

## APPENDIX A

# Writing Mathematical Expressions with LaTeX

LaTeX is extensively used in Python. In this appendix there are many examples that can be useful to represent LaTeX expressions inside Python implementations. This same information can be found at the link <http://matplotlib.org/users/mathtext.html>.

## With matplotlib

You can enter the LaTeX expression directly as an argument of various functions that can accept it. For example, the `title()` function that draws a chart title.

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.title(r'$\alpha > \beta$')
```

## With IPython Notebook in a Markdown Cell

You can enter the LaTeX expression between two '\$\$'.

```
$$c = \sqrt{a^2 + b^2}$$
```

$$c = \sqrt{a^2 + b^2}$$

# With IPython Notebook in a Python 2 Cell

You can enter the LaTeX expression within the Math() function.

```
from IPython.display import display, Math, Latex
display(Math(r'F(k) = \int_{-\infty}^{\infty} f(x) e^{2\pi i k} dx'))
```

## Subscripts and Superscripts

To make subscripts and superscripts, use the ‘\_’ and ‘^’ symbols:

```
r'\alpha_i > \beta_i'
```

$$\alpha_i > \beta_i$$

This could be very useful when you have to write summations:

```
r'\sum_{i=0}^{\infty} x_i'
```

$$\sum_{i=0}^{\infty} x_i$$

## Fractions, Binomials, and Stacked Numbers

Fractions, binomials, and stacked numbers can be created with the \frac{ }, \binom{ }, and \stackrel{ }{ } commands, respectively:

```
r'\frac{3}{4} \binom{3}{4} \stackrel{3}{4}'
```

$$\frac{3}{4} \binom{3}{4}$$

Fractions can be arbitrarily nested:

$$5 - \frac{1}{\frac{x}{4}}$$

Note that special care needs to be taken to place parentheses and brackets around fractions. You have to insert `\left` and `\right` preceding the bracket in order to inform the parser that those brackets encompass the entire object:

$$\left( \frac{5 - \frac{1}{x}}{4} \right)$$

## Radicals

Radicals can be produced with the `\sqrt{[]}` command.

`r'\sqrt{2}$'`

$$\sqrt{2}$$

## Fonts

The default font is italics for mathematical symbols. To change fonts, for example with trigonometric functions as `sin`:

$$s(t) = A \sin(2\omega t)$$

The choices available with all fonts are

```
from IPython.display import display, Math, Latex
display(Math(r'\mathrm{Roman}'))
display(Math(r'\mathit{Italic}'))
display(Math(r'\mathtt{Typewriter}'))
display(Math(r'\mathcal{CALLIGRAPHY}'))
```

**Roman**

*Italic*

**Typewriter**

*CALLIGRAPHY*

# Accents

An accent command may precede any symbol to add an accent above it. There are long and short forms for some of them.

---

<code>\acute a</code> or <code>\'a</code>	$\acute{a}$
<code>\bar a</code>	$\bar{a}$
<code>\breve a</code>	$\breve{a}$
<code>\ddot a</code> or <code>\"a</code>	$\ddot{a}$
<code>\dot a</code> or <code>\.a</code>	$\dot{a}$
<code>\grave a</code> or <code>\`a</code>	$\grave{a}$
<code>\hat a</code> or <code>\^a</code>	$\hat{a}$
<code>\tilde a</code> or <code>\~a</code>	$\tilde{a}$
<code>\vec a</code>	$\vec{a}$
<code>\overline{abc}</code>	$\overline{abc}$

---

***Symbols***

You can also use a large number of the TeX symbols.

***Lowercase Greek***

---

$\alpha$ <code>\alpha</code>	$\beta$ <code>\beta</code>	$\chi$ <code>\chi</code>	$\delta$ <code>\delta</code>	$F$ <code>\digamma</code>
$\epsilon$ <code>\epsilon</code>	$\eta$ <code>\eta</code>	$\gamma$ <code>\gamma</code>	$\iota$ <code>\iota</code>	$\kappa$ <code>\kappa</code>
$\lambda$ <code>\lambda</code>	$\mu$ <code>\mu</code>	$\nu$ <code>\nu</code>	$\omega$ <code>\omega</code>	$\phi$ <code>\phi</code>
$\pi$ <code>\pi</code>	$\psi$ <code>\psi</code>	$\rho$ <code>\rho</code>	$\sigma$ <code>\sigma</code>	$\tau$ <code>\tau</code>
$\theta$ <code>\theta</code>	$\upsilon$ <code>\upsilon</code>	$\varepsilon$ <code>\varepsilon</code>	$\varkappa$ <code>\varkappa</code>	$\varphi$ <code>\varphi</code>
$\varpi$ <code>\varpi</code>	$\varrho$ <code>\varrho</code>	$\varsigma$ <code>\varsigma</code>	$\vartheta$ <code>\vartheta</code>	$\xi$ <code>\xi</code>
$\zeta$ <code>\zeta</code>				

---

**Uppercase Greek**

---

$\Delta$ \Delta	$\Gamma$ \Gamma	$\Lambda$ \Lambda	$\Omega$ \Omega	$\Phi$ \Phi	$\Pi$ \Pi
$\Psi$ \Psi	$\Sigma$ \Sigma	$\Theta$ \Theta	$\Upsilon$ \Upsilon	$\Xi$ \Xi	$\Upsilon$ \mho
$\nabla$ \nabla					

---

**Hebrew**

---

$\aleph$ \aleph	$\beth$ \beth	$\daleth$ \daleth	$\gimel$ \gimel
-----------------	---------------	-------------------	-----------------

---

**Delimiters**

---

$/$ /	$[$ [	$\Downarrow$ \Downarrow	$\Uparrow$ \Uparrow	$\parallel$ \Vert	$\backslash$ \backslash
$\downarrow$ \downarrow	$\langle$ \langle	$\lceil$ \lceil	$\lfloor$ \lfloor	$\llcorner$ \llcorner	$\lrcorner$ \lrcorner
$\rangle$ \rangle	$\rceil$ \rceil	$\rfloor$ \rfloor	$\ulcorner$ \ulcorner	$\uparrow$ \uparrow	$\urcorner$ \urcorner
$\mid$ \vert	$\{$ \{	$\mid$ \mid	$\}$ \}	$\rfloor$ \rfloor	$\mid$ \mid

---

**Big Symbols**

---

$\bigcap$ \bigcap	$\bigcup$ \bigcup	$\bigodot$ \bigodot	$\bigoplus$ \bigoplus	$\bigotimes$ \bigotimes
$\biguplus$ \biguplus	$\bigvee$ \bigvee	$\bigwedge$ \bigwedge	$\coprod$ \coprod	$\int$ \int
