



# ML Advanced

**Featuretools - а вы что, за меня и  
признаки придумывать будете?**

• REC

Проверить, идет ли запись

Меня хорошо видно  
&& слышно?



Ставим "+", если все хорошо  
"-", если есть проблемы

Тема вебинара

# ML Advanced Featuretools

Игорь Стурейко



Teamlead, главный инженер проекта – НИИгазэкономика

Опыт:

Более 15 лет занимался прикладной математикой и математическим моделированием (Data Scientist) (Python, C++) в НИИ ПАО Газпром

Анализ временных рядов, эволюционное развитие сложных систем

+7 (916) 156-07-82 (whatsapp)

@stureiko (TG)

# Правила вебинара



Активно  
участвуем



Off-topic обсуждаем  
в учебной группе



Задаем вопрос  
в чат или голосом



Вопросы вижу в чате,  
могу ответить не сразу

## Условные обозначения



Индивидуально



Время, необходимое  
на активность



Пишем в чат



Говорим голосом

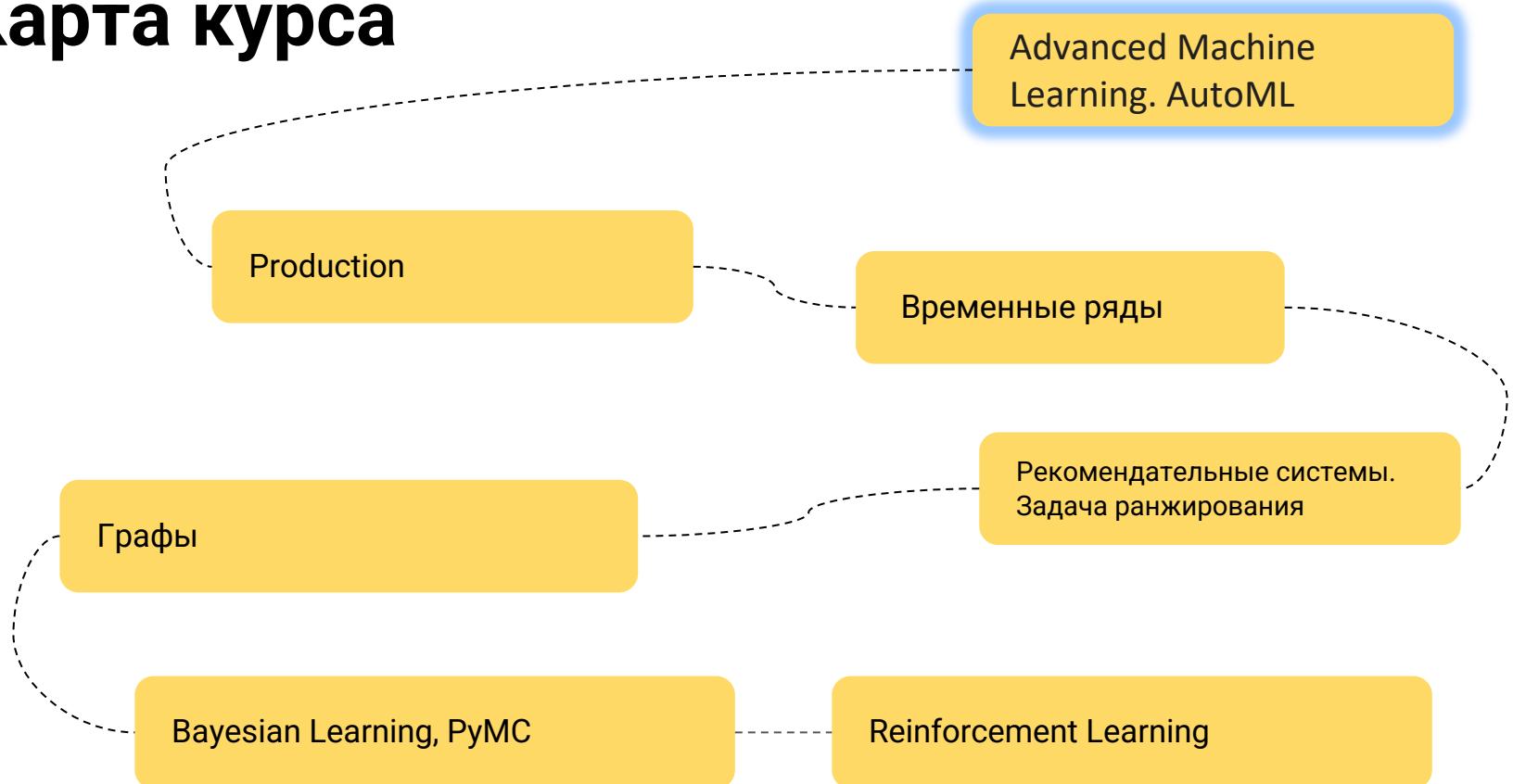


Документ



Ответьте себе или  
задайте вопрос

# Карта курса



# Цели вебинара

К концу занятия вы сможете

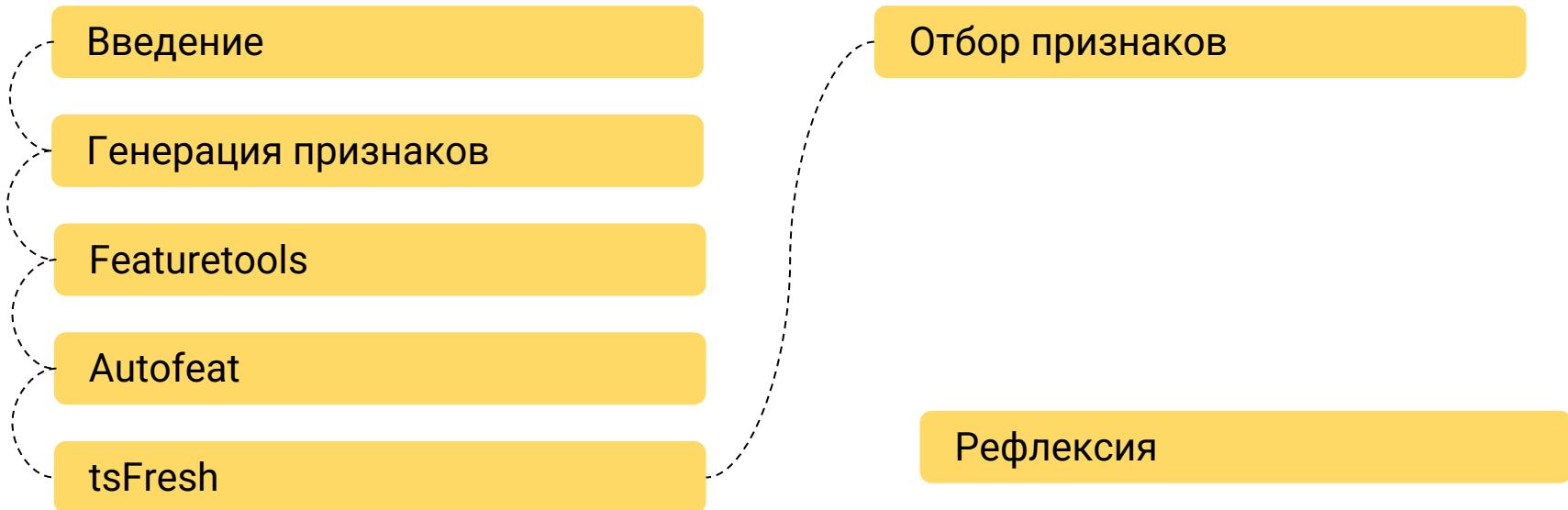
1. Познакомиться с библиотекой Featuretools
  2. Понять как автоматически проводить Feature Engineering
-

# Смысл

Зачем вам это знать

1. Использовать в соревнованиях и хакатонах
  2. Использовать в работе
- 
-

# Маршрут вебинара



# Feature Generation

# Feature Generation



## Traditional

vs.

## Automated

Выполняется специалистами по данным

В значительной степени полагается на выбранную модель и опыт в предметной области

Признаки разработаны методом проб и ошибок.

Выполняется специалистами по данным

Не требует опыта в предметной области

Сокращает время проб и ошибок

Увеличивает ширину и глубину поиска лучших признаков

Может сопровождаться автоматическим подбором модели

# Feature Generation

The Featuretools logo consists of a stylized orange icon on the left followed by the word "Featuretools" in a large, orange, sans-serif font.The tsfresh logo features a black coordinate system with a green line graph showing three data points. Below it, the word "tsfresh" is written in a bold, black, sans-serif font, with "ts" in black and "fresh" in green.The TSFEL logo consists of two blue triangles pointing upwards, with the word "TSFEL" in a bold, black, sans-serif font centered below them.The tsflex logo features a teal circle with a red ring and a black clock face, followed by the word "tsflex" in a bold, red, sans-serif font.

