Вопрос 1

Методами машинного обучения (не статистическими тестами) показать, что разбиение на трейн и тест репрезентативно.

Решение

Использование моделей для предсказания распределения

Метод:

Можно обучить классификационную модель, задача которой определить, из какой выборки (тренировочной или тестовой) принадлежит данный пример. Если модель не может существенно отличить примеры из тренировочной или тестовой выборок, это свидетельствует о том, что распределения этих выборок схожи, а разбиение репрезентативно.

Шаги:

1. Создайте бинарный признак, обозначающий принадлежность примера к тренировочной или тестовой выборке.
2. Объедините тренировочную и тестовую выборки, сохраняя этот бинарный признак.
3. Обучите классификационную модель на всей объединённой выборке, пытаясь предсказать данный бинарный признак.
4. Оцените производительность модели (например, точность, ROC-AUC).
5. Если модель не способна с высокой точностью отличить тренировочную от тестовой выборки (например, точность близка к случайной), разбиение считается репрезентативным.

Визуализация распределения признаков с использованием методов снижения размерности

Метод:

Методы снижения размерности, такие как t-SNE или PCA, позволяют визуально оценить схожесть распределений train и test выборок. Если данные из обеих выборок плотно располагаются в одних и тех же областях после снижения размерности, это свидетельствует о репрезентативности разбиения.

Шаги:

1. Объедините тренировочную и тестовую выборки.
2. Примените метод снижения размерности (например, t-SNE, PCA) к объединённым данным.
3. Визуализируйте данные на 2D или 3D графике, раскрасив точки в зависимости от принадлежности к тренировочной или тестовой выборке.

4. Оцените перекрытие и распределение точек из разных выборок.

Обучение и сравнение нескольких моделей

Метод:

Если различные модели демонстрируют схожие показатели производительности на обеих выборках, это можно рассматривать как подтверждение репрезентативности.

Шаги:

1. Выберите несколько различных моделей (можно использовать простые модели).
2. Обучите каждую модель на тренировочной выборке.
3. Оцените каждую модель на тренировочной и тестовой выборках.
4. Сравните метрики производительности для каждой модели на обеих выборках.

Многократное разбиение на трейн и тест

Если известен способ, которым данные разделены на тренировочную и тестовую выборки, можно повторить процедуру несколько раз с разными значениями зерна (seed). Далее можно обучить модель на различных наборах тренировочных и тестовых данных и сравнить результаты.

Шаги:

1. Повторите разбиение данных на тренировочную и тестовую выборки несколько раз, используя разные значения seed.
2. Для каждого разбиения обучите модель на тренировочной выборке.
3. Оцените модель на соответствующей тестовой выборке.
4. Проанализируйте стабильность и согласованность результатов модели по различным разбиениям.

Вопрос 2

Есть кластеризованный датасет на 4 кластера (1, 2, 3, 4). Бизнес аналитики посчитали, что самым прибыльным является кластер 2. Каждый клиент представлен в виде 10-тимертного вектора, где первые 6 значений транзакции, а оставшиеся: возраст, пол, социальный статус (женат (замужем)/неженат (не замужем)), количество детей. Нужно поставить задачу оптимизации для каждого клиента не из кластера 2 так, чтобы увидеть как должен начать вести себя клиент, чтобы перейти в кластер 2.

Решение

В этой задаче нужно найти такое минимальное изменение ΔX, что $X + \Delta X$ — новый вектор клиента после внесения изменений, принадлежит второму кластеру. То есть задача формулируется следующим образом:

$$||\Delta X||_2 \to min$$
$$X + \Delta X \in C_2$$
$$\Delta X_i = 0, \quad \text{если } i \in \{7, 8, 9, 10\}$$

Важные замечания:

1. *Ограничение изменяемых переменных*. Не все переменные можно изменять произвольно. Например, неразумно требовать от клиента изменения возраста и пола. Соответственно, переменные, которые нельзя изменять, должны иметь $\Delta X_i = 0$.
2. *Масштаб признаков*. Ключевым аспектом задачи является масштабирование признаков, чтобы "стоимость" изменения на 1 была сопоставима для всех признаков. Простого нормирования может быть недостаточно; возможно, потребуется более сложное масштабирование, включающее экспернтые оценки.
3. *Выбор нормы*. В качестве функции расстояния можно использовать разные нормы, например, L_1, L_2 или другие, в зависимости от специфики задачи.

