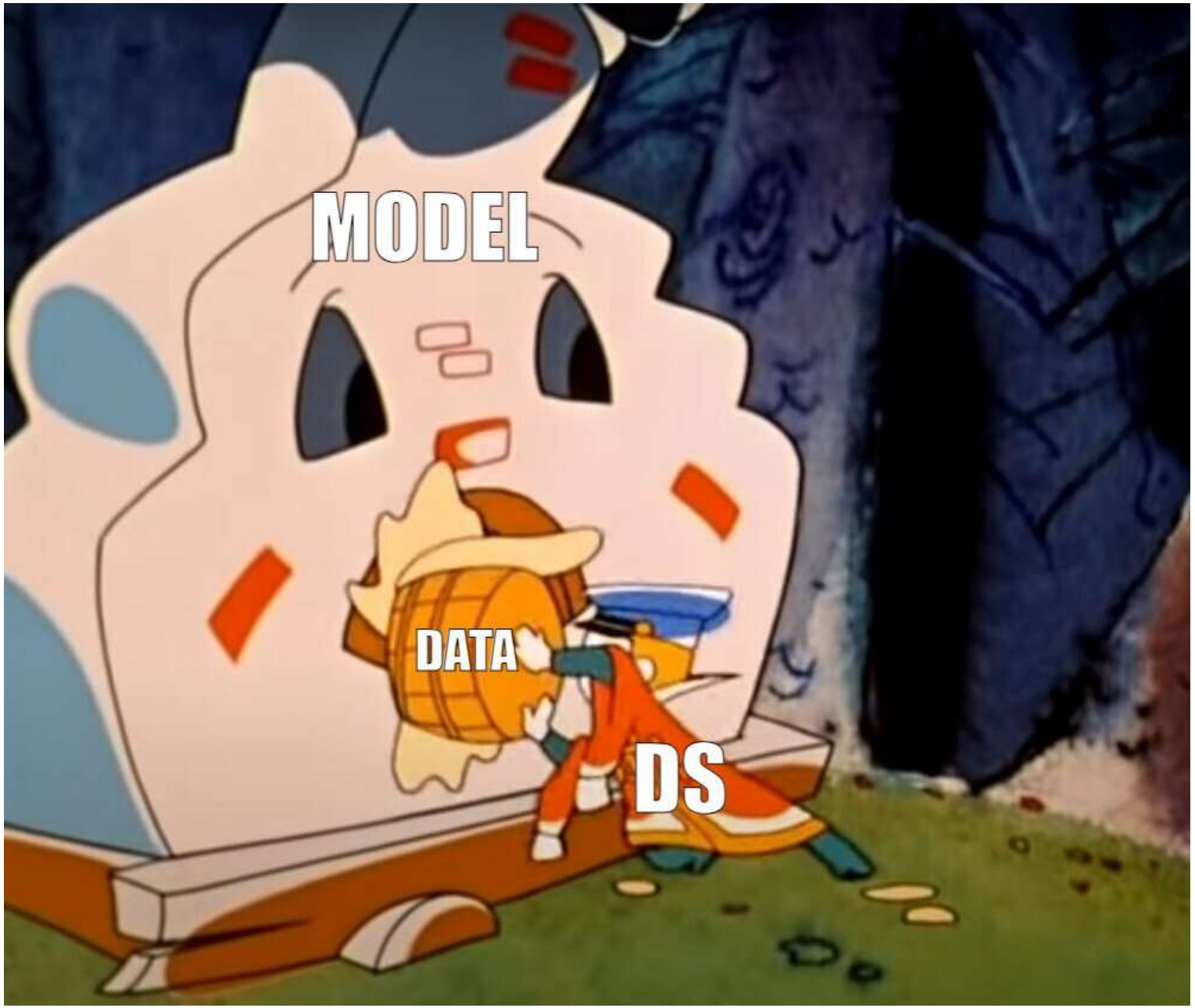
Тверской государственный технический университет

Интеллектуальные информационные системы

Отбор признаков и понижение размерности

Материалы курса доступны по ссылке:
https://github.com/AndreyShpigar/ML-course

2024 г.



