Генерация признаков

Дьяконов А.Г.

Московский государственный университет имени М.В. Ломоносова (Москва, Россия)

Признак

– это некоторая функция определённая на множестве объектов.

Признак пол

Клинты → {М, Ж}

Признак доход

Клинты → {..., 10 000, 20 000, ..., NA}

Значения признака могут быть не опредлены
это тоже важная информация!

Некоторые значения можно восстановить
(например, пол)

Контекстные признаки

– это признаки, смысл которых явно прописан в постановке задачи или понятен из контекста.

Смысл определяет:

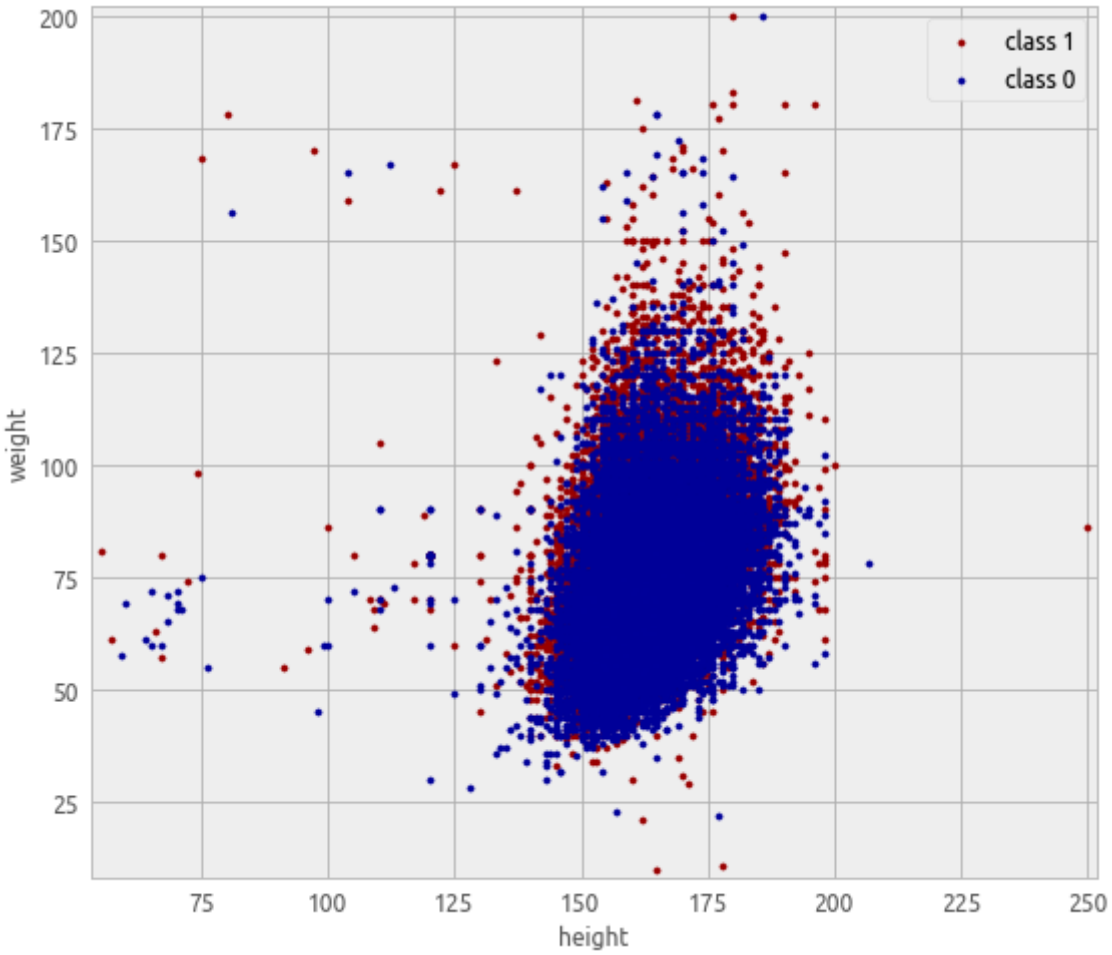
- **область значений**
- **примерное распределение в этой области**

ap_hi	ap_lo	ap_hi_new	ap_lo_new
150	1100	150	110
11	70	110	70
12	80	120	80
11	570	115	70
1	2080	120	80

Пример: диаметр зрачка

Контекстные признаки

Диаграмма рассеивания пар контекстных признаков



- аномальные зоны
- концентрации значений

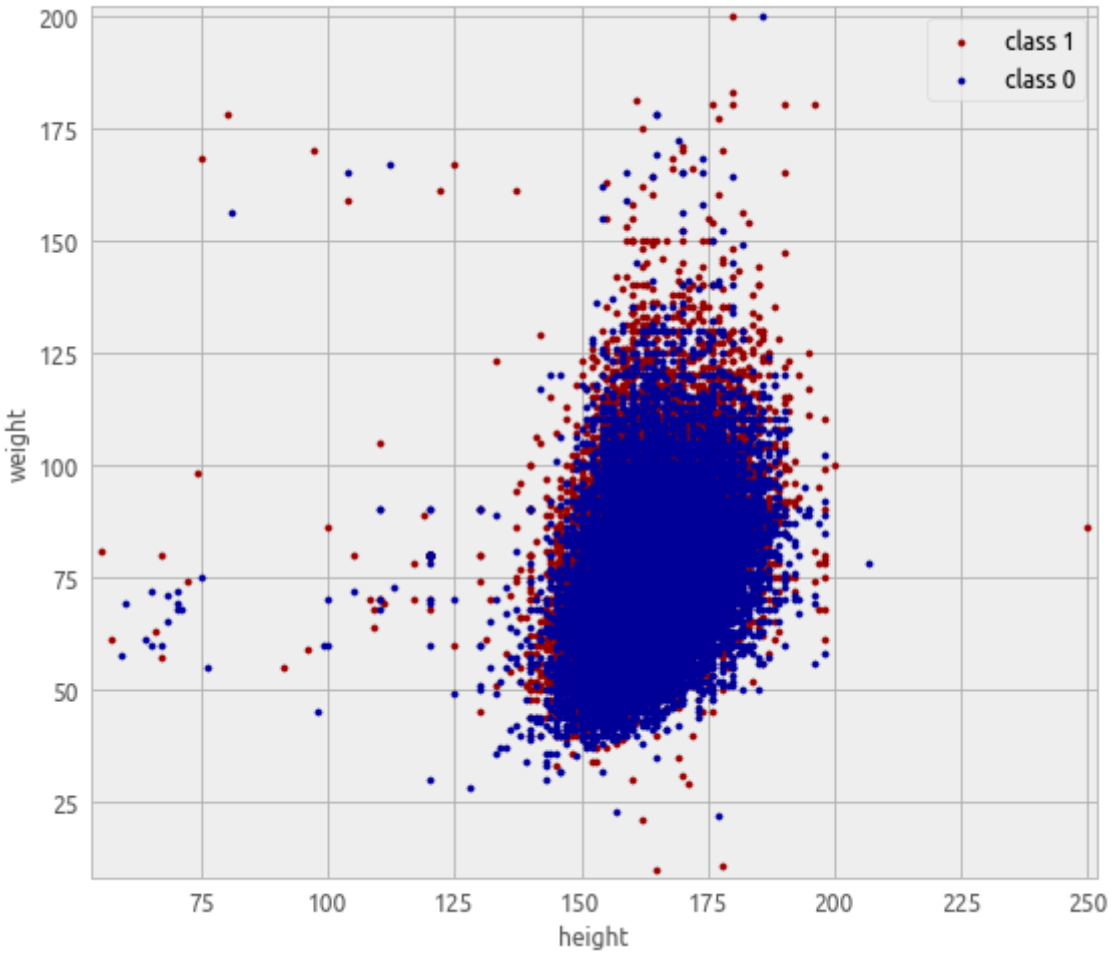
новые признаки

- признак аномальности
- признак частого значения
- отклонение от регрессионной модели построенной по паре

зачем?

