Лекция 3. Задача классификации

Постановка задач классификации

Как мы уже говорили ранее, машинное обучение с учителем является одним из наиболее часто используемых и успешных видов машинного обучения.

Вспомним, что обучение с учителем используется всякий раз, когда мы хотим предсказать определенный результат (ответ) по данному объекту, и у нас есть пары объект-ответ. Мы строим модель машинного обучения на основе этих пар объект-ответ, которые составляют наш обучающий набор данных. Наша цель состоит в том, чтобы получить точные прогнозы для новых, никогда ранее не встречавшихся данных. ИИ с учителем часто требует вмешательства человека, чтобы получить обучающий набор данных, но потом оно автоматизирует и часто ускоряет решение трудоемких или неосуществимых задач.

Есть два основные задачи машинного обучения с учителем: *классификация* и *регрессия*.

Классификация – один из разделов машинного обучения, посвященный решению следующей задачи. Имеется множество объектов (ситуаций), разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется обучающей выборкой. Классовая принадлежность остальных объектов неизвестна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

Классифицировать объект – значит, указать номер (или наименование класса), к которому относится данный объект.

Классификация объекта – номер или наименование класса, выдаваемый алгоритмом классификации в результате его применения к данному конкретному объекту.

В машинном обучении задача классификации относится к разделу обучения с учителем. Существует также обучение без учителя, когда

разделение объектов обучающей выборки на классы не задаётся, и требуется классифицировать объекты только на основе их сходства друг с другом. В этом случае принято говорить о задачах кластеризации или таксономии, и классы называть, соответственно, кластерами или таксонами.

Цель классификации состоит в том, чтобы спрогнозировать метку класса, которая представляет собой выбор из заранее определенного списка возможных вариантов. Классификация иногда разделяется на *бинарную классификацию*, которая является частным случаем разделения на два класса, и *мультиклассовую классификацию*, когда в классификации участвует более двух классов. Бинарную классификацию можно представить как попытку ответить на поставленный вопрос в формате «да/нет». Классификация электронных писем на спам и не-спам является примером бинарной классификации. В данной задаче бинарной классификации ответ «да/нет» дается на вопрос «является ли это электронное письмо спамом?». Пример мультиклассовой классификации – прогнозирование языка веб-сайта. Классами здесь будет заранее определенный список возможных языков.

Самый простой способ отличить классификацию от регрессии – спросить, заложена ли в полученном ответе определенная непрерывность (преемственность). Если полученные результаты непрерывно связаны друг с другом, то решаемая задача является задачей регрессии. Возьмем прогнозирование годового дохода. Здесь ясно видна непрерывность ответа. Разница между годовым доходом в один рубль не существенна, хотя речь идет о разных денежных суммах. Если наш алгоритм предсказывает значение в точности до нескольких рублей, мы не будем настаивать на том, что разница существенна. Наоборот, в задаче распознавании языка веб-сайта (задаче классификации) ответы четко определены. Контент сайта может быть написан либо на одном конкретном языке, либо на другом.

Определение расстояния между объектами класса

Сходство или различие между объектами классификации устанавливается в зависимости от выбранного метрического расстояния между ними. Если каждый объект описывается n свойствами (признаками), то он может быть представлен как точка в n-мерном пространстве, и сходство с другими объектами будет определяться как соответствующее расстояние. При классификации используются различные меры расстояния между объектами.

1. Евклидово расстояние

Это, пожалуй, наиболее часто используемая мера расстояния. Она является геометрическим расстоянием в многомерном пространстве и вычисляется следующим образом:

$$P = \sqrt{\sum_{i=1}^{N}(A_i - B_i)^2}$$

где P – расстояние между объектами А и В;

A_i – значение i-свойства объекта А;

B_i – значение i-свойства объекта B.

Естественное, с геометрической точки зрения, евклидова мера расстояния может оказаться бессмысленной, если признаки измерены в разных единицах. Чтобы исправить положение, прибегают к нормированию каждого признака. Применение евклидова расстояния оправдано в следующих случаях:

- свойства (признаки) объекта однородны по физическому смыслу и одинаково важны для классификации;
- признаковое пространство совпадает с геометрическим пространством.

