



Прикладные задачи анализа данных

**ГЕНЕРАЦИЯ ПРИЗНАКОВ
(FEATURE ENGINEERING /
CONSTRUCTION)**

Дьяконов А.Г.

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

«Applied machine learning» is basically feature engineering.

Andrew Ng

Генерация признаков (feature engineering / construction) – процесс придумывания способов описания данных с помощью простых значений, которые должны отражать характеристики объектов исследований, через которые могут выражаться целевые значения.

Изначально объекты могут быть заданы непризнаковым описанием:

- измерения
- веб-страницы
- файлы
- участники соцсети
- и т.д.

Важно понимать

**Процесс создания признакового пространства зависит от модели,
которую будем использовать**

**ONE-кодирование предпочтительнее для линейных моделей,
умное кодирование категорий – для деревьев
Выбросы можно не удалять для робастной модели
(– и этапы предобработки данных тоже!)**

Следует использовать:

- **контекст (знание предметной области)**
 - **EDA**

Признаки (Features)

Признак – функция на множестве объектов

$$f : X \rightarrow A$$

Признак пол

Клиенты $\rightarrow \{М, Ж\}$

Признак доход

Клиенты $\rightarrow \{..., 10\ 000, 20\ 000, ..., \text{NA}\}$

**Значения признака могут быть не определены
это тоже важная информация!**

**Некоторые значения можно восстановить по другим признакам
(например, пол)**

Типы числовых признаков

- **вещественными (+ временными)**
 - интервальные (Interval)
 - относительные (Ration)
- **категориальными**
 - неупорядоченные категориальные (номинальные – Nominal, факторные)
 - порядковые (Ordinal) или упорядоченные категориальные

отдельно отметим: дискретные, бинарные

Типы числовых признаков

| Тип признака | Операции | Трансформации | Примеры |
|----------------------|---|------------------------------|---|
| номинальные | $==$ перестановки mode, entropy, contingency, correlation, X2-test | перестановка | ID, пол, цвет, профессия |
| порядковые | $>$ median, percentiles, rank correlation, run tests, sign tests | Монотонное преобразование | оценка, рейтинг, место в соревновании |
| интервальные | $+$, $-$ mean, standard deviation, Pearson's correlation, t and F tests | $A * X + B$ | дата, температура по Цельсию |
| относительные | $*$, $/$ geometric mean, harmonic mean, percent variation | $A * X$ | возраст, масса, длина, цена, температура по Кельвину |

Категориальные

Неупорядоченные



Жанры фильмов, имена актёров,
категории товаров

Порядковые

APPAREL

| WEIGHT | BACK LENGTH | CHEST GIRTH | SIZE |
|---------------|-------------|-------------|------|
| Up to 8 lbs | 6-9 | 10-13 | XS |
| Up to 15 lbs | 9-12 | 13-15 | S |
| Up to 25 lbs | 14-18 | 15-20 | M |
| Up to 40 lbs | 20-24 | 20-25 | L |
| Up to 60 lbs | 22-26 | 25-30 | XL |
| Up to 120 lbs | 24-28 | 30-35 | XXL |

COLLARS & BANDANAS

| NECK | SIZE |
|-------|------|
| 6-12 | S |
| 10-16 | M |
| 14-24 | L |



Часто «кольцевой порядок»:
Времена года, дни недели, час

Простые типы нечисловых признаков

- временные отметки (**даты там**)
- множества, наборы (ex: пары координат)
- строки

| | дата | пол | образование | сумма | число просрочек | платёжная строка |
|---|------------|-----|-------------|---------|-----------------|------------------|
| 0 | 12/01/2017 | 1 | высшее | 5000.0 | 0 | 0000 |
| 1 | 13/01/2017 | 1 | высшее | 2500.0 | 1 | 001000 |
| 2 | 13/01/2017 | 0 | | 13675.0 | 3 | 111 |
| 3 | 25/01/2017 | 0 | начальное | NaN | 0 | 0 |

Признаки

- исходные (raw)
- сгенерированные / производные (derived)

пример: возраст = текущая дата – дата рождения

Дальнейший план

Работа с контекстными признаками **Часть EDA**

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 |
|---|-------|-------|-----------|-----------|
| 0 | 150 | 1100 | 150 | 110 |
| 1 | 11 | 70 | 110 | 70 |
| 2 | 12 | 80 | 120 | 80 |
| 3 | 11 | 570 | 115 | 70 |
| 4 | 1 | 2080 | 120 | 80 |

Пример: диаметр зрачка

Контекстные признаки

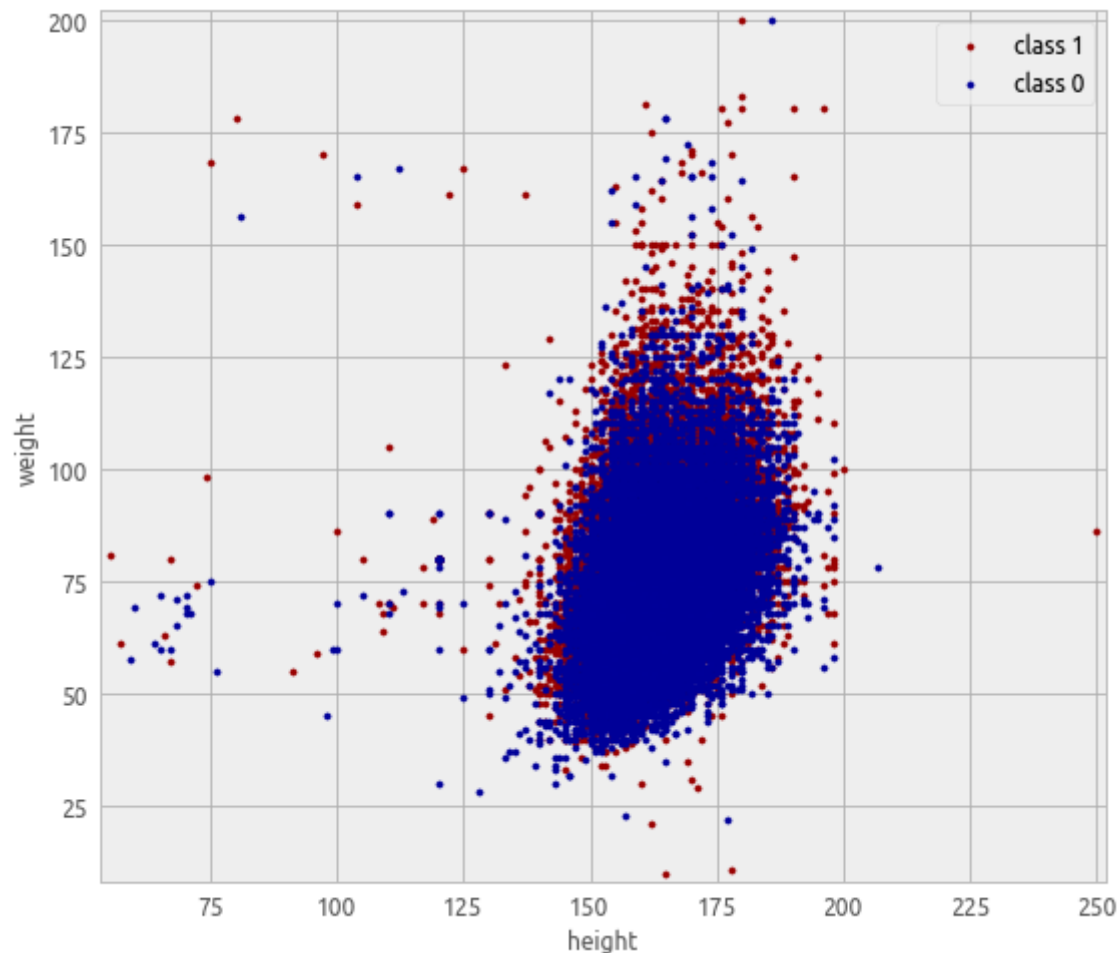
| Признак | Гипотеза |
|--|--|
| Число кликов | Максимальна в рабочие дни, в дневные часы |
| Уровень дохода | Унимодальное распределение, значения положительные |
| Температура | Лежит на отрезке [36, 42] |
| Признаки «верхн_XXX», «нижн_XXX» | $\text{«верхн_XXX»} \geq \text{«нижн_XXX»}$ |

Почему плохо решать задачи без знания предметной области...

- Gett в Израиле
- Хакатон Газпромнефти (второе место из Уфы)
- Пример решения Семёнова для бозона Хигса

Контекстные признаки

Диаграмма рассеивания пар контекстных признаков



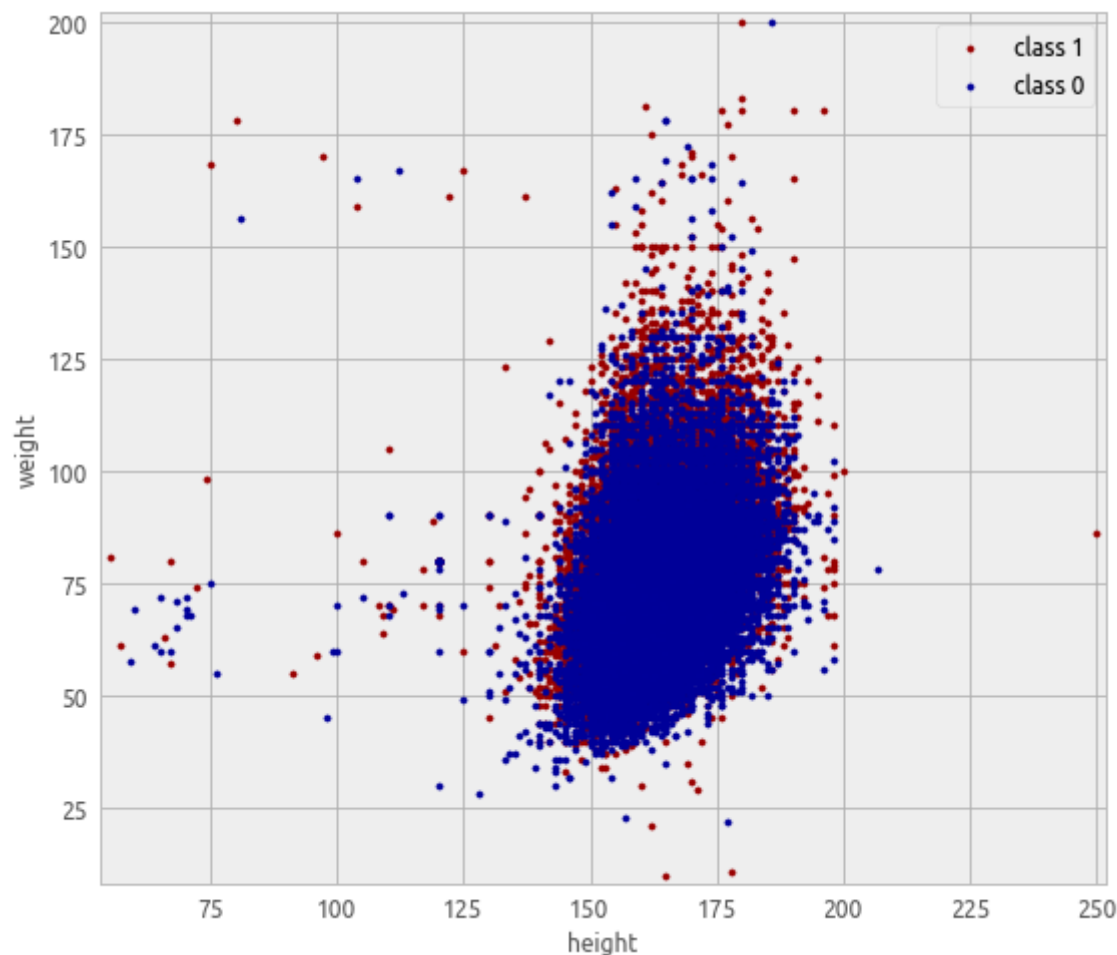
- **аномальные зоны**
- **концентрации значений**

