

plotly

The Best Python Tools for Data Science

The Best Python Tools for Data Science

As data scientists take on new types of projects and aim to solve more complex problems, having a roster of Python tools is essential for meeting today's challenges and innovating for the future. Python libraries and frameworks give data scientists a foundation on which they can create data analysis workflows, build apps, and develop machine learning (ML) systems that support enterprise data strategies at scale.

What Python skills are needed for data science?

Data scientists spend most of their time analyzing and understanding data and building ML prototypes to explore how they can be used in their companies. Because Python is such a flexible coding language, having a diverse skill set when it comes to using it enables data scientists to tackle challenges at the intersection of big data and AI.

One of the main ways you can use Python is to build data visualizations and run analytics. Developing AI and ML algorithms to assist in the data analysis process is also critical as businesses start to scale their analytics and rely on different types of data from many different sources. Data scientists need to be able to use Python in a variety of ways to stay agile as these shifts happen.

For analytics, data scientists should know how to use Python to collect, manipulate, and organize information. They also need to be able to communicate that information in a meaningful way. This is where using Python to build data visualizations like graphs is essential. Data scientists must understand how to pull different data sources together and surface insights that decision-makers and executives can use to formulate business strategy.

Which Python framework is used in data science?

There are several frameworks and libraries you can use for different tasks and many of them can be used together to help you get the job done.

NumPy is one of the most-used open-source Python libraries and is considered the universal standard for working with numerical data in Python. Pandas is another widely-used library that provides data structures and analysis tools. When it comes to ML frameworks, Kaggle found that 80% of data scientists use scikit-learn. They also found that TensorFlow and Keras are top options that are often used in combination.

Copyright © 2023 Plotly. All rights reserved.

What tools can data scientists use to visualize Python analytics?

To build effective Python analytics visualizations, data scientists can combine multiple libraries for maximum flexibility and added capabilities. Python data visualization usually starts with importing packages like NumPy or pandas and visualization libraries like Plotly. Then, data scientists need to import and clean their data to make sure it's accurate. Finally, they need to design visualizations that communicate the information effectively.

Top 10 Python Tools for Data Scientists

Frameworks and libraries make Python analytics and ML development easier and faster for data scientists. Here are the top 10 tools data scientists can use to innovate at scale and take their data analytics capabilities to the next level.

1. NumPY
2. SciPy
3. pandas
4. matplotlib
5. Plotly
6. Cufflinks
7. Black
8. scikit-learn
9. SQLAlchemy
10. TensorFlow

1. NumPy

NumPy (Numerical Python) is a go-to tool in the data science community that enables users to create high-performance multidimensional arrays and matrix data structures. It's an excellent tool for performing mathematical operations on large datasets. It also enables users to take advantage of a unique type of array called ndarray — which is a multidimensional container of items of the same type and size — and provides methods to efficiently operate on it.

Benefits

- The NumPy API can be used in most data science and scientific Python packages including pandas, SciPy, matplotlib, and scikit-learn.
- It also provides a flexible array object that can be used to perform a range of mathematical operations.
- Arrays are faster and more compact than lists in Python. They also take up less memory to store data.

Copyright © 2023 Plotly. All rights reserved.

When to Use It

Data scientists might use NumPy to perform matrix multiplication or to calculate the eigenvectors (a nonzero vector of a linear transformation) and eigenvalues (the factor by which the eigenvector is scaled) of a matrix. They can also use it to do data analysis using NumPy's mathematical functions or to perform a Fourier transform (FT) on a time series.



Using a matrix decomposition to generate a compressed approximation of an image with NumPy. Source.

2. SciPy

For data scientists who need to do things like optimization, linear algebra, integration, interpolation, and statistics, SciPy (Scientific Python) is an open-source scientific computing library for Python. It's built on NumPy and provides additional functionality including a range of statistical functions like hypothesis testing, probability distributions, and descriptive statistics. You can also use its sub-packages for specialized tasks like image processing, sparse matrix manipulation, and clustering algorithms.

