибыли, для этого необходимо продавать автомобили по максимально возможной цене за целевой срок оборачиваемости. Управлять сроками продажи можно с помощью таких инструментов как переоценка и продвижение на классифайдах.

Переоценка. На данный момент существуют сервисы, прогнозирующие наиболее вероятную цену продажи автомобиля на основе алгоритмов машинного обучения. Предварительная оценка дает дилеру возможность вывести автомобиль по завышенной цене и в течение последующих 30 дней переоценивать предложение, приближаясь к наиболее вероятной цене продажи. Такая тактика предотвращает продажу автомобилей по заниженной цене, как это происходит у дилеров первой и второй категории, и, как следствие, способствует уменьшению токсичного склада.

Продвижение на классифайдах. При размещении на классифайдах дилер ставит перед собой цель получить максимальное число звонков, которые в свою очередь зависят от количества просмотров объявления. Таким образом, на самом деле для дилера важно получить как можно больше просмотров. Количество просмотров зависит от даты публикации — новые объявления находятся в начале поисковой выдачи, и от цены предложения, так как переоценка закономерно приводит к увеличению просмотров. Кроме этого, классифайды предлагают различные способы продвижения объявлений, такие как подъем вверх списка выдачи или выделение цветом, а сервис Авито сразу предоставляет услугу увеличения просмотров в несколько раз.

По результатам исследований за пределами данной работы было установлено, что переоценка приводит к увеличению звонков и незначительному росту просмотров. Применение продвижения, напротив,

нацелено на увеличение просмотров, при этом звонки остаются практически на прежнем уровне. Из этого следует, что оптимальным будет одновременное применение переоценки и продвижения. Однако в какой момент времени необходимо переоценивать и продвигать автомобиль пока открытый вопрос. В настоящее время часто используется следующая стратегия:

1. Автомобиль выводят по максимально высокой цене и начинают ее тестировать.
2. Если в течение первой недели объявление не получает звонков, происходит переоценка предложения примерно на 3%.
3. Если звонки были получены, цену оставляют неизменной еще на 3 дня.

В среднем для продажи автомобиля необходимо получить 6-7 звонков. Поэтому при интерпретации результатов модели будем говорить, что требуется применить переоценку не только в случае, когда в первую неделю не ожидается звонков, но и в случае, когда за целевой срок оборачиваемости будет получено меньше 6 звонков. В последнем случае переоценку предлагается сделать на 10-й день с начала продажи.

1.2 Машинное обучение. Решающие деревья

Последние годы алгоритмы машинного обучения (МО) набирают популярность, так как они способны выявить неочевидные взаимосвязи в данных. Самыми популярными моделями МО являются линейная регрессия и случайный лес. Первый хорошо восстанавливает простые зависимости, в то время как второй применяется для решения задач с нелинейными взаимосвязями в данных. Поскольку данные, собранные для выполнения настоящей работы, обладают нелинейной зависимостью, выбор был сделан в пользу модели случайного леса.

Введем определения, необходимые для дальнейшего изучения модели.

Определение 1.4. Признак x_i — независимая переменная, описывающая объект (объясняющая переменная).

Определение 1.5. Категориальный признак — признак, который принимает значения из конечного неупорядоченного множества.

Определение 1.6. Объект $x^i = (x_1,..., x_n)$ — вектор признаков.

Определение 1.7. Целевая переменная y — зависимая переменная, значение которой нужно спрогнозировать.

Определение 1.8. Обучающая выборка X объема N — набор пар $(x^i, y_i)_{i=1}^{N}$, где y_i — истинный ответ на объекте x^i.

Определение 1.9. Переобучение — явление, при котором алгоритм сильно подстраивается под обучающую выборку и при этом не извлекает из нее закономерностей. В результате модель показывает высокое качество на обучении и низкое на тесте.

Определение 1.10. Шум (выбросы) в данных — экстремальные значения в данных, которые сильно отличаются от основной массы наблюдений.

Прежде чем перейти к изучению случайного леса, рассмотрим модель бинарного решающего дерева, у которого:

1. В каждой внутренней вершине m записана функция (предикат) $\beta_m: X \to \{0, 1\}$
2. В каждой листовой вершине записан прогноз $c_m \in Y$

В большинстве случаев используются одномерные предикаты, которые сравнивают значение j-го признака с некоторым порогом t:

$$\beta_m(x, j, t) = [x_j < t]$$

Очевидно, что для любой выборки можно построить дерево, у которого каждый лист будет соответствовать ровно одному объекту выборки. Скорее всего, такое дерево будет переобученным и качество алгоритма на новых данных будет низким. Можно было бы поставить задачу поиска дерева, которое не допускает ошибок на обучающей выборке и при этом содержит наименьшее количество листьев. К сожалению, такая задача является NP-полной, поэтому в машинном обучении применяется жадный способ построения дерева — от корня к листьям.

