Отбор признаков

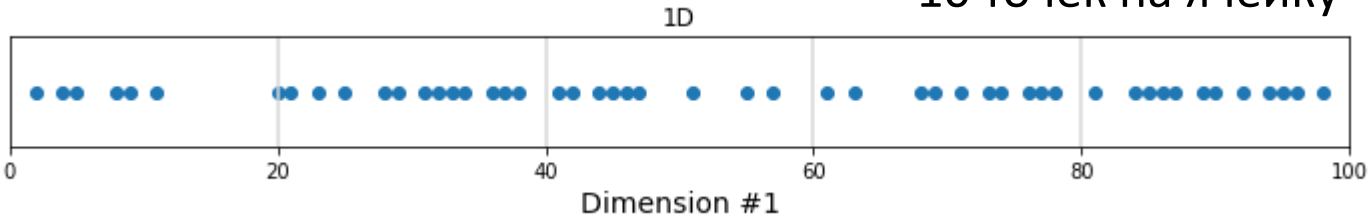
Feature Selection

- Отбор признаков – процесс определения подмножества релевантных признаков для построения модели
- Причины использования отбора признаков:
  - упрощение модели для интерпретации
  - сокращение времени обучения
  - избегание проклятия размерности (curse of dimensionality)
  - улучшение совместимости данных с классом моделей
- Основная гипотеза: данные содержат *избыточные* (redundant) и *нерелевантные* (irrelevant) признаки
  - релевантный признак может стать избыточным при наличии другого релевантного признака (корреляция)

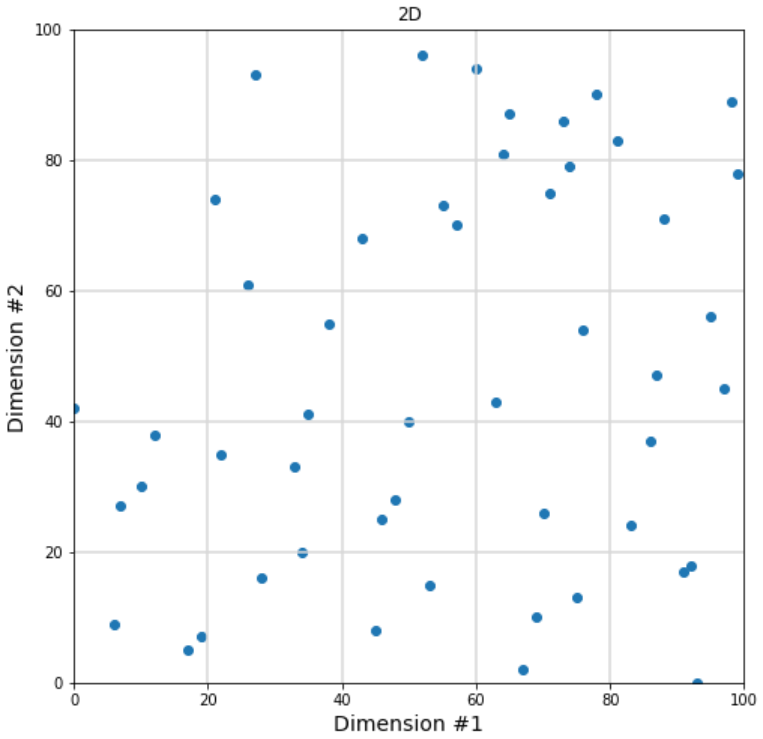
Curse of dimensionality

50 точек:

