142,92

Рейтинг

NIX Solutions

Компания

NIX_Solutions 24 июля в 11:06

Памятки по искусственному интеллекту, машинному обучению, глубокому обучению и большим данным

https://becominghuman.ai/cheat-sheets-for-ai-neural-networks-machine-learning-deep-learning-big-data-678c51b4b463

Машинное обучение, Искусственный интеллект, Big Data, Блог компании NIX Solutions

Перевод

ИНФОРМАЦИЯ

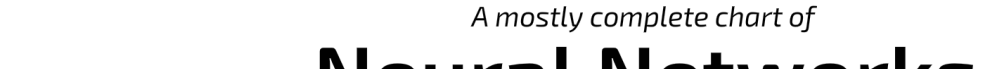
Дата основания	1994 год
Локация	Харьков Украина
Сайт	nixsolutions.com
Численность	1001–5000 человек
Дата регистрации	25 февраля 2015

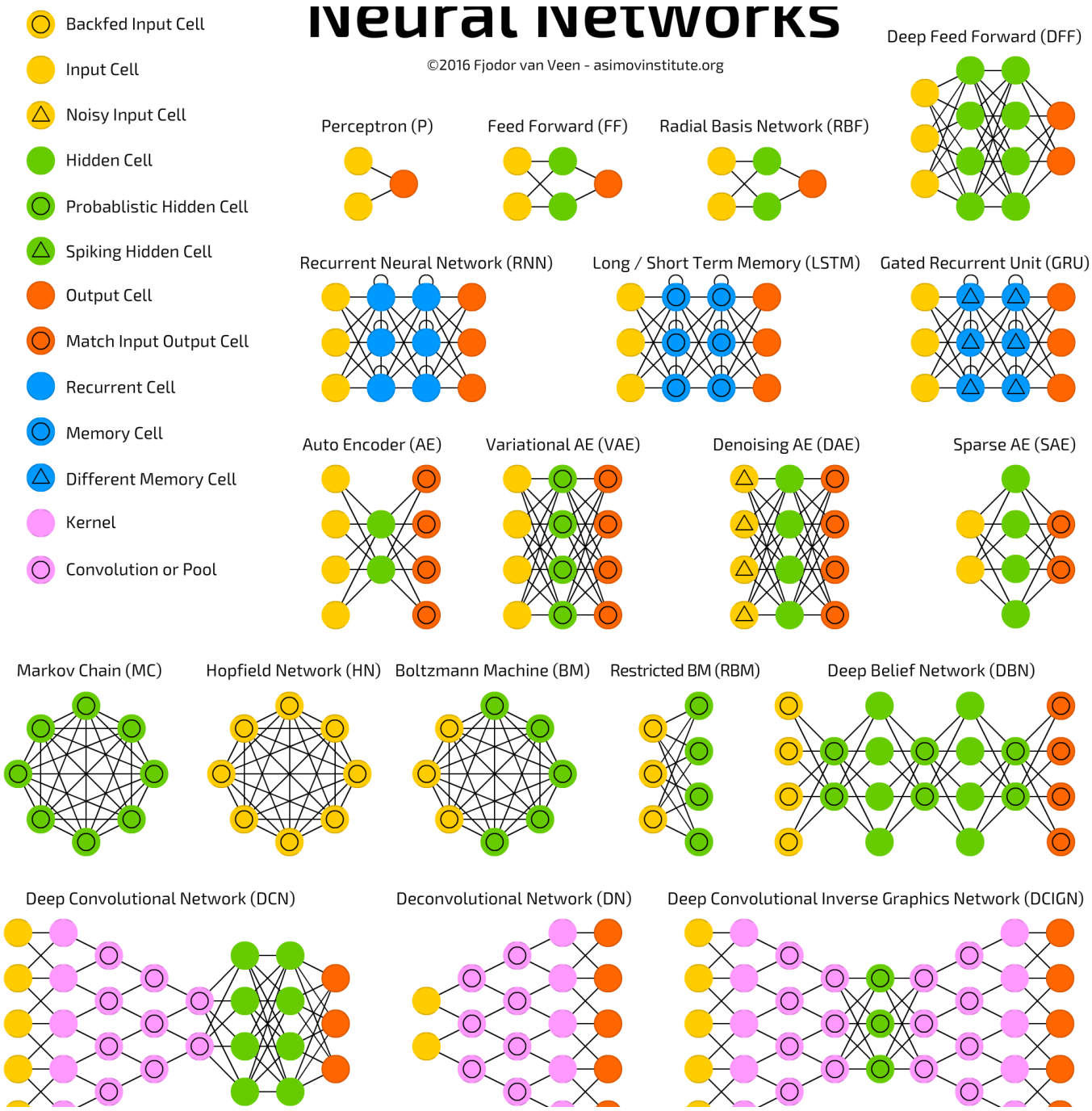
FACEBOOK

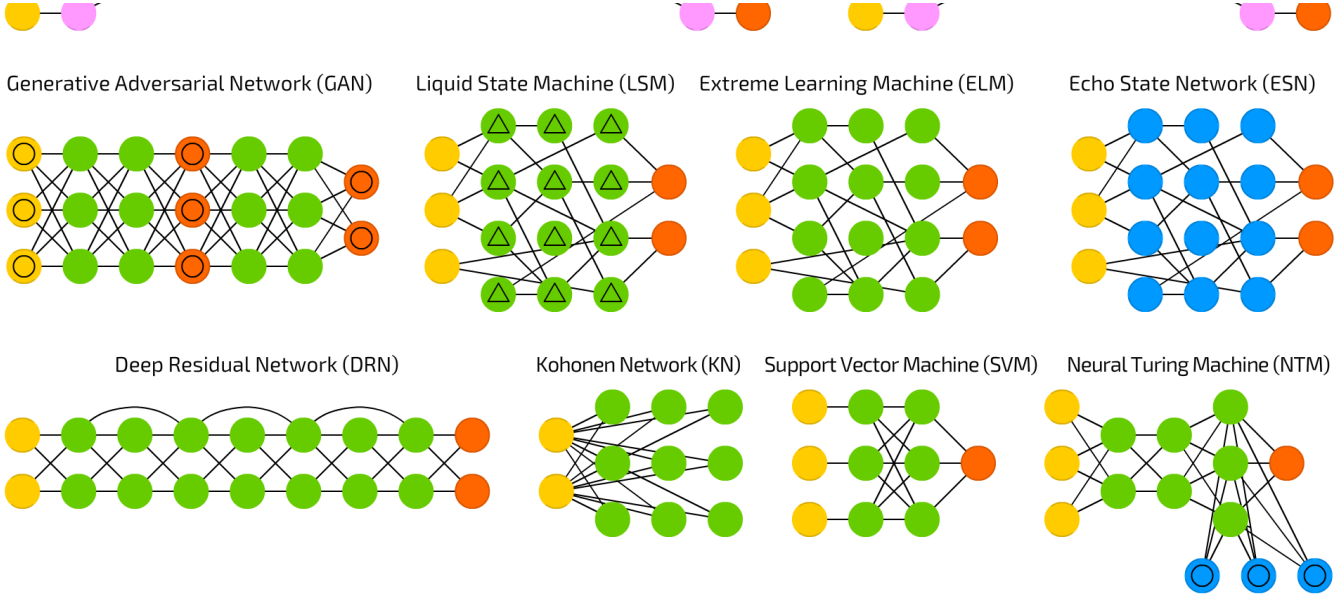
В течение нескольких месяцев мы собирали памятки по искусственному интеллекту, которыми периодически делились с друзьями и коллегами. В последнее время сложилась целая коллекция, и мы добавили к памяткам описания и/или цитаты, чтобы было интереснее читать. А в конце вас ждёт подборка по сложности «О большое» (Big-O). Наслаждайтесь.

UPD. Многие картинки будут читабельнее, если открыть их в отдельных вкладках или сохранить на диск.

Нейронные сети





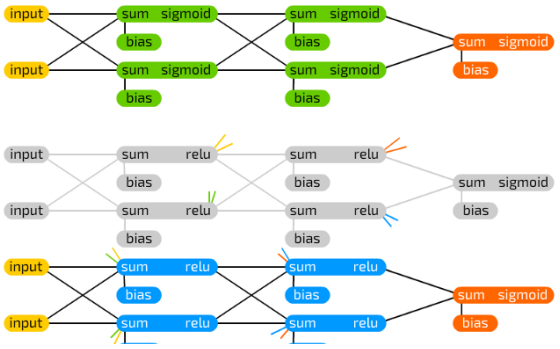


Памятка по нейронным сетям

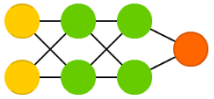
Графы нейронных сетей

An informative chart to build

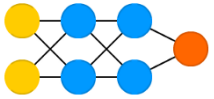
Neural Network Graphs



Deep Feed Forward Example



Deep Recurrent Example
(previous iteration)



Deep Recurrent Example

Как ИИ учится генерировать изображения кошек
8,3k 0

27 отличных open source-инструментов для веб-разработки
38,8k 10

Как в React избавиться от сложности в управлении состоянием — отчёт по итогам поездки на React Amsterdam
5,8k 3

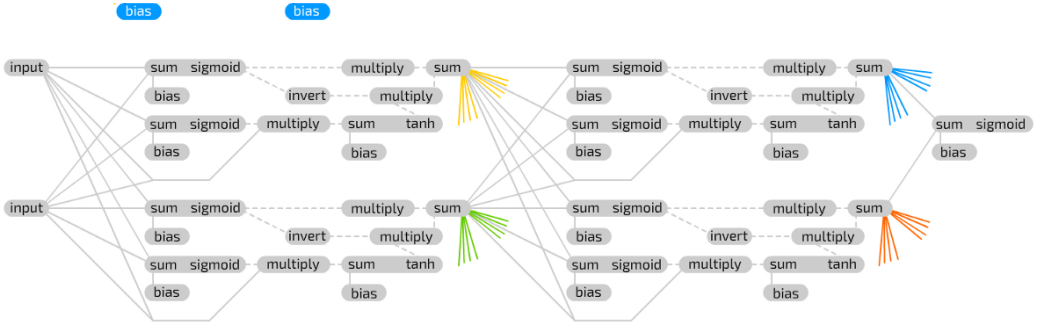
Как машины анализируют большие данные: введение в алгоритмы кластеризации
7,3k 7

Я сделал PWA и выложил в трёх магазинах приложений. И вот что я выяснил
17,4k 58

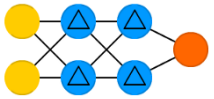
Анонс митапа RubyRoars #1 в Харькове
667 0

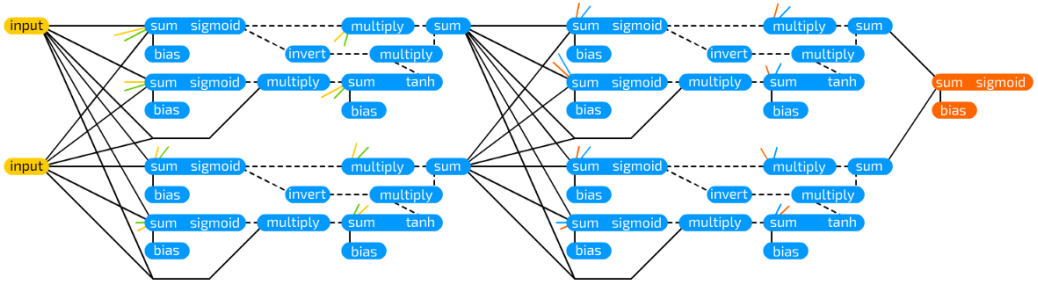
Анонс митапа Sync.NET #6 в Харькове
1,3k 2

Анонс митапа ThinkPHP #16 в Харькове
1,7k 0

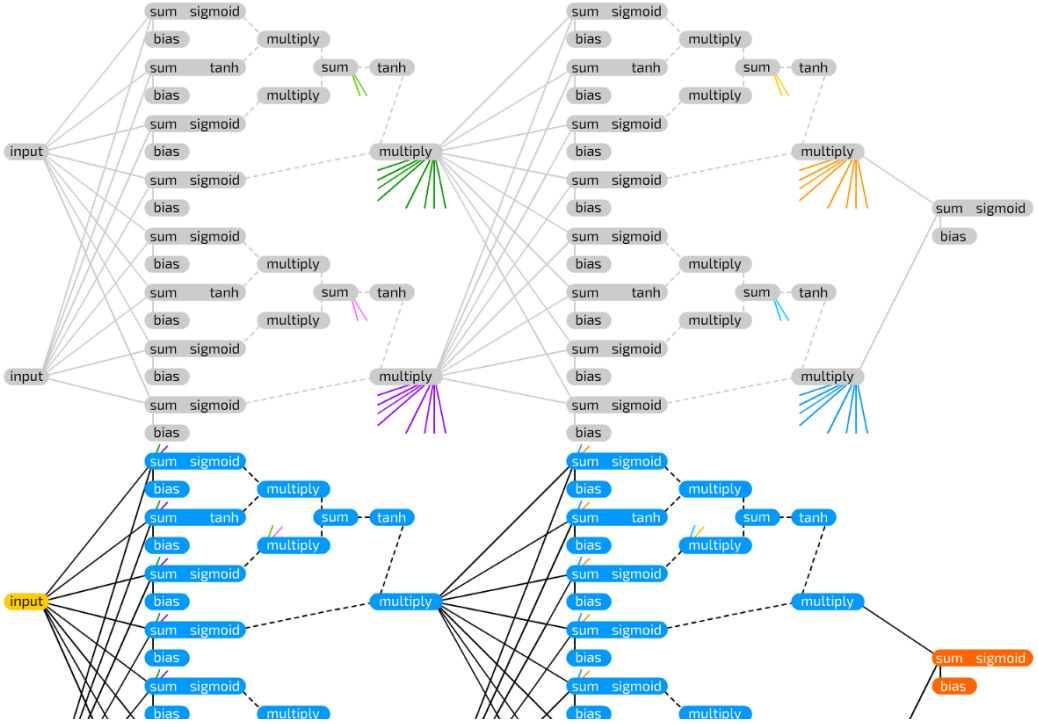


Deep GRU Example
(previous iteration)

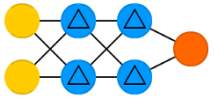




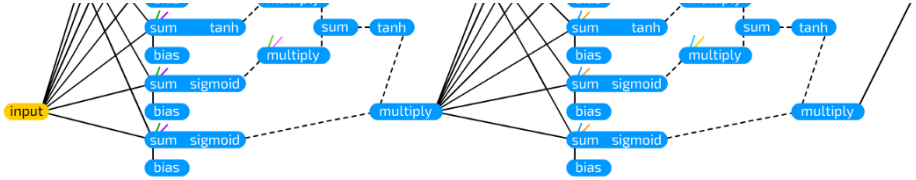
Deep GRU Example



Deep LSTM Example
(previous iteration)



Deep LSTM Example



Памятка по графикам нейронных сетей

Linear Vector Spaces:

Definition: A linear vector space, X, is a set of elements (vectors) defined over a scalar field, F, that satisfies the following conditions:

1) if $x \in X$ and $y \in X$ then $x+y \in X$. **2)** $x+y=y+x$ **3)** $(x+y)+z=x+(y+z)$
4) There is a unique vector $0 \in X$, such that $x+0=x$ for all $x \in X$.
5) For each vector $x \in X$ there is a unique vector in X, to be called $(-x)$, such that $x+(-x)=0$. **6)** multiplication, for all scalars $a \in F$, and all vectors $x \in X$,
7) For any $x \in X$, $1x=x$ (for scalar 1).
8) For any two scalars $a \in F$ and $b \in F$ and any $x \in X$, $a(bx)=(ab)x$.
9) $(a+b)x=ax+bx$. **10)** $a(x+y)=ax+ay$.

