Титульный лист материалов по дисциплине

(заполняется по каждому виду учебного материала)

ДИСЦИПЛИНА	**Технологии извлечения знаний из больших данных** (полное наименование дисциплины без сокращений)
ИНСТИТУТ	ИКБ
КАФЕДРА	**КБ-4 «Интеллектуальные системы информационной безопасности»** полное наименование кафедры)
ВИД УЧЕБНОГО МАТЕРИАЛА	**Лекция** (в соответствии с пп.1-11)
ПРЕПОДАВАТЕЛЬ	**Никонов В.В.** (фамилия, имя, отчество)
СЕМЕСТР	**3 семестр 2023/2024 уч. года** (указать семестр обучения, учебный год)

Базовая работа с табличными данными

Задача регрессии. Введение

Данные представлены в виде таблицы, каждая строка — объект, столбцы — признаки объектов (все объекты описываются одним и тем же набором признаков, но значения признаков у каждого объекта свои). В задаче классификации также имеется отдельный столбец с классами объектов. Этот столбец еще называют **столбцом значений целевой переменной** (англ. target), то есть величины, которую нужно предсказывать.

В задаче регрессии данные имеют такой же вид, но целевая переменная числовая. Иными словами,

задача регрессии состоит в том, чтобы на основании различных признаков предсказать вещественный ответ, т.е. для каждого объекта нужно предсказать число.

Например, в задаче предсказания спроса на товар нужно предсказать, какое количество единиц товара потребуется в торговой точке в определенный промежуток времени, например в конкретную неделю. Другой пример — предсказание стоимости квартиры (стоимость в рублях — числовая величина). Еще один пример — предсказание возраста человека по фотографии (возраст — число). Отличие задачи классификации от задачи регрессии выглядит незначительным и в принципе действительно таковым является, однако, как мы увидим ниже, алгоритмы предсказания и обучения будут работать по-разному для этих двух задач.

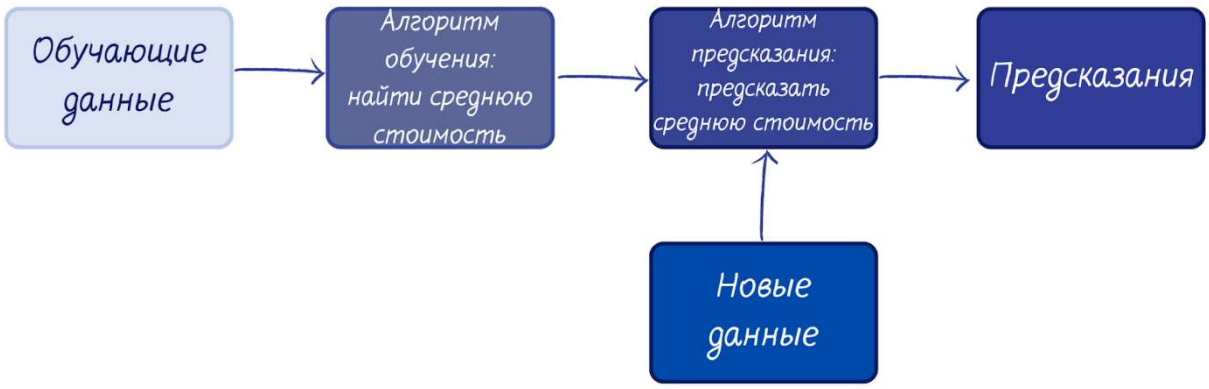
Как и в задаче классификации, потребуются обучающие данные с признаками объектов и известными значениями числовой целевой переменной. Для примера будем рассматривать задачу предсказания стоимости квартиры с данными следующего вида:

Площадь ($м^2$)	**Этаж**	**Число комнат**	**Число лет с последнего ремонта**	**Стоимость (млн)**
115	3	4	2	46.0
55	5	2	5	10.3
72	6	3	12	17.7