[cod3licious/  
autofeat](#)



Linear Prediction Model with Automated Feature Engineering and Selection Capabilities

4  
Contributors

13  
Issues

438  
Stars

59  
Forks



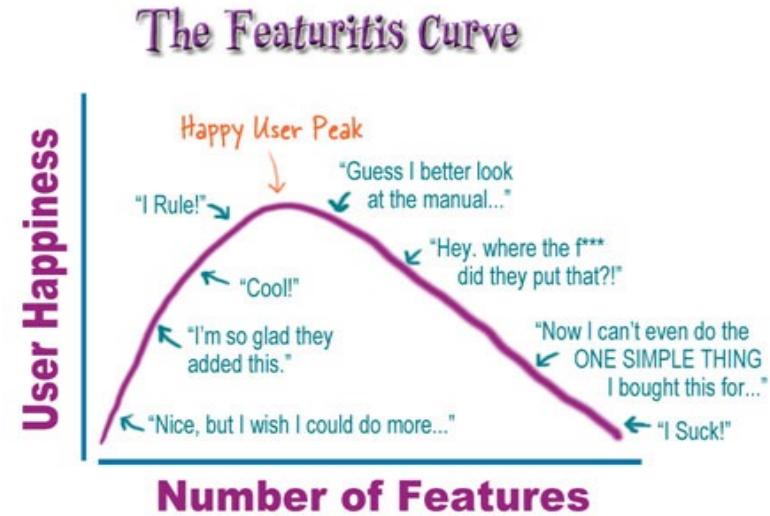
# Как построить хорошие признаки?

## Генератор признаков

- Генераторы доменных признаков
- Генераторы признаков общего назначения

## Алгоритм отбора признаков

- Степень «хорошести» признаков связана как с данными, так и с моделью
- Необходимо сочетать с эффективным выбором модели



# Доменные признаки

Текстовые данные

- Bag of words, TF-IDF, Word2Vec embeddings

Картинки

- Цвет, текстура, вейвлет-коэффициенты, масштабно-инвариативные признаки (SIFT), признаки из нейронной сети

Временные ряды

- Спектральные признаки, motifs, shapelets, discords

# Признаки общего назначения

## Преобразования одной переменной

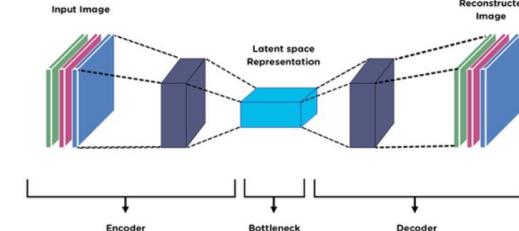
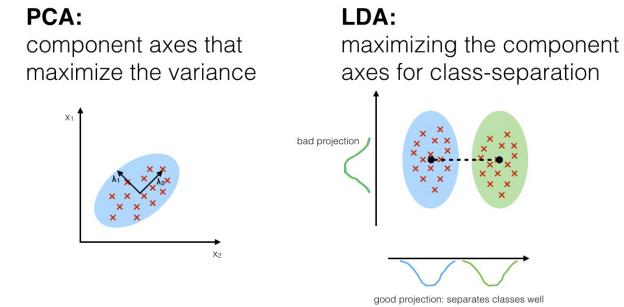
- Логарифмический, экспоненциальный, частотный, однократное кодирование, нормализация

## Комбинация двух переменных

- Сумма, разность, деление, произведение

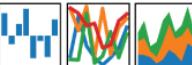
## Многомерные и модельные методы

- Unsupervised learning
  - PCA, случайные проекции, distance/cluster based features
- Supervised learning
  - Линейный дискриминантный анализ (LDA), supervised dictionary learning
- Deep learning
  - Автокодировщики, промежуточные уровни обученных сетей



# Featuretools

фреймворк, используемый для автоматизации процесса генерации признаков.  
Работает путем преобразования транзакционных и реляционных наборов данных в матрицы  
признаков для машинного обучения.