Контекстные признаки

Диаграмма рассеивания пар контекстных признаков



- аномальные зоны
- концентрации значений

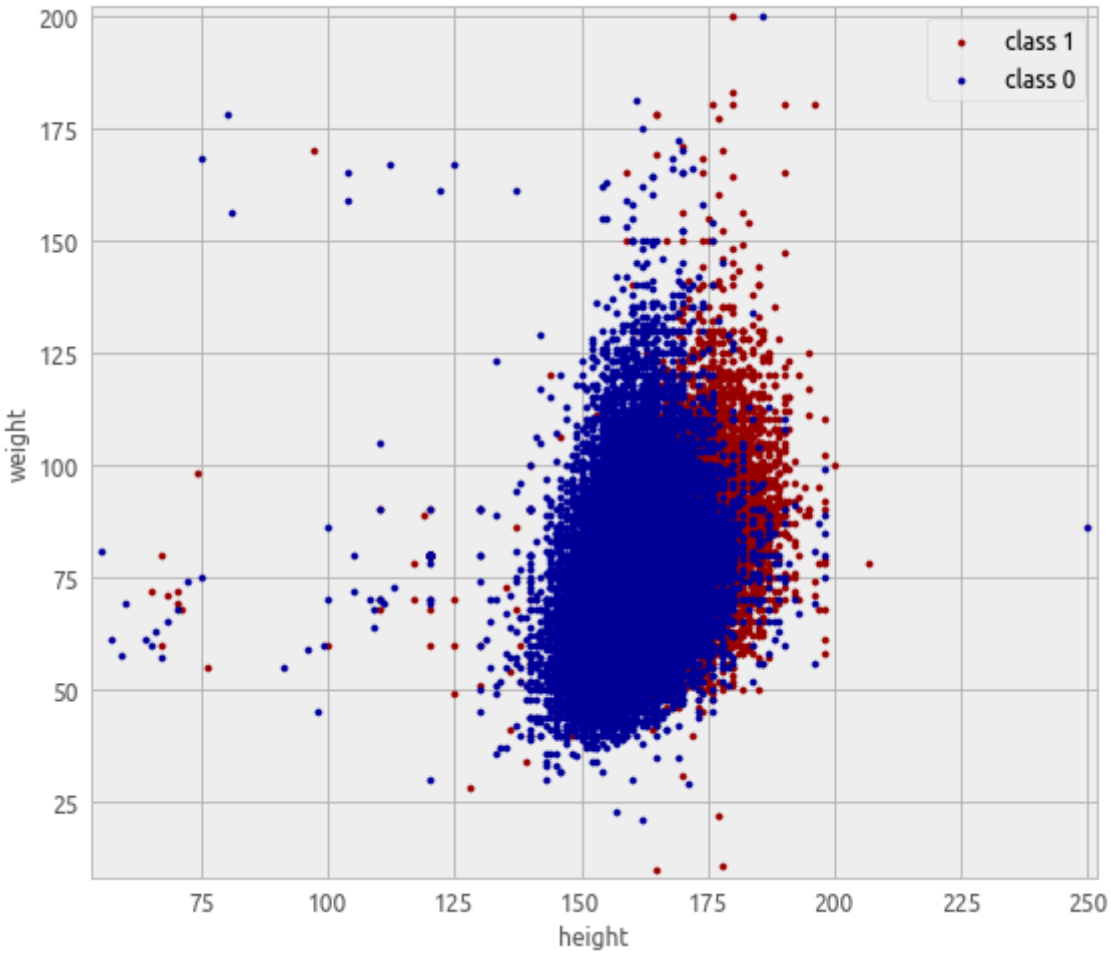
новые признаки

- признак аномальности
- признак частого значения
- отклонение от регрессионной модели построенной по паре

степень переедания/недоедания пациента

Контекстные признаки

Диаграмма рассеивания пар контекстных признаков



Та же диаграмма, но целевой признак здесь – пол

Можно предположить, что 1 – Мужчины

Dummy-признаки

могут не входить в явном виде в признаковую матрицу, но из значения определяются из способа организации данных.

Id	Пол	Рост	Вес
1	М	170	80
20	Ж	NA	70
23	М	167	75
33	М	NA	NA
40	Ж	180	65

Dummy-признаки

могут не входить в явном виде в признаковую матрицу, но из значения определяются из способа организации данных.

#	Id	Пол	Рост	Вес	#NA	Пол=М
1	1	М	170	80	0	1
2	20	Ж	NA	70	1	0
3	23	М	167	75	0	1
4	33	М	NA	NA	2	2
5	40	Ж	180	65	0	0

- **номер строки, id, чётность id**
- **train/test/valid**
- **характеристические признаки**

Казалось бы, логично не рассматривать такие признаки…

Dummy-признаки

Когда Dummy-признак важен…

в.давл	Round_1
120	0
122	2
130	0
111	1
111	1

«Круглые числа» – м.б. неточные/быстрые замеры

Утечка в данных

– информация, которая повышает качество решения задачи машинного обучения, но теряет эти свойства при тестировании на независимом и правильно организованном контроле

Пол	Рост	Вес	Класс
М	170	80	0
Ж	NA	70	0
М	167	75	0
М	NA	NA	1
Ж	180	65	1

Зависимость от Dummy-признаков, в том числе

- **от порядка**
- **от способа представления (названия файлов и т.п.)**
- **от особенностей (наличия пропусков, дубликатов и т.п.)**

Странности в данных

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
1143	1586	23351	1	150	90.0	150	90	3	2	0	0	1	1
1503	2122	16534	1	164	73.0	164	73	1	1	0	0	1	1
3420	4838	14516	1	100	70.0	100	70	1	1	0	0	1	0
3735	5278	17642	1	120	70.0	120	70	1	1	0	0	1	0
3799	5378	23434	1	150	61.0	150	61	1	3	0	0	1	1
4212	5946	16110	1	120	80.0	120	80	1	1	0	0	1	0
7058	10053	21025	1	140	90.0	140	90	3	1	0	0	1	1
7305	10412	15859	1	120	80.0	120	80	1	1	0	0	1	0

Что странного?

Странности в данных

	Id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
1143	1586	23351	1	150	90.0	150	90	3	2	0	0	1	1
1503	2122	16534	1	164	73.0	164	73	1	1	0	0	1	1
3420	4838	14516	1	100	70.0	100	70	1	1	0	0	1	0
3735	5278	17642	1	120	70.0	120	70	1	1	0	0	1	0
3799	5378	23434	1	150	61.0	150	61	1	3	0	0	1	1
4212	5946	16110	1	120	80.0	120	80	1	1	0	0	1	0
7058	10053	21025	1	140	90.0	140	90	3	1	0	0	1	1
7305	10412	15859	1	120	80.0	120	80	1	1	0	0	1	0

рост = верхнее давление

вес = нижнее

Строковые признаки

T	company	client
12C	Shell	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36
14	shel	NA
15	bp	NA
11	B&P	NA
10C	Procter&Gamble	NA

Строковые признаки

T	company	browser	ip	OS
12	Shell Gas station	Mozilla	53.0.2785.143	Mac
14	Shell Gas station	NA	NA	NA
15	BP Gas station	NA	NA	NA
11	BP Gas station	NA	NA	NA
10	P&G Manufacturer	NA	NA	NA

Категориальные признаки

1. Автоматическое определение категориальности

- если значения – строки
- если мало уникальных значений

	city	class	degree	income
0	Moscow	A	1	10.2
1	London	B	1	11.6
2	London	A	2	8.8
3	Kiev	A	2	9.0
4	Moscow	B	3	6.6
5	Moscow	B	3	10.0
6	Kiev	A	1	9.0
7	Moscow	A	1	7.2

```
# найти все признаки, в которых первое значение - строка
def find_cat(data):
    for name in data.columns:
        s = ''
        s += name
        if (type(data[name][0]) == str):
            s += ' строка,'
        if (data[name].nunique()<=3):
            s += ' мало уникальных'
        if (s!=name):
            print (s)

find_cat(data)
```

```
city строка, мало уникальных
class строка, мало уникальных
degree мало уникальных
```

