

Занятие 1.

Введение в машинное обучение

Преподаватели

Рязанов Василий Владимирович

Кандидат Физ-мат наук

Kaggle Master

Опыт в DS 6+ лет



Емчинов Александр Владимирович

Аспирант, МФТИ

Старший специалист по анализу данных



План

1. Что подразумевают под машинным обучением и анализом данных?
2. Что нужно для того, чтобы начать?
3. Виды задач, метрики
4. Семейства алгоритмов
5. Практика

Что нужно для того, чтобы начать

1. python3 (scala)
2. Базовые операции: jupyter, numpy, pandas, sklearn
3. Визуализации: matplotlib, seaborn
4. Глубокое обучение: tensorflow, keras, pytorch
5. Специфические пакеты:
 - a. ближайшие соседи(annoy, faiss...)
 - b. анализ текстов (pymystem, spacy, ...)
 - c. анализ аудио (librosa)
 - d. ...и др. в зависимости от задачи
6. Железо

python3

python

Поисковый запрос

c++

Поисковый запрос

java

Поисковый запрос

+ Добавить сравнение

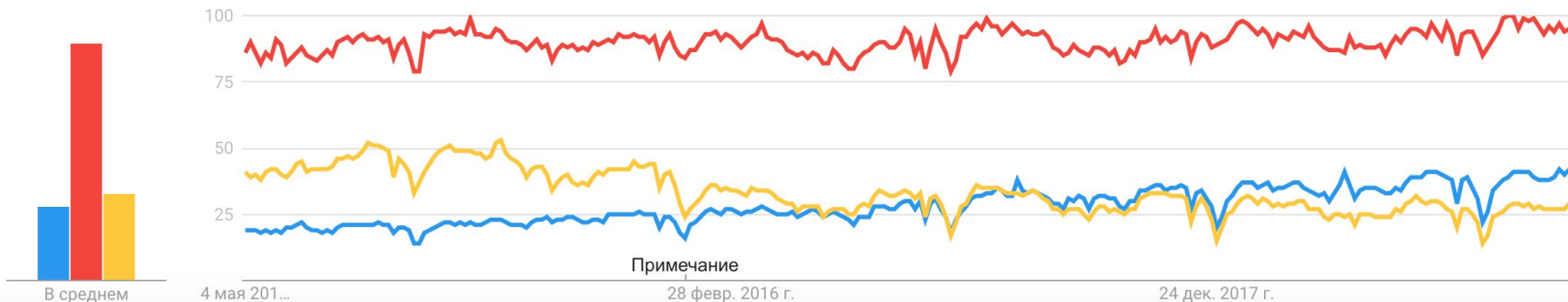
Соединенные Штаты ▼

Последние 5 лет ▼

Все категории ▼












Веб-поиск ▼

Динамика популярности ?



python2 vs python3

Используйте python3

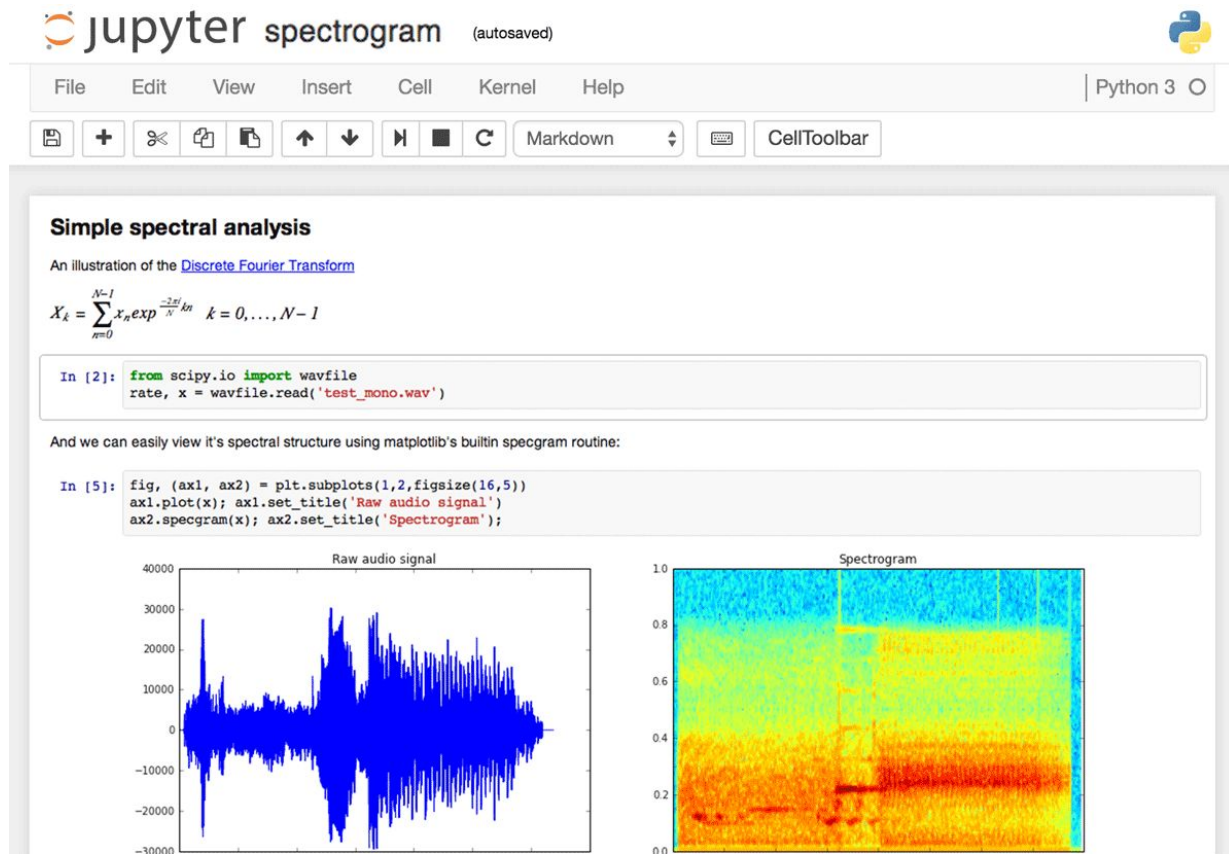
PYTHON 2		PYTHON 3
 Legacy		Future 
It is still entrenched in the software at certain companies		It will take over Python 2 by 2020
 Library		Library 
Many older libraries built for Python 2 are not forwards-compatible		Many of today's developers are creating libraries strictly for use with Python 3
0100 0001 ASCII		Unicode 0000 0000 0100 0001
Strings are stored as ASCII by default		Text strings are Unicode by default
 5/2=2		5/2=2.5 
It rounds your calculation down to the nearest whole number		The expression 5 / 2 will return the expected result
print "hello"		print ("hello")
Python 2 print statement		The print statement has been replaced with a print () function

python3 - обучение

- <https://pythontutor.ru/> / <https://snakify.org/ru/> (базовые упражнения)
- <https://leetcode.com/> (алгоритмические задачи)
- <http://judge.mipt.ru/> (контесты)
- Практика