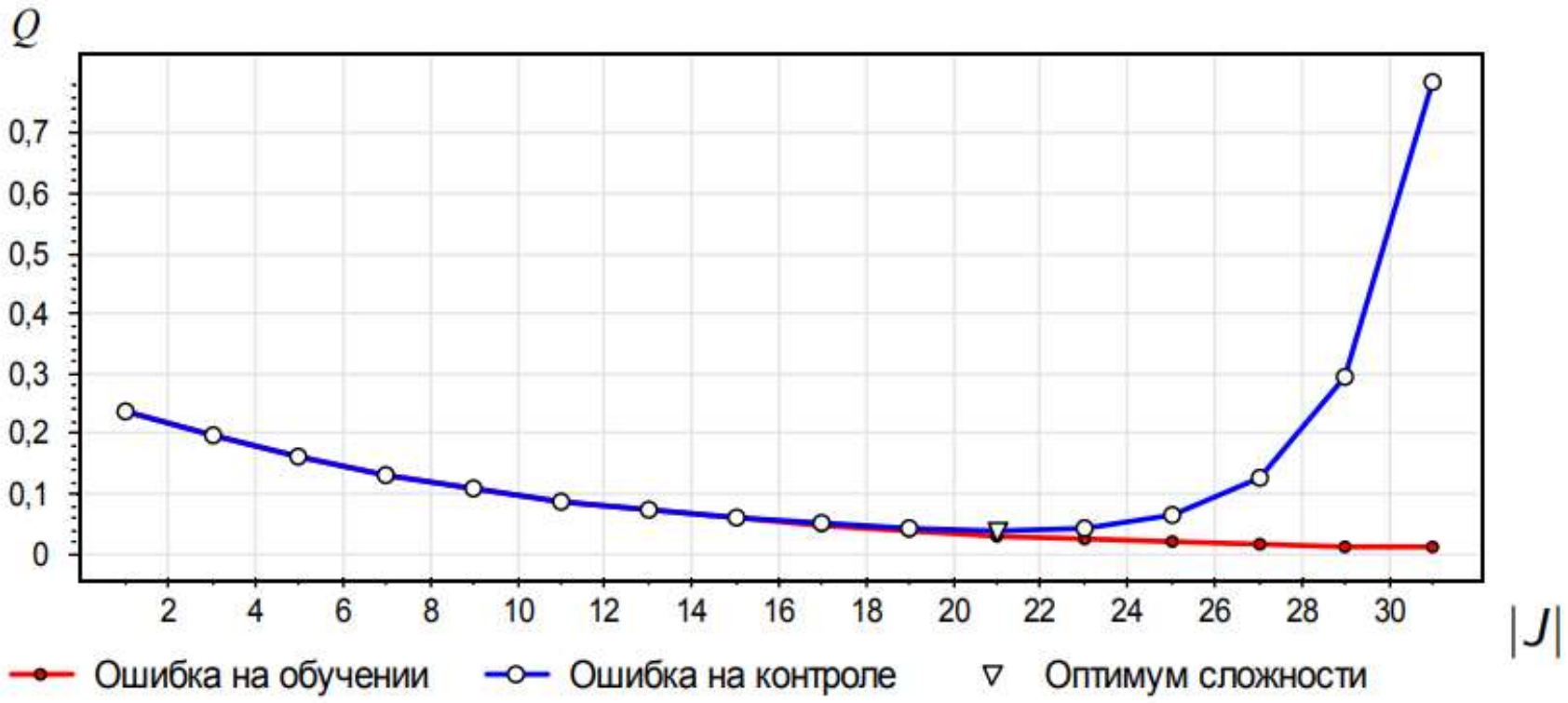
«Засунем всё в модель – и так сойдет»

Мотивация отбора признаков

- Признак может быть *неинформативным* – признак никак не влияет на целевую переменную
- Признак может быть *зависимым* – информация о признаке содержится в других признаках
- Признак может быть сильно *зашумленным* – его использование может повышать риск ошибки
- Минимизация числа признаков позволяет уменьшить затраты *времени и ресурсов* – особенно важно при наличии ограничений на инференс модели
- *Понижение размерности* может уменьшать переобучение
- Меньшее число признаков повышают интерпретируемость модели

Критерий отбора признаков

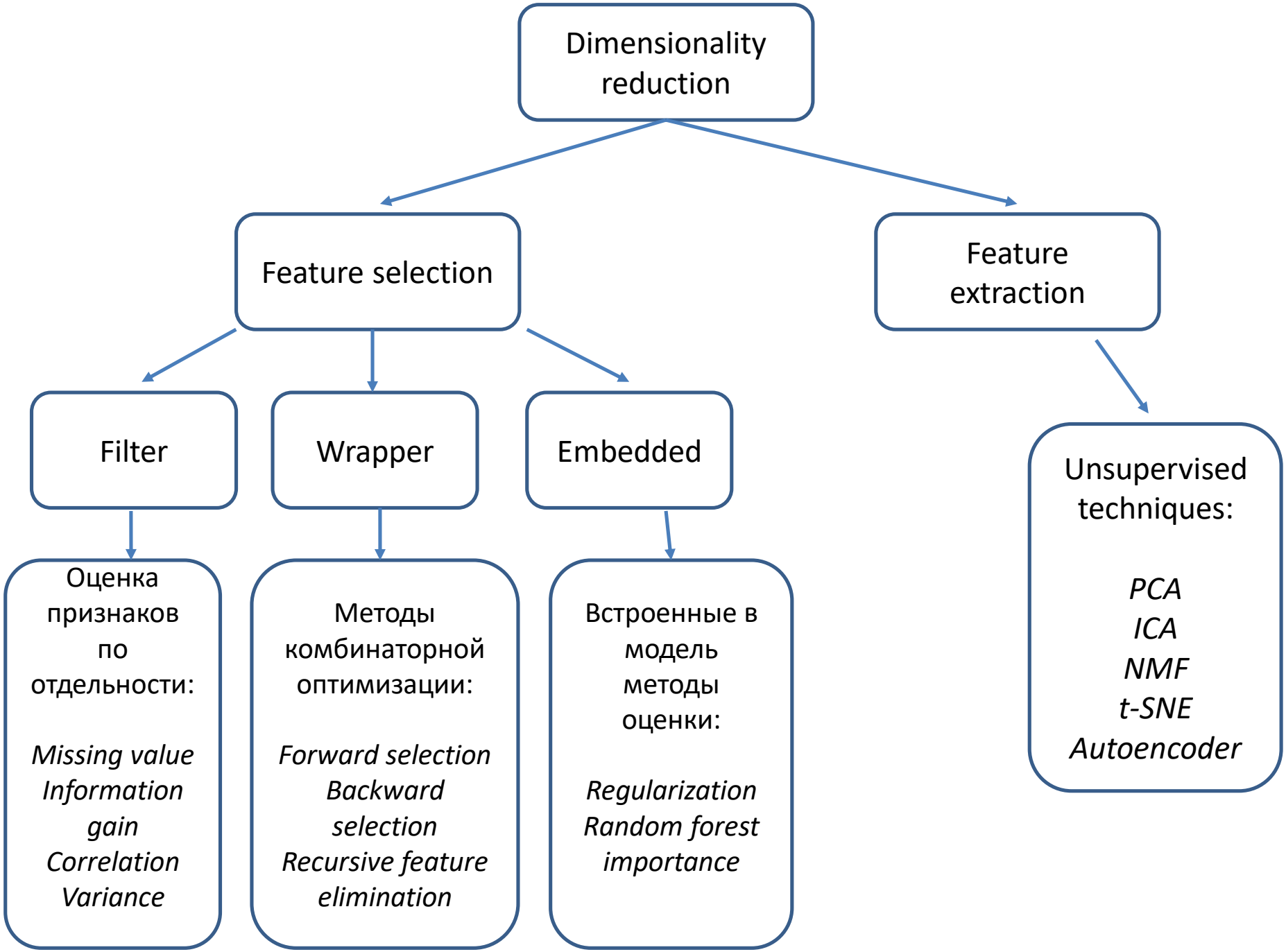
Внутренний критерий и внешний критерий:



Источник: http://www.machinelearning.ru/wiki/images/d/de/Voron-ML-Quality-slides.pdf

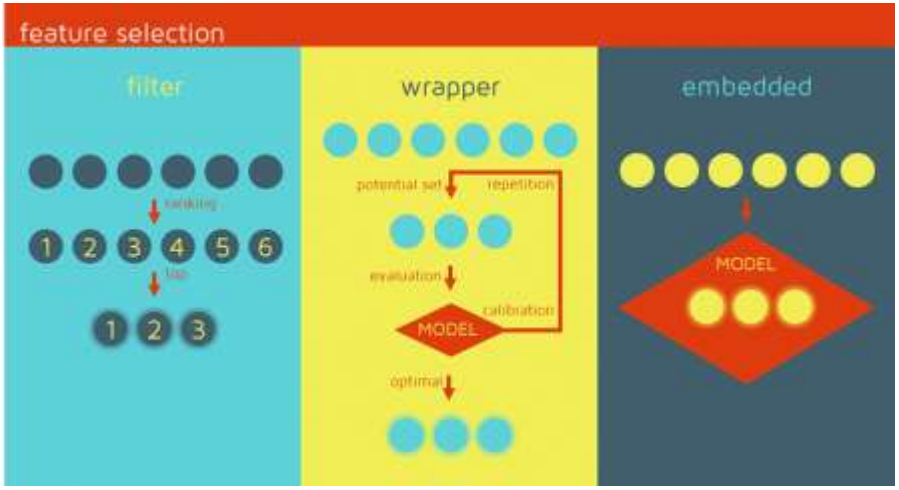
Внешние эмпирические критерии:

1. Проверка на отложенных данных (hold-out)
2. Кросс-валидация
3. Скользящий контроль (LOO)
4. Непротиворечивость моделей
5. Устойчивость модели при малых изменениях данных
6. Согласованность долгосрочных и краткосрочных прогнозов


Dimensionality reduction
Feature selection
Feature extraction
Filter
Wrapper
Embedded
Unsupervised techniques:
PCA
ICA
NMF
t-SNE
Autoencoder
Оценка признаков по отдельности:
Missing value
Information gain
Correlation
Variance
Методы комбинаторной оптимизации:
Forward selection
Backward selection
Recursive feature elimination
Встроенные в модель методы оценки:
Regularization
Random forest importance

- *Feature selection*

поиск оптимального подмножества признаков из исходного множества признаков



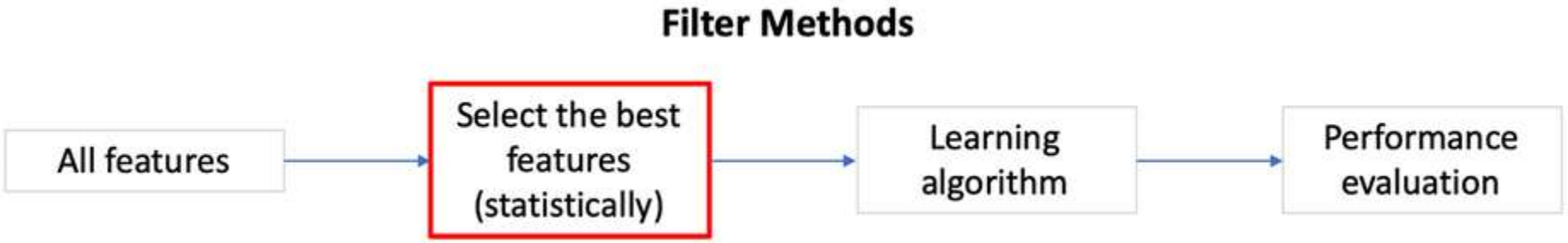
- *Feature extraction*

преобразование исходного признакового пространства в подпространство меньшего размера с максимальным сохранением информации

Filter methods

- *Методы фильтрации признаков* – основаны на оценке информативности каждого признака, с последующей сортировкой и выбором k лучших.

- Не учитывают зависимости между признаками (только парные зависимости)
- Не всегда позволяют оценить «полезность» признака, больше подходят для того, чтобы найти явно «бесполезные» признаки
- Малая вычислительная сложность
- Не зависят от модели
- Чувствительны к выбору критерия отбора (пороговые значения для околонулевой дисперсии, коэффициентов корреляции и т.д.)



