Вопросы на ”до свидания”

1. Полнота, точность

   Полнота, она же Recall - сколько фрода мы поймали vs сколько пропустили.

   Точность, Precision - сколько фрода vs не-фрода мы поймали.

$$Recall = \frac{tp}{tp + fn}$$

$$Precision = \frac{tp}{tp + fp}$$

2. ROC кривая. Коэфициент Джини. AUC

   ROC кривая - зависимость true positive от false positive для разных порогов решающего правила.

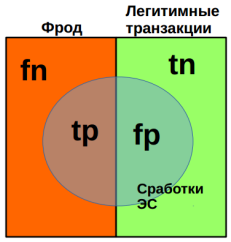
   Коэффициент Джини - максимальное отклонение ROC кривой от диагонали $tp = fp$.

   Если система адекватная (не контрпродуктивная), то кривая всегда будет выше диагонали.

   AUC - ”area under curve” - площадь под кривой ROC. Чем выше, тем лучше.

3. tp, fp, tn, fn

...	Фрод	Не фрод
Думаем, что фрод	tp	fp
Думаем, что не фрод	fn	tn



4. классификация

   Есть несколько категорий (напр. фрод\не фрод). Необходимо объекты расфасовать по категориям. В ”обучающей выборке” заранее известно, в какую категорию какой объект входит.

5. кластеризация

   Категории заранее неизвестны. Необходимо разбить объекты на несколько ”каких-то” категорий.

6. регрессия

   Зная значение функции на каком-то наборе точек, наиболее хорошо приблизить функцию.

7. идентификация

   Определить, к какой из большого числа категорий принадлежит объект.

8. обучение с учителем и обучение без учителя

   Supervised\Unsupervised learning. При обучении с учителем системе дается какая-то обучающая выборка, для которой задача уже решена (например, при задаче классификации). Без учителя обучающей выборки нет, система сама должна искать закономерности (кластеризация).

9. признак

   Строго заданная характеристика объекта (например, сумма транзакции в рублях).

10. класс

    Одна из категорий, принадлежность к которой необходимо определить при решении задачи классификации.

11. событие, действие, решение

    ???

12. обратная связь

    В общем случае: влияние выхода системы на ее дальнейший вход.

13. DENY, REVIEW, ALLOW решения ФМ системы RSA

    DENY - заблокировать транзакцию и профиль клиента.

    REVIEW - потребовать подтверждение в КЦ.

    ALLOW - разрешить без вопросов.

14. примеры применения машинного обучения для задач информационной безопасности

    Имея транзакции банка, определить фрод.

    Зная поведение ботнетов, предсказывать будущие атаки или определять, является ли компьютер частью ботнета.

15. экспертная система

    ???

16. выброс

    Нестандартные данные, которые могут помешать классификации.

17. аномалия

    ???

18. выборка

    Конечный набор объектов, взятый из множества всех возможных объектов.

19. разделяющая гиперповерхность

    Гиперплоскость в пространстве всех возможных объектов, которая разделяет объекты на классы.

20. функция штрафа

    Некая функция, символизирующая, насколько "плохо" был классифицирован объект. Обычно функция расстояния или просто +1 за каждую точку не в своем классе. Задача классификации сводится к задаче минимизации функции штрафа.

21. логистическая функция

    $$f(\pm d(x_1, x_2)) = f(d) = \frac{1}{1 - e^{-\alpha \times d}}$$

    Используется в качестве функции штрафа.

22. ансамбль

    Мнения нескольких систем так или иначе аггрегируются в одно (например, функцией голосования).

23. дерево решений

    Дерево, в каждом узле которого находится условие, определяющее, к какому из детей надо перейти. В листьях содержатся классы. Каждый объект спускается по дереву в соответствии с условиями, пока не попадет в класс.

24. бутстрепинг

    Генерация выборки размера N из подвыборки размера $n \ll N$ путем выбора с повторением.

25. бутстреп аггрегация

    Она же bagging. Тот же ансамбль на классификаторах.

    Обобщение одного и того же классификатора на различных подвыборках или случайных параметрах, которые различны при каждой новой подгонки.

26. подгонка (fitting)

    Настройка параметров классификатора с помощью обучающих выборок.

27. проверка (scoring)

    Оценка качества классификатора на тестовой выборке.