Категориальные признаки

2. Создание новых категориальных признаков

- **конъюнкция признаков**

	city	class	degree	income	city + degree
0	Moscow	A	1	10.2	Moscow + 1
1	London	B	1	11.6	London + 1
2	London	A	2	8.8	London + 2
3	Kiev	A	2	9.0	Kiev + 2
4	Moscow	B	3	6.6	Moscow + 3
5	Moscow	B	3	10.0	Moscow + 3
6	Kiev	A	1	9.0	Kiev + 1
7	Moscow	A	1	7.2	Moscow + 1

```
# конъюнкция двух признаков
def make_conj(data, feature1, feature2):
    data[feature1 + ' + ' + feature2] =
            data[feature1].astype(str) +
            ' + ' +
            data[feature2].astype(str)

    return (data)

# пример использования
make_conj(data, 'city', 'degree')
```

Категориальные признаки

2. Создание новых категориальных признаков

- конъюнкция признаков
- создание новых признаков по контекстным

Пример: верхние уровни иерархии

Категориальные признаки

3. Простейшее кодирование – по номеру категории Label Encoding

- sklearn.preprocessing.LabelEncoder
- dict + map

	city	class	degree	income	city_le
0	Moscow	A	1	10.2	2
1	London	B	1	11.6	1
2	London	A	2	8.8	1
3	Kiev	A	2	9.0	0
4	Moscow	B	3	6.6	2
5	Moscow	B	3	10.0	2
6	Kiev	A	1	9.0	0
7	Moscow	A	1	7.2	2

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le.fit(data.city)
data['city_le'] = le.transform(data.city)
data
```

```
# ручная альтернатива
# словарь для кодировки
dct = {'Kiev': 0, 'London': 1, 'Moscow': 2}
data['city_le'] = data['city'].map(dct)
data
```

```
train_d['Manager_Gender'] = train_d['Manager_Gender'].map({'M':1, 'F':-1, nan:0})
```

Категориальные признаки

3. Простейшее кодирование – по номеру категории Label Encoding

- **случайное кодирование**

многократное случайное кодирование иногда хорошо работает с RF

- **не подходит для линейных алгоритмов**
- **проблема новых категорий**

(средним)

Категориальные признаки

4. Dummy-кодирование / One-hot-encoding

```
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder(sparse=False)
new_ohe_features = ohe.fit_transform(data.city_le.values.reshape(-1, 1))
tmp = pd.DataFrame(new_ohe_features, columns=['city=' + str(i) for i in range(new_ohe_features.shape[1])])
data = pd.concat([data, tmp], axis=1)
data
```

	city	class	degree	income	city=0	city=1	city=2
0	Moscow	A	1	10.2	0	0	1
1	London	B	1	11.6	0	1	0
2	London	A	2	8.8	0	1	0
3	Kiev	A	2	9.0	1	0	0
4	Moscow	B	3	6.6	0	0	1
5	Moscow	B	3	10.0	0	0	1
6	Kiev	A	1	9.0	1	0	0
7	Moscow	A	1	7.2	0	0	1

```
# ручной способ
def code_myohe(data, feature):
    for i in data[feature].unique():
        data[feature + '=' + i] =
            (data[feature] == i).astype(float)
code_myohe(data, 'city')
data
```

OneHotEncoder по умолчанию порождает sparse-ответ

OneHotEncoder работает только с числами (ручное решение и со строками)

Категориальные признаки

- **Для линейных алгоритмов (кодируем N-1 категорию)**
- **Большое число категорий → сильно разреженные матрицы**

Важно: проблема новых категорий

Категориальные признаки

5. По значениям вещественного признака

```
# функция возвращает значения нового признака
def code_mean(data, cat_feature, real_feature):
    return (data[cat_feature].map(data.groupby(cat_feature)[real_feature].mean()))

data['city_mean_income'] = code_mean(data, 'city', 'income')
data
```

	city	class	degree	income	city_mean_income
0	Moscow	A	1	10.2	8.5
1	London	B	1	11.6	10.2
2	London	A	2	8.8	10.2
3	Kiev	A	2	9.0	9.0
4	Moscow	B	3	6.6	8.5
5	Moscow	B	3	10.0	8.5
6	Kiev	A	1	9.0	9.0
7	Moscow	A	1	7.2	8.5

Естественная интерпретация: товары какой категории дороже

Категориальные признаки

6. По значениям категориального признака

- **просто по мощности Count Encoding – одна строка**

```
feature = 'city'
newfeature = 'city_c'
data[newfeature] = data[feature].map(data.groupby(feature).size())
data
```

	city	class	degree	income	city_c
0	Moscow	A	1	10.2	4
1	London	B	1	11.6	2
2	London	A	2	8.8	2
3	Kiev	A	2	9.0	2
4	Moscow	B	3	6.6	4
5	Moscow	B	3	10.0	4
6	Kiev	A	1	9.0	2
7	Moscow	A	1	7.2	4

Категориальные признаки

Count Encoding

- коллизии (несколько категорий – один код)
- проблема шума (мелкие категории)
- проблема новых категорий

Решение

- + шум
- объединение категорий в одну
- кодируем «1»

Категориальные признаки

6. По значениям ДРУГОГО категориального признака

```
import numpy as np
from numpy.linalg import svd
def code_factor(data, cat_feature, cat_feature2):
    ct = pd.crosstab(data[cat_feature], data[cat_feature2])
    u, _, _ = svd(ct.values)
    coder = dict(zip(ct.index, u[:,0])) # если кодировать первой компонентой
    return (data[cat_feature].map(coder))

data['city_degree_code'] = code_factor(data, 'city', 'degree')
```

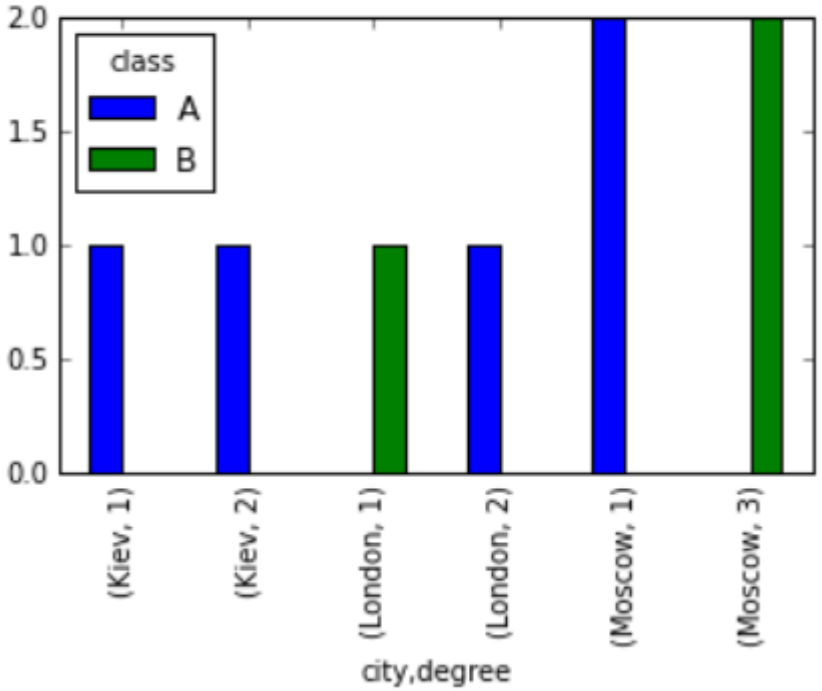
	city	class	degree	income	city_mean_income	city_degree_code
0	Moscow	A	1	10.2	8.5	-0.888074
1	London	B	1	11.6	10.2	-0.325058
2	London	A	2	8.8	10.2	-0.325058
3	Kiev	A	2	9.0	9.0	-0.325058
4	Moscow	B	3	6.6	8.5	-0.888074
5	Moscow	B	3	10.0	8.5	-0.888074
6	Kiev	A	1	9.0	9.0	-0.325058
7	Moscow	A	1	7.2	8.5	-0.888074

Полезный инструмент – функция crosstab

