Оценочные материалы для промежуточной аттестации по дисциплине

Название дисциплины «Машинное обучение»

Код, направление подготовки	09.04.02 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ
Направленность (профиль)	УПРАВЛЕНИЕ ДАННЫМИ
Форма обучения	Очная
Кафедра-разработчик	Информатики и вычислительной техники
Выпускающая кафедра	Информатики и вычислительной техники

Типовые задания для контрольной работы (3 семестр):

Исходные данные для машинного обучения.

1. Вопросы на которые необходимо ответить:
   • Как собрать и подготовить данные для построения модели?
   • Как интерпретировать модель и ее результаты?
   • Как корректно оценить качество модели?
   Когда появилась достаточная выборка, можно смело строить гипотезы, используя алгоритмы машинного обучения.

2. Этапы работы:
   - Сбор данных
   - Препроцессинг
   - Построение модели
   - Анализ качества и интерпретация модели

   Сбор данных

   К задаче подходим серьезно, так как на ее основании строится дальнейший процесс. Во-первых, нужно не упустить важные признаки, описывающие объект, во-вторых, создать жесткие критерии для принятия решения о признаке.

   Булевы (бикатегориальные), ответом на которые является: Да или Нет (1 или 0). Например, ответ на вопрос: есть ли у клиента аккаунт?

   Категориальные, ответом на которые является конкретный класс. Обычно классов больше двух (мультикатегориальные), иначе вопрос можно свести к булевому. Например, цвет: красный, зеленый или синий.

   Количественные, ответами на которые являются числа, характеризующее конкретную меру. Например, количество обращений в месяц: пятнадцать.

   Обычно, когда рассматривают классическую задачу, решаемую алгоритмами машинного обучения, мы имеем дело только с численными данными. Например, распознавание черно-белых рукописных цифр с картинки 20 на 20 пикселей. В этом примере 400 чисел (описывающих яркость черно-белого пикселя) представляют один пример из выборки. В общем случае данные необязательно должны быть числовыми. Дело в том, что при построении модели нужно понимать, с какими типами вопросов алгоритм может иметь дело.

Например: дерево принятия решения обучается на всех типах вопросов, а нейросеть принимает только числовые входные данные и обучается лишь на количественных признаках. Означает ли это, что мы должны отказаться от некоторых вопросов в угоду более совершенной модели? Вовсе нет, просто нужно правильно подготовить данные.
Данные должны иметь следующую классическую структуру: вектор признаков для каждого i-го клиента X(i)= {x(i)1, x(i)2, ..., x(i)n} и класс Y(i) — категория, показывающая купил он или нет. Например: клиент(3) = {зеленый, горький, 4.14, да} — купил.
Основываясь на вышесказанном, попробуем представить формат данных с типами вопросов, для дальнейшей подготовки:

класс: (категория)	цвет: (категория)	вкус: (категория)	вес: (число)	твердый: (bool)
-	красный	кислый	4.23	да
-	зеленый	горький	3.15	нет
+	зеленый	горький	4.14	да
+	синий	сладкий	4.38	нет
-	зеленый	соленый	3.62	нет

Препроцессинг

После того как данные собраны, их необходимо подготовить. Этот этап называется препроцессинг. Основная задача препроцессинга — отображение данных в формат пригодный для обучения модели. Можно выделить три основных манипуляции над данными на этапе препроцессинга:

Создание векторного пространства признаков. По сути, это процесс приведения всех данных в числовую форму. Это избавляет нас от категорийных, булевых и прочих не числовых типов.

Нормализация данных. Процесс, при котором мы добиваемся, например того, чтобы среднее значение каждого признака по всем данным было нулевым, а дисперсия — единичной.

Изменение размерности векторного пространства. Если векторное пространство признаков слишком велико (миллионы признаков) или мало (менее десятка), то можно применить методы повышения или понижения размерности пространства:

Для повышения размерности можно использовать часть обучающей выборки как опорные точки, добавив в вектор признаков расстояние до этих точек. Этот метод часто приводит к тому, что в пространствах более высокой размерности множества становятся линейно разделимыми, и это упрощает задачу классификации.

Для понижения размерности чаще всего используют PCA. Основная задача метода главных компонент — поиск новых линейных комбинаций признаков, вдоль которых максимизируется дисперсия значений проекций элементов обучающей выборки.

