Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Камкова Екатерина Александровна

Проверка самостоятельности студентов онлайн-курсов

Курсовая работа

Санкт-Петербург
2019

Оглавление

Введение

С развитием технологий люди начали задумываться об обучении через интернет. За последние несколько лет были созданы такие сайты как Coursera, Stepik и Open Education [5, 7, 6], где пользователи из разных стран в любое удобное им время могут изучить интересующие их МООК (массовые открытые онлайн-курсы) и получить сертификат о прохождении. В некоторых университетах уже сейчас прохождение МООК является необходимым условием получения зачёта. Но, поскольку обучение происходит онлайн, нет возможности напрямую наблюдать за прохождением. Следовательно, появляется необходимость в проверке самостятельности студентов. На данный момент уже существуют и активно используются различные подходы, такие как отслеживание посещённых страниц, просмотренных видео и др. В этой работе будет рассмотрен альтернативный способ поиска подозрительных на списывание студентов.

1. Постановка задачи

Целью данной работы является оздание приложения для выявления студентов, подозрительных на списывание, путём обучения алгоритмов на логах на сайте Open Education. Для её достижения были поставлены следующие задачи.

1. Извлечение необходимой информации из логов.
2. Поиск подходящей задаче модели машинного обучения.
3. Реализация классификации учеников.
4. Реализация базового консольного интерфейса.

2. Существующие решения

На данный момент каждый из трёх упомянутых выше сайтов имеет свои механизмы проверки студентов.

2.1. Coursera

Чтобы пройти курс на Coursera [5], все страницы курса должны быть помечены как пройденные. Для этого на страницах, содержащих материал в виде видео-лекций, необходимо просмотреть большую часть видео, на странице с текстовым материалом нажать на кнопку “Пометить как выполненное” и пройти тест на страницах с тестом. Более того, существует Кодекс Чести Coursera (Coursera Honor Code), которому каждый студент должен следовать. Но такие проверки можно легко подделать, если перемотать видео в конец, нажать на кнопку, не читая материал, и проигнорировать Кодекс Чести.

2.2. Stepik

На Stepik [7] страницы помечаются как пройденные при их посещении, а видео является просмотренными, если пользователь хоть раз включил их. Эти проверки ещё легче обойти.

2.3. Open Education

На Open Education [6] некоторые курсы предусматривают особые условия прохождения финального экзамена. Студент должен пройти экзамен под наблюдением сотрудника через веб-камеру, предварительно подтвердив свою личность (показав сотруднику удостоверение личности). Такие предосторожности значительно сложнее обойти, но они крайне трудоёмки, их просто невозможно распространить на весь курс.

2.4. Аналогичные исследования

Было решено усовершенствовать проверку самостоятельности студентов с помощью информации, полученной из логов. За последние несколько лет было проведено множество исследований [1, 3, 2, 4] возможностей, предоставляемых логами, но в основном они направлены на поиск оптимального формата курса или предсказание вылета студентов.

3. Реализация

3.1. Используемые технологии

Поскольку предстоит работа с логами, а впоследствии — машинное обучение, был выбран язык Python [11].

Программа была написана в PyCharm — интегрированной среде разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA [10].

Для представления данных в удобном для машинного обучения формате была использоана библиотека Pandas [8].

Также в работе были использованы реализации алгоритмов обучения от Scikit Learn [9].

- Метод опорных векторов (SVM) Идея данного метода заключается в представлении примеров как точек в гиперпространстве (размерность пространства равна количеству параметров, описывающих пример) и разделении его гиперплоскостью на две группы. Поскольку решений данной задачи может быть несколько, SVM выбирает решение таким образом, чтобы расстояние между гиперплоскостью и ближайшими точками данных было максимальным.

- Деревья принятия решений Дерево решений может быть представлено в виде двоичного дерева, каждый узел которого является входной переменной и точкой разделения для этой переменной. Листья таких деревьев — выходные переменные, использующиеся для предсказания. Предсказания производятся путём прохода по дереву к листу и вывода значения этого узла.

3.2. Создание параметров

EdX - крупный провайдер MOOK, на основе его документации была проведена работа с логами.

Если смотреть на информацию о действиях студентов во время прохождения онлайн курса, можно заметить, что поведение списывающего

человека похоже на поведение человека, уже знающего текущий материал. Например, студент, обладающий ответами, скорее всего перемотает или пропустит видео, не будет участвовать в форуме, с первого раза напишет тест, и студент, знакомый с материалом поступит также. Именно поэтому задача выявить людей, **подозрительных**, чтобы учитель потом мог проверить их знание и убедиться в их самостоятельности.