pandas   
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

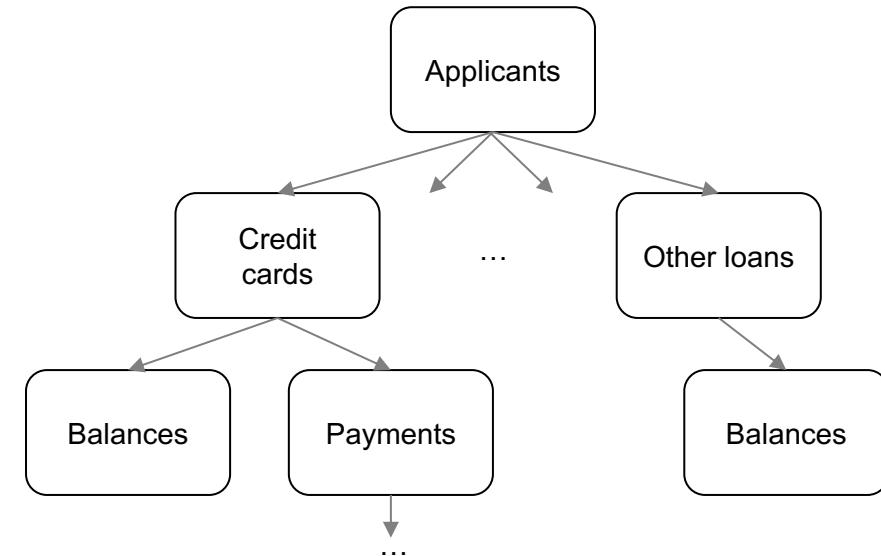
Prepare Your Data

 Featuretools

Feature Engineering



Learn from Your Data



# Основные понятия

## Entity

Entity это таблица (dataframe). У каждой entity обязательно должен быть индекс.

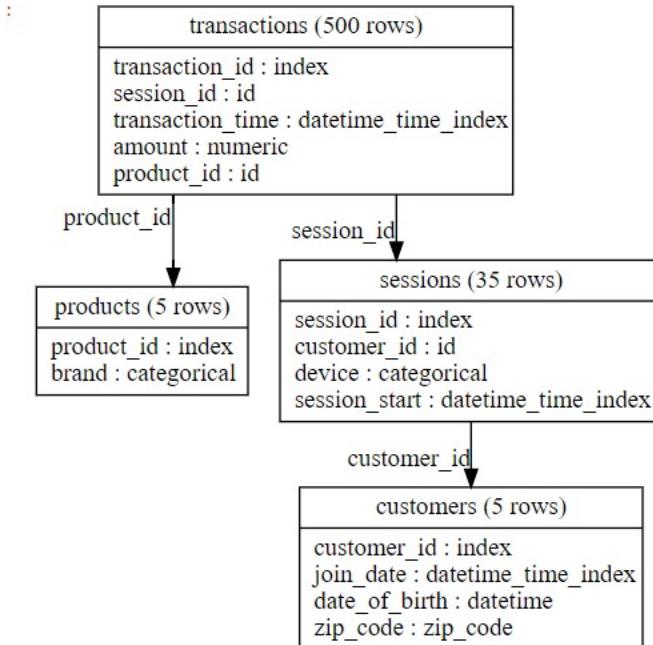
## Entityset

Entityset это структура (словарь) состоящая из множества отдельных entity и связей между ними.

## Relationship

Relationship это отношения между таблицами в Entityset. Отношения родитель-потомок (один ко многим), для одного родителя может быть несколько потомков.

```
: es = ft.demo.load_mock_customer(return_entityset=True)
es.plot()
```



# Основные понятия

## Feature Primitives

Это базовые операции, которые используются для формирования новых признаков, могут применяться к наборам данных и могут накладываться друг на друга для создания сложных признаков.

Две основные группы:

### Aggregations

- Last, num\_unique, max, std, time\_since\_last

### Transformations

- Absolute, isin, numwords

# Глубокий синтез признаков (DFS)

**DFS** – это автоматизированный метод создания признаков для реляционных и многотабличных данных. DFS работает, используя концепцию объединения примитивов для получения более глубоких признаков.

**Глубина признака** – это количество примитивов, необходимых для его создания. Признак, основанный на одном преобразовании или агрегации будет иметь глубину равную единице, признак который создается из двух примитивов будет иметь глубину два.

# Featuretools

## Преимущества:

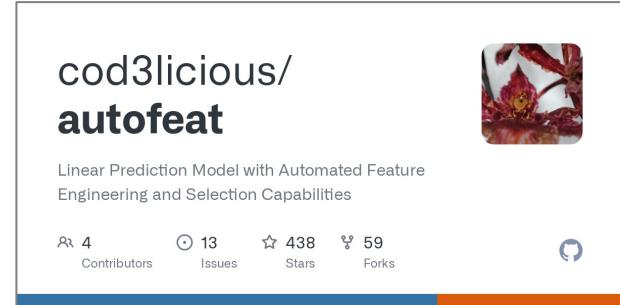
- популярен, много материалов для изучения.
- Примитивы можно создавать самому.
- Позволяет учитывать временные структуры (временные ряды)

## Недостатки:

- Может создавать очень много признаков.
- Нельзя использовать неструктурированные данные.

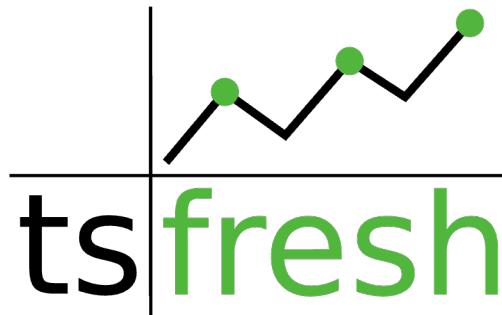
# AutoFeat

**AutoFeat** – библиотека Python, которая предоставляет модели линейной регрессии и классификации в стиле scikit-learn с возможностями автоматического создания и отбора признаков.