2. Квадрат евклидова расстояния

Данная мера расстояния используется в тех случаях, когда требуется придать больше значение более отдаленным друг от друга объектам. Это расстояние вычисляется следующим образом:

$$P = \sum_{i=1}^{N} (A_i - B_i)^2.$$

3. Взвешенное евклидово расстояние

Применяется в тех случаях, когда каждому i-свойству удается приписать некоторый «вес» w_i, пропорционально степени важности признака в задаче классификации:

$$P = \sqrt{\sum_{i=1}^{N} w_i (A_i - B_i)^2}.$$

Определение весов, как правило, связано с дополнительными исследованиями, например, организацией опроса экспертов и обработкой их мнений.

4. Хеммингово расстояние

Также называется манхэттенским, сити-блок расстоянием или расстоянием городских кварталов. Это расстояние является разностью по координатам. В большинстве случаев эта мера расстояния приводит к таким же результатам, как и для обычного расстояния Евклида. Однако отметим, что для этой меры влияние отдельных больших разностей (выбросов) уменьшается (так как они не возводятся в квадрат). Хеммингово расстояние вычисляется по формуле:

$$P = \sum_{i=1}^{N} (|A_i| - |B_i|).$$

5. Расстояние Чебышева

Принимает значение наибольшего модуля разности между значениями соответствующих свойств (признаков) объектов:

$$P = \max |A_i - B_i|.$$

Выбор меры расстояния и весов для классифицирующих свойств – очень важный этап, так как от этих процедур зависят состав и количество формируемых классов, а также степень сходства объектов внутри классов.

Метод ближайших соседей

Алгоритм k ближайших соседей, возможно, является самым простым алгоритмом машинного обучения. Построение модели заключается в запоминании обучающего набора данных. Для того, чтобы сделать прогноз для новой точки данных, алгоритм находит ближайшие к ней точки обучающего набора, то есть находит «ближайших соседей».

Метод k-ближайших соседей используется для решения задачи классификации. Он относит объекты к классу, которому принадлежит большинство из k его ближайших соседей в многомерном пространстве признаков. Это один из простейших алгоритмов обучения классификационных моделей.

Число k – это количество соседних объектов в пространстве признаков, которые сравниваются с классифицируемым объектом. Иными словами, если k=10, то каждый объект сравнивается с 10-ю соседями.

В процессе обучения алгоритм просто запоминает все векторы признаков и соответствующие им метки классов. При работе с реальными данными, т.е. наблюдениями, метки класса которых неизвестны, вычисляется расстояние между вектором нового наблюдения и ранее запомненными. Затем выбирается k ближайших к нему векторов, и новый объект относится к классу, которому принадлежит большинство из них.

Приведем алгоритм метода.

Шаг 1. Выберите значение *k* соседей (скажем, k = 5)

Шаг 2: Найдите ближайшую точку данных K (5) для нашей новой точки данных на основе евклидова расстояния (которое мы обсудим позже)

Шаг 3. Среди этих K точек данных подсчитайте точки данных в каждой категории.

Шаг 4. Назначьте новую точку данных категории, которая имеет наибольшее количество соседей с новой точкой данных.

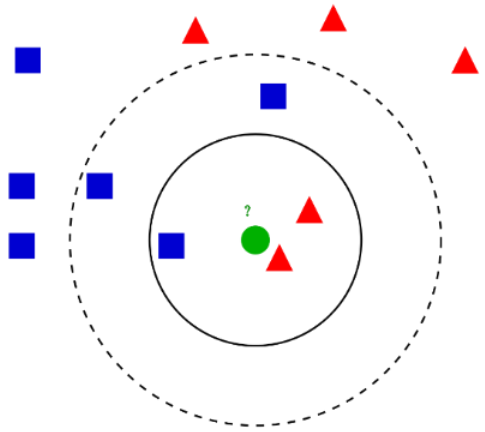

Рисунок 1 – Иллюстрация метода ближайших соседей

Если значения признаков непрерывные, то в качестве меры расстояния между объектами обычно используется расстояние Евклида, а если категориальные, то может использоваться расстояние Хэмминга.

В простейшем варианте алгоритм *k* ближайших соседей рассматривает лишь одного ближайшего соседа – точку обучающего набора, ближе всего расположенную к точке, для которой мы хотим получить прогноз. Прогнозом является ответ, уже известный для данной точки обучающего набора.