```
ct = pd.crosstab([data.city, data.degree], data['class'])
ct.plot(kind='bar')
```

	city	class	degree	income
0	Moscow	A	1	10.2
1	London	B	1	11.6
2	London	A	2	8.8
3	Kiev	A	2	9.0
4	Moscow	B	3	6.6
5	Moscow	B	3	10.0
6	Kiev	A	1	9.0
7	Moscow	A	1	7.2

	class	A	B
city	degree		
Kiev	1	1	0
	2	1	0
London	1	0	1
	2	1	0
Moscow	1	2	0
	3	0	2



Категориальные признаки

8. Хэш-кодирование

- средство против сильно разреженных данных
- могут возникать коллизии (можно выполнять разные хэш-кодирования)

9. По значению целевого – Target Encoding

- это форма стэкинга (по одной переменной)
- подходит для любых алгоритмов (если правильно сделана)
  - cv-кодировка
  - добавление шума
  - сглаживание

Пример: кодирование по куску испорченных данных

10. Экспертное кодирование

см. также кодирование названий географических регионов

11. Вложение категориальных признаков в маломерное пространство (Category Embedding)

Категориальные признаки

Пропуски в категориальных признаках

- **создание отдельной категории ("нет значения")**
- **игнорирование пропусков (например, в dummy-кодировке сопоставить им нулевую строку, т.е. соответствующие объекты не принадлежат ни одной категории)**
- **стандартные методы обработки пропусков (при плотном кодировании, например, по целевому вектору)**

Вещественные признаки

- деформация (функция над признаком)
- нормировка (специальный вид деформации)
- новые признаки (функции над несколькими)
- дискретизация (binning)

Продажа планшетов	Продажа телефонов	Продажа ноутбуков	Общие продажи
10	30	1	32
12	42	2	56
10	20	1	31
15	31	2	48
5	15	0	20

Вещественные признаки

Деформация

- логарифм
- корень

Совет: делать деформацию, если она делает похожим распределение по признаку на нормальное

Нормировка

Приведение всех признаков «в одну шкалу»: k-NN, k-means, SVM

Пример: регуляризация временных рядов

- **стандартизация**
- **на отрезок**
- **по максимуму**
- **по сумме**

Заблуждение: инвариантность RF относительно монотонных преобразований.

Вещественные признаки

Новые признаки

- суммы групп признаков
- мономы

дискретизация (binning)

- округление
- кластеризация (!)

Часто вместо конкретного значения – порядок

Решается проблема выбросов

Временные признаки

```
train_d[:5]
```

	ID	Office_PIN	Application_Receipt_Date	Applicant_City_PIN	Applicant_Gender	Applicant_BirthDate	Applicant_Marital_Status
0	FIN1000001	842001	4/16/2007	844120	M	12/19/1971	M
1	FIN1000002	842001	4/16/2007	844111	M	2/17/1983	S
2	FIN1000003	800001	4/16/2007	844101	M	1/16/1966	M
3	FIN1000004	814112	4/16/2007	814112	M	2/3/1988	S
4	FIN1000005	814112	4/16/2007	815351	M	7/4/1985	M

1. Преобразование признаков