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

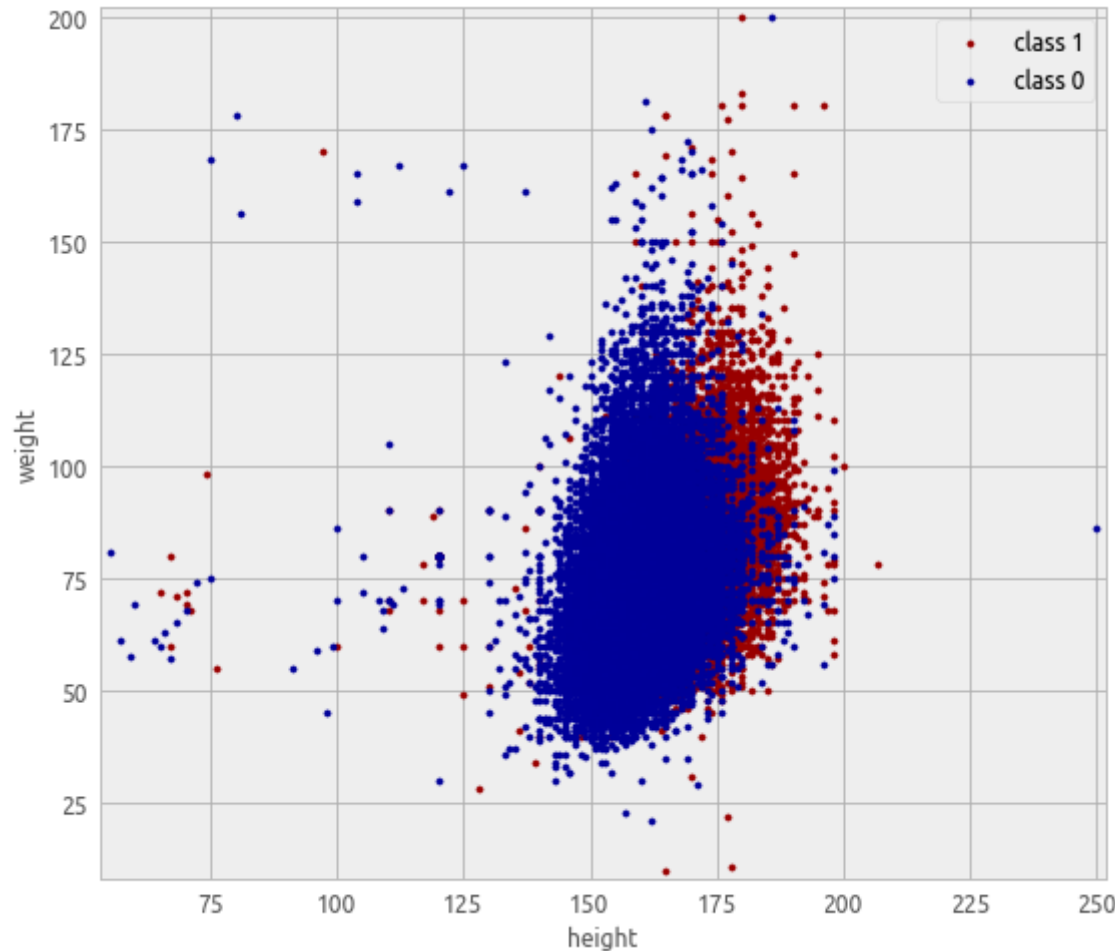
новые признаки

- 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**

Служебные признаки

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

- **номер строки** (а также производные признаки, например, чётность номера строки)
- **номер объекта в какой-то внутренней нумерации** (например, id объекта), производные признаки от этого номера
- **порция данных** (если датасет разбит на несколько частей, например, train / test / valid)
- **константный признак**
- **характеристические признаки** (выполняется ли какое-то свойство)
- **характеристика особенностей данных** (например, число пропусков на объекте)

Казалось бы, логично не рассматривать такие признаки...

Служебные признаки

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

| | Пол | Рост | Вес |
|----|-----|-------|------|
| 1 | М | 170.0 | 80.0 |
| 20 | Ж | NaN | 70.0 |
| 23 | М | 167.0 | 75.0 |
| 33 | М | NaN | NaN |
| 40 | Ж | 180.0 | 65.0 |

| | index | Пол | Рост | Вес | #nan | Пол=M |
|---|-------|-----|-------|------|------|-------|
| 0 | 1 | М | 170.0 | 80.0 | 0 | 0 |
| 1 | 20 | Ж | NaN | 70.0 | 1 | 0 |
| 2 | 23 | М | 167.0 | 75.0 | 0 | 1 |
| 3 | 33 | М | NaN | NaN | 2 | 1 |
| 4 | 40 | Ж | 180.0 | 65.0 | 0 | 0 |

```
data.reset_index(inplace=True)
data['#nan']=data.isnull().sum(axis=1)
data['Пол=M'] = (data['Пол'] == 'М').astype(int)
data['Рост2'] = data['Рост'] / 10
data['Рост2'] = data['Рост'] - 10*np.floor(data['Рост2'].values)
```

Д3 код можно улучшать

Служебные признаки

Когда такие признаки важны:

- **особенность значения \Leftarrow особенность объекта**
«круглый доход» – не знает точного значения
«круглый рост, вес, давление» – не знает точного

- **улучшает качество**
«есть ли пропуск в признаке NAME», «сколько пропусков / аномальных значений»

- **поиск утечек в данных**

- **организация эксперимента**
(разбиение на корзины и т.п.)

Пример: области значений целевого (маленькие, средние, большие) + StratifiedKFold

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 |

Виды утечек

1) зависимость от служебных признаков, в том числе

- от порядка (ех: номера строки)
- от организации данных (от каталога)
- от способа представления (названия файлов, времени его создания и т.п.)
- от особенностей (наличия пропусков, дубликатов и т.п.)

2) неявное использование информации

- о целевом векторе
- из будущего или настоящего

**3) содержание ответа в исходных данных
как правило, из-за заглядывания в будущее
(пример про номер страницы и число страниц в сессии)**

Странности в данных

| | ld | 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?