Определение 1.11. Максимальная глубина дерева — количество вершин в пути от корня дерева до самого удаленного листа.

Опишем жадный алгоритм построения бинарного дерева. Пускай в вершину m попала выборка X_m, необходимо подобрать параметры jи t предиката β_m так, чтобы минимизировать заданный заранее критерий ошибки $Q(X_m, j, t)$. Так как признаков конечное число, а из всех возможных значений порога t имеет смысл рассматривать только те, при которых получаются различные разбиения, параметры можно подобрать перебором. При этом можно показать, что значений порога t столько, сколько различных значений признака x_j на обучающей выборке. После того, как параметры были подобраны, вершине ставится в соответствие предикат

$\beta_m(x, j_m, t_m) = [x_{j_m} < t_m]$, а выборка X_m делится на два подмножества — левое и правое поддерево:

$$X_l = \{x \in X_m \mid [x_{j_m} < t_m]\}, X_r = \{x \in X_m \mid [x_{j_m} \geq t_m]\}.$$

Как было сказано ранее, выбранные параметры должны минимизировать критерий ошибки, который выглядит следующим образом:

$$Q(X_m, j, t) = \frac{|X_l|}{|X_m|} H(X_l) + \frac{|X_r|}{|X_m|} H(X_r)$$

Функция $H(X)$ называется критерием информативности, ее значение характеризует разброс ответов в X. В случае регрессии разброс ответов — это дисперсия, поэтому критерий информативности принимает вид:

$$H(X) = \frac{1}{|X|} \sum_{i \in X} (y_i - \bar{y}(X))^2, \text{ где } \bar{y}(X) = \frac{1}{|X|} \sum_{i \in X} y_i$$

Процесс ветвления продолжается рекурсивно до тех пор, пока не выполнится условие останова. Обычно в качестве критерия останова используют либо ограничение на количество объектов в листе (по умолчанию это единица), либо ограничение на глубину дерева. Последний критерий считается довольно грубым, однако он хорошо зарекомендовал себя в построении композиций, когда много решающих деревьев объединяются в один сложный алгоритм. Подробнее композиции будут рассмотрены в следующем параграфе.

Вершина, для которой выполняется условие останова, объявляется листом. Если в лист попала некоторая подвыборка X_m исходной выборки, требуется определить для нее оптимальный прогноз. В задачах регрессии оптимальным считается средний ответ по данной подвыборке:

$$a_m = \frac{1}{|X_m|} \sum_{i \in X_m} y_i.$$

Таким образом, жадный алгоритм построения бинарного решающего дерева состоит из следующих шагов:

1. Выбор критерия ошибки. В задачах регрессии чаще всего выбирают среднеквадратичную ошибку.
2. Выбор критерия останова
3. Подбор оптимальных параметров j и t в каждой внутренней вершине
4. Вычисление значения листовой вершины, если в ней оказалось больше одного объекта

Теперь, когда с алгоритмом построения мы разобрались, следует уделить внимание признакам, которые мы рассматриваем в вершинах дерева. В машинном обучении признаки можно разделить на вещественные и категориальные. Очевидно, что предикат в виде неравенства, рассмотренный ранее, не применим к категориальным признакам. Поэтому при построении бинарных деревьев такие признаки необходимо предварительно обработать. Рассмотрим один из возможных способов это сделать.

Пускай, нам нужно произвести разбиение в вершине m, при этом категориальный признак x_j может принимать значения $C = \{c_1, ..., c_n\}$. Отсортируем значения по следующему принципу:

$$\frac{\sum_{i \in X_m} [x_{j_i} = c_1] y_i}{\sum_{i \in X_m} [x_{j_i} = c_1]} \leq ... \leq \frac{\sum_{i \in X_m} [x_{j_i} = c_n] y_i}{\sum_{i \in X_m} [x_{j_i} = c_n]}$$

Фактически, сортировка происходит по возрастанию доли объектов с соответствующим значением признака. После этого категории заменяются на натуральные числа от 1 до n, таким образом происходит преобразование в вещественные признаки.

1.3 Случайный лес

Несмотря на то, что решающие деревья способны восстанавливать сложные зависимости, они обладают парой весомых недостатков: легко переобучаются, кроме этого их структура сильно меняется при малейшем изменении обучающей выборки. По этим причинам решающее дерево как самостоятельный алгоритм используется крайне редко, чаще деревья объединяют в композиции для построения одного непереобученного алгоритма.