Проведя все манипуляции над данными, мы получим обучающую выборку, подходящую любой модели. В нашем случае, после применения унитарной кодировки и нормализации данные выглядят так:

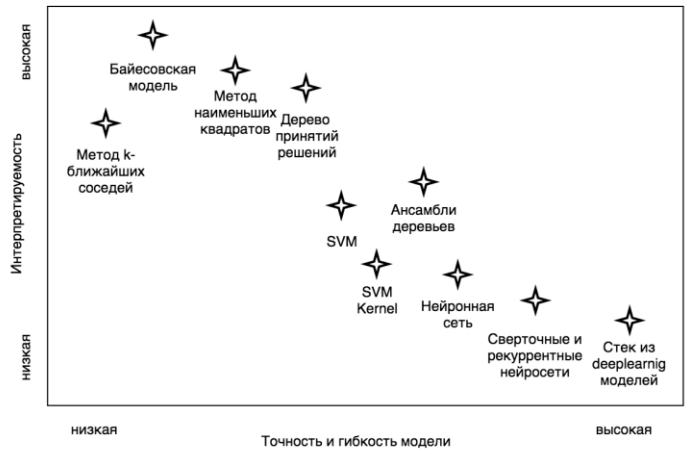
Можно сказать, что препроцессинг — это процесс отображения понятных нам данных в менее удобную для человека, но зато в излюбленную машинами форму.
Формула скоринга чаще всего представляет из себя следующую линейную модель:

$$score = \sum_{k=1}^{|w|} w_k x_k$$

где, k — это номер вопроса в анкете, wk — коэффициент вклада ответа на этот k-ый вопрос в суммарный скоринг, |w| — количество вопросов (или коэффициентов), xk — ответ на этот вопрос.

Выбор модели

Теперь самое важное: выбор модели. На сегодняшний день существует множество алгоритмов машинного обучения, на основе которых можно построить скоринг модель: Decision Tree (дерево принятия решений), KNN (метод k-ближайших соседей), SVM (метод опорных векторов), NN (нейросеть). И выбор модели стоит основывать на том, чего мы от нее хотим. Во-первых, насколько решения, повлиявшие на результаты модели, должны быть понятными. Другими словами, насколько нам важно иметь возможность интерпретировать структуру модели.



Кроме того, не все модели легко построить, для некоторых требуются весьма специфические навыки и очень-очень мощное железо. Но самое важное — это внедрение построенной модели. Бывает так, что бизнес-процесс уже налажен, и внедрение какой-то сложной модели попросту невозможно. Или требуется именно линейная модель, в которой клиенты, отвечая на вопросы, получают положительные или отрицательные баллы в зависимости от варианта ответа. Иногда, напротив, есть возможность внедрения, и даже требуется сложная модель, учитывающая очень неочевидные сочетания входных параметров, находящая взаимосвязи между ними.

Задача.

Построить свою модель на уже готовой библиотеке, которые сейчас есть для любого языка программирования, а также постепенно углублять свои теоретические знания в этом направлении.

Обучение модели

Когда у нас есть и обучающая выборка, и теоретические знания, можем начинать обучение нашей модели. Однако проблема заключается в том, что часто элементы множеств представлены в неравных пропорциях. Купивших может быть 5%, а некупивших — 95%. Как в таком случае производить обучение? Ведь можно добиться 95% достоверности, утверждая, что никто не купит.

Например, если у нас всего 20,000 примеров и из них 1,000 купивших, можно случайным образом выбрать из каждой группы по 500 примеров и использовать их для обучения. И повторять эту операцию раз за разом. Это немного усложняет реализацию процесса обучения, но зато помогает получить грамотную модель.

Выбрав модель и алгоритм обучения, желательно разделить вашу выборку на части: провести обучение на обучающей выборке, составляющей 70% от всей, и пожертвовать 30% на тестовую выборку, которая потребуется для анализа качества полученной модели.

Оценка качества модели

Подготовив модель, необходимо адекватно оценить ее качество. Для этого введем следующие понятия:

TP (True Positive) — истиноположительный. Классификатор решил, что клиент купит, и он купил.

FP (False Positive) — ложноположительный. Классификатор решил, что клиент купит, но он не купил. Это так называемая ошибка первого рода. Она не так страшна, как ошибка второго рода, особенно в тех случаях, когда классификатор — тест на какое-нибудь заболевание.

FN (False Negative) — ложноотрицательный. Классификатор решил, что клиент не купит, а он мог купить (или уже купил). Это так называемая ошибка второго рода. Обычно при создании модели желательно минимизировать ошибку второго, даже увеличив тем самым ошибку первого рода.

TN (True Negative) — истиноотрицательный. Классификатор решил, что клиент не купит, и он не купил.

Кроме прямой оценки достоверности в процентах существуют такие метрики, как точность (англ. precision) и полнота (англ. recall), основанные на вышеприведенных результатах бинарной классификации.