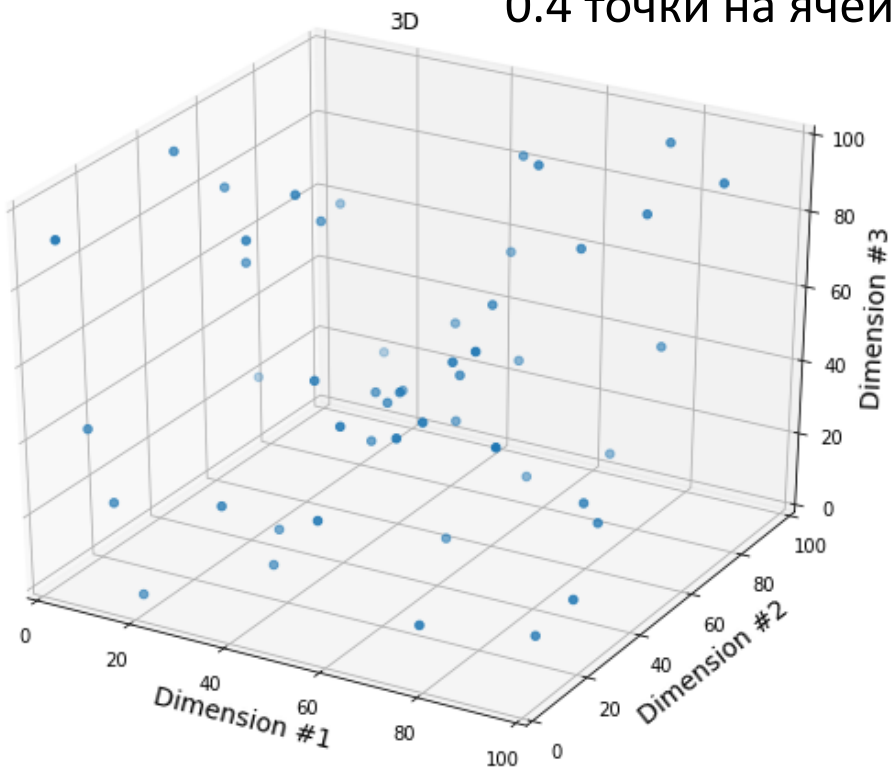
10 точек на ячейку



2 точки на ячейку



0.4 точки на ячейку

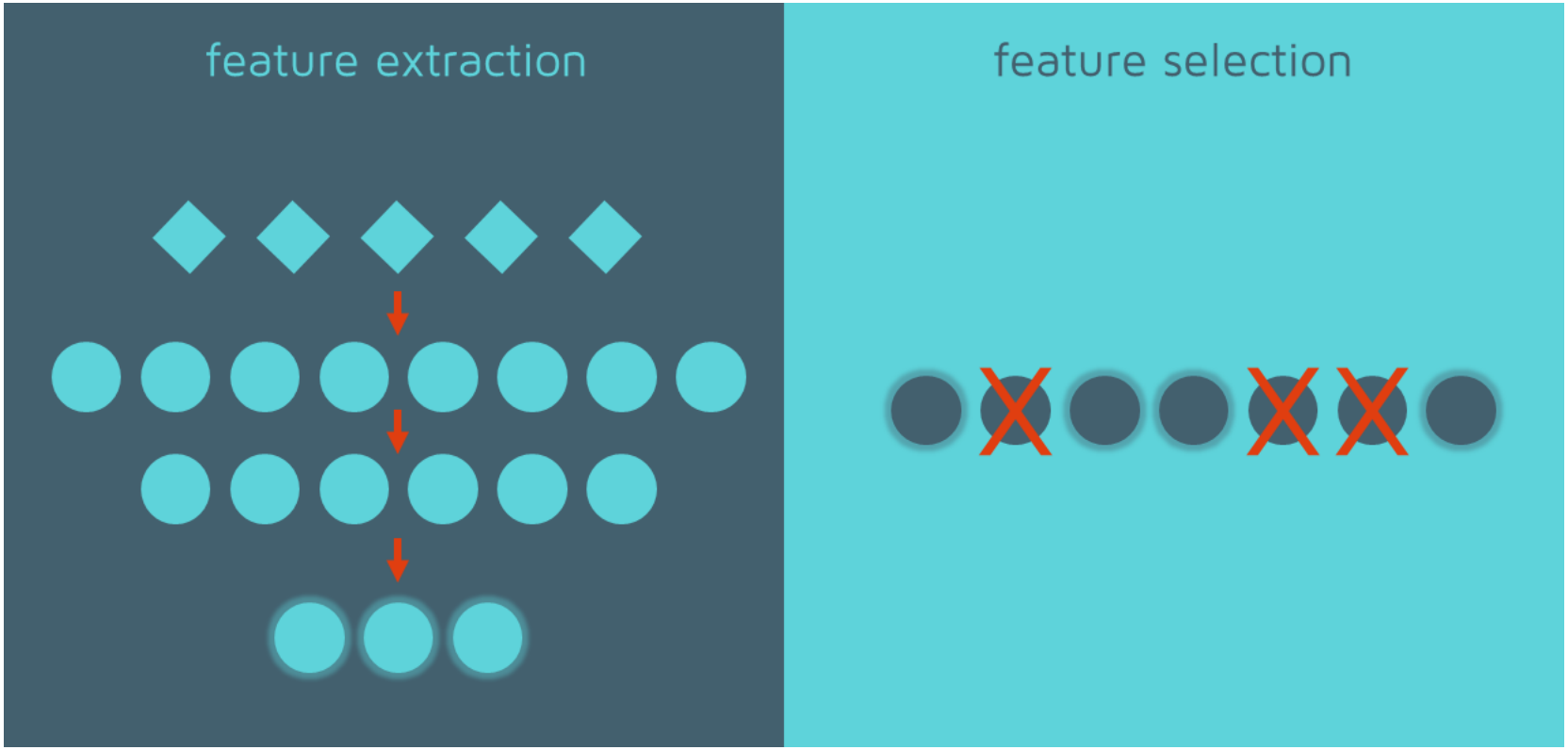


Данные должны расти экспоненциально!