28. переобучение (overfitting)

    Явление, когда модель слишком точно подогнана под обучающие данные, видя закономерности в случайном шуме. Такая модель может работать хуже на больших данных.

29. пакеты pandas, numpy, scikit-learn, matplotlib

    Библиотеки для Python.

    Pandas - структуры данных.

    numpy - матрицы, математика.

    scikit-learn - машинное обучение

    matplotlib - графики

Вопросы 1

1. 111

Вопросы 2 и 3

1. Задачи классификации, кластеризации и регрессиии.

   Классификация:

   Есть набор объектов, каждый из которых принадлежит к какому-то заранее определенному классу. Системе на вход поступает обучающая выборка: набор объектов, для которых класс заведомо известен. Система должна новые поступающие объекты распределять по классам.

   Кластеризация:

   Есть набор объектов. Необходимо их разбить на какое-то количество кластеров, так что в одном кластере объекты "похожи"друг на друга, а между классами различаются.

   Регрессия:

   Известно значение функции в точках. Найти функцию.

2. Задачи регрессии, интерполяции, аппроксимации. Прогнозирование.

   Регрессия:

   Известно значение функции в точках. Найти функцию.

   Интерполяция:

   Известно значение функции в точках. Найти значение в промежуточных точках.

   Аппроксимация:

Известно значение функции в точках. Найти приближенное значение функции в других точках.

~~Пророчество~~ Прогнозирование - попытка узнать значение какого-то показателя в будущем, зная, как он вел себя в прошлом.

3. Задачи ML и DM в информационной безопасности: банковский фрод; pay per click. Суть проблемы и общий подход к решению.

   Задача: среди огромного количества происходящих транзакций выявлять мошеннические и отвергать\валидировать подозрительные.

   Задача: среди кликов на рекламные баннеры выявлять те, которые реально принесли рекламную пользу (а не являются ботами, например). Начислять деньги только за осмысленные клики.

   Решается задача классификации с двумя классами: фрод\не фрод или зачислять\не зачислять.

4. Задачи ML и DM в информационной безопасности: call фрод, обнаружение социальной инженерии. Суть проблемы и общий подход к решению.

   ???

5. Задачи ML и DM в информационной безопасности: поведенческий анализ botnet-ов, обнаружение фазинга. Суть проблемы и общий подход к решению.

   ???

6. Полнота и точность. F-мера. Другие метрики оценки качества классификаторов.

   Полнота, она же Recall - сколько фрода мы поймали.

   Точность, Precision - сколько не-фрода мы пропустили.

$$Recall = \frac{tp}{tp + fn}$$

$$Precision = \frac{tp}{tp + fp}$$

   F-мера - единый критерий, зависящий от полноты и точности.

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

   Другие метрики:

   ROC-кривая, коэффициент Джини. Accuracy ($\frac{tp+tf}{tp+tn+fp+fn}$). Cost-benefit analysis. Value at risk.

7. Расстояние Евклида, манхеттенское расстояние. Расстояние Минковского. Расстояние Чебышева. Расстояние Махаланобиса.

   Расстояние Евклида: $\sqrt{\Delta x^2 + \Delta y^2 + ...}$

   Манхэттенское расстояние: $\Delta x + \Delta y + ...$

   Расстояние Минковского: $P(x, y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$

   Расстояние Махаланобиса: $d(x_1, x_2) = \sqrt{(x_1 - x_2)^T \Delta^{-1} (x_1 - x_2)}$, где Δ - матрица ковариации. Вырождается в евклидово для единичной матрицы ковариации.

8. ROC кривая. Коэфициент Джини. AUC.

   ROC кривая - зависимость true positive от false positive для разных порогов решающего правила.

   Коэффициент Джини - максимальное отклонение ROC кривой от диагонали $tp = fp$.

   Если система адекватная (не контрпродуктивная), то кривая всегда будет выше диагонали.

   AUC - "area under curve" - площадь под кривой ROC. Чем выше, тем лучше.

