Предварительная оценка успешности учащихся на курсе

проблема

Как только в мире начал распространяться коронавирус, правительства начали устанавливать меры безопасности, в т.ч. и прямые либо косвенные запреты на передвижение и сборы в компании. Поэтому настал бум всевозможных онлайновых курсов.

Однако, согласно статистическим данным, 90% всех записавшихся пользователей активны только первые 10 дней, после чего не активны либо вообще уходят с курса. Это плохо как для работников курса, так и для самих учащихся.

Поэтому, на основе данных за несколько первых дней активности пользователя следует определить, сможет ли он пройти хотя бы половину всего курса, чтобы команда курса могла определить, в зависимости от предпочтений, бросить ли неактивного, или, наоборот, помочь ему. Также можно делать персональные рекомендации плохо успевающим учащимся. Это можно делать при помощи алгоритмов машинного обучения.

актуальность

как уже было сказано в описании проблемы, наибольшая популярность онлайн-курсов и распространённость этой проблемы стали выше после начала эпидемии, то есть не более чем год назад. Более того, эта работа нужна для обеих сторон: и для ученика (он сможет получать персональные рекомендации, подсказки или напоминания), и для учителя (он сможет обратить внимание на неактивных учеников, чтобы помочь им, или наоборот, на активных, чтобы заниматься с ними лучше).

цель

Обучить модель на основании данных по активности пользователя за первую неделю обучения и предсказать с её помощью, пройдёт ли пользователь 50% курса или более, или же нет.

задачи

1. Найти подходящий курс с достаточным количеством данных
2. проанализировать данные
3. извлечь из них необходимые характеристики
4. обучить на них модель
5. протестировать, валидировать
6. Получить предсказания
7. интерпретировать предсказания

описание решения

Данное решение представляет собой модель типа «случайный лес», обученную на следующих параметрах: Сколько баллов пользователь набрал за первую неделю, сколько задач решил правильно и неправильно, сколько комментариев оставил за неделю и (целевой признак, в обучение не включался) смог ли он за всё время обучения на курсе набрать хотя бы 50% от общего количества баллов.

техническая составляющая

Данное решение является программным модулем и, теоретически, может быть исполнено на любом компьютере с достаточными техническими требованиями (очень быстро работала модель на компьютере с процессором из 6 ядер частотой 3,17 ГГц, ОЗУ 16 ГБ) и всеми средствами, указанными в разделе «описание платформ и библиотек»

Иные решения, сравнение

К сожалению, иные решения подобных вопросов мне неизвестны, но я предполагаю, что, скорее всего, крупнейшие платформы онлайн-курсов (такие, как Stepik, Coursera и Сириус.Курсы) уже создали аналогичные модели и успешно их применяют. Сравнение их не представляется возможным, так как они, скорее всего, скрыты внутри сервисов и доступ конечного пользователя к ним невозможен, а обратная разработка данного вопроса, скорее всего, противозаконна (потому что найденные решения, весьма возможно, принадлежат сотрудникам курсов, зарегистрировавших свои решения как интеллектуальную собственность) или невозможна, как сказано выше.

описание работы инженерного решения

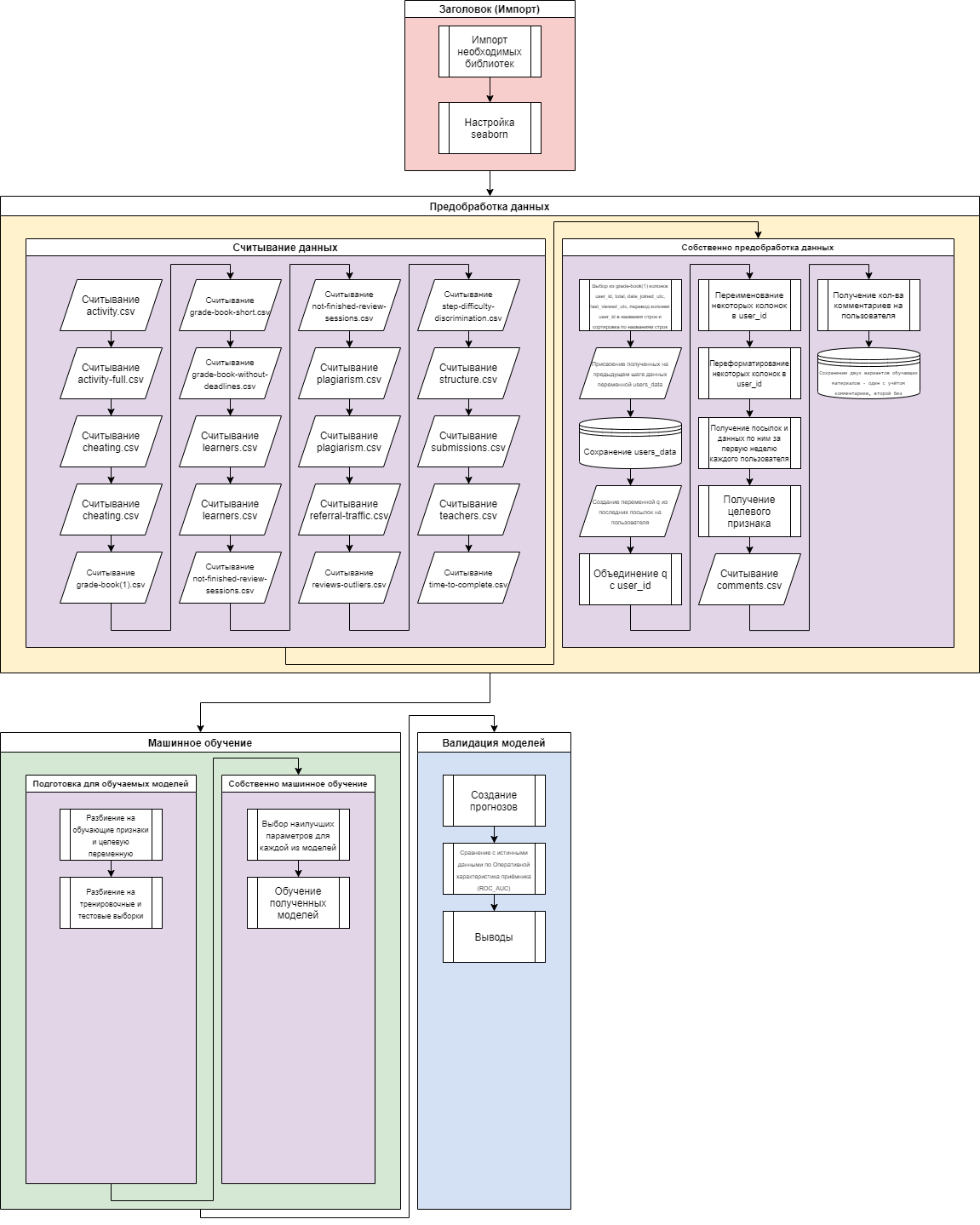
Предложенное мной решение не является техническим устройством, поэтому его детальное описание находится в разделе «программная реализация»