Оболочки и среды разработки

- bash
- IDE (PyCharm, Sublime, ...)
- Jupyter
- Enterprise решения (например Databricks)



numpy - массивы, линейная алгебра

- <https://docs.scipy.org/doc/numpy/user/quickstart.html>
- Быстрое индексирование, операции на векторы
- Быстрые встроенные функции (np.sum, np.mean, np.exp, ...)
- Гибкий и простой

pandas - работа с таблицами

<https://pandas.pydata.org/pandas-docs/stable/10min.html>

Медленнее numpy, но нагляднее

In [33]: data

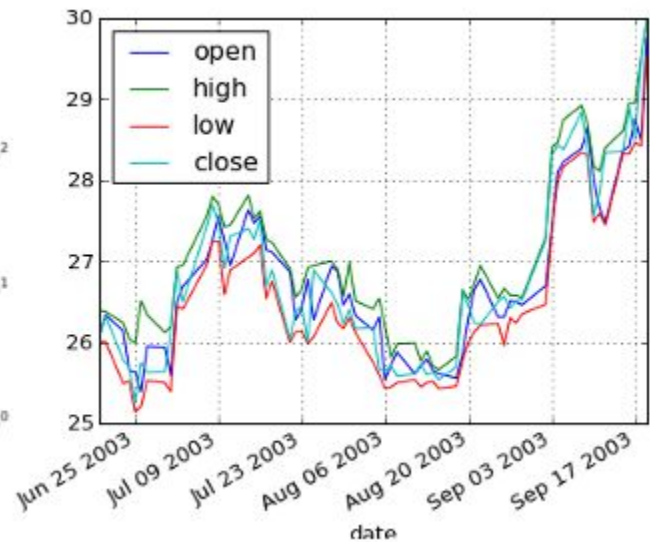
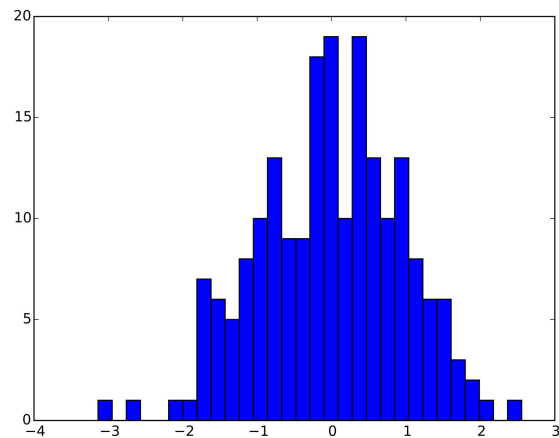
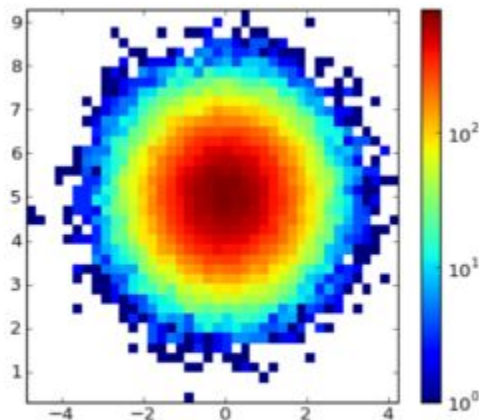
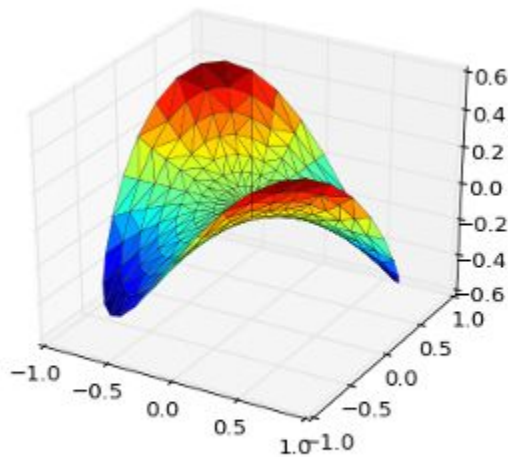
Out[33]:

	Area Abbreviation	Area Code	Area	Item Code	Item	Element Code	Element	Unit	latitude	longitude	...	Y2004	Y2005	Y2006	Y2007	Y2008	Y2009
0	AF	2	Afghanistan	2511	Wheat and products	5142	Food	1000 tonnes	33.94	67.71	...	3249.0	3486.0	3704.0	4164.0	4252.0	4538.0
1	AF	2	Afghanistan	2805	Rice (Milled Equivalent)	5142	Food	1000 tonnes	33.94	67.71	...	419.0	445.0	546.0	455.0	490.0	415.0
2	AF	2	Afghanistan	2513	Barley and products	5521	Feed	1000 tonnes	33.94	67.71	...	58.0	236.0	262.0	263.0	230.0	379.0
3	AF	2	Afghanistan	2513	Barley and products	5142	Food	1000 tonnes	33.94	67.71	...	185.0	43.0	44.0	48.0	62.0	55.0
4	AF	2	Afghanistan	2514	Maize and products	5521	Feed	1000 tonnes	33.94	67.71	...	120.0	208.0	233.0	249.0	247.0	195.0
5	AF	2	Afghanistan	2514	Maize and products	5142	Food	1000 tonnes	33.94	67.71	...	231.0	67.0	82.0	67.0	69.0	71.0

matplotlib - визуализация

Базовая библиотека для визуализаций

Непростой API



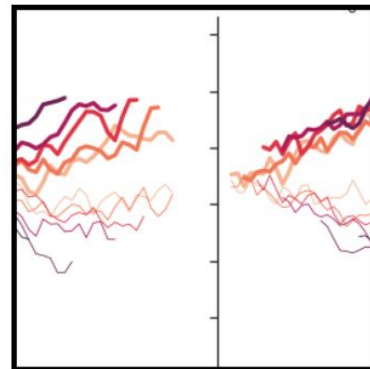
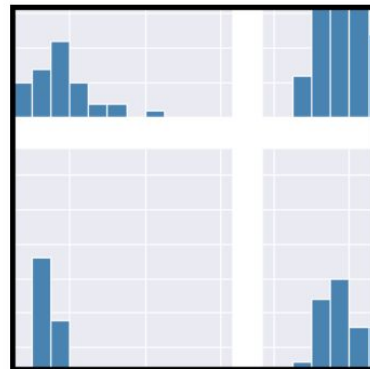
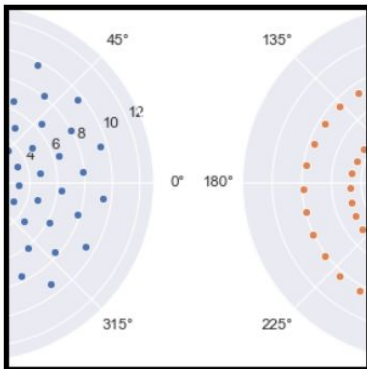
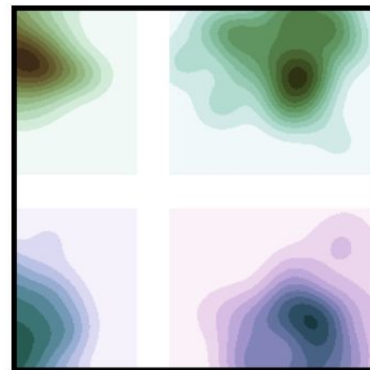
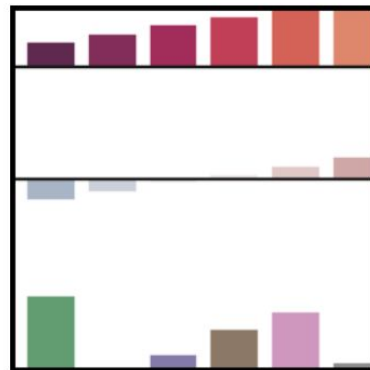
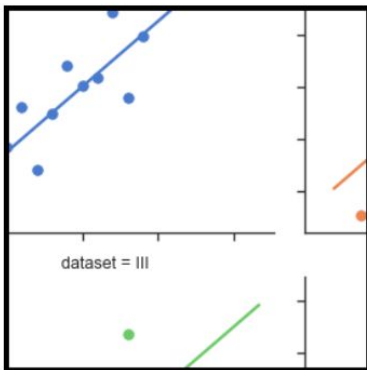
seaborn - визуализация с простым API

