

▼ Install PyCaret

PyCaret is an open-source, low-code machine learning library in Python designed to make the end-to-end machine learning process easier and faster for both beginners and experienced data scientists. It offers a variety of tools and functions for data preparation, model training, model selection, and deployment.

`pip3 install pycaret==2.3.6` is a command used to install a specific version of the PyCaret library in Python. If you encounter an error message, try running the code again.

```
# install slim version (default)
!pip install pycaret
```



Please follow our [blog](#) to see more information about new features, tips and tricks, and featured notebooks such as [Analyzing a Bank Failure with Colab](#).

2025-01-13

- Released version 1.2.0 of the ([Open in Colab Chrome Extension](#)).
- Released minimizable comments with indicators next to cell.
- TPU v5e-1 Runtimes are now available for selection ([tweet](#)).
- GPU prices were decreased ([tweet](#)).

Python package upgrades

- accelerate 1.1.1 -> 1.2.1
- aiohttp 3.10.10 -> 3.11.11
- altair 4.2.2 -> 5.5.0
- bigframes 1.25.0 -> 1.29.0
- cmake 3.30.5 -> 3.31.2
- cvxpy 1.5.3 -> 3.6.0
- earthengine-api 1.2.0 -> 1.4.3
- folium 0.18.0 -> 0.19.3
- holidays 0.60 -> 0.63
- huggingface-hub 0.26.2 -> 0.27.0
- jsonpickle 3.4.2 -> 4.0.1
- kagglehub 0.3.3 -> 0.3.6
- keras 3.4.1 -> 3.5.0
- matplotlib 3.8.0 -> 3.10.0
- openai 1.54.3 -> 1.57.4
- pymc 5.18.0 -> 5.19.1
- safetensors 0.4.5 -> 0.5.0
- scikit-image 0.24.0 -> 0.25.0
- scikit-learn 1.5.2 -> 1.6.0
- sentence-transformers 3.2.1 -> 3.3.1
- tensorflow 2.17.0 -> 2.17.1
- torch 2.5.0 -> 2.5.1
- torchaudio 2.5.0 -> 2.5.1
- torchvision 0.20.0 -> 0.20.1
- transformers 4.46.2 -> 4.47.1
- wandb 0.18.6 -> 0.19.1
- xarray 2024.10.0 -> 2024.11.0

Python package inclusions

- google-genai 0.3.0

2024-11-11

- Users can now import Gemini API keys from AI Studio into their user secrets, all in Colab ([tweet](#)).
- Increased limit to 1000 characters for requests to Gemini in Chat and Generate windows.
- Improved saving notebook to GitHub flow.
- Updated Gemini spark icon to be colorful.

```
Requirement already satisfied: nbclassic>=0.4.7 in /
Requirement already satisfied: notebook-shim>=0.2.3
Requirement already satisfied: beautifulsoup4 in /us
Requirement already satisfied: bleach!=5.0.0 in /usr
Requirement already satisfied: defusedxml in /usr/lc
Requirement already satisfied: jupyterlab-pygments i
Requirement already satisfied: mistune<4,>=2.0.3 in
Requirement already satisfied: nbclient>=0.5.0 in /u
Requirement already satisfied: pandocfilters>=1.4.1
Requirement already satisfied: argon2-cffi-bindings
Requirement already satisfied: webencodings in /usr/
Requirement already satisfied: tinycss2<1.5,>=1.1.0
Requirement already satisfied: jupyter-server<3,>=1.
Requirement already satisfied: cffi>=1.0.1 in /usr/l
Requirement already satisfied: soupsieve>1.2 in /usr
Requirement already satisfied: pycparser in /usr/loc
Requirement already satisfied: anyio>=3.1.0 in /usr/
Requirement already satisfied: websocket-client in /
Requirement already satisfied: sniffio>=1.1 in /usr/
```

```
#Run the below code in your notebook to check the instal
from pycaret.utils import version
version()
```

→ '3.3.2'

Install Analysis Extras for Explainable AI

```
!pip install pycaret[analysis]
```

→

- [uv](#) is pre-installed on the PATH for faster package installs.

• Fixed bugs

- Dropdown text for GitHub repository not visible [#4901](#).
- Pre-installed California housing dataset README not correct [#4862](#).
- Backend execution error for scheduled notebook [#4850](#).
- Drive File Stream issues [#3441](#).
- Linking to the signup page does not preserve the authuser parameter.
- Error messages in Gemini chat are not polished.
- Clicking in Gemini chat feedback causes jitters the UI.
- Hovering over a table of contents entry would show the menu icons for all entries.
- Surveys display over open dialogs.
- Playground mode banner not shown on mobile.