Возможность применения на практике

полученные при помощи обученной модели данные можно использовать во благо учеников (посылать им рекомендации, подсказки, если он надолго застрял, напоминания и т.д.) и учителей (чтобы указать им на тех, кто работает интенсивно и на тех, кто плохо справляется). Также по мере поступления новых данных модель следует дообучать, чтобы она не «теряла хватку», оставалась в курсе новых намечающихся закономерностей и т.д.

описание программы

Здесь я привожу процесс подготовки данных и то, как пользователь видит обучение машины (оно, как и многие другие вещи, скрыто и является чёрным ящиком) в виде блох-схемы средней степени подробности (то есть не сводящейся к общим инструкциям, но и не описывающую полную систему библиотечных функций). Большая часть процессов помечена как «предопределённый процесс», так как являются либо библиотечной функцией-«чёрным ящиком», либо комбинацией таких функций. Файл блок-схемы также доступен в векторном формате SVG.



Полная версия программы доступна в репозитории на Github, с комментариями на русском и английском языке.

список проделанных работ

1. Обучение способам машинного обучения
2. Работа на целевой проект

В первой работе я проходил курс обучения с созданием похожей модели с задачами, подобными задачам текущего проекта. Создавалась она примерно также, как и текущее изделие, за исключением различий в способах обработки данных. Эта модель создавалась исключительно в целях обучения работника основным принципам машинного обучения и работы с данными, и эта модель единственная, которая использовалась практически (её предсказания были посланы в тестирующую систему и оценены).

Вторая работа — мой текущий проект. Этот проект является модификацией старого относительно нового формата данных и способа обработки этих данных, но идеологически он мало чем отличается от предыдущего. К сожалению, моя модель пока не применялась на практике серьёзно (только оценивалась на существующих данных). Он должен пройти дополнительную проверку, что пока невозможно из-за малого количества новых данных, и одобрение заказчика на его внедрение.

описание использованных ресурсов

В качестве среды разработки был использован Visual Studio Code и дополнение Jupyter Notebook, отчёты с одного из курсов на Stepik, библиотеки pandas, numpy, sklearn и seaborn.

Проект основан на свободно распространяемых ресурсах, пока не зарегистрирован как чья-либо интеллектуальная собственность, поэтому сам абсолютно свободен, бесплатен и доброволен. Репозиторий на Github:

Практическая часть

Библиотеки и платформы

Visual Studio Code. Настраиваемый редактор кода для большинства типов файлов и языков. Многофункционален благодаря обширной базе расширений.

Python 3.9.1. Языковая платформа, на которой писался проект, и расширение для VS Code для работы с ним.

Jupyter Notebook. «Блокнот» для работы с Python, визуализациями и форматированным текстом (реализовано при помощи Markdown) в одном файле и расширение для VS Code для работы с Notebook прямо в редакторе.

Pandas 1.2.1. Библиотека для работы с данными.

NumPy 1.19. Библиотека для работы с высокоуровневыми функциями для многомерных массивов данных, обязательное требование pandas и SKLearn.

SciPy 1.6.0 Библиотека для инженерных расчётов. Обязательное требование для SKLearn.

Matplotlib 3.3.3. Библиотека для визуализации графиков, обязательное условие для seaborn и SKLearn.

Seaborn 0.11.1. Надстройка для matblotlib, позволяет строить улучшенные графики.

SKLearn 0.24.1 Библиотека, реализующая алгоритмы машинного обучения, метрики качества обучаемых моделей и т.д.

итоги работы

Данное решение является моделью вида «классификатор на основе случайного леса» для обеспечения знания о возможных успевающих и неуспевающих учениках онлайн-курса. Данная модель создана для применения на практике, но пока не может из-за недостаточной степени проверенности и валидированности модели. Скорее всего, созданы модели подобной направленности и, возможно, даже более хорошего качества, но, как уже было сказано ранее, изучить эти модели не представляется из-за недоступности их для конечных пользователей и незаконности процесса обратной разработки данных систем. В процессе работы над этим решением я научился основным принципам так называемого Data Science (англ. Наука о Данных, или данные как наука), т.е. обработке данных, извлечению из них нужной информации, обучению на ней машинных моделей для получения предсказаний и т.д. Так что этот проект будет полезен не только конечному заказчику, но и работникам, вовлечённым в проект.