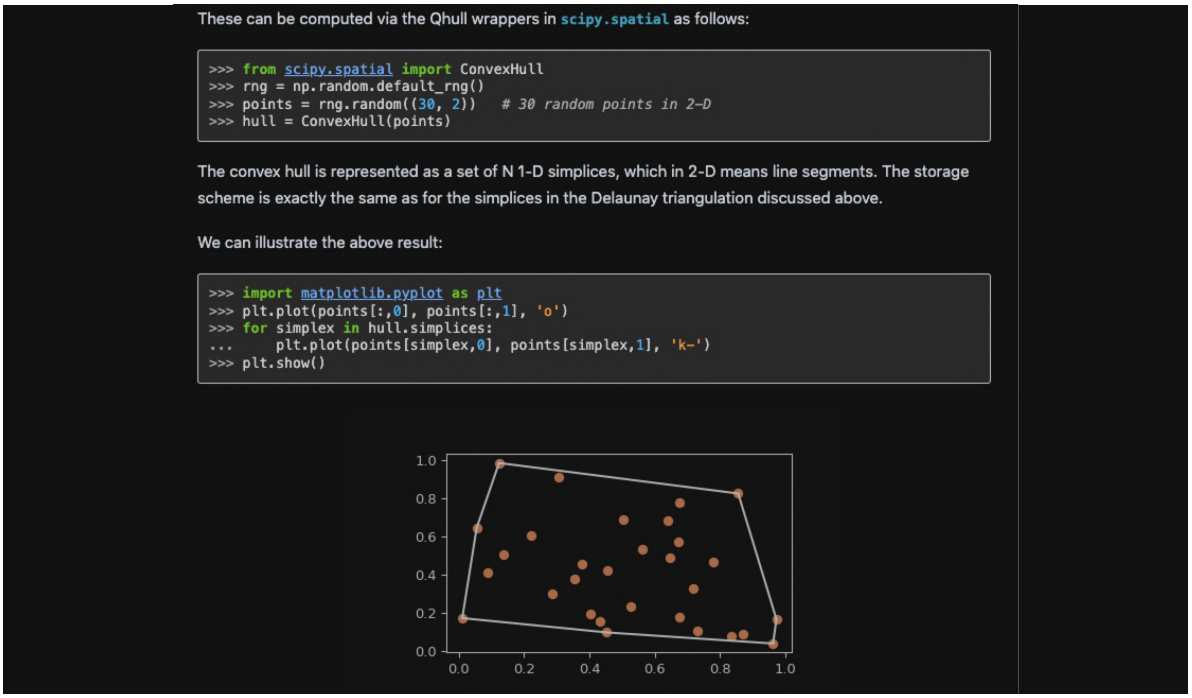
Benefits

- SciPy provides a wide range of mathematical and scientific functions that can be used for highly complex analysis tasks.
- It easily integrates well with other Python libraries, such as matplotlib, pandas, and scikit-learn.
- It's more user-friendly and efficient for scientific calculations than NumPy.

When to Use It

Data scientists can use Scipy to perform tasks like numerical integration or solving differential equations. You can also perform hypothesis testing or estimate the parameters of a probability distribution from a set of observations.

Copyright © 2023 Plotly. All rights reserved.



Working with a convex hull using SciPy. Source.

3. pandas

pandas (Python Data Analysis) is a popular Python package that provides high-performance data manipulation and analysis tools that are built on NumPy. It supports reading and writing data from various file formats, including CSV, Excel, SQL databases, and more. It allows data scientists to leverage both Series and DataFrame structures and makes working with relational and labeled data easy and intuitive. pandas is primarily used for real-world data analysis in Python, allowing users to clean, transform, and reshape data, as well as perform common data analysis tasks, such as grouping, filtering, and aggregation.

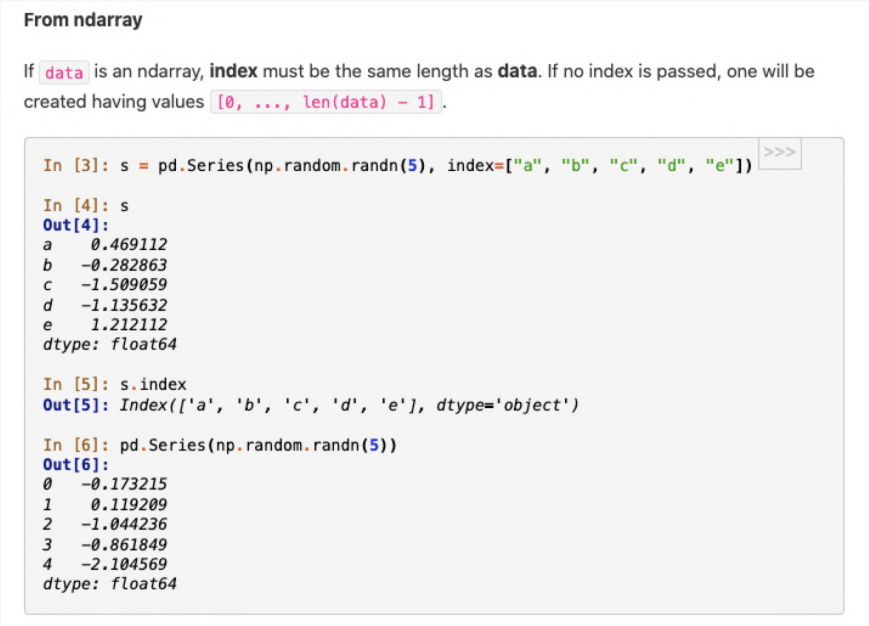
Benefits

- pandas is highly customizable and extensible, with many third-party libraries and tools built on top of it.
- It enables you to create your own function and run it across a series.
- It also allows you to deal with missing data thanks to its syntax and robust functionality.

When to Use It

A data scientist might use pandas to read a large dataset, clean up missing or incorrect values, and perform data transformations to prepare it for further analysis. They could then use it to aggregate the data and perform statistical analysis, generating insights.

Copyright © 2023 Plotly. All rights reserved.



Working with ndarray using pandas. Source.

4. matplotlib

matplotlib is widely used for creating Python graphs, including both static and interactive data visualizations as well as more complex visualizations like sub-plots, grids, and animations. It can be combined with NumPy, pandas, and SciPy. It allows users to create graphs similar to R or Matlab. Matplotlib also has great documentation for beginners, including tutorials as well as a large community of users who contribute to the library.