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

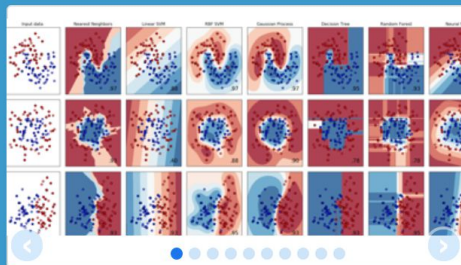
sns.set(style="white", context="talk")
rs = np.random.RandomState(8)

# Set up the matplotlib figure
f, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(7,

# Generate some sequential data
x = np.array(list("ABCDEFGHIJ"))
y1 = np.arange(1, 11)
sns.barplot(x=x, y=y1, palette="rocket", ax=ax1)
ax1.axhline(0, color="k", clip_on=False)
ax1.set_ylabel("Sequential")
```



sklearn - базовые операции из ML



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.

Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

Нейросети

<https://keras.io/>

<https://pytorch.org/>

<https://www.tensorflow.org/>

Deprecated :(

<https://github.com/Lasagne/Lasagne>

<https://github.com/Theano/Theano>



PYTORCH



Железо

Свое (core-i7, 32 gb ram, GPU)

Аренда (Amazon EC2, Microsoft Azure, Google Cloud)

Виды задач

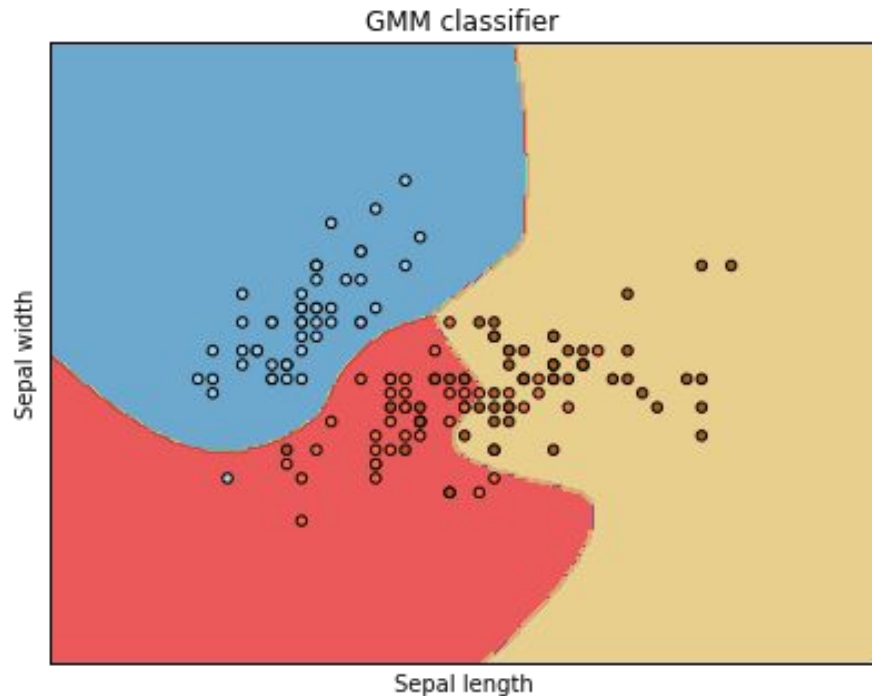
- Классификация (обучение с учителем/supervised)
 - Предсказание одной метки класса из конечного набора
- Регрессия (обучение с учителем/supervised)
 - Предсказание непрерывной величины
