Принятие решений на основе данных
Практика в Microsoft Azure Machine Learning Studio

Azure ML Studio - один из вариантов визуального окружения для работы с задачами машинного обучения. В ней можно как собирать модели из готовых блоков (без программирования), так и добавлять в ваше решение нестандартные шаги путем включения функций, написанных на Python или R. В данном задании мы будем пользоваться только стандартными блоками.

Прежде чем начать выполнять задание, давайте зайдем в нужное нам окружение:

1. Переходим по ссылке: https://studio.azureml.net/.
2. Переходим по следующей последовательности: Sign up here -> Guest Workspace -> New Experiment. Если у вас есть аккаунт Microsoft (или вы готовы его завести), то можно им воспользоваться вместо гостевого. Это бесплатно и добавляет ряд удобств, например, все ваши эксперименты сохраняются на продолжительное время.
3. Во вкладке Experiments нажимаем на кнопку New + в левом нижнем углу и выбираем новый пустой эксперимент - Blank Experiment. Заметьте, что кроме пустого эксперимента, здесь можно выбрать множество готовых примеров решения разных видов задач. Если вам интересно, то можно после выполнения задания посмотреть какие-то из них.
4. Мы на месте - можно начинать строить свою первую модель машинного обучения!

Задача

В данном задании мы будем решать задачу предсказания оттока пользователей в телекоме (churn prediction). У нас есть данные о некотором числе пользователей, о каждом пользователе мы знаем 19 признаковых показателей и значение целевой переменной churn - ушел данный пользователь к другому оператору за определенный период времени или нет (если churn = true, значит ушел). Наша задача: используя эти данные обучить модель, которая будет способна достаточно точно предсказывать вероятность ухода новых пользователей по значению тех же 19 признаковых показателей. Соответственно, перед нами задача бинарной классификации.

Поехали!

Далее идет список шагов, которые нужно сделать, чтобы завершить задание. Если вам не понятно, как делать какой-то шаг, то ниже есть технические подсказки к каждому из них. Некоторые шаги отмечены тегом “вопрос” - это значит, что на этом шаге вам нужно ответить на поставленный вопрос в письме при сдаче задания. Соответственно, чтобы сдать задание, нужно прислать письмо с ответами на вопросы 2, 5, 7, 8, 9 (текстовые ответы + одна приложенная картинка для пункта 8).

Шаг 1. Импортируйте данные, используя блок Import Data. Ссылка на данные: https://raw.githubusercontent.com/nadiinchi/intro_python_sklearn_azure/main/train_mis.csv

Шаг 2 - вопрос. Посмотрите на данные и ответьте на следующие вопросы:

- Какие виды признаков представлены в данных? Если бы система не обрабатывала признаки автоматически при обучении модели, то какие признаки и как нужно было бы привести к числовому формату?
- Есть ли в данных пропущенные значения или выбросы? Напишите, какие проблемы в данных вы видите. Как их можно было бы исправить (например, на что заменить пропуски)?

Шаг 3. Разделите выборку на обучающую и валидационную части в пропорции 80 к 20 с помощью блока Split Data.
Шаг 4. Обучите линейную модель на обучающей части данных. Для этого сначала инициализируйте модель используя блок Two-Class Logistic Regression, а далее обучите ее, используя блок Train Model. Не забудьте назначить в последнем целевую переменную для предсказания.
Шаг 5 - вопрос. Посмотрите на веса полученной обученной модели. Назовите три наиболее важных для предсказания признака по мнению полученной модели. Как увеличение значений этих признаков смещают предсказание модели: в сторону отрицательного или в сторону положительного класса (в задаче положительный класс - клиент уходит)? Согласуются ли полученные результаты со здравым смыслом?
Не забывайте, что сравнивать веса признаков можно только в том случае, когда признаки отнормированы, здесь система сделала это за вас автоматически при обучении модели =)
Шаг 6. Оцените качество модели на обучающей и валидационной выборках. Для этого сначала получите предсказания на каждой из них с помощью блока Score Model (отдельный блок на каждую выборку), а после подсчитайте метрики качества на обеих выборках с помощью блока Evaluate Model (один блок на обе выборки - в нем будет удобно сравнить результаты).
Шаг 7 - вопрос. Какие значения Accuracy у вас получились на обучающей и валидационной выборках? Является ли полученная модель недообученной, нормальной или переобученной?
Шаг 8 - вопрос. Сделайте скриншот полученного схемы (всех использованных выше блоков, связанных между собой) и приложите его к письму при сдаче задания.
Шаг 9 - вопрос. Попробуйте вместо линейной модели применить другие модели классификации (дерево, случайный лес, бустинг на деревьях - смотрите разные модели, начинающиеся с Two-Class ...). Постарайтесь получить как можно более качественную модель. Укажите, какого максимального значения Accuracy на валидационной выборке вам удалось добиться и с помощью какой модели.