Linear Independence: Consider n vectors $\{x_1, x_2, \ldots, x_n\}$. If there exists n scalars $a_1, a_2, \ldots, a_n$, at least one of which is nonzero, such that $a_1x_1+a_2x_2+\ldots+a_nx_n=0$, then the $\{x_i\}$ are linearly dependent.

Spanning a Space:
Let X be a linear vector space and let $\{u_1, u_2, \ldots, u_n\}$ be a subset of vectors in X. This subset spans X if and only if for every vector $x \in X$ there exist scalars $x_1, x_2, \ldots, x_n$ such that $x = x_1u_1+x_2u_2+\ldots+x_nu_n$.

Inner Product: (x,y) for any scalar function of x and y.
1.$(x,y)=(y,x)$ 2. $(x,ay_1+by_2)=a(x,y_1)+b(x,y_2)$
3. $(x,x) \geq 0$, where equality holds iff x is the zero vector.

Norm: Ascalar function $\|x\|$ is called a norm if it satisfies:
1. $\|x\| \geq 0$ 2. $\|x\| = 0$ if and only if $x = 0$.
3. $\|ax\| = |a|\|x\|$ 4. $\|x+y\| \leq \|x\| + \|y\|$

Angle: The angle θ bet. 2 vectors x and y is defined by $\cos\theta = \frac{(x,y)}{\|x\|\,\|y\|}$

Orthogonality: 2 vectors $x, y \in X$ are said to be orthogonal if $(x,y) = 0$.

Gram Schmidt Orthogonalization:
Assume that we have n independent vectors $y_1, y_2, \ldots, y_n$. From these vectors we will obtain n orthogonal vectors $v_1, v_2, \ldots, v_n$.

$$v_1 = y_1, \quad v_k = y_k - \sum_{i=1}^{k-1} \frac{(v_i, y_k)}{(v_i, v_i)} v_i,$$

$$where \ \frac{(v_i, y_k)}{(v_i, v_i)} v_i \ is \ the \ projection \ of \ y_k \ on \ v_i$$

Vector Expansions:

$$x = \sum_{i=1}^{n} x_i v_i = x_1v_1 + x_2v_2 + \cdots + x_nv_n,$$

Perceptron Architecture:

$\mathbf{a} = hardlim(\mathbf{Wp}+\mathbf{b})$, $\mathbf{W} = [{}_1\mathbf{w}^T \ {}_2\mathbf{w}^T \ \ldots \ {}_S\mathbf{w}^T]^T$,

$$a_i = hardlim(n_i) = hardlim({}_i\mathbf{w}^T\,\mathbf{p} + b_i)$$

Decision Boundary: ${}_i\mathbf{w}^T\,\mathbf{p} + b_i = 0$

The decision boundary is always orthogonal to the weight vector.
Single-layer perceptrons can only classify linearly separable vectors.

Perceptron Learning Rule

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{e}\mathbf{p}^T, \ \mathbf{b}^{new} = \mathbf{b}^{old} + \mathbf{e},$$
$$where \quad \mathbf{e} = \mathbf{t} - \mathbf{a}$$

Hebb's Postulate: "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."

Linear Associator: $\mathbf{a} = purelin(\mathbf{Wp})$

The Hebb Rule: *Supervised Form:* $w_{ij}^{new} = w_{ij}^{old} + t_{qi}P_{qi}$

$$\mathbf{W} = \boldsymbol{t}_1\boldsymbol{P}_1^T + \boldsymbol{t}_2\boldsymbol{P}_2^T + \cdots + \boldsymbol{t}_Q\boldsymbol{P}_Q^T$$

$$\mathbf{W} = \begin{bmatrix}\mathbf{t}_1 \ \mathbf{t}_2 \ \ldots \mathbf{t}_Q\end{bmatrix}\begin{bmatrix}\mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_Q^T\end{bmatrix} = \mathbf{T}\mathbf{P}^T$$

Pseudoinverse Rule: $\mathbf{W} = \mathbf{T}\mathbf{P}^+$

When the number, R, of rows of $\mathbf{P}$ is greater than the num ber of columns,Q, of $\mathbf{P}$ and the columns of $\mathbf{P}$ are independent, then the pseudoinverse can be computed by $\mathbf{P}^+ = (\mathbf{P}^T\mathrm{P})^{-1}\mathbf{P}^T$

Variations of Hebbian Learning:

Filtered Learning (Ch.14): $\mathbf{W}^{new} = (1-\gamma)\mathbf{W}^{old} + \alpha\mathbf{t}_q\mathbf{p}_q^T$

Delta Rule (Ch.10): $\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha(\mathbf{t}_q - \mathbf{a}_q)\mathbf{p}_q^T$

Unsupervised Hebb (Ch.13): $\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha\mathbf{a}_q\mathbf{p}_q^T$

Taylor: $F(\mathbf{x}) = F(\mathbf{x}^*) + \nabla F(\mathbf{x})^T|_{\mathbf{x}=\mathbf{x}^*}(\mathbf{x}-\mathbf{x}^*) + \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)\nabla^2 F(\mathbf{x})^T|_{\mathbf{x}=\mathbf{x}^*}(\mathbf{x}-\mathbf{x}^*) + \cdots$

$$for\ orthogonal\ vectors, x_j = \frac{(v_j, x)}{(v_j, v_j)}$$

Reciprocal Basis Vectors:

$$(r_i, v_j) = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}, \quad x_j = (r_j, x)$$

To compute the reciprocal basis vectors: set $\mathbf{B} = [\mathrm{v}_1\ \mathrm{v}_2\ \dots\ \mathrm{v}_n]$,

$\mathbf{R} = [\mathrm{r}_1\ \mathrm{r}_2 \dots\ \mathrm{r}_n]$, $\mathbf{R}^T = \mathbf{B}^{-1}$ In matrix form: $\mathbf{x}^v = \mathbf{B}^{-1}\,\mathbf{x}^s$

Transformations:

A *transformation* consists of three parts:
domain: $X = \{ x_i \}$, range: $Y = \{ y_i \}$, and a rule relating each $x_i \in X$ to an element $y_i \in Y$.

Linear Transformations: transformation A is *linear* if:

1. for all $x_1, x_2 \in X$, $A(x_1 + x_2) = A(x_1) + A(x_2)$
2. for all $x \in X$, $a \in R$, $A(ax) = aA(x)$

Matrix Representations:

Let $\{v_1, v_2, \dots, v_n\}$ be a basis for vector space X, and let $\{u_1, u_2, \dots, u_n\}$ be a basis for vector space Y. Let A be a linear transformation with domain X and range Y: $A(x) = y$

The coefficients of the matrix representation are obtained from

$$A(v_j) = \sum_{i=1}^{m} a_{ij} u_i$$

Change of Basis: $\mathbf{B}_t = [\mathbf{t}_1\ \mathbf{t}_2\ \dots \mathbf{t}_n]$, $\mathbf{B}_w = [\mathbf{w}_1\ \mathbf{w}_2\ \dots \mathbf{w}_n]$

$$\mathbf{A}' = [\mathbf{B}_w^{-1}\mathbf{A}\mathbf{B}_t]$$

Eigenvalues & Eigenvectors: $\mathbf{Az} = \lambda \mathbf{z}$, $|[\mathbf{A} - \lambda \mathbf{I}]| = 0$

Diagonalization: $\mathbf{B} = [\mathbf{z}_1\ \mathbf{z}_2\ \dots \mathbf{z}_n]$,

where $\{\mathbf{z}, \mathbf{z}_2, \dots, \mathbf{z}_n\}$ are the eigenvectors of a square matrix A,

$$[\mathbf{B}^{-1}\mathbf{A}\mathbf{B}] = \mathrm{diag}([\lambda_1\ \lambda_2 \dots \lambda_n])$$

Grad $\nabla F(\mathbf{x}) = \left[\frac{\partial}{\partial \mathrm{x}_1} F(\mathbf{x}) \quad \frac{\partial}{\partial \mathrm{x}_2} F(\mathbf{x}) \quad \dots \quad \frac{\partial}{\partial \mathrm{x}_n} F(\mathbf{x})\right]^T$

Hessian: $\nabla^2 \mathrm{F}(\mathrm{x}) =$

$$\begin{bmatrix} \frac{\partial}{\partial \mathrm{x}_1{}^2} F(\mathbf{x}) & \frac{\partial}{\partial \mathrm{x}_1 \partial \mathrm{x}_2} F(\mathbf{x}) \dots & \frac{\partial}{\partial \mathrm{x}_1 \partial \mathrm{x}_n} F(\mathbf{x}) \\ \frac{\partial}{\partial \mathrm{x}_2 \partial \mathrm{x}_1} F(\mathbf{x}) & \frac{\partial}{\partial \mathrm{x}_2{}^2} F(\mathbf{x}) \dots & \frac{\partial}{\partial \mathrm{x}_2 \partial \mathrm{x}_n} F(\mathbf{x}) \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial \mathrm{x}_n \partial \mathrm{x}_1} F(\mathbf{x}) & \frac{\partial}{\partial \mathrm{x}_n \partial \mathrm{x}_2} F(\mathbf{x}) \dots & \frac{\partial}{\partial \mathrm{x}_n{}^2} F(\mathbf{x}) \end{bmatrix}$$

Directional Derivatives:

1st Dir.Der.: $\frac{\mathbf{p}^T \nabla F(\mathbf{x})}{\|\mathbf{p}\|}$, **2nd Dir.Der.:** $\frac{\mathbf{p}^T \nabla^2 F(\mathbf{x}) \mathbf{p}}{\|\mathbf{p}\|^2}$

Minima:

Strong Minimum: if a scalar $\delta > 0$ exists, such that $F(x) < F(x + \Delta x)$ for all Δx such that $\delta > \|\Delta x\| > 0$.

Global Minimum: if $F(x) < F(x + \Delta x)$ for all $\Delta x \neq 0$

Weak Minimum: if it is not a strong minimum, and a scalar $\delta > 0$ exists, such that $F(x) \leq F(x + \Delta x)$ for all Δx such that $\delta > \|\Delta x\| > 0$.

Necessary Conditions for Optimality:

1st-Order Condition: $\nabla \mathrm{F}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^*} = 0$ (Stationary Points)

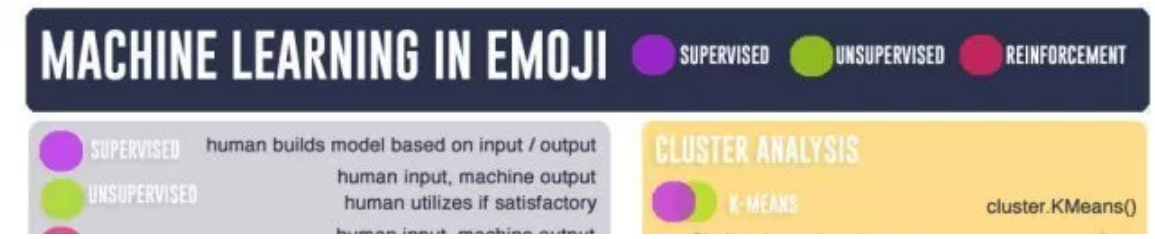
2nd-Order Condition: $\nabla^2 \mathrm{F}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^*} \geq 0$ (Positive Semi-definite Hessian Matrix).

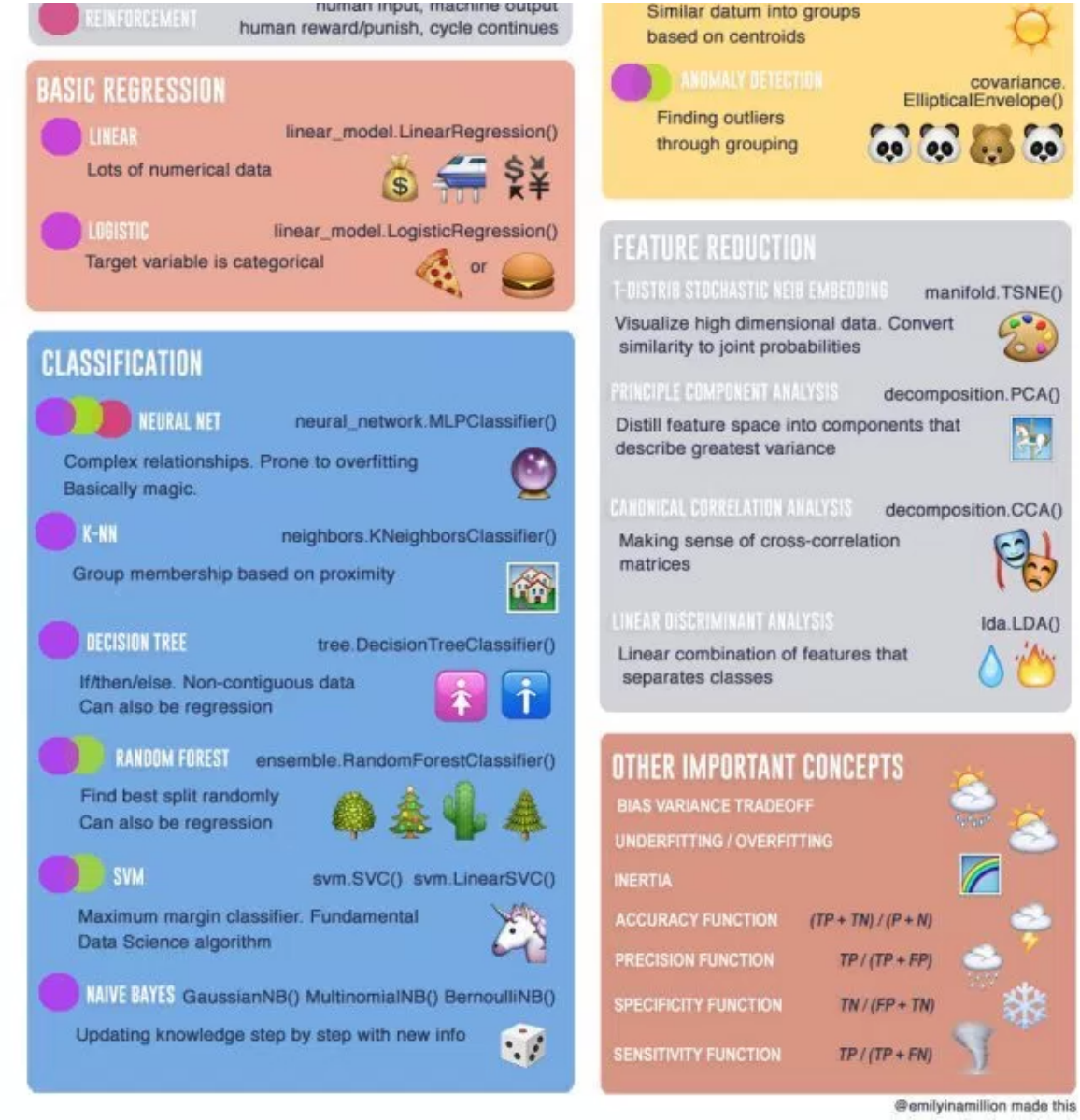
Quadratic fn.: $F(\mathrm{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{A}\,\mathbf{x} + \mathbf{d}^{\mathrm{T}}\mathbf{x} + \mathrm{c}$

$$\nabla \mathrm{F}(\mathrm{x}) = \mathbf{Ax} + \mathbf{d},\ \nabla^2 \mathrm{F}(\mathrm{x}) = \mathbf{A},\ \lambda_{min} \leq \frac{\mathbf{p}^T \mathbf{A}\,\mathbf{p}}{\|\mathbf{p}\|^2} \leq \lambda_{max}$$