- Кластеризация (обучение без учителя/unsupervised)
 - Разбиение существующего набора объектов на группы без исходного набора меток

Классификация

Задача - предсказать метки классов (вероятности для каждого класса) по имеющимся признакам

Множество классов L конечно

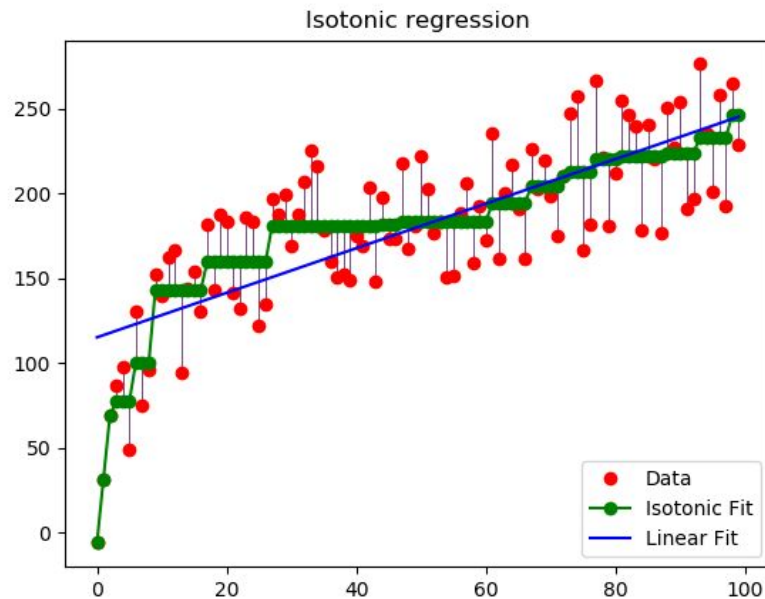
Среди классов нет порядка



Регрессия

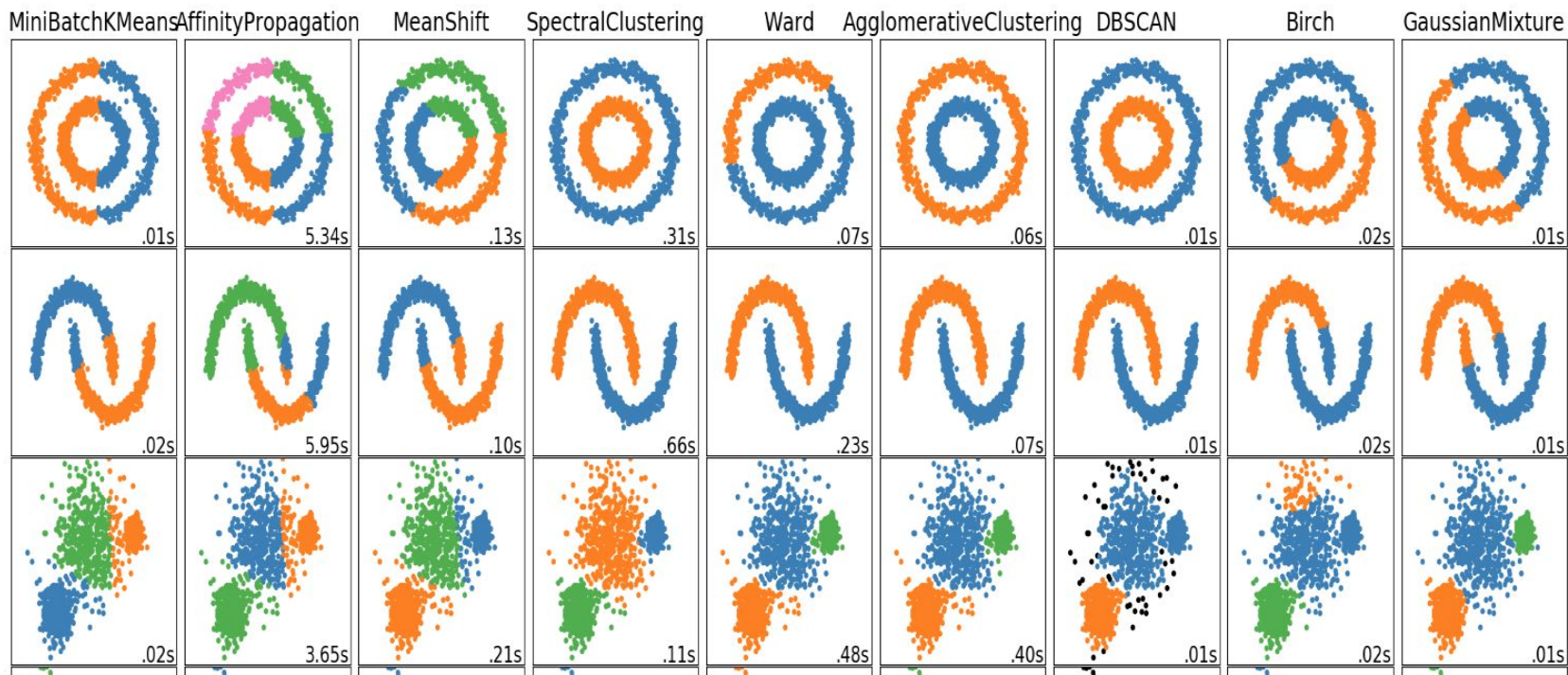
Предсказываем вещественные числа

Метрика, как правило, зависит от удаленности прогноза от истинного значения



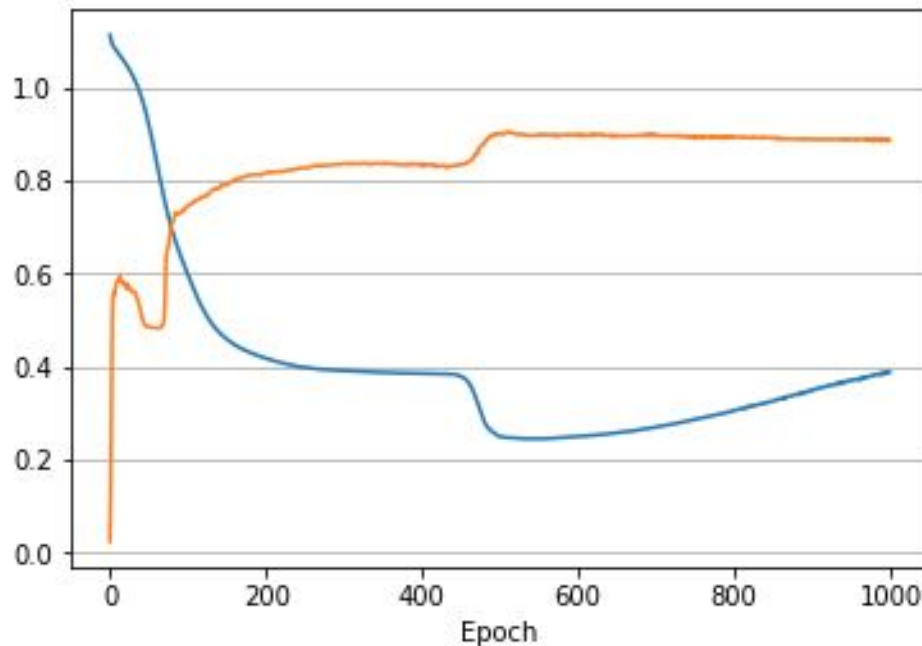
Кластеризация

Нужно сгруппировать объекты без меток



Loss vs Metric

- Target metric это то, что мы хотим оптимизировать
- Optimization loss это то, что модель оптимизирует



Классификация - метрики

- Accuracy
- Precision, Recall, F-мера, Confusion Matrix
- ROC-AUC
- LogLoss

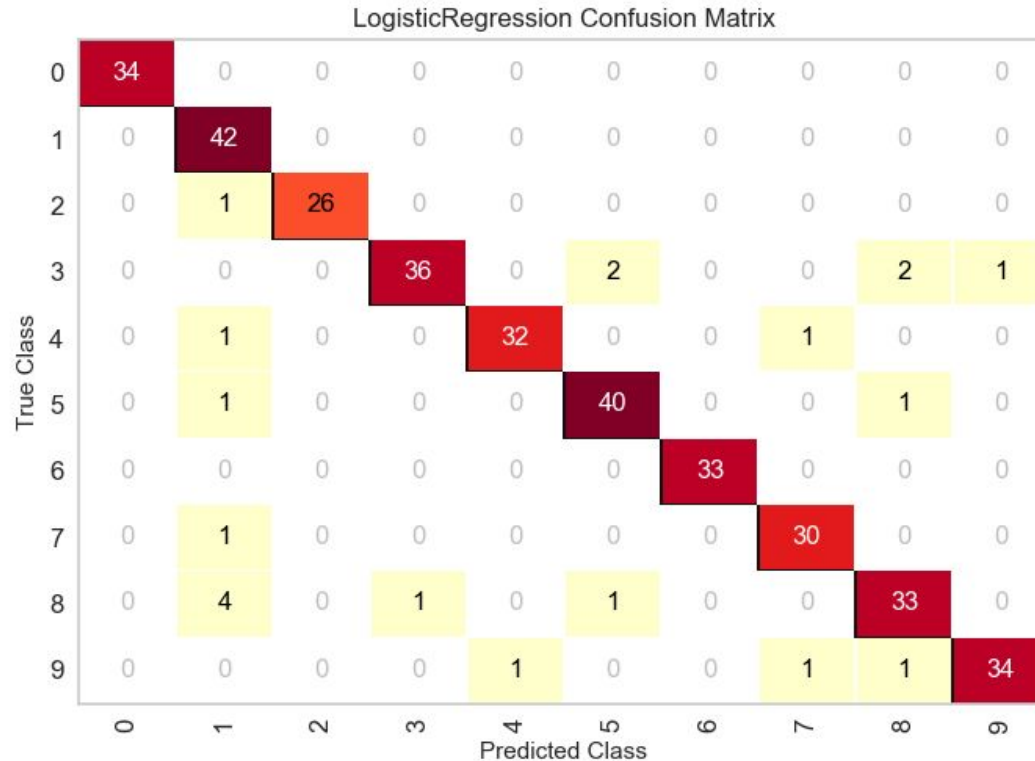
Accuracy - доля правильных ответов

Универсальная метрика

Accuracy = 0.99 - хорошо или нет?

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N [\hat{y}_i = y_i]$$

Confusion Matrix - Матрица ошибок



Confusion Matrix - бинарная классификация