Успех!
Теперь вы можете решать простые задачи с помощью машинного обучения в визуальной среде. Если вы захотите в будущем воспользоваться данной средой, чтобы строить модели на своих данных, то советую вам завести аккаунт, чтобы можно было загружать данные в систему со своего компьютера и сохранять эксперименты. Также напоминаю, что в системе можно посмотреть готовые примеры решения разных задач.

Технические подсказки.

Шаг 1. Для импорта данных нужно сделать следующее:

- Выбираем слева блок Import Data (для поиска можно воспользоваться поисковой строкой) и перетаскиваем его на поле.
- Справа в свойствах блока выбираем в качестве источника данных Web URL via HTTP и указываем ссылку на данные в поле Data source URL.
- Указываем формат данных CSV и помечаем галочкой CSV or TSV has header row. Последнее указывает, что в нашем файле заданы названия колонок, и их нужно воспринимать именно как названия, а не как значения признаков для первого объекта.

Шаг 2. Чтобы посмотреть на данные, нужно запустить эксперимент (кнопка Run внизу), кликнуть правой кнопкой мыши на нижний кружок блока Import Data и выбрать Visualize.

Шаг 3. Для разделения данных нужно сделать следующее:

- Выбираем слева блок Split Data и перетаскиваем его на поле.
- Подаем ему на вход данные.
- Справа в свойствах блока в поле Fraction прописываем долю первой выборки (0.8 в нашем случае).

Шаг 4. Для разделения данных нужно сделать следующее:

- Выбираем слева блоки Two-Class Logistic Regression и Train Model и перетаскиваем их на поле.
- Подаем блоку Train Model на вход слева модель, на вход справа - обучающую часть данных (левый выход Split Data).
- Справа в свойствах блока Train Model запускаем Launch Column Selector: во втором поле выбираем column names, в третьем - churn. Таким образом мы указали модели, какой столбец в данных содержит целевую переменную.

Шаг 5. Чтобы посмотреть на веса модели, нужно запустить эксперимент (кнопка Run внизу), кликнуть правой кнопкой мыши на нижний кружок блока Train Model и выбрать Visualize.

Шаг 6. Для разделения данных нужно сделать следующее:

- Выбираем слева блоки Score Model (две штуки) и Evaluate Model и перетаскиваем их на поле.
- Подаем первому блоку Score Model на вход слева модель (выход блока Train Model), на вход справа - обучающую часть данных (левый выход Split Data).
- Подаем второму блоку Score Model на вход слева модель, на вход справа - валидационную часть данных (правый выход Split Data).
- Подаем блоку Evaluate Model на вход слева выход первого блока Score Model, на вход справа - выход второго блока Score Model.

Шаг 7. Чтобы посмотреть на качество модели, нужно запустить эксперимент (кнопка Run внизу), кликнуть правой кнопкой мыши на нижний кружок блока Evaluate Model и выбрать Visualize. Теперь, кликая на синий и красный квадратик, вы можете переключаться между качеством на обучающей выборке (синяя, левый вход) и валидационной выборке (красная, правый вход). Система показывает значение большого числа метрик, нас интересует цифра Accuracy.

Шаг 8. В полученной схеме нужно просто заменить блок Two-Class Logistic Regression на блок любой другой модели для бинарной классификации и запустить эксперимент.