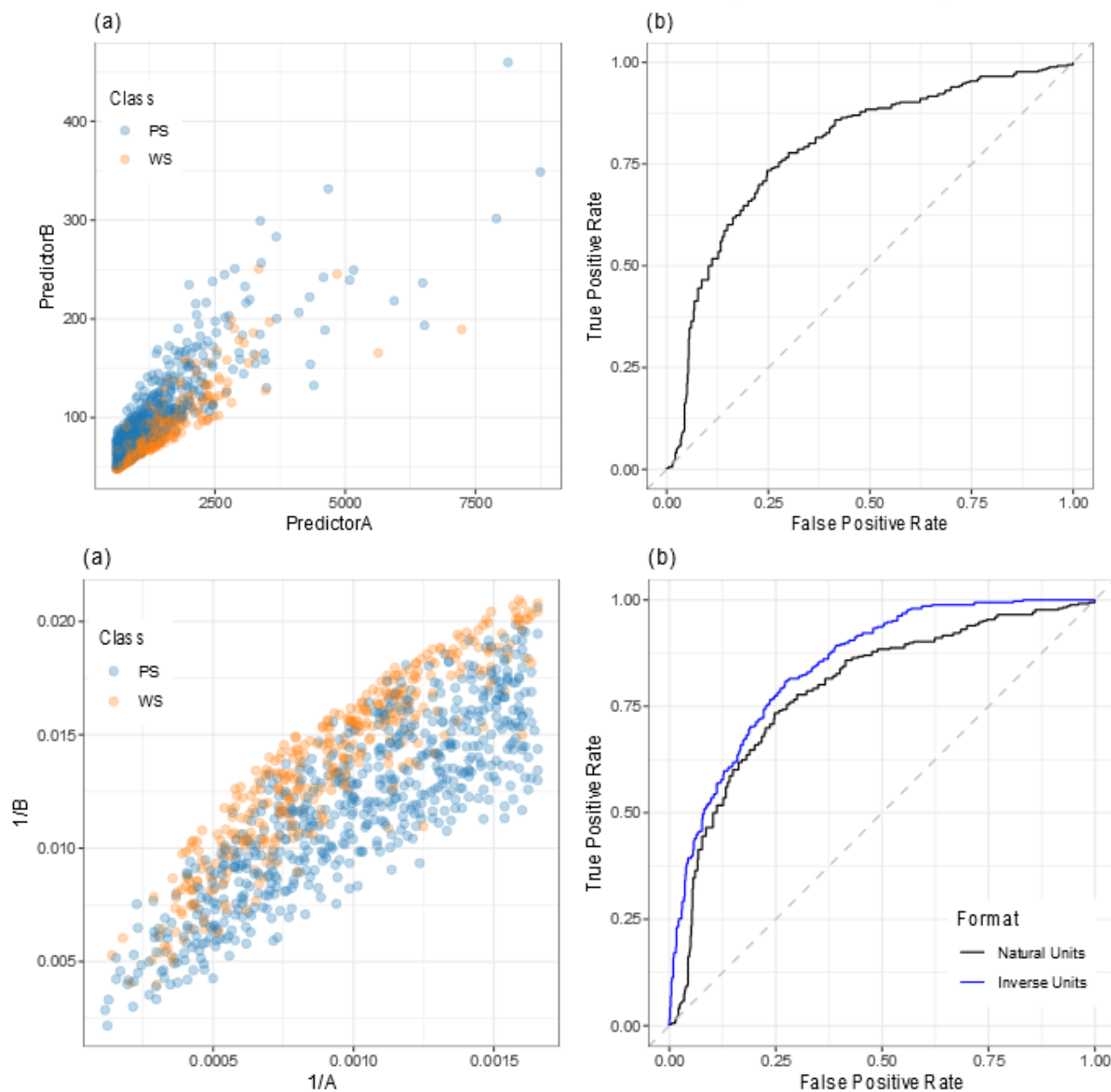
Странности в данных

| | ld | 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

Использование EDA для генерации признаков



Использование EDA для генерации признаков

**использование обратных признаков
повышает качество линейной модели**

<http://www.feat.engineering/intro-intro.html>

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 | sex | income |
|---|--------|-----|--------|
| 0 | Moscow | M | 110 |
| 1 | London | F | 200 |
| 2 | London | M | 140 |
| 3 | Paris | M | 120 |
| 4 | Moscow | F | 190 |

city строка, мало уникальных
sex строка, мало уникальных

```
def find_cat(data):  
    """  
    найти все признаки,  
    в которых первое значение – строка  
    и / или мало значений  
    """  
    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)
```

Категориальные признаки

2. Создание новых категориальных признаков

- конъюнкция признаков

| | city | sex | income | city + sex |
|---|--------|-----|--------|------------|
| 0 | Moscow | M | 110 | Moscow + M |
| 1 | London | F | 200 | London + F |
| 2 | London | M | 140 | London + M |
| 3 | Paris | M | 120 | Paris + M |
| 4 | Moscow | F | 190 | Moscow + F |

```
def make_conj(data, feature1, feature2):  
    """  
    конъюнкция двух признаков  
    """  
    data[feature1 + ' + ' + feature2] =  
        data[feature1].astype(str) + ' + ' +  
        data[feature2].astype(str)  
    return (data)  
  
# пример использования  
make_conj(data, 'city', 'sex')
```

Категориальные признаки

2. Создание новых категориальных признаков

- **конъюнкция признаков**

может быть очень полезно: kNN, линейные алгоритмы

- **создание новых признаков по контекстным**

Пример: верхние уровни иерархии

Категориальные признаки

3. Простейшее кодирование – по номеру категории **Label Encoding**

- **Лексикографический порядок (sklearn)**
`sklearn.preprocessing.LabelEncoder`

- **В порядке появления (pandas)**
`pandas.factorize`

Хорошая идея! Можно использовать сортировки по признакам
⇒ по порядку индуцированным каким-то признаком

- **Случайное кодирование**
`dict + map`

многократное случайное кодирование иногда хорошо работает с RF

Категориальные признаки

| | city | sex | income | city_le | city_fz | city_rnd | sex_le | sex_fz | sex_rnd |
|---|--------|-----|--------|---------|---------|----------|--------|--------|---------|
| 0 | Moscow | M | 110 | 1 | 0 | 0.63 | 1 | 0 | 0.22 |
| 1 | London | F | 200 | 0 | 1 | 0.75 | 0 | 1 | 0.20 |
| 2 | London | M | 140 | 0 | 1 | 0.75 | 1 | 0 | 0.22 |
| 3 | Paris | M | 120 | 2 | 2 | 0.50 | 1 | 0 | 0.22 |
| 4 | Moscow | F | 190 | 1 | 0 | 0.63 | 0 | 1 | 0.20 |

```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
```

```
for name in cols:
    # лексикографический порядок
    data[name + '_le'] = le.fit_transform(data[name])
    # в порядке упоминания
    data[name + '_fz'] = pd.factorize(data[name])[0]
    # случайно
    dct = dict(zip(data[name].unique(),
                   np.random.rand(data[name].nunique()).round(2)))
    data[name + '_rnd'] = data[name].map(dct)
```

Категориальные признаки

3. Простейшее кодирование – по номеру категории **Label Encoding**

- не подходит для линейных алгоритмов
- проблема новых категорий
(средним)

Категориальные признаки

4. Dummy-кодирование / One-hot-encoding

| | city | sex | income | city=Moscow | city=London | city=Paris | sex=M | sex=F |
|---|--------|-----|--------|-------------|-------------|------------|-------|-------|
| 0 | Moscow | M | 110 | 1 | 0 | 0 | 1 | 0 |
| 1 | London | F | 200 | 0 | 1 | 0 | 0 | 1 |
| 2 | London | M | 140 | 0 | 1 | 0 | 1 | 0 |
| 3 | Paris | M | 120 | 0 | 0 | 1 | 1 | 0 |
| 4 | Moscow | F | 190 | 1 | 0 | 0 | 0 | 1 |

Мнение: OHE $\rightarrow m$, dummy $\rightarrow m - 1$
drop_first=True в get_dummies
(теперь False)

Категориальные признаки

OneHotEncoder (по умолчанию sparse, раньше –только с числами)

```
from sklearn import preprocessing
ohe = preprocessing.OneHotEncoder(sparse=False)
tmp = ohe.fit_transform(data[cols]).astype(int)
tmp = pd.DataFrame(tmp,
                    columns=['OHE_' + str(i) for i in range(tmp.shape[1])])
data = pd.concat([data, tmp], axis=1)
```

Ручное решение

```
def code_myoe(data, feature):
    """
    ручной способ ONE
    """
    for i in data[feature].unique():
        data[feature + '=' + i] = (data[feature] == i).astype(int)

for name in cols:
    code_myoe(data, name)
```

Самый простой способ

```
data = create_data(1)
pd.get_dummies(data)
```

Категориальные признаки

ONE

- Хороши для линейных алгоритмов
можно кодировать $N-1$ категорию
- Все признаки в одной шкале (на $[0, 1]$)
но есть тонкость, что категории разной мощности
- Большое число категорий → сильно разреженные матрицы
часто используют sparse-формат
- Плохо, что после ONE значительная часть признаков – бинарные – описывают категориальные зависимости
некоторые алгоритмы (RF) могут терять качество

Проблема мелких и новых категорий

– возникает почти для всех способов кодирования

Часто: мелкие категории → в одну

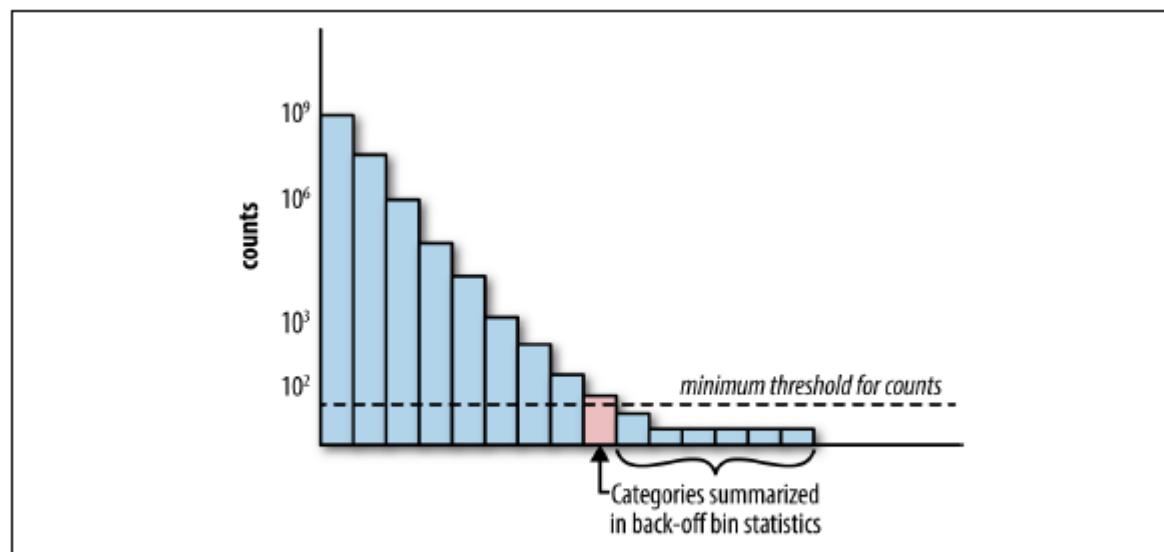


Figure 5-3. If a rare category gains counts, it can move above the threshold for the back-off bin, using its own count statistics for modeling

Alice Zheng, Amanda Casari Feature Engineering for Machine Learning, Principles and Techniques for Data Scientists // O'Reilly Media, 2018, pp. 218

Категориальные признаки

5. По значениям вещественного признака

| | city | sex | income | city_mean_income | sex_mean_income |
|---|--------|-----|--------|------------------|-----------------|
| 0 | Moscow | M | 110 | 150 | 123.3 |
| 1 | London | F | 200 | 170 | 195.0 |
| 2 | London | M | 140 | 170 | 123.3 |
| 3 | Paris | M | 120 | 120 | 123.3 |
| 4 | Moscow | F | 190 | 150 | 195.0 |

```
def code_mean(data, cat_feature, real_feature):  
    """  
    кодирование средним значением  
    """  
    mn = data.groupby(cat_feature)[real_feature].mean()  
    return data[cat_feature].map(mn)  
  
for name in cols:  
    data[name + '_mean_income'] = code_mean(data, name, 'income')
```

Категориальные признаки

5. По значениям вещественного признака

- **Естественная интерпретация:**
товары какой категории дороже
- **можно использовать другие статистики**
они не всегда логичны и лучше интерпретируются
- **можно кодировать по разным признакам**