Python package upgrades

- accelerate 0.34.2 -> 1.1.1
- arviz 0.19.0 -> 0.20.0
- bigframes 1.18.0 -> 1.25.0
- bigquery-magics 0.2.0 -> 0.4.0
- bokeh 3.4.3 -> 3.6.1
- blosc 2.0.0 -> 2.7.1
- cloudpickle 2.2.1 -> 3.1.0
- cudf-cu12 24.4.1 -> 24.10.1
- dask 2024.8.0 -> 24.10.0
- debugpy 1.6.6 -> 1.8.0
- earthengine-api 1.0.0 -> 1.2.0
- folium 0.17.0 -> 0.18.0
- gscfs 2024.6.1 -> 2024.10.0
- geemap 0.34.3 -> 0.35.1
- holidays 0.57 -> 0.60
- huggingface-hub 0.24.7 -> 0.26.2
- kagglehub 0.3.0 -> 0.3.3
- lightgbm 4.4.0 -> 4.5.0
- lxml 4.9.4 -> 5.3.0
- matplotlib 3.7.1 -> 3.8.0
- mizani 0.11.4 -> 0.13.0
- networkx 3.3 -> 3.4.2
- nltk 3.8.1 -> 3.9.1
- pandas 2.1.4 -> 2.2.2
- pillow 10.4.0 -> 11.0.0
- plotnine 0.13.6 -> 0.14.1
- polars 1.6.0 -> 1.9.0
- protobuf 3.20.3 -> 4.25.5
- pyarrow 14.0.2 -> 17.0.0
- pydrive2 1.20.0 -> 1.21.1
- pymc 5.16.2 -> 5.18.0
- torch 2.4.1 -> 2.5.0

```
using cached salt-n-1.0.1-py3-none-any.whl (110 kB)
Using cached wtforms-3.2.1-py3-none-any.whl (152 kB)
Using cached ansi2html-1.9.2-py3-none-any.whl (17 kB)
Using cached puremagic-1.28-py3-none-any.whl (43 kB)
Using cached pywavelets-1.8.0-cp311-cp311-manylinux_
Using cached multiprocess-0.70.17-py311-none-any.whl
Downloading zope.event-5.0-py3-none-any.whl (6.8 kB)
Downloading zope.interface-7.2-cp311-cp311-manylinux
                                                 259.8/25
```

```
Building wheels for collected packages: htmlmin, das
  Building wheel for htmlmin (setup.py) ... done
    Created wheel for htmlmin: filename=htmlmin-0.1.12
      Stored in directory: /root/.cache/pip/wheels/8d/55
  Building wheel for dash-cytoscape (setup.py) ... c
    Created wheel for dash-cytoscape: filename=dash_cy
      Stored in directory: /root/.cache/pip/wheels/99/b1
Successfully built htmlmin dash-cytoscape
Installing collected packages: puremagic, htmlmin, c
  Attempting uninstall: slicer
    Found existing installation: slicer 0.0.8
  Uninstalling slicer-0.0.8:
    Successfully uninstalled slicer-0.0.8
  Attempting uninstall: shap
    Found existing installation: shap 0.47.1
  Uninstalling shap-0.47.1:
    Successfully uninstalled shap-0.47.1
  Attempting uninstall: dash
    Found existing installation: dash 3.0.3
  Uninstalling dash-3.0.3:
    Successfully uninstalled dash-3.0.3
Successfully installed Flask-WTF-1.2.2 PyWavelets-1.
```

Principal Component Analysis

PCA stands for Principal Component Analysis, and it is a widely used technique in data analysis and machine learning for reducing the dimensionality of large datasets. The basic idea behind PCA is to transform a set of high-dimensional variables into a smaller set of uncorrelated variables called principal components, while retaining as much of the original variance as possible.

Import Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style="darkgrid")
import pandas as pd
plt.rcParams['figure.figsize'] = (7,5)

from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

- torchaudio 2.4.1 -> 2.5.0
- torchvision 0.19.1 -> 0.20.0
- transformers 4.44.2 -> 4.46.2
- xarray 2024.9.0 -> 2024.10.0

Python package inclusions

- diffusers 0.31.0
- gitpython 3.1.43
- langchain 0.3.7
- openai 1.54.3
- pygit2 1.16.0
- pyspark 3.5.3
- sentence-transformers 3.2.1
- timm 1.0.11
- wandb 0.18.6

Library and driver upgrades

- drivefs upgraded from 89.0.2 to 98.0.0

2024-09-23

- Improved code snippet search
- Updated Marketplace image and public local runtime container
- Improved the look-and-feel of interactive form dropdowns and checkboxes
- Fixed bugs
 - activating the skip link caused the notebook to scroll out of view
 - toggling a checkbox too much caused the page to crash
 - lightning fast drags could cause orphaned tabs
 - custom widgets snippet would show for local runtimes

Python package upgrades

- accelerate 0.32.1 -> 0.34.2
- arviz 0.18.0 -> 0.19
- autograd 1.6.2 -> 1.7.0
- bigframes 1.14.0 -> 1.18.0
- dask 2024.7.1 -> 2024.8.0
- distributed 2024.7.1 -> 2024.8.0
- duckdb 0.10.3 -> 1.1.0
- earthengine-api 0.1.416 -> 1.0.0
- flax 0.8.4 -> 0.8.5
- gdown 5.1.0 -> 5.2.0
- geemap 0.33.1 -> 0.34.3
- geopandas 0.14.4 -> 1.0.1
- google-cloud-aiplatform 1.59.0 -> 1.67.1
- google-cloud-bigquery-storage 2.25.0 -> 2.26.0
- holidays 0.54 -> 0.57
- huggingface-hub 0.23.5 -> 0.24.7
- ibis-framework 8.0.0 -> 9.2.0
- jax 0.4.26 -> 0.4.33
- jaxlib 0.4.26 -> 0.4.33
- kagglehub 0.2.9 -> 0.3.0

```
print("Pandas version: ", pd.__version__)
print("Seaborn version: ", sns.__version__)
```

