

Тест начат	Четверг, 21 декабря 2023, 16:04
Состояние	Завершены
Завершен	Четверг, 21 декабря 2023, 16:42
Прошло времени	37 мин. 56 сек.
Баллы	6,00/6,00
Оценка	20,00 из 20,00 (100%)

Вопрос 1

Верно

Баллов: 1,00 из 1,00

Прогнозирование с помощью линейной регрессии

В этом задании вам требуется обучить алгоритм линейной регрессии с помощью библиотеки **sklearn**.

Алгоритм линейной регрессии позволяет предсказывать значения зависимой переменной (y) на основе независимых переменных (x_0, x_1, x_2). Модель линейной регрессии имеет вид:

$$y = a * x_0 + b * x_1 + c * x_2 + d,$$

где a, b, c, d — коэффициенты линейной регрессии, которые нужно получить в результате обучения модели.

После обучения модели необходимо спрогнозировать значение зависимой переменной на основе новых входных данных.

Результат сохраните в **numpy**-массив (**ndarray**) в формате:

```
answer: ndarray([a,b,c,y_pred])
```

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 import numpy as np
2 from sklearn.linear_model import LinearRegression
3 #Данные для обучения
4 X = np.array([[1, 1, 1], [1, 2, 2], [2, 2, 3], [2, 3, 5]])
5
6 #Значения функции
7 y = np.array([9, 14, 18, 26])
8
9 #Значение переменных для прогноза функции y_pred
10 X_pred = np.array([[3, 5, 7]])
11
12 #Далее запишите ваш код
13 model = LinearRegression()
14 model.fit(X, y)
15 y_pred = model.predict(X_pred)
16 a, b, c = model.coef_
17 d = model.intercept_
18 answer = np.array([a, b, c, y_pred[0]])
```

	Тест	Ожидается	Получил	
✓	print(np.power(sum(answer - np.array([1., 2., 3., 37.])),2)<1e-10)	True	True	✓

Все тесты пройдены! ✓

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **2**

Верно

Баллов: 1,00 из 1,00

Прогнозирование с помощью логистической регрессии

Необходимо обучить модель классификации с использованием логистической регрессии. Для этого вам предоставлены данные **X_train** и **y_train**, где **X_train** — набор признаков, а **y_train** — целевая переменная.

После обучения модели спрогнозируйте значения целевой переменной для тестовых данных **x_test**. Полученные предсказанные значения сохраните в переменную **y_pred** в виде массива.

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 #Здесь будет выполнена генерация данных для модели
2 #Данный код будет автоматически выполнен при запуске вашей программы
3
4 from sklearn.linear_model import LogisticRegression
5 x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
6
7 #Далее запишите ваш код
8 model = LogisticRegression()
9 model.fit(x_train, y_train)
10 y_pred = model.predict(x_test)
11
```

	Тест	Ожидается	Получил	
✓	from sklearn.metrics import f1_score print(f1_score(y_test,y_pred)>0.9)	True	True	✓

Все тесты пройдены! ✓

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **3**

Верно

Баллов: 2,00 из 2,00

Определение оптимального числа кластеров методом K-means

Предлагаем выполнить это задание самостоятельно в собственном созданном Python-ноутбуке. В LMS требуется ввести лишь результаты выполнения задания.

Напишите программу, которая определит оптимальное количество кластеров в датасете [Electric_Car.csv](#) методом **K-means**.

Для этого необходимо выполнить следующие шаги:

- Загрузить датасет и оставить в Dataframe только столбцы "AccelSec", "TopSpeed_KmH", "Range_Km", "PriceEuro", удалив остальные.
- Выбрать подходящее количество кластеров для разделения данных. Для этого можно использовать метод локтя. Он заключается в запуске алгоритма **K-means** с разным количеством кластеров и в выборе такого числа кластеров, при котором кривая суммы квадратов внутрикластерных расстояний до центра кластера имеет характерный изгиб.
- Реализовать алгоритм **K-means** с выбранным количеством кластеров. Для параметра random_state укажите значение 0.
- Провести кластеризацию данных. Вычислить координаты центров кластеров с помощью атрибута **cluster_centers_** и норму полученного вектора с помощью функции **linalg.norm** из библиотеки **NumPy** с точностью до третьего знака.