Ниже отдельно рассмотрим случай кодирования по целевому признаку

Категориальные признаки

6. По значениям категориального признака

- просто по мощности **Count Encoding**
- по частоте **Frequency Encoding**

•

| | city | sex | income | city_vc | sex_vc | city_vcn | sex_vcn |
|---|--------|-----|--------|---------|--------|----------|---------|
| 0 | Moscow | M | 110 | 2 | 3 | 0.4 | 0.6 |
| 1 | London | F | 200 | 2 | 2 | 0.4 | 0.4 |
| 2 | London | M | 140 | 2 | 3 | 0.4 | 0.6 |
| 3 | Paris | M | 120 | 1 | 3 | 0.2 | 0.6 |
| 4 | Moscow | F | 190 | 2 | 2 | 0.4 | 0.4 |

```
data[name + '_vc'] = data[name].map(data[name].value_counts())
data[name + '_vcn'] =
    data[name].map(data[name].value_counts(normalize=True))

# другой способ
for name in cols:
    data[name + '_vc'] = data[name].map(data.groupby(name).size())
```

Категориальные признаки

Недостатки Count Encoding

- **коллизии (несколько категорий – один код)**
добавляем шум (не всегда это проблема)
- **проблема шума (мелкие категории)**
мелкие категории объединяем в одну, и новые!
- **проблема новых категорий**
см. выше
- **проблема утечки информации**
честно: без заглядывания в будущее
это не страшная утечка (допустима в соревнованиях)

Категориальные признаки

6. По значениям ДРУГОГО категориального признака

| | city | sex | income | city_svd | sex_svd |
|---|--------|-----|--------|----------|---------|
| 0 | Moscow | M | 110 | -0.66 | -0.79 |
| 1 | London | F | 200 | -0.66 | -0.62 |
| 2 | London | M | 140 | -0.66 | -0.79 |
| 3 | Paris | M | 120 | -0.37 | -0.79 |
| 4 | Moscow | F | 190 | -0.66 | -0.62 |

```
pd.crosstab(data['city'],
            data['sex'])
```

| | sex | |
|--------|-----|---|
| | F | M |
| city | | |
| <hr/> | | |
| London | 1 | 1 |
| Moscow | 1 | 1 |
| Paris | 0 | 1 |

```
from numpy.linalg import svd
```

```
def code_factor(data, cat_feature, cat_feature2):
```

```
    """
```

```
    кодирование на основе другого категориального
```

```
    """
```

```
    ct = pd.crosstab(data[cat_feature], data[cat_feature2])
```

```
    u, __, __ = svd(ct.values)
```

```
    coder = dict(zip(ct.index, u[:,0])) # е. кодировать первой компонентой
```

```
    return (data[cat_feature].map(coder))
```

Категориальные признаки

6. По значениям ДРУГОГО категориального признака

```
data['city_svd'] = code_factor(data, 'city', 'sex')  
data['sex_svd'] = code_factor(data, 'sex', 'city')
```

$$X = \| x_{ij} \| = U \Lambda V^T$$

$$\Lambda' = \text{diag}(\lambda_1, \dots, \lambda_r, 0, \dots, 0)$$

$$U \Lambda' V^T = \sum_{i=1}^r \lambda_i u_i v_i^T \equiv X'$$

x_{ij} – сколько i -я категория встречается с j -й

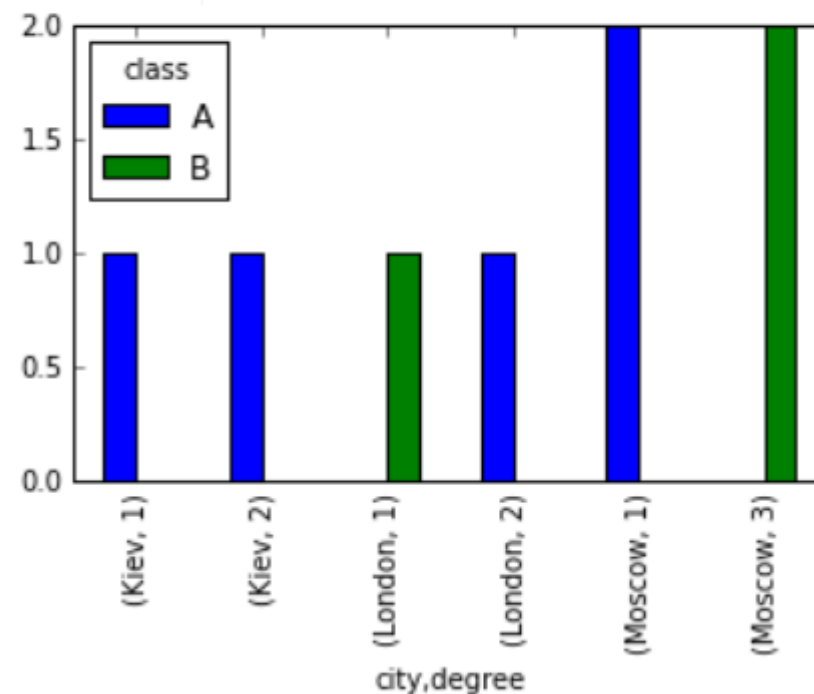
Дьяконов А. Г. Методы решения задач классификации с категориальными признаками // Прикладная математика и информатика. Труды факультета Вычислительной математики и кибернетики МГУ имени М.В. Ломоносова. — 2014. — № 46. — С. 103–127.

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 быть коллизии (можно выполнять разные хэш-кодирования)

| | city | sex | income | city_0 | city_1 | sex_0 | sex_1 |
|---|--------|-----|--------|--------|--------|-------|-------|
| 0 | Moscow | M | 110 | 1.0 | -1.0 | 1.0 | 0.0 |
| 1 | London | F | 200 | -2.0 | 2.0 | -1.0 | 0.0 |
| 2 | London | M | 140 | -2.0 | 2.0 | 1.0 | 0.0 |
| 3 | Paris | M | 120 | 1.0 | -2.0 | 1.0 | 0.0 |
| 4 | Moscow | F | 190 | 1.0 | -1.0 | -1.0 | 0.0 |

```
from sklearn.feature_extraction import FeatureHasher
fh = FeatureHasher(n_features=2, input_type='string')
for name in cols:
    tmp = fh.fit_transform(data[name]).toarray()
    tmp = pd.DataFrame(tmp, columns=[name + '_' + str(i) for i in
range(tmp.shape[1])])
    data = pd.concat([data, tmp], axis=1)
```

Категориальные признаки

9. По значению целевого – **Target Encoding**

- Mean Target Encoding
- Std Target Encoding
- ...

| | city | sex | income | city_mt | sex_mt | target |
|---|--------|-----|--------|---------|--------|--------|
| 0 | Moscow | M | 110 | 0.5 | 0.67 | 1 |
| 1 | London | F | 200 | 0.5 | 0.50 | 1 |
| 2 | London | M | 140 | 0.5 | 0.67 | 0 |
| 3 | Paris | M | 120 | 1.0 | 0.67 | 1 |
| 4 | Moscow | F | 190 | 0.5 | 0.50 | 0 |

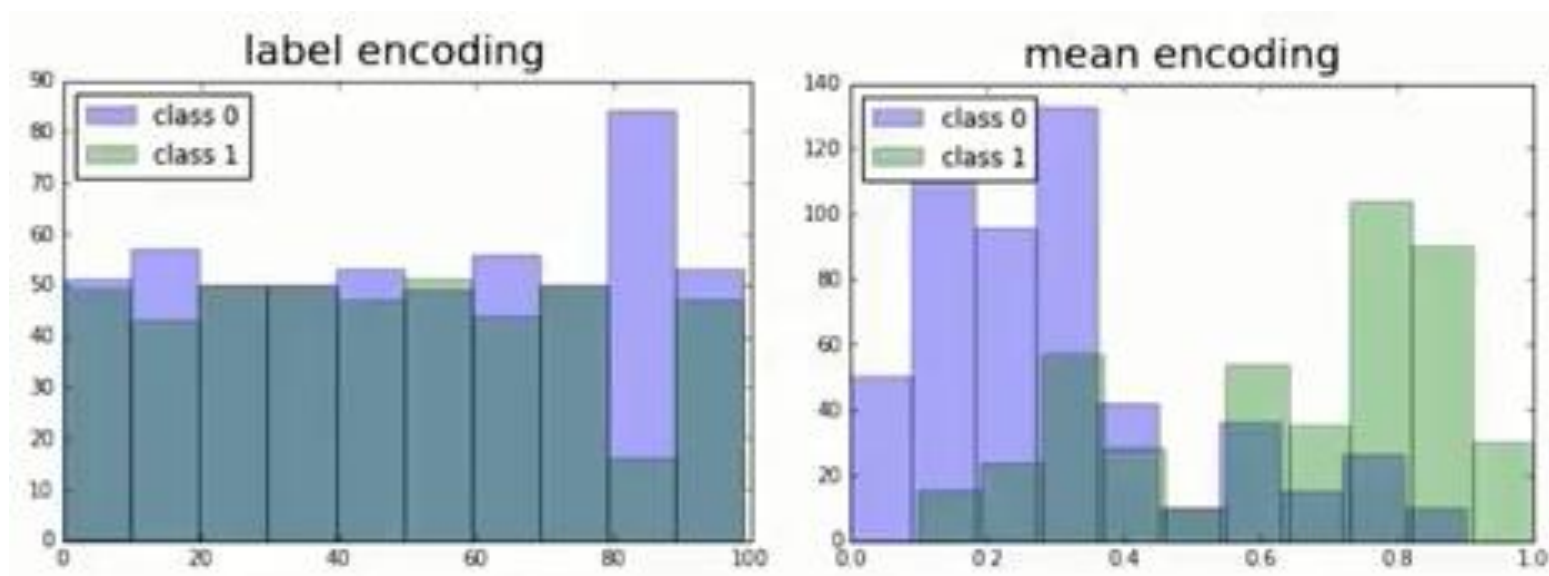
- это форма стэкинга (по одной переменной)
- подходит для любых алгоритмов (если правильно сделана)

Почему так плохо?