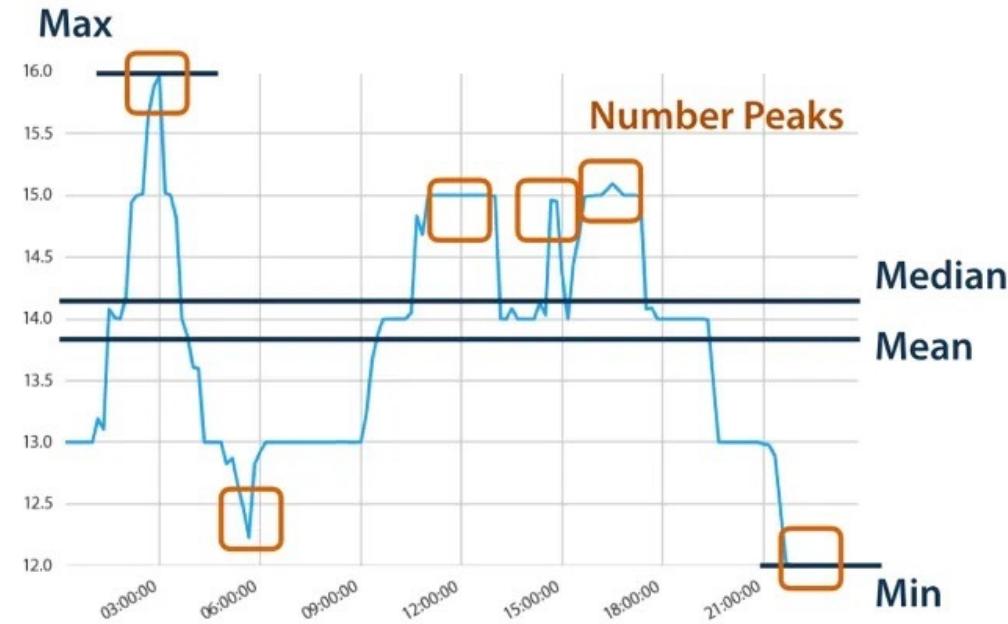


Нелинейные признаки создаются в чередующемся многоэтапном процессе, сначала применяя нелинейные преобразования (могут быть выбраны пользователем) к элементам (например  $\log(x)$ ,  $\sqrt{x}$ ,  $|x|$ ,  $\exp(x)$ ,  $2x$ ,  $\sin(x)$ ), а затем комбинируя их с различными операторами.

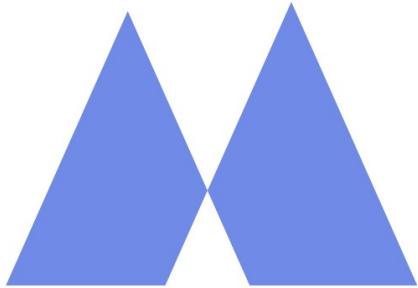
**AutoFeat** также позволяет определять единицы входных переменных, чтобы предотвратить создание физически бессмысленных признаков.



Автоматически извлекает 100 признаков из данных временного ряда, которые описывают как основные, так и сложные характеристики временного ряда (например количество пиков, среднее значение, максимум и минимум, статистика симметрии по обращению времени и т.д)



# Работа с временными рядами



**TSFEL**

[tsFel](#)



**tsflex**

[tsFlex](#)

# Вопросы?



Ставим “+”,  
если вопросы есть



Ставим “-”,  
если вопросов нет

# Feature Selection

# Зачем отбирать признаки?

Легче интерпретировать

Быстрее учиться

Меньше переобучение → лучше обобщение

Мультиколлениарность

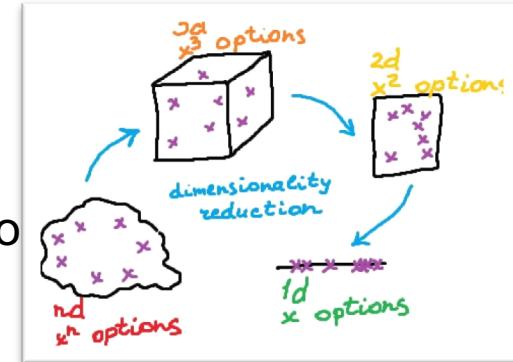


# Feature Selection vs. Dimensionality Reduction

**FeatureSelection** – процесс, который выбирает и исключает некоторые признаки, не изменяя их.



**Dimensionality Reduction** изменяет или преобразует объекты в более низкую размерность. По сути, уменьшение размерности создает совершенно новое пространство признаков, которое описывает примерно то же, что и исходное, но меньше по размерам.