- *Missing value ratio* – оцениваем отношение количества пропущенных значений к общему числу наблюдений для каждого признака

✓ Интуитивно понятный метод

✓ Признаки с большим количеством пропущенных значений лучше выкидывать, чем заполнять

× Чувствительность к значению порога

× Пропущенные значения могут быть связаны с целевой переменной

× Выкидывать или заполнять? Заполнять можно статистиками или с помощью моделей, используя пропущенные значения в качестве целевой переменной

ID	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	count
AB101	1.0	0.0	0.0	1.0	9.84	14.395	81.0	NaN	16
AB102	1.0	NaN	0.0	NaN	9.02	13.635	80.0	NaN	40
AB103	1.0	0.0	NaN	1.0	9.02	13.635	80.0	NaN	32
AB104	NaN	0.0	NaN	1.0	9.84	14.395	75.0	NaN	13
AB105	1.0	NaN	0.0	NaN	9.84	14.395	NaN	16.9979	1
AB106	1.0	0.0	NaN	2.0	9.84	12.880	75.0	NaN	1
AB107	1.0	0.0	0.0	1.0	9.02	13.635	80.0	NaN	2
AB108	1.0	NaN	0.0	1.0	8.20	12.880	86.0	NaN	3
AB109	NaN	0.0	0.0	NaN	9.84	14.395	NaN	NaN	8
AB110	1.0	0.0	0.0	1.0	13.12	17.425	76.0	NaN	14

Variable	Missing value ratio
ID	0%
season	20%
holiday	30%
workingday	30%
weather	30%
temp	0%
atemp	0%
humidity	20%
windspeed	90%
count	0%

Источник: https://www.analyticsvidhya.com/blog/2021/04/beginners-guide-to-missing-value-ratio-and-its-implementation/

- *Variance threshold* – удаление признаков с дисперсией ниже заданного порога

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}\left(f_{i,j} - \overline{f_j}\right)^2$$

где $f_{i,j}$ – значение j – го признака на i – m объекте, $\overline{f_j}$ – среднее значение j – го признака

- ✓ Интуитивно понятный метод
- ✓ Легкая реализация
- × Чувствительность к значению порога
- × Чувствительность к шуму
- × Малая, но ненулевая дисперсия может указывать на связь с целевой переменной

- *Корреляция:*

- высокая степень корреляции между признаком и целевой переменной
- низкая степень корреляции между признаками

Можно использовать различные коэффициенты корреляции:

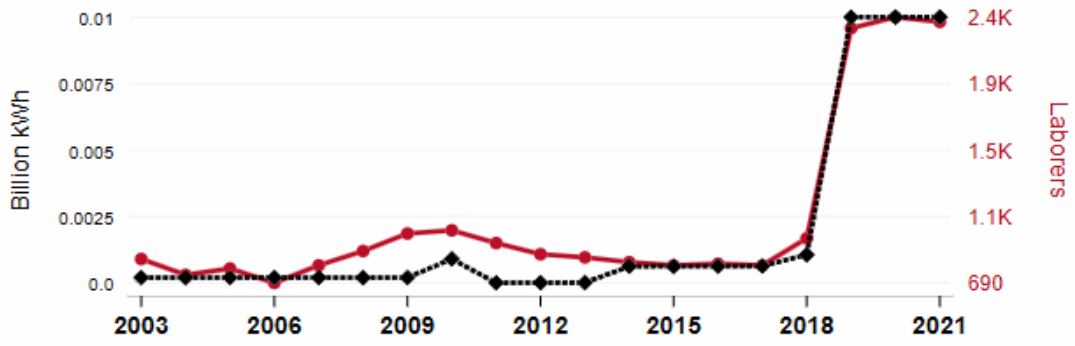
1. Коэффициент корреляции Пирсона (линейный коэффициент)
2. Коэффициент ранговой корреляции Спирмена
3. Коэффициент ранговой корреляции Кендалла
4. Коэффициент корреляции знаков Фехнера

Сам по себе факт корреляционной зависимости не дает основания утверждать, что одна из переменных предшествует или является причиной изменений, или то, что переменные вообще причинно связаны между собой, а не наблюдается действие третьего фактора

Solar power generated in Belize

correlates with

The number of fire inspectors in Florida



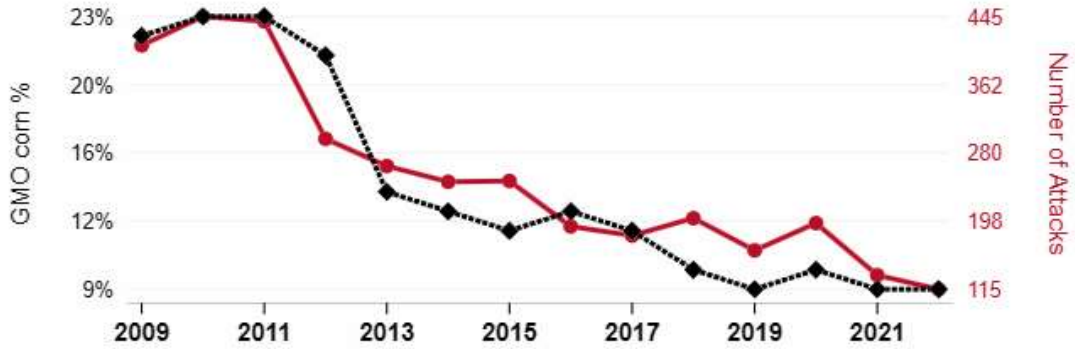
Total solar power generated in Belize in billion kWh · Source: Energy Information Administration

