Прикладные задачи анализа данных

Временные и географические признаки

Дьяконов А.Г.

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

Если есть 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?

2.1. Каждый момент времени:

- какое время суток, день, неделя, месяц, год

- какой день недели

- праздник / выходной / особый день

- какой день года

```
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

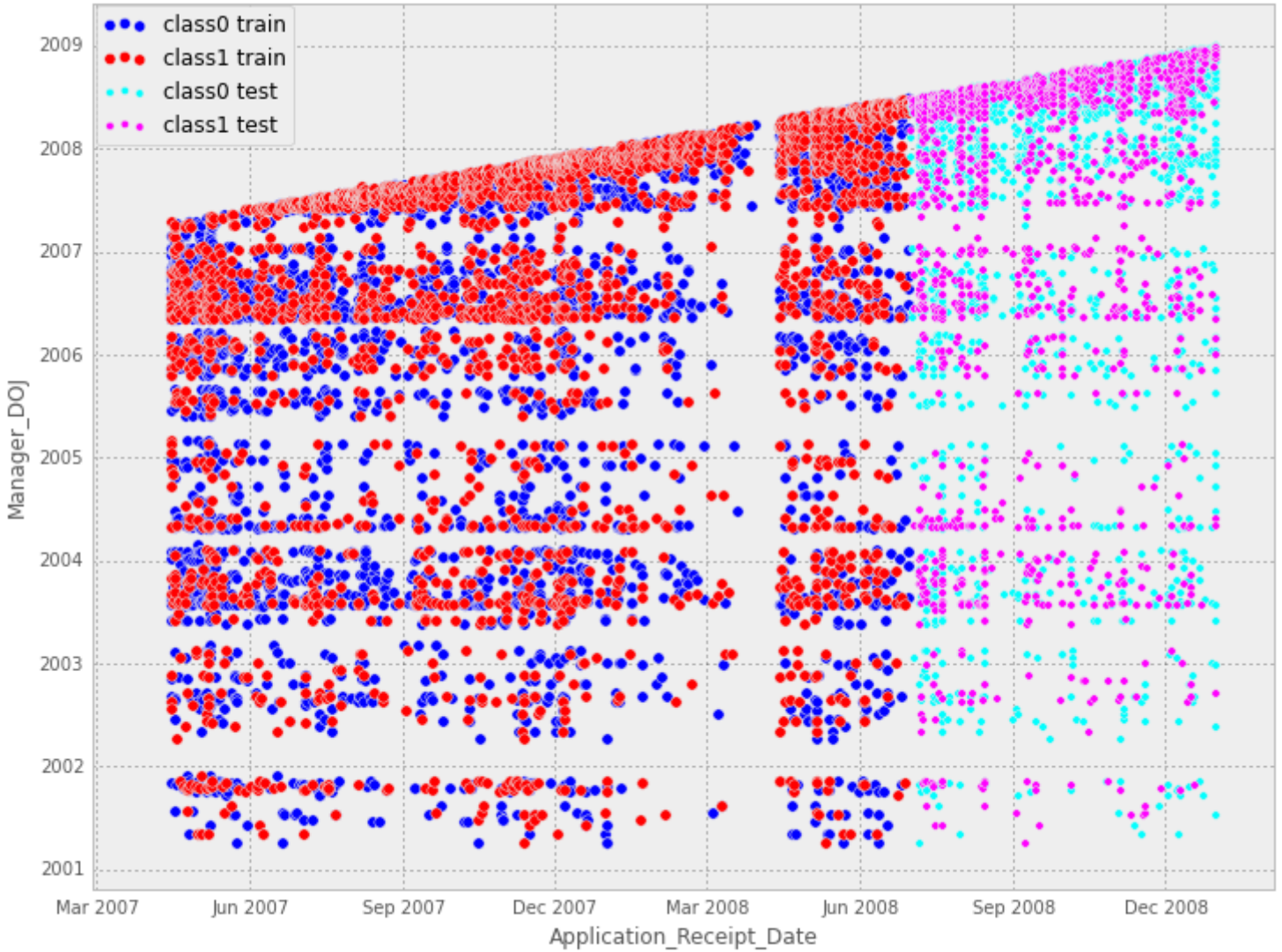
2.2. Взаимодействие пары признаков

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

2.3. Использование для других признаков

- устаревание в весовых схемах
- формирование разбиения обучение / тест

Пример признака



День сделки / когда менеджер начал работать

Разница – опыт менеджера

2.3. Взаимодействие с другими признаками

Время позволяет разбивать выборку на части «естественным образом»

Можно смотреть на стабильность признаков!