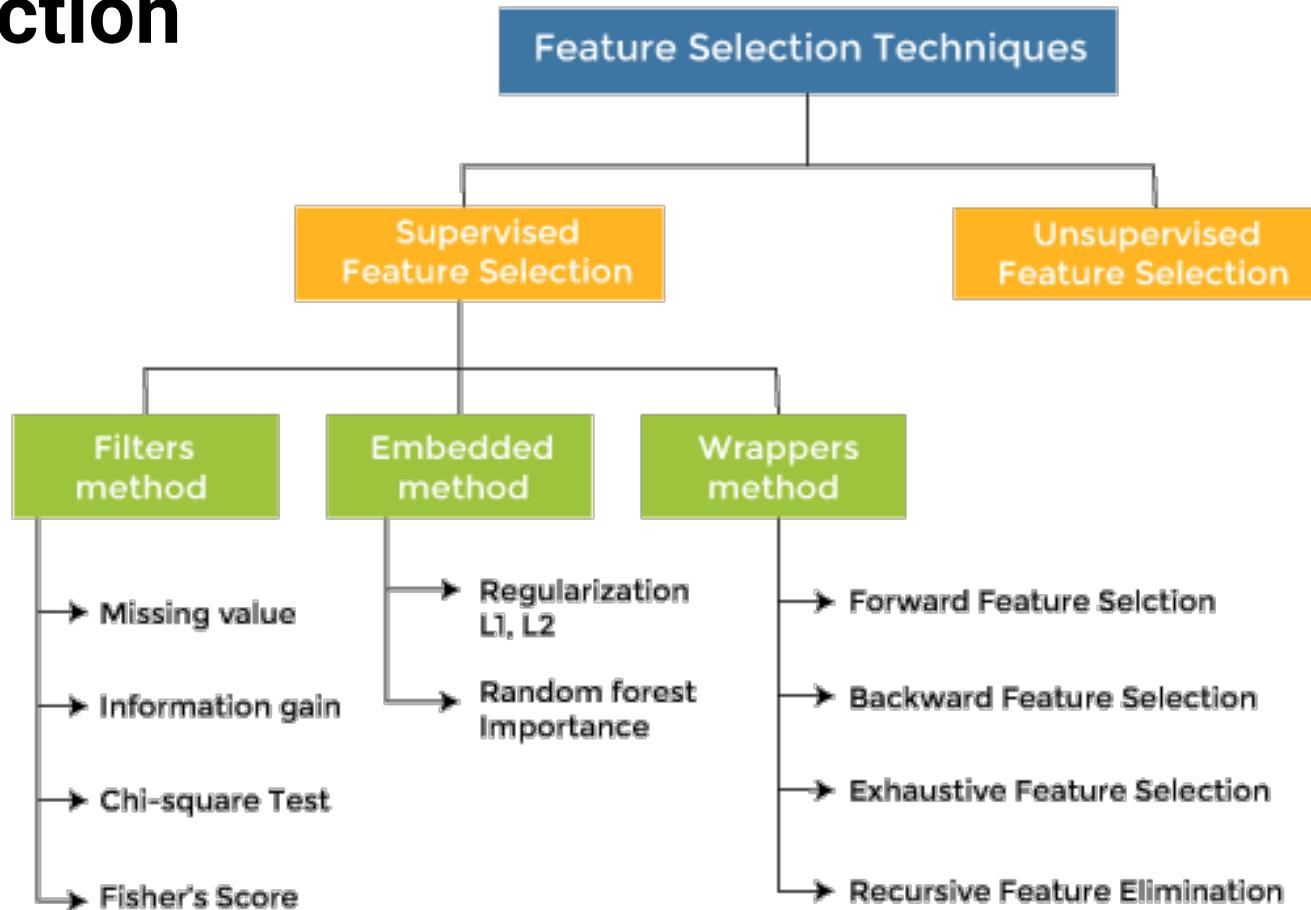


# Feature selection

**Filter methods** – используют только характеристики признаков

**Wrapper methods** – для выбранного метода ML находят наилучший набор признаков, сравнивая качество модели по выбранной метрики