→ Pandas version: 2.1.4
Seaborn version: 0.13.2

Dataset

The examined group comprised kernels belonging to three different varieties of wheat: Kama, Rosa and Canadian, 70 elements each, randomly selected for the experiment. High quality visualization of the internal kernel structure was detected using a soft X-ray technique. It is non-destructive and considerably cheaper than other more sophisticated imaging techniques like scanning microscopy or laser technology. The images were recorded on 13x18 cm X-ray KODAK plates. Studies were conducted using combine harvested wheat grain originating from experimental fields, explored at the Institute of Agrophysics of the Polish Academy of Sciences in Lublin.

The data set can be used for the tasks of classification and cluster analysis.

Attribute Information:

To construct the data, seven geometric parameters of wheat kernels were measured:

1. area A,
2. perimeter P,
3. compactness $C = 4\pi A/P^2$,
4. length of kernel,
5. width of kernel,
6. asymmetry coefficient
7. length of kernel groove. All of these parameters were real-valued continuous.

<https://archive.ics.uci.edu/ml/datasets/seeds>

Read the dataset

`pd.read_csv` is a function in the pandas library in Python that is used to read a CSV (Comma Separated Values) file and convert it into a pandas DataFrame.

```
#read cvs file into dataframe
df = pd.read_csv('https://raw.githubusercontent.com/ecvbe
```

- lightgbm 4.4.0 -> 4.5.0
- matplotlib-venn 0.11.10 -> 1.1.1
- mizani 0.9.3 -> 0.11.4
- Pillow 9.4.0 -> 10.4.0
- plotly 5.15.0 -> 5.24.1
- plotnine 0.12.4 -> 0.13.6
- polars 0.20.2 -> 1.6.0
- progressbar2 4.2.0 -> 4.5.0
- PyDrive2 1.6.3 -> 1.20.0
- pymc 5.10.4 -> 5.16.2
- pytensor 2.18.6 -> 2.25.4
- scikit-image 0.23.2 -> 0.24.0
- scikit-learn 1.3.2 -> 1.5.2
- torch 2.3.1 -> 2.4.1
- torchaudio 2.3.1 -> 2.4.1
- torchvision 0.18.1 -> 0.19.1
- transformers 4.42.4 -> 4.44.2
- urllib3 2.0.7 -> 2.2.3
- xarray 2024.6.0 -> 2024.9.0

Python package inclusions

- bigquery-magics 0.2.0

2024-08-20

- TPU memory usage and utilization can now be checked with `!tpu-info`
- Gemini Chat responses are now grounded in relevant sources
- Added a new "Create Gemini API key" link in the user secrets panel
- Added a new "Gemini: Creating a prompt" snippet and touched up the existing "Gemini: Connecting to Gemini" snippet
- Added the ability to specify custom placeholder text for various interactive form params (see [examples](#))
- Keyboard navigation a11y improvements to comments UI
- Various minor rendering improvements to interactive forms UI
- A11y improvements for the run button and header
- Updated tooltip styling
- A11y improvements for the file browser's disk usage bar
- On mobile, tooltips now trigger on long press
- On mobile, release notes updates will no longer display automatically
- Python package upgrades
 - astropy 5.3.4 -> 6.1.2
 - bigframes 1.11.1 -> 1.14.0
 - bokeh 3.3.4 -> 3.4.3

```
#df = pd.read_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/mpg.csv')
df.head(25)
```

	manufacturer	mpg	cyl	disp	hp	drat	wt	qsec
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.0
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.0
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.0
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.0
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.0
5	Valiant	18.1	6	225.0	105	2.76	3.460	20.0
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.0
7	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.0
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.0
9	Merc 280	19.2	6	167.6	123	3.92	3.440	18.0
10	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.0
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.0
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.0
13	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.0
14	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.0
15	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.0
16	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.0
17	Fiat 128	32.4	4	78.7	66	4.08	2.200	19.0
18	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.0
19	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.0
20	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.0
21	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.0

Next steps:

[Generate code with df](#)[View recommended plots](#)[New](#)

df.info()

→ <class 'pandas.core.frame.DataFrame'>
 RangeIndex: 32 entries, 0 to 31
 Data columns (total 12 columns):

- dask 2023.8.1 -> 2024.7.1
- earthengine-api 0.1.412 -> 0.1.416
- geopandas 0.13.2 -> 0.14.4
- kagglehub 0.2.8 -> 0.2.9
- keras 2.15.0 -> 3.4.1
- lightgbm 4.1.0 -> 4.4.0
- malloy 2023.1067 -> 2024.1067
- numba 0.58.1 -> 0.60.0
- numpy 1.25.2 -> 1.26.4
- opencv-python 4.8.0.76 -> 4.10.0.84
- pandas 2.0.3 -> 2.1.4
- pandas-gbq 0.19.2 -> 0.23.1
- panel 1.3.8 -> 1.4.5
- requests 2.31.0 -> 2.32.3
- scikit-learn 1.2.2 -> 1.3.2
- scipy 1.11.4 -> 1.13.1
- tensorboard 2.15.2 -> 2.17.0
- tensorflow 2.15.0 -> 2.17.0
- tf-keras 2.15.1 -> 2.17.0
- xarray 2023.7.0 -> 2024.6.0
- xgboost 2.0.3 -> 2.1.1
- Python package inclusions
 - einops 0.8.0