```
# ЭТО ПРИЗНАК, КОТОРЫЙ СМОТРИМ
feature = 'Applicant_Qualification'

# два участка
indclass = train_d['Application_Receipt_Date'] > pd.Timestamp('2008-04-01')
train1 = train_d[~indclass]
train2 = train_d[indclass]

print ('UNIQUE TRAIN 1 = ', train1[feature].unique())
print ('UNIQUE TRAIN 2 = ', train2[feature].unique())
print ('UNIQUE TEST = ', test_d[feature].unique())

t1 = train1[[feature, 'Business_Sourced']].fillna(-
1.1).groupby(feature)['Business_Sourced'].agg({'mean':np.mean, 'size':size})
t2 = train2[[feature, 'Business_Sourced']].fillna(-
1.1).groupby(feature)['Business_Sourced'].agg({'mean':np.mean, 'size':size})
t3 = test_d[[feature, 'Business_Sourced']].fillna(-
1.1).groupby(feature)['Business_Sourced'].agg({'mean':np.mean, 'size':size})

# здесь лежат сколько признак встречается и среднее значение целевого
pd.concat([t1, t2, t3], axis=1, keys=['train-1','train-2','test'])
```

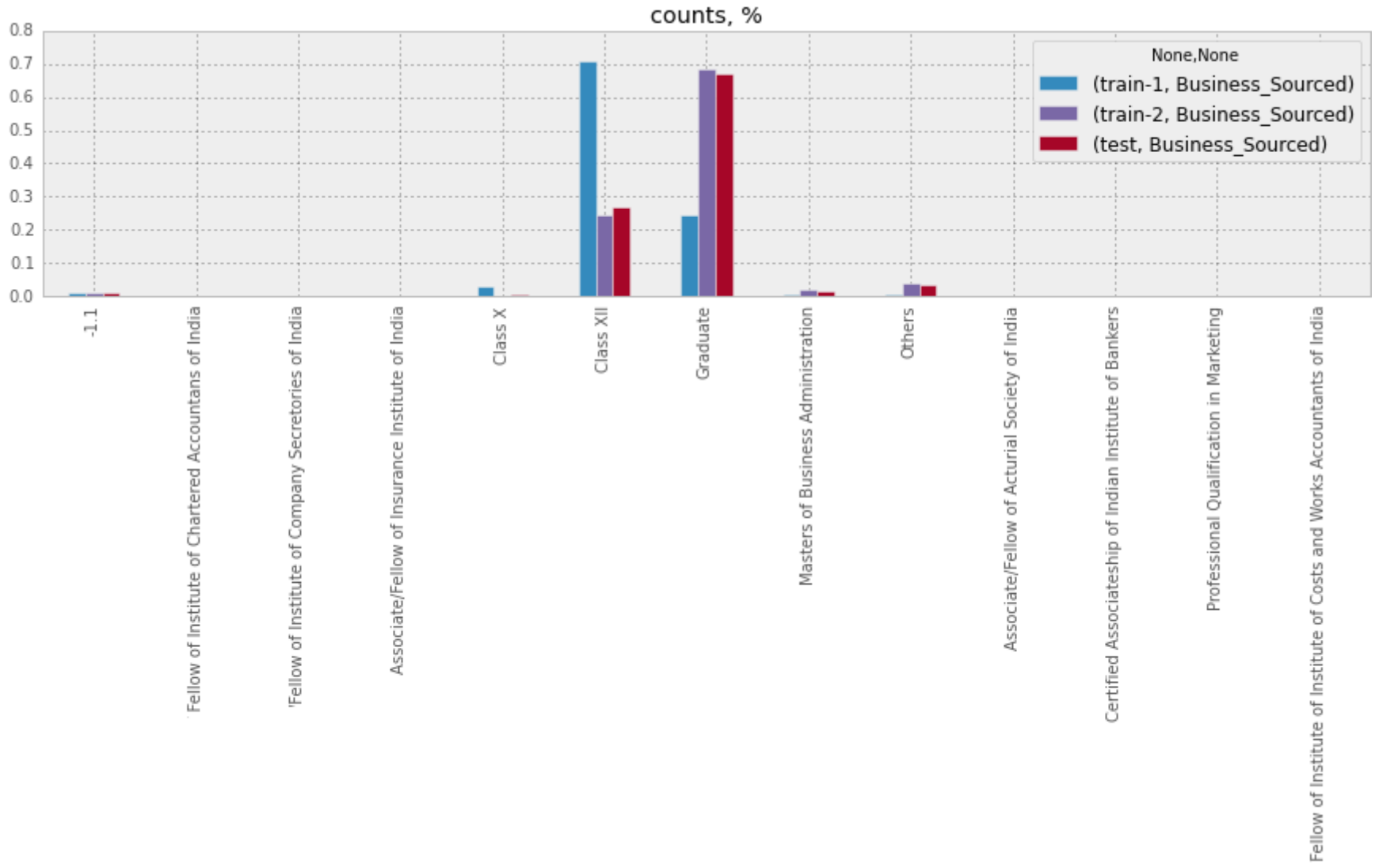
	train-1		train-2		test	
	size	mean	size	mean	size	mean
-1.1	67	0.000000	19	0.000000	44	0.000000
Associate / Fellow of Institute of Chartered Accountans of India	3	0.333333	NaN	NaN	2	1.000000