Памятка по нейронным сетям

Обзор по машинному обучению



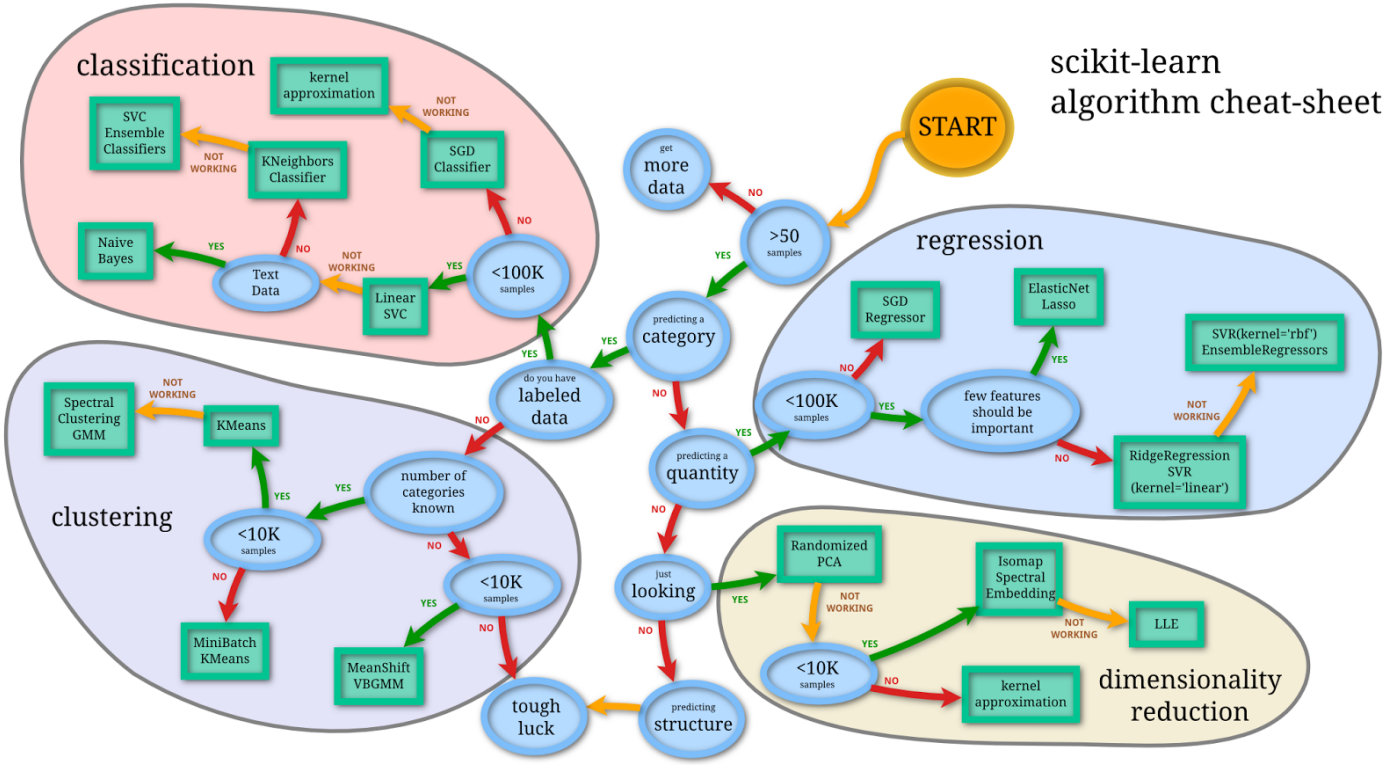


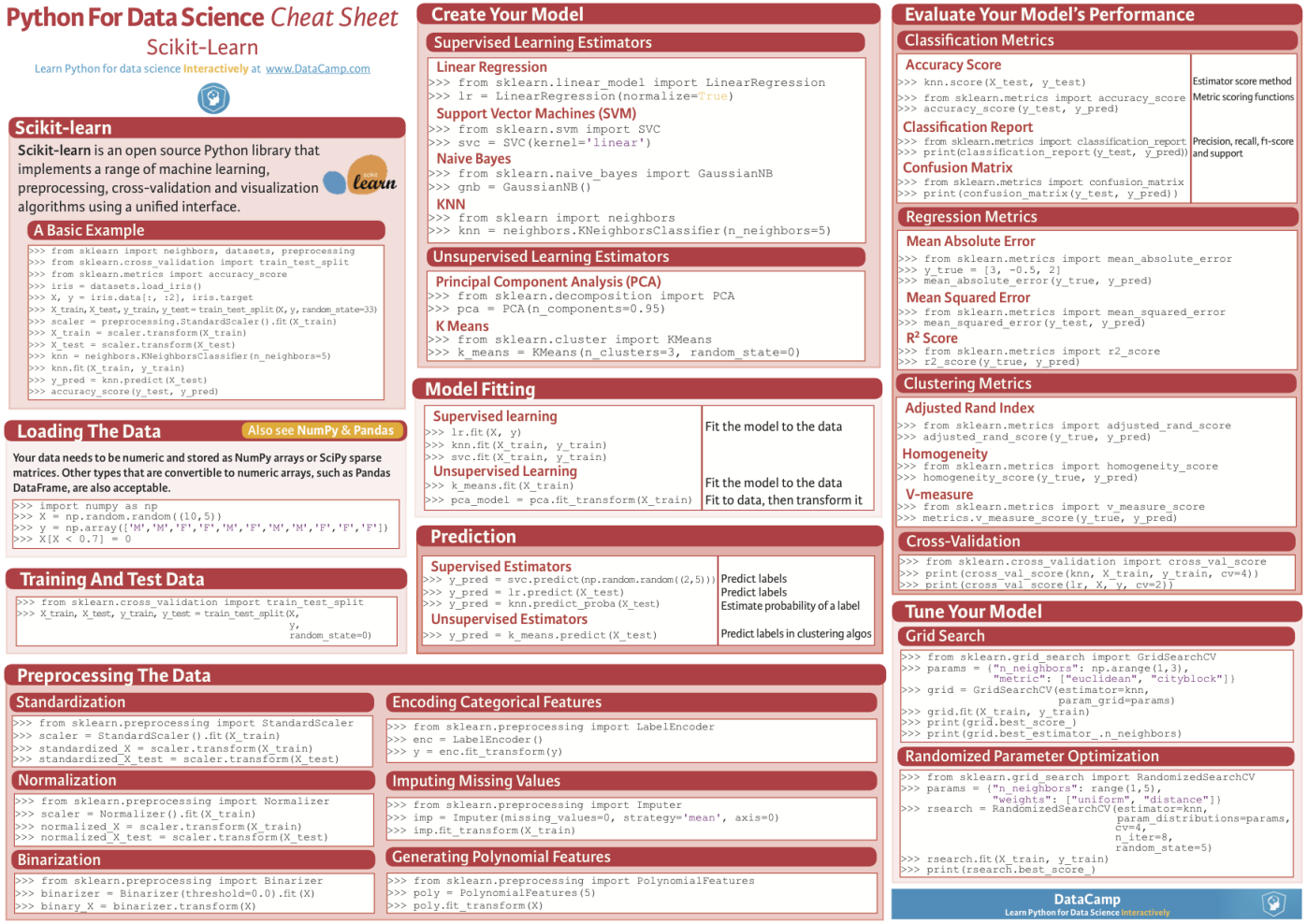
Памятка по машинному обучению

Алгоритм Scikit-learn

Эта памятка по машинному обучению поможет найти подходящий алгоритм для оценки, что является наиболее сложной частью работы. Блок-схема поможет проверить документацию и задаст общее направление по каждому алгоритму. Это позволит лучше понять стоящие перед вами проблемы и способы их решения.

Scikit-learn (ранее известная как **scikits.learn**) — это бесплатная библиотека машинного обучения для Python. В нее входят различные виды классификации, регрессии и алгоритмы кластеризации, включающие метод опорных векторов, алгоритм Random forest («случайный лес»), градиентный бустинг, метод *k*-средних и DBSCAN. Scikit-learn предназначена для взаимодействия с вычислительными и научными библиотеками Python NumPy и SciPy.

