ВМК

Задание 3. Композиции алгоритмов для решения задачи регрессии

Практикум 317 группы, 2019

Начало выполнения задания: 3 декабря 2020 года.
Жёский дедлайн: **27 декабря 2020 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. Написать отчёт о проделанной работе (формат PDF). Отчёт должен быть подготовлен в системе LaTeX.
3. Написать реализацию веб-сервера с требующейся функциональностью. Обернуть своё решение в докер контейнер.
4. Весь код, написанный во время задания, должен быть размещён в приватном репозитории. Требования к ведению репозитория представлены ниже.

Экспериментальная часть задания

Эксперименты этого задания необходимо проводить на датасете данных о продажах недвижимости. Данные можно скачать по ссылке.

Реализация алгоритмов (10 баллов)

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.

Эксперименты (15 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)

Обратите внимание! Для исследования зависимости от количества деревьев не обязательно с нуля переобучать модель.

Инфраструктурная часть

Реализация веб-сервера (15 баллов)

В этом задании вам предлагается спроектировать веб-интерфейс для взаимодействия с вашей моделью. Считайте, что назначение вашего интерфейса — обучение моделей человеком, который не знает языка Python. Это творческое задание, вы можете использовать при реализации всё, что считаете нужным.

1. (7 баллов) В интерфейсе должна быть предусмотрена функция создания новой модели. При создании новой модели должно быть возможно указать её тип (случайный лес или градиентный бустинг), а так же её гиперпараметры. В интерфейсе должна быть предусмотрена функция обучения модели на любом датасете, соответствующем заданному формату. Модель обязательно должна поддерживать обучение по поданному .csv файлу, где каждый столбец задаёт отдельный признак.

2. (4 балла) В интерфейсе обязательно должна быть предусмотрена функция просмотра информации о модели. Пользователь должен иметь возможность получать информацию о гиперпараметрах модели, датасете, на котором она обучалась, а также о значении функции потерь после каждой итерации.

3. (4 балла) В интерфейсе должна быть предусмотрена возможность сделать предсказание обученной моделью на датасете, соответствующем по формату датасету, на котором она обучалась.

Ведение проекта (10 баллов)

Весь код вашего задания должен быть выложен в приватный github репозиторий. В репозитории должен быть указан README.md файл, объясняющий, как необходимо пользоваться вашей системой. Качество кода влияет на итоговую оценку,код должен быть структурированным и понятным. За качественное ведения репозитория (см. соответствующую лекцию) будут назначаться бонусные баллы (до двух баллов). Под качественным ведением подрузамевается:

1. Основная разработка ведётся не в master, а в отдельных ветках. Ветка соответствует решение некоторой глобальной задачи.

2. Одно важное изменение в коде — один коммит в системе.

3. Обновление master ветки происходи посредством pull request и merge.

Решение должен быть обёрнуто в докер контейнер. В репозитории должен содержаться DockerFile, а также инструкция по сборке. Образ вашего контейнера должен быть залит на dockerhub.

Бонусная часть (до 5 баллов)

В этом задании нет чётких условий для бонусной части. Дополнительные баллы могут быть поставлены за любой хорошо реализованный дополнительный функционал, не описанный в задании. Не забудьте в отчёте / при посылке задания написать, что за дополнительный функционал вы реализовали.