Машинное Обучение МФТИ

Практическое задание №2: Решающие деревья

В качестве второй домашней работы студентам предлагается принять участие в соревновании по предсказнию уровня заработной платы на Kaggle. Для выполнения домашнего задания используйте ipython notebook с небольшими заготовками кода.
Соревнование общее – бонусы получают три первые места в каждой учебной группе.

Настройка окружения, инструкция по отправке решения

1. Вы можете загрузить `cart_trees.ipynb` - это ноутбук с кодом, который необходимо доделать, следуя инструкциям.

2. Установите Anaconda для **Python 2.7**.

3. Установите GraphViz.

4. Сохранить ipynb с выполненным экспериментом. Нужно запаковать, переименовать `hw02_<фамилия>.` и отправить в приватный канал в piazza с названием вида `HW2_группа_фамилия`. В ноутбуке должны быть описаны эксперименты, которые показали улучшение, и приведен код, который их выполняет. Можете описать интересные идеи, которые не сработали.

Описание данных и метрики

Для решения задания нужно обучить модель на обучающей выборке `adult.data`, сделать предсказание на `adult.val` и оправить свое решение на kaggle. Для ранжирования решений на `leadboard` используется метрика F_1 – описание её можно посмотреть, например, здесь: Wikipedia.

$$F_1 = \frac{precision + recall}{2\ precision\ recall}$$

Каждый объект в обучающей (`adult.data`) и тестовой (`adult.val`) выборках представляет собой набор из 14 значений как числовых, так и категориальных.

Задания, которые нужно выполнить

1. Визуализируйте взаимоотношения всех числовых характеристих - покажите их взаимное влияние на графике.

2. Реализуйте алгоритм CART с друмя метриками:

   (a) индексом Gini,

   (b) и критерием Twoing,

   (c) можете реализовать ещё метрики, это будет отмечено дополнительными баллами.

   Посмотрите, как работают эти критерии при разных параметрах построения дерева. Описание алгоритма CART можно посмотреть здесь.

3. (По желанию.) Реализуйте прунинг, описание алгоритма есть в ноутбуке, а дополнительное описание можно посмотреть здесь. В некоторых источниках к прунингу относят критерии останова роста дерева, так что в этой терминологии задание - реализовать post-pruning.

4. Визуализируйте ваше получившееся после обучения на датасете дерево (с помощью GraphViz).

5. Визуализируйте эффективность решения разных вариаций решающего дерева, сравните их на одной картинке. Дополнительно можно сравнить с классификатором на основе kNN.

6. Реализуйте алгоритм построения леса. Для этого вам нужно реализовать процедуру бэггинга. Подробнее на эту тему пожно прочитать в книге Elements of Statistical Learning, с. 282 и далее.

7. **Задание считается сданным после отправки ipython notebook, с описанием и кодом проведенных экспериментов, наглядными графиками и правильными выводами**

Методические указания

1. При подборе параметров модели рекомендуется использовать только часть обучающей выборки, для того чтобы сократить время обучения.

2. Согласно правилам соревнований нельзя делать больше 3х коммитов в систему в сутки. Из этого надо сделать следующие выводы:

   (a) Обучаться нужно локально (cross-validation) и только после получения результата, который вы считаете удовлетворительным, нужно делать submit в систему.

   (b) Начать делать домашнее задание стоит заблаговременно.

3. Обратите внимание, что публичные результаты на kaggle рассчитываются только по части контрольной выборки, и будут рассчитаны по всей контрольной выборке после окончания соревнования. Будьте аккуратны с переобучением.

4. Победители получают бонусные балы – шарить решение не выгодно.

Разница между списыванием и помощью товарища иногда едва различима. Мы искренне надеемся, что при любых сложностях вы можете обратиться к семинаристам и с их подсказками самостоятельно справиться с заданием. При зафиксированных случаях списывания (одинаковый код, решение задачи), баллы за задание будут обнулены всем участникам инцидента.