Dimensionality reduction

- *Снижение размерности* (dimensionality reduction) – преобразование данных из высокоразмерного пространства в низкоразмерное с сохранением свойств, важных для решаемой задачи
- Подходы:
  - *отбор признаков* (feature selection) – процесс определения подмножества релевантных признаков для построения модели
  - *извлечение признаков* или *выделение признаков* (feature extraction или feature projection) – переход из исходного пространства признаков в **новое** пространство меньшей размерности

Feature extraction vs. Feature selection



Feature extraction

Методы feature extraction:

- Principal Component Analysis (PCA)
- Kernel Principal Component Analysis (KPCA)
- Linear Discriminant Analysis (LDA)
- Non-negative Matrix Factorization (NMF)
- Autoencoders
- t-distributed Stochastic Neighbor Embedding (t-SNE)
- Uniform Manifold Approximation and Projection (UMAP)

Feature extraction

Связанные термины:

- Feature engineering:
  - создание признаков на основе «сырых» данных:
    - feature creation
    - feature transformation
    - feature extraction
    - feature selection
  - иногда как синоним feature extraction
- Feature construction – создание признаков вручную
- Feature learning – создание признаков в процессе обучения

Feature selection

- Алгоритм feature selection включает:
  - способ формирования подмножества признаков
  - способ оценки качества сформированного множества
- Подходы:
  - методы-фильтры (filter methods)
  - методы-обертки (wrapper methods)
  - встроенные методы (embedded methods)

Filter methods

- *Методы-фильтры* используют информацию, извлеченную из обучающей выборки (без привлечения алгоритмов машинного обучения)
- Примеры: Chi-Square Score, Mutual Information, Pointwise Mutual Information (PMI), ReliefF, Information Gain и др.
- Ранжируют признаки – требуется способ определения оптимального количества признаков
- Преимущества:
  - высокая скорость
  - независимость от конкретного алгоритма машинного обучения
  - в большей степени отражают связи между признаками
- Недостаток:
  - качество предсказания ниже, чем для других методов
- Часто используются перед методами-обертками