Категориальные признаки

9. По значению целевого – **Target Encoding**

**Почему хорошая идея –
упорядочивание категорий исходя из смысла задачи**



<https://ru.coursera.org/learn/competitive-data-science>

**Многие кодировки «случайны»,
а кодирование по значению целевого «логично»**

Категориальные признаки: Target Encoding

Нельзя допустить утечки значений целевого!
особенно проблемно для мелких категорий
способы борьбы аналогичны стекингу

- **кодирование по отложенной выборке**
– сокращаем выборку для обучения

Хороший пример: кодирование по куску испорченных данных

- **k-fold-кодировка**
разбиваем на фолды и кодируем
LOO-кодировка – проблемы утечки остаются
- **кодирование по случайным подвыборкам**
- **кодирование по предыдущим объектам (CatBoost)**

Категориальные признаки: сглаживание

добавляем среднее значение целевого с весом

$$\text{mean} = \frac{k_1 + \alpha \frac{m_1}{m}}{k + \alpha}$$

+ борьба с редкими категориями
на них оценка ненадёжна

**тонкий вопрос: как кодировать на обучении, как на тесте
обычно на тесте пересчитывают (по всему обучению)...**

Категориальные признаки: добавление шума

Категориальные признаки: почему плохо L00-кодировать

Leave-one-out

| | feature | feature_mean | target |
|---|---------|--------------|--------|
| 0 | Moscow | 0.50 | 0 |
| 1 | Moscow | 0.25 | 1 |
| 2 | Moscow | 0.25 | 1 |
| 3 | Moscow | 0.50 | 0 |
| 4 | Moscow | 0.50 | 0 |

Категориальные признаки: Кодирование по предыдущим объектам (CatBoost)

| | sex | sex_cb | target |
|---|-----|--------|--------|
| 0 | M | NaN | 1 |
| 1 | F | NaN | 1 |
| 2 | M | 1.0 | 0 |
| 3 | M | 0.5 | 1 |
| 4 | F | 1.0 | 0 |

- **есть в CatBoost**
там аналогичный приём и для вычисления градиента
- **одна категория в обучении кодируется по-разному, а на контроле фиксировано**
- **нет гиперпараметров**
(хотя напрашивается ввести)
- **можно использовать разные порядки**
~ при построении разных деревьев
случайно сортировать
порядки, индуцированные признаками

Категориальные признаки: Кодирование по предыдущим объектам (CatBoost)

```
gb = data.groupby(name)
data[name + '_cb'] =
    (gb['target'].cumsum() - data['target']) / gb.cumcount()
```

Категориальные признаки: Target Encoding

Примеры кодирования по другим статистикам для бинарного целевого вектора:

$$(y_1, \dots, y_k), k_1 = y_1 + \dots + y_k, k_0 = k - k_1$$

$$\text{mean} = \frac{k_1}{k}$$

$$\text{diff} = k_1 - k_0$$

$$\text{difflog} = \frac{\log k_1}{\log k_0}$$

$$\text{normdiff} = \frac{k_1 - k_0}{k}$$

Использование статистик напрямую не связанных со значением целевого признака (не среднее значений целевых, а дисперсию) уменьшает утечку!

Категориальные признаки: Target Encoding

На практике хороша смесь подходов!

Категориальные признаки

10. Экспертное кодирование

**Если признак порядковый, то есть естественная нумерация
Но хороша для деревьев**

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

были способы генерации признаков (в обработке данных):

- **дискретизация (binning / quantization)**
 - округление (rounding)
- **деформация (функция над признаком)**
- **нормировка (специальный вид деформации)**
- **новые признаки (функции над несколькими)**

| 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: новые признаки

- суммы групп признаков
 - мономы
- расстояние (ядро) до какого-то объекта

| | f | g | f^2 | fg | g^2 |
|---|-----|-----|-------|------|-------|
| 0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 |
| 1 | 2.0 | 2.0 | 4.0 | 4.0 | 4.0 |
| 2 | 3.0 | 1.0 | 9.0 | 3.0 | 1.0 |
| 3 | 4.0 | 1.0 | 16.0 | 4.0 | 1.0 |

```
from sklearn.preprocessing import PolynomialFeatures
```

```
pf = PolynomialFeatures(degree=2,  
                        interaction_only=False,  
                        include_bias=False)
```

```
data = pf.fit_transform(data)
```

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

```
data[name] = pd.to_datetime(data[name])
```

м.б. преобразование в вещественный признак

2. Генерация новых признаков

Каких?

Временные признаки

2.1. Характеристика момента времени:

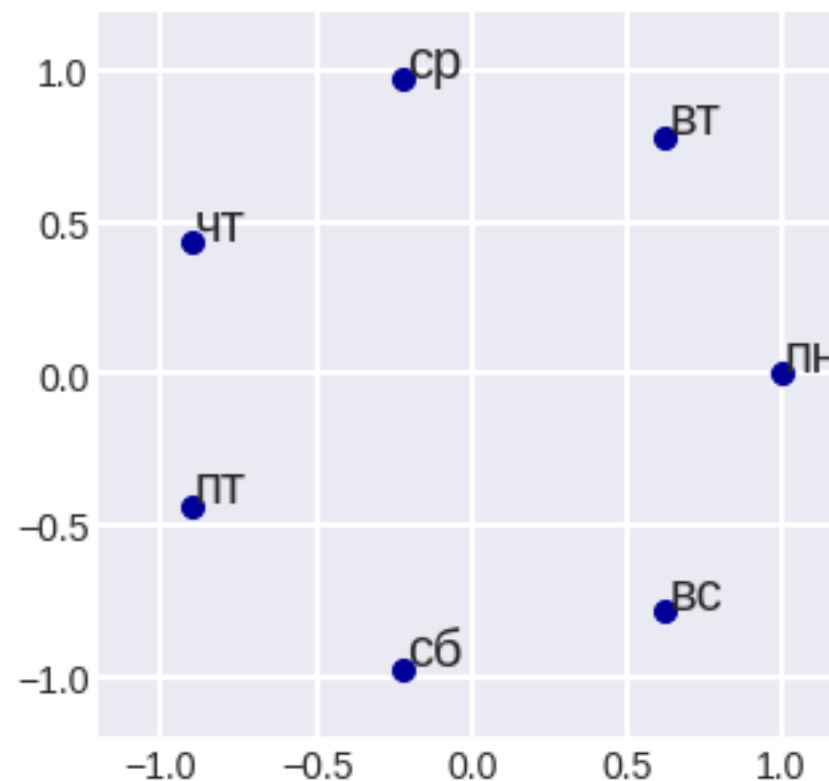
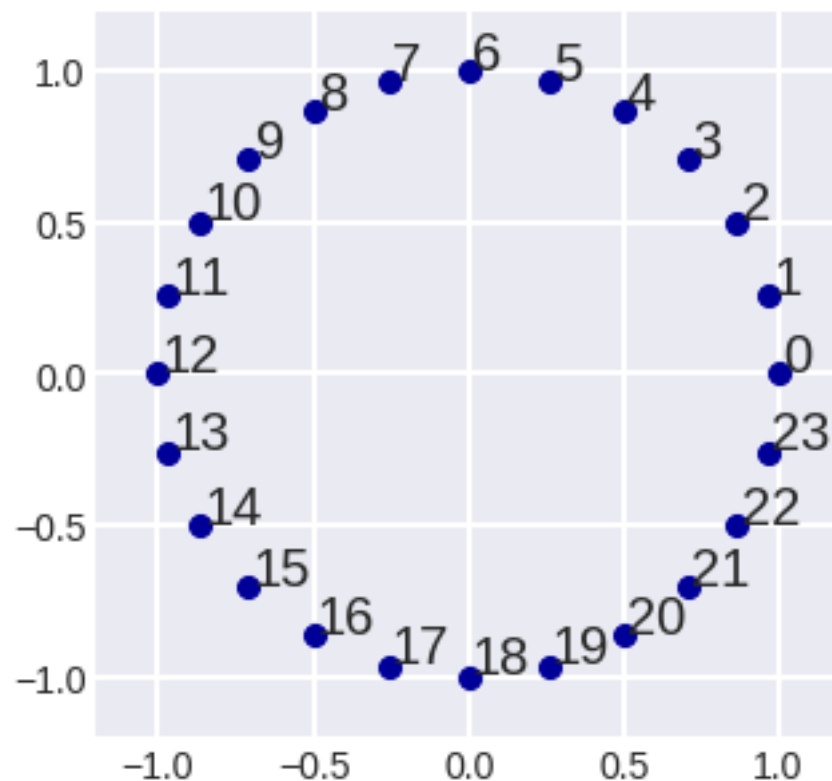
- час, минута, секунда (=0)
- время суток
- день, день недели, день года
- неделя, месяц
- время года, год
- праздник / выходной / особый день
(первый понедельник
месяца, начало Олимпиады)

```
data[name + '_day'] =  
    data[name].dt.day  
data[name + '_dayofweek'] =  
    data[name].dt.dayofweek  
data[name + '_dayofyear'] =  
    data[name].dt.dayofyear  
data[name + '_month'] =  
    data[name].dt.month
```

| | date | date_day | date_dayofweek | date_dayofyear | date_month |
|---|------------|----------|----------------|----------------|------------|
| 0 | 2017-12-10 | 10 | 6 | 344 | 12 |
| 1 | 2016-11-13 | 13 | 6 | 318 | 11 |
| 2 | 2008-01-01 | 1 | 1 | 1 | 1 |
| 3 | 2017-05-06 | 6 | 5 | 126 | 5 |

Временные признаки

Кодирование циклических признаков



```
t = np.linspace(0, 2*np.pi, 8)
x = np.cos(t)
y = np.sin(t)
```

Временные признаки

2.2. Взаимодействие пары признаков

- **разница времён**
- **в один ли день недели/год и т.п.**

м.б. другое время задано неявно:

- **близость к дедлайну ($T_{\max} - T_{\text{день рождения}}$)**
 - **возраст ($T - T_{\text{день рождения}}$)**
- **сколько после/до праздника / большой покупки / регистрации**
+ нормировать на некоторые шаблоны (max разность)