Метрика достоверности

Самая простая метрика — это метрика достоверности (англ. Accuracy). Но эта метрика не должна быть единственной метрикой модели. Особенно в тех случаях, когда существует перекос в выборке, то есть представители разных классов встречаются с разной вероятностью.

$$Accuracy = \frac{tp+tn}{tp+fp+fn+tn}$$

Точность и полнота

Точность (англ. precision) показывает отношение верно угаданных объектов класса ко всем объектам, которые мы определили как объекты класса. Например, мы решили, что купит 115, а из них реально купило 37, значит точность составляет 0.33. Полнота (англ. recall) показывает отношение верно угаданных объектов класса ко всем представителям этого класса. Например, среди нами угаданных реально купило 37, а всего купивших было 43. Значит наша полнота составляет 0.88.

	Gold Class 1	Gold Class 2
Observed Class 1	TP	FP
Observed Class 2	FN	TN

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

F-мера

Также существует F-мера (англ. F1 score) — среднее гармонической точности и полноты. Помогает сравнить модели, используя одну числовую меру.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Интерпретация модели

Когда у нас есть готовая модель, мы можем использовать ее, ожидая той точности, которой нам дал анализа ее качества.

Результаты при решение задач машинного обучения:

Рассмотреть все основные этапы data-mining.

Узнать много полезных приемов как при подготовке данных, так и при обучении.

Достаточно глубоко познакомиться с теорией классических искусственных нейронных сетей.

Рассмотреть разные статистические подходы к анализу качества модели.

Описать все этапы от создания до разбора и внедрения нейронной сети на примере построения линейной скоринг модели.

Показать, как современные алгоритмы машинного обучения могут помочь в решении реальных бизнес-задач.

Темы (контрольная работа).

1. Задачи и терминология машинного обучения: supervised и unsupervised задачи; регрессия, классификация, кластеризация.
2. Объект, признак, типы признаков, методы работы с ними. Метрики качества. Инструменты интеллектуального анализа данных.
3. Изучение основ работы с векторными данными и визуализацией. Градиент.
4. Методы оптимизации гладких функций.
5. Реализация градиентного спуска для линейной регрессии.
6. Выделение признаков из текста: one-hot encoding, стемминг, лемматизация, tf-idf преобразование.
7. Логистическая регрессия на примере задачи эмоциональной окраски текстов. L1 и L2 регуляризация.
8. Способы оценки качества моделей: holdout и кросс-валидация.
9. Метод ближайших соседей.
10. Деревья решений, случайный лес, градиентный бустинг.
11. Решение соревнования на платформе Kaggle.
12. Способы построения композиций моделей.
13. Алгоритмы кластеризации: K-means, DBscan, агломеративная кластеризация.
14. Методы понижения размерности на основе матричных разложений (PCA и SVD). T-SNE.

15. Введение в нейронные сети - полносвязные нейросети, метод обратного распространения ошибки, инициализация весов, нелинейности.
16. Обзор стохастических методов оптимизации первого порядка.
17. Можно предложить свой вариант темы

(согласовать с преподавателем обязательно)!

При оформлении необходимо соблюдать следующую структуру:

1. Титульный лист
2. Оглавление (сформированное автоматически)
3. Введение
4. Содержательная часть
5. **Теория, решение индивидуальной задачи или обязательно проработать самостоятельно**
6. Заключение
7. Список литературы

Типовые вопросы к экзамену (3 семестр):

1. Задачи и терминология машинного обучения: supervised и unsupervised задачи; регрессия, классификация, кластеризация.
2. Объект, признак, типы признаков, методы работы с ними. Метрики качества. Инструменты интеллектуального анализа данных.
3. Изучение основ работы с векторными данными и визуализацией. Градиент.
4. Методы оптимизации гладких функций.
5. Реализация градиентного спуска для линейной регрессии.
6. Выделение признаков из текста: one-hot encoding, стемминг, лемматизация, tf-idf преобразование.
7. Логистическая регрессия на примере задачи эмоциональной окраски текстов. L1 и L2 регуляризация.
8. Способы оценки качества моделей: holdout и кросс-валидация.
9. Метод ближайших соседей.
10. Деревья решений, случайный лес, градиентный бустинг.
11. Решение соревнования на платформе Kaggle.
12. Способы построения композиций моделей.
13. Алгоритмы кластеризации: K-means, DBscan, агломеративная кластеризация.
14. Методы понижения размерности на основе матричных разложений (PCA и SVD). T-SNE.
15. Введение в нейронные сети - полносвязные нейросети, метод обратного распространения ошибки, инициализация весов, нелинейности.
16. Обзор стохастических методов оптимизации первого порядка.