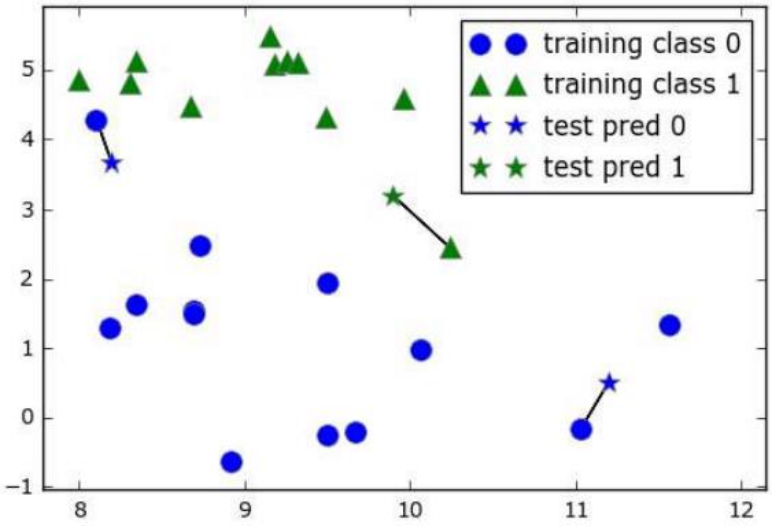

Рисунок 2 – Иллюстрация метода ближайших соседей (k = 1)

На рисунке 2 приведена иллюстрация метода ближайших соседей (k = 1). Здесь мы добавили три новые точки данных, показанные в виде звездочек. Для каждой мы отметили ближайшую точку обучающего набора. Прогноз, который дает алгоритм одного ближайшего соседа – метка этой точки.

Вместо того, чтобы учитывать лишь одного ближайшего соседа, мы можем рассмотреть произвольное количество (*k*) соседей. Отсюда и происходит название алгоритма *k* ближайших соседей. Когда мы рассматриваем более одного соседа, для присвоения метки используется голосование. Это означает, что для каждой точки тестового набора мы подсчитываем количество соседей, относящихся к классу 0, и количество соседей, относящихся к классу 1. Затем мы присваиваем точке тестового набора наиболее часто встречающийся класс: другими словами, мы выбираем класс, набравший большинство среди *k* ближайших соседей. В примере, приведенном ниже, используются три ближайших соседа (рисунок 3).

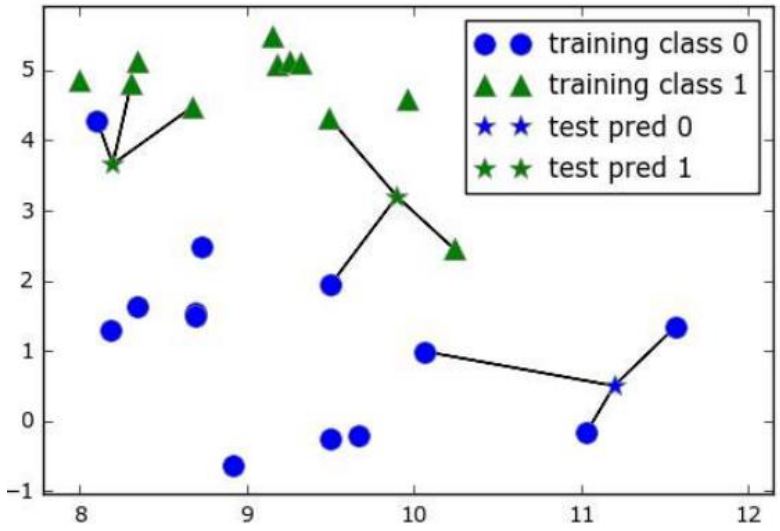

Рисунок 3 – Иллюстрация метода ближайших соседей (k = 3)

Пример

Рассмотрим простой численный пример работы алгоритма (рисунок 4).