```
train_d['Application_Receipt_Date'] =
pd.to_datetime(train_d.Application_Receipt_Date)
train_d['Applicant_BirthDate'] = pd.to_datetime(train_d.Applicant_BirthDate)
train_d['Manager_DOJ'] = pd.to_datetime(train_d.Manager_DOJ)
train_d['Manager_DoB'] = pd.to_datetime(train_d.Manager_DoB)
```

2. Генерация новых признаков

Каких?

Временные признаки

2.1. Характеристика момента времени:

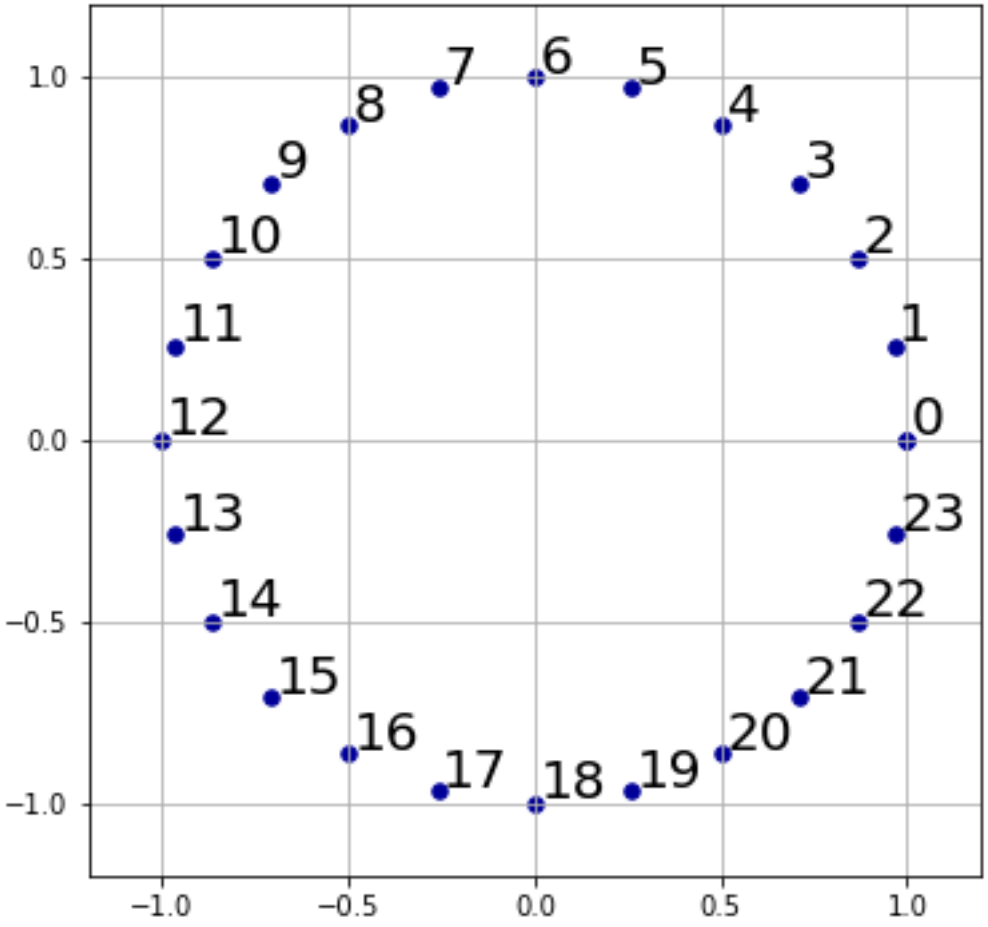
- **час, минута, секунда (=0)**
- **время суток**
- **день, день недели, день года**
- **неделя, месяц**
- **время года, год**
- **праздник / выходной / особый день** (первый понедельник месяца, начало Олимпиады)

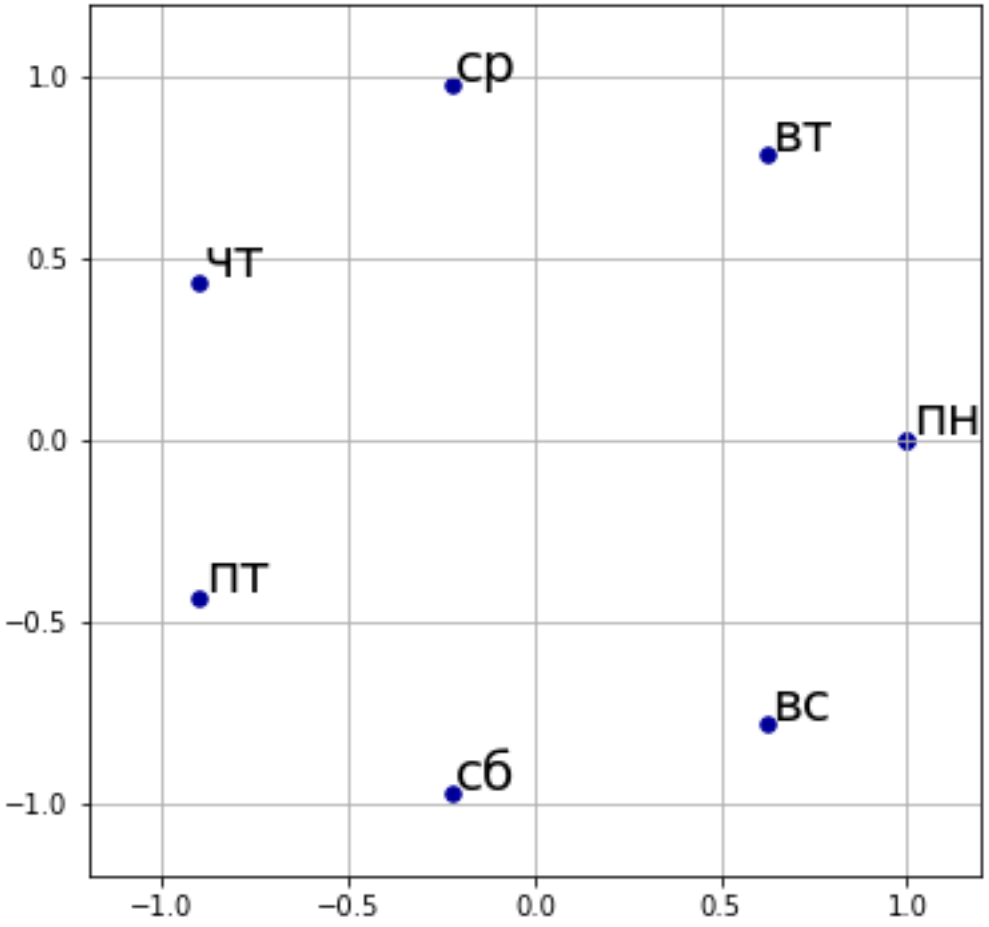
```
tmp = test_d.Manager_DOJ.dt
pd.DataFrame({'day': tmp.day,
              'dayofweek': tmp.dayofweek,
              'dayofyear': tmp.dayofyear,
              'month': tmp.month})
```

	day	dayofweek	dayofyear	month
0	26	0	147	5
1	24	1	176	6
2	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN
4	4	1	338	12

Временные признаки

Кодирование циклических признаков





Временные признаки

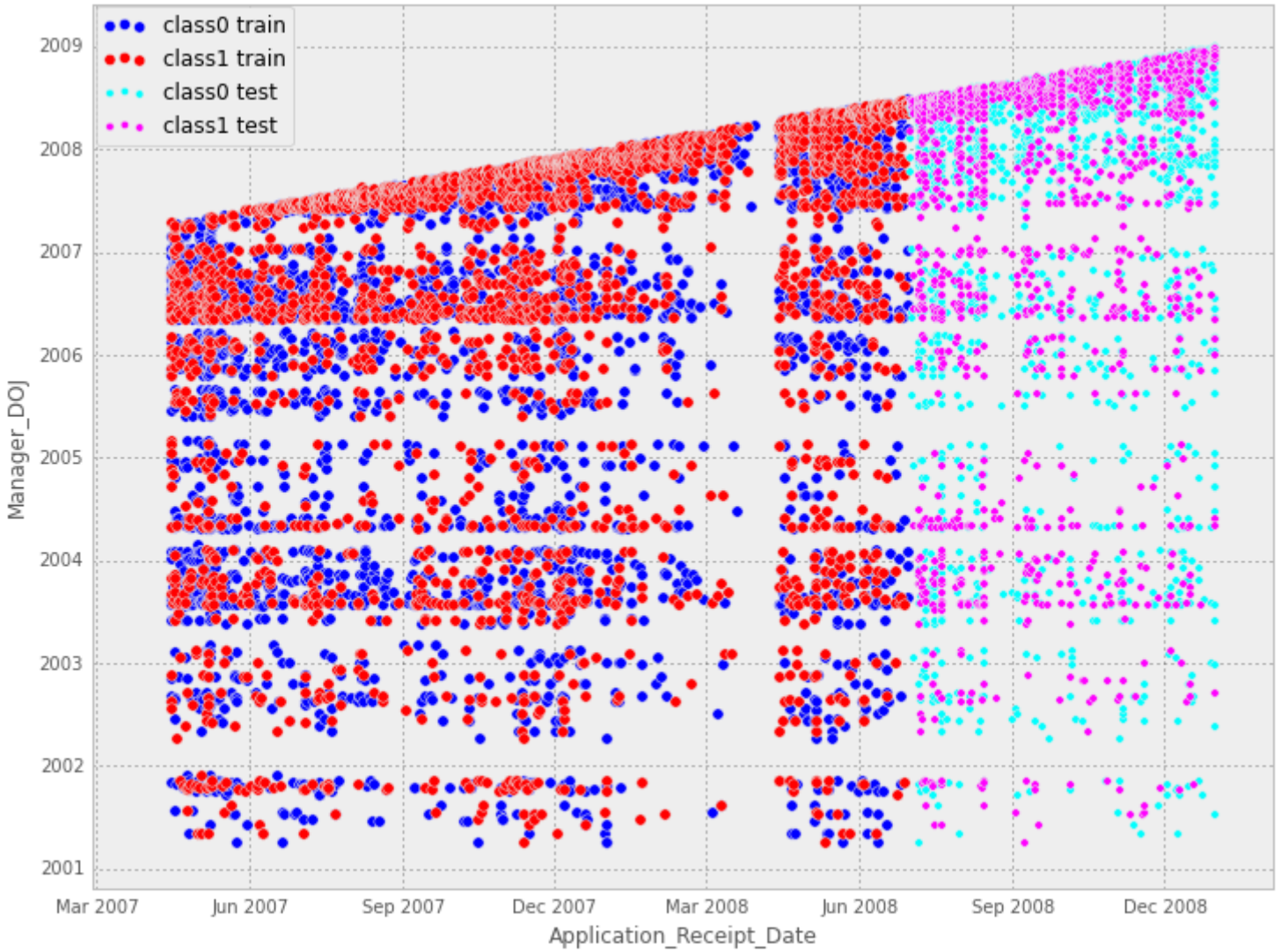
2.2. Взаимодействие пары признаков

- разница времён
- близость к дедлайну
- в один ли день недели/год и т.п.

2.3. Использование для других признаков

- устаревание в весовых схемах
- что за последний промежуток T (операции по карте за последний месяц)
- формирование разбиения обучение / тест

Пример признака



День сделки / когда менеджер начал работать

Разница – опыт менеджера

Временные признаки