2024-07-22

- You can now embed Google sheets directly into Colab to streamline interactions with data with InteractiveSheet.
- Example:

```
from google.colab import sheets
sh = sheets.InteractiveSheet()
df = sh.as_df()
```

- Fixed multiple rendering bugs in cell editors with wide text content (i.e. text is no longer hidden or clipped)
- Fixed multiple accessibility issues in Colab's comments feature (e.g. proper keyboard focus management, added accessibility landmarks, etc)
- Fixed bug where AI code generation would fail for extremely long broken code snippets
- Fixed multiple scrollbar bugs in the user secrets panel
- Added the ability for workspace admin to purchase Colab Pro and Pro+ Subscriptions for users
- Fixed bug where user secrets couldn't be moved to a tab
- Fixed several focus management accessibility issues in tabs, the table

#	Column	Non-Null Count	Dtype
0	manufacturer	32 non-null	object
1	mpg	32 non-null	float64
2	cyl	32 non-null	int64
3	disp	32 non-null	float64
4	hp	32 non-null	int64
5	drat	32 non-null	float64
6	wt	32 non-null	float64
7	qsec	32 non-null	float64
8	vs	32 non-null	int64
9	am	32 non-null	int64
10	gear	32 non-null	int64
11	carb	32 non-null	int64
dtypes:		float64(5), int64(6), object(1)	
memory usage:		3.1+	KB

```
print("Number of duplicated rows is: ", df.duplicated())
```

→ Number of duplicated rows is: 0

```
print("Number of rows with NaNs is: ", df.isna().any(ax=
```

→ Number of rows with NaNs is: 0

Exploratory Data Analysis

sns.pairplot is a function in the seaborn library in Python that is used to plot pairwise relationships between multiple variables in a dataset. The resulting plot is a grid of scatterplots, with each variable plotted against every other variable.

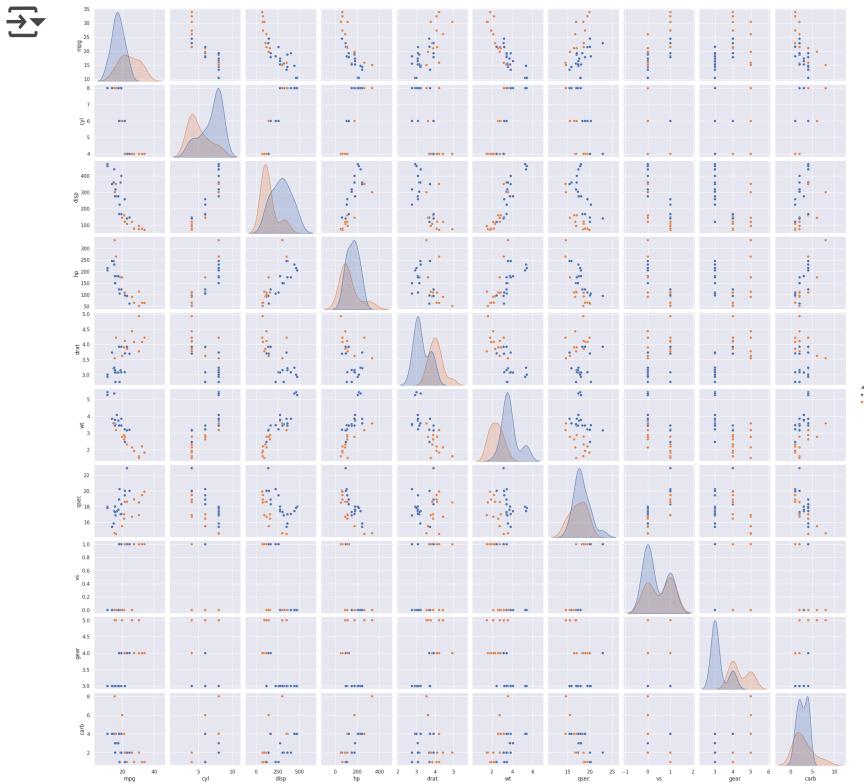
```
sns.pairplot(df, hue='am')
plt.show()
```

of contents, the left toolbar, and the run button

- Fixed bug where overflowing cells may be omitted when pasting from Google Sheets
- Fixed bug where the generate code button did not activate on touch
- Python package upgrades
 - bigframes 1.9.0 -> 1.11.1
 - cvxpy 1.3.4 -> 1.5.2
 - earthengine-api 0.1.408 -> 0.1.412
 - google-api-core 2.11.1 -> 2.19.1
 - google-api-python-client 2.84.0 -> 2.137.0
 - google-cloud-aiplatform 1.56.0 -> 1.59.0
 - google-cloud-bigquery 3.21.0 -> 3.25.0
 - google-cloud-core 2.3.3 -> 2.4.1
 - google-cloud-datastore 2.15.2 -> 2.19.0
 - google-cloud-firebase 2.11.1 -> 2.16.1
 - google-cloud-functions 1.13.3 -> 1.16.4
 - google-generativeai 0.5.4 -> 0.7.2
 - kagglehub 0.2.5 -> 0.2.8
 - pip 23.1.2 -> 24.1.2
 - setuptools 67.7.2 -> 71.0.4
 - sympy 1.12.1 -> 1.13.1
 - torch 2.3.0 -> 2.3.1
 - transformers 4.41.2 -> 4.42.4
- Python package inclusions
 - accelerate 0.32.1