Определение 1.12. Композиция — это объединение N алгоритмов $b_1(x),\ldots, b_N(x)$ в один, при этом алгоритмы $b_1(x),\ldots, b_N(x)$ называются *базовыми*. Идея заключается в независимом обучении базовых алгоритмов и последующем усреднении их ответов. Для задачи регрессии ответ композиции $a(x)$ будет иметь вид:

$$a(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x).$$

Определение 1.13. Смещение — отклонение прогноза данной модели от прогноза идеальной модели.

Определение 1.14. Разброс — дисперсия ответов базовых алгоритмов.

Для начала разберемся, как обучить Nбазовых алгоритмов. Очевидно, нельзя обучать алгоритмы на всей обучающей выборке, в противном случае они получатся одинаковыми, и создание композиции потеряет смысл. Поскольку решающие деревья сильно меняются при небольших изменениях в обучающей выборке, одним из способов сделать базовые алгоритмы различными — использовать рандомизацию при построении обучающей выборки. В машинном обучении популярным подходом является бутстрап, заключается он в следующем: из обучающей выборки длины l выбираются с возвращением l объектов. Таким образом, выборка по-прежнему будет иметь длину l, но некоторые объекты в ней будут повторяться, а некоторые

объекты исходной выборки в нее не попадут. В среднем в бутстрапированной выборке содержится около 63% уникальных объектов исходной выборки.

Решающие деревья характеризуются низким смещением и высоким разбросом. При построении композиции смещение все еще будет низким, при этом разброс уменьшается и вычисляется как

$$\begin{pmatrix}\text{разброс}\\ \text{композиции}\end{pmatrix} = \frac{1}{N}\begin{pmatrix}\text{разброс одного}\\ \text{базового алгоритма}\end{pmatrix} + \begin{pmatrix}\text{корреляция между}\\ \text{базовыми алгоритмами}\end{pmatrix}$$

Видно, что разброс композиции тем меньше, чем ниже корреляция базовых алгоритмов. Использования бутстрапированных выборок в данном случае недостаточно, поэтому имеет смысл рандомизировать сам процесс построения дерева. Как было сказано ранее, в каждой вершине происходит подбор параметров разбиения j и t. Если в задаче поиска оптимальных параметров выбирать j из случайного подмножества признаков размера q, корреляция заметно снизится. В задаче регрессии рекомендованное значение параметра $q = \frac{d}{3}$, где d — общее число признаков.

Перейдем к алгоритму построения случайного леса:

1. Построить с помощью бутстрапа N случайных выборок $\overline{X_n},\ n = \overline{1, N}$
2. Получившиеся на первом шаге обучающие выборки используются для построения соответствующих решающих деревьев $b_n,\ n = \overline{1, N}$
3. Построенные деревья объединяются в композицию

После построения композиции необходимо оценить качество полученного алгоритма. В задачах регрессии наиболее часто используются следующие метрики:

1. Коэффициент детерминации R^2— доля дисперсии зависимой переменной, объясняемая рассматриваемой моделью:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2},$$

где y_i, $\hat{y}_i$— фактические и прогнозируемые значения объясняемой переменной соответственно, а $\bar{y}_i$ — выборочное среднее. При оценке регрессионных моделей коэффициент R^2 интерпретируется как соответствие модели данным. Для адекватных моделей значения коэффициента лежат в промежутке от 0 до 1. В общем случае хорошими принято считать модели, для которых $R^2 > 0.8$.

2. RMSE — корень из среднеквадратической ошибки. Квадратный корень извлекается с целью выравнивания масштаба ошибок и объясняемой переменной:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

3. MAE — средняя абсолютная ошибка. Главное преимущество данной метрики перед RMSE заключается в меньшей чувствительности к шуму в данных, то есть одно очень плохое предсказание не сильно увеличит значение ошибки модели. Вычисляется метрика как

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|.$$

Важно отметить, что оценивать качество модели необходимо на новых данных, так как на обучающей выборке модель ожидаемо покажет высокий результат. Поэтому предлагается разделить исходную выборку на

обучающую и тестовую в соотношении 80:20 соответственно. После этого на основе обучающей выборки строится модель, а на тестовой оценивается ее качество.

Глава 2. Построение модели случайного леса

2.1 Предварительная обработка и анализ данных

Для решения поставленной в данной работе задачи была сформирована выборка автомобилей, проданных в период с 1 января 2020 по 28 февраля 2021 года. Все автомобили размещались на классифайде Авито и были проданы в течение 90 дней, при этом услуги продвижения на классифайде к ним не применялись.

Начнем с разделения признаков на категориальные и вещественные, а также с анализа зависимости целевой переменной (количества просмотров) от объясняющих.