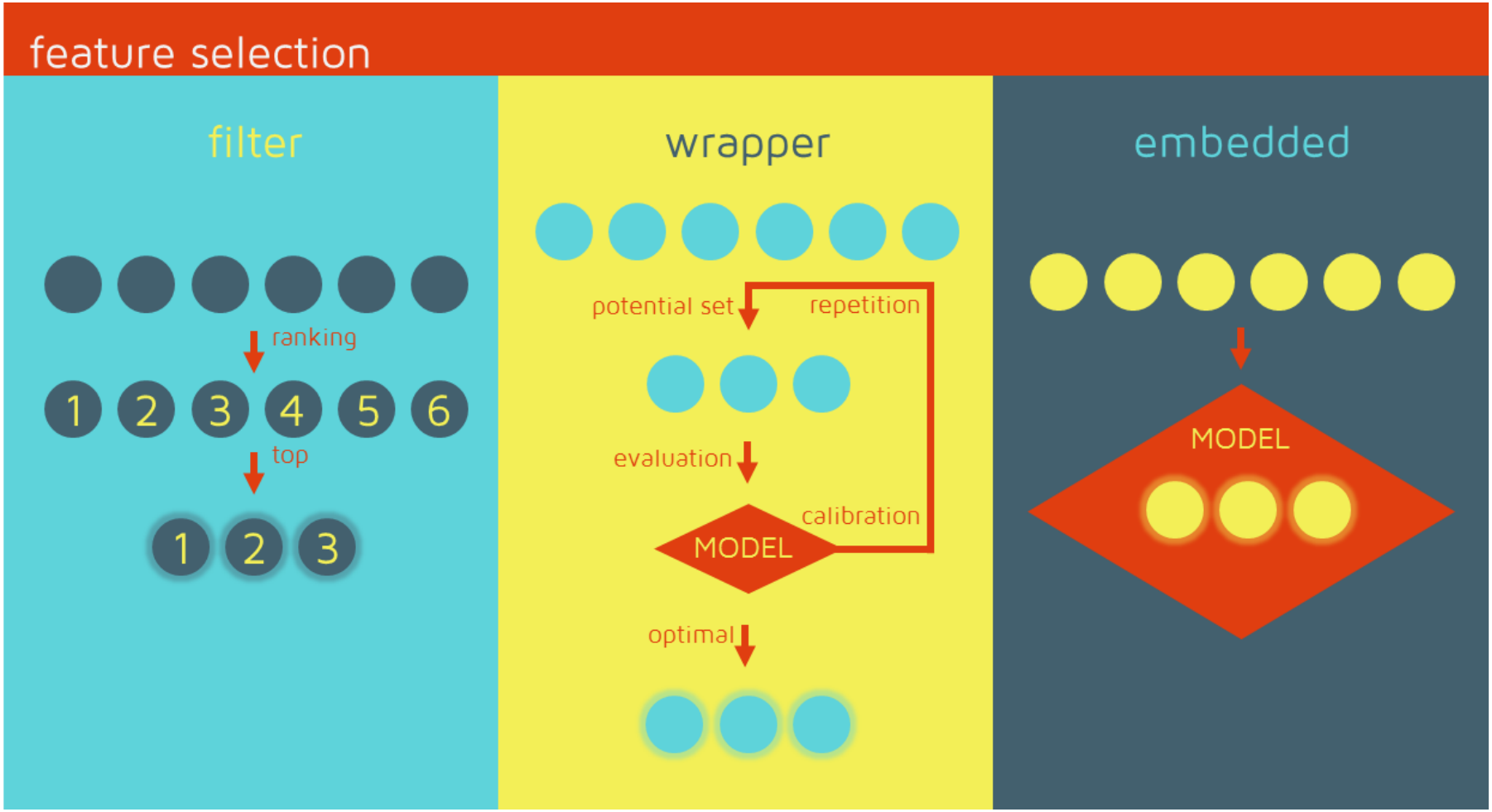
Wrapper methods

- *Методы-обертки* используют алгоритм машинного обучения для оценки качества текущего подмножества признаков
- На каждом шаге требуется обучать модель
- Качество оценивается на отложенной выборке
- Примеры: Genetic algorithm, Ant colony, Simulated annealing, Recursive Feature Elimination (RFE)
- Преимущество: высокое качество предсказания
- Недостаток: низкая скорость

Embedded methods

- *Встроенные методы* являются частью процесса построения модели машинного обучения
- Примеры: LASSO, Decision trees
- Преимущество: менее вычислительно сложные, чем методы-обертки
- Недостаток: связаны с алгоритмом машинного обучения

Feature selection



Методы-фильтры

Variance Threshold

- Исключаем признаки с дисперсией меньше заданного порога
- Пример: бинарные признаки, $D[X] = p(1-p)$, где p – вероятность единиц (или нулей)
- Пусть $p = 0.8$, тогда $D[X] = 0.16$
- `sklearn.feature_selection.VarianceThreshold`
- См. feature_selection.ipynb

N	X1	X2	X3
1	0	0	1
2	1	0	1
3	0	1	1
4	1	0	1
5	1	0	1
Дисперсия	**0.24**	**0.16**	**0.00**

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html

Univariate feature selection

- `SelectKBest` – удаляет все признаки, кроме k признаков с наивысшими оценками
  - `score_func=f_classif` – функция для оценки признаков
  - `k=10` – количество наилучших признаков
- `SelectPercentile` – удаляет все признаки, кроме заданного процента признаков с наивысшими оценками
  - `score_func=f_classif` – функция для оценки признаков
  - `percentile=10` – процент наилучших признаков

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectPercentile.html

Univariate feature selection

Функции оценки признаков:

- классификация:
  - `chi2`
  - `f_classif`
  - `mutual_info_classif`
- регрессия:
  - `r_regression`
  - `f_regression`
  - `mutual_info_regression`

Pearson correlation coefficient

- Коэффициент корреляции Пирсона:

$$\rho_{xy} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\mathbb{E}[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

- `sklearn.feature_selection.r_regression` – вычисляет коэффициент корреляции между каждым признаком и целевой переменной

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.r_regression.html

chi2

- Вычисление статистики χ^2 между каждым неотрицательным признаком и классом
  - Признак – частота, количество или бинарный
- Нулевая гипотеза: признак и класс независимы (признак не релевантен)
- Для таблицы сопряженности (contingency table) с r строками и c столбцами:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{\left(O_{i,j} - E_{i,j}\right)^2}{E_{i,j}},$$