$\oint$ \oint	$\prod$ \prod	$\sum$ \sum		

---

Standard Function Names

$\Pr$ \Pr	$\arccos$ \arccos	$\arcsin$ \arcsin	$\arctan$ \arctan
$\arg$ \arg	$\cos$ \cos	$\cosh$ \cosh	$\cot$ \cot
$\coth$ \coth	$\csc$ \csc	$\deg$ \deg	$\det$ \det
$\dim$ \dim	$\exp$ \exp	$\gcd$ \gcd	$\hom$ \hom
$\inf$ \inf	$\ker$ \ker	$\lg$ \lg	$\lim$ \lim
$\liminf$ \liminf	$\limsup$ \limsup	$\ln$ \ln	$\log$ \log
$\max$ \max	$\min$ \min	$\sec$ \sec	$\sin$ \sin
$\sinh$ \sinh	$\sup$ \sup	$\tan$ \tan	$\tanh$ \tanh

Binary Operation and Relation Symbols

$\bumpeq$ \Bumpeq	$\cap$ \Cap	$\cup$ \Cup
$\doteq$ \Doteq	$\Join$ \Join	$\subseteq$ \Subset
$\supseteq$ \Supset	$\Vdash$ \Vdash	$\Vvdash$ \Vvdash
$\approx$ \approx	$\approxeq$ \approxeq	$\ast$ \ast
$\asymp$ \asymp	$\backepsilon$ \backepsilon	$\backsimeq$ \backsimeq
$\backsimeq$ \backsimeq	$\barwedge$ \barwedge	$\because$ \because
$\between$ \between	$\bigcirc$ \bigcirc	$\bigtriangledown$ \bigtriangledown
$\bigtriangleup$ \bigtriangleup	$\blacktriangleleft$ \blacktriangleleft	$\blacktriangleright$ \blacktriangleright
$\bot$ \bot	$\bowtie$ \bowtie	$\boxdot$ \boxdot
$\boxminus$ \boxminus	$\boxplus$ \boxplus	$\boxtimes$ \boxtimes
$\bullet$ \bullet	$\bumpeq$ \bumpeq	$\cap$ \cap
$\cdot$ \cdot	$\circ$ \circ	$\circ$ \circ
$\coloneq$ \coloneq	$\cong$ \cong	$\cup$ \cup
$\curlyeqprec$ \curlyeqprec	$\curlyeqsucc$ \curlyeqsucc	$\curlyvee$ \curlyvee
$\curlywedge$ \curlywedge	$\dagger$ \dag	$\dashv$ \dashv

(continued)

---

$\ddagger$	<code>\ddag</code>	$\diamond$	<code>\diamond</code>	$\div$	<code>\div</code>
$\div$	<code>\divideontimes</code>	$\doteq$	<code>\doteq</code>	$\doteqdot$	<code>\doteqdot</code>
$\dotplus$	<code>\dotplus</code>	$\overline{\wedge}$	<code>\doublebarwedge</code>	$\eqcirc$	<code>\eqcirc</code>
$\equiv$	<code>\eqcolon</code>	$\approx$	<code>\eqsim</code>	$\gtrsim$	<code>\gtrsim</code>
$\lessapprox$	<code>\lesapprox</code>	$\equiv$	<code>\equiv</code>	$\fallingdotseq$	<code>\fallingdotseq</code>
$\frown$	<code>\frown</code>	$\geq$	<code>\geq</code>	$\geqq$	<code>\geqq</code>
$\gtrless$	<code>\gtrless</code>	$\gg$	<code>\gg</code>	$\ggg$	<code>\ggg</code>
$\gtrapprox$	<code>\gtrapprox</code>	$\gneq$	<code>\gneq</code>	$\gnsim$	<code>\gnsim</code>
$\gtrless$	<code>\gtrless</code>	$\gtrdot$	<code>\gtrdot</code>	$\gtreqless$	<code>\gtreqless</code>
$\in$	<code>\in</code>	$\gtrless$	<code>\gtrless</code>	$\gtrsim$	<code>\gtrsim</code>
$\leq$	<code>\leq</code>	$\intercal$	<code>\intercal</code>	$\leftthreetimes$	<code>\leftthreetimes</code>
$\lessapprox$	<code>\lessapprox</code>	$\leqq$	<code>\leqq</code>	$\leslant$	<code>\leslant</code>
$\lesseqgtr$	<code>\lesseqgtr</code>	$\lessdot$	<code>\lessdot</code>	$\lesseqgtr$	<code>\lesseqgtr</code>
$\ll$	<code>\ll</code>	$\lessgtr$	<code>\lessgtr</code>	$\lesssim$	<code>\lesssim</code>
$\lneqq$	<code>\lneqq</code>	$\lll$	<code>\lll</code>	$\lnapprox$	<code>\lnapprox</code>
$\mid$	<code>\mid</code>	$\lnsim$	<code>\lnsim</code>	$\ltimes$	<code>\ltimes</code>
$\nVDash$	<code>\nVDash</code>	$\models$	<code>\models</code>	$\mp$	<code>\mp</code>
$\ncong$	<code>\ncong</code>	$\nVdash$	<code>\nVdash</code>	$\napprox$	<code>\napprox</code>
$\neq$	<code>\neq</code>	$\neq$	<code>\neq</code>	$\neq$	<code>\neq</code>
$\ngtr$	<code>\ngtr</code>	$\nequiv$	<code>\nequiv</code>	$\ngeq$	<code>\ngeq</code>
$\nless$	<code>\nless</code>	$\ni$	<code>\ni</code>	$\nleq$	<code>\nleq</code>
$\nparallel$	<code>\nparallel</code>	$\nmid$	<code>\nmid</code>	$\notin$	<code>\notin</code>
		$\nprec$	<code>\nprec</code>	$\nsim$	<code>\nsim</code>

---

(continued)

$\not\subset$	$\not\subseteq$	$\nsucc$
$\not\supset$	$\not\supseteq$	$\ntriangleleft$
$\ntrianglelefteq$	$\ntriangleright$	$\ntrianglerighteq$
$\nvDash$	$\nvdash$	$\odot$
$\ominus$	$\oplus$	$\oslash$
$\otimes$	$\parallel$	$\perp$
$\pitchfork$	$\pm$	$\prec$
$\preccurlyeq$	$\preccurlyeq$	$\preceq$
$\precnapprox$	$\precnsim$	$\precapprox$
$\propto$	$\rightthreetimes$	$\risingdotseq$
$\rtimes$	$\sim$	$\simeq$
$/$	$\smile$	$\sqcap$
$\sqcup$	$\sqsubset$	$\sqsubseteq$
$\sqsubseteq$	$\sqsupset$	$\sqsupseteq$
$\sqsupseteq$	$\star$	$\subset$
$\subseteq$	$\subseteqq$	$\subsetneq$
$\subsetneqq$	$\succ$	$\succapprox$
$\succcurlyeq$	$\succeq$	$\succapprox$
$\succnsim$	$\succsim$	$\supset$
$\supseteq$	$\supseteqq$	$\supsetneq$
$\supsetneqq$	$\therefore$	$\times$
$\top$	$\triangleleft$	$\trianglelefteq$
$\trianglelefteq$	$\triangleright$	$\trianglerighteq$
$\uplus$	$\Vdash$	$\varpropto$
$\vartriangleleft$	$\vartriangleright$	$\Vdash$
$\vee$	$\veebar$	$\wedge$
$\wr$		



**Arrow Symbols**


---

$\Downarrow$ <code>\Downarrow</code>	$\Leftarrow$ <code>\Leftarrow</code>
$\Leftrightarrow$ <code>\Leftrightarrow</code>	$\Lleftarrow$ <code>\Lleftarrow</code>
$\Longleftarrow$ <code>\Longleftarrow</code>	$\Longleftrightarrow$ <code>\Longleftrightarrow</code>
$\Rightarrow$ <code>\Rightarrow</code>	$\Uparrow$ <code>\Uparrow</code>
$\nearrow$ <code>\nearrow</code>	$\nwarrow$ <code>\nwarrow</code>
$\Rightarrow$ <code>\Rightarrow</code>	$\Rrightarrow$ <code>\Rrightarrow</code>