```
In [14]: # Важно: первый аргумент - true values, второй - predicted values  
# получаем матрицу 2x2  
print(metrics.confusion_matrix(y_test, y_pred_class))  
  
[[118  12]  
 [ 47  15]]
```

N = 192	Предсказан: 0	Предсказана: 1
Истинный: 0	118	12
Истинный: 1	47	15

TP TN FP FN

N = 192		Предсказан: 0	Предсказана: 1
Истинный: 0		118	12
Истинный: 1		47	15

- True Positives (TP): мы правильно предсказали 1
 - 15
- True Negatives (TN): мы правильно предсказали 0
 - 118
- False Positives (FP): мы неправильно предсказали 1
 - 12
 - Ошибка 1 типа
- False Negatives (FN): мы неправильно предсказали 0
 - 47
 - Ошибка 2 типа

Метрики из Confusion Matrix

- Sensitivity: $\frac{TP}{(TN+FN)} = 0.242$
 - Когда исходное значение позитивны(1), как часто предсказания верны?
 - Как "sensitive" классификатор к обнаружению положительных классу?
 - Также известен как "True Positive Rate (TPR)" или "Recall"
 $\frac{TP}{(TP+FN)} = 0.556$
- Precision:
 - Когда мы предсказываем 1, как часто это действительно 1?
 $\frac{TP}{(TP+FP)} = 0.337$
- F1-score:
 - Это комбинация Precision и Recall
 - Подходит, если есть дисбаланс в выборке

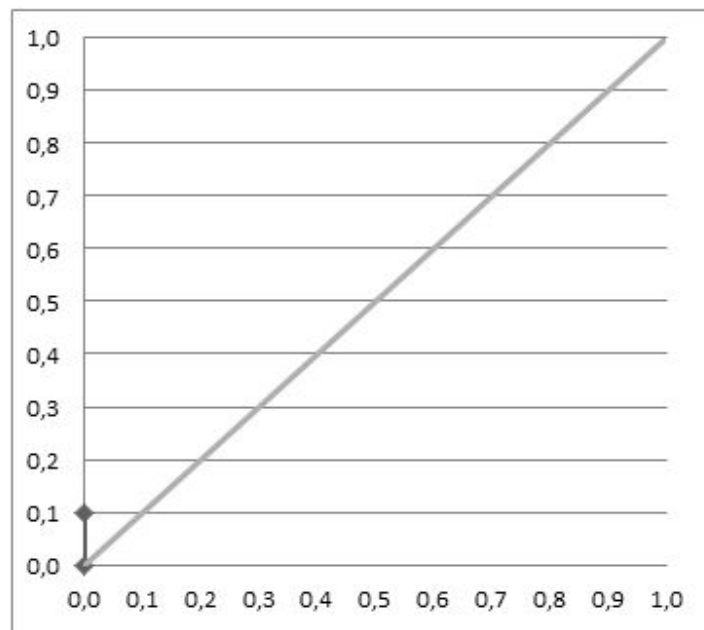
ROC-AUC

Учитывая набор данных и классификатор, который может ранжировать:

- Упорядочить тестовые примеры по шкале от самой высокой до самой низкой
- Начинаем в (0, 0) и с максимума вероятности
- Для каждого примера x в упорядоченном множестве
 - Если x положительный(=1), движемся $1/\text{pos}$ вверх