где $O_{i,j}$ – наблюдаемое (observed) значение в ячейке i, j,

$E_{i,j}$ – ожидаемое (expected) значение в ячейке i, j

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html

chi2: пример про «Титаник»

Sex/Survived	Observed	Expected	O-E	(O-E)^2	(O-E)^2/E
Female Survived	233	120.5	112.47	12650.57	104.96
Female not Survived	81	193.5	-112.47	12650.57	65.39
Male Survived	109	221.5	-112.47	12650.57	57.12
Male not Survived	468	355.5	112.47	12650.57	35.58
					263.05
	Survived	Not survived			
Female	233	81	314	35.2%	
Male	109	468	577	64.8%	
	342	549	891		
Chi2 (probability)	9.8E-57				

chi2: пример про «Титаник»

Sex/Survived	Observed	Expected	O-E	(O-E)^2	(O-E)^2/E
Female Survived	110	120.5	-10.53	110.78	0.92
Female not Survived	204	193.5	10.53	110.78	0.57
Male Survived	232	221.5	10.53	110.78	0.50
Male not Survived	345	355.5	-10.53	110.78	0.31
					2.30
	Survived	Not survived			
Female	110	204	314	35.2%	
Male	232	345	577	64.8%	
	342	549	891		
Chi2 (probability)	0.5118				

ANOVA F-value

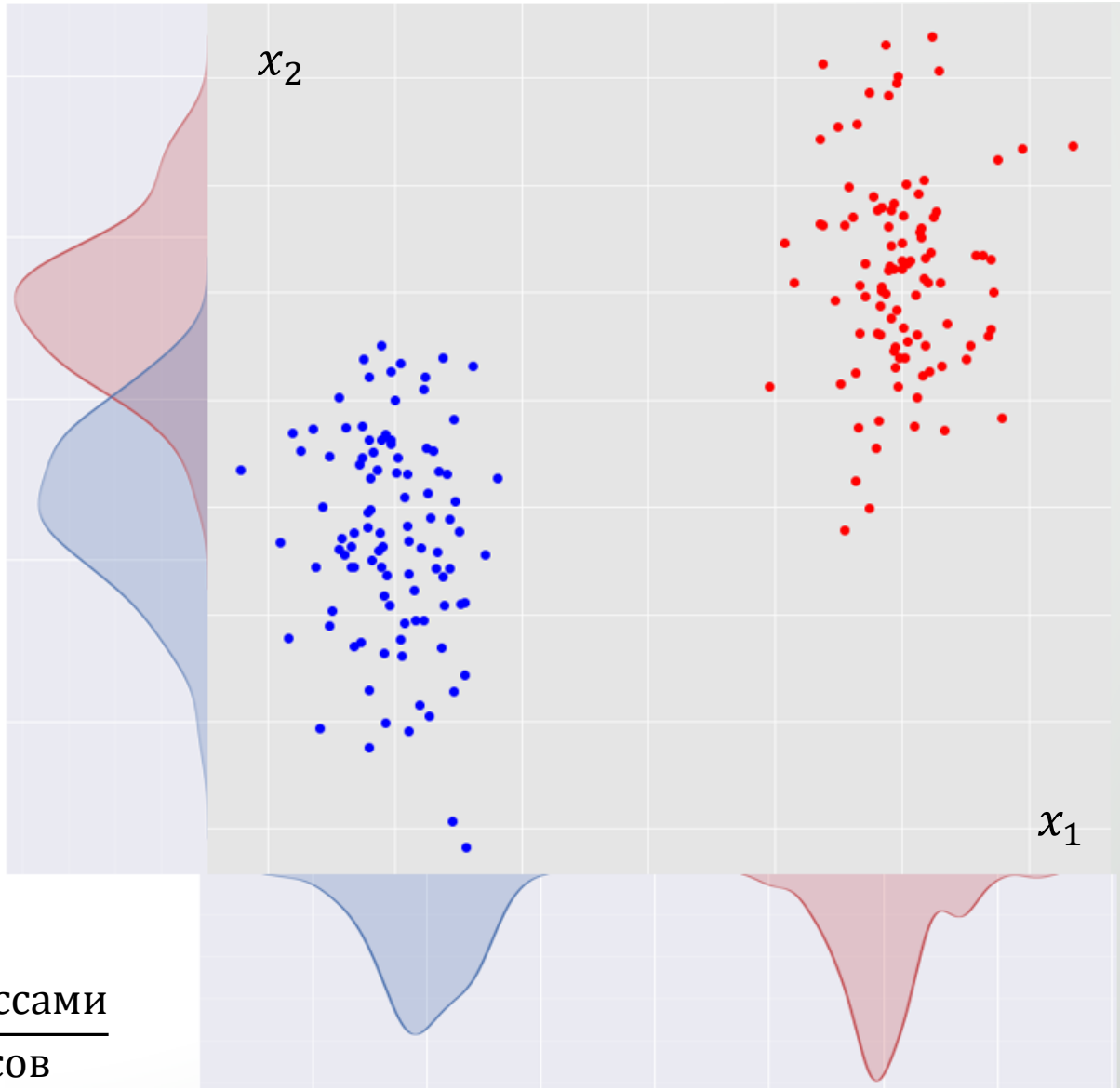
- scikit-learn:
  - `sklearn.feature_selection.f_classif`
  - `sklearn.feature_selection.f_regression`
- ANOVA (ANalysis Of VAriance) – применяется для определения значимости различий средних значений
  - Используется F-тест (критерий Фишера)
- Нулевая гипотеза: средние значения всех выборок равны
- Альтернативная гипотеза: среднее значение хотя бы одной выборки отличается от остальных

