



Национальный  
исследовательский  
Томский  
государственный  
университет



# Загрузка, подготовка и предварительное изучение данных. Практика оценки конкурсного задания

Сергей Аксёнов, к.т.н., доцент кафедры  
Теоретических основ информатики ТГУ

# Пример: Классификация

---

Выборка: US 2009 Residential Energy Survey

**12000** Жилищ (Дома & квартиры)

**934** Признаков для каждого дома

- Год постройки,
- Материал стен, крыши,
- Площадь комнат, подвала, гаража,
- Тип топлива для печи для приготовления еды, обогрева,
- Подключение телевизоров, холодильников, компьютеров, тостеров, систем кондиционирования,
- Результаты мониторинга энергоэффективности и т.д.



**Целевая переменная:** Класс региона, для которого спроектирован дом (Холодный, Умеренный, Жаркий-Влажный, Жаркий-Сухой, Морской)

# Работа до соревнования

---

1. Поиск выборки (достаточно крупный и/или сложный набор данных)
2. Решение задачи экспертами самостоятельно
  - ☐ Понять, какие могут быть проблемы
  - ☐ Оценка выполнимости задачи
  - ☐ Определение критериев оценки
  - ☐ Оценка возможностей вычислительной техники
3. Подготовка выборки
  - ☐ Добавление шумов, пропущенных значений (повторное решение)
  - ☐ Разделение выборки на обучающую (с метками классов) и тестовую (с метками (не известна конкурсантам) и без меток – получают конкурсанты)
  - ☐ Описание признаков
4. Подготовка документации

# Информация о данных в выборке (codebook)

```
additional_info = pd.read_csv('codebook.csv', encoding = "utf-8", sep=',')
additional_info.head()
add_data = pd.DataFrame(data.columns)
for index, row in additional_info.iterrows():
    if row['Variable Name'] in add_data[0].unique():
        print('-----')
        print('Name:', row['Variable Name'])
        print('Description:', row['Variable Description'])
        print('Labels', row['Response Codes and Labels'])
```

```
-----
Name: Climate_Region_Pub
Description: Building America Climate Region (collapsed for public file)
Labels 1
2
3
4
5
```

```
-----
Name: YEARMADE
Description: Year housing unit was built
Labels 1600 - 2009
```

```
-----
Name: WALLTYPE
Description: Major outside wall material
Labels 1
2
3
4
5
```

Отсутствие описания может привести к непониманию признаков и возможных зависимостей между ними.

Конструированию фантастических гипотез и заведомо неверных путей решения задачи.

# Просмотр типов данных в наборе

---

```
df.dtypes
Climate_Region_Pub    int64
DIVISION              int64
REPORTABLE_DOMAIN     int64
DOLELCOL              object
TOTALDOLCOL           int64
KWHCOL                float64
BTUELCOL              float64
TOTALBTUCOL           int64
TOTALDOLSPH           int64
TOTALBTUSPH           int64
CELLAR                int64
NWEIGHT               float64
TOTHSQFT              int64
HEATHOME              int64
NUMPC                 int64
DOLLAREL              int64
DOLELOTH              float64
CUFEETNGSPH           float64
..                   ..
```

Какие типы данных используются в наборе?

Нужно ли преобразовывать категориальные признаки? Как их конвертировать?

Какие модели обучения использовать при выполнении задания?

# Автоматизация обработки признаков

---

```
# категориальные признаки
categorical_features = []
for col in data.columns:
    if not (isinstance(data[col].iloc[0], np.float64)
            or isinstance(data[col].iloc[0], np.int64)):
        categorical_features.append(col)
categorical_features
```

```
['DOLELCOL',
 'DOLFOSPH',
 'DOLELSPH',
```

Создание списков признаков

Ускорение обработки

Критически важно, если  
выборка большого размера



# Дубликаты строк / Уникальные значения признаков

Можно сгенерировать дубликаты строк в исследуемую выборку для оценки работы участников.

```
duplicates = data.duplicated()
for i in range(len(duplicates)):
    if duplicates[i]:
        print(i)
```

Очень часто удаление дубликатов записей  
есть среди критериев оценки.

