Ответы на задачи

1. В данном SQL запросе допущена ошибка, исправьте её и объясните, почему она возникает.

```sql
SELECT id, WEEK(created_at) AS register_week
FROM customers
WHERE register_week > 30
```

Ответ:

В указанном виде будет выдаваться ошибка, что «register_week» не существует, она появляется из-за того, что в sql сначала обрабатывается FROM, потом WHERE и далее SELECT

правильный вариант:

```sql
SELECT  id, WEEK(created_at) AS register_week
FROM customers
WHERE WEEK(orderDate) > 30
```

2. Есть 2 таблицы: Фильмы и Оценки. Напишите SQL запрос, который получит среднюю оценку, медиану оценок и последнюю оценку каждого фильма.

Ответ:

```sql
WITH med AS (
SELECT movie_id,
AVG(rating) AS median_val
FROM (
SELECT
movie_id,
rating,
row_number() OVER (partition by movie_id order by rating) rn,
count(*) over(partition by movie_id) cnt
from Raiting) as dd
WHERE rn in (FLOOR((cnt + 1) / 2), FLOOR( (cnt + 2) / 2) )
GROUP BY movie_id),

last_record AS (
SELECT id, movie_id, rating AS Last_Raiting
FROM Raiting
WHERE id IN (
SELECT MAX(id)
FROM Raiting
GROUP BY movie_id))

SELECT Movie.name,
AVG(Raiting.rating) AS AVERAGE,
med.median_val AS MEDIAN, last_record.Last_Raiting
FROM Movie, Raiting, med, last_record
WHERE Movie.id = Raiting.movie_id AND Movie.id =
med.movie_id AND Movie.id =last_record.movie_id
GROUP BY Movie.name
```

Movie

id	name
1	A
2	B
3	C
4	D
5	E

Raiting

id	movie_id	rating
1	1	3
2	2	3
3	1	5
4	1	4
5	2	4
6	3	3
7	4	5
8	4	4
9	5	3
10	2	5

3. Вероятность увидеть хотя бы 1 падающую звезду в течение каждых 15-ти минут равна 20%. Какова вероятность увидеть хотя бы 1 падающую звезду в течение 1 часа. Объясните решение.

   Ответ:

   p = 0.2 - вероятность увидеть звезду в течение 15 минут
   Используем формулу полной вероятности, это сумма следующих вероятностей:
   p - вероятность увидеть в первые 15 минут
   +
   p*(1-p) - вероятность увидеть во вторые 15 минут, не видя в первые
   +
   p*(1-p)^2 - вероятность увидеть в третье 15 минут, не видя в первые и вторые
   +
   p*(1-p)^3 - вероятность увидеть в четвертые 15 минут, не видя в первые, вторые и третьи.

   Итого получаем:

   0.2 + 0.2*0.8 + 0.2*0.8^2 + 0.2*0.8^3 = 0,5904 - искомая вероятность

4. Как сгенерировать 1 целое случайное число от 1-го до 7 [1, 7] с помощью 1-го игрального кубика.

   Ответ:

   Бросаем кубик два раза. В таком случае у нас будет максимум 36 возможных вариантов, для того чтобы получит равновероятный результат, необходимо уменьшить общее число исходов до кратного 7, ближайшее подходящее число 35, следовательно мы «запретим» один исход из 36. В результате получаем 7x5 комбинаций для получения числа от 1 до 7 и одну комбинацию для переброса.
   Общая схема такая:
   1) Если второй бросок не равен 6, используем для определения числа первый бросок.
   (1:1–1:5) → 1
   (2:1–2:5) → 2 и т.д
   2) Если второй бросок равен 6 и первый не равен 6, то получаем число 7
   (16,26,36,46,56) → 7
   3) Если первый и второй броски равны 6, тогда перебрасываем
   В итоге получаем способ равновероятно сгенерировать число от 1 до 7 используя один кубик.

5. Задача на Python.
   Дана строка, напишите функцию, которая вернет длину самой длинной подстроки без повторяющихся символов.

   Примеры:
   “dbcdbcbb” ответ: **3** (“dbc” длиной 3)
   “cccccc” ответ: **1** (“c” длиной 1)
   “**bwwkew**” ответ: **3** (“wke” длиной 3, обратите внимание, что ответ должен быть подстрокой, “bwke” это часть последовательности символов, но это не подстрока)

Ответ:

```
def longest(string):
    result = 0
    for start in range(len(string)):
        container = set()
        for letter in string[start:]:
            if letter in container:
                break
            container.add(letter)
        result = max(result, len(container))
    return result

x = input()
print('Длина самой длинный подстроки без повторяющихся символов', longest(x))
```

6. Задача на Python.
Даны 2 отсортированных списка целых чисел. Напишите функцию, которая, не подключая дополнительных библиотек, вернет медиану для этих двух списков. Заранее известно, что списки не могут быть пустыми.

Примеры:
a = [2, 4]
b = [3]
ответ: 3.0

a = [2, 3]
b = [4, 5]
ответ: (3 + 4) / 2 = 3.5

Ответ:

```
def mediana (list1,list2):
    list3 = sorted(list1 + list2)
    length = len(list3)
    if length % 2 == 0:
        mediana = (list3[length//2-1]+list3[length//2])/2
    else:
        mediana = list3[length//2+1]
    return mediana
```

7. У вас есть набор данных со следующей структурой и длиной 15 млн строк:

DateTime	Customer Id	RestaurantId	City	Revenue
2020-06-15 20:00:00	23	123	Москва	250.0

Перед вами стоит задача, прогноз дневной выручки в каждом ресторане в будущем.