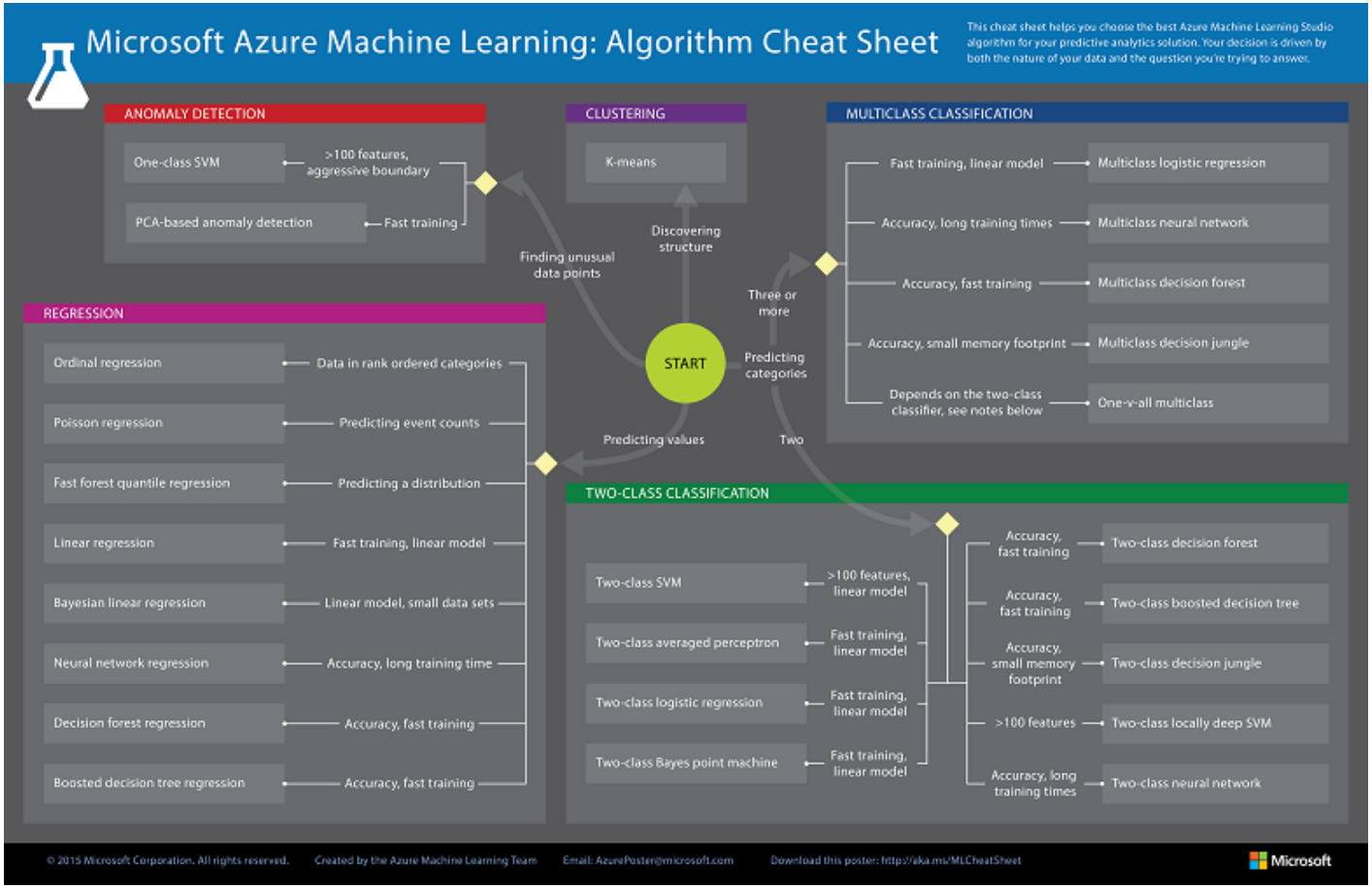




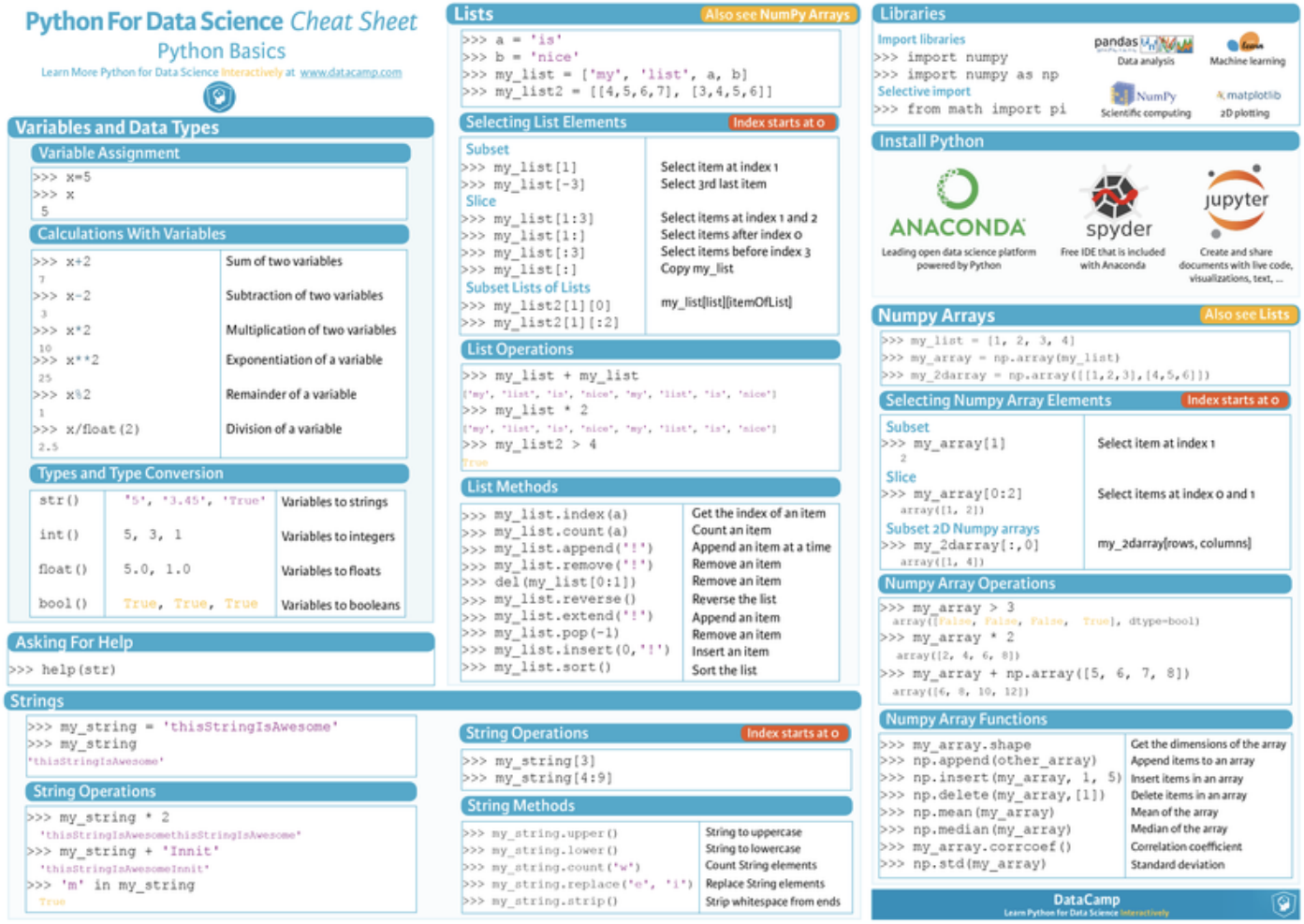
Памятка по Scikit-learn

Памятка по алгоритмам машинного обучения

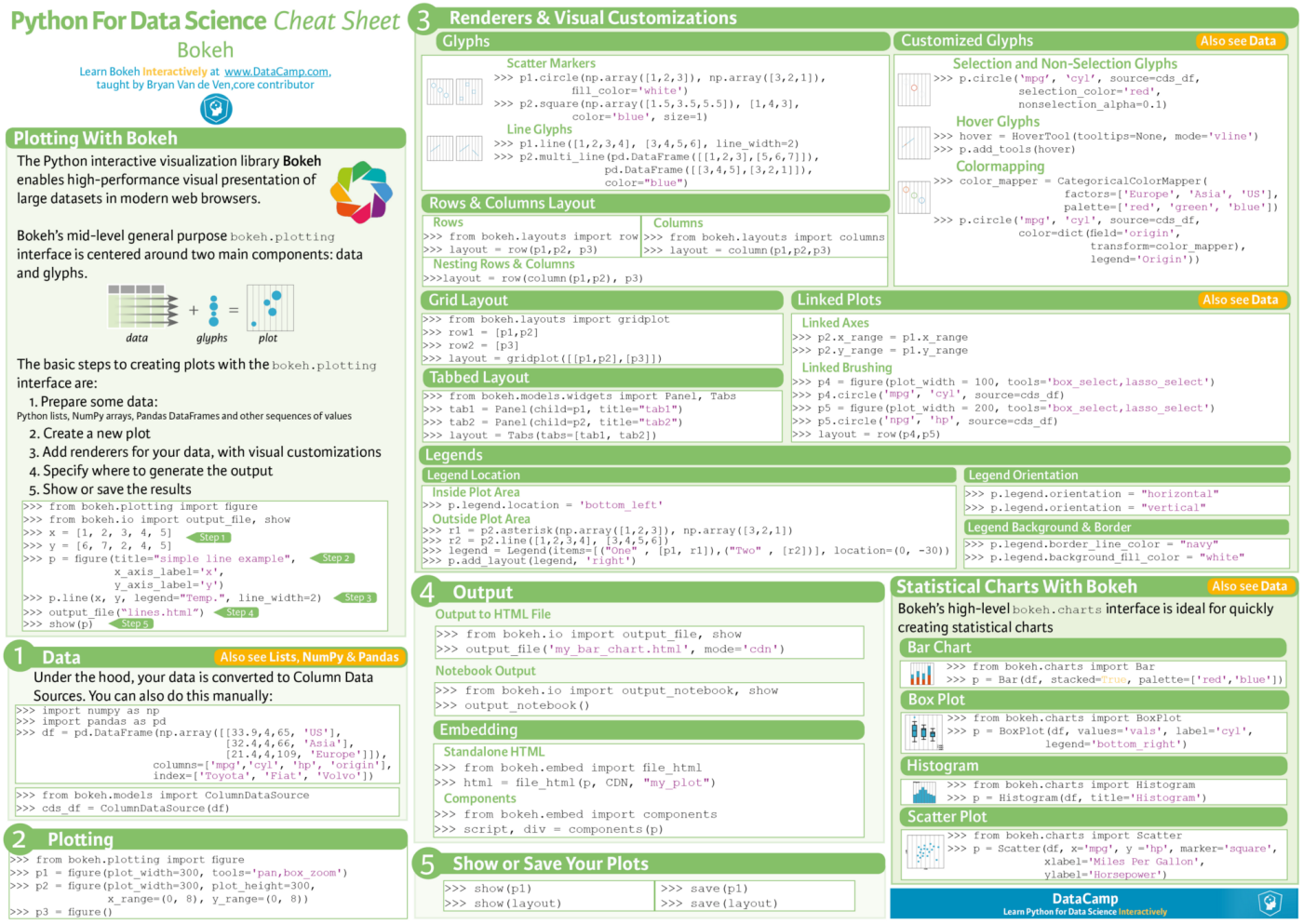
Эта памятка от Microsoft Azure поможет с выбором подходящих алгоритмов машинного обучения для вашего предсказательного аналитического решения. Вначале памятка спросит о природе данных, а затем посоветует наилучший алгоритм.



Python для Data Science



Памятка по Python для Data Science



Памятка по большим данным

TensorFlow

В мае 2017 года Google анонсировал TPU второго поколения, а также их доступность в Google Compute Engine. TPU второго поколения обладают производительностью до 180 терафлопов, а при кластеризации по 64 TPU — до 11,5 петафлопов.

About

TensorFlow

TensorFlow™ is an open source software library for numerical computation using data flow graphs. TensorFlow was originally developed for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

Skflow

Scikit Flow provides a set of high level model classes that you can use to easily integrate with your existing Scikit-learn pipeline code. Scikit Flow is a simplified interface for TensorFlow, to get people started on predictive analytics and data mining. Scikit Flow has been merged into TensorFlow since version 0.8 and now called TensorFlow Learn.

Keras

Keras is a minimalist, highly modular neural networks library, written in Python and capable of running on top of either TensorFlow or Theano

Installation

How to install new package in Python:

`pip install <package-name>`

Example: `pip install requests`

How to install tensorflow?

device = **cpu/gpu**

python_version = **cp27/cp34**

```
sudo pip install
https://storage.googleapis.com/
tensorflow/linux/$device/tensorflow-
0.8.0-$python_version-none-linux_x86
_64.whl
```

How to install Skflow

`pip install sklearn`

How to install Keras

`pip install keras`

update ~/.keras/keras.json - replace “theano” by “tensorflow”

Helpers

Python helper

Important functions

`id(object)`
Return the identity of an object. This is guaranteed to be unique among simultaneously existing objects.
`import __builtin__`
`dir(__builtin__)`
Other built-in functions

TensorFlow

Main classes

`tf.Graph()`
`tf.Operation()`
`tf.Tensor()`
`tf.Session()`

Some useful functions

`tf.get_default_session()`
`tf.get_default_graph()`
`tf.reset_default_graph()`
`ops.reset_default_graph()`
`tf.device(“/cpu:0”)`
`tf.name_scope(value)`
`tf.convert_to_tensor(value)`

TensorFlow Optimizers

`GradientDescentOptimizer`
`AdadeltaOptimizer`
`AdagradOptimizer`
`MomentumOptimizer`
`AdamOptimizer`
`FtrlOptimizer`
`RMSPropOptimizer`

Reduction

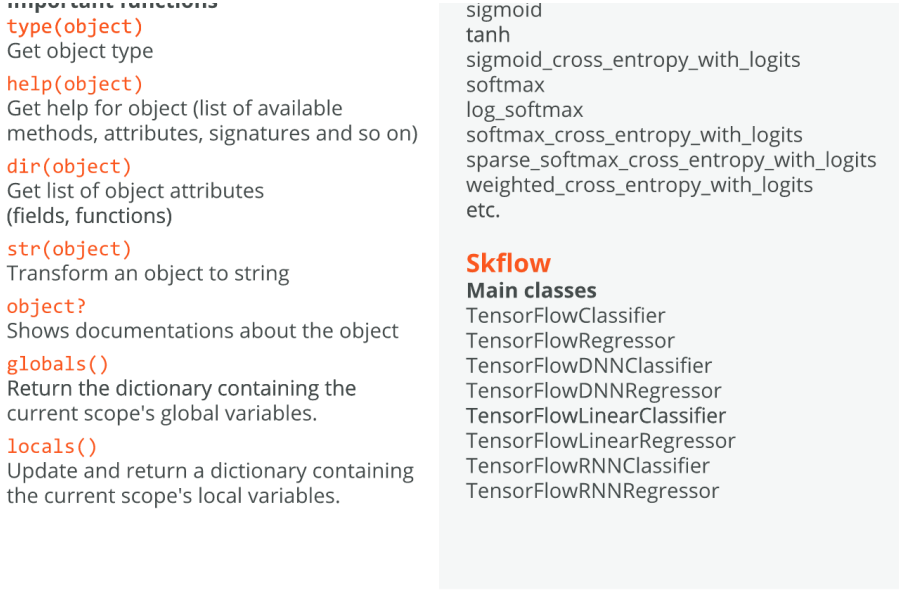
`reduce_sum`
`reduce_prod`
`reduce_min`
`reduce_max`
`reduce_mean`
`reduce_all`
`reduce_any`
`accumulate_n`

Activation functions

`tf.nn?`
relu
relu6
elu
softplus
softsign
dropout
bias_add

TensorFlowEstimator

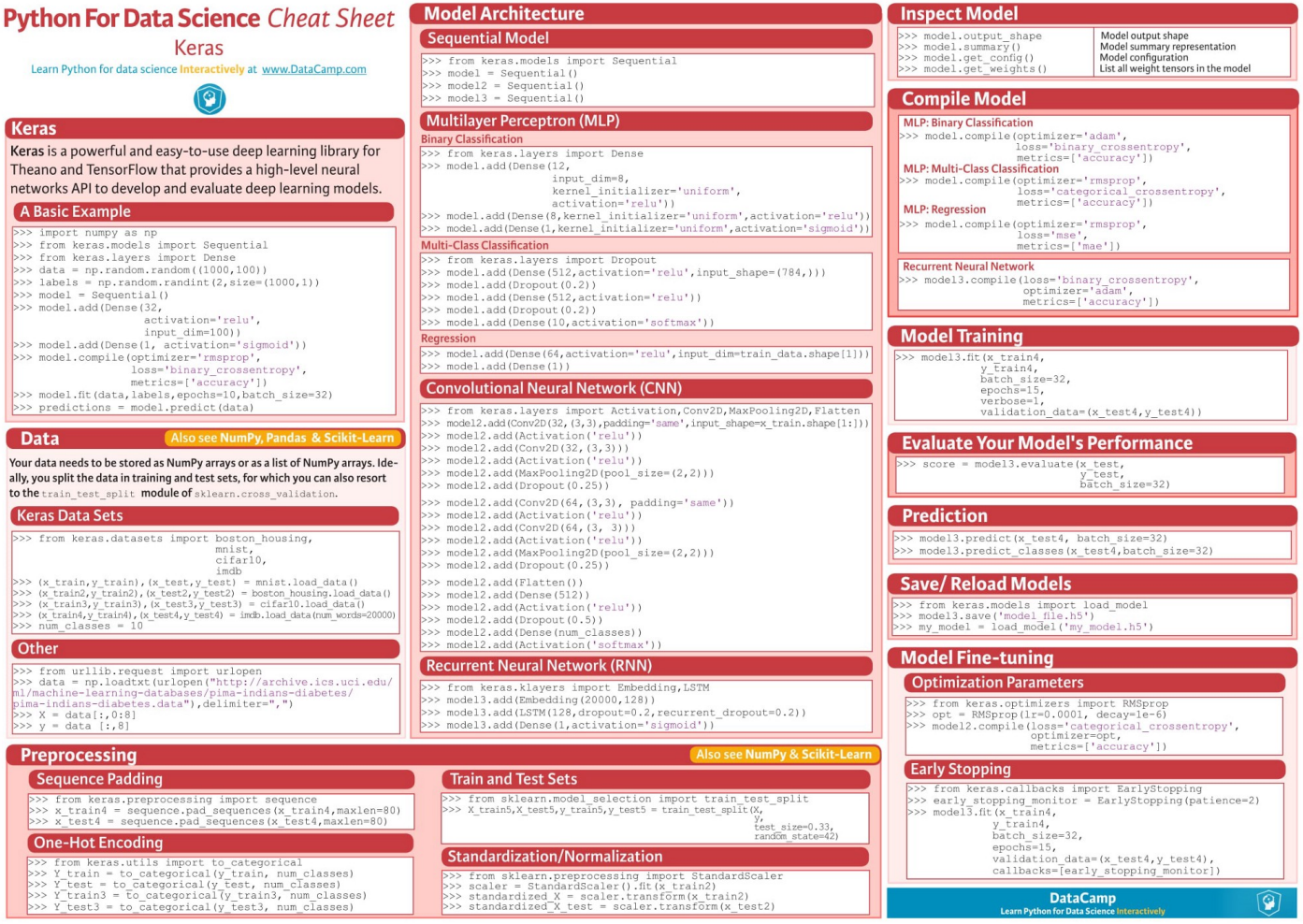
Each classifier and regressor have following fields
n_classes=0 (Regressor), n_classes are expected to be input (Classifiers)
batch_size=32,
steps=200, // except TensorFlowRNNClassifier - there is 50
optimizer='Adagrad',
learning_rate=0.1,



