Вопрос № 1. Рассмотрите основные особенности обучения моделей машинного обучения на размеченных данных.

Вопрос № 2. Проанализируйте виды признаков, используемых в машинном обучении.

Вопрос № 3. Рассмотрите возможности использования линейных моделей в задачах регрессии.

Вопрос № 4. Рассмотрите возможности использования линейных моделей в задачах классификации.

Вопрос № 5. Задача регрессии - предсказание значений непрерывной целевой переменной

Вопрос № 6. Стохастический градиентный спуск

Вопрос № 7. K ближайших соседей

Вопрос № 8. Простая линейная регрессия

Вопрос № 9. Логистическая регрессия.

Вопрос № 10. Деревья принятия решений.

Вопрос № 11. Нейронные сети прямого распростанения

Вопрос № 12. Нейронные сети обратного распространения.

**Вопрос №1.**

1.1 Задайте функцию, которая не принимает никаких аргументов и просто выводит на экран строку: It's my first function

В конце программы вызовите эту функцию.

*def my\_function():*

*print("It's my first function")*

*# вызов функции*

*my\_function()*

1.2 Имеется список с оценками студента:

m = [2, 3, 5, 5, 2, 2, 3, 3, 4, 5, 4, 4]

Необходимо с помощью срезов выбрать элементы с 3-го по 7-й (включительно) и вывести их на экран в обратном порядке.

*m = [2, 3, 5, 5, 2, 2, 3, 3, 4, 5, 4, 4]*

*# выбираем элементы с 3-го по 7-й (включительно) с помощью среза*

*selected\_elements = m[2:7]*

*# выводим выбранные элементы на экран в обратном порядке*

*print(selected\_elements[::-1])*

**Вопрос №2.**

2.1 Вводятся целые числа в одну строчку через пробел. Необходимо преобразовать их в список lst , затем, удалить последнее значение и если оно нечетное, то в список (в конец) добавить True, иначе - False. Отобразить полученный список на экране командой: print(\*lst)

Реализовать программу без использования условного оператора.

*# вводим числа через пробел и преобразуем их в список*

*lst = list(map(int, input().split()))*

*# определяем четность последнего значения*

*last\_element\_is\_odd = lst[-1] % 2 == 1*

*# удаляем последнее значение из списка*

*lst.pop()*

*# добавляем в список значение, соответствующее четности последнего элемента*

*lst.append(last\_element\_is\_odd)*

*# выводим список на экран*

*print(\*lst)*

2.2 На вход программы поступают данные в виде набора строк в формате:

ключ1=значение1  
ключ2=значение2  
...  
ключN=значениеN

Ключами здесь выступают целые числа (см. пример ниже). Необходимо их преобразовать в словарь d (без использования функции dict()) и вывести его на экран командой:

print(\*sorted(d.items()))

P. S. Для считывания списка целиком в программе уже записаны начальные строчки.

*# начальные строки программы*

*input\_data = ['1=apple', '2=banana', '3=cherry']*

*# создаем пустой словарь*

*d = {}*

*# проходим по каждой строке в списке input\_data*

*for line in input\_data:*

*# разделяем строку на ключ и значение*

*key, value = line.split('=')*

*# преобразуем ключ в целое число*

*key = int(key)*

*# добавляем ключ и значение в словарь*

*d[key] = value*

*# выводим отсортированный словарь на экран*

*print(\*sorted(d.items()))*

**Вопрос №3.**

1. Вводятся оценки студента (числа от 2 до 5) в одну строку через пробел. Необходимо определить количество двоек и вывести это значение на экран.
2. Вводятся данные в формате ключ=значение в одну строчку через пробел. Необходимо на их основе создать словарь, затем проверить, существуют ли в нем ключи со значениями: 'house', 'True' и '5' (все ключи - строки). Если все они существуют, то вывести на экран ДА, иначе - НЕТ.

**Вопрос №4.**

1. Вводятся данные в формате ключ=значение в одну строчку через пробел. Необходимо на их основе создать словарь d, затем удалить из этого словаря ключи 'False' и '3', если они существуют. Ключами и значениями словаря являются строки. Вывести полученный словарь на экран командой: print(\*sorted(d.items()))
2. Вводятся номера телефонов в одну строчку через пробел с разными кодами стран: +7, +6, +2, +4 и т.д. Необходимо составить словарь d, где ключи - это коды +7, +6, +2 и т.п., а значения - список номеров (следующих в том же порядке, что и во входной строке) с соответствующими кодами. Полученный словарь вывести командой: print(\*sorted(d.items()))

**Вопрос №5.**

1. Пользователь вводит в цикле целые положительные числа, пока не введет число 0. Для каждого числа вычисляется квадратный корень (с точностью до сотых) и значение выводится на экран (в столбик). С помощью словаря выполните кэширование данных так, чтобы при повторном вводе того же самого числа результат не вычислялся, а бралось ранее вычисленное значение из словаря. При этом на экране должно выводиться: значение из кэша: <число>
2. Имеется закодированная строка с помощью азбуки Морзе. Коды разделены между собой пробелом. Необходимо ее раскодировать, используя азбуку Морзе из предыдущего занятия. Полученное сообщение (строку) вывести на экран.