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**

Временные признаки

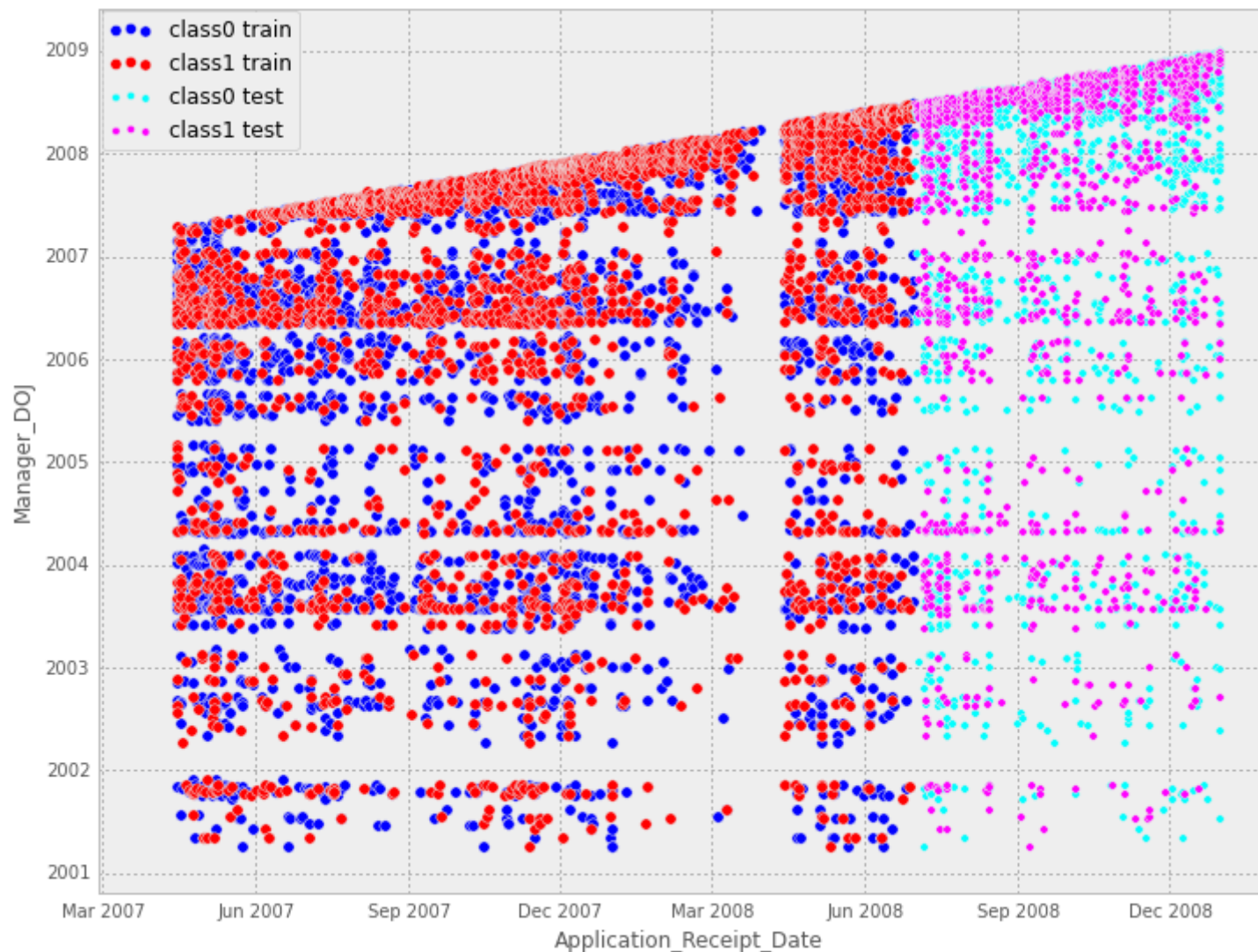
2.4. Использовать для генерации других признаков

сколько транзакций в этот день

число транзакций клиента в день / число всех транзакций

какой по счёту закрыл сделку перед концом торгов

Пример признака



День сделки / когда менеджер начал работать

Разница – опыт менеджера

Временные признаки

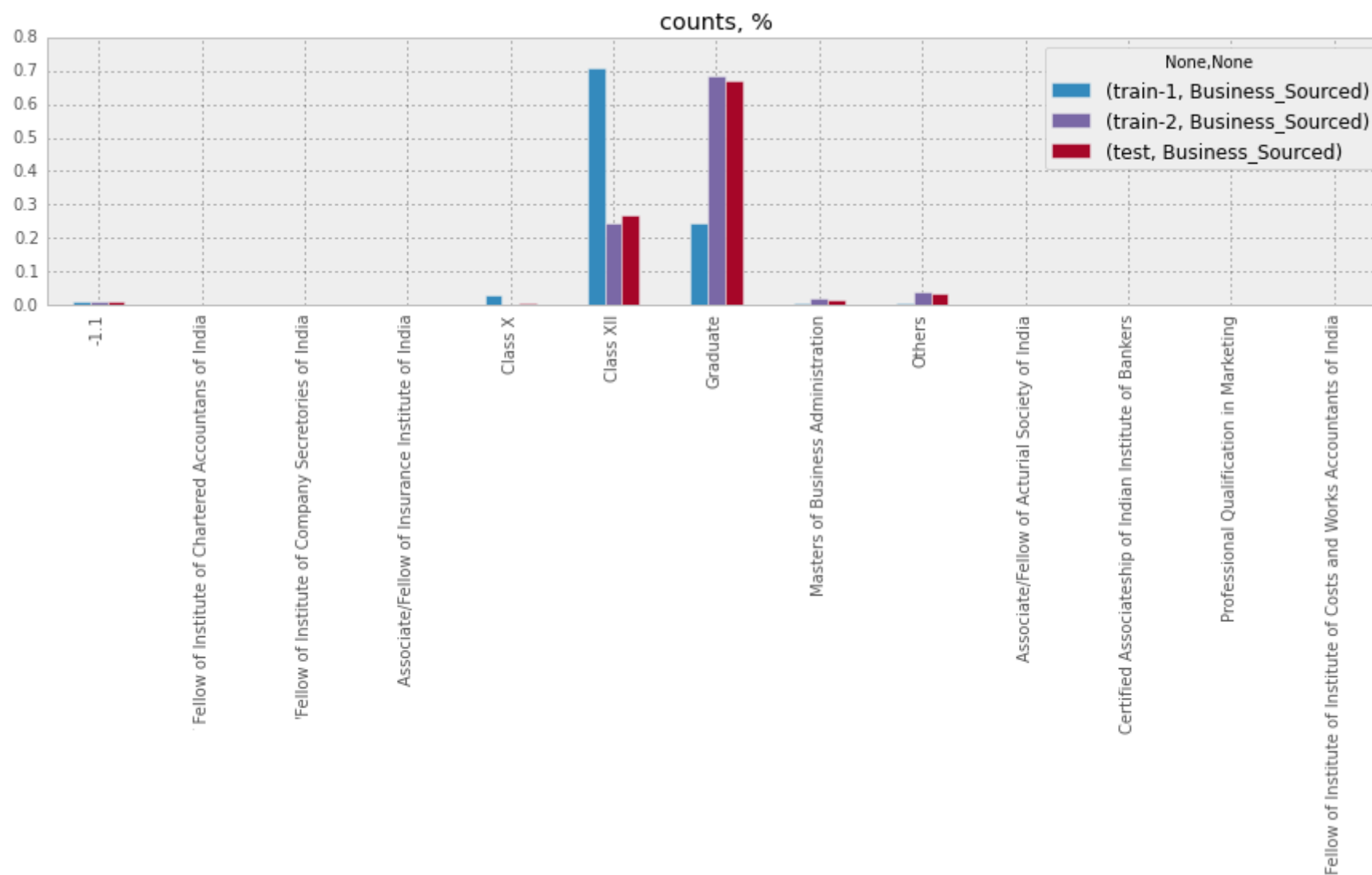
2.3. Взаимодействие с другими признаками

Можно смотреть на стабильность признаков!

| | | train-1 | | train-2 | | test | |
|--|--------------|---------|----------|---------|----------|------|----------|
| | | size | mean | size | mean | size | mean |
| -1.1 | нулевая цель | 67 | 0.000000 | 19 | 0.000000 | 44 | 0.000000 |
| Associate / Fellow of Institute of Chartered Accountants of India | | 3 | 0.333333 | NaN | NaN | 2 | 1.000000 |
| Associate/Fellow of Institute of Company Secretaries 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 |