Сложность решения этой задачи во многом зависит от сложности формы области C_2. Эта форма определяется используемым алгоритмом кластеризации. Если мы использовали k-means, то форма задаётся несколькими линейными неравенствами (разделяющими гиперплоскостями). В этом случае задача решается относительно просто – у нас всего несколько точек-кандидатов. Это точка нормальной проекции точки клиента на гиперплоскость, разделяющую текущий кластер и кластер 2 и угловые точки области кластера 2.

Вопрос 3

Что лучше 2 модели случайного леса по 500 деревьев или одна на 1000, при условии, что ВСЕ параметры кроме количества деревьев одинаковы?

Решение

В алгоритме случайного леса создаётся множество деревьев решений, каждое из них по своей bootstrap выборке и случайному разбору признаков, затем объединяем их предсказание.

- **Вычислительная сложность**:
  Оба варианта требуют обучения одинакового количества деревьев (1000), следовательно, объём вычислений схож. Также важно отметить, что случайный лес хорошо поддаётся параллельной обработке. Таким образом, с точки зрения производительности, значительной разницы между двумя вариантами нет.
- **Качество модели**
  В обоих вариантах мы обучим 1000 независимых одинаковых деревьев, затем агрегируем их предсказания. Если мы не теряем информацию при двукратном усреднении, результат будет одинаков. Например, мы решаем задачу классификации и итоговое предсказание получаем методом простого голосования большинства. Если мы в качестве результатов работы лесов возвращаем просто результат голосования, мы теряем информацию о числе голосов и результат 2х лесов по 500

будет хуже, чем у одного на 1000. А если мы возвращаем количество голосов, то мы можем агрегировать результаты 2х лесов без потери информации.

Вопрос 4

В наличии датасет с данными по дефолту клиентов. Как, имея в инструментарии только алгоритм kmeans получить вероятность дефолта нового клиента.

Решение

Алгоритм K-Means является методом кластеризации. С его помощью можно разделить клиентов на несколько кластеров. Затем рассчитать характеристики этих кластеров. Нового клиентам можно отнести к какому-то из существующих кластеров и на основе этого определить вероятность его дефолта.

Способ 1

Этап 1. Подготовка данных

- обработка пропущенных значений и выбросов
- кодирование категориальных признаков
- стандартизация и нормализация числовых признаков
- разделение данных на трейн и тест

Этап 2. Отбор признаков
Выбираем релевантные признаки, которые могут повлиять на вероятность дефолта. Например, с помощью анализа корреляции

Этап 3. Определение числа кластеров
Например, методом локтя. Тут нужно учитывать бизнес-логику и интерпретируемость кластеров.

Этап 4. Обучение модели k-means

Этап 5. Валидация и оценка результатов

Этап 6. Анализ кластеров
Пытаемся проинтерпретировать результат и проверить его на адекватность/ связь с реальностью.

На этом этапе также вычисляется вероятность дефолта в каждом кластере как доля дефолтов к общему числу элементов кластера.

Этап 6. Оценка вероятности дефолта нового клиента

- предобрабатываем нового клиента так же, как и данные
- назначаем клиенту кластер с помощью
- возвращаем вероятность дефолта для этого кластера

Способ 2

Действуем так же, только на последнем этапе присваиваем клиенту не кластер, а набор вероятностей принадлежать к кластерам на основе расстояния до центров нескольких ближайших кластеров. В качестве вероятности дефолта возвращаем сумму произведений вероятности отнесения к кластеру на вероятность дефолта в этом кластере.

Для дополнительной интерпретируемости можно с помощью PCA визуализировать в 2D кластеры и положение нового клиента по отношению к ним.

Способ 3

Отдельно решаем задачу кластеризации для клиентов с дефолтом и клиентов без дефолта.

Далее рассчитываем вероятность и визуализируем так же, как в способе 2.

Вопрос 5

Есть выборка клиентов с заявкой на кредитный продукт. Датасет состоит из персональных данных: возраст, пол и т.д. Необходимо предсказывать доход клиента, который представляет собой непрерывные данные, но сделать это нужно используя только модель классификации.

Решение

Этап 0. Предварительный анализ данных.
Например, размер выборки, доля пропусков, в каком диапазоне и насколько неравномерно распределены доходы и т.п.