**Вопрос №6.**

1.В список: a = [5.4, 6.7, 10.4]

добавить в конец вложенный список со значениями, вводимыми в программу (целые числа вводятся в строчку через пробел). Результирующий список lst вывести на экран командой: print(lst)

2. Имеется кортеж: t = (3.4, -56.7)

Вводится последовательность целых чисел в одну строчку через пробел. Необходимо их добавить в кортеж t. Результат вывести на экран командой: print(t)

**Вопрос №7.**

1. Вводятся целые числа в одну строку через пробел. На их основе формируется кортеж. Необходимо найти и вывести все индексы неуникальных (повторяющихся) значений в этом кортеже. Результат отобразите в виде строки чисел, записанных через пробел.
2. Вводятся три строчки стихотворения (каждая с новой строки). Сохранить его в виде вложенного списка с разбивкой по строкам и словам (слова разделяются пробелом). Результирующий список lst вывести на экран командой: print(lst)

**Вопрос №8.**

1. Вводится  матрица чисел из трех строк. В каждой строке числа разделяются пробелом. Необходимо вывести на экран последний столбец этой матрицы в виде строки из трех чисел через пробел.
2. Имеется двумерный кортеж, размером 5 x 5 элементов:

t = ((1, 0, 0, 0, 0),

 (0, 1, 0, 0, 0),

 (0, 0, 1, 0, 0),

 (0, 0, 0, 1, 0),

 (0, 0, 0, 0, 1))

Вводится натуральное число N (N < 5). Необходимо на основе кортежа t сформировать новый аналогичный кортеж t2 размером N x N элементов. Результат вывести на экран в виде таблицы чисел.

**Вопрос №9.**

1. Вводятся два вещественных числа в одну строку через пробел. Вывести на экран наибольшее из чисел. Задачу решить с помощью условного оператора.
2. Вводится текст в одну строку, слова разделены пробелом. Необходимо подсчитать число уникальных слов (без учета регистра) в этом тексте. Результат (число уникальных слов) вывести на экран.

**Вопрос №10.**

1. Вводится слово. Необходимо определить, является ли это слово палиндромом (одинаково читается вперед и назад, например, АННА). Регистр букв не учитывать. Если введенное слово палиндром, на экран вывести ДА, иначе - НЕТ.
2. Вводятся два списка целых чисел каждый с новой строки (в строке наборы чисел через пробел). Необходимо выбрать и отобразить на экране уникальные числа, присутствующие в первом списке, но отсутствующие во втором. Результат выведите на экран в виде строки чисел, записанных по возрастанию через пробел.

**Вопрос №11.**

1. Вводятся два целых положительных числа m и n в одну строку через пробел. Если число m делится нацело на число n, то вывести на экран частное от деления (результат деления) в виде целого числа. В противном случае вывести сообщение «m на n нацело не делится» (без кавычек) и вместо m и n подставить соответствующие числа, например: «13 на 2 нацело не делится».
2. Вводятся два списка целых чисел каждый с новой строки (в строке наборы чисел через пробел). Необходимо выбрать и отобразить на экране уникальные числа, присутствующие в первом или втором списках, но отсутствующие одновременно в обоих. Результат выведите на экран в виде строки чисел, записанных по возрастанию через пробел.

**Вопрос №12.**

1. Вводится список оценок студента - его ответов у доски по предмету "Информатика" в виде чисел от 2 до 5 в одну строку через пробел. Если студент имеет хотя бы одну двойку, то он не допускается до экзамена. Определить на основе введенного списка, допущен ли студент. Если допущен, то вывести слово ДОПУЩЕН, иначе - НЕ ДОПУЩЕН. При реализации задачи используйте множество для определения наличия двойки.
2. Вводится натуральное число, которое может содержать только простые множители 1, 2, 3, 5 и 7 (любые из них, не обязательно все). Необходимо разложить введенное число на простые множители и проверить, содержит ли оно множители 2, 3 и 5 (обязательно все их, хотя бы один раз). Если это так, то вывести ДА, иначе - НЕТ.