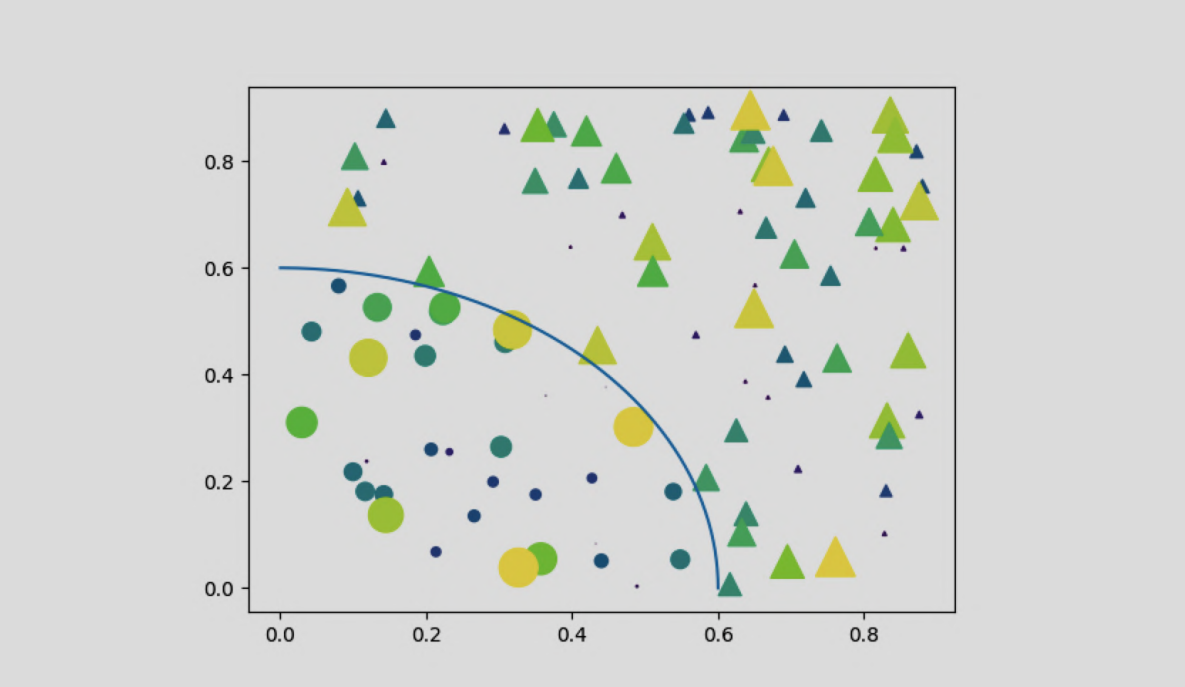
Benefits

- When using matplotlib visualizations alongside ML, it enables you to easily catch outliers in your data.
- It has low memory consumption for enhanced runtime and can be used on almost any operating system.
- You can easily embed data visualizations in JupyterLab and Graphical User Interfaces (GUIs), like a website.

When to Use It

Data scientists can use matplotlib to create effective charts for showing business metrics like sales figures for different product categories, making it easy to identify top-selling products and areas for improvement. They can easily create multiple visualizations to bring data into a dashboard for non-technical users to view.

Copyright © 2023 Plotly. All rights reserved.



Creating a scatter plot with a masked region with matplotlib. Source.

5. Plotly

Plotly makes Dash, a platform that allows you to build interactive, dynamic data visualizations and data apps for Python. It has over 40 different chart types that cover a variety of use cases from statistics to 3D modeling. Plotly enables data scientists to create beautiful web-based visualizations that can be displayed in Jupyter notebooks, saved to standalone HTML files, or served as part of pure Python-built web applications using Dash.