Памятка по TensorFlow

Keras

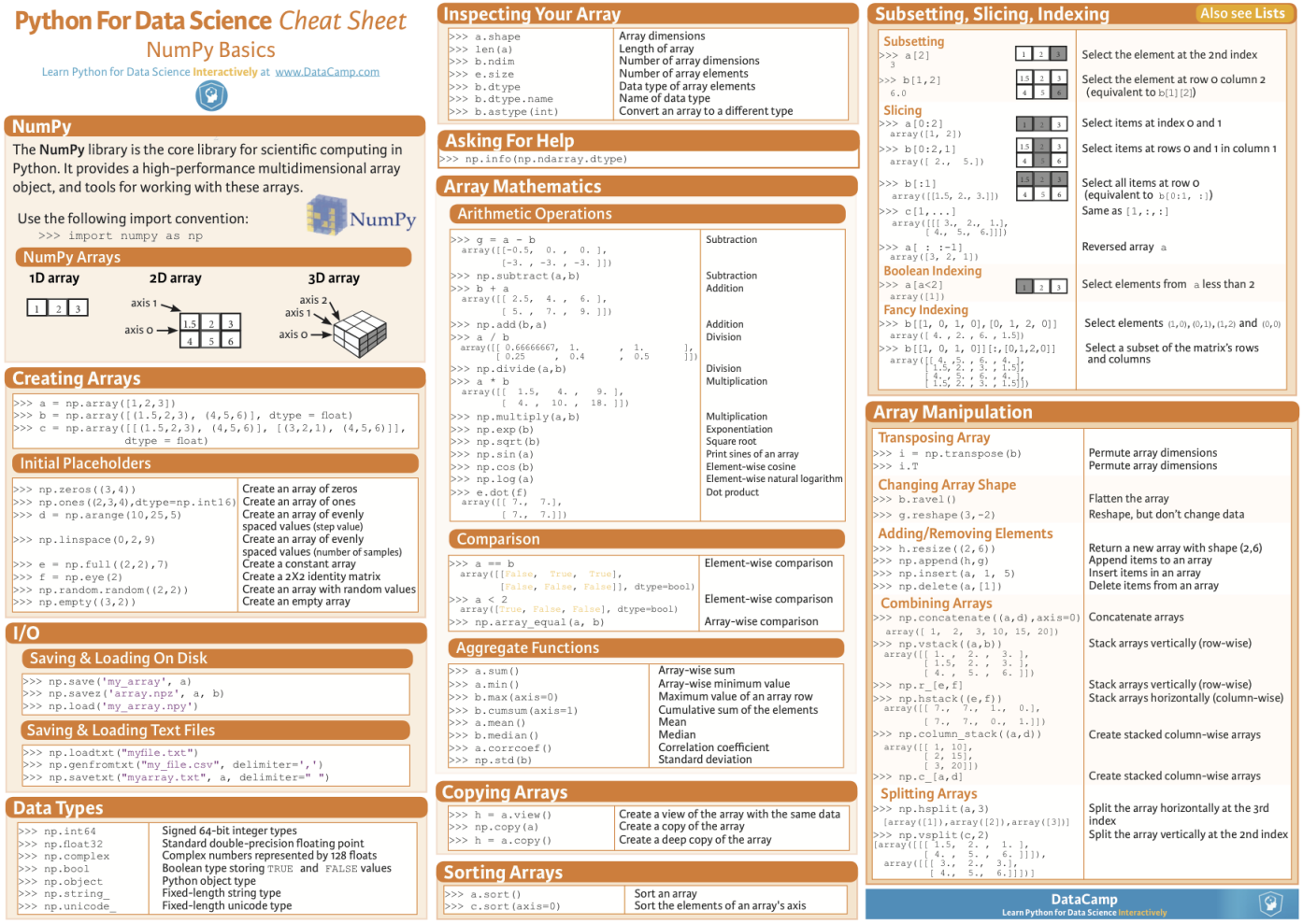
В 2017 году команда TensorFlow в Google решила внедрить поддержку Keras в основную библиотеку TensorFlow. Шолле (Chollet) объяснил, что Keras является, скорее, интерфейсом, чем сквозной системой машинного обучения. Он предоставляет высокоуровневый, более интуитивный набор абстракций, который упрощает конфигурирование нейронных сетей, вне зависимости от используемой в бэкенде библиотеки научных вычислений



NumPy

NumPy предназначен для Cpython, эталонной реализации Python, которая является не оптимизирующим интерпретатором байт-кода. Математические алгоритмы, написанные для этой версии Python, часто работают гораздо медленнее скомпилированных аналогов. Библиотека NumPy частично решает проблему скорости за счет многомерных массивов, а также функций и операторов, оптимизированных для работы с массивами. Необходимо будет переписать часть кода с

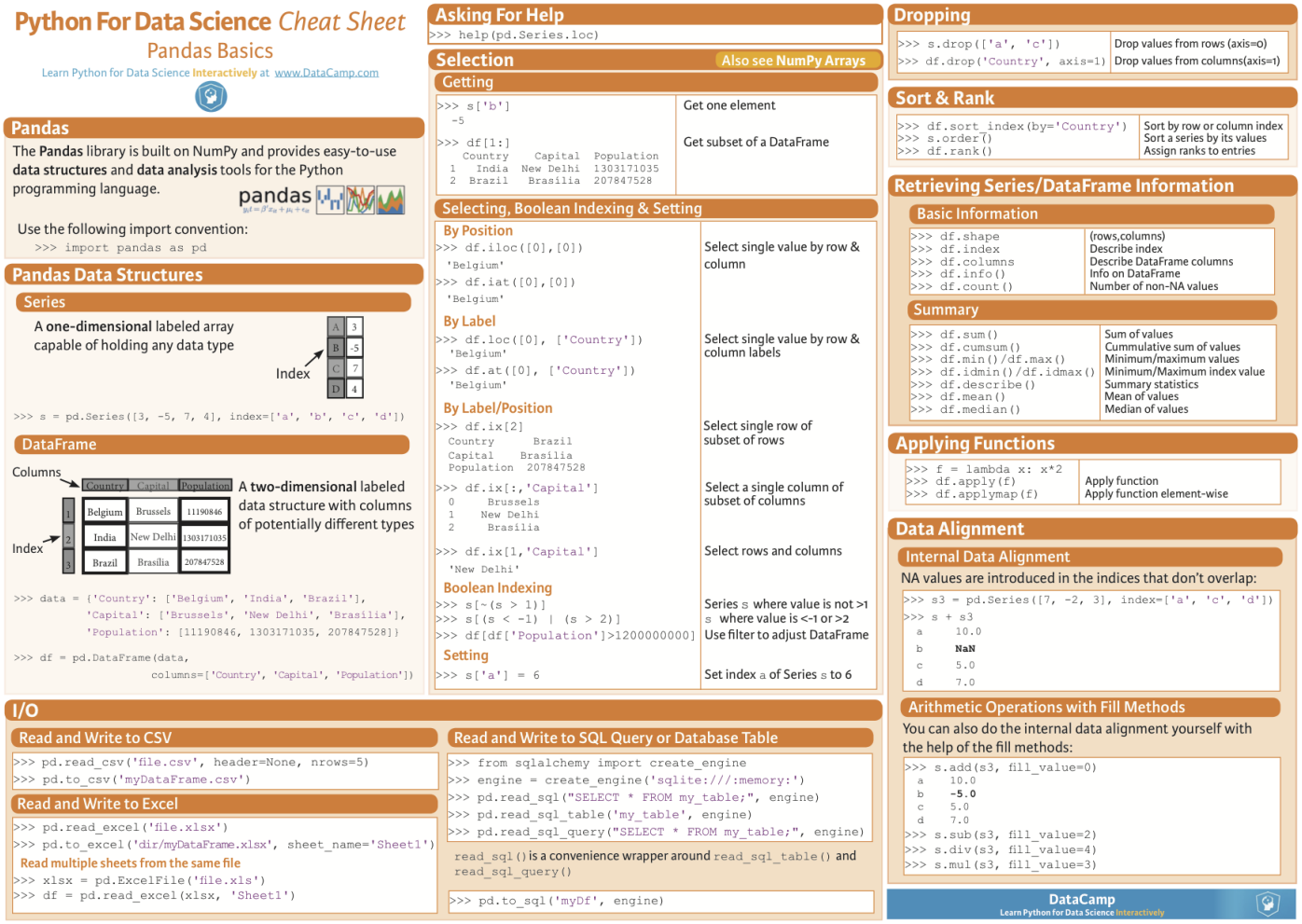
использованием NumPy, в основном внутренние циклы.



Памятка по NumPy

Pandas

Название «Pandas» происходит от эконометрического термина "panel data", который применяется для многомерных структурированных наборов данных.



Памятка по Pandas

Data Wrangling

Data Wrangling (*«выпас» данных, первичная обработка данных*) — этот термин начинает проникать в поп-культуру. В фильме 2017 «Конг: Остров черепа» один из героев представлен как «Стив Вудвард, наш data wrangler».

Data Wrangling with pandas Cheat Sheet

http://pandas.pydata.org

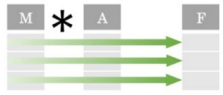
Tidy Data – A foundation for wrangling in pandas

In a tidy data set:

Each **variable** is saved in its own **column**

Each **observation** is saved in its own **row**

Tidy data complements pandas's **vectorized operations**. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas.



M * A

Syntax – Creating DataFrames

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
        {"a" : [4 ,5, 6],
         "b" : [7, 8, 9],
         "c" : [10, 11, 12]},
       index = [1, 2, 3])
```

Specify values for each column.

```
df = pd.DataFrame(
    [[4, 7, 10],
     [5, 8, 11],
     [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
```

Specify values for each row.

		a	b	c
n	v			
d	1	4	7	10
	2	5	8	11
e	2	6	9	12

```
df = pd.DataFrame(
        {"a" : [4 ,5, 6],
         "b" : [7, 8, 9],
         "c" : [10, 11, 12]},
index = pd.MultiIndex.from_tuples(
        [('d',1),('d',2),('e',2)],
          names=['n','v'])))
```

Create DataFrame with a MultiIndex

Method Chaining

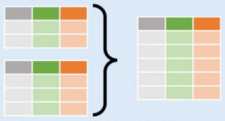
Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.

