Лабораторная 3 - unsupervised

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

Задание

1. Изучите теорию 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 без учителя
   Это может быть одна из разобранных на лекциях моделей. Можете выбрать и изучить самостоятельно и другую модель, если она не проще разобранных
5. Найдите реализацию модели в каком-либо фреймворке
   Рекомендуется sklearn, но если у вас специфические запросы, то можно использовать любой другой
6. 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
7. Измерьте качество модел
   Выберите подходящие метрики качества для вашей задачи. Измерьте метрики. Если модель переобучена, исправьте это. Покажите качество модели на каждой эпохе обучения с помощью графиков, если алгоритм это позволяет
8. Реализуйте эту же модель самостоятельно
   В реализацию добавьте как минимум 2 гиперпараметра. Повторите обучение, перебор гиперпараметров и измерение качества для вашей реализации модели
9. Сравните качество вашей модели с моделью из фреймворка
   Если качество значительно различается, то ищите ошибку. Идеального совпадения метрик не требуется

   Вы можете показать примерный вид латентного пространства с помощью метода снижения размерности, например PCA. Если при этом данные явно плохо кластеризуются или кластеры сильно пересекаются, то попробуйте другой метод, например umap

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, какие сценарии применения

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

1. что такое проклятие размерности
2. модель k-means
3. модель dbscan
4. модель agglomerative clustering
5. метрики качества кластеризации: внутри- и межкластерное расстояния, силуэт
6. влияние признаков с разным размахом вариации на k-means
7. Методы снижения размерности зачем нужны? На чем лучше обучать модели: на данных до снижения или после?
8. модель pca
9. модель t-sne
10. как использовать алгоритмы обучения без учителя для поиска выбросов
11. что такое approximated nearest neighbors (ann) search? Как использовать kd-tree для реализации этого алгоритма? Какое время работы у ann?
12. как автоматически определить количество кластеров в данных?
13. если для каждой точки в датасете заранее известна метка кластера, к которой она принадлежит, то как сопоставить результаты кластеризации с истинными метками? То есть как построить маппинг между метками кластеров и метками классов