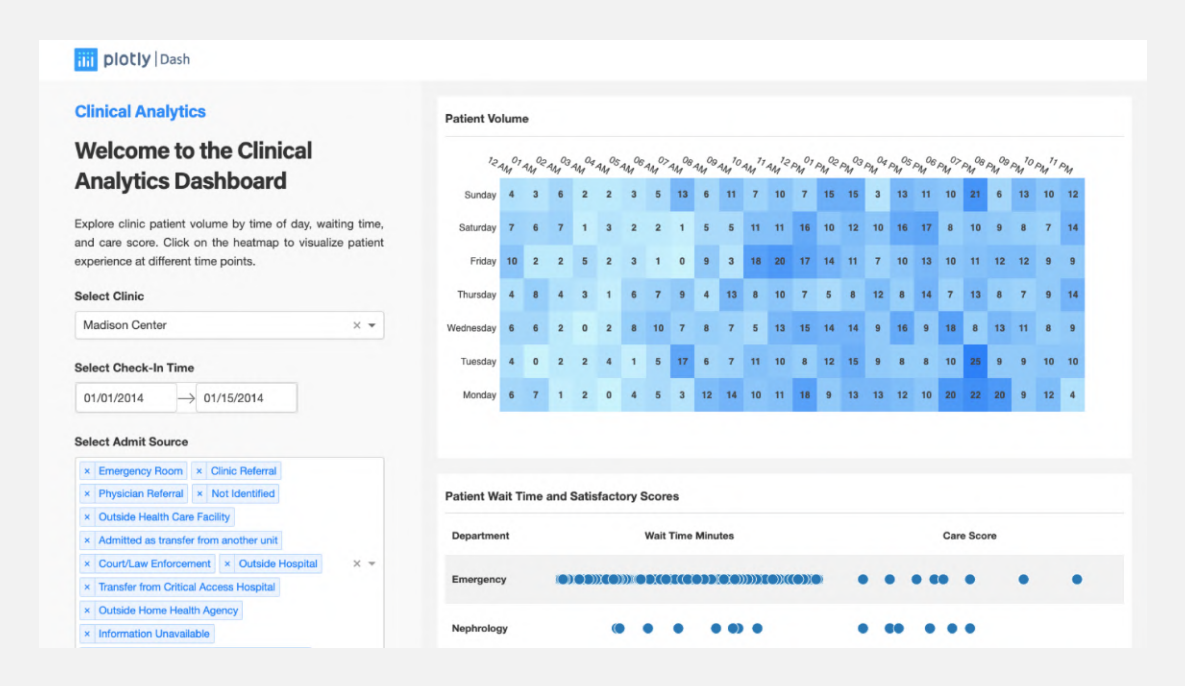
Benefits

- Plotly is highly customizable, allowing users to create visualizations that fit their specific needs and style preferences to communicate information in an engaging way.
- It has a large community of users who constantly contribute to its development and provide support.
- Plotly's interactivity features make it ideal for exploring and presenting complex datasets to gain deeper insights.
- Its Dash Enterprise platform has one-click deployment and stateless architecture that enables the ability to install on any cloud VPC including AWS, Azure, or GCP, making it efficient and highly scalable.

Copyright © 2023 Plotly. All rights reserved.

When to Use It

Dash is great for making customized, interactive data visualizations when you need to present data to executives or set up visualizations that can be used on an ongoing basis to make business decisions across departments. It makes it easy for data scientists to get creative with data visualization so that the insights they find really make an impact on their audience.



Data visualization app made with Dash. Source.

6. Cufflinks

Cufflinks is a Python library that connects Plotly and pandas so that users can build data visualizations directly on data frames. It makes it easy for data scientists and analysts to create interactive data visualizations without the need for advanced programming skills. Cufflinks' ability to create interactive plots allows users to explore their data in real time, enabling them to discover insights that may not be apparent in static charts or graphs.

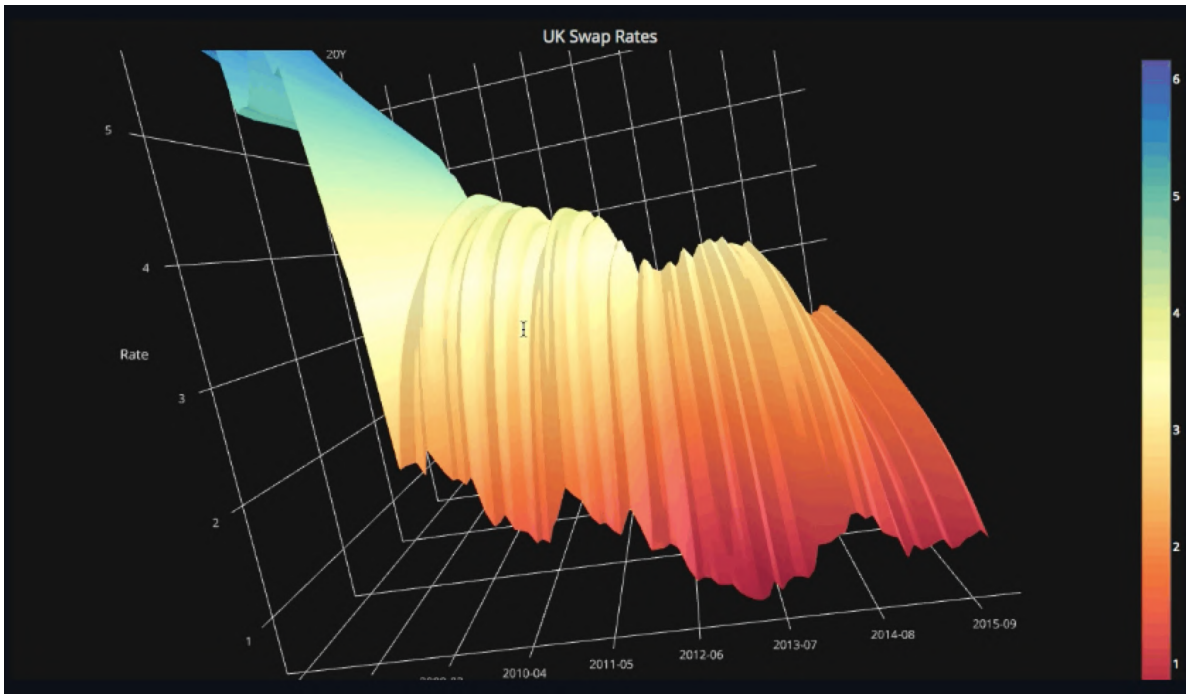
Benefits

- Cufflinks supports a variety of customization options, including chart titles, axis labels, and colors, and can be used to create interactive plots in Jupyter notebooks, as well as standalone HTML files.
- It enables users to quickly explore large datasets and identify trends and patterns.

Copyright © 2023 Plotly. All rights reserved.

When to Use It

With Cufflinks, data scientists can create interactive plots so that their audience can easily explore their data on their own—no matter what their level of technical skill might be. It can also be used to build internal web-based tools by allowing developers to embed interactive plots into web pages.



Plotting with pandas and Plotly using Cufflinks. Source.

7. Black

Black is a Python formatter that ensures your code adheres to a set of style guidelines. It reduces the amount of time you spend formatting your code so you can focus on more important tasks, like writing the code itself. It ensures everyone stays on the same page about code formatting. There isn't much room for customization, but this helps reinforce the same formatting rules from user to user.

