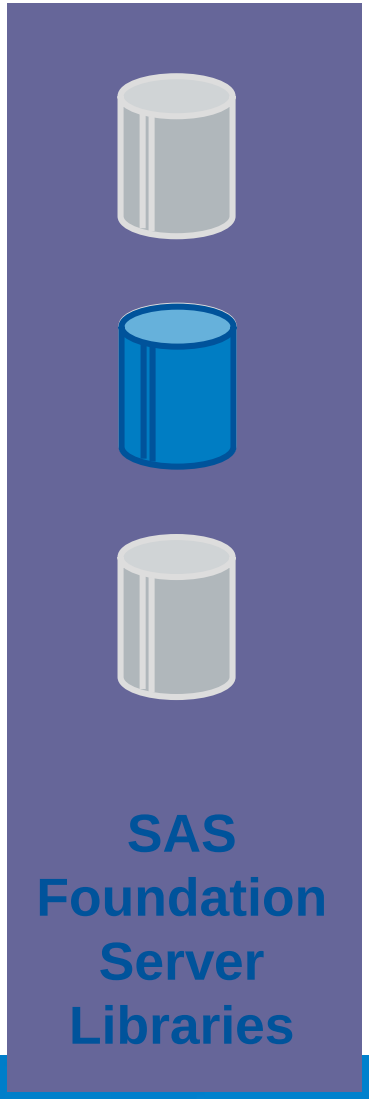
SAS ENTERPRISE MINER

ПРЕДОБРАБОТКА ДАННЫХ

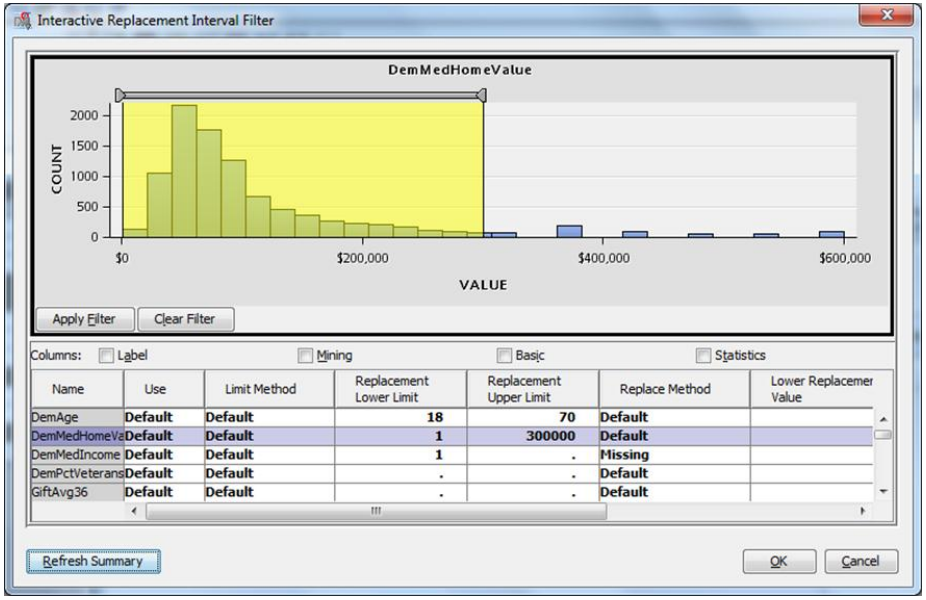
ПОДКЛЮЧЕНИЕ ИСТОЧНИКА ДАННЫХ



- Выбрать источник.
- Определить роли переменных.
- Определеить типы переменных.
- Определить роль источника.