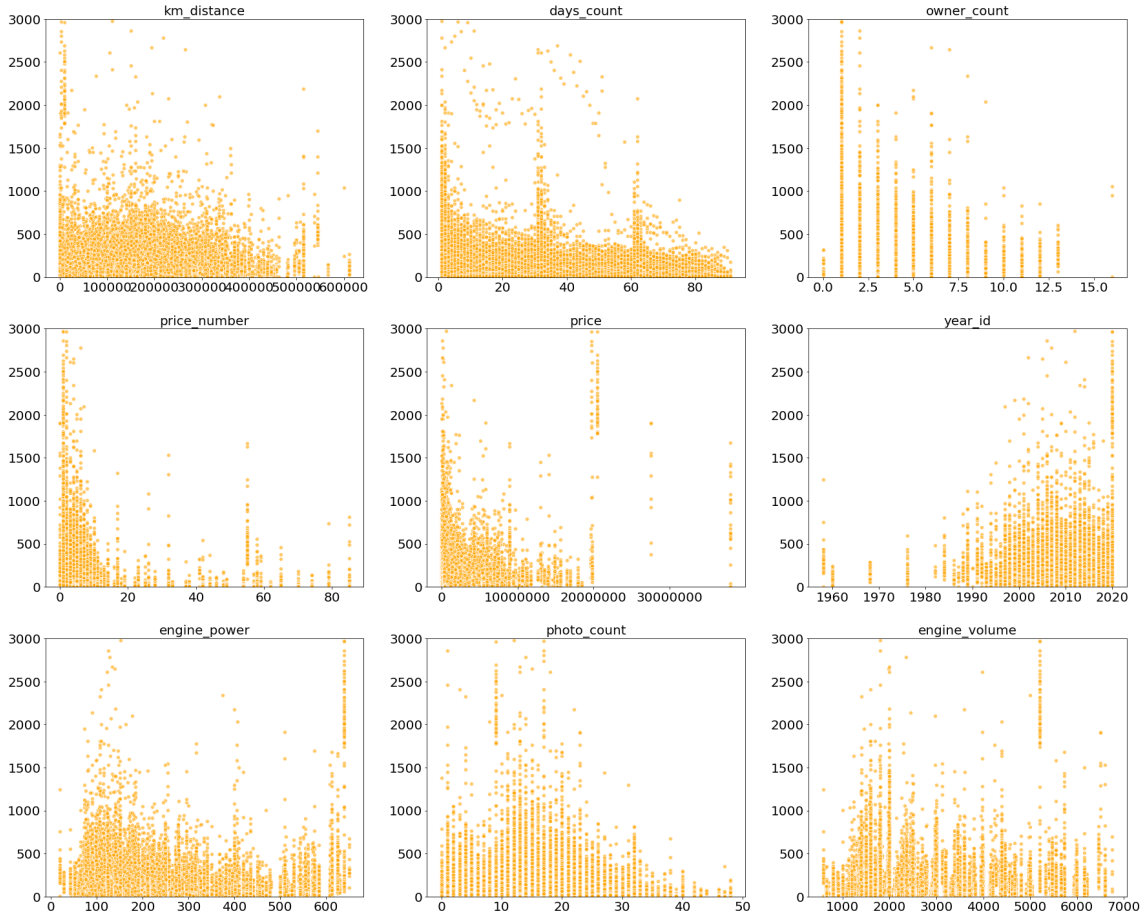

Рис. 1: Зависимость количества просмотров от вещественных признаков

Из графиков выше можно сделать следующие выводы:

1. Распределение просмотров по дням с начала продажи (Рис. 1 «days_count») похоже на экспоненциальное, также примерно каждые 30 дней наблюдается резкое увеличение просмотров, связанное с необходимостью обновлять объявления по истечении их срока публикации. Кроме того, на графике видно довольно высокую зашумленность данных.
2. На графике с годом выпуска автомобиля (Рис. 1 «year_id») можно заметить, что в выборке присутствуют машины 1960-1980 годов — это также является шумом.
3. Особой популярностью пользуются автомобили с количеством предыдущих владельцев 1-2 (Рис. 1 «owner_count»).

4. Наибольшее количество просмотров получают машины после переоценки, 2-3 по счету цена является для потребителя наиболее привлекательной (Рис. 1 «price_number»).
5. Остальные признаки выглядят логично: интересны те машины, у которых меньше пробег (Рис. 1 «km_distance»), позже год выпуска, а также присутствует около 15 фотографий в объявлении (Рис. 1 «photo_count»).