Если дубликатов очень много, то увеличивается время построения моделей и остается меньше времени на исследование.

```
df['building_class'].unique()

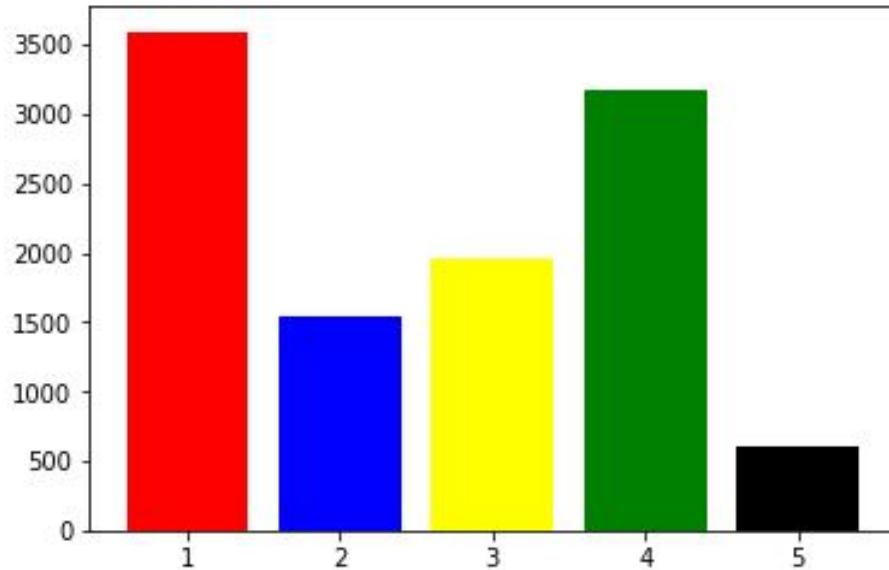
array(['R5', 'G7', 'O6', 'K6', 'H8',
       'RK', 'D3', 'E1', 'RB', 'V0',
       'E7', nan, 'N2', 'F1', 'D6',
       'P9', 'C1', 'D9', 'D7', 'O5',
       'D4', 'U2', 'F4', 'G5', 'G6',
```

## Понимание возможных значений признаков для генерации гипотез

# Распределение классов в выборке

```
plt.bar(np.unique(result_Y, return_counts=True)[0],  
        np.unique(result_Y, return_counts=True)[1],  
        color=['red', 'blue', 'yellow', 'green', 'black'])
```

<BarContainer object of 5 artists>



Какое количество примеров  
есть для каждого из классов  
(сбалансированная или  
несбалансированная  
выборка)?

Метрики используемые для  
оценки качества модели?



# Оценка ситуации с пропущенными значениями -1

```
columns_with_null = data.isnull().sum()
for i in range(len(columns_with_null)):
    if columns_with_null[i] != 0:
        print(i)
```

```
df.isnull().sum()
```

GALLONFO	0
DOLFOSPH	0
DOLLARFO	0
KWHSPH	0
BTUELSPH	0
FOWARM	0
USEFO	0
TOTUSQFT	0
TOTALDOL	0
NUMTHERM	0
DOLELSPH	0
CONCRETE	0

Много ли пропусков в выборке?

Нужно ли сокращать количество признаков, если много пропущенных значений этого признака?

Как бороться с пропущенными значениями в строках (удалить, заменить)?

# Оценка ситуации с пропущенными значениями -2

---

Использование функции удаления всех строк, где есть хотя бы одно пропущенное значение допустимо, только в случае если удаляется очень небольшая часть выборки, не нарушается существенно распределение классов.

```
for i in df.columns:
    percent_of_null = df[i].isnull().sum() * 100 / df.shape[0]
    if percent_of_null > 35:
        df.drop(i, axis = 1, inplace = True)
```

Удаление столбцов, где процент пропуска >35%


# Преобразование категориальных признаков - 1

- Порядковые (можно сравнить их и упорядочить)
- Номинальные (несравнимые)