**Embedded methods** – отбор признаков происходит как часть процесса построения модели



# Filter Methods

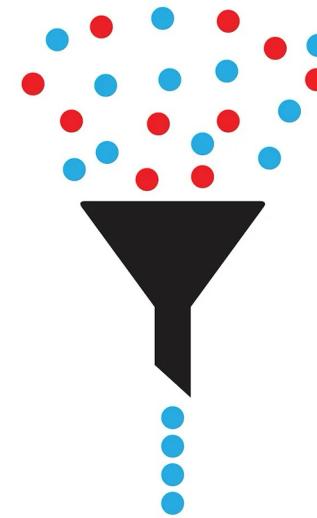
## Basic Filter methods

- Constant Features
- Quasi-Constant Features
- Duplicated Features

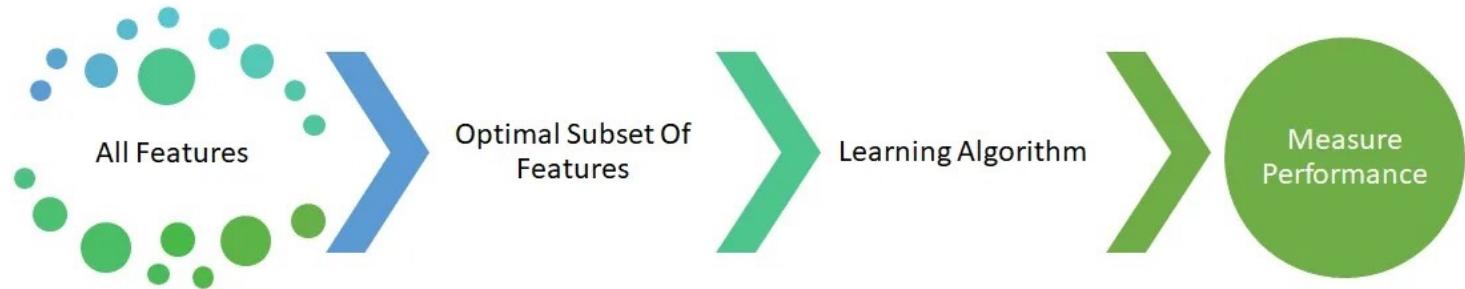
## Correlation Filter methods

## Statistical & Ranking Filter methods

- Mutual information
- Chi-squared Score
- Univariate ROC-AUC / RMSE



# Wrapper Methods: Process



**Выбор подмножества признаков** – используя один из алгоритмов поиска, выбираем подмножество признаков

**Обучение модели** – обучаем на выбранном подмножестве

**Оценивание качества модели**

**Повторение** – повторяем до достижения заданного критерия качества модели

# Wrapper Methods: Methods

**Forward Feature Selection** – начинает с пустого множества и добавляет один признак за итерацию.

**Backward Feature Elimination** – начинает с полного набора и удаляет один признак за итерацию.

**Exhaustive Feature Selection** – оценивает все возможные комбинации признаков.

**Sequential Floating** – попеременно проводит forward и backward feature selection.

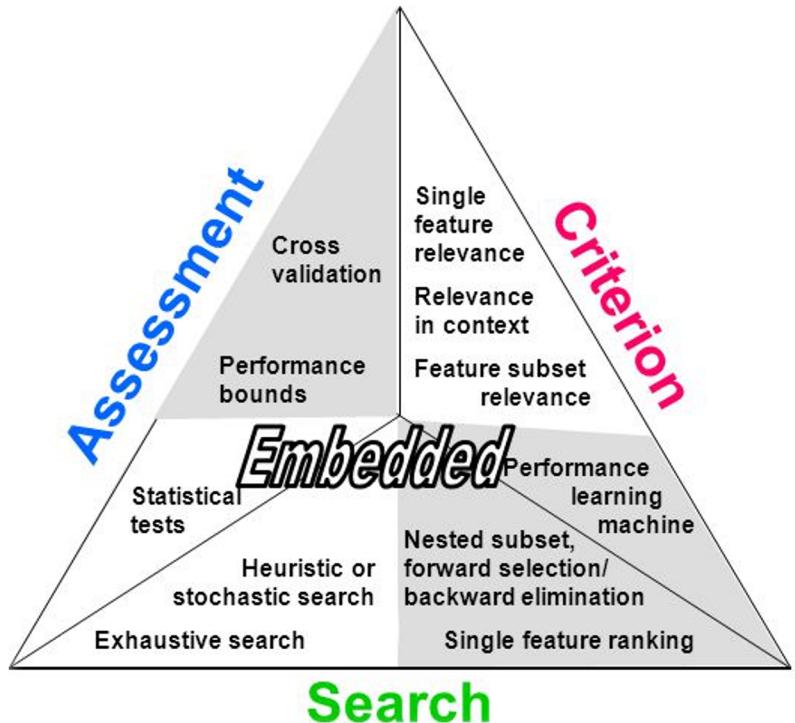
# Embedded Methods: Advantages

Учитывают межпризнаковое  
взаимодействие как wrapper методы

Быстры как filter методы

Точнее filter методов

Находят лучший набор признаков  
для своего алгоритма



# Embedded Methods: Process

Обучаем модель на всех признаках

Подсчитываем feature importance модели и оцениваем важность каждого из признаков

Удаляем бесполезные по feature importance признаки

# Embedded Methods: Methods

## L1-regularization

### L1 Regularization

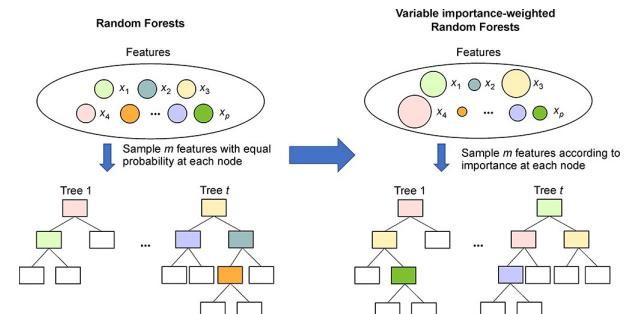
$$\text{Modified loss function} = \text{Loss function} + \lambda \sum_{l=1}^n |W_l|$$

### L2 Regularization

$$\text{Modified loss function} = \text{Loss function} + \lambda \sum_{l=1}^n W_l^2$$