ФИЛЬТРАЦИЯ И ЗАМЕНА ДАННЫХ

- Цель – поиск и удаление из выборки артефактов и выбросов



Правила фильтрации задаются для отдельных переменных:

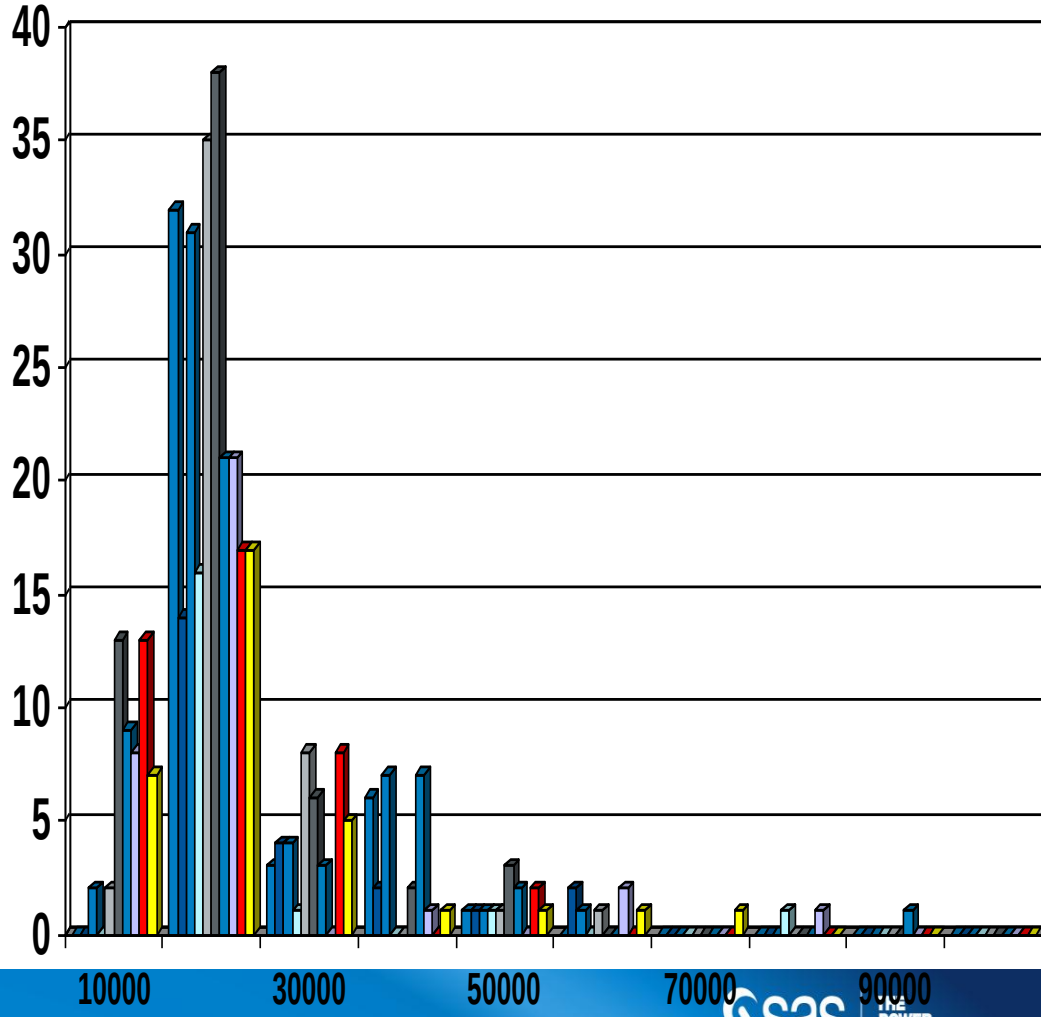
- Ручные – задаются недопустимые значения переменных (диапазоны для числовых, список для категориальных)
- Редкие значения для категориальных
- Нетипичные значения для числовых (задается допустимое отклонение от мат. ожидания или допустимое отклонение от медианы или экстремальные процентили и другое).