 - Если x отрицательный(=0), движемся $1/\text{neg}$ вправо

Где pos и neg доли положительных и отрицательных примеров.

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1



LogLoss

- Binary: $LogLoss = \frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$ $y_i \in \mathbb{R}, \quad \hat{y}_i \in \mathbb{R}$
- Multiclass: $LogLoss = -\frac{1}{N} \sum_{i=1}^N \sum_{l=1}^L y_{i,l} \log(\hat{y}_{i,l})$ $y_i \in \mathbb{R}^L, \quad \hat{y}_i \in \mathbb{R}^L$
- На практике: $LogLoss = -\frac{1}{N} \sum_{i=1}^N \sum_{l=1}^L y_{i,l} \log(\min(\max(\hat{y}_{i,l}, 10^{-15}), 1 - 10^{-15}))$

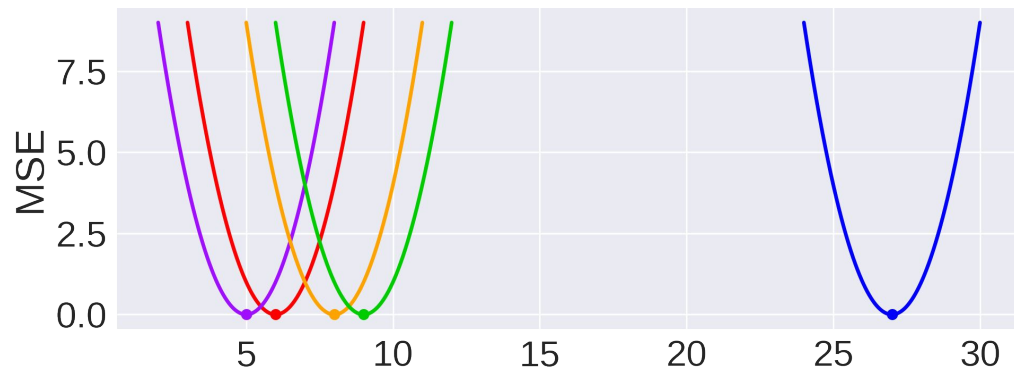
Регрессия - метрики

- MSE, RMSE, R-squared
- MAE
- (R)MSPE, MAPE
- (R)MSLE

MSE: Mean Squared Error

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

X	Y
...	5
...	9
...	8
...	6
...	27

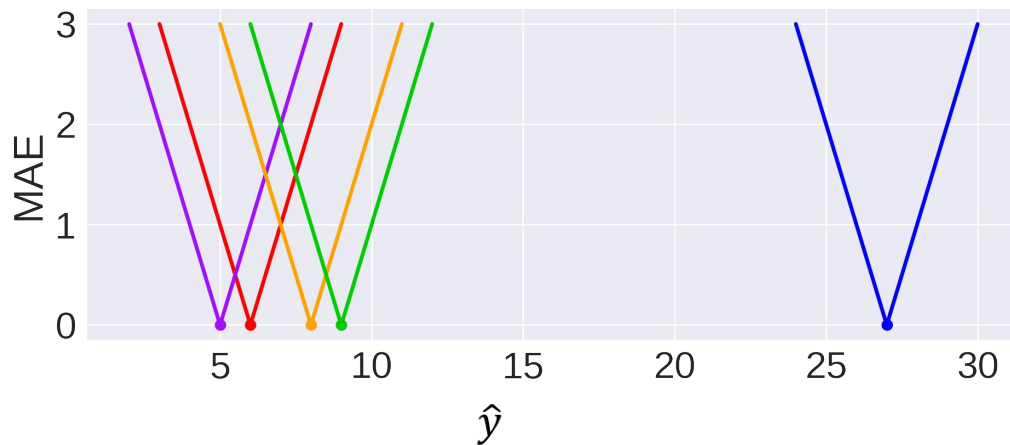


MAE: Mean Absolute Error

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

Data:

X	Y
-1	5
1	9
-2	8
3	6
3	27



Семейства алгоритмов

- Линейные модели
- Деревянные модели
- Модели на основе метрической близости
- Нейросетевые модели
- Вероятностные модели

Линейные модели

Классификация

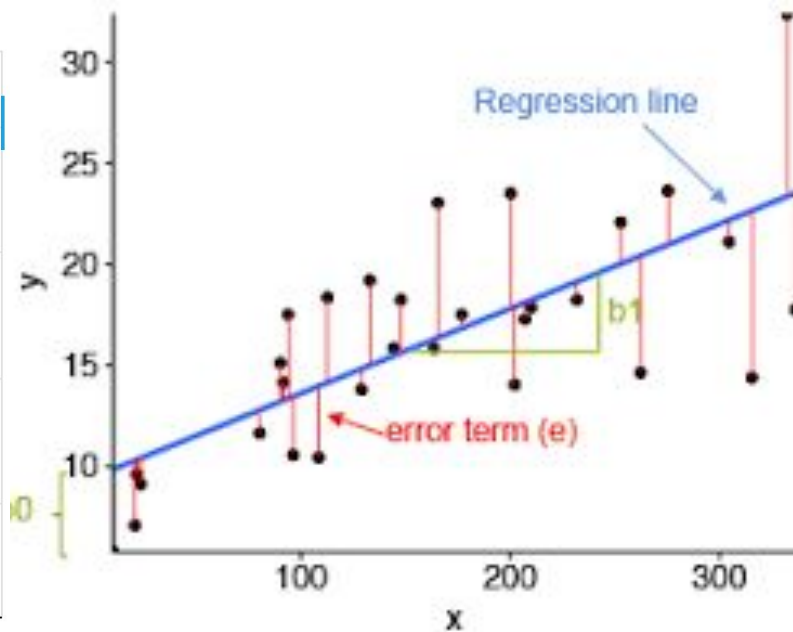
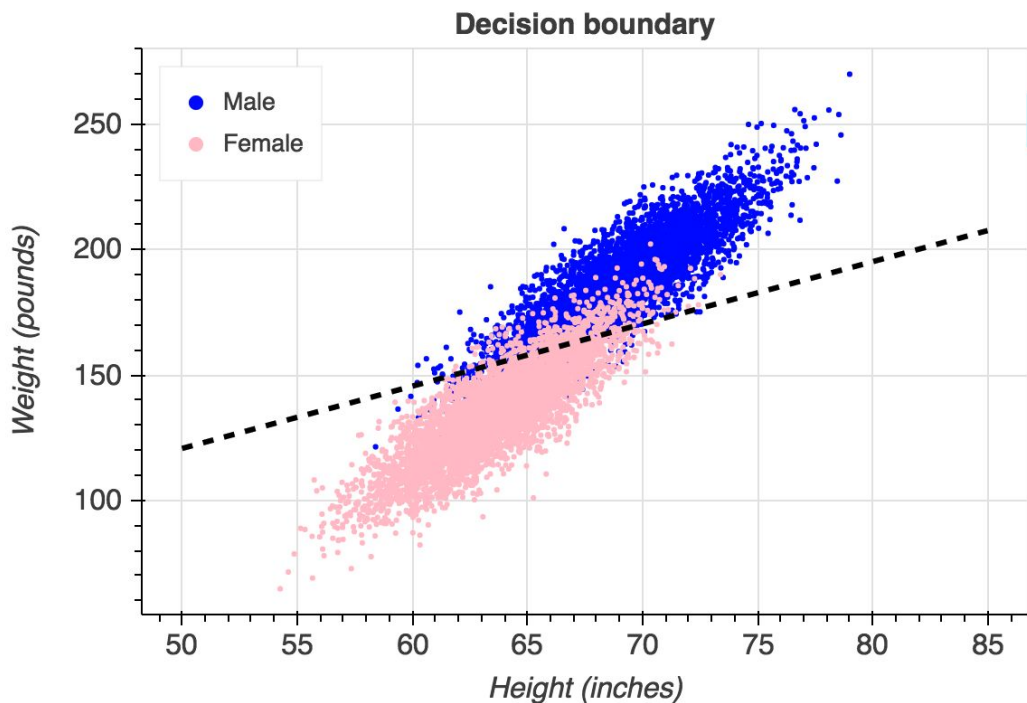
1. Логистическая регрессия
2. SVM
3. Linear Discriminant Analysis (LDA)
4. ...

Регрессия

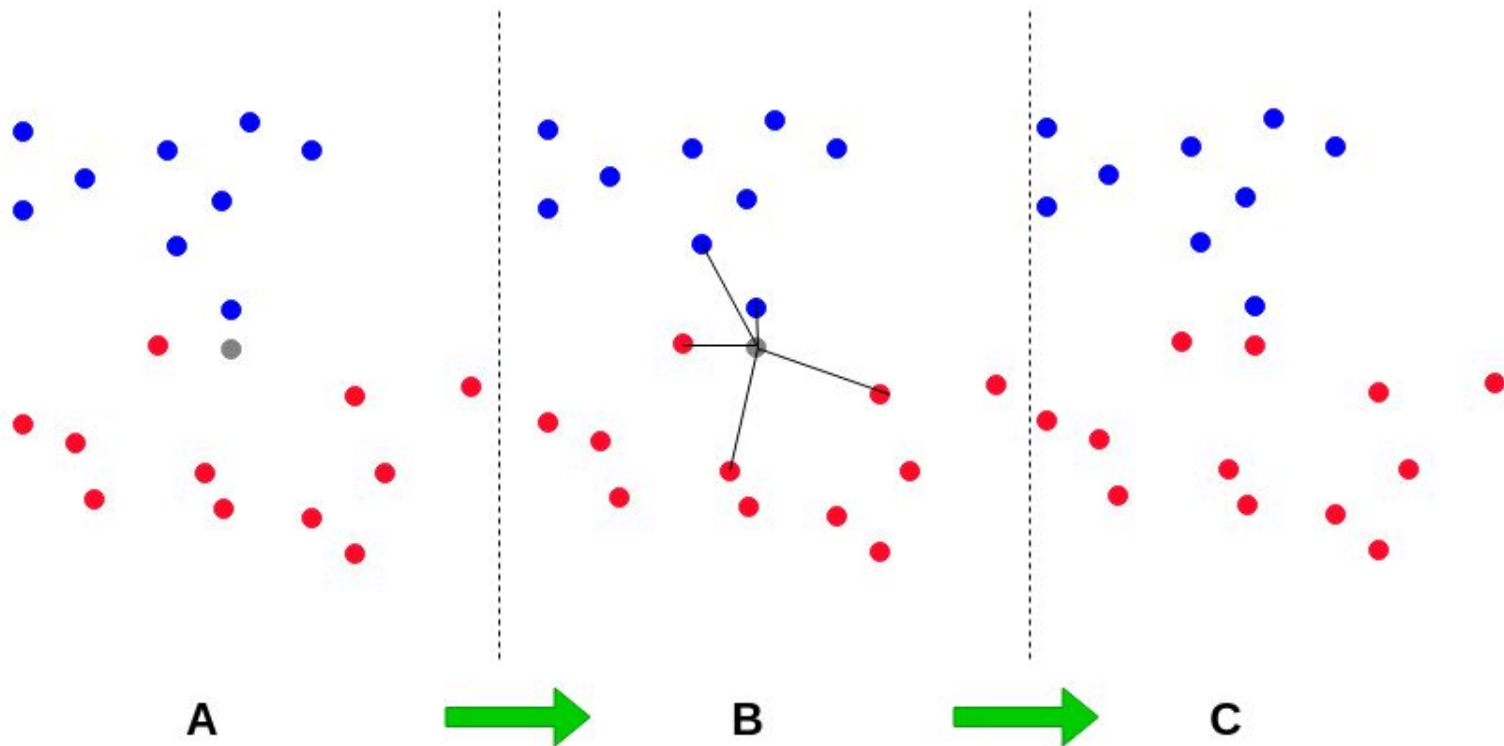
1. Линейная регрессия
2. ...

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
```