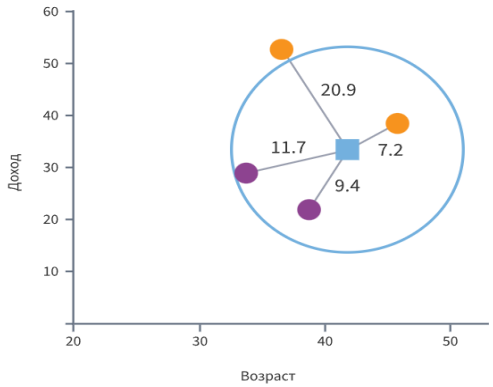

Рисунок 4 – Численный пример метода ближайших соседей

Пусть имеется набор данных о заёмщиках банка часть из которых допустили просрочку по платежу. Признаками являются возраст и среднемесячный доход. Метками класса в поле «Просрочено» будут «Да» и «Нет».

Таблица 1 – Метки класса

X1-возраст	**X2-доход (тыс. руб)**	**Y-Просрочено**
46	40	Нет
36	54	Нет
34	29	Да
38	23	Да

На рисунке оранжевыми кружками представлены объекты класса «Нет», а фиолетовыми класса «Да». Синим квадратом отображается классифицируемый объект (новый заёмщик).

Задача заключается в том, чтобы выполнить классификацию нового заёмщика, для которого X1 = 42 и X2 = 34 с целью оценить возможность просрочки им платежей.

1. Зададим значение параметра k = 3.
2. Рассчитаем расстояние между вектором признаков классифицируемого объекта и векторами обучающих примеров по формуле $D(A,X) = \sqrt{(A_1 - X_1)^2 + (A_2 - X_2)^2}$ и установим для каждого примера его ранг (таблица 2).
3. Исключим из рассмотрения пример, который при k=3 не является соседом и рассмотрим классы оставшихся (таблица 3).

Таблица 2 – Таблица начальных данных

X1-возраст	**X2-доход (тыс. руб)**	**Расстояние**	**Ранг**	**Сосед**
46	40	7.2	1	Нет
36	54	20.9	4	Нет
34	29	9.4	2	Да
38	23	11.7	3	Да

Таблица 3 – Приведенные данные

X1-возраст	**X2-доход (тыс. руб)**	**Расстояние**	**Ранг**	**Сосед**
46	40	7.2	1	Нет
34	29	9.4	2	Да
38	23	11.7	3	Да

Одним из преимуществ метода ближайших соседей является то, что эту модель очень легко интерпретировать и, как правило, этот метод дает приемлемое качество без необходимости использования большого количества настроек. Он является хорошим базовым алгоритмом, который нужно

попробовать в первую очередь, прежде чем рассматривать более сложные методы.

Метрики в задачах классификации

В задачах машинного обучения для оценки качества моделей и сравнения различных алгоритмов используются метрики. Их выбор и анализ – непременная часть работы аналитика. В заключении лекции мы рассмотрим некоторые критерии качества в задачах классификации. Обсудим, что является важным при выборе метрики.

Перед переходом к самим метрикам введем важную концепцию для описания этих метрик – матрицу ошибок (confusion matrix). Матрица ошибок (или матрица неточностей) — это таблица, которая позволяет визуализировать эффективность алгоритма классификации путем сравнения прогнозируемого значения целевой переменной с ее фактическим значением (рисунок 4).

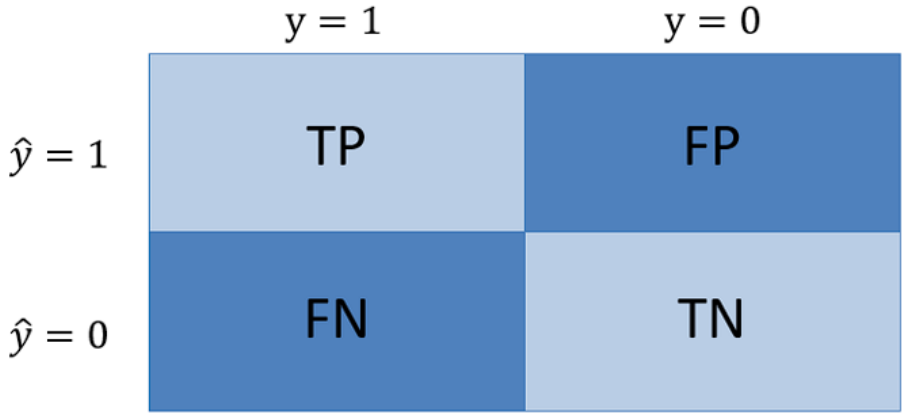

Рисунок 5 – Матрица ошибок

Допустим, что у нас есть два класса и алгоритм, предсказывающий принадлежность каждого объекта одному из этих классов: “+” или “-”, 0 или 1, TRUE или FALSE, собака или кошка.

