**Описание проекта**

Сравнение Москвы и Петербурга окружено мифами:

* Москва — мегаполис, подчинённый жёсткому ритму рабочей едели;
* Петербург — город своеобразной культуры, непохожий на Москву.

Некоторые мифы отражают действительность. Другие — пустые стереотипы. Бизнес должен отличать первые от вторых, чтобы принимать рациональные решения. На реальных данных Яндекс Музыки вы проверите гипотезы и сравните поведение пользователей двух столиц.

**Гипотезы**

1. Активность пользователей зависит от дня недели. Причём в Москве и Петербурге это проявляется по-разному.
2. Утром в понедельник в Москве преобладают одни жанры музыки, а в Петербурге — другие. Это верно и для вечера пятницы.
3. Москва и Петербург предпочитают разные жанры музыки. В Москве чаще слушают поп-музыку, в Петербурге — русский рэп.

**Данные**

Данные хранятся в файле yandex\_music\_project.csv. На платформе он находится по пути /datasets/yandex\_music\_project.csv. Скачать датасет для работы с ним вне проекта можно [здесь](https://code.s3.yandex.net/datasets/yandex_music_project.csv).

**Описание колонок:**

* userID — идентификатор пользователя;
* Track — название трека;
* artist — имя исполнителя;
* genre — название жанра;
* City — город пользователя;
* time — время начала прослушивания;
* Day — день недели.

**Инструкция**

Проект этого курса проверяет базовые знания Python и библиотеки pandas, поэтому будет проверен автоматически в Jupyter-тренажёре. Задания в таких проектах чаще всего предполагают один вариант решения — его и проверит тренажёр.

Решайте задания в несколько шагов:

1. Напишите код для задания в одной или нескольких ячейках.
2. Запустите ячейки, связанные с текущим заданием.
3. Нажмите кнопку «Проверить задание».
4. Если решение прошло проверку, нажмите кнопку «Следующее задание».
5. Вернитесь к шагу 1 и решите следующее задание.

Проект завершён, когда засчитаны все задания в тренажёре.

Обратите внимание, что при запуске ячеек тетрадки в произвольном порядке вы можете столкнуться с ошибками. Например, в том случае, если попытаетесь получить объект, который в данный момент не загружен в память. Прежде чем обращаться к ревьюеру или преподавателю, попробуйте удалить все объекты из памяти и перезапустить весь код. Сделать это можно в Jupyter Notebook с помощью команды Kernel → Restart & Run All.