$\Rsh$ <code>\Rsh</code>	$\searrow$ <code>\searrow</code>
$\swarrow$ <code>\swarrow</code>	$\Uparrow$ <code>\Uparrow</code>
$\Updownarrow$ <code>\Updownarrow</code>	$\circlearrowleft$ <code>\circlearrowleft</code>
$\circlearrowright$ <code>\circlearrowright</code>	$\curvearrowleft$ <code>\curvearrowleft</code>
$\curvearrowright$ <code>\curvearrowright</code>	$\dashleftarrow$ <code>\dashleftarrow</code>
$\dashrightarrow$ <code>\dashrightarrow</code>	$\downarrow$ <code>\downarrow</code>
$\Downarrow$ <code>\Downarrow</code>	$\Downarrow$ <code>\Downarrow</code>
$\downharpoonright$ <code>\downharpoonright</code>	$\downharpoonleft$ <code>\downharpoonleft</code>
$\hookrightarrow$ <code>\hookrightarrow</code>	$\hookleftarrow$ <code>\hookleftarrow</code>
$\leftarrow$ <code>\leftarrow</code>	$\leadsto$ <code>\leadsto</code>
$\leftharpoonup$ <code>\leftharpoonup</code>	$\leftarrowtail$ <code>\leftarrowtail</code>
$\Lleftarrow$ <code>\Lleftarrow</code>	$\leftharpoonup$ <code>\leftharpoonup</code>
$\Leftrightarrow$ <code>\Leftrightarrow</code>	$\leftrightarrow$ <code>\leftrightarrow</code>
$\Leftrightarrow$ <code>\Leftrightarrow</code>	$\leftrightharpoons$ <code>\leftrightharpoons</code>
$\leftrightsquigarrow$ <code>\leftrightsquigarrow</code>	$\leftsquigarrow$ <code>\leftsquigarrow</code>
$\longleftarrow$ <code>\longleftarrow</code>	$\longleftrightarrow$ <code>\longleftrightarrow</code>
$\longmapsto$ <code>\longmapsto</code>	$\longrightarrow$ <code>\longrightarrow</code>
$\looparrowleft$ <code>\looparrowleft</code>	$\looparrowright$ <code>\looparrowright</code>
$\mapsto$ <code>\mapsto</code>	$\multimap$ <code>\multimap</code>
$\nLeftarrow$ <code>\nLeftarrow</code>	$\nLeftrightarrow$ <code>\nLeftrightarrow</code>
$\nrightarrow$ <code>\nrightarrow</code>	$\nearrow$ <code>\nearrow</code>
$\nleftarrow$ <code>\nleftarrow</code>	$\nleftrightarrow$ <code>\nleftrightarrow</code>

---

*(continued)*

---

$\rightarrow$ <code>\rightarrow</code>	$\nwarrow$ <code>\nwarrow</code>
$\rightarrowtail$ <code>\rightarrowtail</code>	$\rightarrowtail$ <code>\rightarrowtail</code>
$\rightharpoonup$ <code>\rightharpoonup</code>	$\rightharpoonup$ <code>\rightharpoonup</code>
$\rightleftarrows$ <code>\rightleftarrows</code>	$\rightleftarrows$ <code>\rightleftarrows</code>
$\rightleftharpoons$ <code>\rightleftharpoons</code>	$\rightleftharpoons$ <code>\rightleftharpoons</code>
$\rightrightarrows$ <code>\rightrightarrows</code>	$\rightrightarrows$ <code>\rightrightarrows</code>
$\rightsquigarrow$ <code>\rightsquigarrow</code>	$\searrow$ <code>\searrow</code>
$\swarrow$ <code>\swarrow</code>	$\rightarrow$ <code>\to</code>
$\twoheadleftarrow$ <code>\twoheadleftarrow</code>	$\twoheadrightarrow$ <code>\twoheadrightarrow</code>
$\uparrow$ <code>\uparrow</code>	$\updownarrow$ <code>\updownarrow</code>
$\updownarrow$ <code>\updownarrow</code>	$\Uparrow$ <code>\Uparrow</code>
$\Uparrow$ <code>\Uparrow</code>	$\Uparrow$ <code>\uparpoonleft</code>
	$\Uparrow$ <code>\uparrows</code>

---

*Miscellaneous Symbols*

---

$\$$ <code>\\$</code>	$\text{\AA}$ <code>\AA</code>	$\Finv$ <code>\Finv</code>
$\Game$ <code>\Game</code>	$\Im$ <code>\Im</code>	$\P$ <code>\P</code>
$\Re$ <code>\Re</code>	$\S$ <code>\S</code>	$\angle$ <code>\angle</code>
$\backprime$ <code>\backprime</code>	$\bigstar$ <code>\bigstar</code>	$\blacksquare$ <code>\blacksquare</code>
$\blacktriangle$ <code>\blacktriangle</code>	$\blacktriangledown$ <code>\blacktriangledown</code>	$\cdots$ <code>\cdots</code>
$\checkmark$ <code>\checkmark</code>	$\text{\textcircled{R}}$ <code>\circledR</code>	$\text{\textcircled{S}}$ <code>\circledS</code>
$\clubsuit$ <code>\clubsuit</code>	$\complement$ <code>\complement</code>	$\copyright$ <code>\copyright</code>
$\ddots$ <code>\ddots</code>	$\diamondsuit$ <code>\diamondsuit</code>	$\ell$ <code>\ell</code>
$\emptyset$ <code>\emptyset</code>	$\eth$ <code>\eth</code>	$\exists$ <code>\exists</code>
$\flat$ <code>\flat</code>	$\forall$ <code>\forall</code>	$\hbar$ <code>\hbar</code>
$\heartsuit$ <code>\heartsuit</code>	$\hslash$ <code>\hslash</code>	$\iiint$ <code>\iiint</code>

---

(continued)

---

$\int\!\!\int$ <code>\iint</code>	$\int\!\!\int$ <code>\iint</code>	$\mathfrak{z}$ <code>\imath</code>
$\infty$ <code>\infty</code>	$\mathcal{J}$ <code>\jmath</code>	$\dots$ <code>\ldots</code>
$\measuredangle$ <code>\measuredangle</code>	$\natural$ <code>\natural</code>	$\neg$ <code>\neg</code>
$\nexists$ <code>\nexists</code>	$\iiint$ <code>\iiint</code>	$\partial$ <code>\partial</code>
$\prime$ <code>\prime</code>	$\sharp$ <code>\sharp</code>	$\spadesuit$ <code>\spadesuit</code>
$\sphericalangle$ <code>\sphericalangle</code>	$\beta$ <code>\ss</code>	$\nabla$ <code>\triangledown</code>
$\varnothing$ <code>\varnothing</code>	$\triangle$ <code>\vartriangle</code>	$\vdots$ <code>\vdots</code>
$\wp$ <code>\wp</code>	$\yen$ <code>\yen</code>	

---

## APPENDIX B

# Open Data Sources

## Political and Government Data

### **Data.gov**

<http://data.gov>

This is the resource for most government-related data.

