Лабораторная 2 - supervised

github classroom task
Порядок сдачи и защиты лабы описан тут
Дедлайн указан в таблице с баллами

Задание

1. Изучите теормин к лабораторной работе
2. Исследуйте и подготовьте свой датасет из 1 лабы
   Постройте графики по вашим данным, чтобы понять как они устроены. Обработайте отсутствующие значения, выбросы. Извлеките данные из текстовых признаков, если они у вас есть. Сделайте признаки из категориальных переменных
3. Реализуйте baseline-решение
   Оно позволит понять, насколько обученная далее модель лучше, чем if-else
4. Выберите модель обучения с учителем
   Это может быть одна из разобранных на лекциях моделей: линейная/логистическая регрессии, дерево решений, случайный лес, бустинг. Можете выбрать и изучить самостоятельно и другую модель, если она не проще перечисленных
5. Найдите реализацию модели в sklearn
6. Разделите датасет на валидационную, тестовую и обучающую выборки
   Не забудьте о стратификации и не создавайте data leak по времени
7. Обучите модель из фреймворка и подберите лучшие гиперпараметры
   Изучите гиперпараметры модели из фреймворка. Реализуйте обучение модели на тренировочной выборке. Далее реализуйте перебор гиперпараметров модели, чтобы найти гиперпараметры, максимизирующие качество модели на валидационной выборке
   Перебирать гиперпараметры можно наивно, можно воспользоваться sklearn.grid search или optuna
8. Измерьте качество модели
   Выберите подходящие метрики качества для вашей задачи. Измерьте метрики на тренировочной и валидационной выборке, сравните их. Если модель переобучена, исправьте это. Покажите качество модели на каждой эпохе обучения с помощью графиков
9. Реализуйте эту же модель самостоятельно
   В реализацию добавьте как минимум 2 гиперпараметра. Повторите обучение, перебор гиперпараметров и измерение качества для вашей реализации модели
10. Сравните качество вашей модели с моделью из фреймворка
    Если качество значительно различается, то ищите ошибку. Идеального совпадения метрик не требуется

Использование jupyter notebook

Попытка показать графики, метрики и сэмплы датафрейма без юпитер ноута сделает проверку сильно неудобнее. В юпитер можно импортировать питон файлы. Можно написать основной код в них, а визуализировать в ноуте, если вам так удобнее. Визуализации только в ноуте - strictly recommended

Теория

В каждом вопросе про метрики, формулы и тп нужно знать как это рассчитывается и почему формулы именно такие. Зубрить сами формулы не нужно

В вопросах про модели подразумевается: Как устроена, как обучается, какие гиперпараметры, какие требования к данным и к чему уязвима

Преподаватель может задать доп вопрос об использованном вами термине, чтобы понять, что вы понимаете его смысл

1. Чем обучение с учителем отличается от обучения без учителя. Какое отношение оно имеет к кластеризации, классификации, регрессии, ранжирования, генерации
2. Что такое параметры модели? Что такое гиперпараметры?
3. Как и зачем перебирать гиперпараметры
4. Что такое мягкая классификация
5. Что такое переобучение
6. Что такое бейзлайн? Что такое наивный алгоритм? Какие примеры наивных алгоритмов можно привести для задач классификации и регрессии
7. Как и зачем дискретизировать, бинаризовать, нормализовывать и взвешивать признаки?
8. Как и зачем делать one-hot encoding? Можно ли это делать функцией pandas.get_dummies? Чем one-hot отличается от binary-encoding?
9. Как вычисляется cross-entropy и почему она cross?
10. Зачем делить данные на train, test, val? Чем val отличается от test? Кросс-валидация, ее виды. Что такое data leak
11. Что такое регуляризация? Как устроены L1 и L2 регуляризации
12. Метрики классификации: precision, recall, accuracy, F1, F-beta, roc auc кривая, precision-recall кривая и как они строится, confusion matrix
13. Метрики регрессии: mae, mse, (s-)mape
14. Вопрос-приз: можно ли использовать модели машинного обучения для заполнения пропущенных данных в датасете?
15. Дисбаланс классов: как влияет на метрики и модели, как с этим справиться
16. Градиентный спуск. Как оптимизирует параметры моделей, что такое пространство ошибок, чем отличаются (мини-)батчевый и стохастический
17. Модель линейная регрессия
18. Модель логистическая регрессия
19. Модель дерево решений, модель случайный лес. Как считается важность признаков в дереве

20. Ансамблирование алгоритмов. Бустинг. Другие алгоритмы понятно описаны тут и тут