ВМК

Задание 3. Композиции алгоритмов для решения задачи регрессии

Практикум 317 группы, 2018

Начало выполнения задания: 2 декабря 2018 года.
Жёсткий дедлайн: **23 декабря 2018 года, 23:59.**
Дедлайн по отчёту: **24 декабря 2018 года, 23:59.**

Формулировка задания

Данное задание направлено на ознакомление с алгоритмами композиций.

В задании необходимо:

1. Написать на языке Python собственную реализацию методов случайных лес и градиентный бустинг. Прототипы функций должны строго соответствовать прототипам, описанным в спецификации и проходить все выданные тесты. Задание, не проходящее все выданные тесты, приравнивается к невыполненному. При написании необходимо пользоваться стандартными средствами языка Python, библиотеками numpy, scipy и matplotlib. Библиотекой scikit-learn пользоваться запрещается, если это не обговорено отдельно в пункте задания.
2. Провести описанные ниже эксперименты с выданными данными.
3. Поучаствовать в соревновании на платформе kaggle inclass. При подготовке решения разрешается пользоваться любыми библиотеками. Количество посылок за день ограничено!
4. Написать отчёт о проделанной работе (формат PDF). Отчёт должен быть подготовлен в системе LaTeX.

Список экспериментов

Эксперименты этого задания необходимо проводить на датасете данных о продажах недвижимости. Данные можно скачать со страницы соревнования https://www.kaggle.com/c/cmc-msu-317-prac-fall-2018/.

Первая часть (20 баллов)

1. Проведите минимальную обработку имеющихся данных. Разделите данные на обучение и контроль, переведите данные в numpy ndarray.
2. Исследуйте поведение алгоритма случайный лес. Изучите зависимость RMSE на отложенной выборке и время работы алгоритма в зависимости от следующих факторов:
   - количество деревьев
   - размерность подвыборки признаков для одного дерева
   - максимальная глубина дерева (+случай, когда глубина неограничена)
3. Исследуйте поведение алгоритма градиентный бустинг. Изучите зависимость RMSE на отложенной выборке и время работы алгоритма в зависимости от следующих факторов:
   - количество деревьев
   - размерность подвыборки признаков для одного дерева
   - максимальная глубина дерева (+случай, когда глубина неограничена)
   - выбранный learning_rate
     (Каждый новый алгоритм добавляется в композицию с коэффициентом γ * learning_rate)

Вторая часть (25 баллов)

В этом задании вам предлагается поучавствовать в учебном конкурсе, посвящённому предсказанию цены на недвижимость. Доступ к контесту можно получить у преподавателя. Метрика оценивания — RMSE.

Для получения баллов за задания, необходимо преодолевать бейзлайны. На преодоление каждого бейзлайна отводится некоторый срок. Если к этому сроку бейзлайн не преодолён, от итоговой оценки отнимается 5 штрафных баллов.

По окончании конкурса необходимо будет сдать отчёт:

- найденные в данных закономерности
- краткое описание выбранных моделей/использованных библиотек
- качество моделей на локальном контроле и на public/privat лидерборде
- выбранную методику тестирования моделей

После окончания контеста трём лучшим (на private лидерборде) из числа преодолевших **все** бейзлайны назначается 15, 13 и 10 баллов соответственно. Остальным участником, преодолевшим все бейзлайны, назначается от 1 до 8 бонусных баллов. Количество баллов линейно зависит от позиции в лидерборде (лучшему 8, худшему 1).

Участникам, проявлявшим какую-либо полезную активность (выкладывание решений в открытый доступ, активное обсуждение в форуме соревнования), могут быть назначены дополнительные бонусные баллы (на усмотрение преподавателя).

Требования к реализации

Прототипы всех функций описаны в файлах, прилагающихся к заданию.

Среди предоставленных файлов должны быть следующие модули и функции в них:

1. Модуль `ensembles.py` с реализациями случайного леса и градиентного бустинга. Алгоритмы должны соответствовать классическим реализациям, разобранным на лекции.

   Для одномерной оптимизации используйте функцию minimize_scalar. Разрешается использовать класс DecisionTreeRegressor из библиотеки scikit-learn.