2024-06-18

- Inline AI completions are now available to users on the free-of-charge tier
- Reduced latency for LSP and terminal connections
- Improved quality of inline completions
- Visual improvements to switch controls across Colab
- Various bug fixes, performance and a11y improvements to the user secrets panel
- Improved tooltip UX behavior
- Improved behavior when copying data from Google Sheets and pasting in Colab



```
y =df['am']
y.value_counts().plot(kind='pie')
plt.ylabel('')
plt.show()
```

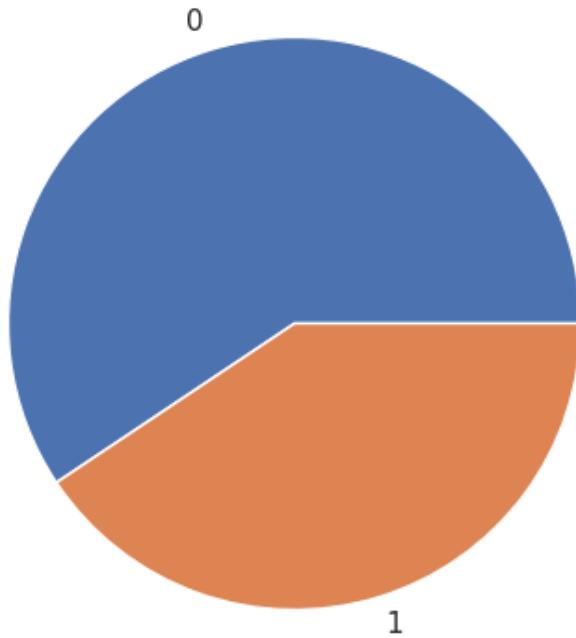
- Scroll to cell fixes for single tabbed view and jump to cell command
- Improved tab header behavior
- A11y improvements for notebook-focused cells
- Python package upgrades
 - torch 2.2.1 -> 2.3.0
 - torchaudio 2.2.1 -> 2.3.0
 - torchvision 0.17.1 -> 0.18.0
 - torchtext 0.17.1 -> 0.18.0
 - google-cloud-aiplatform 1.51.0 -> 1.56.0
 - bigframes 1.5.0 -> 1.8.0
 - regex 2023.12.25 -> 2024.5.15

2024-05-13

- Code actions are now supported to automatically improve and refactor code. Code actions can be triggered by the keyboard shortcut "Ctrl/⌘ + ."
- Python package upgrades
 - bigframes 1.0.0 -> 1.5.0
 - google-cloud-aiplatform 1.47.0 -> 1.51.0
 - jax[tpu] 0.4.23 -> 0.4.26
- Python package inclusions
 - cudf 24.4.1

2024-04-15

- TPU v2 runtime is now available
- L4 runtime is now available for paid users
- New distributed fine-tuning Gemma tutorial on TPUs ([GitHub](#))
- Symbol rename is now supported with keyboard shortcut F2
- Fixed bug causing inability to re-upload deleted files
- Fixed breaking bug in colabtools %upload_files_async
- Added syntax highlighting to %%writefile cells
- Cuda dependencies that come with Torch are cached for faster downloads for packages that require Torch and its dependencies ([GitHub issue](#))
- Python package upgrades
 - bigframes 0.24.0 -> 1.0.0
 - duckdb 0.9.2 -> 0.10.1
 - google-cloud-aiplatform 1.43.0 -> 1.47.0
 - jax 0.4.23 -> 0.4.26



Data Matrix

```
X = df.drop(columns=['am'])
X = X.drop(columns=['manufacturer'])

X = X.drop(columns=['drat'])
```

```
X.head(10)
```

	mpg	cyl	disp	hp	wt	qsec	vs	gear	carb	
0	21.0	6	160.0	110	2.620	16.46	0	4	4	
1	21.0	6	160.0	110	2.875	17.02	0	4	4	
2	22.8	4	108.0	93	2.320	18.61	1	4	1	
3	21.4	6	258.0	110	3.215	19.44	1	3	1	
4	18.7	8	360.0	175	3.440	17.02	0	3	2	
5	18.1	6	225.0	105	3.460	20.22	1	3	1	
6	14.3	8	360.0	245	3.570	15.84	0	3	4	
7	24.4	4	146.7	62	3.190	20.00	1	4	2	
8	22.8	4	140.8	95	3.150	22.90	1	4	2	
9	19.2	6	167.6	123	3.440	18.30	1	4	4	

Next steps:

[Generate code with X](#)

[View recommended plots](#)

[New](#)