СОКРАЩЕНИЕ ОБУЧАЮЩЕЙ ВЫБОРКИ – СЛУЧАЙНАЯ ВЫБОРКА (SAMPLING)

- Цель – выбрать «представительное» подмножество примеров:
  - В идеале с тем же распределением
  - Просто случайная выборка работает плохо – не удается сохранить характеристики всего набора
- Адаптивные методы случайной выборки:
  - В соответствии с «грубой» моделью, например, кластерной
  - Случайная выборка в рамках «срезов», построенных по классу, высоко селективному атрибуту или их комбинации
  - Основная особенность – выборка в рамках среза или кластера пропорциональна размеру среза или кластера

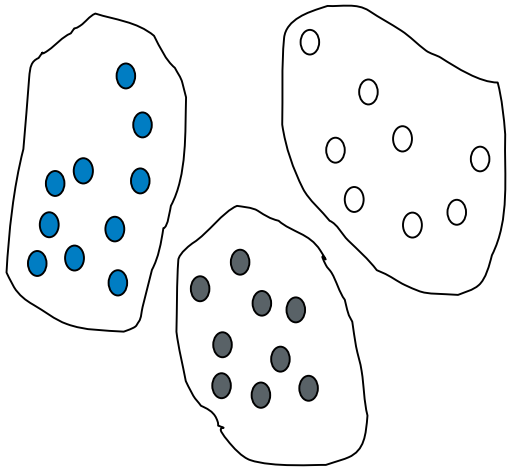
СОКРАЩЕНИЕ ОБУЧАЮЩЕЙ ВЫБОРКИ (SAMPLING) – МЕТОД ГИСТОГРАММ

- Задается процент исходной выборки
- Для выбранной категориальной переменной (переменная стратификации) строится частотная диаграмма (для числовой необходима предварительная дискретизация)
- Наблюдения случайным образом выбрасываются так, чтобы сохранить распределение переменной стратификации

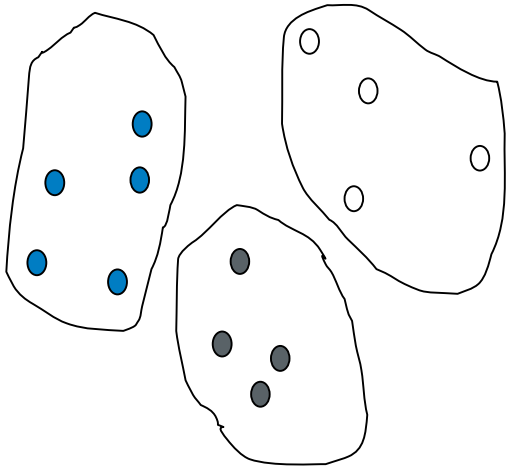


СОКРАЩЕНИЕ ОБУЧАЮЩЕЙ ВЫБОРКИ (SAMPLING) – КЛАСТЕРИЗАЦИЯ

«Сырые» данные



Кластерная/стратифицированная случайная выборка



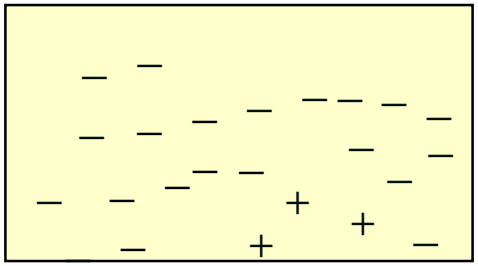
- Кластеризуем данные
- Каждому наблюдению присваивем номер его кластера
- Далее переменная с номером кластера рассматривается как переменная стратификации

«БАЛАНСИРОВКА» КЛАССОВ

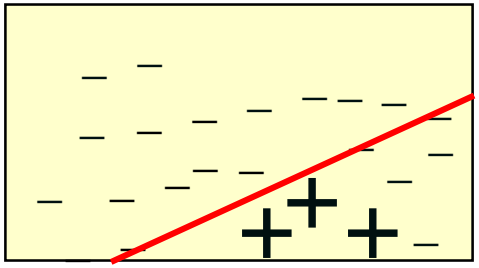
- Обычная ситуация – число примеров одного класса может на порядки отличаться от числа примеров другого
- Если решать напрямую – ничего не получится
- Три варианта:
  - Разный «штраф» за ошибку наиболее популярный метод
  - Under sampling – «искусственно» увеличивать число примеров «маленького» класса – можно испортить распределение и закономерности
  - Oversampling – «искусственно» уменьшить число примеров «большого» класса - можно потерять важную информацию, но тоже популярный метод

