

Задача

Ваш сотовый оператор предлагает следующую акцию: Вы можете купить любое число SMS на месяц по льготной цене 0.8 р/штука. Если Вам потребуется больше SMS, то вы будете платить за них по обычной цене 1 р/штука. Если Вы не используете все купленные SMS за месяц, то заплаченные за них деньги пропадут.

За последние 7 месяцев число отправленных Вами SMS составило: 102, 83, 93, 86, 88, 81, 95. Сколько SMS Вам следует купить по льготной цене?

Рекомендация по решению

1. Реализуйте модель для восстановления параметров распределения числа отправляемых SMS, например:
$$\text{Число SMS}_i \sim \text{Poisson}(c),$$
$$c \sim \text{Uniform}(50, 150).$$
2. Убедитесь, что модель работает, посмотрите на плотность распределения параметра c (ожидаемого значения числа SMS в месяц).
3. Математически сформулируйте функцию потерь для данной задачи. Функция должна иметь два параметра (два входа): настоящее число SMS отправленных в месяц и прогноз (число закупленных SMS). Просто посчитайте, сколько денег переплачивается: если купить больше SMS чем нужно, то теряются деньги за эти SMS; если купить меньше, чем нужно, то теряется $1 - 0.8 = 0.2$ рубля с каждой «лишней» SMS.
4. Реализуйте функцию `sms_loss(true_count, prediction)`, которая бы вычисляла построенную Вами функцию потерь. Эта функция должна быть написана таким образом, чтобы в качестве аргумента `true_count` (истинное значение числа отправленных SMS) можно было подать массив значений, а в качестве параметра `prediction` (число спрогнозированных и закупленных заранее SMS) подавалось одно значение. Функция должна вернуть массив значений, соответствующих потерям для каждого истинного значения из `true_count`.
5. Реализуйте функцию `expected_loss(prediction)`, которая вычисляет ожидаемое значение потерь по заданному прогнозу: передайте функции `sms_loss()` след параметра c и значение прогноза `prediction`. Верните среднее значение полученного параметра.
6. Постройте график функции `expected_loss()`:
 - a. сгенерируйте массив возможных значений прогноза, используя `numpy.arange(min, max)` или `numpy.linspace(min, max, steps)`;
 - b. для каждого значения прогноза вычислите значение функции `expected_loss()` и поместите результаты в ещё один массив;
 - c. постройте график получившейся зависимости.
7. Используя функцию `scipy.optimize.fmin()` вычислите минимум функции `expected_loss()`. В качестве параметров функции `fmin` надо передать функцию `expected_loss` и начальное приближение к ответу (его можно придумать, посмотрев на график). Напечатайте полученное значение.
8. Постройте гистограмму следа параметра c и обозначьте на нём вертикальной линией значение минимума функции ошибки (Байесовское действие). Вертикальную линию можно нарисовать функцией `plt.vlines(x, min_y, max_y)`.