GeekBrains

Курс Python для DataScience

Практическое задание

Инструкция к сдаче

1. Настоятельно рекомендуем сдавать практическое задание в виде ссылки на личный репозиторий на github.
2. Рекомендуемый способ организации данных в репозитории: создать отдельные папки по темам и помещать в них отдельные файлы для каждой задачи с правильным расширением.

Ссылка на инструкцию по работе с git и сдачу практики:

https://docs.google.com/document/d/1RAT_ukE39iOfbz1xa39QXae2hBUEZ4U6Fko_wFDdrsM/edit

Ссылка на видеокурс по Git:

https://geekbrains.ru/courses/66

Если остались сложности с системой git, то обратитесь к преподавателю или наставнику.

Тема “Вычисления с помощью Numpy”

Задание 1

Импортируйте библиотеку Numpy и дайте ей псевдоним np.

Создайте массив Numpy под названием a размером 5x2, то есть состоящий из 5 строк и 2 столбцов. Первый столбец должен содержать числа 1, 2, 3, 3, 1, а второй - числа 6, 8, 11, 10, 7. Будем считать, что каждый столбец - это признак, а строка - наблюдение. Затем найдите среднее значение по каждому признаку, используя метод mean массива Numpy. Результат запишите в массив mean_a, в нем должно быть 2 элемента.

Задание 2

Вычислите массив a_centered, отняв от значений массива “a” средние значения соответствующих признаков, содержащиеся в массиве mean_a. Вычисление должно производиться в одно действие. Получившийся массив должен иметь размер 5x2.

Задание 3

Найдите скалярное произведение столбцов массива a_centered. В результате должна получиться величина a_centered_sp. Затем поделите a_centered_sp на N-1, где N - число наблюдений.

Задание 4**

Число, которое мы получили в конце задания 3 является ковариацией двух признаков, содержащихся в массиве “a”. В задании 4 мы делили сумму произведений центрированных признаков на N-1, а не на N, поэтому полученная нами величина является несмещенной оценкой ковариации.

Подробнее узнать о ковариации можно здесь:

https://studopedia.ru/9_153900_viborochnaya-kovariatsiya-i-viborochnaya-dispersiya.html

В этом задании проверьте получившееся число, вычислив ковариацию еще одним способом - с помощью функции np.cov. В качестве аргумента m функция np.cov должна принимать транспонированный массив “a”. В получившейся ковариационной матрице (массив Numpy размером 2x2) искомое значение ковариации будет равно элементу в строке с индексом 0 и столбце с индексом 1.

Тема “Работа с данными в Pandas”

Задание 1

Импортируйте библиотеку Pandas и дайте ей псевдоним pd. Создайте датафрейм authors со столбцами author_id и author_name, в которых соответственно содержатся данные: [1, 2, 3] и ['Тургенев', 'Чехов', 'Островский'].

Затем создайте датафрейм book со столбцами author_id, book_title и price, в которых соответственно содержатся данные:

[1, 1, 1, 2, 2, 3, 3],

['Отцы и дети', 'Рудин', 'Дворянское гнездо', 'Толстый и тонкий', 'Дама с собачкой', 'Гроза', 'Таланты и поклонники'],

[450, 300, 350, 500, 450, 370, 290].

Задание 2

Получите датафрейм authors_price, соединив датафреймы authors и books по полю author_id.

Задание 3

Создайте датафрейм top5, в котором содержатся строки из authors_price с пятью самыми дорогими книгами.

Задание 4

Создайте датафрейм authors_stat на основе информации из authors_price. В датафрейме authors_stat должны быть четыре столбца:

author_name, min_price, max_price и mean_price,

в которых должны содержаться соответственно имя автора,минимальная, максимальная и средняя цена на книги этого автора.

Задание 5**

Создайте новый столбец в датафрейме authors_price под названием cover, в нем будут располагаться данные о том, какая обложка у данной книги - твердая или мягкая. В этот столбец поместите данные из следующего списка:

['твердая', 'мягкая', 'мягкая', 'твердая', 'твердая', 'мягкая', 'мягкая'].

Просмотрите документацию по функции pd.pivot_table с помощью вопросительного знака.Для каждого автора посчитайте суммарную стоимость книг в твердой и мягкой обложке. Используйте для этого функцию pd.pivot_table. При этом столбцы должны называться "твердая" и "мягкая", а индексами должны быть фамилии авторов. Пропущенные значения стоимостей заполните нулями, при необходимости загрузите библиотеку Numpy.

Назовите полученный датасет book_info и сохраните его в формат pickle под названием "book_info.pkl". Затем загрузите из этого файла датафрейм и назовите его book_info2. Удостоверьтесь, что датафреймы book_info и book_info2 идентичны.