ПРИМЕР «БАЛАНСИРОВКИ» КЛАССОВ

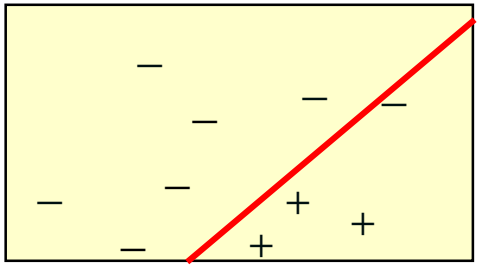
- Пусть “-” в 1000 раз больше чем «+», тогда точность «константного классификатора (всегда «–») 99.9%

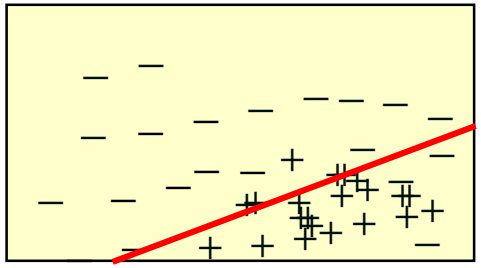


- Если «штраф» на «+» за ошибку увеличить в 1000



- Over sampling и under sampling:





ФОРМИРОВАНИЕ ОБУЧАЮЩИХБ ВАЛИДАЦИОННЫХ И ТЕСТОВЫХ ВЫБОРОК

- Переобучение:
  - нельзя строить и проверять модель на одних и тех же данных
- Обычный подход в DM – случайное разбиение на 3 набора
  - Тренировочный - для построения семейства моделей – кандидатов на финальную модель
  - Валидационный – для выбора из кандидатов финальной модели
  - Тестовый – для оценки качества финальной модели на «новых» данных
  - Иногда валидационный=тестовый
- Замечания:
  - Необходимо сохранить «пропорцию» значений отклика – это просто для задач классификации, сложнее для регрессии, еще сложнее для ранжирования и других
  - Необходимо учитывать специфические атрибуты, например, время, место и другие …

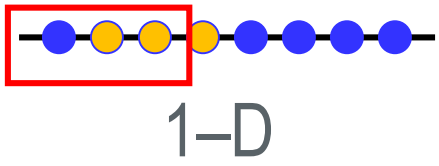
ДРУГИЕ ПОДХОДЫ К ФОРМИРОВАНИЮ ВЫБОРОК

- Cross валидация – перекрестная проверка:
  - Если недостаточно данных, разбиваем на равные блоки с сохранением «пропорции» отклика
  - Строим модели для всех комбинаций
  - Результат усредняем

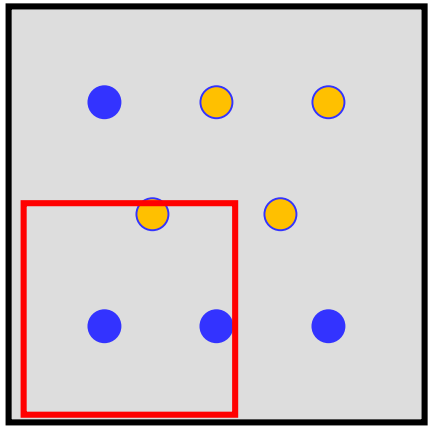
1	2	3	4	5
Train	**Train**	**Valid**	**Test**	**Train**

- Bootstrapping:
  - Из набора размера N формируем с помощью случайной выборки с возвратом M наборов, каждый размера N
  - В каждый из M какие-то элементы не попадают, какие-то входят по несколько раз
  - Строим модели для всех наборов, считаем оценки для всех моделей, но на исходном наборе
  - Результат оценки усредняем