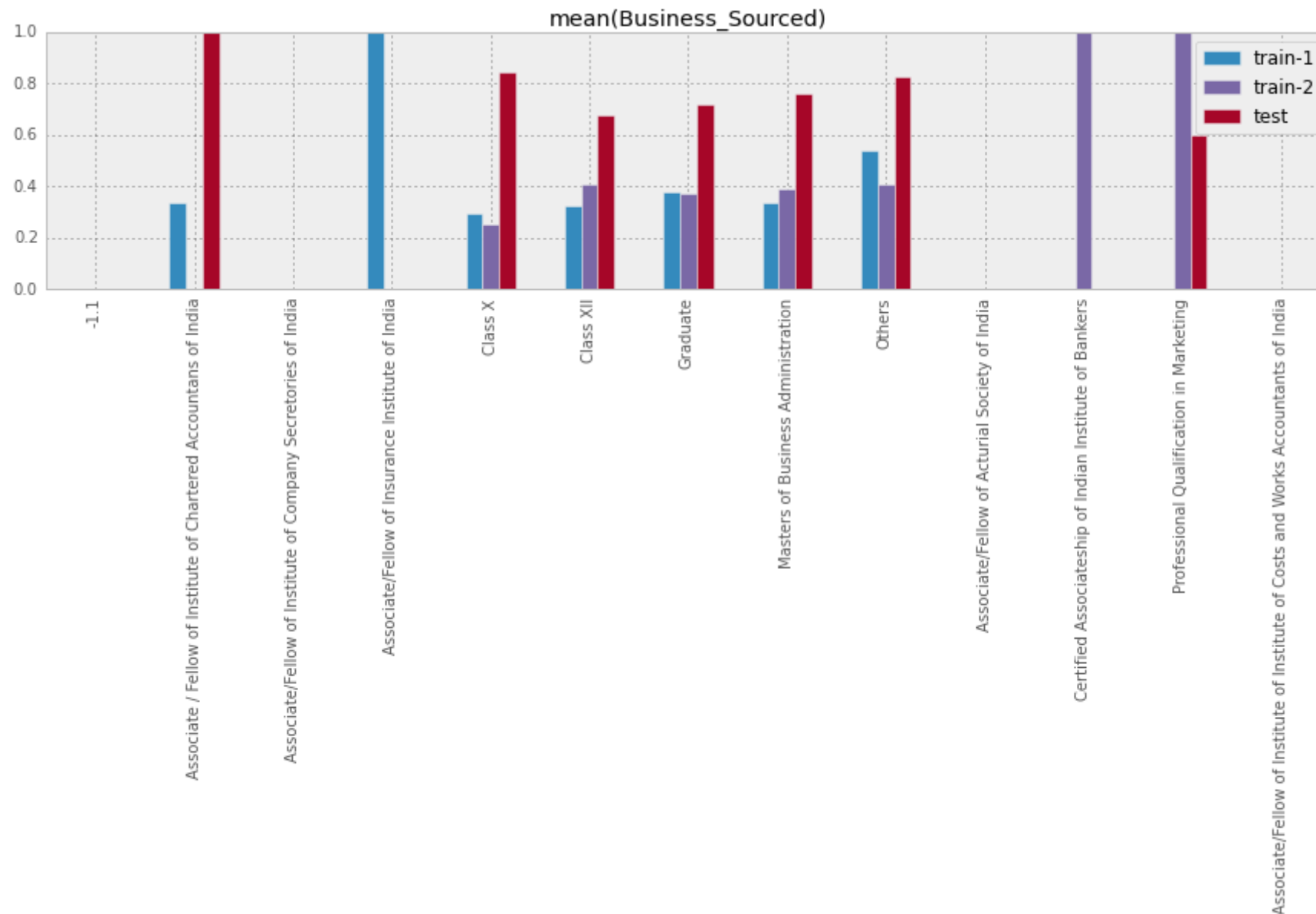
Временные признаки

Как часто встречаются такие значения признаков



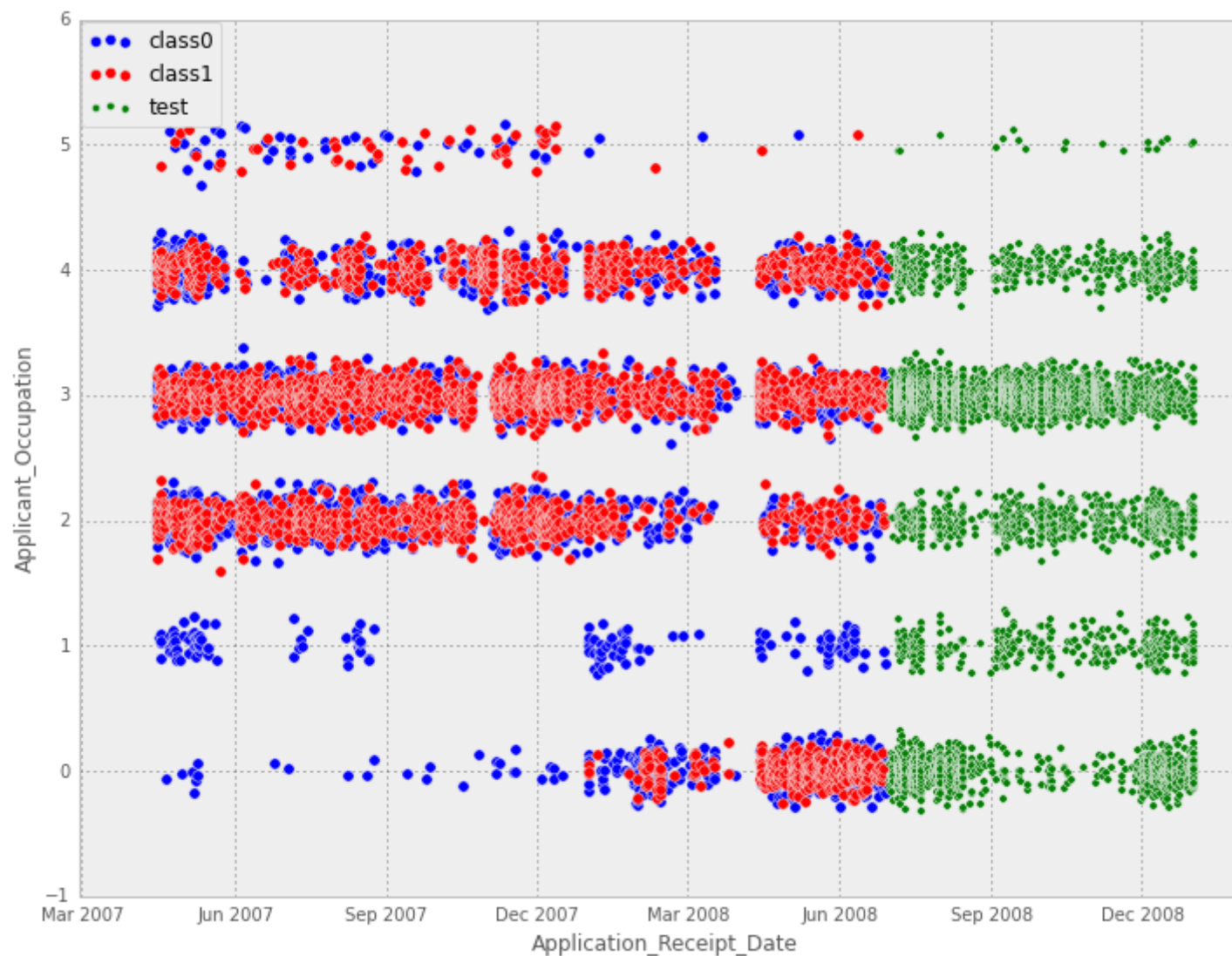
Временные признаки

Как меняется среднее значение целевой переменной



Временные признаки

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



Временные признаки

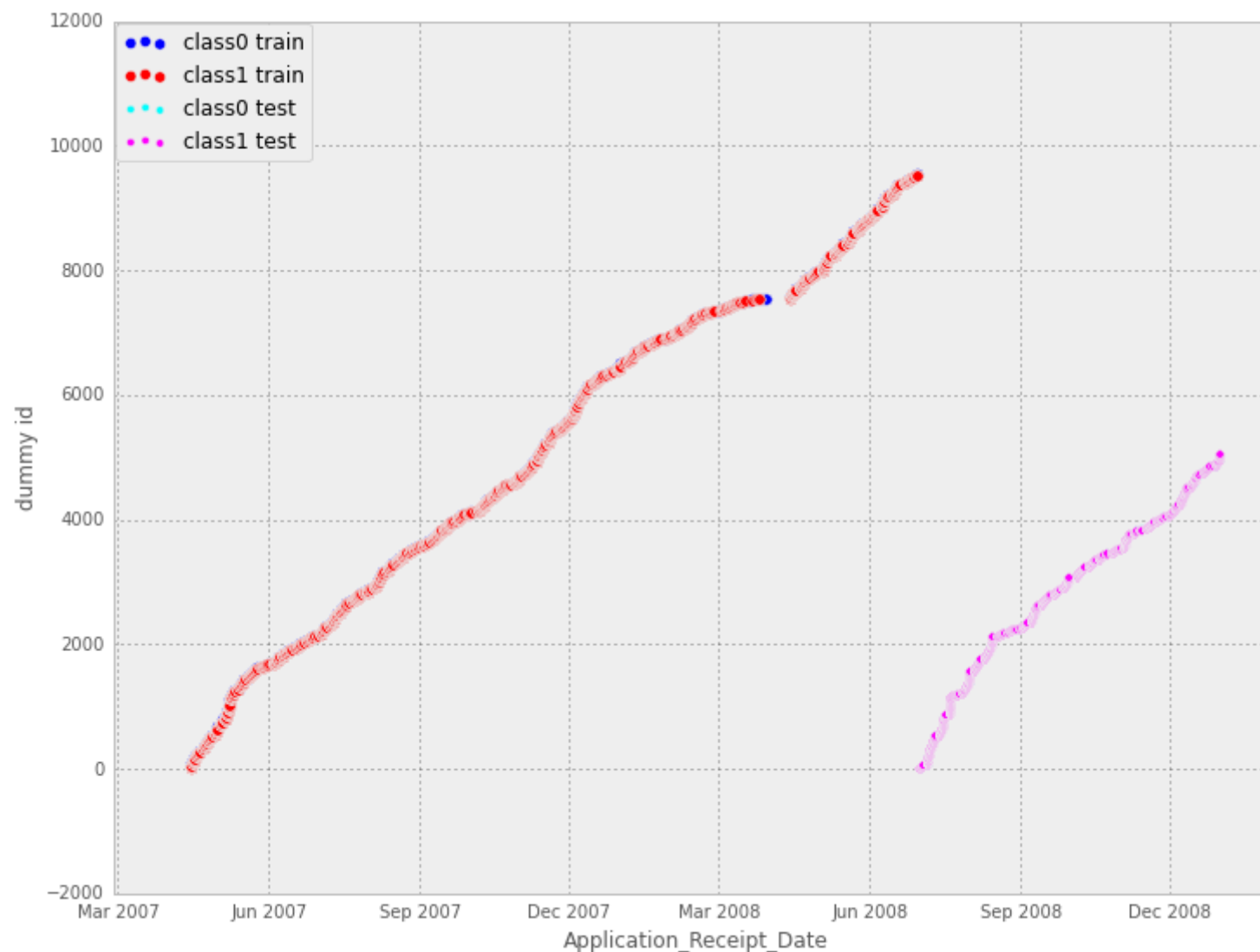
2.3. Смотрим, как меняются другие признаки во времени...

Это позволяет:

- выявить стабильные признаки**
- правильно сформировать разбиения обучение / тест**