### **Socrata**

<http://www.socrata.com/resources/>

Socrata is a good place to explore government-related data. Furthermore, it provides some visualization tools for exploring data.

### **US Census Bureau**

<http://www.census.gov/data.html>

This site provides information about US citizens covering population data, geographic data, and education.

### **UN3ta**

<https://data.un.org/>

UNdata is an Internet-based data service which brings UN statistical databases.

### **European Union Open Data Portal**

<http://open-data.europa.eu/en/data/>

This site provides a lot of data from European Union institutions.

### **Data.gov.uk**

<http://data.gov.uk/>

This site of the UK Government includes the British National Bibliography: metadata on all UK books and publications since 1950.

### **The CIA World Factbook**

<https://www.cia.gov/library/publications/the-world-factbook/>

This site of the Central Intelligence Agency provides a lot of information on history, population, economy, government, infrastructure, and military of 267 countries.

## **Health Data**

### **Healthdata.gov**

<https://www.healthdata.gov/>

This site provides medical data about epidemiology and population statistics.

### **NHS Health and Social Care Information Centre**

<http://www.hscic.gov.uk/home>

Health data sets from the UK National Health Service.

## **Social Data**

### **Facebook Graph**

<https://developers.facebook.com/docs/graph-api>

Facebook provides this API which allows you to query the huge amount of information that users are sharing with the world.

### **Topsy**

<http://topsy.com/>

Topsy provides a searchable database of public tweets going back to 2006 as well as several tools to analyze the conversations.

**Google Trends**

<http://www.google.com/trends/explore>

Statistics on search volume (as a proportion of total search) for any given term, since 2004.

**Likebutton**

<http://likebutton.com/>

Mines Facebook’s public data—globally and from your own network—to give an overview of what people “Like” at the moment.

## Miscellaneous and Public Data Sets

**Amazon Web Services public datasets**

<http://aws.amazon.com/datasets>

The public data sets on Amazon Web Services provide a centralized repository of public data sets. An interesting dataset is the 1000 Genome Project, an attempt to build the most comprehensive database of human genetic information. Also a NASA database of satellite imagery of Earth is available.

**DBPedia**

<http://wiki.dbpedia.org>

Wikipedia contains millions of pieces of data, structured and unstructured, on every subject. DBPedia is an ambitious project to catalogue and create a public, freely distributable database allowing anyone to analyze this data.

**Freebase**

<http://www.freebase.com/>

This community database provides information about several topics, with over 45 million entries.

## **Gapminder**

<http://www.gapminder.org/data/>

This site provides data coming from the World Health Organization and World Bank covering economic, medical, and social statistics from around the world.

# **Financial Data**

## **Google Finance**

<https://www.google.com/finance>

Forty years' worth of stock market data, updated in real time.

# **Climatic Data**

## **National Climatic Data Center**

<http://www.ncdc.noaa.gov/data-access/quick-links#loc-clim>

Huge collection of environmental, meteorological, and climate data sets from the US National Climatic Data Center. The world's largest archive of weather data.

## **WeatherBase**

<http://www.weatherbase.com/>

This site provides climate averages, forecasts, and current conditions for over 40,000 cities worldwide.

## **Wunderground**

<http://www.wunderground.com/>

This site provides climatic data from satellites and weather stations, allowing you to get all information about the temperature, wind, and other climatic measurements.

## Sports Data

### Pro-Football-Reference

<http://www.pro-football-reference.com/>

This site provides data about football and several other sports.

## Publications, Newspapers, and Books

### New York Times

<http://developer.nytimes.com/docs>

Searchable, indexed archive of news articles going back to 1851.

### Google Books Ngrams

<http://storage.googleapis.com/books/ngrams/books/datasetv2.html>

This source searches and analyzes the full text of any of the millions of books digitized as part of the Google Books project.

## Musical Data

### Million Song Data Set

<http://aws.amazon.com/datasets/6468931156960467>

Metadata on over a million songs and pieces of music. Part of Amazon Web Services.