Associate/Fellow of Institute of Company Secretories of India	1	0.000000	NaN	NaN	NaN	NaN
Associate/Fellow of Insurance 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

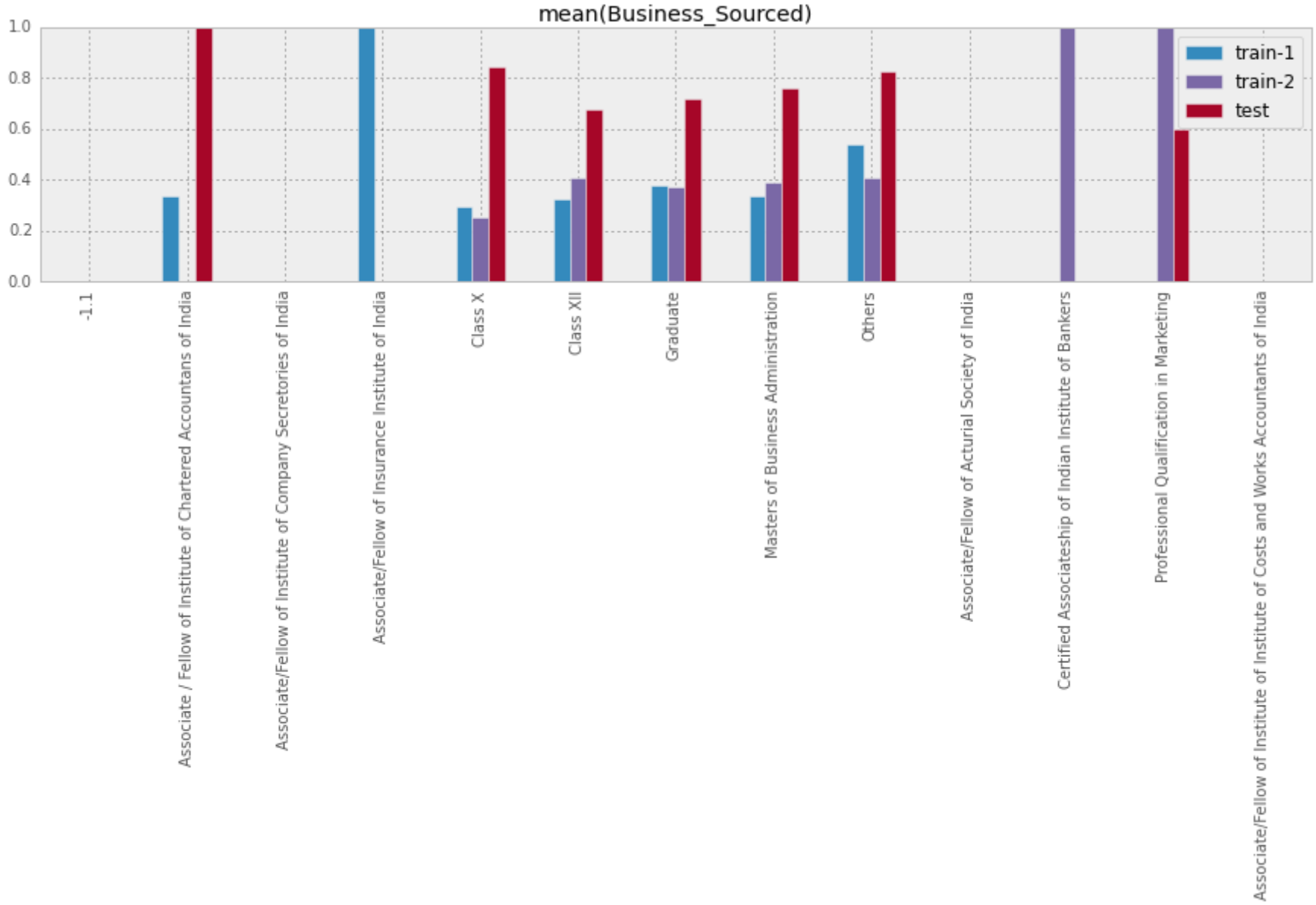
редкие признаки

нестабильность: по частотам и по цели

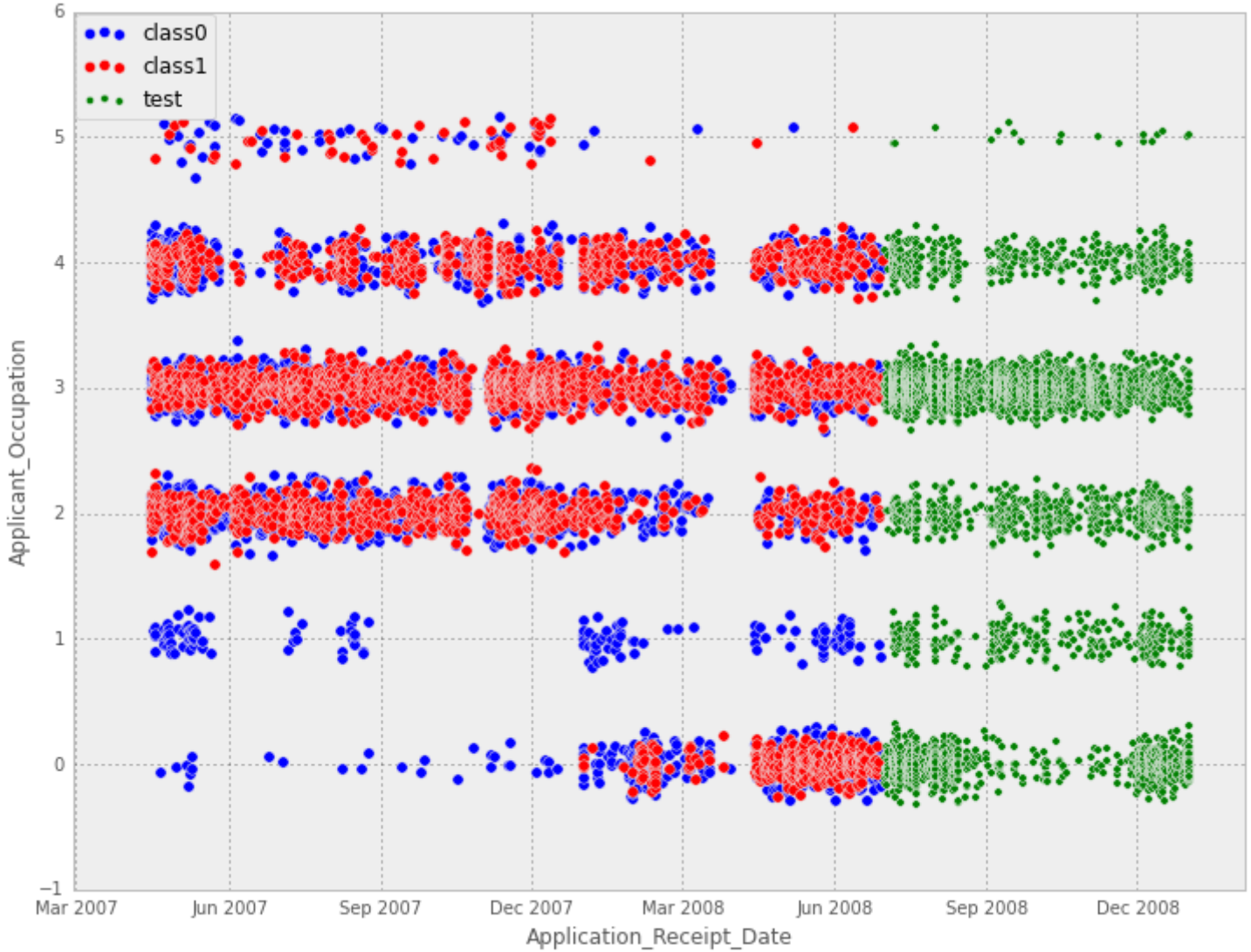
Как часто встречаются такие значения признаков



Как меняется средне значение целевой переменной



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



2.3. Смотрим, как меняются другие признаки во времени…