Для отслеживания действий студентов были предложены следующие параметры (было введено следующее обозначение: сессия — отрезок времени, когда события создавались не реже чем раз в 30 минут):

- Количество событий, созданных студентом
- Количество действий на форуме
- Количество переходов по предоставленным ссылкам
- Время, прошедшее от первого клика до последнего
- Время обучения (суммарное время сессий)
- Количество сессий
- Средняя продолжительность сессии
- Количество возвращений на предыдущие страницы
- Отношение количества посещённых страниц в блоке к количеству страниц в блоке
- Количество посещённых блоков
- Количество действий во время просмотра видео-лекций
- Среднее количество действий во время просмотра видео-лекций
- Среднее время, потраченное пользователем на просмотр видео-лекций

- Количество изменений скорости видео
- Среднее количество переходов вперёд во время просмотра видео-лекций
- Среднее количество переходов назад во время просмотра видео-лекций
- Количество просмотренных видео-лекций
- Среднее количество пауз
- Среднее количество возвращений к видео-лекциям
- Среднее количество попыток, необходимых для прохождения теста
- Количество пройденных тестов
- Количество начатых тестов

Средняя скорость обработки событий

$$= \frac{1200000\ events}{1\ min} \tag{1}$$

3.3. Исследование моделей

Для решения текущей задачи была необходима модель, производящая бинарную классификацию (0 — студент подозрителен, 1 — вне подозрений). Среди моделей, удовлетворяющих этому описанию, после анализа их принципов и основываясь на статье "Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention" [3] для рассмотрения были выбраны две модели:

3.3.1. Метод опорных векторов (SVM)

Для определения эффективности модели в текущей задаче используются меры:

TP (true positive) — истинно-положительное решение

TN (true negative) — истинно-отрицательное решение

FP (false positive) — ложно-положительное решение

FN (false negative) — ложно-отрицательное решение

- $$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$
- $$precision = \frac{TP}{TP + FP} \tag{3}$$
- $$recall = \frac{TP}{TP + FN} \tag{4}$$

Алгоритм SVM был протестирован на параметрах (добавление других параметров ухудшает предсказание):

Количество действий на форуме

Количество просмотренных видео-лекций

Среднее количество попыток, необходимых для прохождения теста

Количество пройденных тестов

Количество начатых тестов

В среднем алгоритм показал следующие результаты:

Accuracy: 0.68

value	precision	recall
0	0.99	0.15
1	0.65	0.99

То есть точность предсказания довольно низкая.

3.3.2. Деревья принятия решений

Данный алгоритм был простестирован на параметрах:

Количество действий на форуме

Количество возвращений на предыдущие страницы

Отношение количества посещённых страниц в блоке к количеству страниц в блоке

Количество посещённых блоков

Среднее время, потраченное пользователем на просмотр видео-лекций

Количество просмотренных видео-лекций

Среднее количество возвращений к видео-лекциям

Среднее количество попыток, необходимых для прохождения теста

Количество пройденных тестов

Количество начатых тестов

В среднем алгоритм показал следующие результаты:

Accuracy: 0.91

value	precision	recall
0	0.87	0.89
1	0.94	0.93

Точность предсказания данной модели значительно выше, поэтому она была выбрана для программы

3.4. Интерфейс

Созданный интерфейс поможет пользователю проверить студентов на списывание

1. Загрузка обучающей таблицы
   (a) Преобразовать логи и загрузить уже имеющиеся оценки
   (b) Воспользоваться уже существующей таблицей
2. Загрузка тестируемой таблицы

(a) Преобразовать логи

(b) Воспользоваться уже существующей таблицей без результатов

3. Возможность сохранения таблицы протестированных студентов

4. Добавление проверенных студентов в обучающую таблицу

Заключение

В рамках данной работы были решены следующие задачи:

1. Реализован алгоритм для извлечения необходимой информации из логов
2. Исследованы различные модели машинного обучения, выбрана подходящая
3. Реализована классификация учеников
4. Реализован базовый консольный интерфейс

Текущие результаты можно посмотреть по адресу [12]

Список литературы

[1] Hu Hui, Zhang Guofeng, Gao Wanlin, Wang Minjuan. Big data analytics for MOOC video watching behavior based on Spark. — URL: https://link.springer.com/article/10.1007%2Fs00521-018-03983-z (online; accessed: 07.06.2019).

[2] Brinton Christopher G., Chiang Mung. MOOC Performance Prediction via Clickstream Data and Social Learning Networks. — URL: https://ieeexplore.ieee.org/document/7218617 (online; accessed: 07.06.2019).

[3] Xing Wanli, Du Dongping. Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention. — URL: https://www.researchgate.net/publication/323784695 (online; accessed: 07.06.2019).

[4] Sinha Tanmay, Li Nan, Jermann Patrick, Dillenbourg Pierre. Your Click Decides Your Fate: Inferring Information Processing and Attrition Behavior from MOOC Video Clickstream Interactions. — URL: https://www.researchgate.net/publication/266141299 (online; accessed: 07.06.2019).

[5] Домашняя страница сайта Coursera. — URL: https://www.coursera.org/ (online; accessed: 07.06.2019).

[6] Домашняя страница сайта Open Education. — URL: https://openedu.ru (online; accessed: 07.06.2019).

[7] Домашняя страница сайта Stepik. — URL: https://welcome.stepik.org/ru (online; accessed: 07.06.2019).

[8] Домашняя страница сайта библиотеки Pandas. — URL: https://pandas.pydata.org/ (online; accessed: 07.06.2019).

[9] Домашняя страница сайта библиотеки scikit-learn. — URL: https://scikit-learn.org/stable/ (online; accessed: 07.06.2019).

[10] Домашняя страница сайта среды разработки PyCharm. — URL: https://www.jetbrains.com/pycharm/ (online; accessed: 07.06.2019).

[11] Домашняя страница сайта языка Python. — URL: https://www.python.org/ (online; accessed: 07.06.2019).

[12] Репозиторий с курсовой работой. — URL: https://github.com/katerina-kamkova/CheatCheck (online; accessed: 07.06.2019).