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_regression.html

ANOVA F-value

- Насколько хорошо признаки x_1 и x_2 разделяют классы?
- Критерии:
  1. Классы далеко друг от друга (расстояние между средними велико)
  2. Классы компактны (дисперсия классов невелика)



$$score = \frac{\text{расстояние между классами}}{\text{компактность классов}}$$

ANOVA F-value

- Расстояние между классами (числитель):

$$MSB = \frac{1}{df_G} SSB = \frac{1}{df_G} \sum_{i=1}^{k} n_i (\bar{x}_i - \bar{x})^2 ,$$

где MSB – Mean Square Between groups,

SSB – Sum of Squares Between groups,

df_G – Degree of freedom for groups: $df_G = k - 1$,

k – number of groups,

$\bar{x}_i$ – ith group mean,

$\bar{x}$ – grand mean,

n_i – number of observations in ith group

ANOVA F-value

- Компактность классов (знаменатель):

$$MSW = \frac{1}{df_w} SSW = \frac{1}{df_w} \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(x_{ij} - \bar{x}_i\right)^2,$$

где MSW – Mean Square Within groups,

SSW – Sum of Squared Within groups,

df_w – Degrees of freedom within groups: $df_w = n - k$,

n – total number of observations,

n_i – number of observations in ith group

ANOVA F-value

- F-тест (критерий Фишера):

$$F = \frac{MSB}{MSW}$$

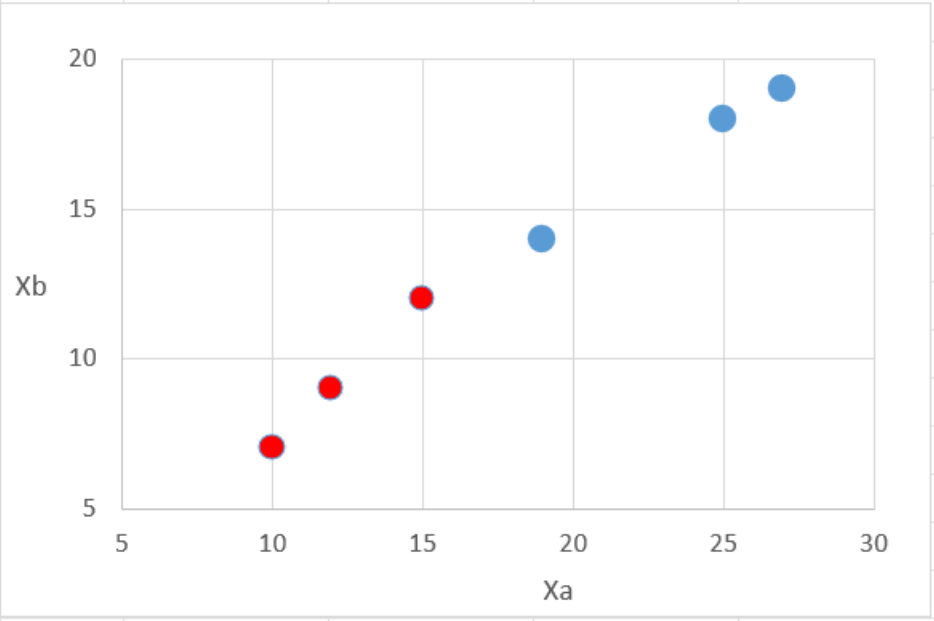
- Чем ближе MSB к MSW, тем более вероятна нулевая гипотеза о равенстве средних
- Если $F > F_{critical}$, то нулевая гипотеза отвергается

