Задача:

Построить модель машинного обучения, предсказывающую траты у посетителей магазина.

План решения:

1. Загрузите **данные** и подготовьте их:
   1. Сделайте из текстового признака, обозначающего пол, бинарный флаг 0/1
   2. Переименуйте признаки так, чтобы в названиях не было пробелов.
2. Доработайте функцию построения сплита (разработанную в рамках скринкаста):
   1. Занесите расчет хаотичности целевого признака в выборке в саму функцию построения сплита
   2. Уберите промежуточные выводы
   3. В качестве результата отработки функции возвращайте пару “признак”+”значение для сплита”, при которых достигается наилучший прирост информации. Если таких не нашлось (если выборка однородна с точки зрения всех признаков и ее невозможно разделить), возвращайте, например, None.
   4. Сделаем функцию похожей на библиотечную: пусть она принимает на вход отдельно датафрейм с признаками (назовите параметр X ) и столбец со значениями целевого признака (назовите y). Доработайте код функции под новые входные данные. Подсказка: чтобы понять, какие значения целевого признака будут принимать объекты, попавшие в левую/правую часть подвыборки, создайте маску из столбца с фичей и примените ее к столбца с целевым признаком:
      1. Чтобы получить маску, к столбцу в признаком, по которому происходит деление примените операцию сравнения со значением, по которому хотите выполнить сплит. Получится столбец из True/False длины, равной количеству элементов в исходной выборке, подлежащей сплиту.
      2. Примените полученную маску к столбцу с целевыми значениями target[mask]. В итоге останутся только значения целевой переменной для наблюдений, которым в маске соответствовало значение True.
3. Реализуйте функцию построения решающего дерева fit\_regression\_tree, допускающую ограничения на максимальную глубину дерева, а также минимальное число наблюдений в выборке, подлежащей сплиту. В результате запуска функции должны быть выведены правила, по которым должен происходить сплит, а в листьях - предикт (то есть среднее значение целевого признака на подвыборке обучающего датафйрема, попавшей в этот лист). Бинарное дерево строится рекурсивной, поэтому функция его построения тоже будет рекурсивной:
   1. Реализуйте часть для выхода из рекурсии. Здесь должна выполняться проверка на достижение максимально допустимой глубины дерева/достаточное количество наблюдений в выборке для выполнения очередного сплита/возможность построения сплита. Невозможно построить сплит, если все объекты в выборке одинаковы с точки зрения значений фичей. Если одно из этих условий выполняется, выводите предикт в следующем формате:  
      "| " \* <текущая глубина дерева>"|---" value: <предикт с 2 знаками после запятой>]'
   2. Реализуйте альтернативный путь для дальнейшего построения дерева:
      1. Выведите правило, по которому наблюдения “идут влево” в формате:  
         "| " \* <текущая глубина дерева>"|---" <фича для сплита> <= <значение для сплита>
      2. Постройте сплит “пошедшей влево” подвыборки
      3. Выведите правило, по которому наблюдения “идут вправо” в формате:  
         "| " \* <текущая глубина дерева>"|---" <фича для сплита> > <значение для сплита>
      4. Постройте сплит “пошедшей вправо” подвыборки.
4. Проверьте корректность реализованного алгоритма
   1. Убедитесь, что без ошибок строится дерево без ограничений (то есть до последнего возможного сплита)
   2. Убедитесь, что без ошибок отрабатывает алгоритм для построения дерева с ограничением на глубину. Выберите максимально допустимую глубину, равную 2. Также постройте дерево с помощью библиотечного алгоритма (DecisionTreeRegressor) на тех же входных данных. Выведите его с помощью функции export\_text и сравните построенные деревья. Они должны совпадать.
   3. Убедитесь, что без ошибок отрабатывает алгоритм для построения дерева с ограничением на минимальное количество элементов в выборке, над которой производится в сплит. Также сравните построенное дерево с деревом, построенным с помощью библиотечной реализации, при ограничении min\_samples\_split=70.