2024-03-13

- Fixed bug that sometimes caused UserSecrets to move / disappear
- Improved messaging for mounting drive in an unsupported environment ([GitHub issue](#))
- Python package upgrades
 - torch 2.1.0 -> 2.2.1
 - torchaudio 2.1.0 -> 2.2.1
 - torchvision 0.16.0 -> 0.17.1
 - torchtext 0.16.0 -> 0.17.1
 - PyMC 5.7.2 -> 5.10.4
 - BigFrames 0.21.0 -> 0.24.0
 - google-cloud-aiplatform 1.42.1 -> 1.43.0
 - tornado 6.3.2 -> 6.3.3

2024-02-21

- Try out Gemma on [Colab!](#)
- Allow unicode in form text inputs
- Display documentation and link to source when displaying functions
- Display image-like ndarrays as images
- Improved UX around quick charts and execution error suggestions
- Released Marketplace image for the month of February ([GitHub issue](#))
- Python package upgrades
 - bigframes 0.19.2 -> 0.21.0
 - regex 2023.6.3 -> 2023.12.25
 - spacy 3.6.1 -> 3.7.4
 - beautifulsoup4 4.11.2 -> 4.12.3
 - tensorflow-probability 0.22.0 -> 0.23.0
 - google-cloud-language 2.9.1 -> 2.13.1
 - google-cloud-aiplatform 1.39.0 -> 1.42.1
 - transformers 4.35.2 -> 4.37.2
 - pyarrow 10.0.1 -> 14.0.2

2024-01-29

- New [Kaggle Notebooks <-> Colab updates!](#)! Now you can:
 - Import directly from Colab without having to download/re-upload
 - Upload via link, by pasting Google Drive or Colab URLs
 - Export & run Kaggle Notebooks on Colab with 1 click
- Try these notebooks that talk to Gemini:
 - [Gemini and Stable Diffusion](#)

```
X.info()
```

```
→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 9 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   mpg      32 non-null     float64
 1   cyl      32 non-null     int64  
 2   disp     32 non-null     float64
 3   hp       32 non-null     int64  
 4   wt       32 non-null     float64
 5   qsec     32 non-null     float64
 6   vs       32 non-null     int64  
 7   gear     32 non-null     int64  
 8   carb     32 non-null     int64  
dtypes: float64(4), int64(5)
memory usage: 2.4 KB
```

```
X.describe().transpose()
```

	count	mean	std	min	25%	max
mpg	32.0	20.090625	6.026948	10.400	15.42500	1
cyl	32.0	6.187500	1.785922	4.000	4.00000	8
disp	32.0	230.721875	123.938694	71.100	120.82500	19
hp	32.0	146.687500	68.562868	52.000	96.50000	12
wt	32.0	3.217250	0.978457	1.513	2.58125	5
qsec	32.0	17.848750	1.786943	14.500	16.89250	1
vs	32.0	0.437500	0.504016	0.000	0.00000	1
gear	32.0	3.687500	0.737804	3.000	3.00000	4
carb	32.0	2.812500	1.615200	1.000	2.00000	8

Standardize the Data

```
X.info()
```

```
→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 9 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   mpg      32 non-null     float64
 1   cyl      32 non-null     int64  
 2   disp     32 non-null     float64
 3   hp       32 non-null     int64  
 4   wt       32 non-null     float64
 5   qsec     32 non-null     float64
 6   vs       32 non-null     int64  
 7   gear     32 non-null     int64  
 8   carb     32 non-null     int64  
dtypes: float64(4), int64(5)
memory usage: 2.4 KB
```

- [Learning with Gemini and ChatGPT](#)
- [Talk to Gemini with Google's Speech to Text API](#)
- [Sell lemonade with Gemini and Sheets](#)
- [Generate images with Gemini and Vertex](#)

- Python package upgrades
 - google-cloud-aiplatform 1.38.1 -> 1.39.0
 - bigframes 0.18.0 -> 0.19.2
 - polars 0.17.3 -> 0.20.2
 - gdown 4.6.6 -> 4.7.3 ([GitHub issue](#))
 - tensorflow-hub 0.15.0 -> 0.16.0
 - flax 0.7.5 -> 0.8.0
- Python package inclusions
 - sentencepiece 0.1.99

2024-01-08

- Avoid nested scrollbars for large outputs by using [google.colab.output.no_vertic](#) [Example notebook](#)
- Fix [bug](#) where downloading models from Hugging Face could freeze
- Python package upgrades
 - huggingface-hub 0.19.4 -> 0.20.2
 - bigframes 0.17.0 -> 0.18.0

2023-12-18

- Expanded access to AI coding has arrived in Colab across 175 locales for all tiers of Colab users
- Improvements to display of ML-based inline completions (for eligible Pro/Pro+ users)
- Started a series of [notebooks](#) highlighting Gemini API capabilities
- Enable ⌘/Ctrl+L to select the full line in an editor
- Fixed [bug](#) where we weren't correctly formatting output from multiple execution results
- Python package upgrades

- CUDA 11.8 to CUDA 12.2
- tensorflow 2.14.0 -> 2.15.0
- tensorboard 2.14.0 -> 2.15.0
- keras 2.14.0 -> 2.15.0
- Nvidia drivers 525.105.17 -> 535.104.05
- tensorflow-gcs-config 2.14.0 -> 2.15.0

```

    7 gear      32 non-null      int64
    8 carb      32 non-null      int64
dtypes: float64(4), int64(5)
memory usage: 2.4 KB