2.3. Взаимодействие с другими признаками

Можно смотреть на стабильность признаков!

	train-1		train-2		test	
	size	mean	size	mean	size	mean
-1.1	67	0.000000	19	0.000000	44	0.000000
Associate / Fellow of Institute of Chartered Accountans of India	3	0.333333	NaN	NaN	2	1.000000
Associate/Fellow of Institute of Company Secretories of India	1	0.000000	NaN	NaN	NaN	NaN
Associate/Fellow of Insurance Institute of India	1	1.000000	NaN	NaN	NaN	NaN
Class X	221	0.294118	4	0.250000	19	0.842105
Class XII	5318	0.322114	488	0.409836	1357	0.677966
Graduate	1829	0.374522	1367	0.373811	3375	0.715852
Masters of Business Administration	33	0.333333	41	0.390244	71	0.760563
Others	56	0.535714	76	0.407895	171	0.824561
Associate/Fellow of Acturial Society of India	NaN	NaN	1	0.000000	NaN	NaN
Certified Associateship of Indian Institute of Bankers	NaN	NaN	1	1.000000	NaN	NaN
Professional Qualification in Marketing	NaN	NaN	1	1.000000	5	0.600000
Associate/Fellow of Institute of Institute of Costs and Works Accountants of India	NaN	NaN	NaN	NaN	1	0.000000

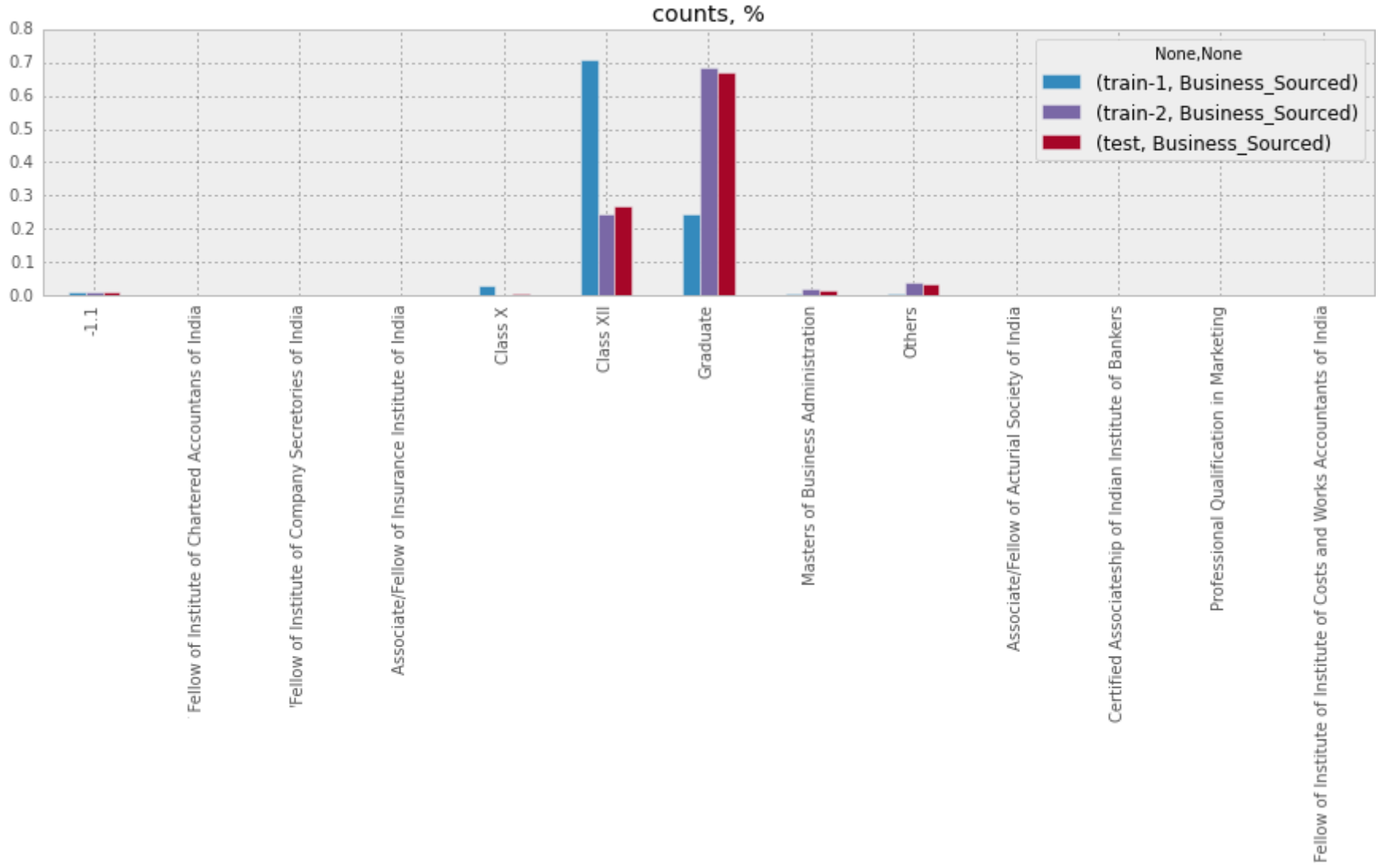
нулевая цель

редкие признаки

нестабильность: по частотам и по цели

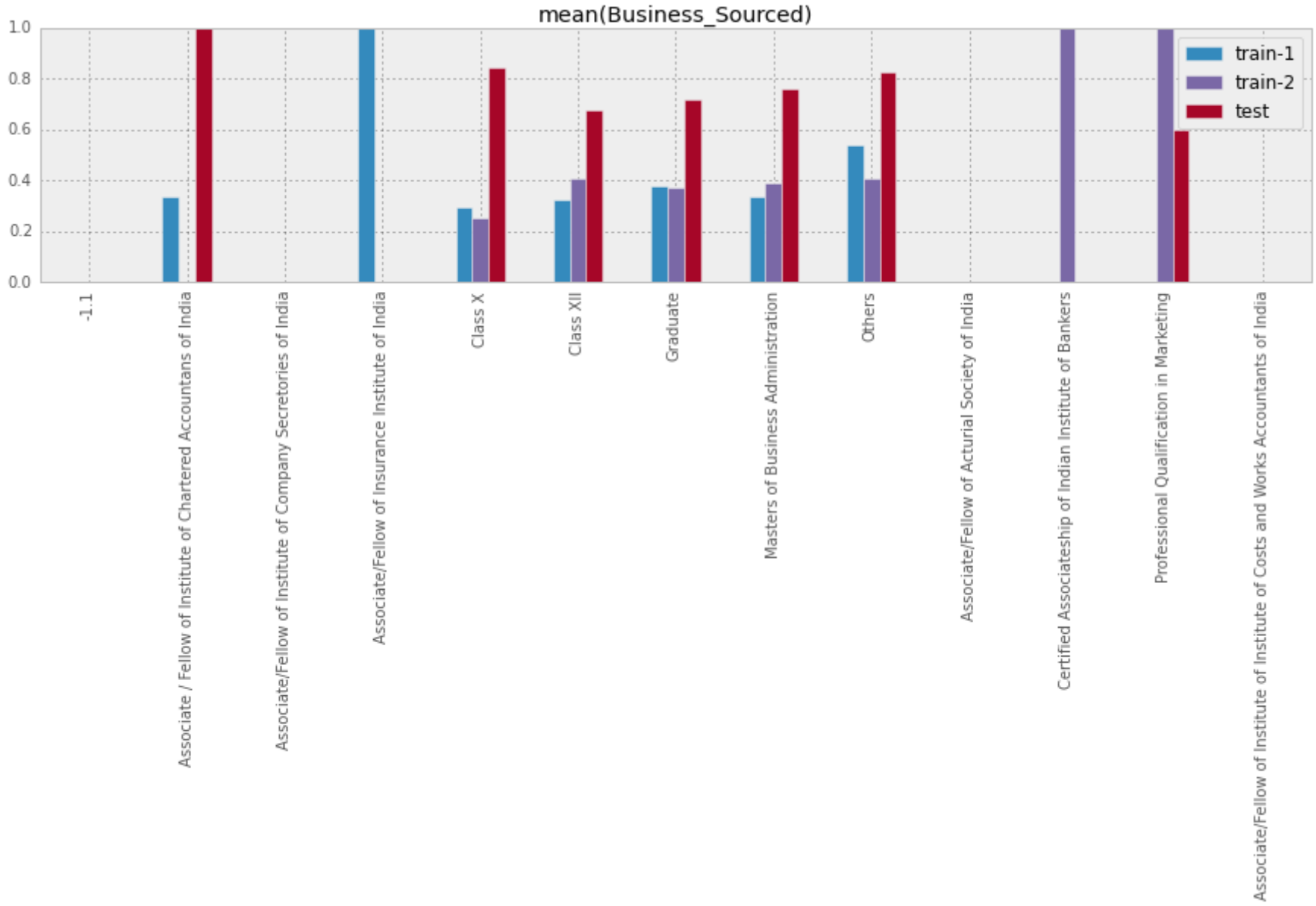
Временные признаки

Как часто встречаются такие значения признаков



Временные признаки

Как меняется средне значение целевой переменной



Временные признаки

2.3. Смотрим, как меняются другие признаки во времени...

Временные признаки

2.3. Смотрим, как меняются другие признаки во времени…

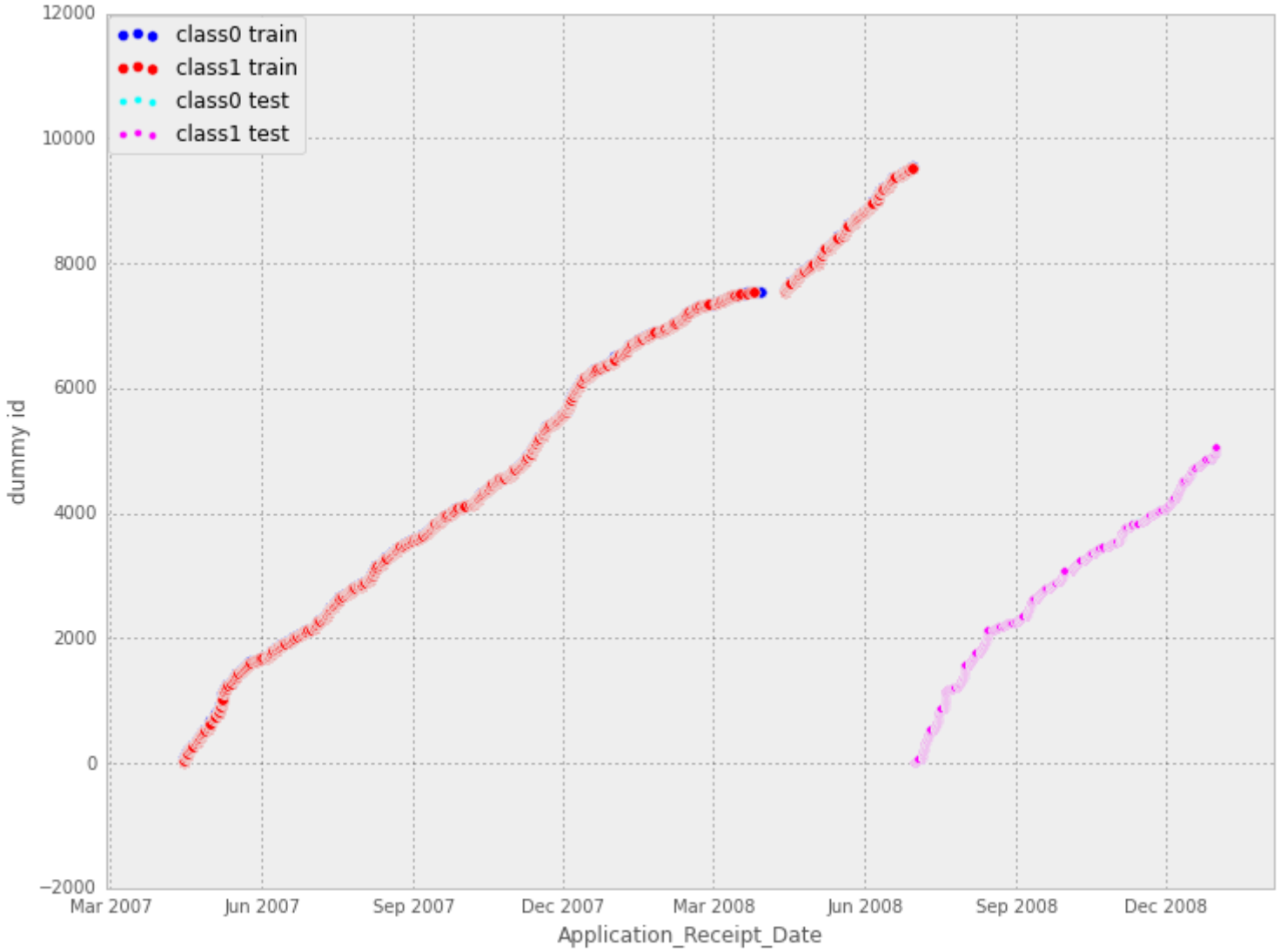