# Index

## A

- Accents, LaTeX, 540–547
- Advanced Data aggregation
  - apply() functions, 225
  - transform() function, 226
- Anaconda, 24, 88
- Anderson Iris Dataset, *see* Iris flower dataset
- Array manipulation
  - joining arrays
    - column\_stack() and row\_stack(), 72
    - hstack() function, 71
    - vstack() function, 71
  - splitting arrays
    - hsplit() function, 72
    - split() function, 73–74
    - vsplit() function, 72
- Artificial intelligence, 5, 350
  - schematization of, 352
- Artificial neural networks
  - biological networks, 361
  - edges, 356
  - hidden layer, 357
  - input and output layer, 357
  - multi layer perceptron, 360
  - nodes, 356
  - schematization of, 355–356
  - SLP (*see* Single layer perceptron (SLP))
  - weight, 356

## B

- Bar chart
  - 3D, 306–307
  - error bars, 281
  - horizontal, 281–282
  - matplotlib, 278
  - multiserial, 282–284
  - multiseries stacked bar, 286–290
  - pandas DataFrame, 285–286
  - representations, 291
  - stacked bar charts, 290
  - x-axis, 280
  - xticks() function, 279
- Bayesian methods, 4
- Big Data, 353
- Bigrams, 498
- Biological neural networks, 361
- Blending operation, 520

## C

- Caffe2, 355
- Chart typology, 267
- Choropleth maps
  - D3 library, 459
  - geographical representations, 459
  - HTML() function, 461–462
  - jinja2, 462–463
  - JSON and TSV, 463

Choropleth maps (*cont.*)

- JSON TopoJSON, 460
- require.config(), 461
- results, 464
- US population, 2014
  - data source census.gov, 467
  - file TSV, codes, 466
  - HTML() function, 468
  - jinja2.Template, 469
  - pop2014\_by\_county
    - dataframe, 465–466
  - population.csv, 467–468
  - render() function, 470–471
  - SUMLEV values, 464

Classification and regression trees, 12

Classification models, 12

Climatic data, 552

Clustered bar chart

- IPython Notebook, 454–455
- jinja2, 455, 457–458
- render() function, 458–459

Clustering models, 4, 11–12

Collocations, 498

Computer vision, 507

Concatenation

- arrays, 188
- combining, 191, 193
- concat() function, 189–190
- dataframe, 191
- keys option, 190
- pivoting, 193
  - hierarchical indexing, 193
  - long to wide format, 195
  - stack() function, 194
  - unstack() function, 194
- removing, 196

Correlation, 129–131

Covariance, 129–131

Cross-validation, 13

Cython, 22

## D

Data aggregation

- apply() functions, 226, 228–229
- GroupBy, 217
  - groupby() function, 219
  - operations, 218
  - output of, 220
  - SPLIT-APPLY-COMBINE, 218
- hierarchical grouping, 220–221
- merge(), 226
- numeric and string values, 219
- price1 column, 219
- transform() function, 225

Data analysis

- charts, 2
- data visualization, 2
- definition, 1
- deployment phase, 2
- information, 6
- knowledge, 6
- knowledge domains
  - computer science, 3
  - disciplines, 3
  - fields of application, 5
  - machine learning and artificial intelligence, 5
  - mathematics and statistics, 4
  - problems of, 3
- open data, 15–16
- predictive model, 2
- process
  - data sources, 9
  - deployment, 13
  - exploration/visualization, 10–11

- extraction, 9–10
  - model validation, 13
  - planning phase, 9
  - predictive modeling, 12
  - preparation, 10
  - problem definition, 8
  - stages, 6–8
- purpose of, 1
- Python and, 17
- quantitative and qualitative, 14
- types
  - categorical data, 6
  - numerical data, 6
- DataFrame
  - pandas
    - definition, 103–105
    - nested dict, 111
    - operations, 121
    - structure, 103
    - transposition, 111
  - structure, 102
- Data manipulation
  - aggregation (*see* Data aggregation)
  - concatenation, 188
  - discretization and binning, 204
  - group iteration, 222
  - permutation, 210
  - phases of, 181
  - preparation (*see* Data preparation)
  - string (*see* String manipulation)
  - transformation, 197
- Data preparation, 181
  - DataFrame, 182
  - merging operation, 182
  - pandas.concat(), 182
  - pandas.DataFrame.combine\_
    - first(), 182
  - pandas.merge(), 182
  - procedures of, 181
- Data structures, operations
  - DataFrame and series, 121–122
  - flexible arithmetic
    - methods, 120–121
- Data transformation
  - drop\_duplicates() function, 199
  - mapping
    - adding values, 201
    - axes, 202
    - dict objects, 199
    - replacing values, 199
  - remove duplicates, 198–199
- Data visualization
  - adding text
    - axis labels, 251–252
    - informative label, 254
    - mathematical expression, 254–255
    - modified of, 252
    - text() function, 253
  - bar chart (*see* Bar chart)
  - chart typology, 267
  - contour plot/map, 297–299
  - data analysis, 231
  - 3D surfaces, 302, 304
  - grid, 256
  - grids, subplots, 309
  - handling date values, 264–267
  - histogram, 277–278
  - installation, 233
  - IPython and IPython
    - QtConsole, 233, 235
  - kwargs
    - figures and axes, 249
    - horizontal subplots, 249–250
    - linewidth, 248

Data visualization (*cont.*)

- plot() function, 249
- vertical subplots, 250–251

legend

- chart of, 258
- legend() function, 257, 258
- multiseries chart, 259
- upper-right corner, 257

line chart (*see* Line chart)

- matplotlib architecture and NumPy, 247

- matplotlib library (*see* matplotlib library)

- mplot3d, 302

multi-panel plots

- grids, subplots, 309, 311
- subplots, 307–309

pie charts, 292

- axis() function, 293
- modified chart, 294
- pandas DataFrame, 296
- pie() function, 292
- shadow kwarg, 295

plotting window

- buttons of, 241
- commands, 241
- matplotlib and NumPy, 246
- plt.plot() function, 242, 243
- properties, 243
- QtConsole, 241–242

- polar chart, 299, 301

- pyplot module, 239

saving, charts

- HTML file, 262–263
- image file, 264
- source code, 260–261
- scatter plot, 3D, 304–305

Decision trees, 11

Deep learning, 349, 532

- artificial (*see* Artificial neural networks)

- artificial intelligence, 350

- data availability, 353

- machine learning, 351

- neural networks and GPUs, 352

Python

- frameworks, 354

- programming language, 354

- schematization of, 352

- TensorFlow (*see* TensorFlow)

Digits dataset

- definition, 475

- digits.images array, 477

- digit.targets array, 478

- handwritten digits, 477

- handwritten number images, 475

- matplotlib library, 477

- scikit-learn library, 476

Discretization and binning, 204

- any() function, 210

- categorical type, 206

- cut() function, 205–206, 208–209

- describe() function, 209

- detecting and filtering

- outliers, 209

- qcut(), 208–209

- std() function, 210

- value\_counts() function, 206

Django, 17