Логистическая регрессия и линейная регрессия



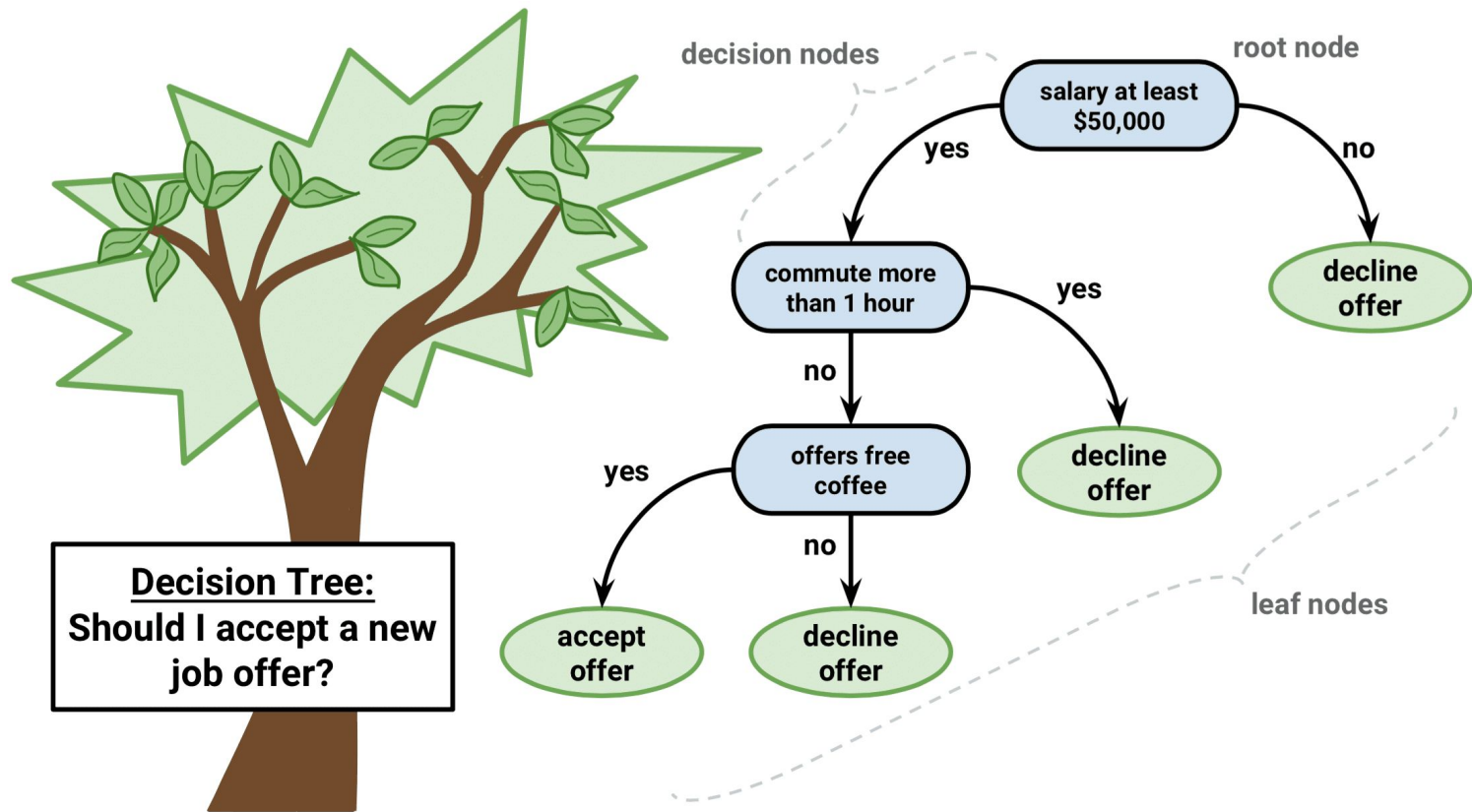
KNN - метод ближайших соседей



Деревянные модели

- Дерево решений
- Случайный лес
- Градиентный бустинг

Бинарное дерево решений

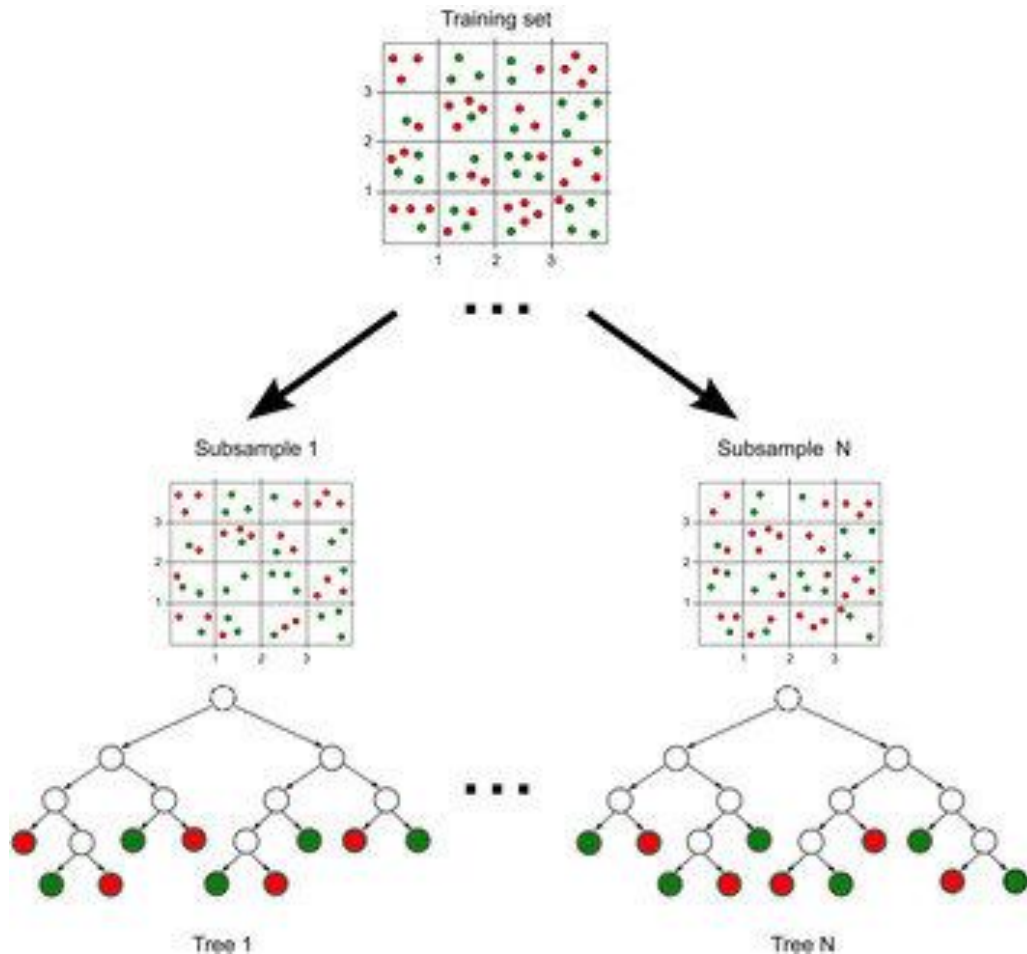


Случайный лес

Алгоритм

- 1) Выбирается случайная подвыборка объектов
- 2) Выбирается случайная подвыборка признаков
- 3) Строится дерево

ExtraTrees (EXTremely RAndomized Trees) - split дерева делается по случайному значению



XGBoost, LightGBM, CatBoost - timeline

Март 2014

Январь 2017

Апрель 2017

XGBoost

Исследовательский проект Tianqi Chen (стал популярным в 2016)



15,198



6,440

Microsoft выпускает стабильную версию **LightGBM**



8,062



2,164