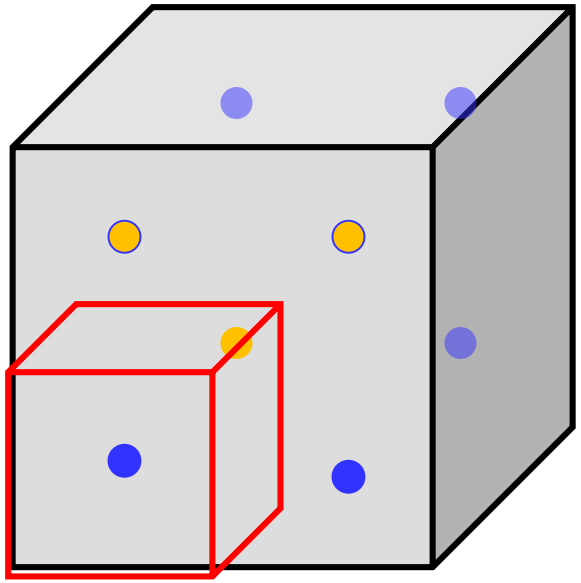
«ПРОКЛЯТИЕ» РАЗМЕРНОСТИ



1–D



2–D

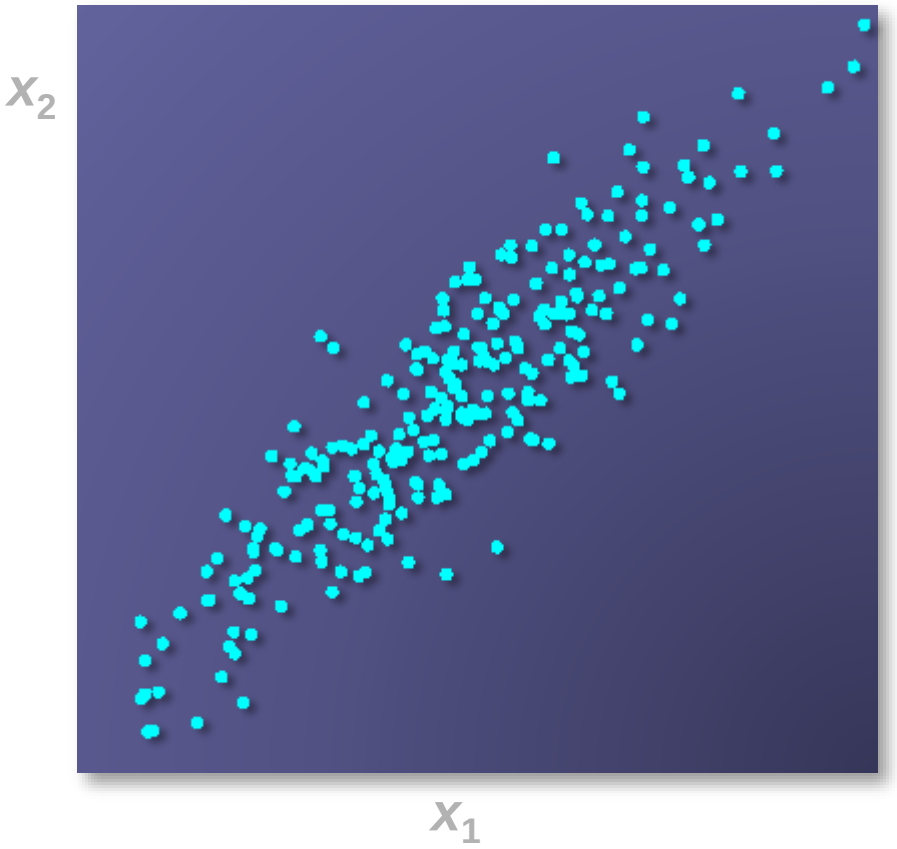


3–D

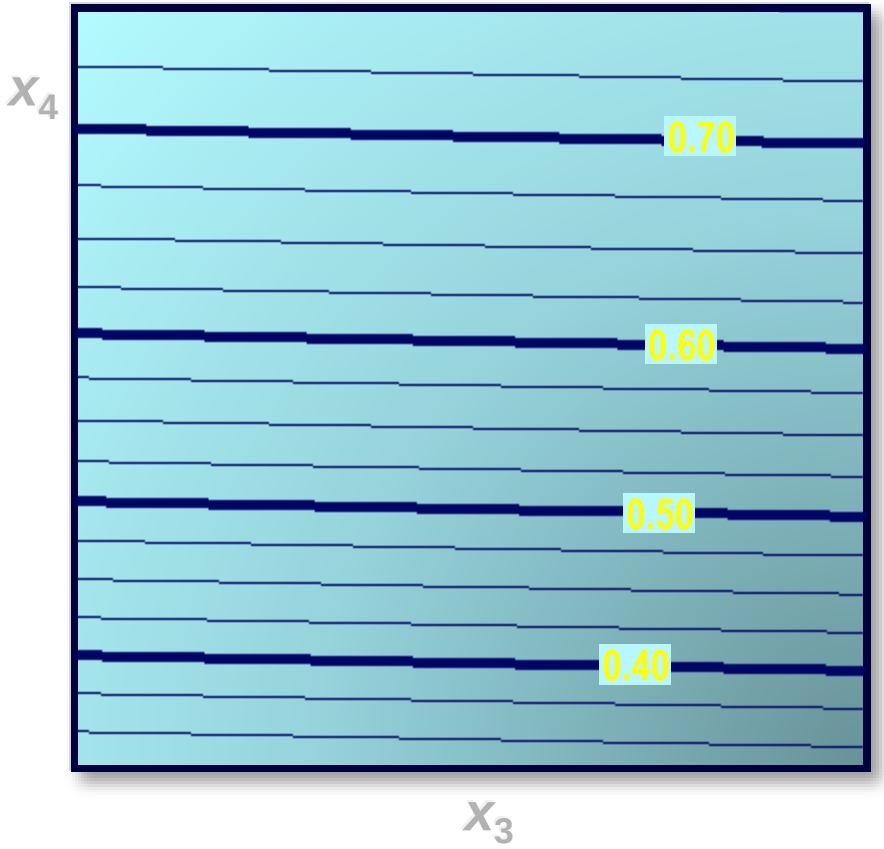
- $E_p(r)=r^{1/p}$
- $E_{10}(0.01)=0.63$
- $E_{10}(0.1)=0.8$

ПРОБЛЕМЫ ВХОДНЫХ ПЕРЕМЕННЫХ

Зависимость



Не релевантность



Выхода два: либо преобразование либо исключение

СОРАЩЕНИЕ РАЗМЕРНОСТИ

Дано: входные переменные {x1,…,xn} и выходная (числовая или бинарная) y

Задача: оставить только значимые и независимые xi

Работает в два этапа:

1. Уделяет все x_i, где $R^2(x_i)<T1$

   удаление незначимых

2. Forward stepwise регрессия

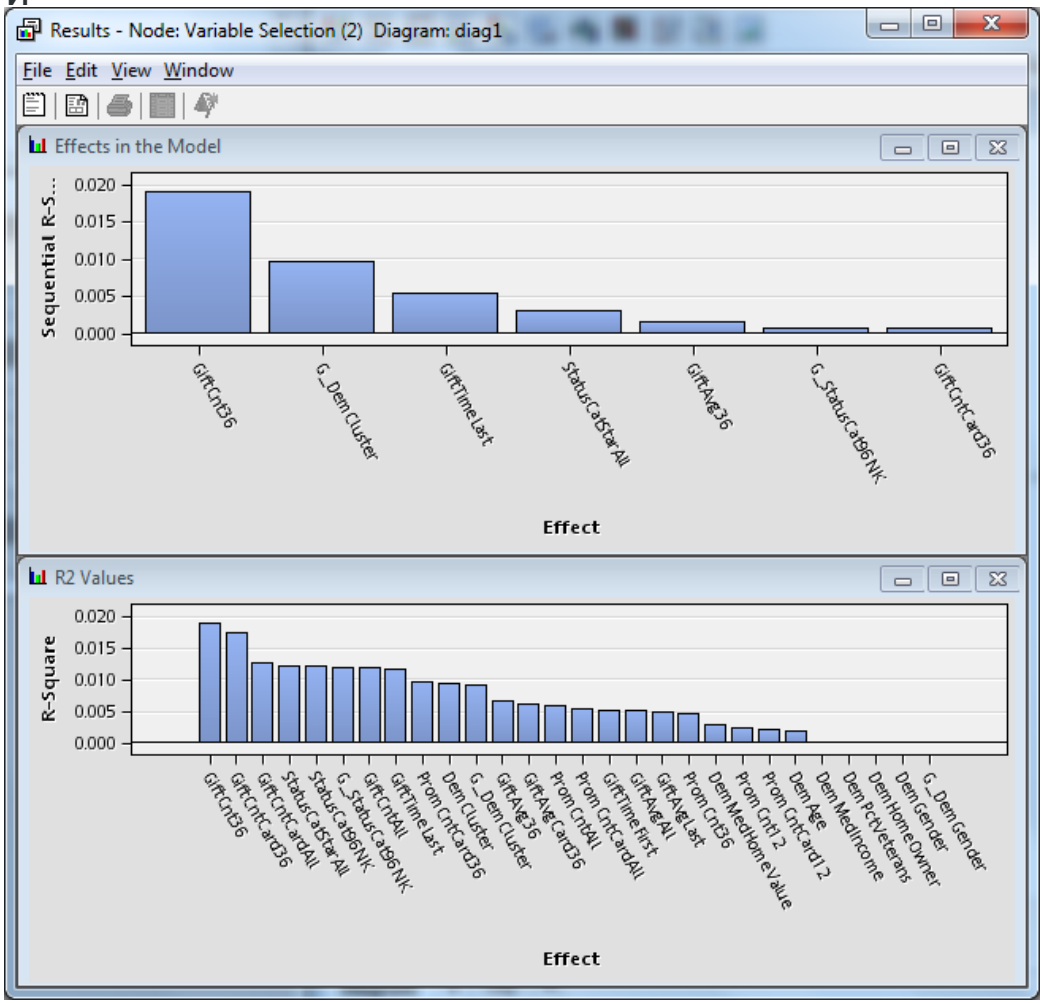
   $f(x_{i1},\ldots x_{ik})$ пока

   $R^2\,(f(x_{i1},\ldots xi_{ik}))-R^2\,(f(x_{i1},\ldots x_{ik-1}))>T2$

   удаление зависимых

Преобразования переменных:

- Дискретизация непрерывных
- Группировка категориальных



ПРОПУЩЕННЫЕ ЗНАЧЕНИЯ

- Не все значения атрибутов известны или достоверны
  - Наиболее важная задача, так как многие к ней сводятся (удаление шума, не консистентностей и т.д.)
- Причины появления пропущенных значений
  - Ошибки «оборудования» и/или ПО при получении данных от датчиков и из экспериментов
  - Удаление несогласованных значений атрибутов
  - Просто не введены в систему из-за халатности или ошибки
  - Часть данных может быть опциональна с точки зрения бизнес процессов организации, но важна для анализа
  - Не хранится правильная история изменений – невозможно правильно определить значение на момент анализа
- Пропущенные данные:
  - Ведут к неточным результатам анализа
  - Допускаются не всеми алгоритмами анализа