55	20	1	0	32.0

В этой задаче объектом является квартира, признаки: площадь, число комнат, этаж и число лет с последнего ремонта, целевая переменная — стоимость квартиры. На основе обучающих данных алгоритм обучения составляет алгоритм предсказания. Алгоритм предсказания по признакам новой квартиры (площадь, число комнат, этаж и информация о ремонте) определяет ее стоимость. Отметим, что хотя мы рассматриваем конкретную задачу для примера, компьютер видит данные как набор чисел, не зная, что за этими числами стоит. Соответственно, все, что мы обсуждаем на примере задачи предсказания стоимости квартиры, можно выполнять для любой другой задачи регрессии, с другими объектами, признаками и целевой переменной — это будет просто другой набор чисел.

Итак, предположим, мы формализовали задачу в виде задачи регрессии и собрали обучающие данные. Сосредоточимся на том, как на основе обучающих данных выполнить обучение и создать алгоритм предсказания. Самый простой (и на самом деле бесполезный) алгоритм предсказания, который можно придумать, — это такой, который для всех объектов предсказывает одно и то же. В нашей задаче это означает, что для всех квартир предсказывается одна и та же стоимость. Определить эту стоимость можно по обучающим данным, например, можно использовать среднюю стоимость квартир в обучающих данных — в вычислении этой средней стоимости и будет заключаться обучение; все лучше, чем предсказывать, например, стоимость ноль.



Конечно, описанный алгоритм не принесет никакой пользы на практике, потому что никак не учитывает особенности объектов, но его можно использовать в качестве базового решения, бейзлайна (англ. baseline): более сложный алгоритм не должен делать предсказания хуже, чем базовое решение. Понять, насколько хорошие предсказания делает алгоритм, можно, измерив среднюю ошибку — об этом мы поговорим в блоке «Метрики».

Линейные модели

Самый известный метод регрессии — это линейные модели, мы остановимся на них подробно. Основной механизм предсказания с помощью линейной модели формулируется следующим образом: необходимо умножить значения всех признаков на веса и сложить.

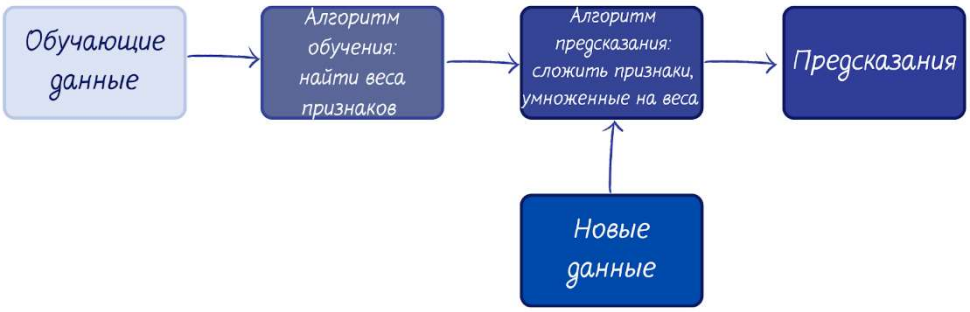
Предположим, что мы хотим предсказать стоимость квартиры со следующими значениями признаков:

Площадь ($м^2$)	**Этаж**	**Число комнат**	**Число лет с последнего ремонта**
70	2	3	5

Также предположим, что мы знаем веса признаков: 0.25 для площади, 1.8 для этажа, 0.5 для числа комнат и (–0.2) для числа лет со дня ремонта. Получается, что вес признака задает вклад признака в предсказание: например, вес 0,25 означает, что каждый квадратный метр квартиры добавляет 0,25 единицы в ее стоимость, а вес 0,5 — что каждая комната в квартире добавляет 0,5 единиц в ее стоимость. Вес -0,2 означает, что каждый год, прошедший с последнего ремонта, вычитает 0,2 единицы из стоимости квартиры.