```
df = (pd.melt(df)
        .rename(columns={
                'variable' : 'var',
                'value' : 'val'})
        .query('val >= 200')
    )
```

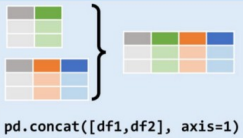
Reshaping Data – Change the layout of a data set

pd.melt(df)
Gather columns into rows.

df.pivot(columns='var', values='val')
Spread rows into columns.



pd.concat([df1,df2])
Append rows of DataFrames



pd.concat([df1,df2], axis=1)
Append columns of DataFrames

df.sort_values('mpg')
Order rows by values of a column (low to high).

df.sort_values('mpg',ascending=False)
Order rows by values of a column (high to low).

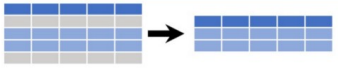
df.rename(columns = {'y':'year'})
Rename the columns of a DataFrame

df.sort_index()
Sort the index of a DataFrame

df.reset_index()
Reset index of DataFrame to row numbers, moving index to columns.

df.drop(['Length','Height'], axis=1)
Drop columns from DataFrame

Subset Observations (Rows)



df[df.Length > 7]
Extract rows that meet logical criteria.

df.drop_duplicates()
Remove duplicate rows (only considers columns).

df.head(n)
Select first n rows.

df.tail(n)
Select last n rows.

df.sample(frac=0.5)
Randomly select fraction of rows.

df.sample(n=10)
Randomly select n rows.

df.iloc[10:20]
Select rows by position.

df.nlargest(n, 'value')
Select and order top n entries.

df.nsmallest(n, 'value')
Select and order bottom n entries.

Logic in Python (and pandas)			
<	Less than	!=	Not equal to
>	Greater than	df.column.isin(*values*)	Group membership
==	Equals	pd.isnull(*obj*)	Is NaN
<=	Less than or equals	pd.notnull(*obj*)	Is not NaN
>=	Greater than or equals	&,|,~,^,df.any(),df.all()	Logical and, or, not, xor, any, all

Subset Variables (Columns)

df[['width','length','species']]
Select multiple columns with specific names.

df['width'] *or* **df.width**
Select single column with specific name.

df.filter(regex='*regex*')
Select columns whose name matches regular expression *regex*.

regex (Regular Expressions) Examples	
'\.'	Matches strings containing a period '.'
'Length$'	Matches strings ending with word 'Length'
'^Sepal'	Matches strings beginning with the word 'Sepal'
'^x[1-5]$'	Matches strings beginning with 'x' and ending with 1,2,3,4,5
''^(?!Species$).*'	Matches strings except the string 'Species'

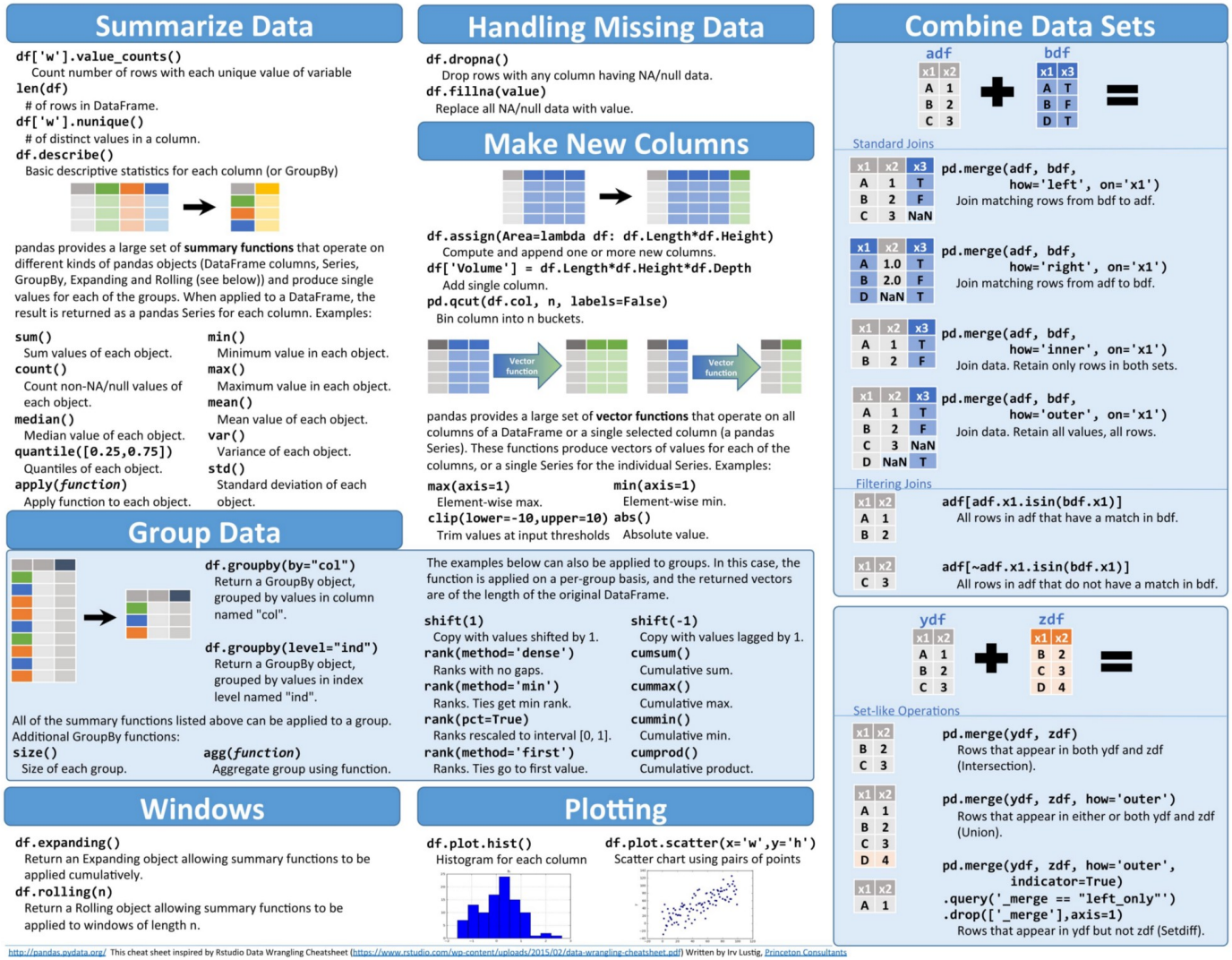
df.loc[:,'x2':'x4']
Select all columns between x2 and x4 (inclusive).

df.iloc[:,[1,2,5]]
Select columns in positions 1, 2 and 5 (first column is 0).

df.loc[df['a'] > 10, ['a','c']]
Select rows meeting logical condition, and only the specific columns .

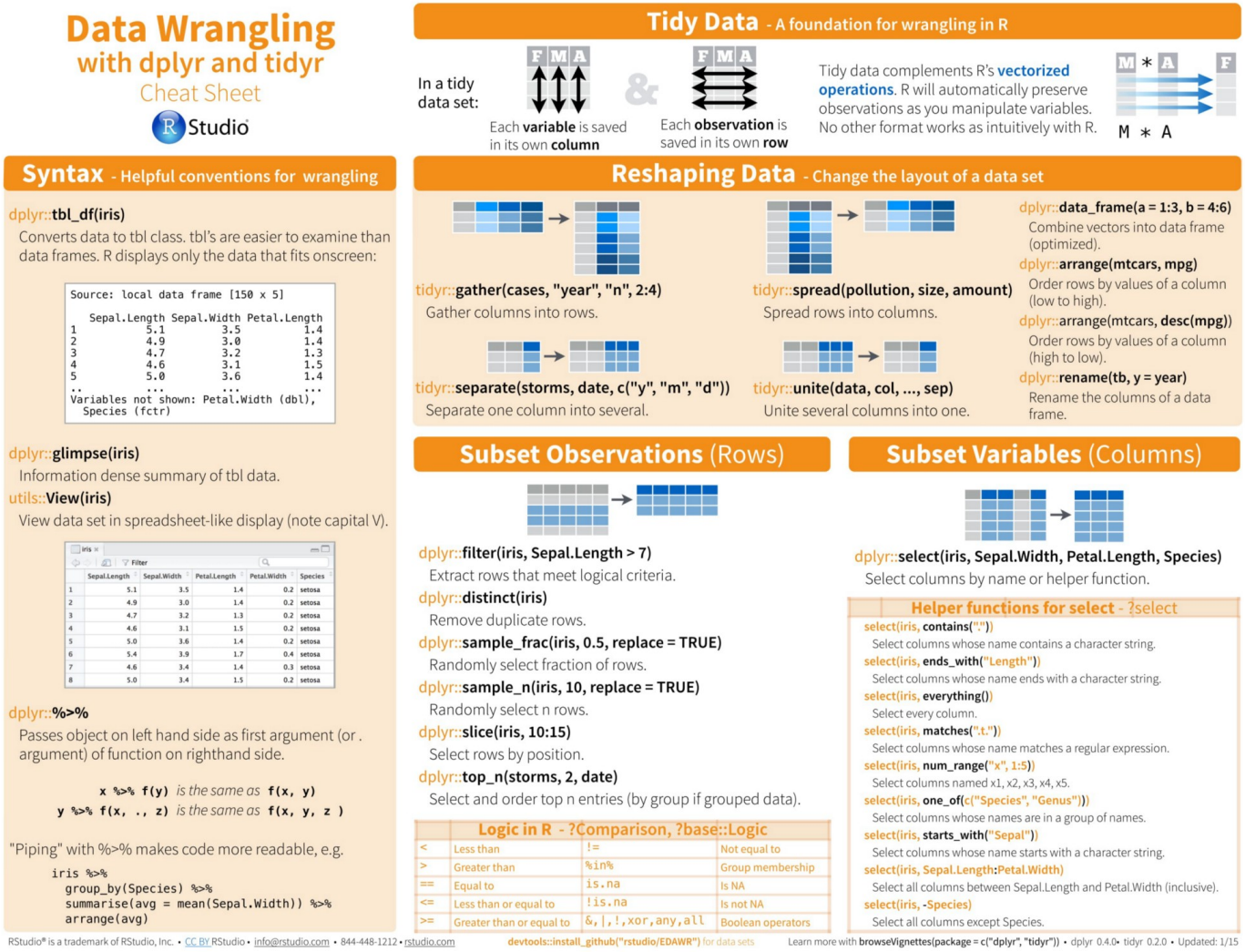
http://pandas.pydata.org/ This cheat sheet inspired by Rstudio Data Wrangling Cheatsheet (https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf) Written by Irv Lustig, Princeton Consultants

Памятка по Data Wrangling

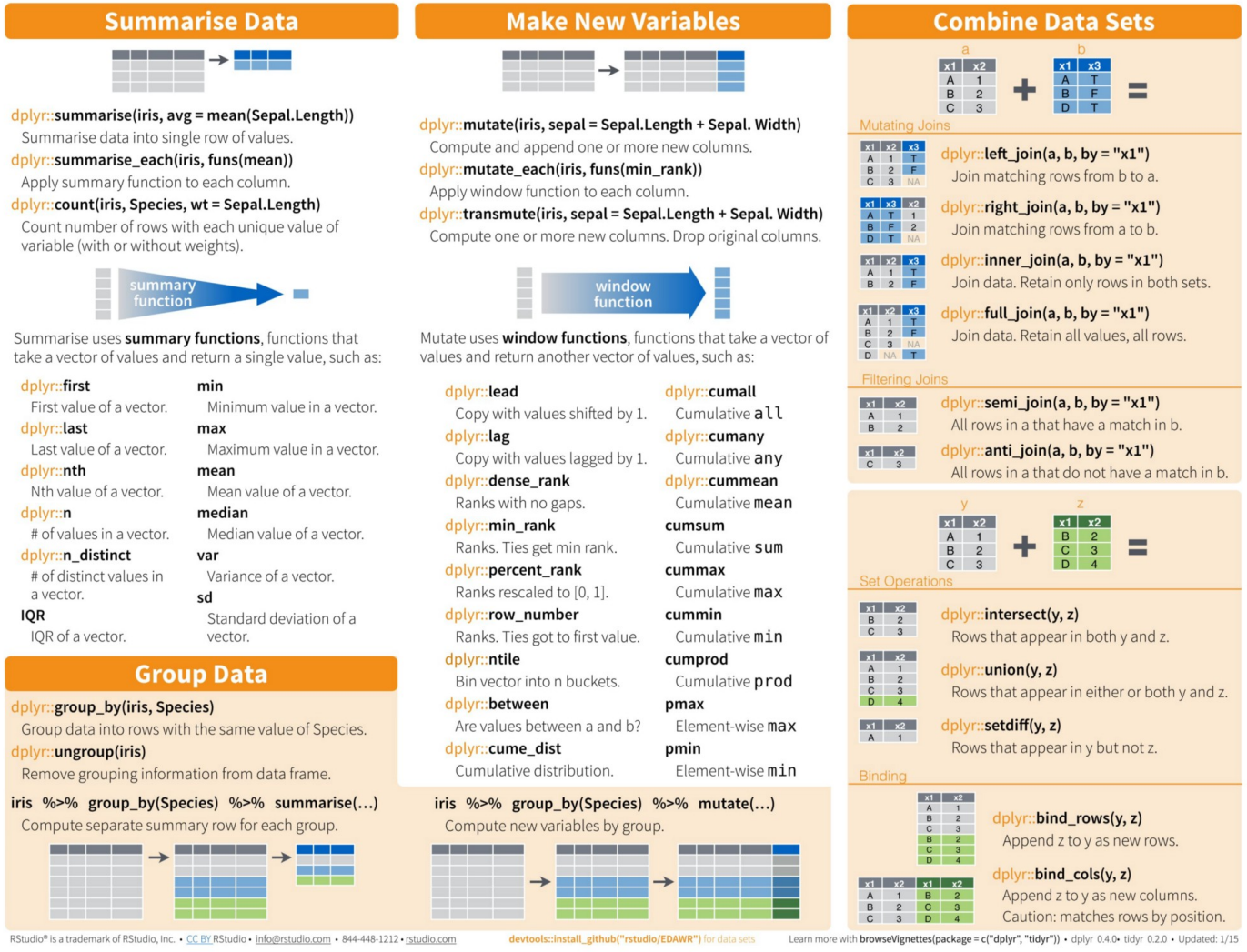


Памятка по Pandas Data Wrangling

Data Wrangling с помощью dplyr и tidyr



Памятка по Data Wrangling с dplyr и tidyr

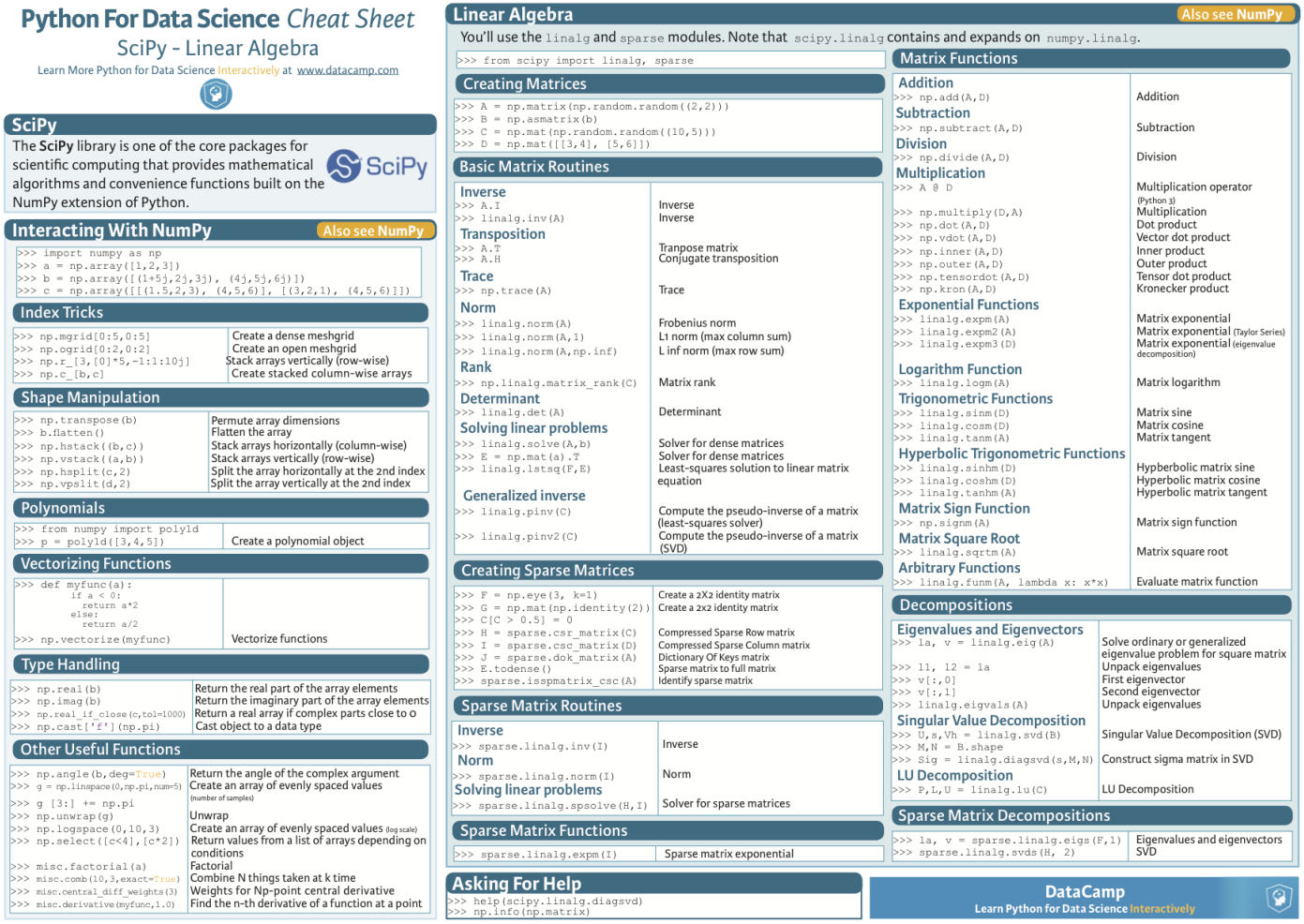


Памятка по Data Wrangling с dplyr и tidyr

SciPy

В основе SciPy лежит объект-массив NumPy. Эта библиотека является частью стека NumPy, который включает такие инструменты, как Matplotlib, Pandas и SymPy, а также расширяющийся набор библиотек