Это позволяет:

- выявить стабильные признаки

- правильно сформировать разбиения обучение / тест

Приём: по старой истории кодировать признаки, по новой обучать!

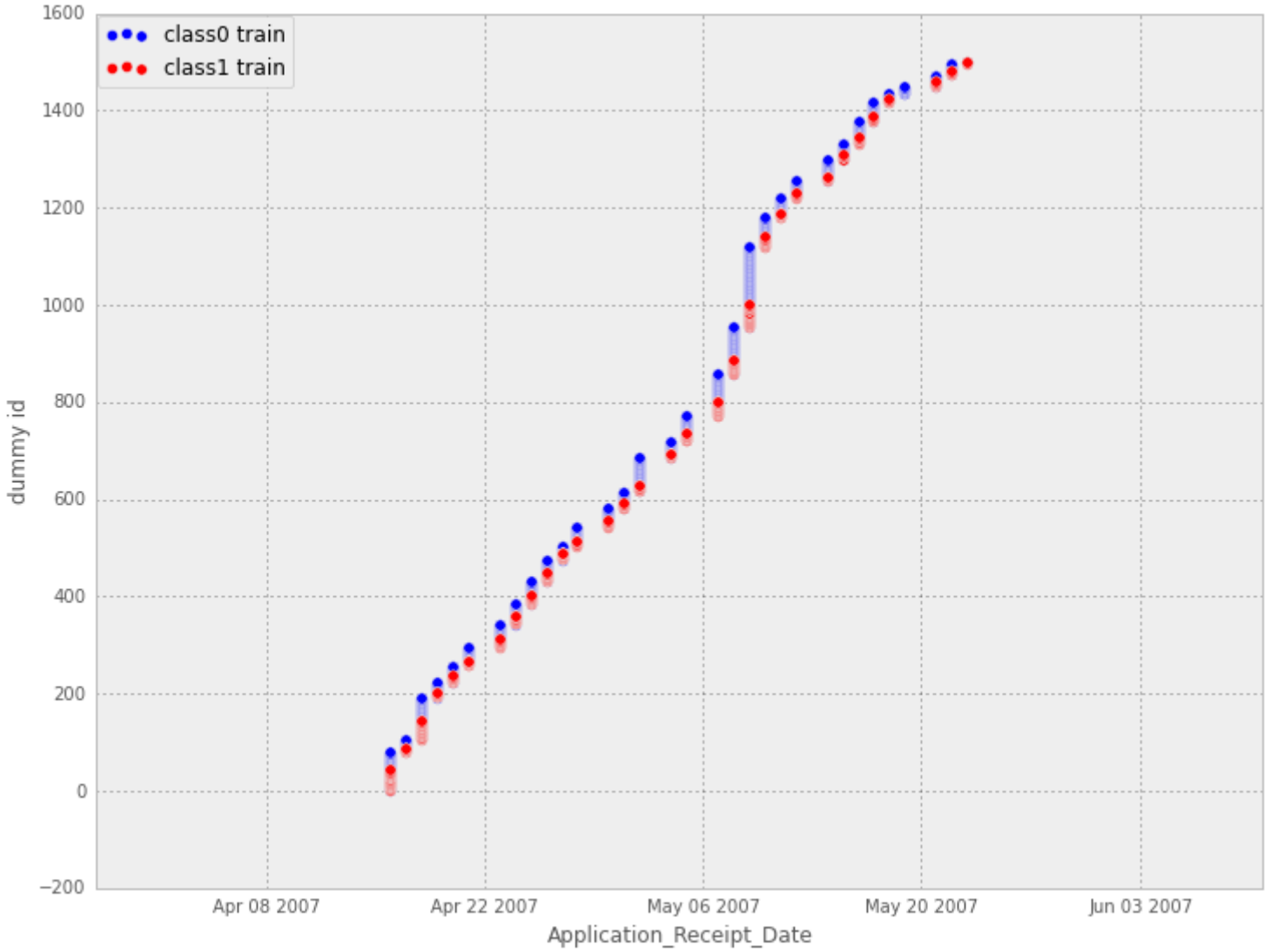
Совет: визуализация id (номер в таблице) – время



Позволяет много чего выявить.

Здесь – разрыв, разную скорость заполнения данными

Совет: визуализация id (номер в таблице) – время



Если присмотреться очень внимательно, то вообще видна «утечка»

Совет: визуализация id (номер в таблице) – время

	Application_Receipt_Date	Business_Sourced		Application_Receipt_Date	Business_Sourced
0	2007-04-16	0	79	2007-04-17	1
1	2007-04-16	1	80	2007-04-17	1
2	2007-04-16	0	81	2007-04-17	1
3	2007-04-16	0	82	2007-04-17	1
4	2007-04-16	0	83	2007-04-17	1
5	2007-04-16	1	84	2007-04-17	1
6	2007-04-16	1	85	2007-04-17	1
7	2007-04-16	0	86	2007-04-17	1
8	2007-04-16	1	87	2007-04-17	0
9	2007-04-16	1	88	2007-04-17	0
10	2007-04-16	1	89	2007-04-17	0
11	2007-04-16	1	90	2007-04-17	0
12	2007-04-16	1	91	2007-04-17	0

Это можно не заметить при беглом просмотре таблицы… слева – первые строки, справа – последующие.

Задача для программирования

	date
0	0
1	0
2	0
3	1
4	2
5	2
6	2
7	2
8	2
9	3
10	3
11	3
12	3

Дано: признаковая таблица, записи упорядочены по времени добавления, один из признаков – дата добавления (монотонно неубывает)

Надо: добавить новые признаки. Для каждой записи посчитать порядковый номер в этот день, сколько записей ещё будет сделано в этот день, сколько процентов записей этого дня на момент добавления записи сделано.

Решение