BLS estimate of fire inspectors and investigators in Florida · Source: Bureau of Larbor Statistics

GMO use in corn

correlates with

Pirate attacks globally



Percent of all corn planted in United States that is genetically modified to be herbicide-tolerant (HT), but not insect-resistant (Bt) · Source: USDA

Global Pirate Attack Count · Source: Statista

2009-2022, r=0.948, r^2=0.899, p<0.01 · tylervigen.com/spurious/correlation/2051

Источник: https://www.tylervigen.com/spurious-correlations

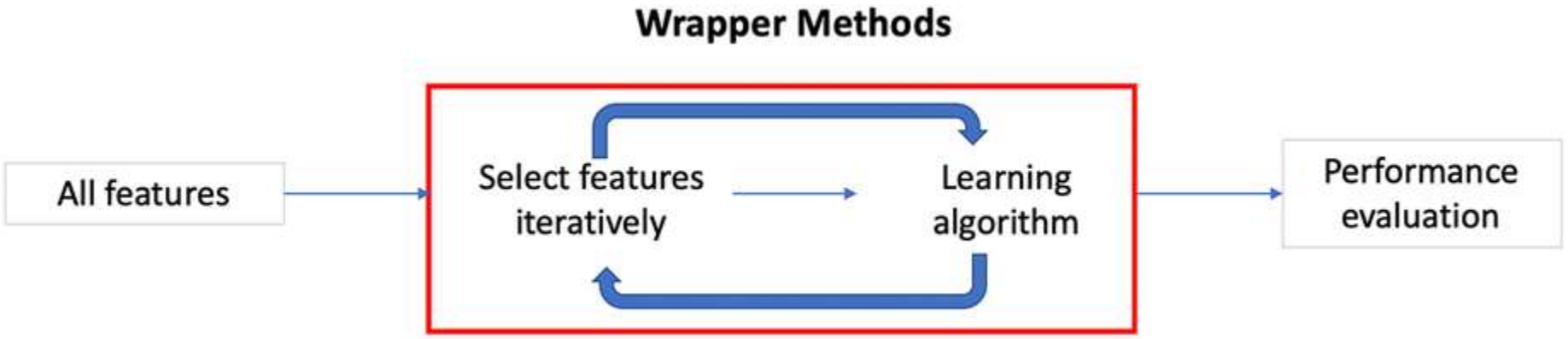
- *Статистические характеристики (univariate feature selection)* – оценка взаимосвязи признаков с целевой переменной на основе статистических тестов. *Взаимная информация – статистическая функция двух случайных величин, описывающее количество информации, содержащееся в одной случайной величине относительно другой*

1. F-test (ANOVA) – анализ дисперсии. Большой F-score указывает на большую разницу в средних значениях между группами и на сильную связь между признаком и целевой переменной.
2. T-score – оценка статистической значимости между средними значениями двух групп (бинарная классификация) $$R_j = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$
3. Chi square – оценка ожидаемой вероятности наступления событий A и B, если мы предполагаем, что они независимы (признак – таргет). Вычисляем насколько эта оценка отличается от ожидаемой $$\chi^2 = \sum \frac{(observed - expected)^2}{expected}$$
4. Information Gain – оценка на основе прироста информации - снижение энтропии целевой переменной при использовании выбранного признака $$IG(Y|X) = H(Y) - H(Y|X)$$

Wrapper methods

- *Методы комбинаторной оптимизации* – переборные методы, основаны на выборе оптимального подмножества признаков из множества признаков

- Учитывают зависимости между признаками
- Часто обеспечивают хорошее качество
- Высокая вычислительная сложность (особенно при больших данных)
- Зависят от модели – подмножества перебираются для выбранной модели. Это подмножество не обобщается на все модели.



- *Алгоритм полного перебора (exhaustive feature selection)* – использует все возможные подмножества признаков для выявления лучшего подмножества

✓ Простота реализации

✓ Гарантированный результат

× Очень высокая вычислительная сложность – в общем случае $O(2^n)$

× Чем больше перебирается вариантов, тем больше переобучение

- *Прямые алгоритмы (step forward feature selection, алгоритмы жадного добавления)* – основаны на постепенном добавлении новых признаков в модель

1. Строится модель, содержащая один признак x_1
2. К признаку x_1 по очереди добавляются оставшиеся признаки x_j, строятся модели, содержащая два признака. На основе выбранной метрики оценивается качество всех моделей и выбирается лучшая модель из двух признаков
3. Итеративное повторение для N > 2 признаков
4. Критерии остановки:

- Минимальное улучшение качества
- Ограничение на количество признаков (максимальное)

- *Обратные алгоритмы (step backward feature selection, алгоритмы жадного удаления)* – основаны на постепенном удалении признаков из модели

1. Каждый признак x_i удаляется из множества признаков X, для оставшихся признаков строятся модели. На основе выбранной метрики оценивается качество всех моделей и выбирается лучшая модель, содержащая все признаки, кроме одного
2. Из подмножества X^{-1} по очереди удаляется по одному из оставшихся признаков x_j. Строятся модели модели на основе исходного множества без двух признаков. Оценивается качество всех моделей, выбирается лучшая модель, содержащая все признаки, кроме двух
3. Итеративное повторение для N > 2 признаков
4. Критерии остановки:

- Минимальное улучшение качества
- Ограничение на количество признаков (минимальное)

Другие алгоритмы добавления или удаления признаков

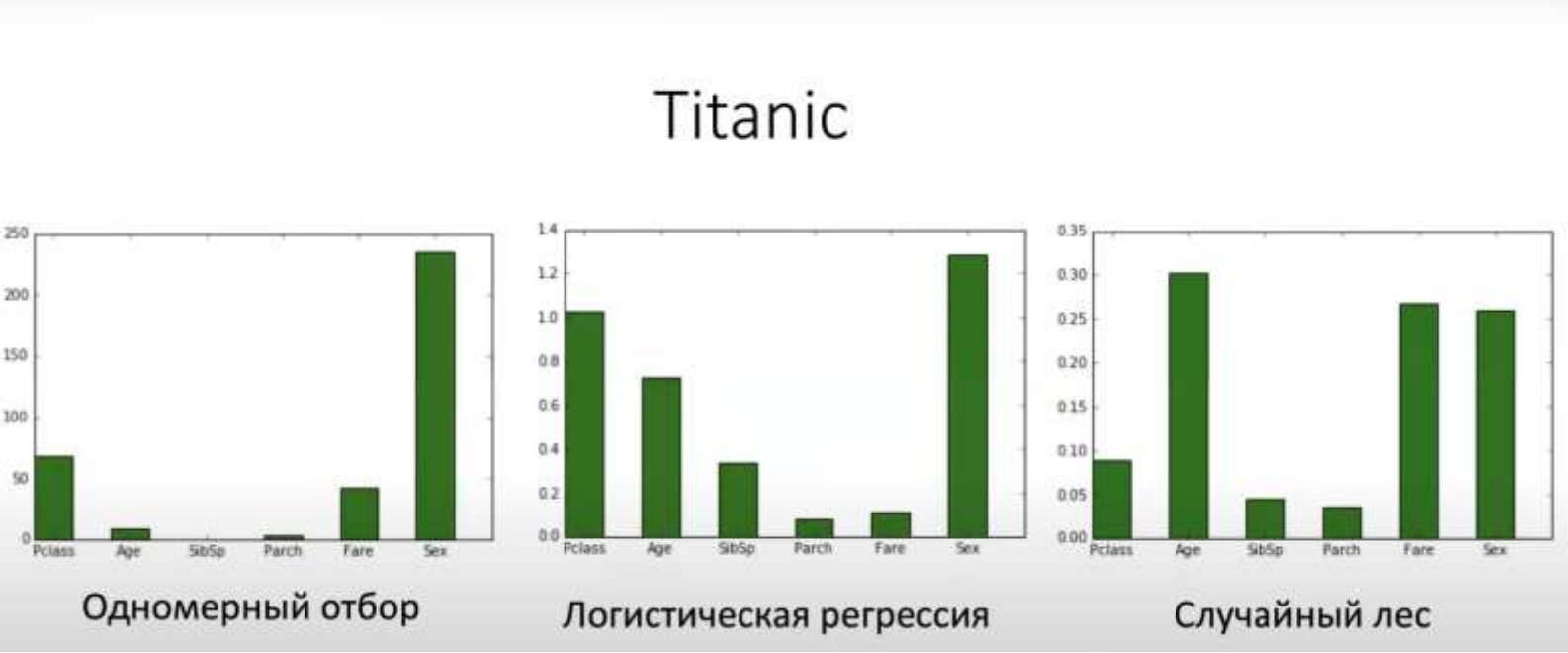
1. Recursive feature elimination – ранжирование признаков с рекурсивным исключением. Признакам назначаются веса, затем наименее значимые признаки удаляются
2. Генетические алгоритмы
3. Стохастические методы отбора
4. Permutation feature importance - определение важности признаков на основе перемешивания данных. Пример – алгоритм Boruta – метод-обертка на основе случайного леса. Создает теневые признаки путем перемешивания значений в каждом столбце, затем оценивает значимость признака.