для научных вычислений. У стека NumPy и приложений MATLAB, GNU Octave и Scilab одна и та же аудитория пользователей. Стек NumPy также иногда называют стеком SciPy.



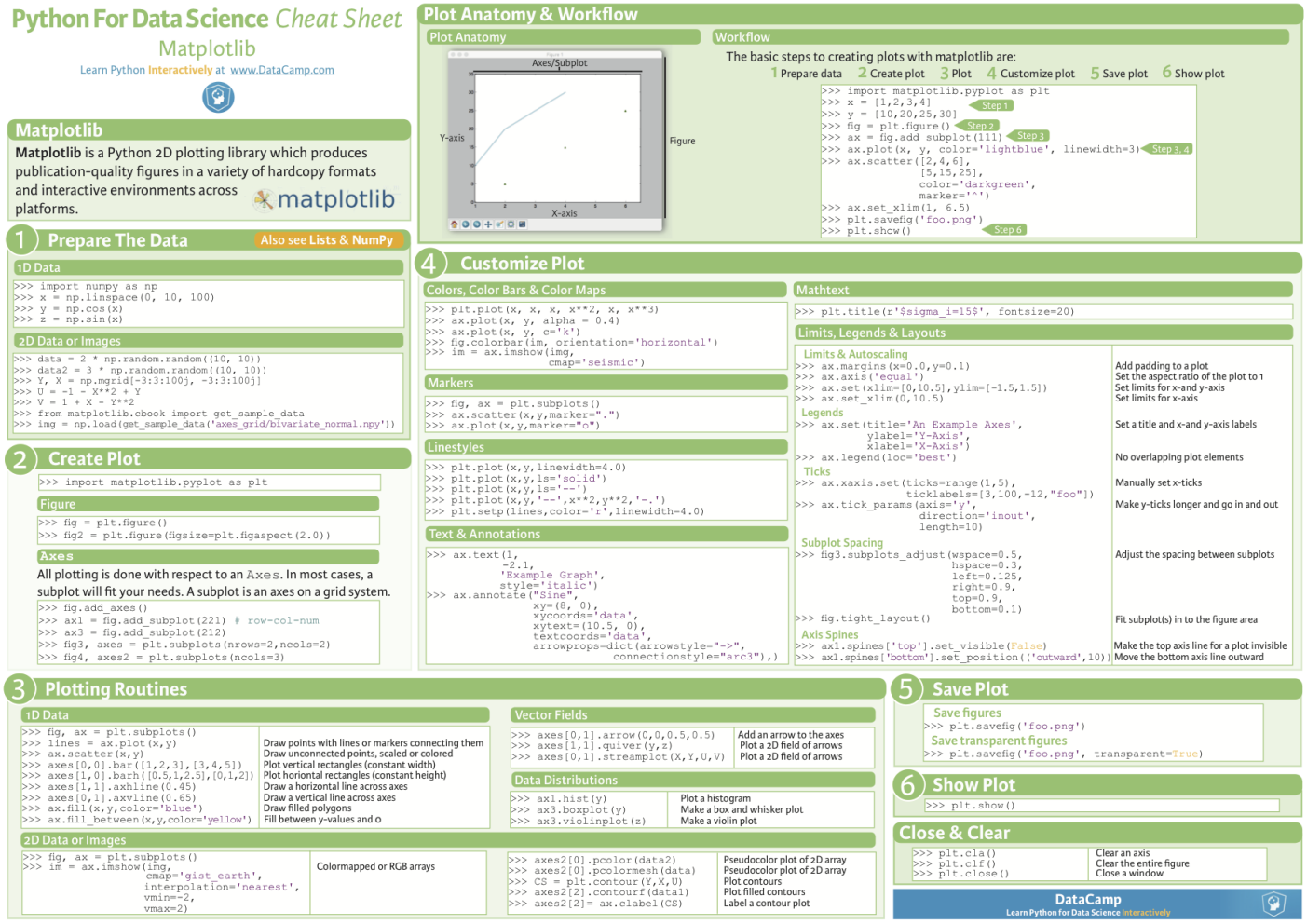
Памятка по SciPy

Matplotlib

Matplotlib — это библиотека для построения графиков для Python и его вычислительного

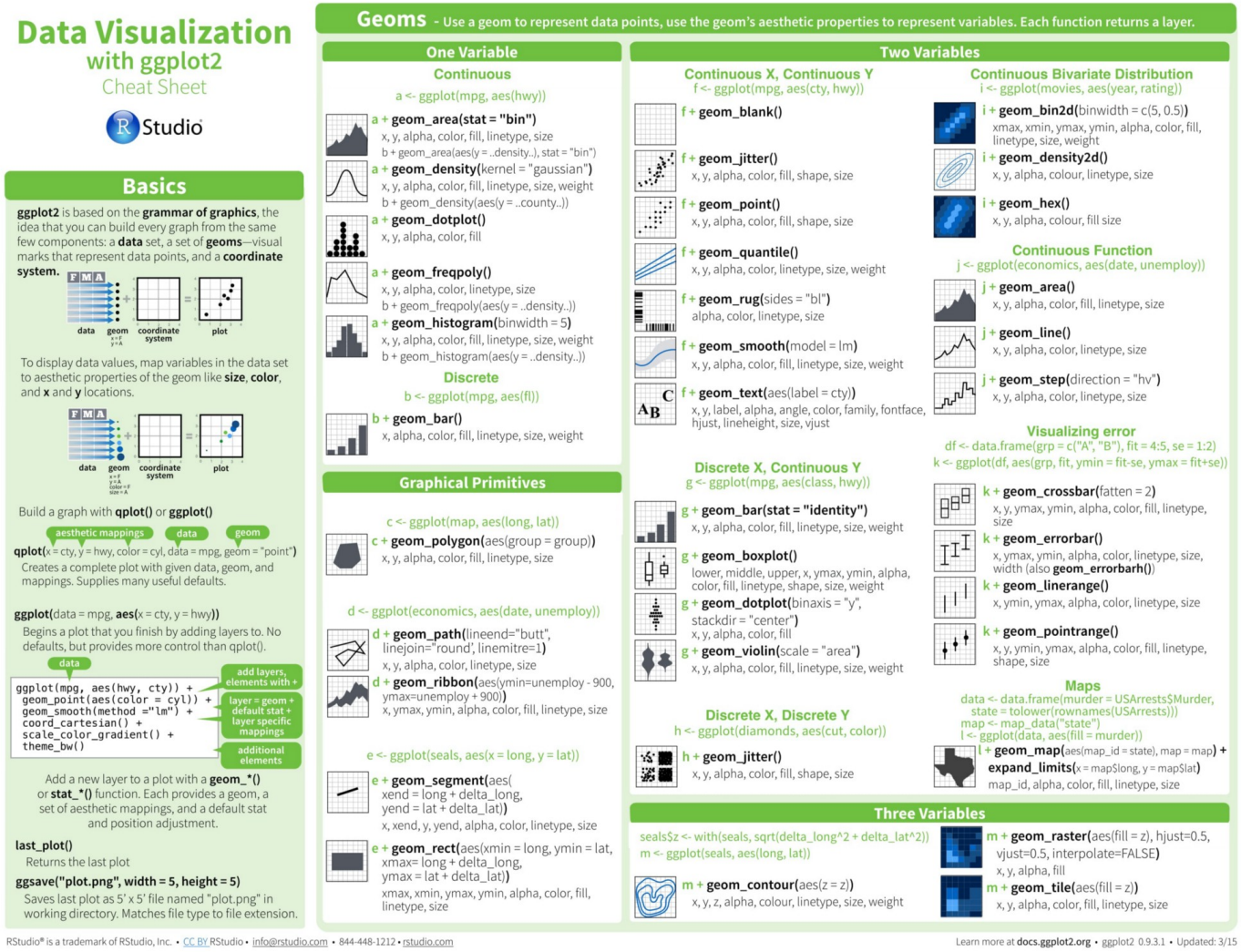
математического расширения NumPy. Она предоставляет объектно-ориентированный API для встраивания графиков в приложения с использованием универсальных GUI-инструментов, таких как Tkinter, wxPython, Qt, или GTK+. Существует также процедурный интерфейс «pylab» на основе конечного автомата (например, OpenGL), разработанный так, чтобы походить MATLAB, хотя его использование не рекомендуется. SciPy использует matplotlib.

Pyplot — это модуль matplotlib, который предоставляет интерфейс наподобие MATLAB. Matplotlib применяется так же, как и MATLAB, позволяет использовать Python, и к тому же бесплатен.

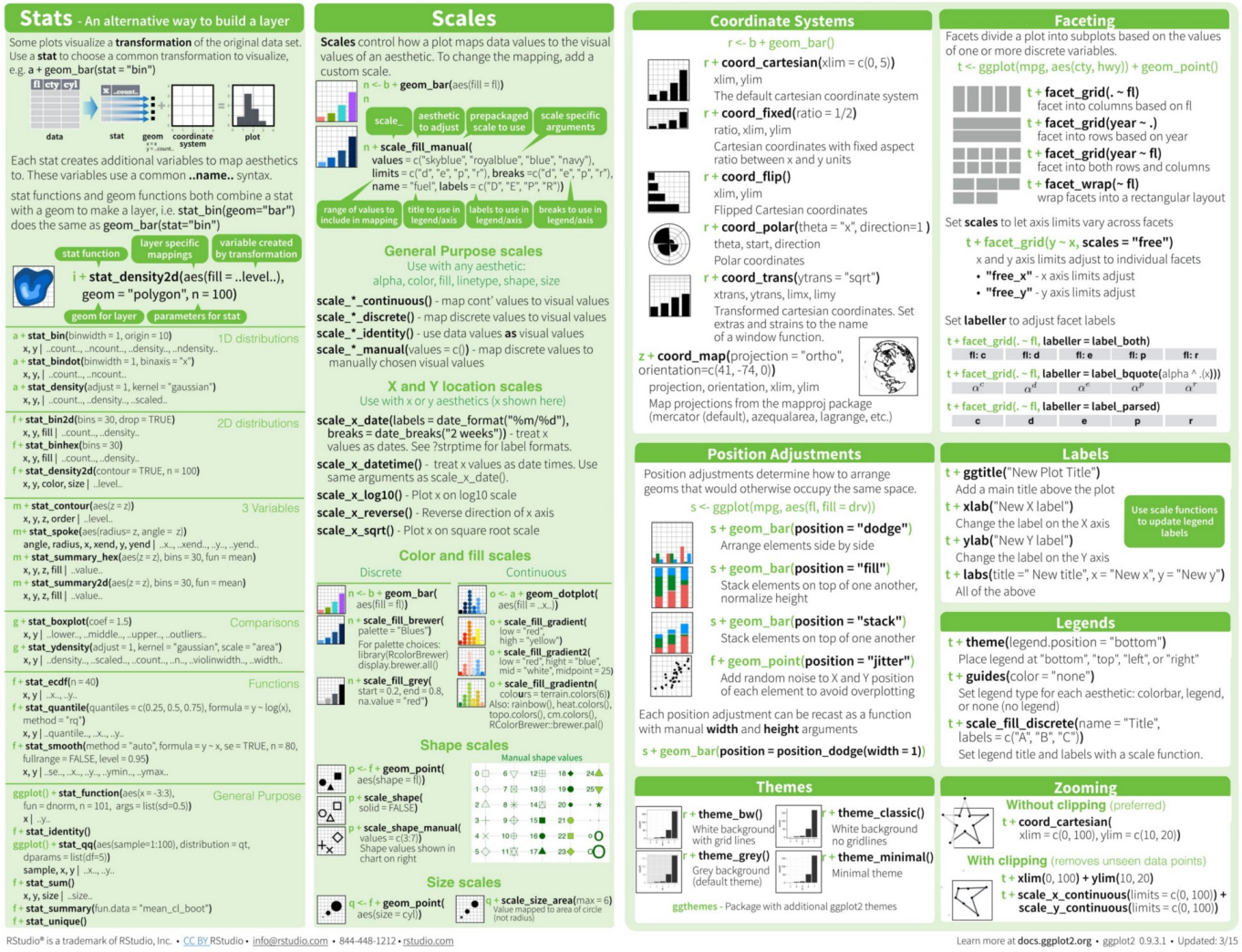


Памятка по Matplotlib

Визуализация данных

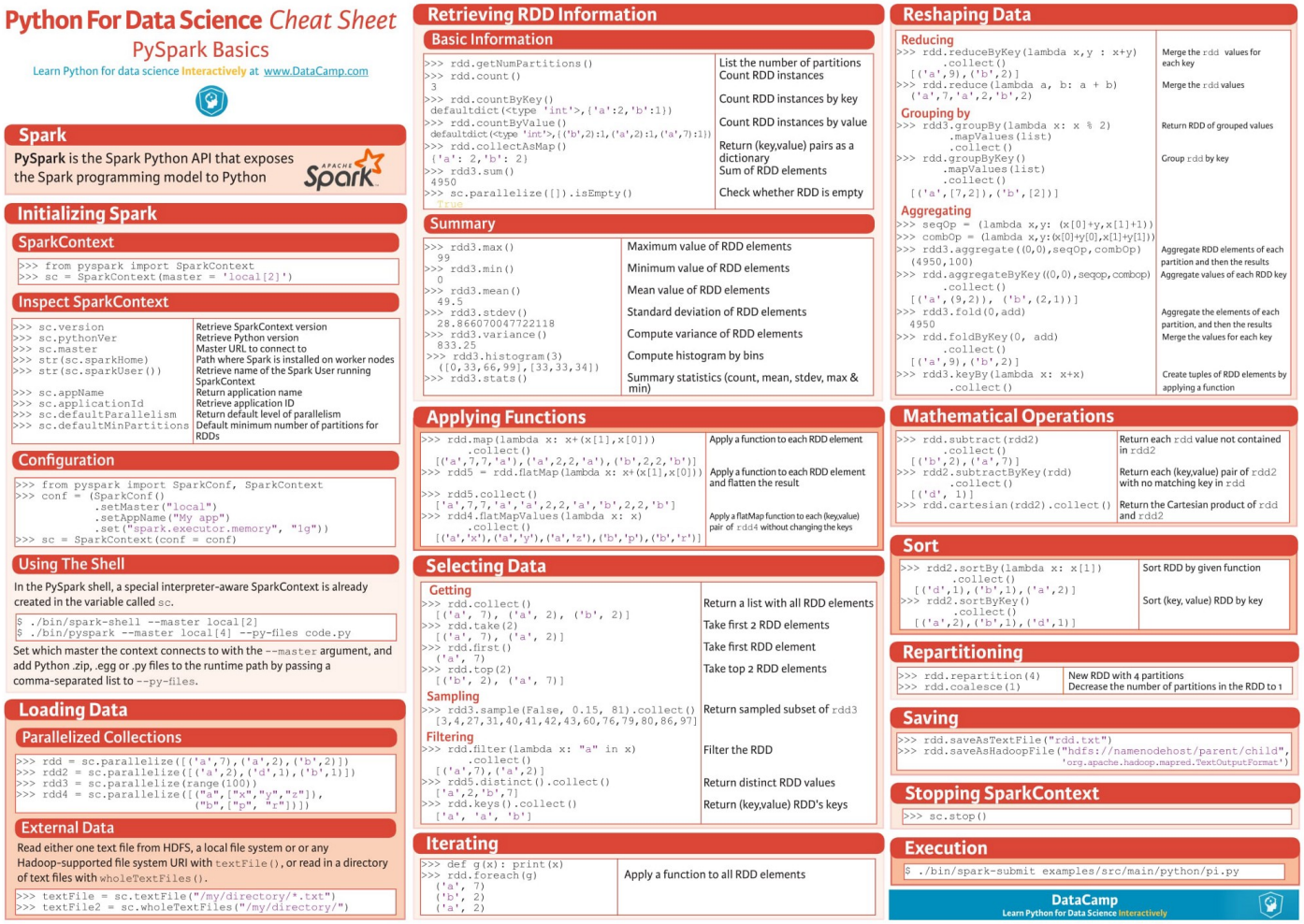


Памятка по визуализации данных



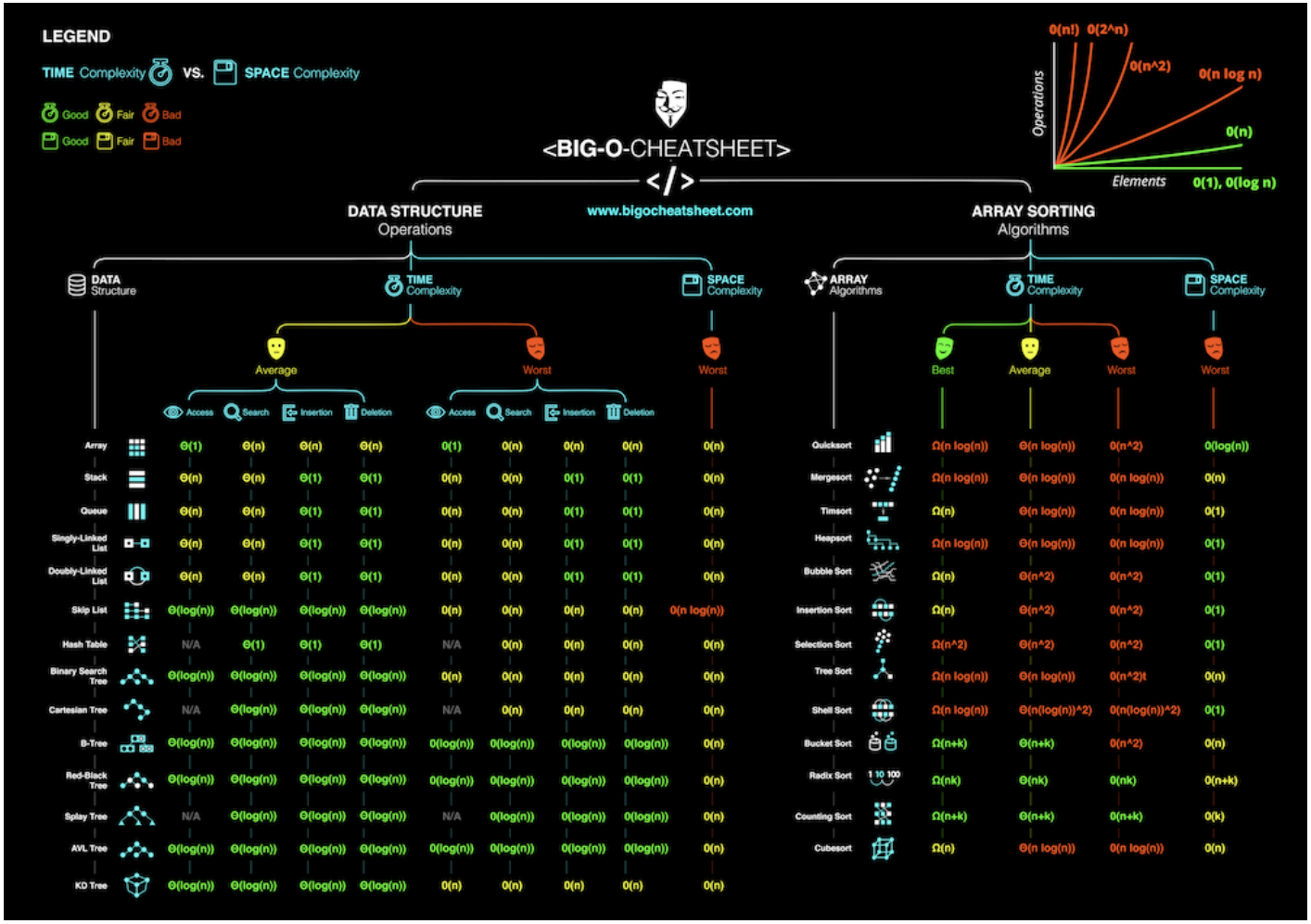
Памятка по ggplot

PySpark

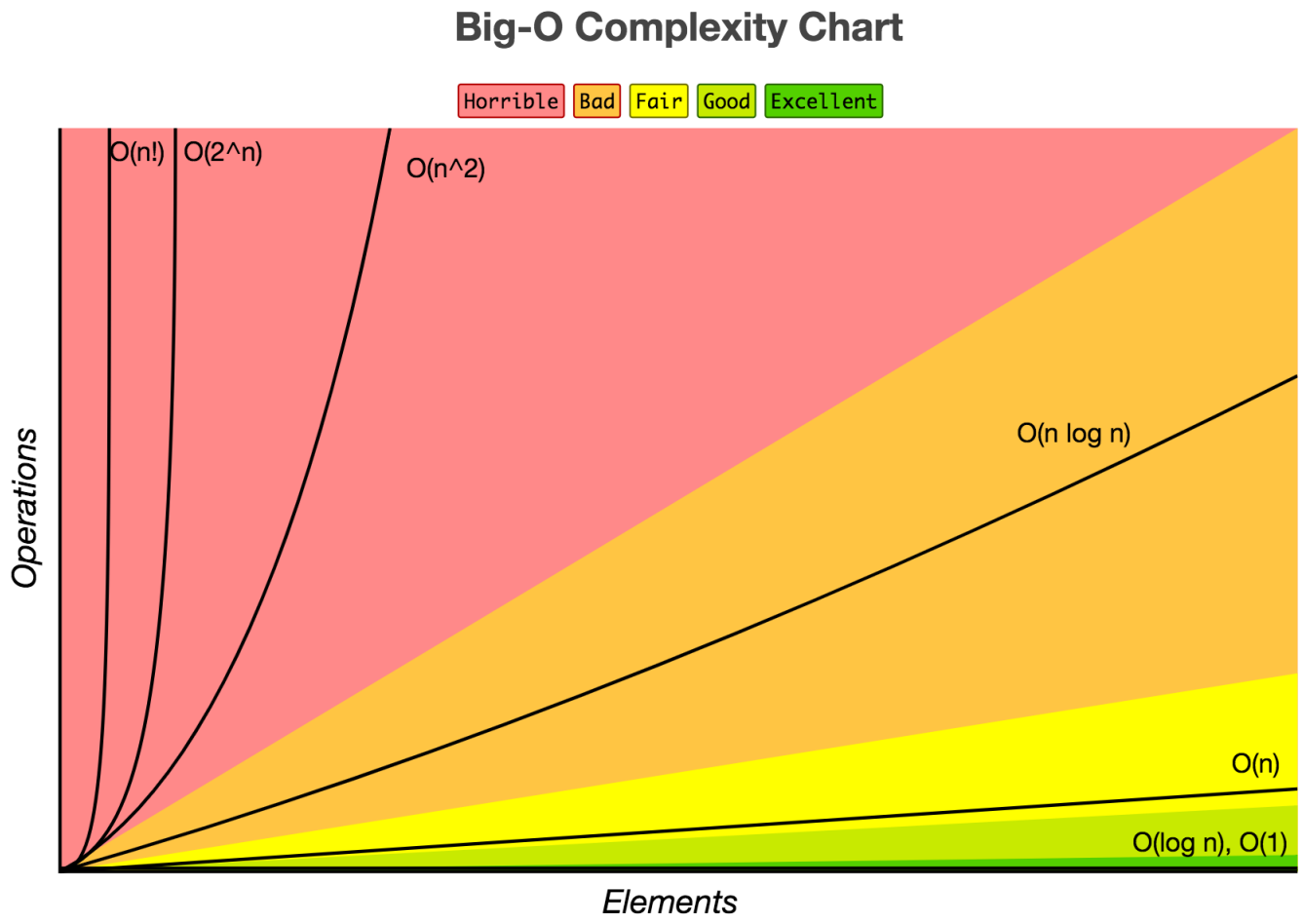


Памятка по PySpark

«О большое» (Big-O)



Памятка по сложности алгоритмов



Памятка по сложности алгоритмов

Common Data Structure Operations

Data Structure	Time Complexity								Space Complexity
	Average				Worst				Worst
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
Array	Θ(1)	Θ(n)	Θ(n)	Θ(n)	O(1)	O(n)	O(n)	O(n)	O(n)
Stack	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Queue	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Singly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Doubly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Skip List	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n log(n))
Hash Table	N/A	Θ(1)	Θ(1)	Θ(1)	N/A	O(n)	O(n)	O(n)	O(n)
Binary Search Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)
Cartesian Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(n)	O(n)	O(n)	O(n)
B-Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Red-Black Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Splay Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(log(n))	O(log(n))	O(log(n))	O(n)
AVL Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
KD Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)

Памятка по сложности операций со структурами данных в алгоритмах

Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	θ(n log(n))	O(n log(n))	O(n)

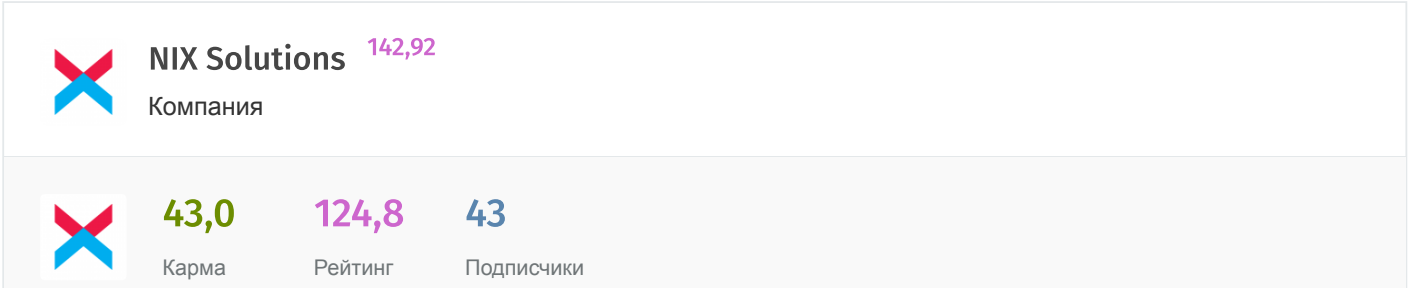
Памятка по сложности алгоритмов сортировки массива

Источники

Памятка по сложности алгоритмов

Памятка по Bokeh
Памятка по Data Science
Памятка по Data Wrangling
Памятка по Ggplot
Памятка по Keras
Памятка по машинному обучению
Памятка по машинному обучению
Памятка по машинному обучению
Памятка по Matplotlib
Памятка по нейросетям
Памятка по графам нейросетей
Нейросети
Памятка по Numpy
Памятка по Pandas
Памятка по Pandas
Памятка по Pyspark
Памятка по Scikit
Памятка по Scikit-learn
Памятка по Scipy
Памятка по TensorFlow