МЕТОДЫ ОБРАБОТКИ ПРОПУЩЕННЫХ ЗНАЧЕНИЙ

- Игнорировать объект или запись:
  - Можем потерять важные объекты (например, опорные вектора)
  - Можем «испортить» выборочное распределение
  - В некоторых задачах процент пропущенных значений велик (>50%)
- Заполнение пропущенных значений «вручную»:
  - Нужен очень грамотный эксперт
  - Полностью «вручную» невозможно для больших объемов
  - Правила заполнения (импутации) трудно формулировать – проблема полноты, противоречивости, достоверности
- Использование глобальной спец. константы типа “unknown”
  - Не всеми алгоритмами анализа реализуемо
- Импутация «среднего» или «наиболее ожидаемого» значения
  - По всей выборке, по страту (срезу), по классу, по кластеру и т.д.
  - Наиболее популярный метод
  - но можем «испортить» выборочное распределение
- Методы импутации на основе DM
  - Будем рассматривать

ВОЗМОЖНОСТИ ИНСТРУМЕНТАРИЯ IMPUTE

- Импутация константным значением - все пропуски для переменной заменяются на:
  - Моду (для категориальных) или мат. ожидание, или пользовательскую константу или робастные оценки
- Импутация псевдослучайным значением:
  - В соответсвии с распределением
- Импутация прогнозом (оценкой)
  - Только деревья решений (но можно делать свои модели)

Распределения

Оценки

$x_i = f(x_1, \ldots, x_p)$

Для неслучайных пропусков – индикаторные переменные

- Одна на все наблюдение
- Своя для каждой переменной

ПРЕОБРАЗОВАНИЕ НЕПРЕРЫВНЫХ ПЕРЕМЕННЫХ

- Простые преобразования:
  - Функции от исходной (log, exp, …)

  - Нормализация (z-score, центрирование, сведение на [0,1])

$$v' = \frac{v - mean_A}{stand\_dev_A} \qquad v' = \frac{v - min_A}{max_A - min_A}$$

  - Дискретизация (ранве интервалы, равные группы и т.д.)
- Адаптивные преобразования – перебор простых и выбор лучшего по некоторому криетрию:
  - Нормальность распределения результата
  - Корреляция с откликом
  - Оптимальная дискретизация

ОБЪЕДИНЕНИЕ РЕДКИХ ЗНАЧЕНИЙ КАТЕГОРИАЛЬНОЙ ПЕРЕМЕННОЙ

Level	N_i	ΣY_i	p_i
A	1562	430	0.28
B	970	432	0.45
C	223	45	0.20
D	111	36	0.32
E	85	23	0.27
F	50	20	0.40
G	23	8	0.35
H	17	5	0.29
I	12	6	0.50
J	5	5	1.00

БИНАРНОЕ КОДИРОВАНИЕ КАТЕГОРИАЛЬНЫХ ПЕРЕМЕННЫХ

Level	D_A	D_B	D_C	D_D	D_E	D_F	D_G	D_H	D_I
A	1	0	0	0	0	0	0	0	0
B	0	1	0	0	0	0	0	0	0
C	0	0	1	0	0	0	0	0	0
D	0	0	0	1	0	0	0	0	0
E	0	0	0	0	1	0	0	0	0
F	0	0	0	0	0	1	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	0	0	1	0
I	0	0	0	0	0	0	0	0	1

ГРУППИРОВКА ЗНАЧЕНИЙ КАТЕГОРАЛЬНОЙ ПЕРЕМЕННОЙ (ПО ОТКЛИКУ ИЛИ ЭКСПЕРТНО)

Level	D_{ABCD}	D_B	D_C	D_D	D_{EF}	D_F	D_{GH}	D_H	D_I
A	1	0	0	0	0	0	0	0	0
B	1	1	0	0	0	0	0	0	0
C	1	0	1	0	0	0	0	0	0
D	1	0	0	1	0	0	0	0	0
E	0	0	0	0	1	0	0	0	0
F	0	0	0	0	1	1	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	0	1	1	0
I	0	0	0	0	0	0	0	0	1

это делать умеет компонента