Dropping, 117–118

## E

Eclipse (pyDev), 41–42

Element-wise computation, 47

Expression-oriented programming, 33

**F**

Financial data, 552  
 Flexible arithmetic methods, 120–121  
 Fonts, LaTeX, 539

**G**

Gradient theory, 523  
 Graphics Processing Unit (GPU), 353  
 Grouping, 11  
 Group iteration  
   chain of transformations, 222, 224  
   functions on groups  
     mark() function, 224–225  
     quantiles() function, 224  
 GroupBy object, 222

**H**

Handwriting recognition  
   digits dataset, 475–478  
   handwritten digits, matplotlib  
     library, 478  
   learning and predicting, 478, 480, 482  
   OCR software, 473  
   scikit-learn, 474–475  
   svc estimator, 480  
   TensorFlow, 480  
   validation set, six digits, 479  
 Health data, 550  
 Hierarchical indexing  
   arrays, 136–137  
   DataFrame, 135  
   reordering and sorting levels, 137–138  
   stack() function, 136  
   statistic levels, 138  
   structure, 134  
   two-dimensional structure, 134

**I**

IDEs, *see* Interactive development environments (IDEs)  
 Image analysis  
   concept of, 521  
   convolutions, 523  
   definition, 507  
   edge detection, 522, 525  
     blackandwhite.jpg  
       image, 526–529, 531  
     black and white system, 525  
     filters function, 528  
     gradients.jpg image, 532  
     gray gradients, 525  
     Laplacian and Sobel filters, 531  
     results, 528  
     source code, 530  
   face detection, 532  
   gradient theory, 523  
   OpenCV (*see* Open Source Computer Vision (OpenCV))  
   operations, 508  
   representation of, 522  
 Indexing functionalities  
   arithmetic and data  
     alignment, 118, 120  
   dropping, 117–118  
   reindexing, 114, 116  
 Integration, 47  
 Interactive development environments (IDEs)  
   Eclipse (pyDev), 41–42  
   Komodo, 45  
   Liclipse, 43–46  
   NinjaIDE, 44–45  
   Spyder, 41  
   Sublime, 42–43

## INDEX

- Interactive programming language, 20
- Interfaced programming language, 20
- Internet of Things (IoT), 353
- Interpreted programming language, 20
- Interpreter
  - characterization, 21
  - Cython, 22
  - Jython, 22
  - PVM, 21
  - PyPy, 22
  - tokenization, 21

- IPython
  - and IPython QtConsole, 233–234
  - Jupyter project logo, 37
  - Notebook, 39, 474
    - DataFrames, 420
  - QtConsole, 38
  - shell, 36
  - tools of, 35

- Iris flower dataset
  - Anderson Iris Dataset, 316
  - IPython QtConsole, 316
  - Iris setosa features, 318–319
  - length and width,
    - petal, 319–320
  - matplotlib library, 318
  - PCA decomposition, 320
  - target attribute, 317
  - types of analysis, 316
  - variables, 319–320

## J

- JavaScript D3 Library
  - bar chart, 454
  - CSS definitions, 450–451
  - data-driven documents, 449
  - HTML importing library, 450

- IPython Notebooks, 449
- Jinja2 library, 451–453
  - pandas dataframe, 453
  - render() function, 453
  - require.config() method, 450
  - web chart creation, 450

- Jinja2 library, 451–453
- Jython, 22

## K

- K-nearest neighbors classification
  - decision boundaries, 325–326
  - 2D scatterplot, sepals, 324
  - predict() function, 323
  - random.permutation(), 323
  - training and testing set, 322

## L

- LaTeX
  - accents, 540–547
  - fonts, 539
  - fractions, binomials, and stacked numbers, 538–539
  - with IPython Notebook
    - in Markdown Cell, 537
    - in Python 2 Cell, 538
  - with matplotlib, 537
  - radicals, 539
  - subscripts and superscripts, 538
  - symbols
    - arrow symbols, 540, 545–546
    - big symbols, 542
    - binary operation and relation symbols, 542–543
    - Delimiters, 540–541
    - Hebrew, 541

- lowercase Greek, 540
- miscellaneous symbols, 540
- standard function names, 542
- uppercase Greek, 541
- Learning phase, 378
- Liclipse, 43–46
- Linear regression, 12
- Line chart
  - annotate(), 274
  - arrowprops kwarg, 274
  - Cartesian axes, 273
  - color codes, 270–271
  - data points, 267
  - different series, 269
  - gca() function, 273
  - Greek characters, 272
  - LaTeX expression, 274
  - line and color styles, 270
  - mathematical expressions, 275
  - mathematical function, 268
  - pandas, 276
  - plot() function, 268
  - set\_position() function, 273
  - xticks() and yticks() functions, 271
- Linux distribution, 90
- LOD cloud diagram, 16
- Logistic regression, 12

## M

- Machine learning (ML), 5
  - algorithm development process, 313
  - deep learning, 351
  - diabetes dataset, 327–328
  - features/attributes, 314
  - Iris flower dataset, 316
  - learning problem, 314
  - linear/least square regression
    - coef\_ attribute, 329
    - fit() function, 329
    - linear correlation, 330
    - parameters, 328
    - physiological factors and
      - progression of diabetes, 332–333
      - single physiological factor, 330
    - schematization of, 352
    - supervised learning, 314
    - SVM (*see* Support vector machines (SVMs))
    - training and testing set, 315
    - unsupervised learning, 314–315
- Mapping
  - adding values, 201–202
  - inplace option, 204
  - rename() function, 204
  - renaming, axes, 202, 204
  - replacing values, 199, 201
- Mathematical expressions with LaTeX,
  - see* LaTeX
- MATLAB, 17
- matplotlib, 48
- matplotlib library
  - architecture
    - artist layer, 236–238
    - backend layer, 236
    - functions and tools, 235
    - layers, 235
    - pylab and pyplot, 238–239
    - scripting layer (pyplot), 238
  - artist layer
    - graphical representation, 237
    - hierarchical structure, 236
    - primitive and composite, 237
  - graphical representation, 231, 233
  - LaTeX, 232
  - NumPy, 246

## INDEX

Matrix product, 60

Merging operation

- DataFrame, 183–184

- dataframe objects, 183

- index, 187

- join() function, 187–188

- JOIN operation, 182

- left\_index/right\_index

  - options, 187

- left join, right join and

  - outer join, 186

- left\_on and right\_on, 185, 187

- merge() function, 183, 184

Meteorological data, 409

- Adriatic Sea and Po Valley, 410

  - cities, 412

  - Comacchio, 413

  - image of, 411

  - mountainous areas, 410

  - reference standards, 412

  - TheTimeNow website, 413

- climate, 409

- data source

  - JSON file, 414

  - Weather Map site, 414

- IPython Notebook

  - chart representation, 425, 429, 431

  - CSV files, 421

  - DataFrames, 422, 432

  - humidity function, 433–435

  - linear regression, 431

  - matplotlib library, 423

  - Milan, 423

  - read\_csv() function, 421

  - result, 423

  - shape() function, 422

  - SVR method, 428–429