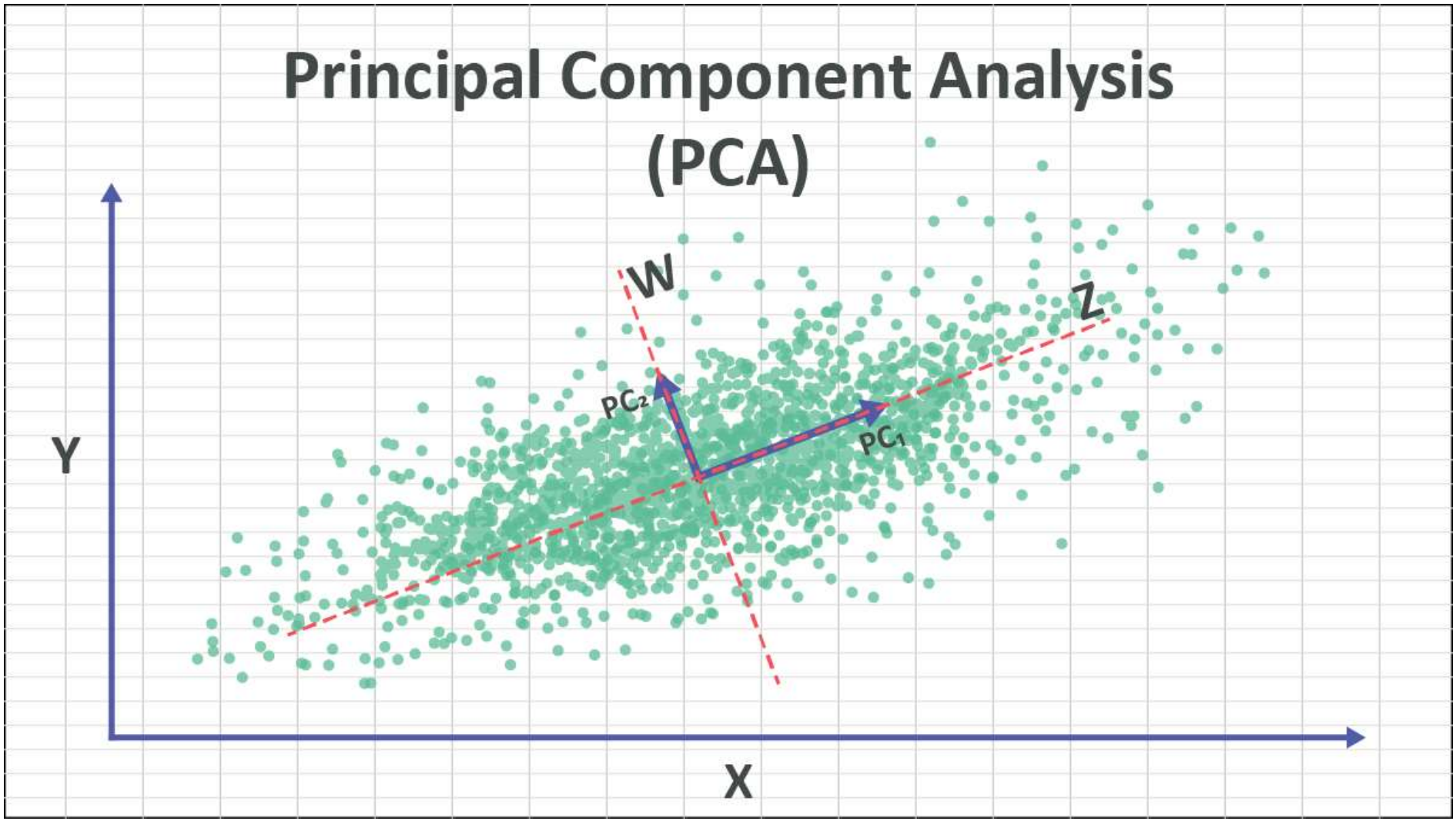
Embedded methods

- *Встроенные методы отбора признаков* – отбор признаков на основе оценки важности признаков в процессе построения модели

➢ Оценка значимости признаков выполняется вместе с обучением модели

➢ Зависят от модели – разные модели могут давать разную оценку значимости признаков




Principal Component Analysis
(PCA)
W
Z
PC2
PC1
Y
X

Метод главных компонент

Основан на предположении о линейности отношений данных и их проекции на подпространство ортогональных векторов, в которых дисперсия будет максимальной. Такие вектора называются главными компонента и определяют направления наибольшей информативности данных

Пусть $X \in R^{l \times D}$- матрица «объекты-признаки», l – число объектов, D – число признаков. Задача понижения размерности до d. Данные центрированы, т.е. среднее значение в каждом столбце матрицы X равно нулю. Будем искать главные компоненты $u_1, \dots, u_d \in R^D$, которые удовлетворяются следующим требованиям:

1. Они ортогональны: $\langle u_i, u_j \rangle = 0, i \neq j$
2. Они нормированы: $\|u_i\|^2 = 1$
3. При проецировании выборки на компоненты получается максимальная дисперсия среди всех возможных способов выбрать d компонент

Реализация алгоритма PCA на основе SVD

1. Центрирование данных и определение числа компонент
2. Сингулярное разложение матрицы данных M = USV. Где сингулярные значения — это квадратные корни собственных значений ковариационной матрицы правая сингулярная матрица V будет соответствовать собственным векторам ковариационной матрицы данных, а левая U будет являться проекцией исходных данных на главные компоненты, определённые матрицей V. Таким образом, сингулярное разложение также позволяет выделить главные компоненты, но без необходимости в вычислении ковариационной матрицы. Помимо того, что такое решение более эффективно, оно считается более численно стабильным, поскольку не требует вычисления ковариационной матрицы напрямую, которая может быть плохо обусловлена в случае сильной корреляции признаков.
3. В матрице U находим максимальные по модулю элементы в каждом столбце, извлекаем знаки и умножаем матрицу U на эти знаки, чтобы гарантировать детерминированный вывод
4. Объясненная дисперсия для каждой главной компоненты вычисляется как возведенные в квадрат соответствующие сингулярные значения, разделенные на количество сэмплов

Дополнительные возможности PCA:

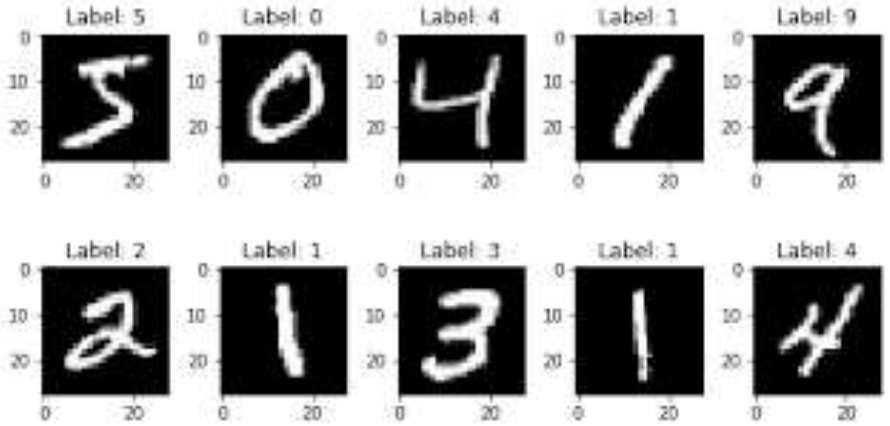
- *Коэффициент объясненной дисперсии каждой главной компоненты* – указывает долю дисперсии датасета, лежащей вдоль оси каждой главной компоненты
- *Восстановление данных и оценка ошибки восстановления (reconstruction error)*
- *Kernel PCA* – использует ядерные функции для нелинейных проекций

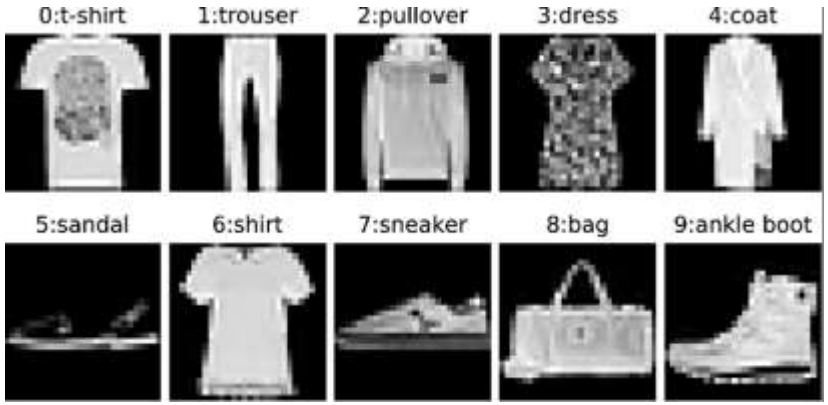
Достоинства и недостатки PCA:

- ✓ Понижение размерности, устранение корреляции между признаками
- ✓ Существуют эффективные реализации
- ✓ Можно использовать для визуализации
- × Ограничения на линейность
- × Чувствительность к масштабу данных
- × Трудно интерпретировать

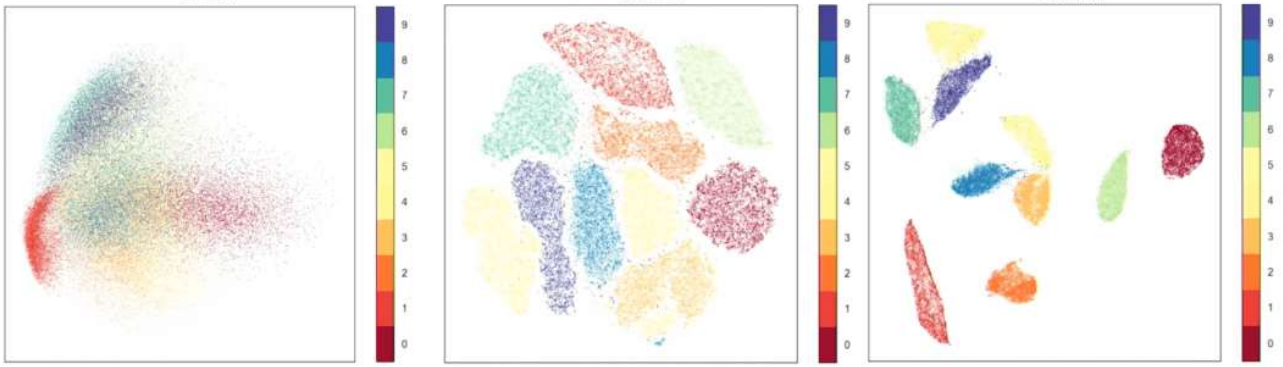
Другие методы понижения размерности

- *Locally Linear Embedding (LLE)* – алгоритм создания линейных комбинаций каждой точки из её соседей с последующим восстановлением этих комбинаций в пространстве более низкой размерности, что позволяет сохранить нелинейную геометрию данных и быть полезным для некоторых задач, где глобальные свойства менее важны. С другой стороны, такой подход имеет высокую вычислительную сложность и может быть чувствителен к шуму.
- *t-SNE (t-Distributed Stochastic Neighbor Embedding)* — алгоритм, который преобразует сходства между данными в вероятности и в дальнейшем пытается минимизировать расхождение между распределениями вероятностей в пространстве высокой и низкой размерности. t-SNE эффективен при визуализации данных высокой размерности, однако может искажать глобальную структуру данных, поскольку не учитывает линейные зависимости, а лишь их близость в исходном пространстве.
- *UMAP (Uniform Manifold Approximation and Projection)* — ещё один алгоритм, подходящий для визуализации данных, который основан на идеи, что данные лежат на некотором однородном многообразии, которое можно аппроксимировать с помощью графа соседей. Такой подход учитывает глобальную структуру данных и позволяет лучше адаптироваться к различным типам данных, а также лучше справляться с шумом и выбросами, чем t-SNE.
- *Автоэнкодеры* — тип нейронных сетей, основанный на обучении кодировщика преобразовывать входные данные в низкоразмерное представление, с последующим обучением декодера восстанавливать исходные данные из этого представления. Autoencoders могут также использоваться для сжатия данных, удаления шума и многих других целей.





MNIST Digits



Fashion MNIST