Теперь рассмотрим зависимость просмотров от категориальных признаков, таких как дилер («user_id»), разместивший объявление, марка («brand_id») и класс («body_type») автомобиля, цвет корпуса («body_color»). Хорошим способом визуализации категориальных признаков является диаграмма размаха, которая показывает медиану, верхний и нижний квартили, минимальное и максимальное значение выборки, и, что важно, на ней хорошо видны выбросы (обозначаются отдельными точками). На графике ниже (Рис. 2) можно увидеть, что некоторые дилеры и определенные марки машин более популярны. Аналогично графикам для вещественных признаков, наблюдается высокая зашумленность данных.

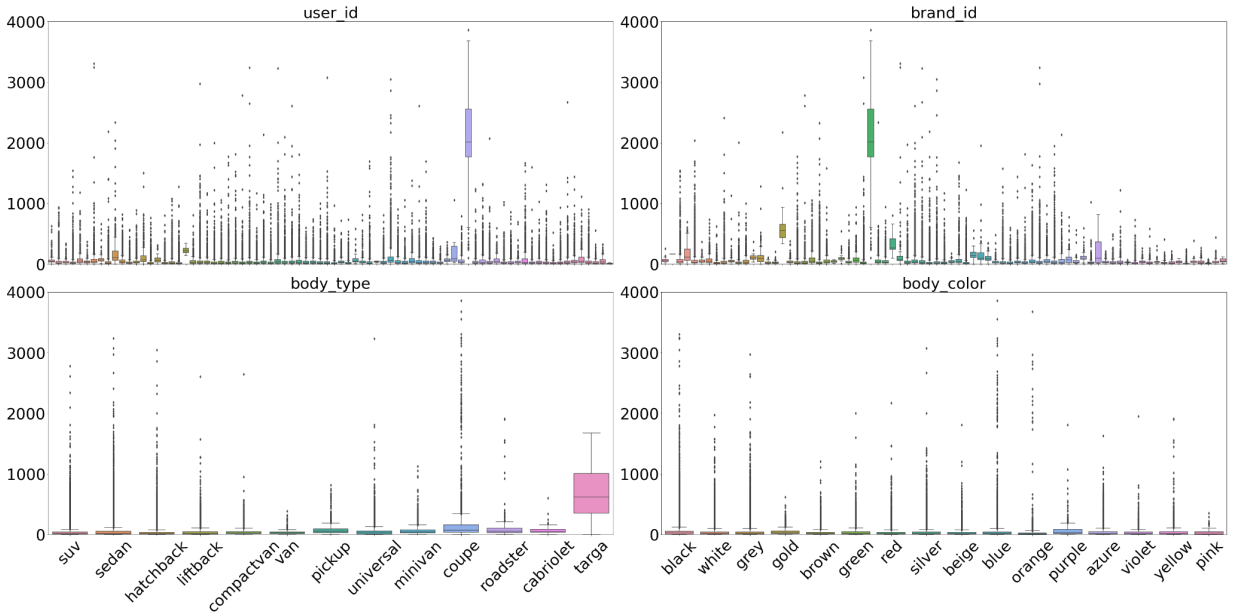

Рис. 2: Зависимость количества просмотров от категориальных признаков

Кроме шума в данных присутствуют пропущенные значения, которые необходимо обработать отдельно. В случае вещественных признаков пропуски можно заполнить средним выборочным значением, а в случае категориальных пропущенные значения интерпретируются как еще одно значение признака, что позволяет сохранить информацию о наличии пропусков.

Теперь закодируем категориальные признаки, разделим выборку на обучающую (X_train, y_train) и тестовую (X_test, y_test), после чего построим модель случайного леса с настройками по умолчанию (Рис. 3):

```
# Создаем модель рандомного леса с базовыми настройками
frm = ensemble.RandomForestRegressor()
# Обучаем модель
frm.fit(X_train, y_train)
# Измеряем качество
predictions = frm.predict(X_test)
print('MAE: ', round(metrics.mean_absolute_error(y_test, predictions), 3))
print('RMSE: ', round(np.sqrt(metrics.mean_squared_error(y_test, predictions)), 3))
print('R2: ', round(frm.score(X_test, y_test), 3))
```

```
MAE:  18.553
RMSE:  52.164
R2:  0.642
```

Рис. 3: Качество модели с базовыми настройками на данных с шумом

Как видно, коэффициент детерминации для такой модели довольно низкий, чтобы улучшить качество модели попробуем снизить уровень шума в данных. Для начала избавимся от выбросов, которые были замечены при анализе зависимости целевой переменной от объясняющих: удалим из выборки машины с годом выпуска раньше 1990 и пробегом меньше 1000 километров. Однако это лишь малая часть шума в данных, чтобы действительно повысить их качество оставим только те машины, количество просмотров которых находится между 20-м и 80-м перцентилем. После этого еще раз обучим модель с базовыми настройками. Коэффициент детерминации заметно увеличился (Рис. 4):

```
frm = ensemble.RandomForestRegressor()
frm.fit(X_train, y_train)
# Измеряем качество
predictions = frm.predict(X_test)
print('MAE: ', round(metrics.mean_absolute_error(y_test, predictions), 3))
print('RMSE: ', round(np.sqrt(metrics.mean_squared_error(y_test, predictions)), 3))
print('R2: ', round(frm.score(X_test, y_test), 3))
```

```
MAE:  6.663
RMSE:  10.146
R2:  0.939
```