ANOVA F-value

$$MSB = \frac{1}{df_G}\sum_{i=1}^{k} n_i(\bar{x}_i - \bar{x})^2$$

$$MSW = \frac{1}{df_w}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_i)^2$$

	Xa	Xb	Y		(Xai-Xcp0)^2	(Xai-Xcp1)^2	(Xbi-Xcp0)^2	(Xbi-Xcp1)^2
Пример 1	10	7	0		5.44		5.44	
Пример 2	12	9	0		0.11		0.11	
Пример 3	15	12	0		7.11		7.11	
Пример 4	19	14	1			21.78		9.00
Пример 5	25	18	1			1.78		1.00
Пример 6	27	19	1			11.11		4.00
				Сумма	12.67	34.67	12.67	14.00
Среднее всех объектов	18.00	13.17						
Среднее по классу 0	12.33	9.33						
Среднее по классу 1	23.67	17.00						
Количество объектов класса 0	3	3						
Количество объектов класса 1	3	3						
Дисперсия средних	192.67	88.17						
Коэффициент	0.25	0.25						
Дисперсия внутренняя	11.83	6.67						
F-test	16.28	13.23						



$$F = \frac{MSB}{MSW}$$

Методы-обертки

SelectFromModel

- SelectFromModel – выбор признаков на основе результатов обучения модели, которая умеет возвращать значимость признаков
  - `coef_, feature_importances_`
  - `Lasso, SVC("l1"), GradientBoosting, DecisionTree`
- Исключаются признаки, значимость которых ниже заданного порога (по умолчанию для L1 10^{-5}, иначе `mean`)
- Используются эвристики:
  - `mean` – значимость ниже среднего значения
  - `median` – значимость ниже медианы
  - `k*mean`, `k*median` – с учетом коэффициента

SelectFromModel

```
class sklearn.feature_selection.SelectFromModel(
    estimator,          # fitted or non-fitted
    *,
    threshold=None,     # mean or 1e-5
    prefit=False,       # fitted model?
    norm_order=1,       # if dimension of coef_ > 1
    max_features=None,  # only this if threshold=-np.inf
    importance_getter='auto’
)
```

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html

Recursive Feature Elimination (RFE)