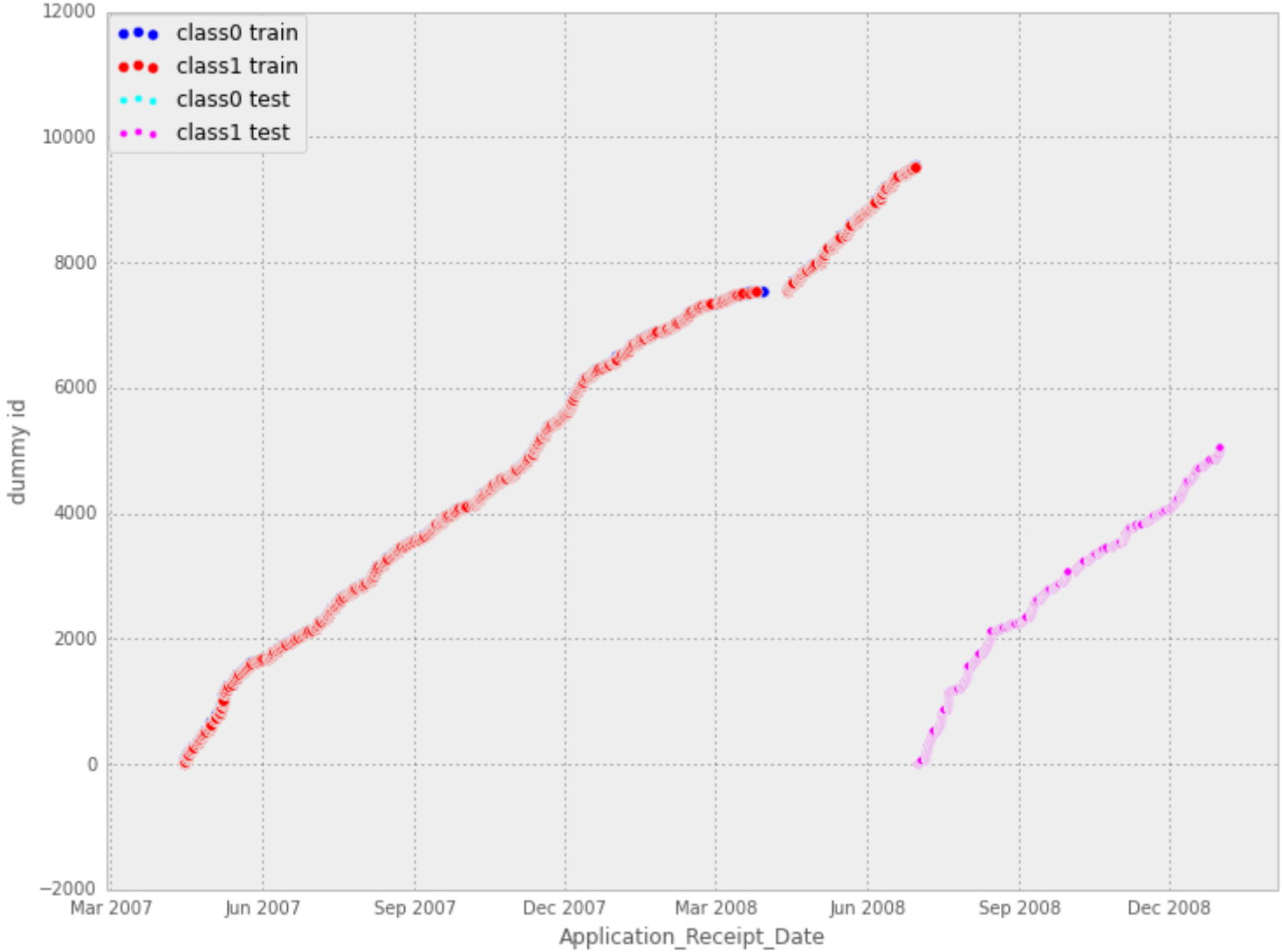
Это позволяет:

- выявить стабильные признаки

- правильно сформировать разбиения обучение / тест

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

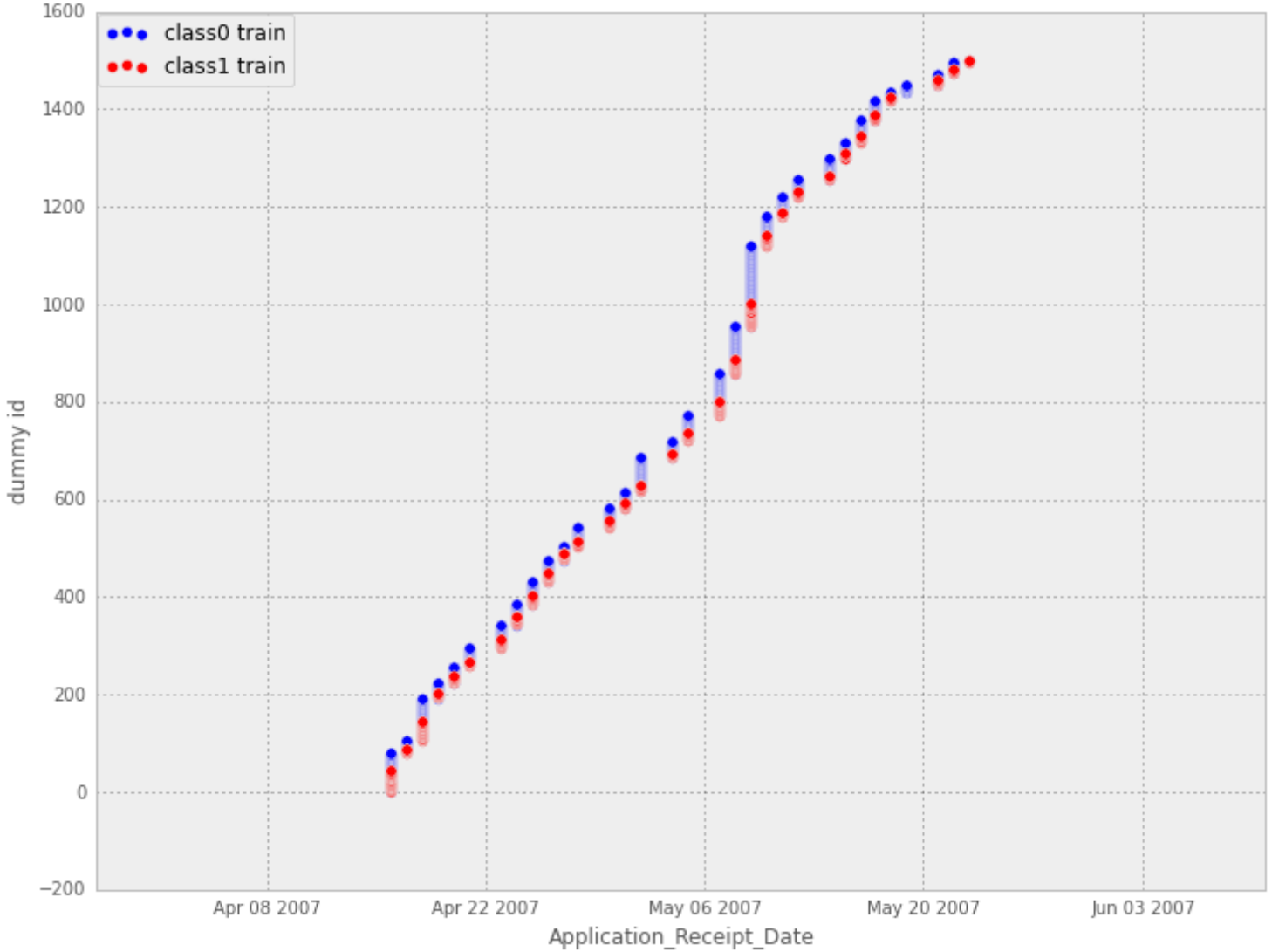
Совет: визуализация id (номер в таблице) – время



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

Если есть географические признаки

Как их использовать?

Если есть географические признаки

Проекции на разные оси

Чтобы реализовывались более сложные «поверхности разделения»