Рис. 4: Качество модели с базовыми настройками на обработанных данных

На данном этапе также можно произвести отбор признаков. Из графика ниже (Рис. 5) очевидно, что такие признаки как тип коробки передач (gearbox_type), количество передач (gearbox_gear_count), количество предыдущих владельцев (owner_count) и класс автомобиля (body_type) практически не влияют на целевую переменную. Уменьшение количества признаков повышает скорость обучения алгоритма, что крайне важно при работе с большими данными.

Кроме этого, данный график показывает какие признаки являются ключевыми для модели: мощность двигателя (engine_power), цена (price) и номер дня с начала продажи (days_count).

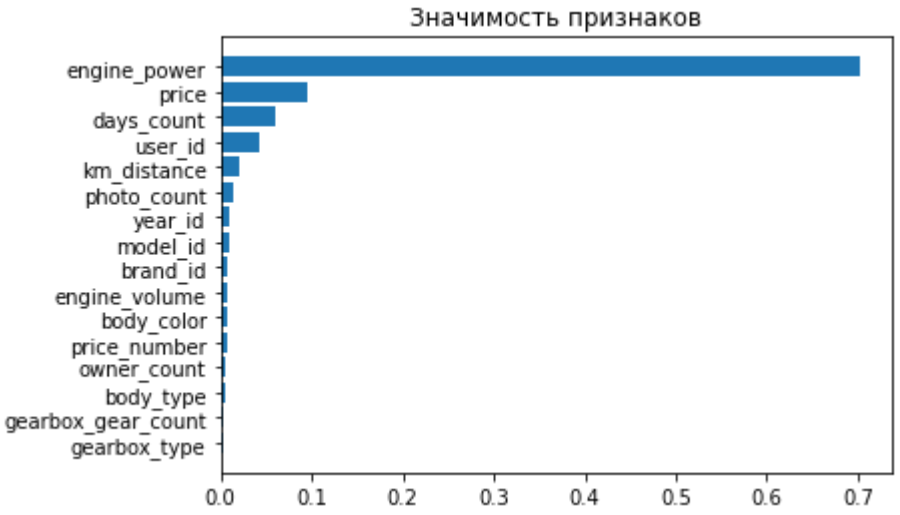

Рис.5: Сравнение значимости признаков

2.2 Настройка модели

После обработки данных стоит попробовать улучшить качество модели с помощью настройки ее параметров[1]. Рассмотрим несколько ключевых параметров случайного леса. В скобках указаны значения параметров по умолчанию:

1. n_estimators (= 10) — количество базовых алгоритмов
2. max_depth — максимальная глубина дерева. Как было сказано ранее, это один из возможных критериев останова, если его не настраивать, дерево будет строиться до тех пор, пока количество объектов в вершине не станет меньше минимального числа объектов, необходимых для деления вершины.
3. min_samples_split (= 2) — минимальное количество объектов, необходимое для осуществления деления вершины.
4. min_samples_leaf (= 1) — минимальное число объектов в листовой вершине
5. max_features — количество признаков, рассматриваемых при поиске оптимального деления вершины. По умолчанию рассматриваются все признаки, однако существует рекомендованное значение, равное трети от общего количества признаков.

Начнем с определения оптимальной глубины деревьев в лесу. По рисунку (Рис. 6) видно, что искомое значение находится около 29, однако следует рассмотреть также ближайшие значения. В результате оптимальной глубиной деревьев оказалось 33.