  - temperature, 424, 426–427, 432

- Jupyter Notebook, 415

  - access internal data, 417

  - command line, 415

  - dataframe, 419–420

  - extraction procedures, 418

  - Ferrara, 416

  - JSON file, 416

  - json.load() function, 415

  - parameters, 419

  - prepare() function, 420

- RoseWind (*see* RoseWind)

- wind speed, 441

Microsoft excel files

- dataframe, 162

- data.xls, 160, 162

- internal module xlrd, 160

- read\_excel() function, 161

MongoDB, 178–179

Multi Layer Perceptron (MLP)

- artificial networks, 360

- evaluation of, 404

- experimental data, 404

- hidden layers, 397

- IPython session, 387

- learning phase, 389

- model definition, 387

- test phase and accuracy

  - calculation, 395, 402

Musical data, 553

## N

Natural Language Toolkit (NLTK)

- bigrams and collocations, 498

- common\_contexts() function, 493

- concordance() function, 493

- corpora, 488

- downloader tool, 489



- fileids() function, 491
  - HTML pages, text, 501
  - len() function, 491
  - library, 489
  - macbeth variable, 491
  - Python library, 488
  - request() function, 502
  - selecting words, 497
  - sentimental analysis, 502
  - sents() function, 492
  - similar() function, 494
  - text, network, 500
  - word frequency, 494
    - macbeth variable, 495
    - most\_common() function, 494
    - nltk.download() function, 495
    - nltk.FreqDist() function, 494
    - stopwords, 495
    - string() function, 496
  - word search, 493
  - Ndarray, 47
    - array() function, 51–53
    - data, types, 53–54
    - dtype (data-type), 50, 54
    - intrinsic creation, 55–57
    - type() function, 51–52
  - NOSE MODULE, 91
  - “Not a Number” data
    - filling, NaN occurrences, 133
    - filtering out NaN
      - values, 132–133
    - NaN value, 131–132
  - NumPy library
    - array manipulation (*see* Array manipulation)
    - basic operations
      - aggregate functions, 62
      - arithmetic operators, 57–59
      - increment and decrement operators, 60–61
      - matrix product, 59–60
      - ufunc, 61
    - broadcasting
      - compatibility, 77
      - complex cases, 78–79
      - operator/function, 76
    - BSD, 50
    - conditions and Boolean arrays, 69
    - copies/views of objects, 75
    - data analysis, 49
    - indexing, 63
      - bidimensional array, 64
      - monodimensional ndarray, 63
      - negative index value, 63
    - installation, 50
    - iterating an array, 67–69
    - ndarray (*see* Ndarray)
    - Numarray, 49
    - python language, 49
    - reading and writing array data, 82
    - shape manipulation, 70–71
    - slicing, 65–66
    - structured arrays, 79
    - vectorization, 76
- O**
- Object-oriented programming language, 20
  - OCR, *see* Optical Character Recognition (OCR) software
  - Open data, 15–16
  - Open data sources, 353
    - climatic data, 552
    - demographics
      - IPython Notebook, 446
      - matplotlib, 449

## INDEX

### Open data sources (*cont.*)

- pandas dataframes, 446–447
- pop2014\_by\_state dataframe, 448
- pop2014 dataframe, 447–448
- United States Census Bureau, 445–446

financial data, 552

health data, 550

miscellaneous and public

data sets, 551–552

musical data, 553

political and government

data, 549–550

publications, newspapers,

and books, 553

social data, 550–551

sports data, 553

### Open Source Computer Vision (OpenCV)

deep learning, 509

image processing and analysis, 509

add() function, 515

blackish image, 518

blending, 520

destroyWindow() method, 512

elementary operations, 514

imread() method, 510

imshow() method, 511

load and display, 510

merge() method, 513

NumPy matrices, 519

saving option, 514

waitKey() method, 511

working process, 512

installation, 509

MATLAB packages, 508

start programming, 510

### Open-source programming

language, 21

### Optical Character Recognition (OCR)

software, 473

order() function, 127

## P

Pandas dataframes, 446, 453

Pandas data structures

DataFrame, 102–105

assigning values, 107–109

deleting column, 110

element selection, 105–107

filtering, 110

membership value, 109–110

nested dict, 111

transposition, 111

evaluating values, 98–99

index objects, 112

duplicate labels, 112–113

methods, 112

NaN values, 99, 101

NumPy arrays and existing

series, 96–97

operations, 120–122

operations and mathematical

functions, 97–98

series, 93

assigning values, 95

declaration, 94

dictionaries, 101

filtering values, 97

index, 93

internal elements, selection, 95

operations, 102

Pandas library, 87

correlation and covariance, 129–131

data structures (*see* Pandas data structures)

- function application and mapping
  - element, 123
  - row/column, 123, 125
  - statistics, 125
- getting started, 92
- hierarchical indexing and
  - leveling, 134–135, 137–138
- indexes (*see* Indexing functionalities)
- installation
  - Anaconda, 88–89
  - development phases, 91
  - Linux, 90
  - module repository, Windows, 90
  - PyPI, 89
  - source, 90
  - testing, 91
- “Not a Number” data, 131–134
- python data analysis, 87–88
- sorting and ranking, 126–129
- Permutation
  - new\_order array, 211
  - np.random.randint() function, 211
  - numpy.random.permutation()
    - function, 210
  - random sampling, 211
    - DataFrame, 211
  - take() function, 211
- Pickle—python object serialization
  - cPickle, 168
  - frame.pkl, 170
  - pandas library, 169
  - stream of bytes, 168
- Political and government
  - data, 549–550
- pop2014\_by\_county dataframe, 465
- pop2014\_by\_state dataframe, 448–449
- pop2014 dataframe, 447–448
- Portable programming language, 20
- PostgreSQL, 174
- Principal component analysis
  - (PCA), 320, 322
- Public data sets, 551–552
- PVM, *see* Python virtual machine (PVM)
- pyplot module
  - interactive chart, 239
  - Line2D object, 240
  - plotting window, 240
- show() function, 240
- PyPy interpreter, 22
- Python, 17
  - data analysis library, 87–88
  - deep learning, 354
  - frameworks, 354
  - module, 91
  - OpenCV, 508
- Python Package Index (PyPI), 39, 89
- Python’s world
  - code implementation, 28
  - distributions, 24
    - Anaconda, 24
    - Enthought Canopy, 26
    - Python(x,y), 26
  - IDEs (*see* Interactive development environments (IDEs))
  - installation, 23–24
  - interact, 28
  - interpreter (*see* Interpreter)
  - IPython (*see* IPython)
  - programming language, 19–21
  - PyPI, 39
  - Python 2, 23
  - Python 3, 23
  - running, entire program code, 27
  - SciPy
    - libraries, 46
    - matplotlib, 48

## INDEX

### Python's world (*cont.*)

- NumPy, 47