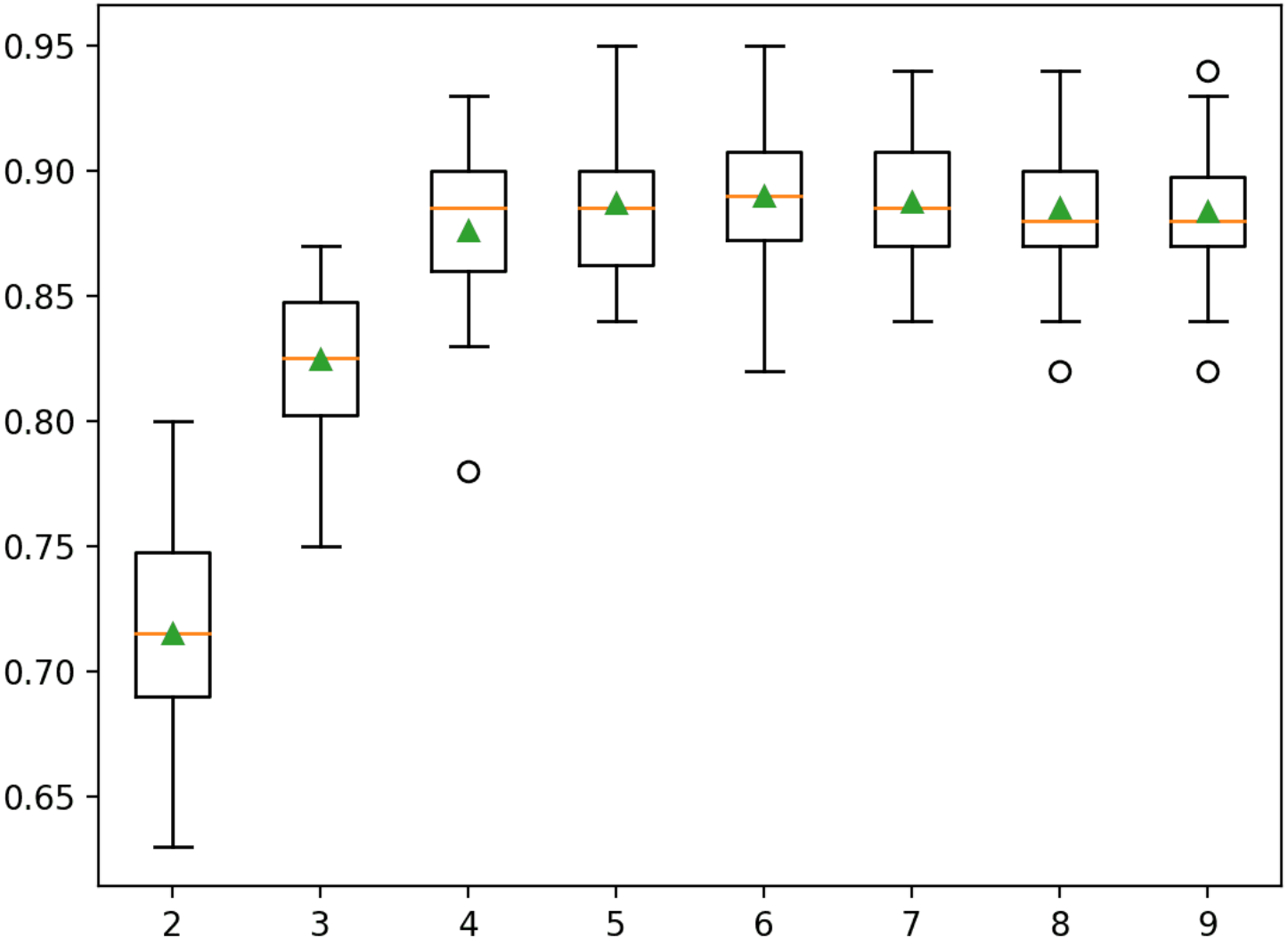
- Рекурсивное исключение признаков – метод-обертка над моделью, которая умеет возвращать значимость признаков
  - `coef_, feature_importances_`
  - `LinearRegression, SVC, GradientBoosting, DecisionTree`
- Алгоритм:
  1. Модель обучается на исходном множестве признаков, возвращая значимость признаков
  2. Наименее значимые `step` признаков исключаются
  3. Модель заново обучается
  4. Шаги 2–3 повторяются до тех пор, пока не получено `n_features_to_select` признаков

Recursive Feature Elimination (RFE)

```
class sklearn.feature_selection.RFE(
     estimator,
     *,
     n_features_to_select=None,  # half of the features
     step=1,
     verbose=0,
     importance_getter='auto’
)
```

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

RFE: cross-validation



RFECV: поиск оптимального количества

```
class sklearn.feature_selection.RFECV(
     estimator,
     *,
     step=1,
     min_features_to_select=1,
     cv=None,
     scoring=None,
     verbose=0,
     n_jobs=None,
     importance_getter='auto’
)
```

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html

Sequential Feature Selection (SFS)

- Последовательный отбор признаков – жадный метод-обертка над моделью, которая не обязана возвращать важность признаков
- Forward-SFS:
  1. Установить количество признаков = 0
  2. Обучить модель с использованием каждого одного признака из m в кросс-валидации
  3. Выбрать признак с наилучшим качеством
  4. Повторять шаги 2–3, выбирая каждый раз из уменьшенного на единицу множества признаков, пока не выполнится условие останова
- Условия останова:
  - достигнуто заданное количество признаков
  - изменение качества на итерации не превышает `tol`

Sequential Feature Selection (SFS)

- Backward-SFS – аналогичная процедура, которая начинает с количества признаков = m и последовательно исключает один признак на каждом шаге
- Forward-SFS и Backward-SFS возвращают неодинаковые результаты
- Выбор Forward-SFS или Backward-SFS может зависеть от требуемого количества признаков – откуда быстрее прийти
- Отличия от SelectFromModel и RFE:
  - не требует, чтобы модель предоставляла значимость признаков
  - может быть медленнее: в Backward-SFS на первом шаге требуется обучить mk моделей для k-fold cross-validation

Sequential Feature Selection (SFS)

```
class sklearn.feature_selection.SequentialFeatureSelector(
    estimator,
    *,
    n_features_to_select='warn’,  # ‘auto’, int, float
    tol=None,
    direction='forward’,          # ‘backward’
    scoring=None,
    cv=5,
    n_jobs=None
)
```

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html