Benefits

- Black is extremely simple to set up.
- It can save developers time by automatically formatting code, eliminating the need to manually format code or argue about formatting choices with team members.
- Black always verifies that the formatted output creates the same internal representation in Python, so it never applies changes that could result in a different code interpretation.
- Makes code review faster for data science teams.

Copyright © 2023 Plotly. All rights reserved.

When to Use It

Black is great for projects where not everyone may be familiar with the desired coding standards. Data science teams can ensure that all code submissions follow the same style guidelines, making it easier to review and integrate new code into the project.



```
# in:

def very_important_function(template: str, *variables, file: os.PathLike, engine: str, heade
    """Applies `variables` to the `template` and writes to `file`."""
    with open(file, 'w') as f:
        ...

# out:

def very_important_function(
    template: str,
    *variables,
    file: os.PathLike,
    engine: str,
    header: bool = True,
    debug: bool = False,
):
    """Applies `variables` to the `template` and writes to `file`."""
    with open(file, "w") as f:
        ...
```

Line wrapping formatting with Black. Source.

8. scikit-learn

scikit-learn provides many of the most commonly used machine learning algorithms in its open-source library. It offers a range of supervised and unsupervised learning algorithms, including regression, classification, clustering, and dimensionality reduction. Additionally, it has tools for data preprocessing, feature selection, and model evaluation.

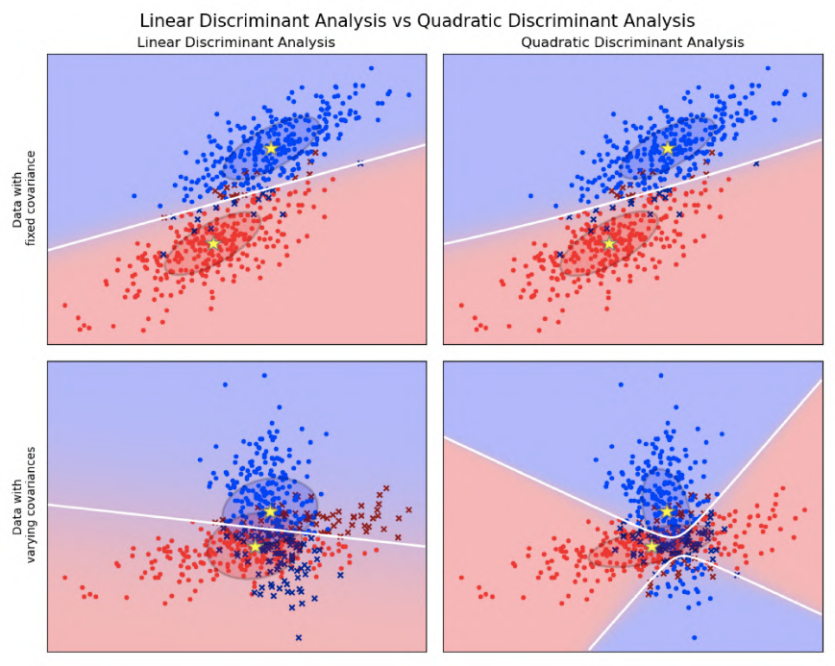
Benefits

- scikit-learn provides a simple and consistent interface for working with machine learning models.
- It integrates well with other Python libraries, such as NumPy, pandas, and matplotlib.
- scikit-learn also has built-in functionality for preprocessing and feature selection, making it easy to prepare data for machine learning.

Copyright © 2023 Plotly. All rights reserved.

When to Use It

One of the main ways data scientists and ML engineers use scikit-learn is to build machine learning models for predictive modeling tasks. Data scientists can also use it to perform clustering analysis on a dataset using clustering algorithms. This can help identify groups of similar data points that can be used for further analysis or segmentation.



Linear and Quadratic Discriminant Analysis with covariance ellipsoid using scikit-learn. Source.

9. SQLAlchemy

If you need to combine the power of SQL and Python, look no further than SQLAlchemy. This tool is an Object Relational Mapper that offers data scientists and developers the full capability and flexibility of SQL, allowing communication between Python programs and relational databases. It does this by translating Python classes to tables and turning function calls into SQL statements. It allows data scientists to use Python to query their database rather than SQL, making it easier to write more complex queries.