9. SSI. VaR, CBA

   System Stability Index: задается отношение двух величин на момент времени t : x и $y = f(x, t)$. Выбираются значения x_i : $\{x_1, x_2, ...x_n\}$.

$$SSI(t_1, t_2) = \sum_{i=1..n} (f(x_i; t_1) - f(x_i; t_2)) * \log_q \left(\frac{f(x_i; t_1)}{f(x_i; t_2)} \right)$$

   Cost-Benefit Analysis: оцениваем свойства системы в условных единицах. Определяются затраты того или иного поведения системы. Решается СЛАУ, максимизирующая получаемые единицы.

   Value at risk: определяются риски и их ущерб, составляется СЛАУ, минимизирующая этот ущерб.

10. Признаки, характеристики, контрибьютеры. Сравнимые и несравнимые признаки.

Сложные и простые. Непрерывные, дискретные, бинарные.

Харакеристика — некое обобщенное свойство предмета.

Признак — конкретно математически заданное свойство предмета.

Контрибьютор — совокупность признаков (возможно один), вносящих определенный вклад в априорную вероятность определения класса.

Сравнимые признаки — те, на пространстве которых можно осмысленно задать полный порядок (например, возраст).

Несравнимые — для которых никакой порядок не имеет смысла (например, город).

Сложные — некая композиция простых признаков (например, дата состоит из года, месяца...)

Непрерывные\дискретные — день, когда я не смогу назвать разницу между непрерывным и дискретным без шпаргалки будет днем, когда я уйду на пенсию.

Бинарные — 2 возможных значения.

11. Пакеты Python. NumPy, SciPy, Pandas, Matplotlib. Jupyther.

    Pandas — структуры данных.

    numpy — матрицы, математика.

    scikit-learn — машинное обучение.

    matplotlib — графики.

    Jupyter — web-приложение для создания интерактивных книг с кодом.

12. Функция штрафа.

    Некая функция, символизирующая, насколько "плохо" был классифицирован объект. Обычно функция расстояния или просто +1 за каждую точку не в своем классе. Задача классификации сводится к задаче минимизации функции штрафа.

13. Байесовский подход. Преимущества и недостатки. Разбиновка. Преимущества и недостатки

    Байесовский подход: имеем априорную вероятность попадания объекта в тот или иной класс. Вычисляем или задаем вероятность, что объект будет иметь определенный набор признаков (отношение правдоподобия). Кладем в теорему байеса, перемешиваем ==> proft.

    Наиболее математически "корректный", но отношения правдоподобия надо откуда-то взять.

    Разбиновка: разбиваем пространство на прямоугольные области. Каждую из областей помечаем каким-то классом. Новые объекты кладем в тот класс, в чей бин он попал.

    Крайне легкий в вычислении и реализации, но ограничен в том, какие именно зависимости он может поймать.

14. Дерево решений. Лес. Случайный лес. Построение случайного леса. Критерии остановки. «Эмпирическая теорема» о случайном лесе.

    Дерево решений — дерево, в каждом узле которого находится условие, определяющее, к какому из детей надо перейти. В листьях содержатся классы. Каждый объект спускается по дереву в соответствии с условиями, пока не попадет в класс.

    Лес — ансамбль деревьев.

    Случайный лес — лес, построенный случайным образом.

    Построение — случайным образом строится разбиновка. Останавливаемся, если в бине мало элементов, если дерево слишком глубокое или если коэффициент Джини мал. Проверяется ее качество. Если достаточно — кладем в лес. Повторяем до готовности.

    "Эмпирическая теорема" — случайный лес хоть как-то работает, если вы не можете придумать ничего лучше.

15. Бутстрепинг, бутстреп, бэггинг и бустинг

    Бутстрепинг — генерация выборки размера N из подвыборки размера $n \ll N$ путем выбора с повторением.

    Бутстреп-агрегация или бэггинг — обобщение одгого и того же классификатора на различных подвыборках или случайных параметрах, которые различны при каждой новой подгонке.

    Бустинг — последовательный ансамбль, каждый последующий член которого корректирует предыдущий.

16. Восстановление плотности. Парзеновская плотность. Ядро. Виды ядер.

    Восстановление плотности — по выборке из неизвестного распределения восстановить плотность этого распределения.

”Честный способ” — для каждой точки строим поле. Плотность равна потенциалу в этой точке.

Упрощение — заменить группы точек на геометрические центры или центры масс.

Функция ядра — некая гладкая четная функция, определяющая потенциал точки.