Вопрос 1. Запишите в поле ввода ответа рассчитанное количество кластеров, полученное при решении задачи (ответ запишите в виде числа, например, 123).

4

✓

Вопрос 2. Вычислите координаты центров кластеров и норму полученного вектора. В поле ввода запишите результат рассчитанной нормы, округлив значение до трех знаков после запятой. В качестве разделителя дробной и целой части используйте символ точки "." (например, 1.234).

209833.221

✓

Вопрос **4**

Верно

Баллов: 1,00 из 1,00

Определение матрицы рейтингов с использованием SVD

Задание состоит в заполнении нулевых значений в матрице **rating** с использованием метода сингулярного разложения (**SVD**).

Исходная матрица **rating** представляет собой двумерный массив, содержащий оценки пользователей для определенных объектов. Однако в некоторых ячейках матрицы отсутствуют значения, обозначенные нулями.

Для заполнения этих нулевых значений используйте метод **SVD** — метод разложения матрицы на три составляющих: матрицу левых сингулярных векторов (**U**), матрицу сингулярных значений (**S**) и матрицу правых сингулярных векторов (**V**).

В результате выполнения программы будет получена матрица **rating_prediction**, в которой нулевые значения будут заменены предсказанными значениями с использованием **SVD**.

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 import numpy as np
2 from scipy.sparse.linalg import svds
3
4
5 ratings = np.array([[3, 1, 0, 1],
6                    [0, 3, 0, 1],
7                    [1, 2, 0, 5],
8                    [1, 0, 0, 2],
9                    [0, 1, 3, 1]]).astype(float)
10
11 #Далее запишите ваш код
12 U, s, Vt = svds(ratings, k=2) # k - количество сингулярных значений
13 # Переформирование s в диагональную матрицу
14 S = np.diag(s)
15 # Предсказание полной матрицы рейтингов
16 rating_prediction = np.dot(np.dot(U, S), Vt)
17
```

	Тест	Ожидается	Получил	
✔	rating_approve = np.array([[1.3183674 , 0.6341322 , -0.76682067, 1.95292953], [0.25769658, 1.34955709, 0.94417223, 1.73267927], [1.84666953, 2.40446815, 0.27938119, 4.4068027], [0.95743862, 0.68557226, -0.35599688, 1.66633836], [-0.79235923, 1.62553782, 2.25215506, 1.0381396]]) print(np.power(sum(sum(rating_prediction - rating_approve)),2) <1e-10)	True	True	✔

Все тесты пройдены! ✔

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **5**

Верно

Баллов: 1,00 из 1,00

Применение фильтра к изображению

Напишите программу, которая создает фильтр для изображения. Входные данные — изображение в виде матрицы **image**, где каждый элемент представляет собой пиксель изображения.

Ваша задача — применить фильтр **filter** к изображению и выдать результат в виде новой матрицы **feature_map**.

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 import numpy as np
2
3 image = np.array([[1, 2, 1, 1, 3],
4                  [0, 1, 2, 3, 4],
5                  [0, 0, 1, 2, 3],
6                  [3, 2, 1, 5, 0],
7                  [1, 2, 3, 1, 7]])
8
9 filter = np.array([[1, 0, 1],
10                  [0, 1, 0],
11                  [1, 0, 1]])
12
13 #Далее запишите ваш код
14 image_height, image_width = image.shape
15 filter_height, filter_width = filter.shape
16
17 # Размеры результирующей матрицы
18 output_height = image_height - filter_height + 1
19 output_width = image_width - filter_width + 1
20
21 # Инициализация матрицы feature_map
22 feature_map = np.zeros((output_height, output_width))
23
24 # Применение свертки
25 for i in range(output_height):
26     for j in range(output_width):
27         feature_map[i, j] = np.sum(image[i:i+filter_height, j:j+filter_width] * filter)
28
29
```

	Тест	Ожидается	Получил	
✓	<pre>feature_map_prove = np.array([[4, 7, 11], [6, 12, 9], [7, 6, 19]]) print(np.power(sum(sum(feature_map_prove - feature_map)),2) <1e-10)</pre>	True	True	✓

Все тесты пройдены! ✓

Верно
Баллы за эту попытку: 1,00/1,00.