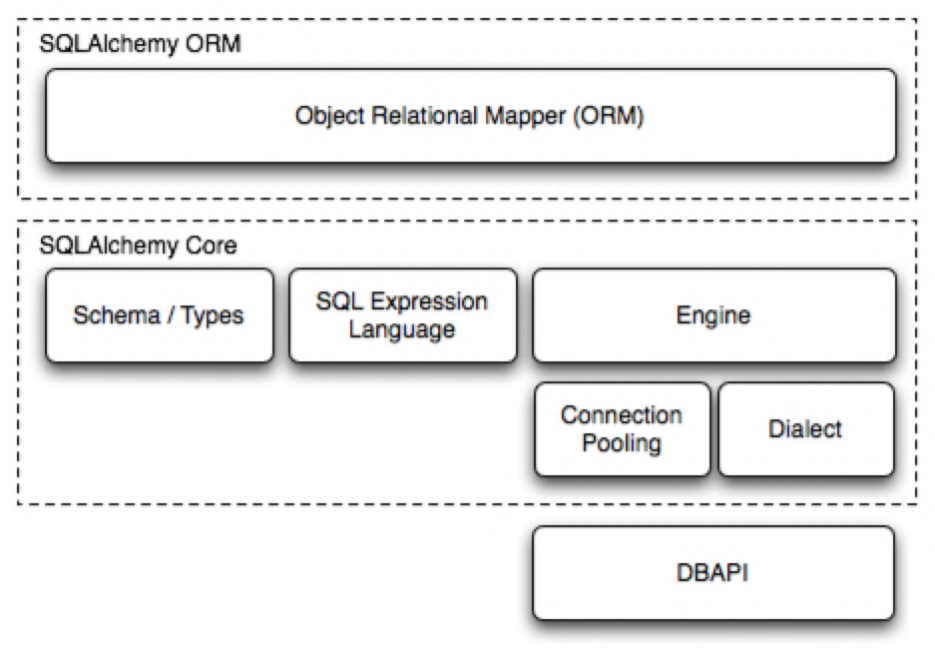
Benefits

- SQLAlchemy makes it much easier to work with databases in Python.
- It supports many common database backends, including MySQL, PostgreSQL, SQLite, and Oracle.
- Features include database transactions and connection pooling making it ideal for building high-performance, scalable database applications.

Copyright © 2023 Plotly. All rights reserved.

When to Use It

SQLAlchemy is great for data scientists who want to bypass the need to set up a relational database management system and write SQL to carry out database operations. It allows users to do things like building data analysis apps using Python that interact directly with a database.



SQLAlchemy components. Source.

10. TensorFlow

TensorFlow is an open-source platform for building ML models that's great for both beginners and seasoned data scientists, developers, and ML engineers. It enables users to work on a variety of ML, deep learning, and NLP tasks. ML models made with TensorFlow can be deployed on desktop, mobile, web, and in the cloud. Because clean data is at the heart of any successful ML project, TensorFlow also includes tools for consolidating, cleaning, and processing data at scale.

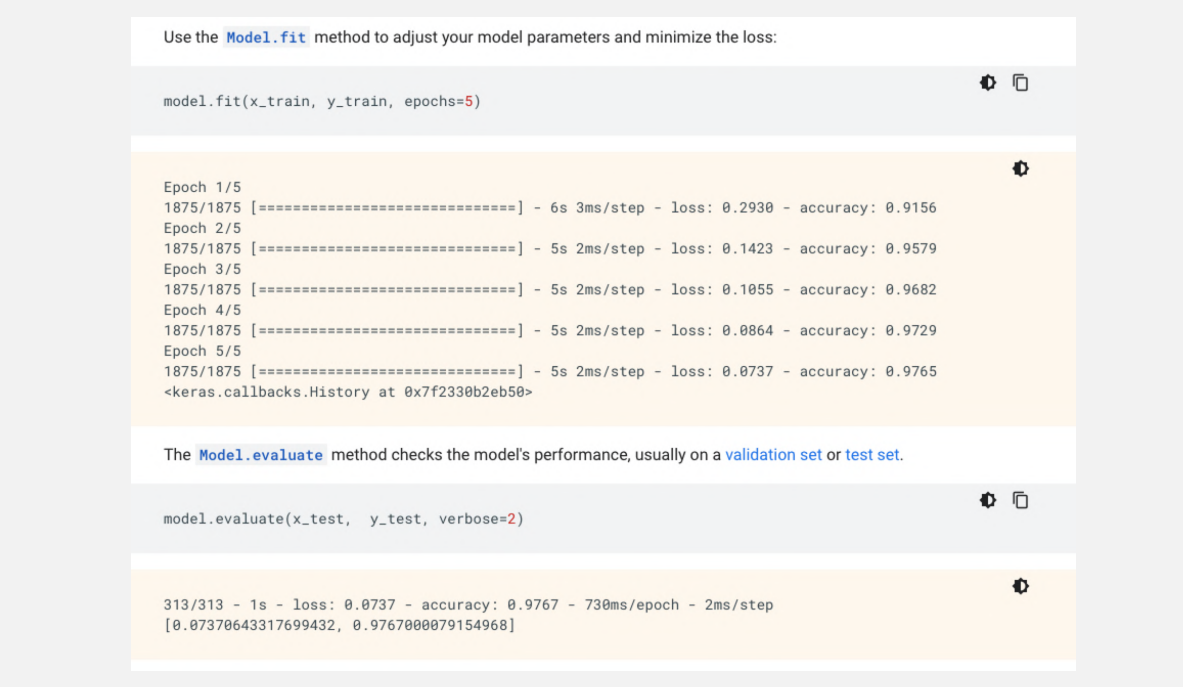
Benefits

- TensorFlow supports many data types, including images, text, and time series data.
- It also facilitates automatic differentiation, making it easier to implement more complex algorithms.
- It has GPU support for faster training of deep learning models.
- TensorFlow enables users to scale quickly by making it possible to train large models on distributed systems.

Copyright © 2023 Plotly. All rights reserved.

When to Use It

Data scientists and ML engineers can use TensorFlow to build models that can generate or translate natural language text. It can also be used for speech recognition projects, helping data scientists turn spoken words into text data for analysis.



Train and evaluate your ML model with TensorFlow. Source.

BONUS: 5 Tools to Boost Your Data Science Projects

Low-Customization Plotting: Seaborn

When it comes to Python data visualization, Seaborn is a solid tool built specifically for statistical visualizations. It's a library based on matplotlib that provides an interface for creating various types of graphs. It supports customizable visualizations that work for many types of reporting and analytics, including scatter plots, line plots, bar plots, and heat maps. It also provides built-in functionality for visualizing statistical relationships, such as regression models and distribution plots.