```
data['num_of_sale'] = np.arange(data.shape[0])

tmp = data['date'].map(data.groupby('date').num_of_sale.min())

data['sales_in_day'] = data['date'].map(data.groupby('date').num_of_sale.max()) - tmp + 1

data['num_of_sale'] = data['num_of_sale'] - tmp

data['invert_num_of_sale'] = data['sales_in_day'] - data['num_of_sale'] - 1

data['per_of_sale'] = data.num_of_sale/data['date'].map(data.groupby('date').num_of_sale.max())

data['per_of_sale'] = data['per_of_sale'].fillna(0.0)
```

Результат

	date	sales_in_day	num_of_sale	invert_num_of_sale	per_of_sale
0	0	3	0	2	0.000000
1	0	3	1	1	0.500000
2	0	3	2	0	1.000000
3	1	1	0	0	0.000000
4	2	5	0	4	0.000000
5	2	5	1	3	0.250000
6	2	5	2	2	0.500000
7	2	5	3	1	0.750000
8	2	5	4	0	1.000000
9	3	4	0	3	0.000000
10	3	4	1	2	0.333333
11	3	4	2	1	0.666667
12	3	4	3	0	1.000000

Географические (пространственные) признаки: Spatial Variables

– отражают локализация в пространстве

- **GPS-координаты**
- **города**
- **страны**
- **адреса**
- **траектории / скорости перемещения и т.п.**

Можно сконвертировать в координаты

Географические (пространственные) признаки

Проекции на разные оси

Чтобы реализовывались более сложные «поверхности разделения»

Кластеризация

Чтобы выделить отдельные регионы

Идентификация, привязка

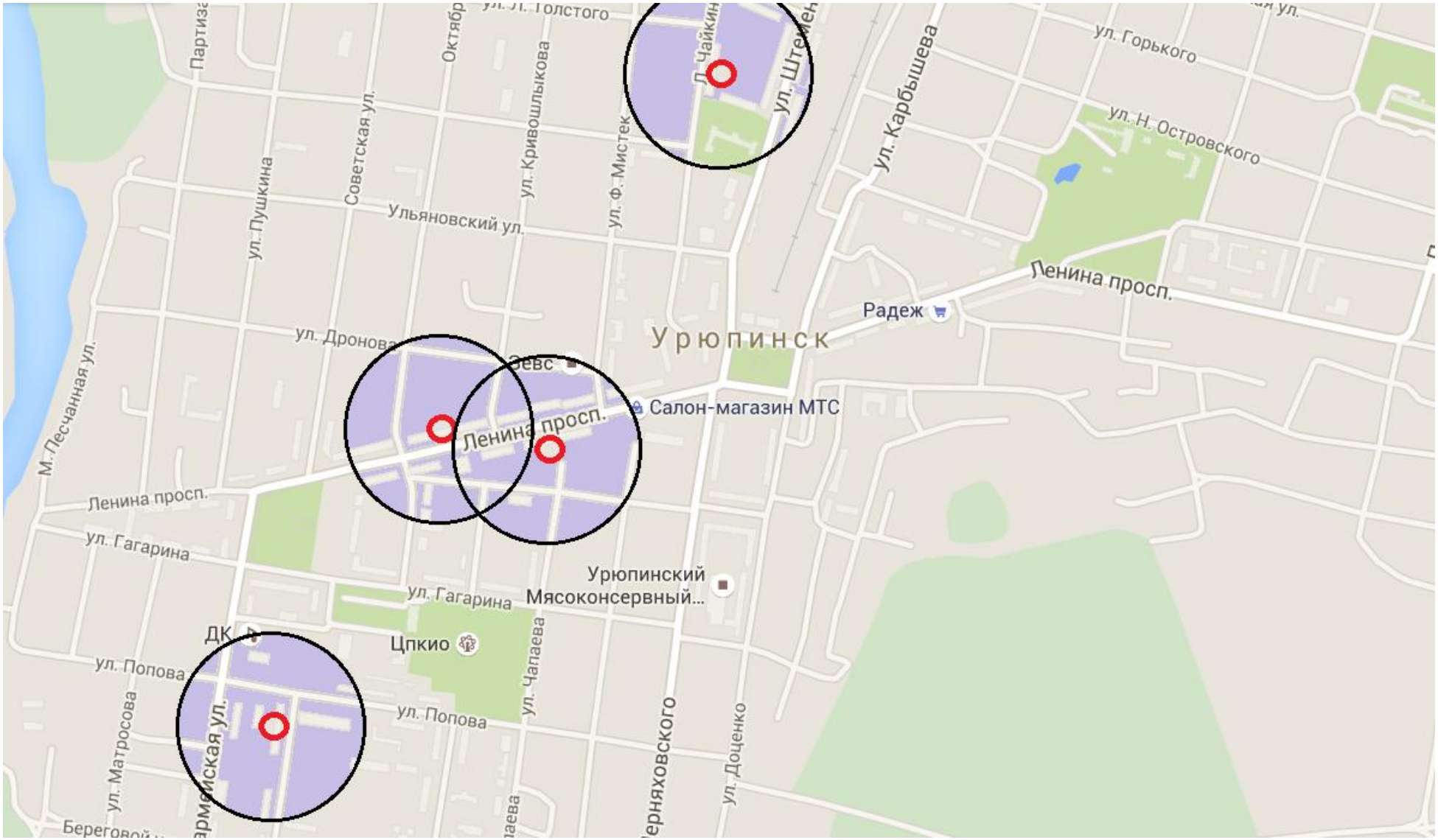
В случае точных координат:

- где находится объект
- какие объекты также рядом (плотность объектов)
- что ещё рядом

Анализ траекторий

если изменение координат во времени

Пример: анализ трафика и конверсии в различных точках продаж



Пример: анализ трафика и конверсии в различных точках продаж

Данные заказчика

- статистика посещений
  - визиты
  - покупки/конверсия
  - ...
- данные магазина
  - площадь
  - персонал
  - категория
  - ...

Наши данные

- Анализ окрестности салона
  - наличие остановок / метро
  - конкурентов
  - где находится магазин (ТЦ)
  - численность населения

Агрегация и слияние

Анкета

id	пол	возраст	сумма	карт
12	М	34	10000	0
15	М	23	50000	1
37	Ж	37	90000	2

БКИ

id	дата	сумма	просрочек
12	10-11-12	1000	0
12	01-02-13	2000	1
15	19-10-11	1000	0
15	05-03-12	2000	0
15	03-07-13	3000	1
15	09-09-13	2000	0
37	23-11-13	5000	0

сумма + веса

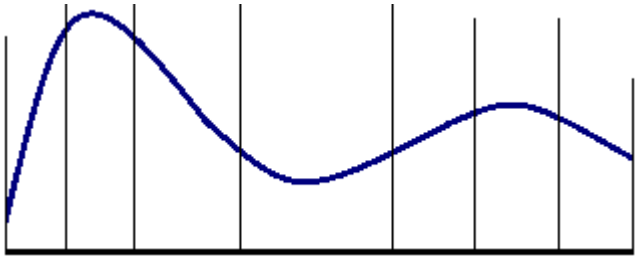
среднее

максимум

минимум

медиана

Контроль с соблюдением пропорций…



Есть стратифицированные разбиения

Группы можно формировать по значениям целевого признака

Отличается ли контроль от обучения

Создание целевого вектора «объект принадлежит контролю»