## Tree-based Feature importance

- Random Forest
- Gradient boosting (LightGBM, CatBoost)



# Hybrid Methods: Recursive Feature Elimination

- Обучаем модель на всех признаках. Считаем feature importance и оцениваем качество по выбранной метрике.
- Удаляем наименее важный признак и повторно обучаем модель на оставшихся.
- Если значение метрики упало, значит признак важен. Иначе удаляем его.
- Повторяем пока не останется признаков для удаления.

# Advanced Methods: Permutation importance

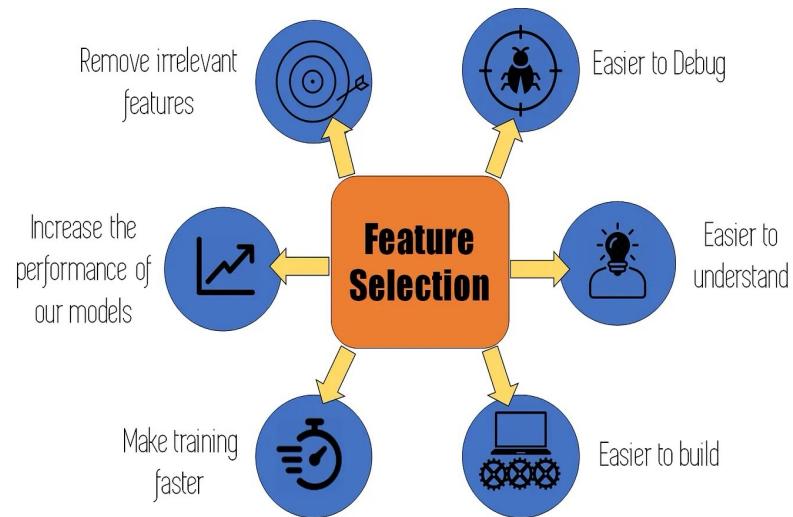
Оцениваем важность признака, измеряя увеличение ошибки прогнозирования модели после перестановки его значений. Таким образом мы нарушаем связь признака и таргета.

Признак является «важным», если перетасовка его значений увеличивает ошибку модели. В этом случае модель полагается на признак при прогнозировании.

Признак считается «не важным», если при перетасовке его значений ошибка модели остается неизменной.

# Основные мысли

1. Генерировать признаки руками
2. Генерировать автоматически
3. TsFresh for TS
4. Отфильтровать признаки
5. Отобрать признаки моделью
6. Повторять до сходимости



# Вопросы?



Ставим “+”,  
если вопросы есть



Ставим “-”,  
если вопросов нет

# Рефлексия

# Рефлексия



С какими впечатлениями уходите с вебинара?



Как будете применять на практике то,  
что узнали на вебинаре?

**Заполните, пожалуйста,  
опрос о занятии  
по ссылке в чате**

Спасибо за внимание!

# Приходите на следующие вебинары

H2O, ТРОТ, Automatic Model Selection and Pipeline Building

Игорь Стурейко



Teamlead, главный инженер проекта – НИИгазэкономика

Опыт:

Более 15 лет занимался прикладной математикой и математическим моделированием  
(Data Scientist) (Python, C++) в НИИ ПАО Газпром

Анализ временных рядов, эволюционное развитие сложных систем

+7 (916) 156-07-82 (whatsapp)

@stureiko (TG)