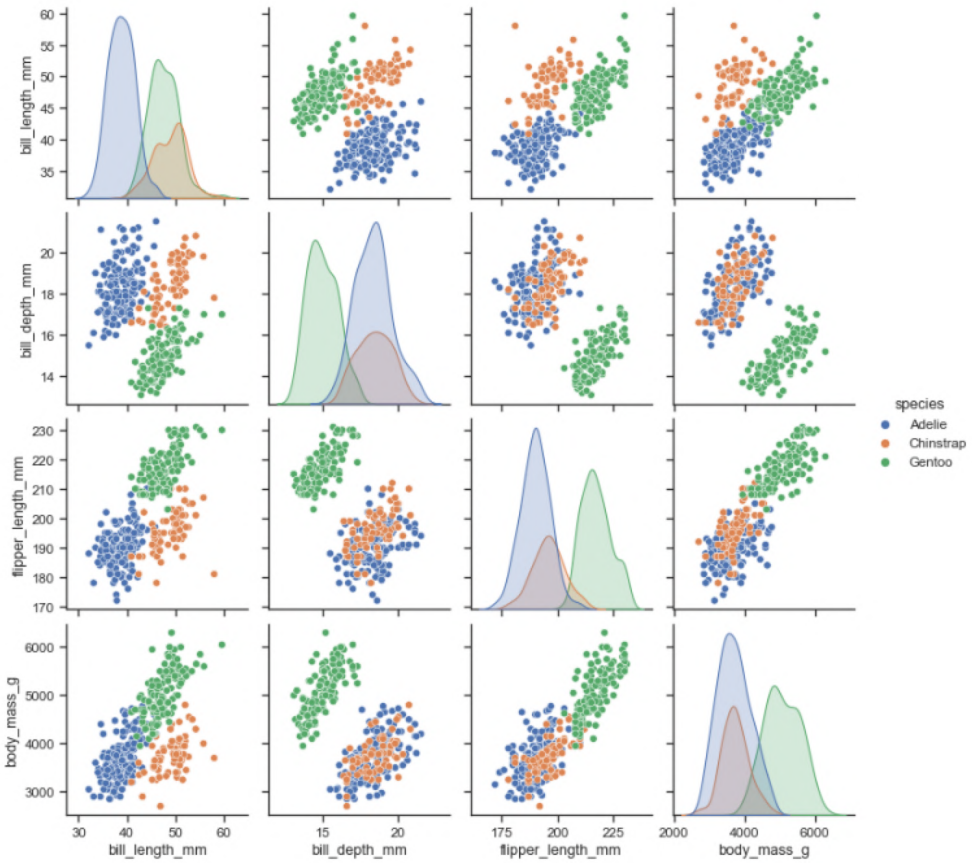
Copyright © 2023 Plotly. All rights reserved.

Benefits

- Seaborn is easy to use and provides a high-level interface for creating complex visualizations.
- It offers a range of built-in functions that simplify data exploration and visualization.
- Seaborn's ready-to-use plots usually don't need too much customization, so they are ideal for rapid prototyping and data exploration.
- It integrates well with other Python libraries, such as pandas and matplotlib.
- It provides robust documentation and tutorials for beginners.

When to Use It

Data scientists might use Seaborn to create a scatter plot showing the relationship between two variables, along with a regression line and confidence interval. You could then add labels and annotations to the plot, making it easy to interpret and communicate the results. Seaborn is also commonly used for data visualization in "print" material like research papers and presentations because it produces visually appealing graphics that are informative and engaging.



Scatterplot matrix using Seaborn. Source.

Copyright © 2023 Plotly. All rights reserved.

Reduce Code Reviews: Pylint

Pylint is a static code analysis tool that checks Python code for errors, bugs, and style issues. It integrates with popular development tools, including IDEs and text editors, and can be run from the command line. Pylint can save developers time by automatically checking code for issues, reducing the need for manual code reviews. It can also make it easier to maintain code as it identifies complex or convoluted code that may be difficult to understand or modify.

Benefits

- Pylint checks the length of each line, whether variable names are well-formed according to the project's coding standard, and whether declared interfaces are truly implemented.
- It also detects duplicate code and checks if imported modules are used.
- Pylint is highly customizable so you can indicate which errors and code conventions are important to you. You can also write plugins to further customize its functionality.

When to Use It

Similar to Black, Pylint is great for increasing productivity. Data scientists and developers can use it to streamline their coding sessions by reducing the need to go back and check their code. It's also great for teams who need to make sure that every piece of code meets the same standards.

```
#!/usr/bin/env python
"""This script prompts a user to enter a message to encode or decode
using a classic Caeser shift substitution (3 letter shift)"""

import string

shift = 3
choice = raw_input("would you like to encode or decode?")
word = (raw_input("Please enter text"))
letters = string.ascii_letters + string.punctuation + string.digits
encoded = ''
if choice == "encode":
    for letter in word:
        if letter == ' ':
            encoded = encoded + ' '
        else:
            x = letters.index(letter) + shift
            encoded = encoded + letters[x]
if choice == "decode":
    for letter in word:
        if letter == ' ':
            encoded = encoded + ' '
        else:
            x = letters.index(letter) - shift
            encoded = encoded + letters[x]

print encoded
```

Code edited with PyLint. Source.