- pandas, 47
- shell, 26
- source code
  - data structure, 30
  - dictionaries and lists, 31
  - functional programming, 33
  - Hello World, 28
  - index, 32
  - libraries and functions, 30
  - map() function, 33
  - mathematical operations, 29
  - print() function, 29
- writing python code,
  - indentation, 34–35
- Python virtual machine (PVM), 21
- PyTorch, 355

## Q

- Qualitative analysis, 14
- Quantitative analysis, 14

## R

- R, 17
- Radial Basis Function (RBF), 340
- Radicals, LaTeX, 539
- Ranking, 128–129
- Reading and writing array
  - binary files, 82
  - tabular data, 83–84
- Reading and writing data
  - CSV and textual files
    - header option, 144
    - index\_col option, 145
    - myCSV\_01.csv, 143

- myCSV\_03.csv, 145
- names option, 145
- read\_csv() function, 143, 145
- read\_table() function, 143
- .txt extension, 142

### databases

- create\_engine() function, 171
- dataframe, 175
- pandas.io.sql module, 171
- pgAdmin III, 175–176
- PostgreSQL, 174
- read\_sql() function, 172
- read\_sql\_query() function, 177
- read\_sql\_table() function, 177
- sqlalchemy, 171
- sqlite3, 171

### DataFrame objects, 141

### functionalities, 141

### HDF5 library

- data structures, 167
- HDFStore, 167
- hierarchical data format, 166
- mydata.h5, 167

### HTML files

- data structures, 152
- read\_html (), 155
- web\_frames, 156
- web pages, 152
- web scraping, 152

### I/O API Tools, 141–142

### JSON data

- books.json, 164
- frame.json, 164
- json\_normalize() function, 165
- JSONViewer, 162–163
- normalization, 164
- read\_json() and to\_json(), 162
- read\_json() function, 164–165

- Microsoft excel files, 159
  - NoSQL database
    - insert() function, 179
    - MongoDB, 178–180
  - pickle—python object serialization, 168
  - RegExp
    - metacharacters, 146
    - read\_table(), 146
    - skiprows, 148
  - TXT files, 147–148
    - nrows and skiprows options, 149
    - portion by portion, 149
    - writing (*see* Writing data)
    - XML (*see* XML)
  - Regression models, 4, 12
  - Reindexing, 114–116
  - RoseWind
    - DataFrame, 436–437
    - hist array, 438–439
    - polar chart, 438, 440–441
    - scatter plot representation, 438
    - showRoseWind() function, 439, 441
- S**
- Scikit-learn library, 473
    - data analysis, 474
    - k-nearest neighbors classification, 322
    - PCA, 320
    - Python module, 313
    - sklearn.svm.SVC, 475
    - supervised learning, 315
    - svm module, 475
  - SciPy
    - libraries, 46
    - matplotlib, 48
    - NumPy, 47
    - pandas, 47
  - Sentimental analysis, 502
    - document\_features()
      - function, 504
    - documents, 503
    - list() function, 504
    - movie\_reviews, 503
    - negative/positive opinion, 505
    - opinion mining, 502
  - Shape manipulation
    - reshape() function, 70
    - shape attribute, 70
    - transpose() function, 71
  - Single layer perceptron (SLP), 371
    - accuracy, 359
    - activation function, 358, 359
    - architecture, 357
    - cost optimization, 382
    - data analysis, 372
    - evaluation phase, 359
    - learning phase, 359, 378
    - model definition, 374
      - explicitly, 376
      - implicitly, 376
      - learning phase, 375
      - placeholders, 376
      - tf.add() function, 377
      - tf.nn.softmax() function, 377
    - modules, 372
    - representation, 358
    - testing set, 385
    - test phase and accuracy
      - calculation, 383
    - training sets, 359
  - Social data, 550
  - sort\_index() function, 126–128, 138
  - Sports data, 553
  - SQLite3, 171
  - stack() function, 136

## INDEX

### String manipulation

#### built-in methods

- count() function, 214
- error message, 214
- index() and find(), 213
- join() function, 213
- replace() function, 214
- split() function, 212
- strip() function, 212

#### regular expressions

- findall() function, 215–216
- match() function, 216
- re.compile() function, 215
- regex, 214
- re.split() function, 215
- split() function, 215

### Structured arrays

- dtype option, 79, 81
- structs/records, 79

### Subjective interpretations, 14

### Subscripts and superscripts,

- LaTeX, 538

### Supervised learning

- machine learning, 314
- scikit-learn, 315

### Support vector classification (SVC), 475

- decision area, 336
- effect, decision boundary, 338–339
- nonlinear, 339–341
- number of points, C parameter, 337–338
- predict() function, 336–337
- regularization, 337
- support\_vectors array, 337
- training set, decision space, 334–336

### Support vector machines (SVMs)

- decisional space, 334
- decision boundary, 334
- Iris Dataset

#### decision boundaries, 342

- linear decision boundaries, 342–343
- polynomial decision boundaries, 344
- polynomial kernel, 343–344
- RBF kernel, 344
- training set, 342

### SVC (*see* Support vector classification (SVC))

### SVR (*see* Support vector regression (SVR))

### Support vector regression (SVR)

- curves, 347
- diabetes dataset, 345
- linear predictive model, 345
- test set, data, 345
- swaplevel() function, 137

## T

### TensorFlow, 349, 354, 362, 480

- data flow graph, 362
- Google's framework, 362
- installation, 363
- IPython QtConsole, 364
- MLP (*see* Multi Layer Perceptron (MLP))
- model and sessions, 364
- SLP (*see* Single layer perceptron (SLP))
- tensors
  - operation, 370
  - parameters, 366
  - print() function, 368
  - representations of, 367
  - tf.convert\_to\_tensor() function, 368
  - tf.ones() method, 369
  - tf.random\_normal() function, 369
  - tf.random\_uniform() function, 369
  - tf.zeros() method, 368

Text analysis techniques  
     definition, 487  
     NLTK (*see* Natural Language Toolkit  
         (NLTK))  
     techniques, 488  
 Theano, 355  
 trigrams() function, 499

## U, V

United States Census Bureau, 445–446  
 Universal functions (ufunc), 61  
 Unsupervised learning, 314

## W

Web Scraping, 4, 10  
 Wind speed  
     polar chart representation, 442  
     RoseWind\_Speed() function, 441  
     ShowRoseWind() function, 442

    ShowRoseWind\_Speed() function, 442  
     to\_csv() function, 443  
 Writing data  
     HTML files  
         myFrame.html, 155  
         to\_html() function, 153–154  
     na\_rep option, 151  
     to\_csv() function, 150–151

## X, Y, Z

XML  
     books.xml, 157–158  
     getchildren(), 158  
     getroot() function, 158  
     lxml.etree tree structure, 159  
     lxml library, 157  
     objectify, 158  
     parse() function, 158  
     tag attribute, 158  
     text attribute, 159