Проект по курсу "Теоретические основы нейронных сетей и машинного обучения"

Задание:

Предсказать вероятность того, что заемщик будет испытывать финансовые трудности в ближайшие 2 года ("Будет", "Не будет"). Это поможет банку определить, одобрять ли кредит.

Описание переменных:

Target | Заемщик имел просрочку платежа на 90 дней или хуже (1 - имел, 0 - не имел)

Age | Возраст заемщика (целое число)

FamilySize | Количество иждивенцев в семье заёмщика, исключая самого заёмщика (супруг, дети и т. д.) (целое число)

PastDueLess_60 | Сколько раз заемщик просрочивал платеж на 30-59 дней, но не более того за последние 2 года (целое число)

PastDue60_90 | Сколько раз заемщик просрочивал платежи на 60-89 дней, но не более того за последние 2 года. (целое число)

PastDue90_More | Сколько раз заемщик просрочивал платеж на 90 дней и более (целое число)

DebtRatio | Ежемесячные платежи по долгам, алименты, расходы на проживание, разделенные на ежемесячный доход (процент)

MonthlyIncome | Ежемесячный доход (число)

TotalBalanceDivideCreditLimits | Общий баланс по кредитным картам и личным кредитным линиям, за исключением недвижимости и без долгов в рассрочку, таких как автокредиты, разделенный на сумму кредитных лимитов (процент)

OpenLoans | Количество открытых кредитов (в рассрочку, таких как автокредит или ипотека) и кредитных линий (например, кредитные карты) (целое число)

RealEstateLoans | Количество ипотечных кредитов и кредитов на недвижимость, включая кредитные линии под залог жилья (целое число)

Инструкции по оформлению:

Каждый этап должен быть выделен заголовком, например:

Предобработка данных

К-ближайших соседей.

Заголовок создается с помощью знака решетки (одна решетка - большой заголовок, две решетки - заголовок поменьше.)

Выводы должны быть выделены, например: "**Вывод**: эта модель работает лучше так как...". Текст выделяется жирным шрифтом с помощью кнопки "B" в шапке редактирования текстовой ячейки или с помощью знака звездочки: **Вывод**.

Старайтесь для "чистовика" писать код последовательно и понятно. Если видите, что какой-то кусок повторяется - попробуйте выделить его в функцию и используйте функцию. Например, процесс обучения модели может быть выделен в функцию такого типа:

```
def train_best(model, X_train, y_train, X_test, y_test, parameters_dict):
    # YOUR CODE ---
    # do GridSearch
    # make prediction
    # plot metrics

    return best_model
```

И далее использован каким-то подобным образом:

```
knn = KNN()
params_knn = {'k': range(1,10)}
best_knn = train_best(knn, X_train, y_train, X_test, y_test, params_knn)
prediction = best_knn.predict(X_test)
```

Пошаговое задание:

1. Определите тип задачи. Объясните своими словами, что за метрика выбрана в качестве функции качества (AUC), почему выбрана она, как она считается и какие у нее худшее и лучшее значения.
2. Подготовьте данные:
    - идентифицируйте и уберите пропуски в данных
    - при необходимости переведите в подходящий тип данных
    - при необходимости отшкалируйте данные
    - при необходимости проведите создание и/или кодирование категориальных признаков
3. Изучите данные:
    - постройте не менее 10 графиков (вы можете построить для себя больше и оставить лучшие 10 или более)
    - каждый график должен иметь смысл и соответствующий вывод - какая зависимость признаков найдена, какое у данных особое распределение и что это означает, и т.д., можете попробовать сразу найти с помощью логистической регрессии самые важные признаки и/или убрать ненужные
    - у графиков должны обязательно быть подписаны оси, присутствовать легенда, название. Можно использовать seaborn: http://seaborn.pydata.org/introduction.html
    - например, найдите, есть ли у какого-либо признака зависимость с целевой переменной? или, есть ли в выборке выбросы исходя из какого-то признака? или, есть ли линейная/нелинейная зависимость между несколькими признаками?
4. Основываясь на полученных выводах из анализа данных:
    - при необходимости, уберите выбросы или отметьте их в качестве нового признака

- при необходимости, примените линеаризующие преобразования к данным (если это необходимо для модели)
- попробуйте придумать новые признаки - это может быть кластеризация в качестве нового признака (функция Kmeans), деление каких-то признаков в "корзины-категории", полиномиальные признаки. При добавление новых признаков создавайте отдельные переменные, чтобы можно было понять, помогают ли признаки какой-то модели или наоборот мешают. Тут же можно провести feature importance анализ.

5. Разделите данные на обучающую и тестовую выборку. Их вы будете использовать для валидирования ваших моделей. Так как разделение нужно только для вашего решения задачи, то фиксировать определенным образом разделение на трейн/тест необязательно.
6. Поиск лучшей модели:
    - Попробуйте алгоритмы машинного обучения, пройденные на курсе (k-ближайших соседей, линейные модели, SVM, SGD, решающие деревья)
    - для каждого найдите лучшую модель с помощью GridSearch с кроссвалидацией на 3 сегментах
    - постарайтесь перебирать как можно большее число параметров, но осмысленно!
    - попробуйте к моделям построить график зависимости качества от перебираемого параметра (пользуйтесь словарями cv_results, которые получаются после gridsearch), при возможности сделайте вывод, почему какие-то параметры могут работать в данной задаче лучше или хуже.
7. Выберите 2 лучших модели. Заново обучите их, но используйте весь доступный train датасет. Сделайте предсказания на test датасете. Сохраните предсказания моделей (по примеру sample_submission.csv). Отправьте в соревнование kaggle.
8. Попробуйте соединить предсказания двух ваших лучших моделей - возьмите среднее предсказаний этих двух моделей. Отправьте в соревнование. Стало ли качество лучше? Как вы думаете почему?

9. Бонус. Изучите модель случайных деревьев и модель градиентного бустинга. Попробуйте их применить к данным. Лучше или хуже работают эти модели? Вы можете отправить лучшее предсказание в соревнование.

Дедлайны:

Защита проектов с презентацией **24 апреля 2025 в 8.00.** На защите вы демонстрируете результаты разведывательного анализа данных (в т.ч. графики и самые интересные результаты по ним), рассказываете об алгоритмах, которые применяете для прогнозирования, **без** демонстрации кода и результатов моделирования.

Крайний срок отправки файла с кодом и описанием результатов моделирования – **27 апреля 2025, 23.59**. Файл нужно прикрепить в СДО в раздел Проект. Можно прикрепить ссылку на colab.

Тот же срок для отправки результатов прогнозирования в kaggle.

14 апреля 20025 в течение дня на лабораторных работах необходимо показать промежуточные результаты по проекту (как минимум выполнение пунктов 1-4 пошагового задания).

Оценивание:

Максимальная оценка за проект – 10 баллов. Вес проекта в итоговой оценке – 20%.

Команде, чья модель покажет лучшую способность предсказывать финансовые трудности заемщика, гарантируется оценка «отлично» по дисциплине с учетом сдачи всех лабораторных работ.