

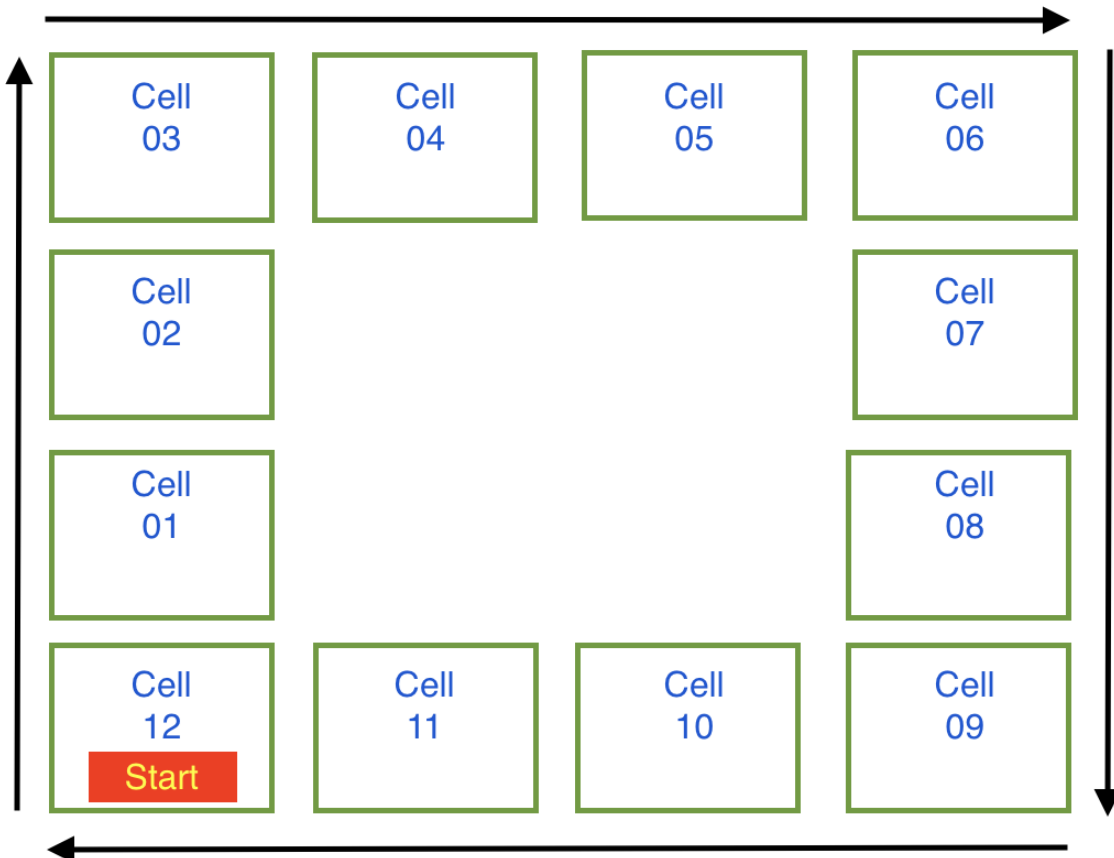
Тестовое задание DS\ML-ops.

Задание 1

Есть настольная игра, в которой:

- игроки перемещаются по полю посредством бросков игральных шестигранных кубиков
- движение идет по часовой стрелке
- игрок начинает игру с поля 12 и перемещается между полями на количество ходов, равное сумме выпавших при броске кубиков баллов.
- если сумма баллов превышает количество полей, игрок продолжает движение по полям по часовой стрелке. Пример: находясь на стартовом поле (Cell 12) при броске 3-х кубиков игрок получает суммарное количество баллов - 16, соответственно конечной точкой будет Cell 04.

Структура игрового поля



Рассчитать вероятность попадания на каждую ячейку игрового поля при разном количестве бросаемых игральных кубиков:

1. Используется 3 куба
2. Используется 6 кубов
3. Используется 12 кубов

Процентные значение попадания на каждую ячейку вписать в таблицу ниже.

	3 Cubes	6 Cubes	12 Cubes
Cell 01			
Cell 02			
Cell 03			
Cell 04			
Cell 05			
Cell 06			
Cell 07			
Cell 08			
Cell 09			
Cell 10			
Cell 11			
Cell 12			

Код представить в виде .py файла, помимо кода необходимо подробно расписать логику (отдельным текстовым файлом)

Задание 2

У нас есть таблица с агрегированными по 30 минут данными – **dataframe_updates.csv**

Поля в таблице содержат следующие данные:

- x - время по UTC,
- id - фиксированная точка в пространстве,
- scooters_on_parking – количество самокатов в данной точке на начало указанного временного промежутка,
- timezone – опорная точка по которой можно ориентироваться на таймзону.
- starts - количество стартов с парковки

Данную таблицу нужно преобразовать:

1. Сгруппировать записи стартов по 3 часа и по id пространственных точек, чтобы получить суммарное количество стартов для данного временного промежутка в этой точке
2. Помимо поля с общим количеством стартов, добавить значение количества самокатов за последний временной промежуток на основании scooters_on_parking (последние полчаса до заданного временного промежутка)
3. Привести время к нужной таймзоне, сейчас оно в формате UTC
4. Предложить решение, как инкрементально обновлять эти данные, если исходная таблица будет обновляться каждые 3 часа

Все манипуляции с данными производить SQL - запросами.

Задание 3

У нас есть онлайн-игра с нелинейной структурой уровней, т.е. уровни можно проходить в любом порядке, и у каждого уровня есть свой номер и сложность.

Необходимо разработать классификационную модель, которая будет предсказывать, покинет ли пользователь в течение следующих 7 дней

В прилагаемом датасете (user_activity.csv) - пользовательские данные.

Описание столбцов таблицы:

- **platform** - игровая платформа.
- **maxIncLevel** - максимальный уровень, пройденный игроком за все время с момента установки игры.
- **maxMainMapLevel** - номер наибольшего уровня, пройденного игроком за рассматриваемый временной период.
- **totalNumOfAttempts** - общее количество стартов уровней.
- **numberOfAttemptedLevels** - количество уникальных уровней, которые пытался пройти игрок.
- **attemptsOnTheHighestLevel** - количество попыток пройти уровень с наибольшим номером.
- **doReturnOnLowerLevels** - перепроходил ли игрок старые уровни? 1 = "Да", 0 = "Нет".
- **playedDays** - количество дней, в которые игрок стартовал хотя бы один уровень.
- **timePassedInMainMapGrind** - общее время, проведенное пользователем на игровых уровнях.
- **timePassedTotal** - общее время, проведенное в игре
- **spentBoosters** - количество потраченных бустеров (платные улучшения)
- **spentTotal** - сумма денег, потраченных игроком за все время с момента установки игры.
- **spentAtPeriod** - сумма денег, потраченных игроком за рассматриваемый временной промежуток.
- **purchases** - количество покупок, совершенных игроком за рассматриваемый временной промежуток.
- **churn** - таргет, остался ли пользователь в игре в течение последних семи дней, либо же ушел из игры. 1 - покинул игру. 0 - остался.

Код представить в виде .ipynb файла, помимо кода необходимо подробно расписать логику, выбранную модель (почему был сделан выбор в пользу этой модели), описать обнаруженные корреляции к уходу из игры; любые другие инсайты.

Задание 4

Стоит задачи минимизировать время в пути для водителя. Как построить маршрут так, чтобы водитель доезжал быстрее? Какие алгоритмы маршрутизации будете использовать, на что будете обращать внимание, и как измерите общую эффективность

Вопрос теоретический.