Кластеризация

Чтобы выделить отдельные регионы

Идентификация, привязка

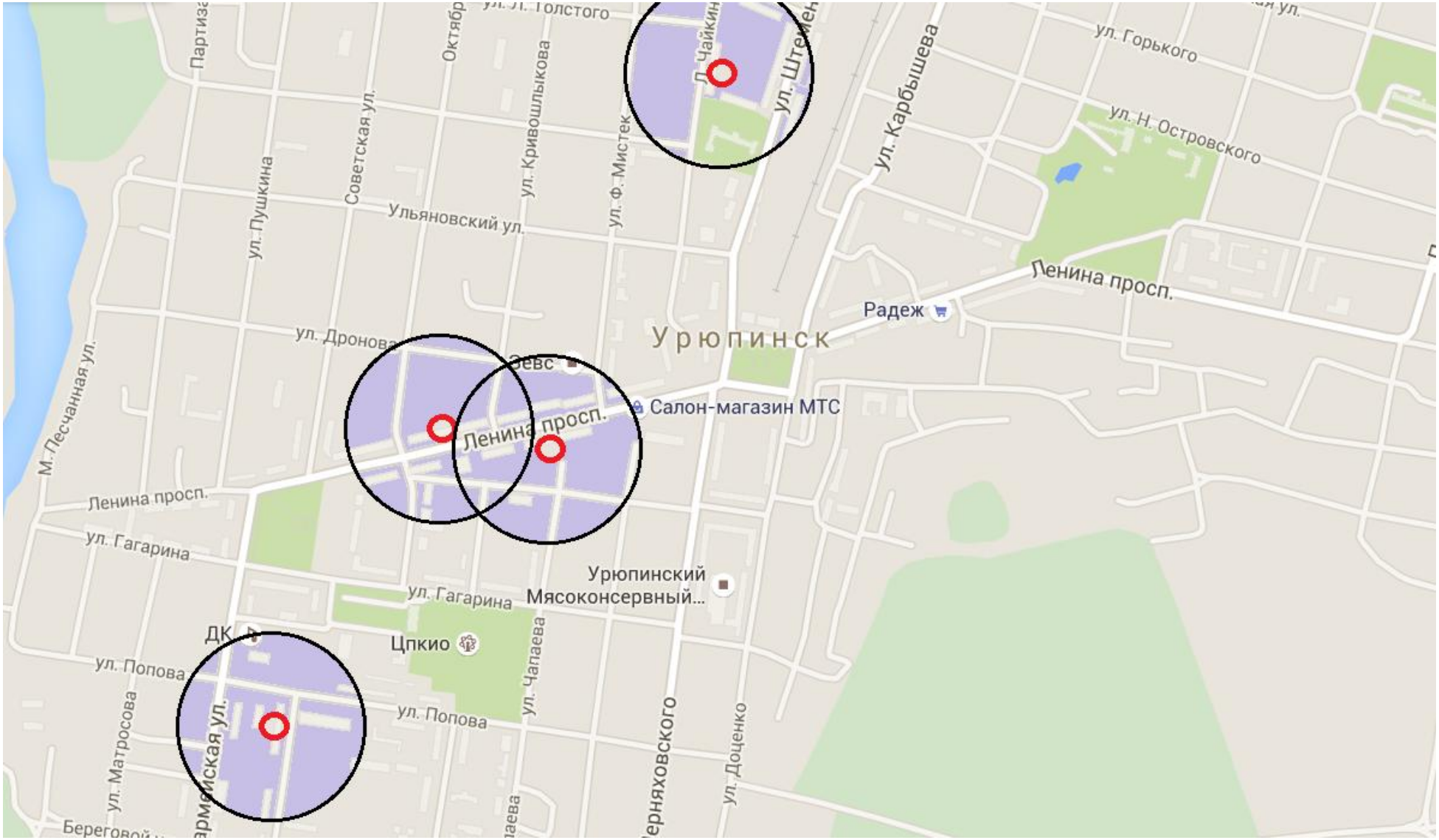
В случае точных координат:

- 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

Снова смотреть на данные!

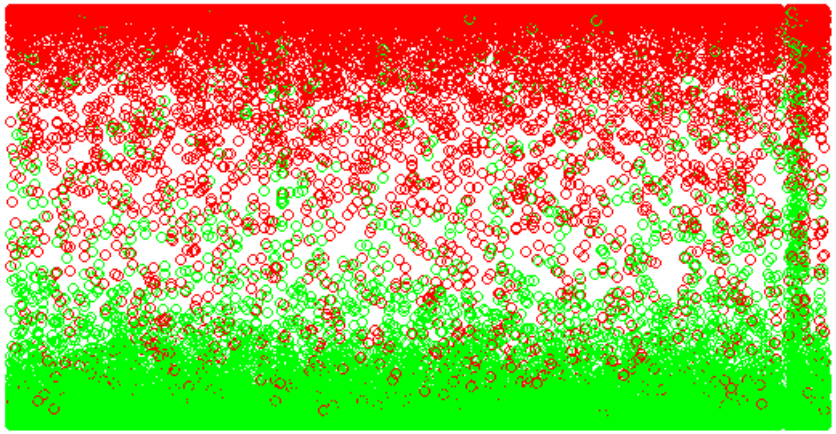
	п.їorder_id	create_time	good_id	price	utm_medium	utm_source	sessionkey_id	category_id	parent_id	root_id	model_id
1	1219899	(75-12-01 12:05:32)	9896348	666	6	4	118678774	139	133	124	123517

model_create_time	is_callcenter
1971-04-14 00:15:20.000	0