Тогда матрица ошибок классификации будет выглядеть так, как показано на рисунке 5. Здесь$\hat{y}$— это прогнозные значения алгоритма, а y – истинная метка класса на этом объекте, т.е. исходные данные.

Рассматриваемые нами метрики основаны на использовании следующих исходов: истинно положительные (TP), истинно отрицательные (TN), ложно положительные (FP) и ложно отрицательные (FN). Ложно положительный и

истинно отрицательный исход еще называют ошибками первого и второго рода соответственно. Таким образом, ошибки классификации бывают двух видов: False Negative (FN) и False Positive (FP).

Например, ваша цель — предсказать, является ли домашнее животное собакой или кошкой, на основе некоторых физических и поведенческих атрибутов. Если имеется тестовый набор данных, содержащий 30 собак и 20 кошек, то матрица ошибок может быть похожа на следующую иллюстрацию (рисунок 6).

		Предсказание модели		
		Собака	**Кошка**	
Реальные данные	**Собака**	**24 (TP)**	**6 (FN)**	30
	Кошка	**2 (FN)**	**18 (TN)**	20

Рисунок 6 – Пример матрицы ошибок

Числа в голубых ячейках представляют собой правильные прогнозы. Как можно видеть, модель правильно прогнозируется более высокий процент фактических кошек.

Теперь, рассмотрим основные метрики классификации.

Доля правильных ответов. Интуитивно понятной, очевидной и почти неиспользуемой метрикой является accuracy — доля правильных ответов алгоритма:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

Эта метрика бесполезна в задачах, когда классы сильно разнятся в размерах. Это легко показать на примере. Допустим, мы хотим оценить работу спам-фильтра почты. У нас есть 100 не-спам писем, 90 из которых наш классификатор определил верно (True Negative = 90, False Positive = 10), и 10 спам-писем, 5 из которых классификатор также определил верно (True Positive = 5, False Negative = 5). Тогда доля правильных ответов будет равна 86,4.

Однако если мы просто будем предсказывать все письма как не-спам, то получим более высокую долю правильных ответов = 90,9.

Precision, recall. Для оценки качества работы алгоритма на каждом из классов по отдельности введем метрики precision (точность) и recall (полнота):

$$Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN}.$$

Точность можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а полнота показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм. Именно введение показателя точности не позволяет нам записывать все объекты в один класс, так как в этом случае мы получаем рост уровня False Positive. Recall демонстрирует способность алгоритма обнаруживать данный класс вообще, а precision — способность отличать этот класс от других классов. Precision и recall не зависят, в отличие от доли правильных ответов (первого показателя), от соотношения классов и потому применимы в условиях несбалансированных выборок.

F-мера. В том случае, если точность и полнота являются одинаково значимыми, то для получения оценки результатов можно использовать их среднее гармоническое. Понятно, что чем выше полнота и точность, тем наверное лучше. Но в реальной жизни максимальная точность и полнота практически не достижимы одновременно и приходится искать некий баланс. Поэтому, хотелось бы иметь некую метрику, которая объединяла бы в себе информацию о точности и полноте нашего алгоритма. F-мера представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю:

$$F = 2\frac{Precision * Recall}{Precision + Recall}.$$

Формула выше придает одинаковый вес точности и полноте, поэтому F-мера будет падать одинаково при уменьшении и точности и полноты. Возможно рассчитать F-меру придав различный вес точности и полноте, если вы осознанно отдаете приоритет одной из этих метрик при разработке алгоритма.

Контрольные вопросы по теме:

1. Основные понятия задач классификации в машинном обучении.
2. Опишите алгоритм метода k-ближайших соседей.
3. Приведите пример функции для расчета расстояния между классифицируемыми объектами.
4. Зависит ли сложность метода k-ближайших соседей от количества классов? От числа точек? От значения константы k?
5. Приведите пример практического применения классификации.
6. Приведите пример решения задачи классификации методом k-ближайших соседей.
7. Перечистите плюсы и минусы метода k-ближайших соседей.