Яндекс выкладывает в open source **CatBoost**



3,803



560

Как работает бустинг

Упрощенная формула бустинга

$$f(x) = \sum_{m=0}^M f_m(x) = f_0(x) + \sum_{m=1}^M \theta_m \phi_m(x),$$

Решается задача
(L - выбран нами)

$$\{\theta_m, \phi_m\} = \arg \min_{\{\theta_m, \phi_m\}} \sum_{i=1}^n L(y_i, f^{(m-1)}(x_i) + \theta_m \phi_m(x_i))$$

Новое дерево ищется как решение
задачи **регрессии** (L = MSE)

$$\phi_m = \arg \min_{\phi} \sum_{i=1}^n [(-g_m(x_i)) - \phi(x_i)]^2.$$

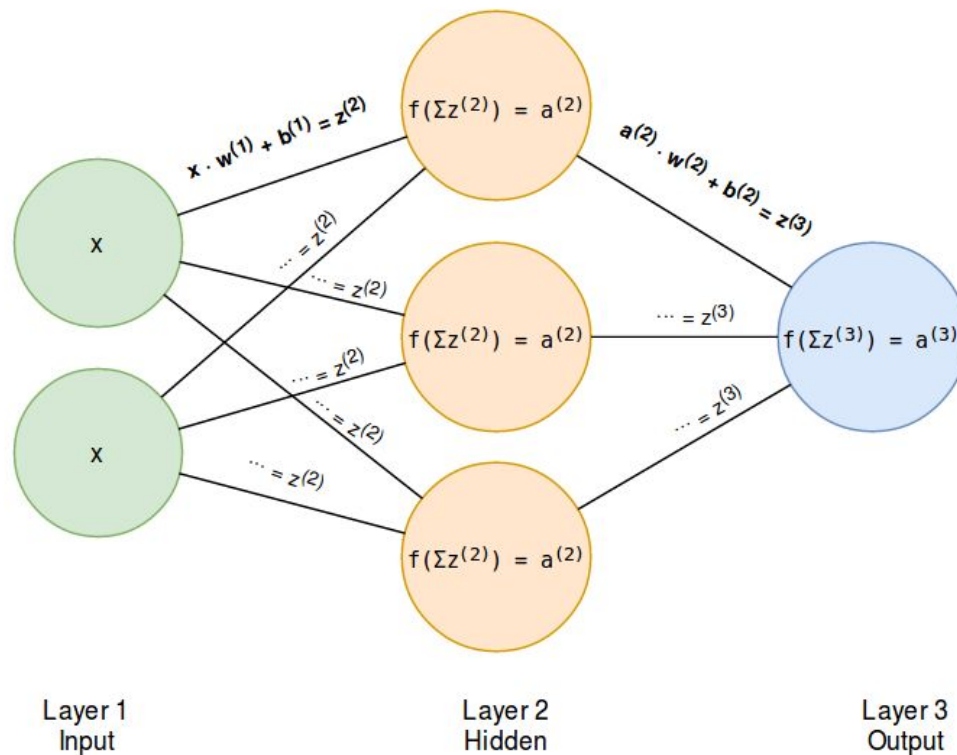
Финальное новое дерево

$$f_m(x) = \eta \rho_m \phi_m(x)$$

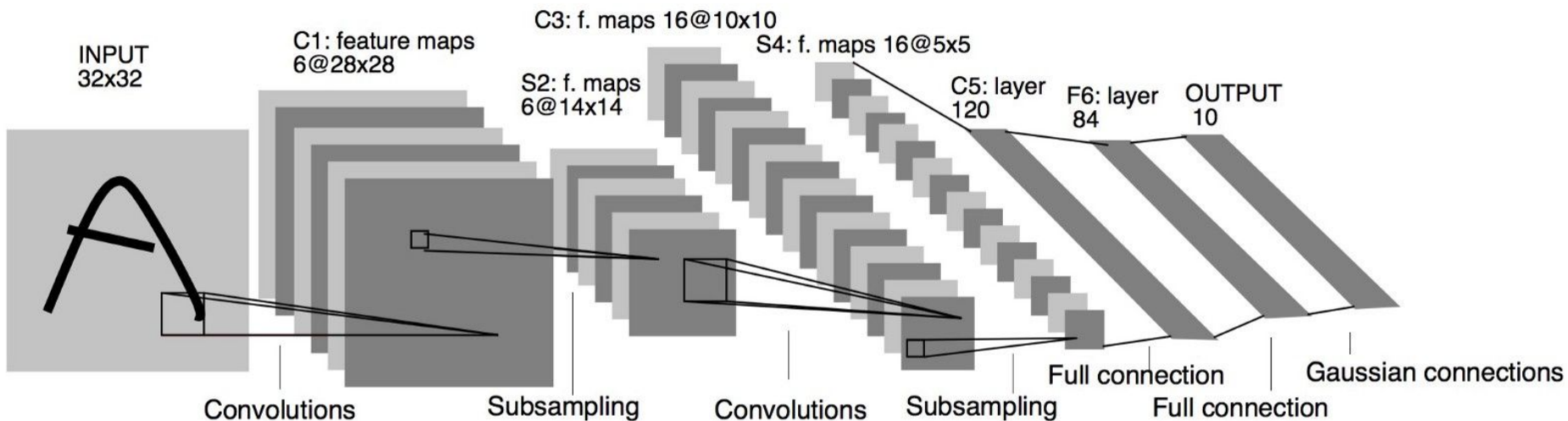
градиенты

learning rate

Нейронные сети



Архитектуры нейронных сетей



Архитектуры нейронных сетей