Метод парензовской плотности — в основе подхода лежит идея о том, что плотность выше в тех точках, рядом с которыми находится большое количество объектов выборки. Если мощность множества элементарных исходов много меньше размера выборки, то в качестве восстановленной по выборке плотности мы вполне можем взять и гистограмму значений выборки.

Парензовская плотность:

$$p(x; y_i, h) = c(h, y_i) \times \sum_{\forall x: y(x_i)=y_i} K\left(\frac{r(x, x_i)}{h}\right)$$

$K(r(x, x_i)/h) = K(r_0)$ - ядро.

17. Pandas. Задачи библиотеки. DataFrame, Series, Panel

    Pandas — библиотка Python для работы со структурами данных.

    Series — одномерная структура данных. DataFrame — двумерная. Panel — трехмерная.

18. Pandas. Задачи библиотеки. loc, iloc, ix

    loc — взять значение по лэйблу. iloc — по индексу. ix — сначала по лейблу, fallback по индексу. Можно подать массив или слайс лэйблов.

19. Pandas. Задачи библиотеки. Типы данных

    Численные (float64, integer64), символьные (object).

20. Pandas. Задачи библиотеки. apply функция

    $apply(func, axis = 0, broadcast = False, raw = False, reduce = None, args = (), **kwds)$

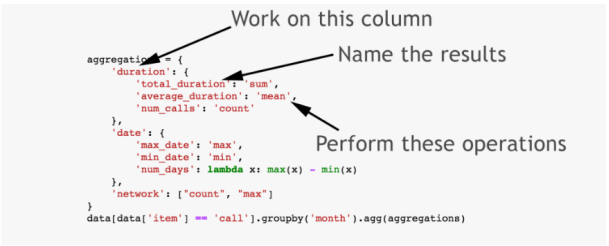
    apply — возвращает результат применения функции к структуре данных (к каждом объекту в Series, каждой Series в DataFrame, каждому DataFrame в Panel.)

21. Pandas. Задачи библиотеки. Группировка, аггрегация, фильтрация.

    $groupby(by = None, axis = 0, level = None, as_index = True, sort = True, group_keys = True, squeeze = False, **kwargs)$

    Группирует элементы структуры данных по какому-то предикату.

    Aggregate.

    $filter(items = None, like = None, regex = None, axis = None)$

    Оставляет только те элементы, которые совпадают с like или regexp.

    Также можно фильтровать элементы оператором

    : df[df.a>5]

22. Scikit-learn. Задачи библиотеки. sklearn.model_selection

    Scikit-learn — простые и эффективные инструменты машинного обучения для Python.

    model_selection — инструменты для кросс-валидации класификатора.

    sklearn.model_selection.train_test_split — разделять выборку на обучающую и тестовую.

    sklearn.model_selection.cross_val_score — самый простой способ провести кросс-валидацию.

    sklearn.model_selection.KFold — делит выборку на группы по k, и кросс-валидирует классификатор, используя одну группу как тестовую, а остальные как тестовые.

    ...

23. Scikit-learn. Задачи библиотеки. Сокращение размерности. PCA. IncrementalPCA

    Сокращение размерности — уменьшение количества признаков без (значительной) потери качества модели.

    PCA — метод сокращения размерности путем превращения нескольких линейно кореллирующих переменных в меньшее количество линейно независимых.

sklearn.decomposition.PCA обрабатывает все пространство разом; sklearn.decomposition.IncrementalPCA менее корректен, но потребляет меньше памяти.

24. Scikit-learn. Задачи библиотеки. Сокращение размерности. ProjectedGradientNMF. FactorAnalysis

    ???

25. Scikit-learn. Задачи библиотеки. sklearn.preprocessing

    sklearn.preprocessing содержит инструменты для предварительной обработки данных, как то:

    - приведение данных к 0 среднему и 1 дисперсии.
    - нормализация
    - бинаризация численных данных
    - кодирование категориальных данных
    - дополнение отсутствующих данных
    - добавление полиномиальных композиций фич

26. Scikit-learn. Задачи библиотеки. sklearn.ensemble

    sklearn.ensemble — инструменты для составления ансамблей нескольких классификаторов.

    Например, BaggingClassifier — использует классификатор на нескольких подвыборках и аггрегирует результаты. RandomForestClassifier — реализует случайный лес. AdaBoostClassifier — алгоритм AdaBoost.