	п.їsessionkey_id	visitor_id	date_time	user_agent	duration_sec	pageviews	cartadds	internal_searches	page_type
1	118678774	1.372664e+17	1975-12-01 11:28:01.587	Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)	2403	39	6	0	2

	п.їsessionkey_id	date_time	page_type	pageview_number	pageview_duration_sec	category_id	model_id	good_id	price
24	118678774	(75-12-01 11:58:52)	1	27	6	139	123517	9896348	666
25	118678774	(75-12-01 11:58:59)	3	28	10	NULL	NULL	NULL	NULL
26	118678774	(75-12-01 11:59:08)	1	29	6	139	123517	9896348	666
27	118678774	(75-12-01 11:59:15)	3	30	5	NULL	NULL	NULL	NULL
28	118678774	(75-12-01 12:06:10)	3	35	9	NULL	NULL	NULL	NULL
29	118678774	(75-12-01 12:06:20)	3	36	8	NULL	NULL	NULL	NULL

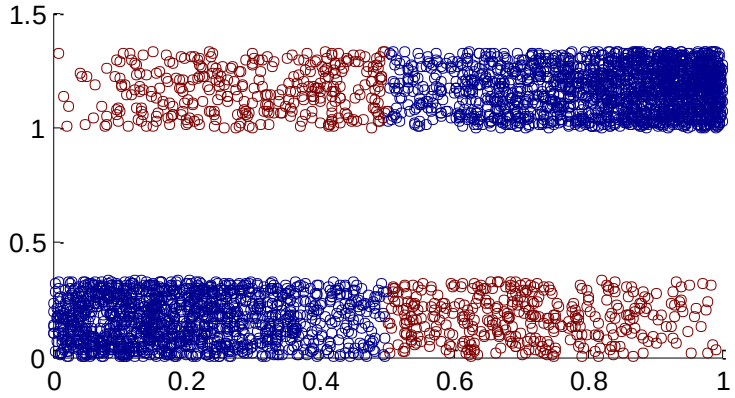
Как догадаться до новых признаков?



признак – ответы алгоритма

Что видно?

Как догадаться до новых признаков?



ответы алгоритма – целевой признак

```
plot(D[,1], a, col=c(rgb(0,1,0),rgb(1,0,0))[D$is_callcenter+1])
which((a<0.5)&(D$is_callcenter==1))[1:10]
```

детальный просмотр – где ошибается алгоритм!

(находим объекты с большой ошибкой и просматриваем)

«Ручной бустинг»