```

```

Xs = StandardScaler().fit_transform(X)
Xcols = X.columns
X = pd.DataFrame(Xs)
X.columns = Xcols
X.head(10)

```

	mpg	cyl	disp	hp	wt	
0	0.153299	-0.106668	-0.579750	-0.543655	-0.620167	-0.7
1	0.153299	-0.106668	-0.579750	-0.543655	-0.355382	-0.4
2	0.456737	-1.244457	-1.006026	-0.795570	-0.931678	0.4
3	0.220730	-0.106668	0.223615	-0.543655	-0.002336	0.9
4	-0.234427	1.031121	1.059772	0.419550	0.231297	-0.4
5	-0.335572	-0.106668	-0.046906	-0.617748	0.252064	1.3
6	-0.976163	1.031121	1.059772	1.456847	0.366285	-1.1
7	0.726459	-1.244457	-0.688779	-1.254944	-0.028296	1.2
8	0.456737	-1.244457	-0.737144	-0.765933	-0.069830	2.8
9	-0.150138	-0.106668	-0.517448	-0.351014	0.231297	0.2

Next steps:

[Generate code with X](#)[View recommended plots](#)[New](#)

```
X.describe().transpose()
```

	count	mean	std	min	25%
mpg	32.0	-4.996004e-16	1.016001	-1.633610	-0.786514
cyl	32.0	2.775558e-17	1.016001	-1.244457	-1.244457
disp	32.0	1.665335e-16	1.016001	-1.308518	-0.900892
hp	32.0	0.000000e+00	1.016001	-1.403130	-0.743705
wt	32.0	-4.718448e-16	1.016001	-1.769642	-0.660403
qsec	32.0	-1.471046e-15	1.016001	-1.903996	-0.543694
vs	32.0	0.000000e+00	1.016001	-0.881917	-0.881917
---	---	-2.775558e-	-----	-----	-----

Observations and variables

- bigframes 0.13.0 -> 0.17.0
- geemap 0.28.2 -> 0.29.6
- pyarrow 9.0.0 -> 10.0.1
- google-generativeai 0.2.2 -> 0.3.1
- jax 0.4.20 -> 0.4.23
- jaxlib 0.4.20 -> 0.4.23
- Python package inclusions
 - kagglehub 0.1.4
 - google-cloud-aiplatform 1.38.1

2023-11-27

- Removed warning when calling await to make it render as code
- Added "Run selection" to the cell context menu
- Added highlighting for the %%python cell magic
- Launched AI coding features for Pro/Pro+ users in more locales
- Python package upgrades
 - bigframes 0.12.0 -> 0.13.0
- Python package inclusions
 - transformers 4.35.2
 - google-generativeai 0.2.2

2023-11-08

- Launched Secrets, for safe storage of private keys on Colab ([tweet](#))
- Fixed issue where TensorBoard would not load ([#3990](#))
- Python package upgrades
 - lightgbm 4.0.0 -> 4.1.0
 - bigframes 0.10.0 -> 0.12.0
 - bokeh 3.2.2 -> 3.3.0
 - duckdb 0.8.1 -> 0.9.1
 - numba 0.56.4 -> 0.58.1
 - tweepy 4.13.0 -> 4.14.0
 - jax 0.4.16 -> 0.4.20
 - jaxlib 0.4.16 -> 0.4.20

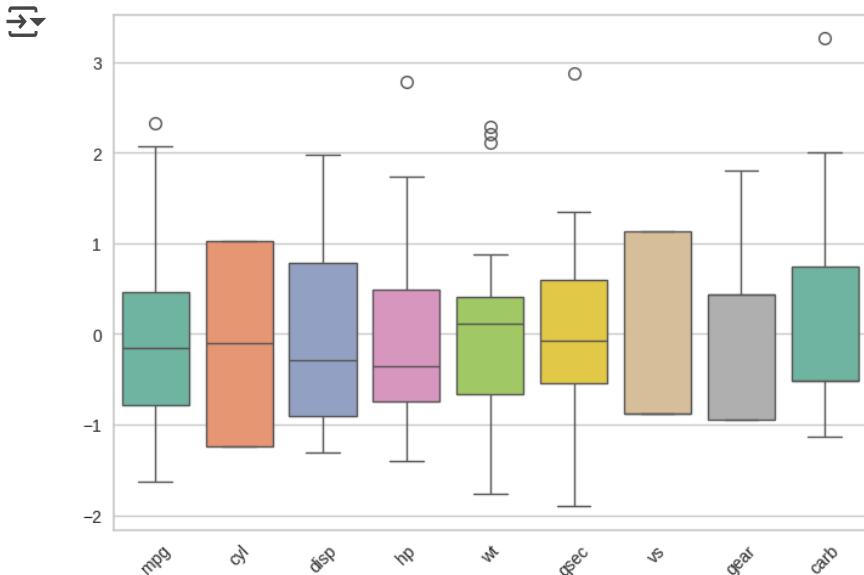
2023-10-23

- Updated the **Open notebook** dialog for better usability and support for smaller screen sizes
- Added smart paste support for data from Google Sheets for R notebooks
- Enabled showing release notes in a tab
- Launched AI coding features for Pro/Pro+ users in Australia 🇦🇺 Canada 🇨🇦 India 🇮🇳 and Japan 🇯🇵 ([tweet](#))
- Python package upgrades

```
observations = list(df.index)
variables = list(df.columns)
```