Этап 1. Постановка задачи
Это самый главный этап и он должен выполняться в плотной работе с заказчиком. Определить ДНК (Дано-Найти-Критерий) задачи.

Понятно, что т.к. мы решаем задачу классификации, клиенты будут как-то разбиваться на классы по уровню дохода. И эти результаты в дальнейшем будут как-то использоваться в процессе принятия решения о выдаче кредита. То, на какие именно классы нужно разбивать клиентов тесно связано с тем, как именно будут в дальнейшем использоваться результаты модели.

Например, если у заказчика уже есть обученная система принятия решения, она сделана на основе решающего дерева и этой системе есть бинарное правило "доход больше определённой суммы", в этом случае в нашей задаче естественным способом получается бинарная классификация вокруг этой суммы. Совершенно другая ситуация, когда подобной системы нет и её собираются обучать. В этом случае может быть удобна равночастотная дискретизация. Каждый бин будет содержать примерно одинаковое количество наблюдений, что может быть важно для моделей чувствительных к дисбалансу классов, но интервалы будут неравномерны по ширине и с "некрасивыми"

границами, что может затруднить интерпретацию. Третья ситуация - если результаты работы модели будут использовать люди. В этом случае классы должны быть максимально удобны для интерпретации. Например, разбиение на равные по ширине интервалы.

Также для целей выдачи конкретного продукта может не иметь смысла различать классы людей с доходом выше(ниже) определённого уровня и их нужно слеплять в один класс.

При выработке функции потерь классы могут быть не симметричными. Например, для бизнеса выдать невозвратный кредит может быть связано с существенно большими потерями, чем не выдать возвратный. Поэтому ошибочное отнесение к более высокому классу должно штрафоваться сильнее, чем к низкому.

Также от постановки задачи зависит как нужно обращаться с неполными данными. Стоит ли их игнорировать или нужно заполнять средними или заполнить чем-то другим.

Эти и многие другие вопросы постановки задачи, хоть напрямую и не относятся к машинному обучению имеют огромную важность. И от качества постановки зависит результат не в меньшей степени, чем от самого решения.

Этап 2. Выбор модели классификации
Для этой задачи хорошо пойдут случайный лес, градиентный бустинг, решающее дерево, логистическая регрессия. В случае большого количества доступных данных можно также рассмотреть нейронные сети.

Решающее дерево и логистическая регрессия проще в реализации и хорошо интерпретируемы.

Случайные лес и градиентный бустинг отличаются большей точностью, но более дорогие, в плане трудовых и вычислительных ресурсов, а также менее интерпретируемы.

Возможно, имеет смысл начать с более простых моделей (в качестве "прототипа"), потом, при необходимости переходить к более сложным.

Этап 3. Подготовка данных

- разбиение данных на классы (бины) по доходу
- обработка пропущенных значений (заполнение/удаление) и выбросов
- кодирование категориальных признаков
- стандартизация и нормализация числовых признаков
- разделение данных на обучающую и тестовую выборки. Или на подвыборки для кросс-валидации

Этап 4. Обучение модели.
Если ресурсы позволяют, можно несколько раз обучить модель на разных параметрах и сравнить результаты.

Этап 5. Валидация на тесте и оценка результатов
Можно использовать следующие метрики оценки качества задачи классификации

1. Точность (Accuracy):
    - Доля правильно предсказанных классов.
2. Матрица ошибок (Confusion Matrix):
    - Позволяет видеть, какие классы путаются.
3. Precision, Recall, F1-Score:
    - Особенно полезны при несбалансированных классах.
4. ROC AUC (для многоклассовых задач):
    - Можно использовать "one-vs-rest" подход.
5. Log Loss:
    - Учитывает уверенность модели в своих предсказаниях.

Этап 6. Обратное преобразование

При необходимости можно перейти от классов обратно к непрерывным данным. Для этого каждому классу присваивается какое-то значение. Например, медиана среди данных, попавших в этот класс в изначальном датасете. Можно использовать и другие средние, это зависит от постановки задачи.

После этого этапа можно ещё раз оценить качество решения, но уже используя метрики качества регрессии, например MSE (среднеквадратическое отклонение).

Заключительный этап –
Предоставление результатов заказчику в удобной для него форме.