[1] ссылка на код программы: https://github.com/dowhile26/Diploma

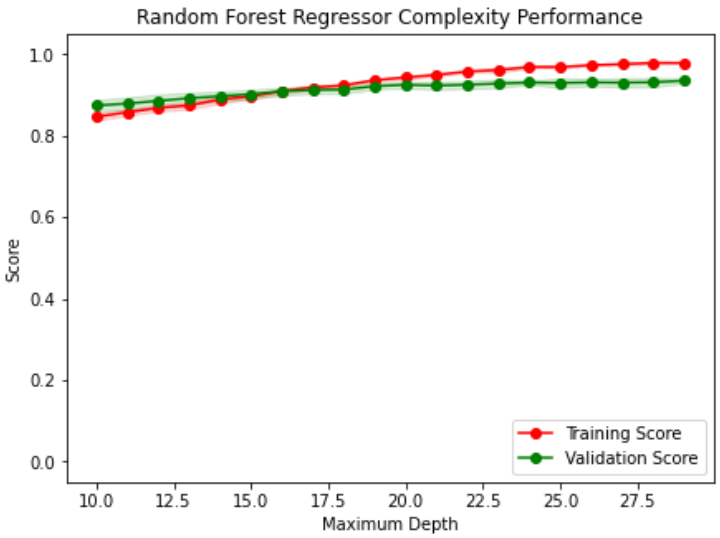

Рис. 6: Зависимость качества случайного леса от глубины деревьев в нем

Теперь с помощью поиска по сетке значений подберем оптимальные параметры для модели. Данный метод с помощью перебора различных значений параметров ищет их комбинацию, которая дает наилучший результат. При этом диапазон значений, среди которых происходит поиск оптимального, необходимо задать самостоятельно. Из графиков ниже (Рис. 7) можно заключить, что:

1. Оптимальное количество базовых алгоритмов в случайном лесу 100, так как при дальнейшем увеличении данного параметра качество не меняется.
2. Увеличение минимального количества объектов, необходимых для разделения вершины, а также объектов в листе приводит к снижению качества.
3. Оптимальным количеством признаков для выбора наилучшего деления вершины является 4.

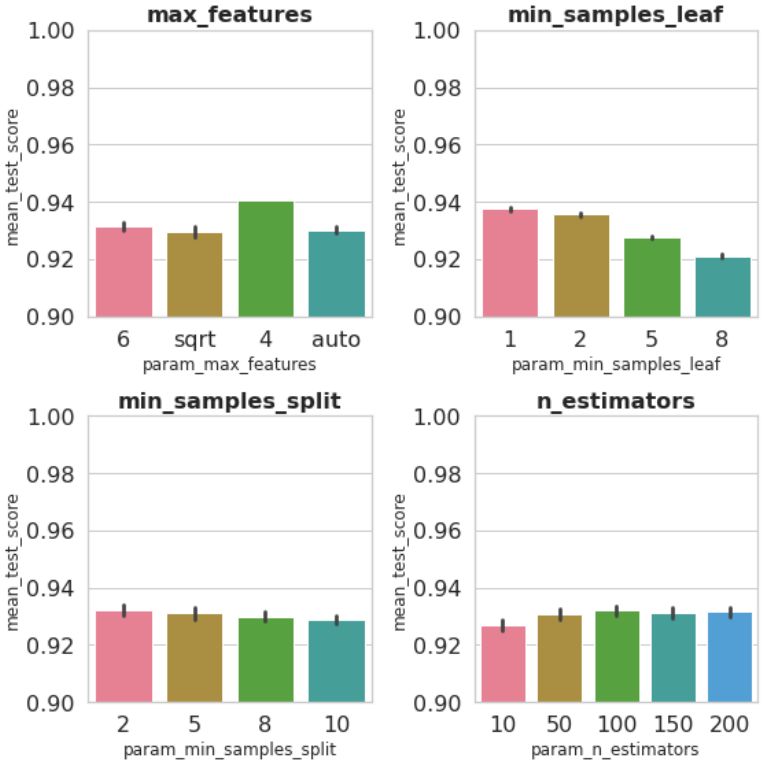

Рис. 7: Зависимость качества модели при различных значениях параметров

В результате поиска по сетке оптимальные параметры оказались следующими:

1. n_estimators = 100
2. max_depth = 33
3. max_features = 4
4. min_samples_split = 2
5. min_samples_leaf = 1

Обучим модель с подобранными параметрами. Как видно (Рис. 8), качество выросло незначительно. Вполне вероятно, что на имеющихся данных построить более точную модель невозможно:

```
frm_c3 = ensemble.RandomForestRegressor(n_estimators=100, max_depth = 33, max_features=4)
frm_c3.fit(X_train, y_train)
predictions = frm_c3.predict(X_test)
print('MAE: ', round(metrics.mean_absolute_error(y_test, predictions), 3))
print('RMSE: ', round(np.sqrt(metrics.mean_squared_error(y_test, predictions)), 3))
print('R2: ', round(frm_c3.score(X_test, y_test), 3))
```

```
MAE:  6.415
RMSE:  10.036
R2:  0.946
```

Рис. 8: Случайный лес с настроенными параметрами

Несмотря на то, что метрики MAE и RMSE имеют те же единицы измерения, что и целевая переменная, оценить, насколько такая ошибка критична довольно сложно. Рассмотрим работу модели на конкретном примере: будем предсказывать количество просмотров на каждый день, после чего проверим, будут ли звонки по данному автомобилю в течение первой недели.