Вопросы:

- Напишите общий план ваших действий;
- Какой способ валидации при настройке гиперпараметров, на ваш взгляд наиболее предпочтителен в такой задаче?

Ответ:

1) Искомая величина Revenue (выручка) - непрерывная переменная, решаем задачу регрессии :
Общий план:
- Проанализировать данные: проверить наличие пропусков, какое распределение, есть ли выбросы, визуализировать данные и т.д.
- В зависимости от результатов анализа обрабатываем данные (переводим все текстовые все значения в числовой формат (label encoding или one-hot encoding), пропуски - отдельные категория, либо удаляем, нормализуем данные и т.д.)
-Анализируем полученное признаковое пространство.
- Подбираем оптимальную модель (линейная или полиномиальная регрессия, дерево решений, рандомный лес, градиентный бустинг и т.д) и параметры.
- Тестируем наилучшую модель.
2) При настройке гиперпараметров используем кросс-валидацию.

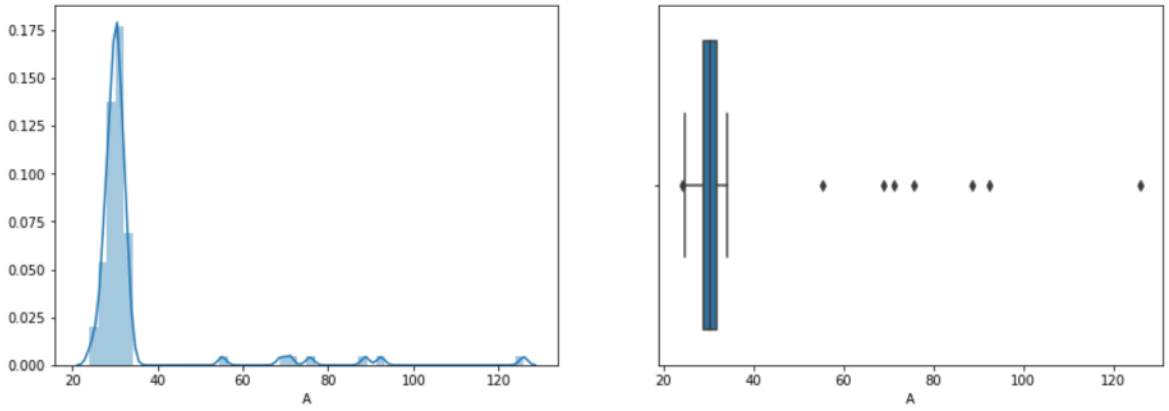
8. В файле для_теста.xlsx есть 2 распределения A и B.
    - Опишите данные распределения.
    - Допустим у нас есть задачи, в которых данные распределения являются целевой переменной, которую нам нужно прогнозировать. Какой тип задачи машинного обучения стоит перед нами в каждом случае.
    - Какую метрику качества в каждом случае вы бы предпочли?

Ответ:
Ответ:
Столбец A - непрерывная переменная, вещественные числа (float), каждое значение уникально, среднее распределения - 33.63, медиана - 30.34, среднеквадратичное отклонение - 14.740618, мин значение - 24.010432, максимальное - 125.828221, размах - 101,817789, распределение отличается от нормального, имеются сильные выбросы в правом хвосте.

Метрики оценки -
Среднеквадратичная ошибка (MSE):
-легко минимизировать
-сильно штрафует за большие ошибки
Средняя абсолютная ошибка (MAE):
-выше устойчивость к выбросам
-сложнее минимизировать.

Если больше ошибки критичны - выбираем MSE, если важнее корректно учитывать выбросы - MAE. При выборе MSE рекомендуется дополнительно оценивать коэффициент детерминации (R^2) - показывает, какую долю дисперсии (разнообразия ответов) модель смогла предсказать:
0<=R^2<=1 - разумная модель
R^2 = 1 - идеальная модель
R^2 = 0 - модель на уровне константной
R^2 < 0 - модель хуже константной.
Задача машинного обучения - регрессия.

График распределения и диаграмма размаха (box plot), столбец A



Столбец B - категориальная переменная, целые числа (int), всего 7 уникальных значений (7 категорий), наиболее часто встречающееся значения - 1,2,3; наиболее редкое - 6.

Метрика оценки:
Accuracy(доля правильных ответов), Precision (точность), Recall (полнота), F1-score (объединяет precision и recall), ROC_AUC-score (площадь (*A*rea *U*nder *C*urve) под кривой ошибок (*R*eceiver *O*perating *C*haracteristic curve).
ROC_AUC-score показывает качество работы алгоритма, идеальный классификатор который не ошибается ROC_AUC-score =1, худший - 0.5. В целом необходимо найти оптимальный для конкретной задачи баланс между precision и recall, в случае несбалансированных классов выбираем модель с максимальным F1-score.

Задача машинного обучения - классификация.

График распределения и диаграмма размаха (box plot), столбец B