Copyright © 2023 Plotly. All rights reserved.

Analyze NL Data: Natural Language Toolkit

Natural Language Toolkit (NLTK) is one of the best platforms for working with natural language, or human language, data. It has extremely robust natural language processing (NLP) libraries as well as libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, which can help data scientists build ML systems around NLP. It also has a suite of interfaces to over 50 corpora and lexical resources such as WordNet.

Benefits

- NLTK provides a wide range of tools and algorithms for working with text data, making it a versatile library for NLP tasks.
- It includes pre-trained models and corpora that can be used to quickly get started with NLP tasks for different languages and text types.
- NLTK provides an intuitive interface for working with text data in Python, making it easy to manipulate and analyze text data.
- It is extensible and modular so you can easily customize and extend the library to meet your specific needs.

When to Use It

Data scientists working with natural language data can use it for text classification tasks, such as sentiment analysis or spam filtering. A data scientist might use NLTK to preprocess and clean text data, extract features from the text, and train an ML model to classify text data into different categories.

Tokenize and tag some text:

```
>>> import nltk
>>> sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',
'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
>>> tagged = nltk.pos_tag(tokens)
>>> tagged[0:6]
[('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'),
('Thursday', 'NNP'), ('morning', 'NN')]
```

Try out basic NL functions like tokenizing and tagging text with NLTK. Source.

Copyright © 2023 Plotly. All rights reserved.

Launch Deep Learning: PyTorch

PyTorch is an open-source framework that provides an easy-to-use interface for building and training deep learning models. It's easy for people already familiar with Python to use and is great for experimenting and prototyping ML and NLP projects. It enables users to build various deep learning models including neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs). PyTorch is built with a strong focus on flexibility and ease of use, providing an intuitive interface for building and deploying machine learning models.

Benefits

- PyTorch supports automatic differentiation, which can help speed up model development and reduce errors during backpropagation.
- Its dynamic computational graph feature makes it possible to define models that can handle varying input sizes and shapes, making it more flexible than other deep learning frameworks.
- It also provides support for distributed computing, which makes it possible to train models on large-scale datasets while maintaining high performance.

When to Use It

A data scientist might use PyTorch to build and train a deep learning model that can accurately classify images based on their content. You can also use it for NLP tasks or to build models that can analyze and classify text data.

```
import torch.distributed as dist
from torch.nn.parallel import DistributedDataParallel

dist.init_process_group(backend='gloo')
model = DistributedDataParallel(model)
```

Take advantage of distributed training across teams using PyTorch. Source.

Copyright © 2023 Plotly. All rights reserved.

Launch Deep Learning: PyTorch

Keras provides a high-level API for the TensorFlow library, making it easier for users to build and train deep learning models and to implement (and experiment with) neural networks. It has a library of commonly used neural network building blocks including layers, objectives, activation functions, and optimizers, as well as tools that make simplifying image and text data a little easier. It also has support for convolutional and recurrent neural networks. Keras is built on top of TensorFlow, allowing it to take advantage of TensorFlow's computational graph and automatic differentiation capabilities.

Benefits

- Keras has a user-friendly interface for building and training deep learning models, making it easy for data scientists to get started with deep learning.
- It offers a range of pre-built layers and models that can be used to build and train deep learning models quickly and easily.
- Keras provides a high-level API that abstracts away many of the low-level details of deep learning, allowing developers to focus on building and training models rather than on low-level implementation details.

When to Use It

Data scientists might want to use Keras to build and train a deep learning model that can accurately classify images into different categories, such as animals, vehicles, or buildings. The model would be trained on a large dataset of labeled images, and then used to classify new images in real time. You can also use it for many other deep learning applications including object detection and NLP.

```
import os

num_skipped = 0
for folder_name in ("Cat", "Dog"):
    folder_path = os.path.join("PetImages", folder_name)
    for fname in os.listdir(folder_path):
        fpath = os.path.join(folder_path, fname)
        try:
            fobj = open(fpath, "rb")
            is_jfif = tf.compat.as_bytes("JFIF") in fobj.peek(10)
        finally:
            fobj.close()

        if not is_jfif:
            num_skipped += 1
            # Delete corrupted image
            os.remove(fpath)

print("Deleted %d images" % num_skipped)
```

Filtering out corrupted images to perform image classification with Keras. Source.

Copyright © 2023 Plotly. All rights reserved.

Why Python Tools Matter for Data Scientists

No matter what projects you're working on, Python tools enable you to speed up and scale the work you're doing. Data scientists have a huge role to play in the future of enterprise success and growth, as well as innovation and development in the realms of ML and deep learning.

Python tools give data scientists the flexibility to be as creative as they need to be to make an impact. With engaging data visualizations, businesses can make better use of their data to make decisions that help them pull ahead of the competition. Using tools that make ML more approachable for more data scientists, enterprise teams can develop models that make businesses far more productive and sustainable in the long run.

Learn more about using Plotly to bring your Python data visualizations to life with low-code tools for building data apps.

About Plotly

Plotly is a software company whose mission is to enable every company, around the world, to build data apps. Our product, Dash Enterprise, is a platform of best-in-class development tools to quickly and easily visualize data in Python from virtually any data source. With customers across the Fortune 500, Plotly is a category-defining leader in enabling data-driven decisions from advanced analytics, machine learning, and artificial intelligence. For more information, visit https://plotly.com.

Copyright © 2023 Plotly. All rights reserved.