Датасет: Футболки

	Цвет	Размер	Цена	Метка
0	зеленый	M	10.1	класс1
1	красный	L	13.5	класс2
2	синий	XL	15.3	класс1

$$XL > L > M$$



Номинальный  
признак



Порядковый  
признак

Пример из: Себастьян Рашка Python и машинное обучение, ДМК, Москва-2017, ISBN 978-5-97060-409-0 (категорически рекомендую для студентов-Beginners)

# Преобразование категориальных признаков - 2

- Для порядковых – задание соответствия вручную исходя из понимания признака

$$XL = L + 1 = M + 2$$

```
size_mapping = {    # словарь соответствий
                  'XL': 3,
                  'L'  : 2,
                  'M'  : 1}
df['размер'] = df['размер'].map(size_mapping)
```

	Цвет	Размер	Цена	Метка
0	зеленый	1	10.1	класс1
1	красный	2	13.5	класс2
2	синий	3	15.3	класс1

Результат

# Преобразование категориальных признаков - 3

□ Для номинальных – прямое кодирование

```
pd.get_dummies(df[['цена', 'цвет', 'размер']])
```

	Цена	Размер	Цвет_зеленый	Цвет_красный	Цвет_синий
0	10.1	1	1	0	0
1	13.5	2	0	1	0
2	15.3	3	0	0	1

Модели дерева решений и случайный лес не требуют преобразований категориальных признаков

# Приведение данных к одному масштабу

Кластеризация и **многие** алгоритмы обучения с учителем крайне нуждаются в приведении признаков к одной шкале, т.к. их вычисления завязаны на учет расстояния между объектами в выборке.

Нормализация:

$$x_{\text{norm}}^{(i)} = \frac{x^{(i)} - x_{\min}}{x_{\max} - x_{\min}}$$

$x_{\min}$  — наименьшее значение

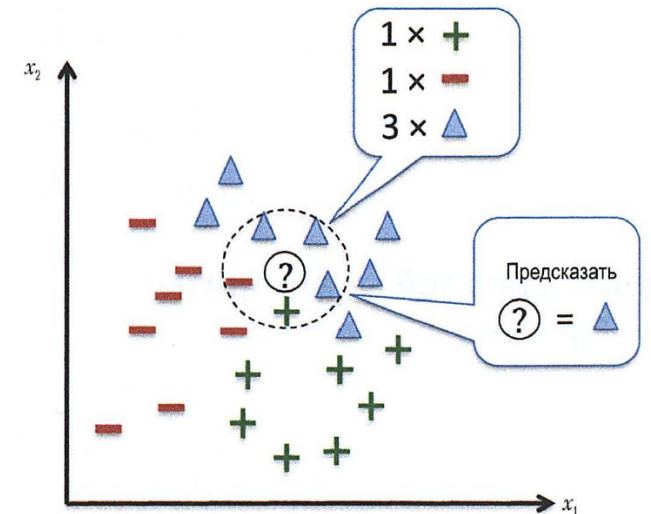
$x_{\max}$  — наибольшее значение

Стандартизация:

$$x_{\text{std}}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

$\mu_x$  — эмпирическое среднее

$\sigma_x$  — стандартное отклонение.



Пример



# Удаление выбросов

---

```
# убираем выбросы через 3sigma.  
#Все, что за пределами 3sigma, будем считать выбросом  
  
# получим стандартное отклонение столбцов  
std = data_without_aim.std()  
  
# получим имена столбцов  
cols = std.iloc[0:len(std)].index.tolist()  
  
# для каждого значения столбцов, проверим на  
#выбросы и уберем, если это так  
for col in cols:  
    data_without_aim[np.abs(data_without_aim[col]  
                           -data_without_aim[col].mean())  
                     <= (3*data_without_aim[col].std())]
```

Устранение значений,  
значительно отличающихся от  
других:

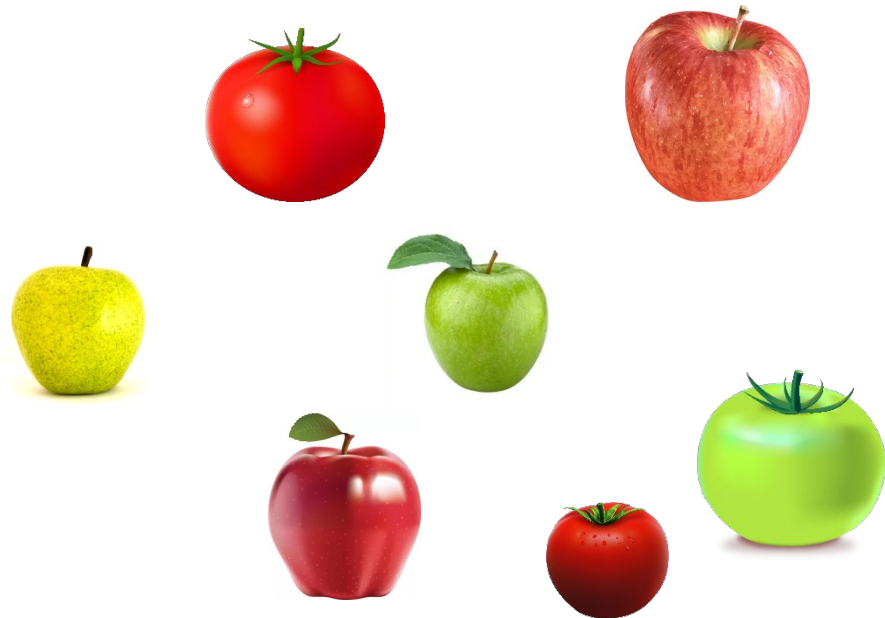
- Повышение адекватности модели
- Есть риск удаления ценной информации



# Пример из обучения без учителя

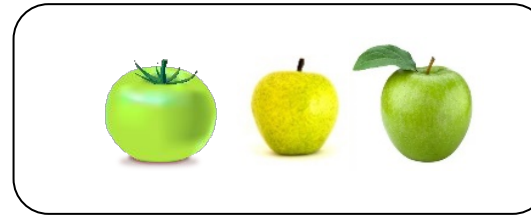
Особенность: Субъективность кластеризации.

**Задача:** Разложить объекты на две группы

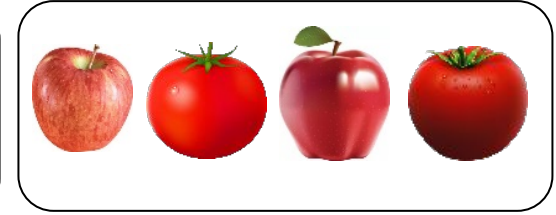


Решение А

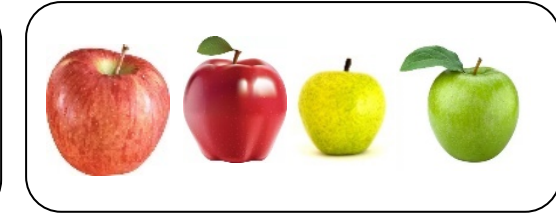
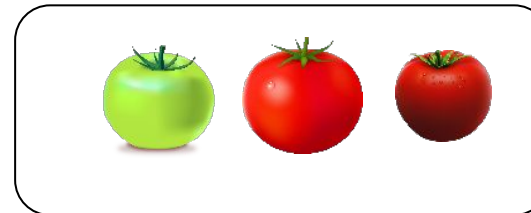
Группа 1



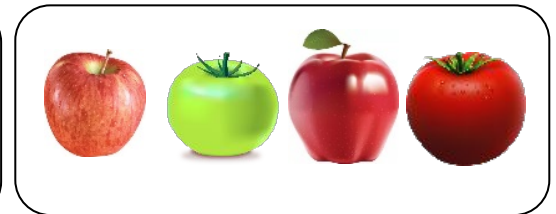
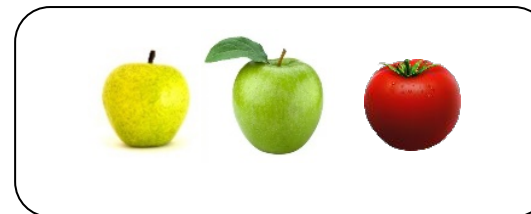
Группа 2



Решение В



Решение С



Разные решения!!!