Тогда будет предсказана стоимость $0,25 \cdot 70 + 1,8 \cdot 2 + 0,5 \cdot 3 - 0,2 \cdot 5 = 21,$ условных единиц, т.е. мы умножили признаки на веса и сложили. Большое по модулю значение веса означает, что соответствующий признак вносит сильный вклад в предсказание, а вес, близкий к нулю, означает, что соответствующий признак практически не влияет на предсказание.

Так выполняются предсказания в линейных моделях, однако не решен вопрос с подбором значения весов. Именно их мы будем искать в процессе обучения по данным: алгоритм обучения автоматически подберет значения весов, такие, что ошибка предсказания с этими весами будет меньше, чем с другими весами. При этом ошибка измеряется по обучающим данным: чем больше обучающих объектов (квартир в обучающих данных), тем в большем количестве случаев будет хорошо работать обученный алгоритм. Если обучающих данных достаточно много, найденные в процессе обучения веса будут точно лучше, чем веса, которые предложил бы использовать специалист по недвижимости, потому что алгоритм обучения «видел» гораздо больше примеров, чем специалист.



Преимущества линейных моделей

Линейные модели легко интерпретируемы: человеку легко понять, почему для объекта выполнено именно такое предсказание. Линейные модели, как правило, решают задачу с приемлемым уровнем качества, однако уступают более мощным алгоритмам, ансамблям решающих деревьев и нейронным сетям, которые мы обсудим во второй половине курса. С другой стороны, качество линейных моделей можно значительно повысить, придумав новые признаки, вычисляемые на основе исходных признаков (например, добавив квадраты признаков), — при этом свойство интерпретируемости сохраняется. Благодаря своей интерпретируемости линейные модели очень популярны в бизнес-задачах, например в кредитном скоринге.

Линейные модели для решения задачи регрессии: сложив значения признаков, умноженные на веса, мы получаем число, и в задаче регрессии

нам и нужно предсказать число. Этот метод называется линейная регрессия (LinearRegression). Линейные модели можно использовать и в задаче классификации, однако для этого потребуется преобразовать число в класс. Если классов всего два, можно сравнивать предсказанное числовое значение с порогом: если оно меньше порога, предсказывать один класс, иначе предсказывать второй класс. По такому принципу работает, например, алгоритм SVM. Альтернативно можно предсказывать вероятности классов, такой алгоритм называется логистическая регрессия (LogisticRegression).

В задаче регрессии требуется по признакам объекта предсказать число. Линейные модели подразумевают, что предсказание вычисляется как сумма значений признаков объекта, умноженных на веса. Веса настраиваются по данным в процессе обучения и определяют важность признаков. Популярность линейных моделей обосновывается тем, что их предсказания легко интерпретировать, однако линейные модели не являются рекордсменами по точности предсказаний.

Эффект переобучения регрессии

Ранее мы много говорили об обучающих данных — тех, которые используются для создания алгоритма предсказания. По результатам обучения мы ожидаем, что алгоритм предсказания делает более-менее точные предсказания для объектов из обучающих данных — иначе зачем мы вообще обучали этот алгоритм. Однако может случиться, что алгоритм делает хорошие предсказания только для обучающих данных. Иными словами, алгоритм запомнил, зазубрил классы/числа для обучающих объектов, но не нашел никаких зависимостей между признаками и целевой переменной (классом/числом). Такой алгоритм будет плохо работать на шаге внедрения и называется переобученным.

Вспомним пример с предсказанием стоимости квартир: переобучение означает, что алгоритм хорошо знает стоимости квартир из обучающих данных, но если мы рассматриваем квартиру, хотя бы чуть-чуть

отличающуюся от тех, что входили в обучающие данные, алгоритм предскажет совершенно неверную стоимость.

Чтобы контролировать возникновение переобучения, на практике всегда выделяют два набора данных: обучающие и тестовые. Первый набор используется для обучения алгоритма, а второй — для контроля ошибки обученного алгоритма на новых данных, тех, которые не входили в обучение.