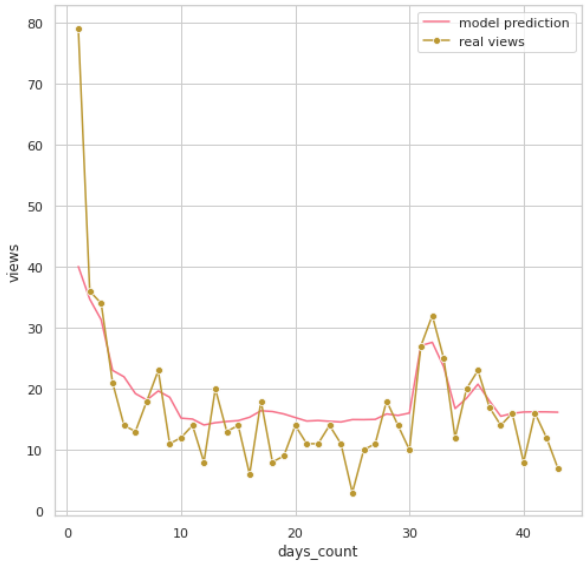

Рис. 9: Сравнение предсказаний модели с реальными данными

Как и показывал коэффициент детерминации, модель довольно хорошо описывает данные, улавливая основные зависимости. Теперь сравним количество предсказанных и фактических просмотров с последующей конверсией в звонки. Из постановки задачи имеет смысл сравнивать значения полученные за период в 7, 10 и 30 дней:

	days_count	views_pred	views_real	calls_pred	calls_real
0	7	188	215	2	2
1	10	241	261	3	3
2	30	545	498	7	6

Рис. 10: Просмотры с конверсией в звонки

Поскольку в среднем машина продается за 6-7 звонков, расхождение в результатах за период в 30 дней не повлияет на интерпретацию результатов модели. Из этого можно сделать вывод, что полученные значения метрик качества являются приемлемыми для данной задачи.

Результаты, полученные моделью в рассмотренном примере, интерпретируются следующим образом: в первые 7 дней будет получено 2 звонка, а за 30 дней прогнозируется целевое количество звонков для дилера, значит, применять переоценку нет необходимости, так как скорее всего машина будет продана в первый месяц, то есть за целевой срок оборачиваемости.

Выводы

В данной работе изучены теоретические аспекты построения модели случайного леса, а также рассмотрены ключевые моменты работы автодилеров, в частности размещение на классифайде.

В практической части установлено, что кроме номера дня с начала продажи автомобиля на количество просмотров объявления больше всего влияют такие характеристики как мощность мотора, цена и пробег. Также произведена предварительная обработка собранных данных и построена модель случайного леса. Далее модель настроена, после чего проведена оценка качества ее работы, а полученные результаты интерпретированы.

Исходя из рассмотренного примера, можно сделать вывод, что качество полученной в результате работы модели удовлетворяет требованиям поставленной задачи.

Заключение

На основе изученной теории в работе получена модель для предсказания количества просмотров объявления по автомобилю в зависимости от номера дня с начала продаж. Также выявлены наиболее значимые характеристики автомобилей, влияющие на количество просмотров объявления.

В дальнейшем результат работы может быть использован при построении модели динамического ценообразования на рынке автомобилей.

Список литературы

1. Hastie, Tibshirani, Friedman. The elements of statistical learning // Stanford, California, 2008. p.587-601.
2. Murphy. Machine learning a probabilistic perspective // The MIT Press Cambridge, Massachusetts, 2012. p.543-550.
3. Harrington. Machine Learning in Action // Manning Publications, Shelter Island, 2012. p.179-205.
4. МФТИ, Яндекс. Обучение на размеченных данных (курс лекций) // https://www.coursera.org/learn/supervised-learning
5. Richert, Coelho. Building Machine Learning Systems with Python // Packt Publishing, Birmingham, 2015. p.1-32.
6. Philipp Probst, Marvin Wright, Anne-Laure Boulesteix. Hyperparameters and Tuning Strategies for Random Forest // WIREs Data Mining and Knowledge Discovery, Vol. 9, No. 3.
7. Philipp Probst, Anne-Laure Boulesteix, Bernd Bischl. Tunability: Importance of Hyperparameters of Machine Learning Algorithms // Journal of Machine Learning Research, Vol. 20.
8. Gérard Biau. Analysis of a Random Forests Model // Journal of Machine Learning Research, Vol. 13.
9. James Bergstra, Yoshua Bengio. Random Search for Hyper-Parameter Optimization // Journal of Machine Learning Research, Vol. 13.
10. Lucas Mentch, Siyu Zhou. Randomization as Regularization: A Degrees of Freedom Explanation for Random Forest Success // Journal of Machine Learning Research, Vol. 21.