Метки: машинное обучение, модель, подсказка, памятка, большие данные, нейросеть



@NIX_Solutions
Пользователь

Поделиться публикацией

ПОХОЖИЕ ПУБЛИКАЦИИ

1 декабря 2017 в 18:49
Структура и модель выполнения .NET Core приложений
+40 20,8k 195 10

23 мая 2017 в 11:01
Как организовать большое React-приложение и сделать его масштабируемым
+29 25,2k 151 16

15 марта 2017 в 11:29
Модели памяти, лежащие в основе языков программирования
+33 24,4k 191 10

Комментарии 9

DmitriyDev 24.07.18 в 17:42 0

Большое человеческое спасибо!

phenik 24.07.18 в 17:44 +3

Напоминает шпаргалки для сдачи экзаменов.

НЛО прилетело и опубликовало эту надпись здесь

delph 24.07.18 в 22:18 0

Отличная сборка, спасибо!

Dark_Daiver 25.07.18 в 08:03 +4

Если честно, то я бы убрал первую картинку, т.к. она откровенно вредная. Мало того, что она толком ничего не объясняет — одной топологии недостаточно чтобы объяснить чем отличается один тип нейронной сети от другой, так еще и топология местами нарисована с ошибками.

К примеру
— Что такое kernel в сверточной сети (deep convolutional network), и почему он идет перед первыми свертками?
— Чем отличаются feed forward и radial basis network?
— Почему у SVM несколько скрытых слоев? Если речь идет о нелинейных SVM, то возможно стоило влепить kernel сюда? Что за хитрые скрытые слои, которые принимают по одному входу? Ну и строго говоря причислять SVM к Neural Networks это несколько грубо.
— Почему в GAN дважды встречается Match input output Cell? Что такое Match input output Cell в принципе? Если это просто штука которая сравнивает поэлементно входы-выходы, то в GAN такого в чистом виде не происходит.
— Почему skip connections в Deep Residual Network связывают только часть нейронов? Строго говоря, не совсем корректно показывать skip connections так же как и обычные соединения. В DRN skip connections это всегда сложение, более сложные случаи уже не называют residual (Dense connections, например).

> **Calc** 25.07.18 в 11:56 0
>
> Наоборот, тем кто далек от этого это визуальное представление, что есть что то кроме простого перцептрона и 2х уровневой сетки. А дальше можно самому искать информацию по каждому типу. Много

раз видел поддержку сетей в opencv, но не так интересен просто список, сколько визуальное представление.
ps работа не связана с нейронными сетями, просто интерес

Dark_Daiver 25.07.18 в 20:21 0

имхо, это не повод городить плохие «шпаргалки». На крайний случай можно было вставить топологии тех же Inception/ResNet — смотрятся солидно и при этом отображают реальную топологию.

lumaxy 25.07.18 в 18:17 0

Только у меня некоторые картинки слишком мелкие, чтобы прочитать часть текста? «Открыть картинку в новой вкладке» не помогает, надо идти по ссылке в конце статьи.

NIX_Solutions 26.07.18 в 12:42 0

Картинки перезалили.

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

САМОЕ ЧИТАЕМОЕ

Сутки | Неделя | Месяц

Как я переехал в Израиль после блокировки Telegram

+131 54,6k 86 364

От разговорника вместо физики до CarPrice, или как я обосновался в Японии

+54 22,6k 60 82

Сапожники с сапогами: какие смартфоны выбирают для себя эксперты

+17 35k 24 72

Средства разработки для платформы «Байкал-Т1» перешли на российский дистрибутив АЛЬТ

+40 20,1k 21 128

«Ручной» манипулятор

+60 12,6k 86 15