Лабораторная 3 - supervised

[github classroom task](https://classroom.github.com/a/dlBloc3S)

Порядок сдачи и защиты лабы описан [тут](https://docs.google.com/document/d/1pj0LIO8kuOzqxgxW-dfZkQFhwZ9dkQ1hIxg5Va39x7U/edit?usp=drive_link)

Дедлайн указан в таблице с баллами

# Тайный метод быстрой сдачи лабы на полный балл

1. Проверьте, что код загружен в main ветку
2. Проверьте, что ничего из раздела Популярные замечания не нужно вашему коду
3. Перед вопросом в чат проверить faq лабы
4. Если видите в чате вопрос 2+ раз, пингуйте практика добавить вопрос в faq

# Задание

Обучите две модели обучения с учителем и сравните их предсказания и качество. На лекции не разбирался код моделей, поэтому можно выбрать любую

Первую модель возьмите из библиотеки (например из рекомендованных ниже библиотек). В качестве второй модели реализуйте первую самостоятельно. Нужно реализовать:

* конфигурацию модели гиперпараметрами
* обучение модели - настройку параметров  
  алгоритм оптимизации можно взять готовый
* предсказание моделью

Не забудьте исследовать свой датасет: построить графики по признакам, таблицу корреляции. По ним будет легче придумать новые признаки, чтобы улучшить предсказание модели. Придумайте хотя бы 1 признак

Метрики качества моделей должны быть одинаковыми или иметь пренебрежимо малую разницу, допустим 5% отклонения одной метрики от другой. Данные должны быть поделены на train, test, val, метрики считать на test части

# Задание (для групп 33-37)

1. Разбейте набор данных на три части (train, valid, test).
2. Выберите по одной модели из каждого типа:
   * Ближайшего соседа.
   * Линейной.
   * Деревья принятия решений.
   * Ансамбль.
   * Выберите функцию качества или ошибки.
3. Для каждой модели постройте график зависимости выбранной функции качества или ошибки от одного из числового гиперпаарметра на train, valid, test множествах. Обратите внимание, что некоторые гиперпараметры лучше перебирать в логарифмированном пространстве (перебирать в линейном, а перед подстановкой брать экспоненту), например: learning rate, коэффициент регуляризации, коэффициент сглаживания и т.д. Остальные гиперпараметры можно оставить по умолчанию, но лучше тоже понастраивать руками.

## Данные

В датасете из 1 лабы выберите target - признак, который полезно предсказать  
Это такие данные, которые сервис не знает заранее, но хотел бы знать. Например, для авто ру это будет цена автомобиля. Для циана - цена квартиры или аренды. Для шахматных партий - выигрыш первого или второго игрока или пат. Для сервиса кино - рейтинг фильма

Брать другой датасет нельзя

Используйте остальные признаки для предсказания таргета. Не забудьте предобработать данные. Умение удалить выбросы, нормализовать, логарифмировать, one-hot’нуть, перемножить, дискретизовать, бинаризовать требующие этого признаки позволяет перегонять соперников в соревнованиях, получать наибольшие премии на работе и, самое важное, лучшие баллы за лабу. Халатное отношение к работе с данными будет отражаться на баллах

Опционально: нарисуйте decisionBoundary график. Можете выбрать 2 произвольных признака или [изучить](https://skine.ru/articles/235125/) и применить алгоритмы снижения размерности, чтобы отобразить все признаки на двухмерном графике. Мы разберем эти алгоритмы на лекции чуть позже

# Рекомендации

* **не стесняйтесь задавать вопросы в чат курса**
* не выбирайте нейросетевые модели, если не понимаете на 101% как они работают и обучаются
* проверять 350 деревьев решений не хочется, поэтому выбирайте разные алгоритмы, оригинальность засчитывается в карму

Для реализации рекомендуются эти python библиотеки:

* sklearn
* fastai
* catboost
* xgboost

Но вы можете брать и другие

# Популярные замечания

* используйте конфиг файл для констант
* не коммитить .idea
* функции длиннее 20 строк - плохо. Их можно декомпозировать
* в начале кода лабы нет комментария со ссылкой на ваш репозиторий с 1 лабой
* данные не предобработаны
* не показано, что модель не переобучена
* нет сравнения предсказаний (например для классификации - гистограмм скоров) и метрик моделей
* если на scatterplot'е точки сливаются, стройте hist2d
* не рисуйте сравнение ошибок как plot() для ошибок либы и вашей. Посчитайте ошибки и по ним постройте гистограмму

# Теормин

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

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

1. Чем обучение с учителем отличается от обучения без учителя. Какое отношение оно имеет к кластеризации, классификации, регрессии, ранжирования, генерации
2. Что такое мягкая классификация
3. Что такое бейзлайн? Что такое наивный алгоритм? Какие примеры наивных алгоритмов можно привести для задачи классификации, регрессии, генерации, рекомендации
4. Линейная и нелинейная зависимость. Как выглядит мультимодальное распределение, чем оно неудобно для ML-алгоритмов. Придумайте пример признака, имеющего мультимодальное распределение (например из своего датасета из 1 лабы)
5. Как и зачем дискретизировать, бинаризовать, нормализовывать и взвешивать признаки?
6. Как и зачем делать one-hot encoding? Можно ли это делать функцией pandas.get\_dummies? Чем one-hot отличается от binary-encoding?
7. Как вычисляется cross-entropy и почему она cross?
8. Зачем делить данные на train, test, val? Чем val отличается от test? Кросс-валидация, ее виды. Что такое data leak
9. L1, L2 регуляризации
10. Метрики классификации: precision, recall, accuracy, F1, F-beta, roc auc (+roc auc кривая (+как строится)), confusion matrix
11. Метрики регрессии: mae, mse, (s-)mape
12. Вопрос-приз: можно ли использовать нейросети для генерации данных для других ML-алгоритмов?
13. Дисбаланс классов: как влияет на метрики и модели, как с этим справиться
14. Вопрос от которого можно отказаться: метрики расстояний в неевклидовом пространстве. Чем оно лучше евклидова для алгоритмов машинного обучения. Можно начать с [этого](https://dawn.cs.stanford.edu/2018/03/19/hyperbolics/) сайта
15. Градиентный спуск (батчевый). Как оптимизирует параметры моделей
16. Модель линейная регрессия
17. Модель логистическая регрессия
18. Модель kNN, **ядра kNN**
19. (исключен) Модель Байесовский классификатор
20. Модель дерево решений, лес. Как считается важность признаков в дереве
21. Ансамблирование алгоритмов, бэггинг. Понятно описано [тут](https://academy.yandex.ru/handbook/ml/article/ansambli-v-mashinnom-obuchenii) и [тут](https://alexanderdyakonov.wordpress.com/2017/03/10/c%D1%82%D0%B5%D0%BA%D0%B8%D0%BD%D0%B3-stacking-%D0%B8-%D0%B1%D0%BB%D0%B5%D0%BD%D0%B4%D0%B8%D0%BD%D0%B3-blending/)

# FAQ

Будет дополняться

1. как найти новые фичи?  
   изучите графики пар фич. Например scatterplot, hist2d. Особенно полезно будет посмотреть на пары (признак, таргет). Так можно увидеть новые закономерности