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

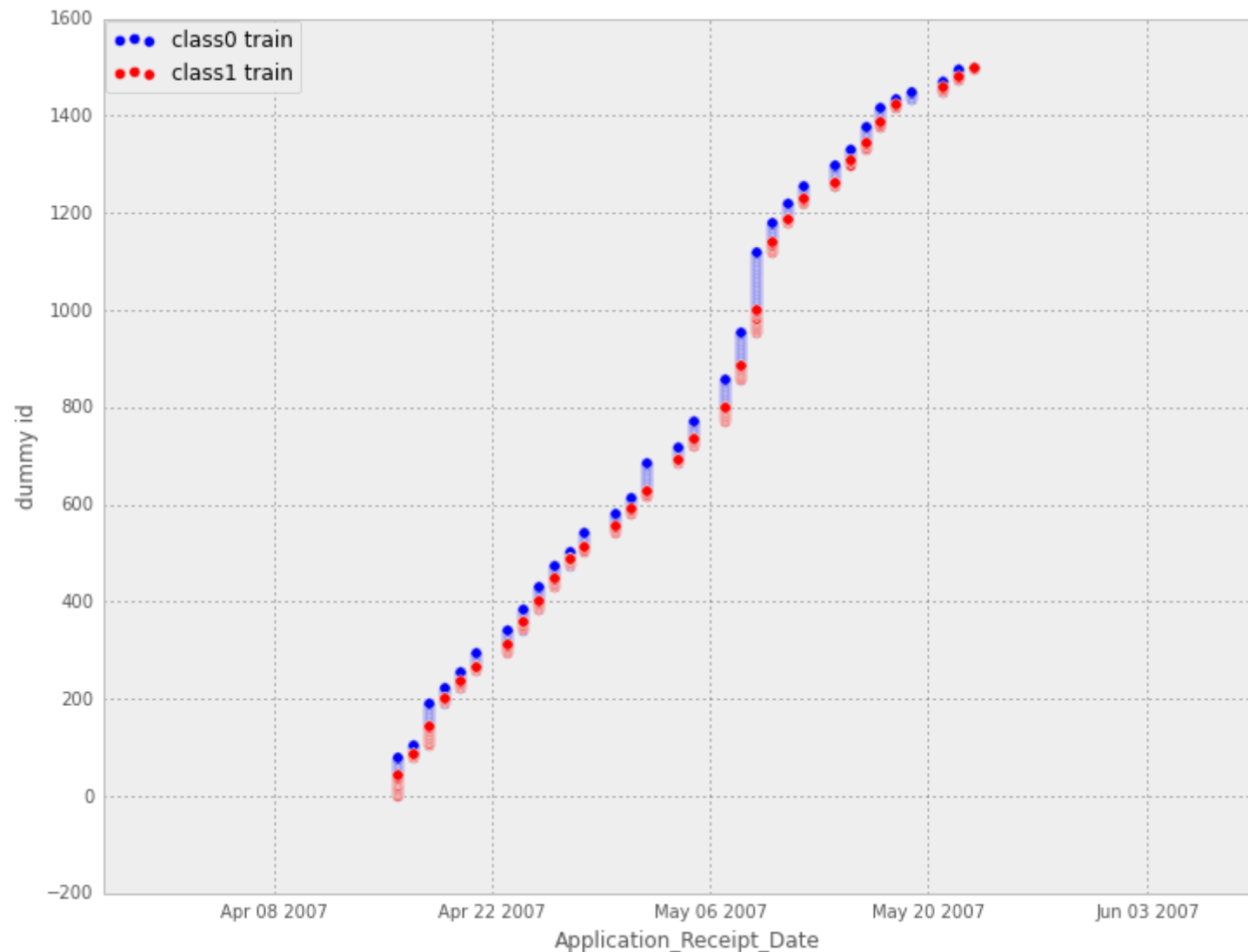
Совет: визуализация id (номер в таблице) – время



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)
```

Д3 Можно ли проще?

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»

картинка

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

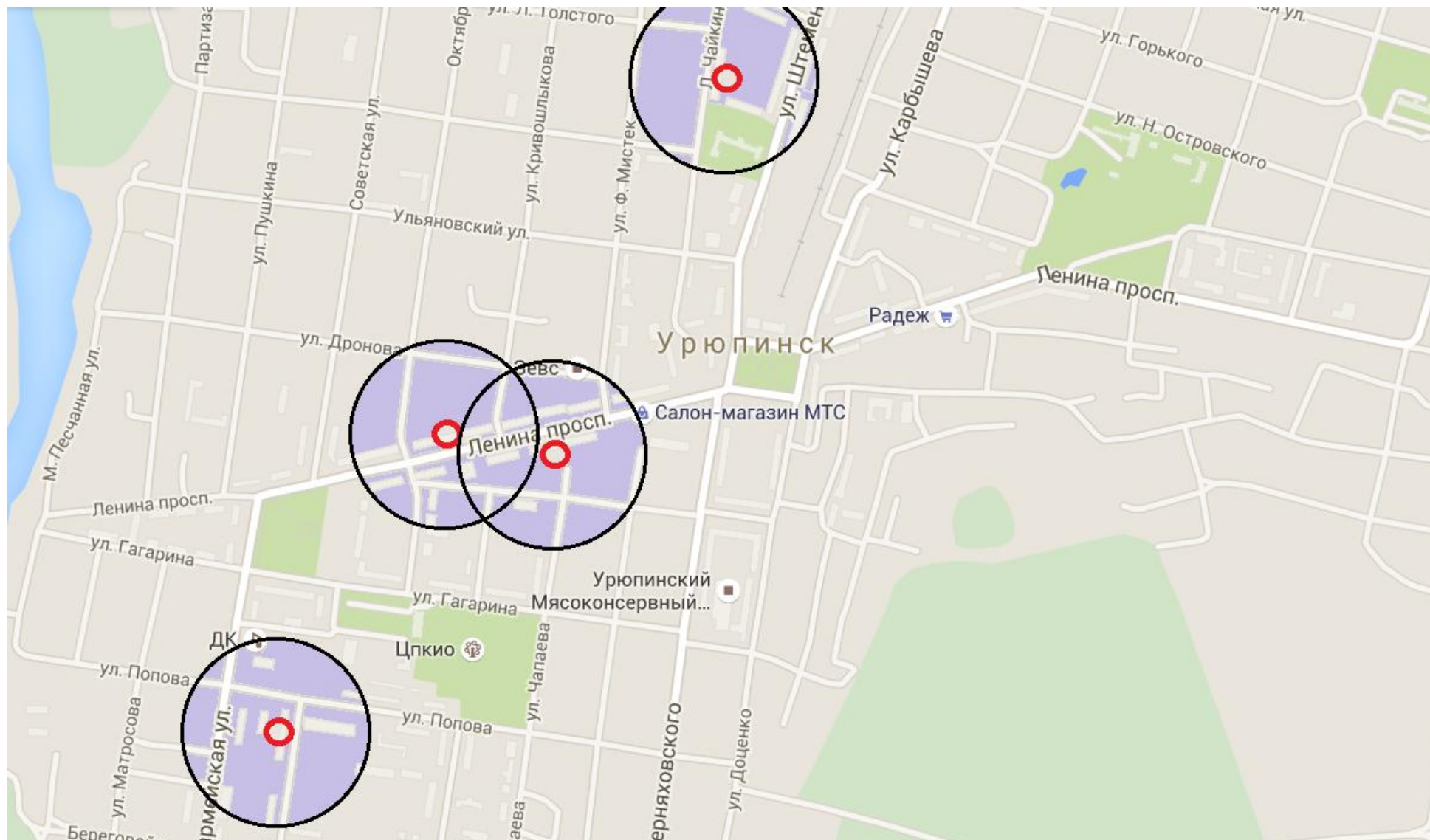
если изменение координат во времени

Географические (пространственные) признаки

Генерация признаков

- **расстояния до**
 - **объектов** (дорога, АЗС, метро, банк, город и т.п.)
 - **вычисленных объектов** (самого дорогого дома, скопления людей)
 - **границ**
 - **кластеров**
- **использовать для генерации других признаков**
средняя цена квартиры в районе, число школ в районе, плотность населения

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**

Деньги, количество и т.п.

- **абсолютные суммы → относительные**
ex: стоимость, площадь → стоимость кв. м.

- **остатки от деления / округления**

-

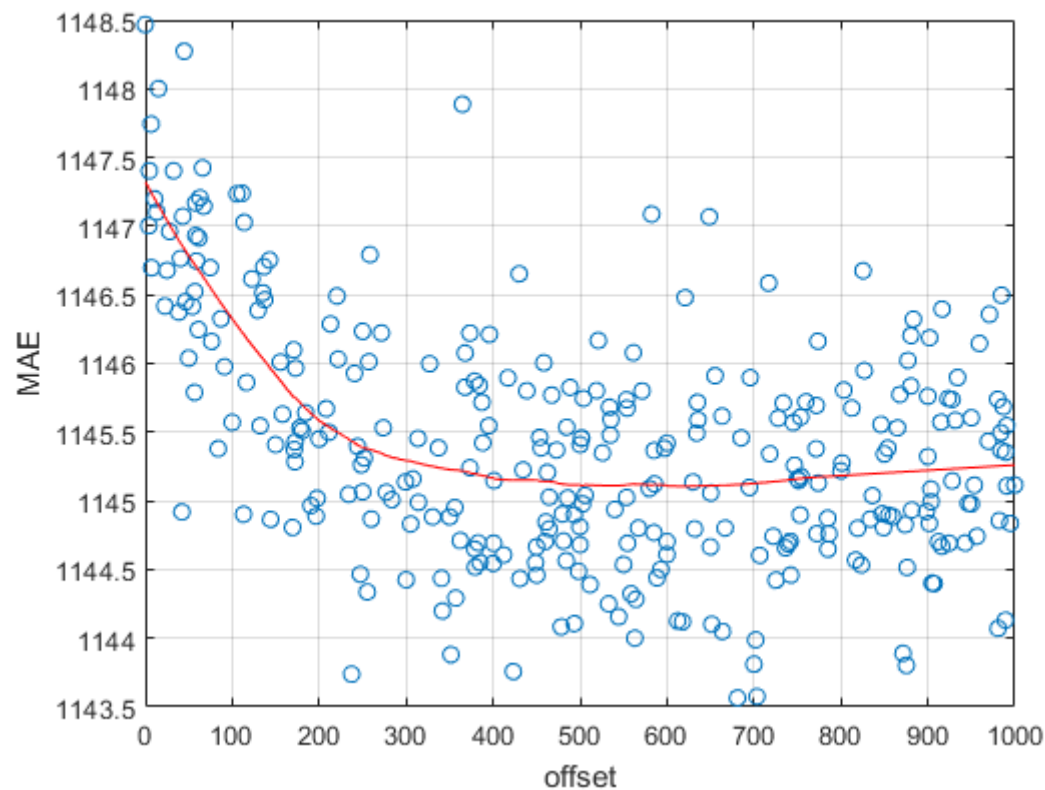
| | price | f_p |
|---|-------|------|
| 0 | 12.50 | 0.50 |
| 1 | 0.99 | 0.99 |
| 2 | 10.00 | 0.00 |
| 3 | 18.01 | 0.01 |

позволяет отделить точные от приблизительных

- **статистики транзакций**

Генерация целевого признака

Исследование сдвига



Преобразование целевого признака $\log(y + \text{offset})$

Литература

Серия постов Understanding Feature Engineering

<https://towardsdatascience.com/understanding-feature-engineering-part-1-continuous-numeric-data-da4e47099a7b>

Книга по генерации признаков

Alice Zheng, Amanda Casari Feature Engineering for Machine Learning, Principles and Techniques for Data Scientists // O'Reilly Media, 2018, pp. 218

Книга по генерации признаков

<http://www.feat.engineering>

Курс «How to Win a Data Science Competition: Learn from Top Kagglers»

<https://ru.coursera.org/learn/competitive-data-science>