Box and Whisker Plots

```
ax = plt.figure()
ax = sns.boxplot(data=X, orient="v", palette="Set2")
ax.set_xticklabels(ax.get_xticklabels(), rotation=45);
```



```
# Use swarmplot() or stripplot to show the datapoints on
# plt. figure()
ax = plt.figure()
ax = sns.boxplot(data=X, orient="v", palette="Set2")
ax = sns.stripplot(data=X, color=".25")
ax.set_xticklabels(ax.get_xticklabels(), rotation=45);
```

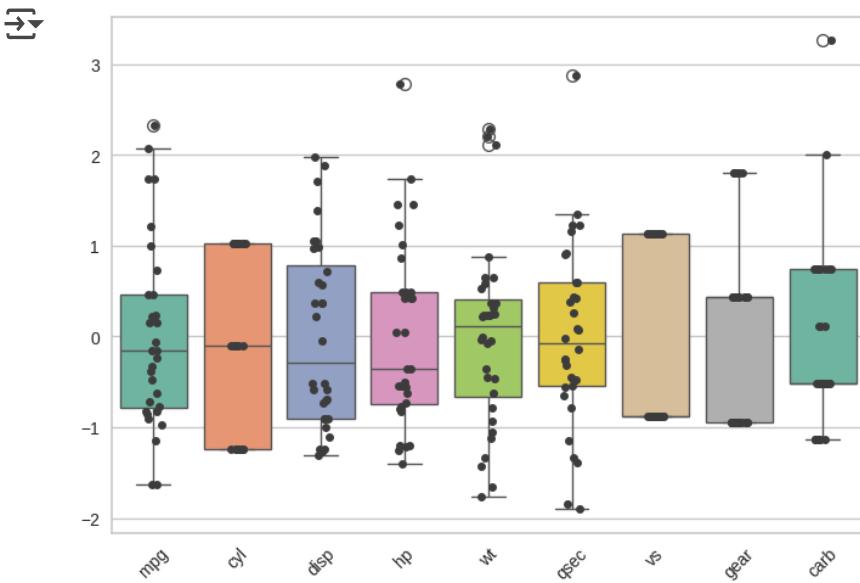
- earthengine-api 0.1.357 -> 0.1.375
- flax 0.7.2 -> 0.7.4
- geemap 0.27.4 -> 0.28.2
- jax 0.4.14 -> 0.4.16
- jaxlib 0.4.14 -> 0.4.16
- keras 2.13.1 -> 2.14.0
- tensorboard 2.13.0 -> 2.14.1
- tensorflow 2.13.0 -> 2.14.0
- tensorflow-gcs-config 2.13.0 -> 2.14.0
- tensorflow-hub 0.14.0 -> 0.15.0
- tensorflow-probability 0.20.1 -> 0.22.0
- torch 2.0.1 -> 2.1.0
- torchaudio 2.0.2 -> 2.1.0
- torchtext 0.15.2 -> 0.16.0
- torchvision 0.15.2 -> 0.16.0
- xgboost 1.7.6 -> 2.0.0
- Python package inclusions
 - bigframes 0.10.0
 - malloy 2023.1056

2023-09-22

- Added the ability to scope an AI generated suggestion to a specific Pandas dataframe ([tweet](#))
- Added Colab link previews to Docs ([tweet](#))
- Added smart paste support for data from Google Sheets
- Increased font size of dropdowns in interactive forms
- Improved rendering of the notebook when printing
- Python package upgrades
 - tensorflow 2.12.0 -> 2.13.0
 - tensorboard 2.12.3 -> 2.13.0
 - keras 2.12.0 -> 2.13.1
 - tensorflow-gcs-config 2.12.0 -> 2.13.
 - scipy 1.10.1 -> 1.11.2
 - cython 0.29.6 -> 3.0.2
- Python package inclusions
 - geemap 0.26.0

2023-08-18

- Added "Change runtime type" to the menu in the connection button
- Improved auto-reconnection to an already running notebook ([#3764](#))
- Increased the specs of our highmem machines for Pro users
- Fixed add-apt-repository command on Ubuntu 22.04 runtime ([#3867](#))
- Python package upgrades
 - bokeh 2.4.3 -> 3.2.2



- cmake 3.25.2 → 3.27.2
- cryptography 3.4.8 → 41.0.3
- dask 2022.12.1 → 2023.8.0
- distributed 2022.12.1 → 2023.8.0
- earthengine-api 0.1.358 → 0.1.364
- flax 0.7.0 → 0.7.2
- ipython-sql 0.4.0 → 0.5.0
- jax 0.4.13 → 0.4.14
- jaxlib 0.4.13 → 0.4.14
- lightgbm 3.3.5 → 4.0.0
- mkl 2019.0 → 2023.2.0
- notebook 6.4.8 → 6.5.5
- numpy 1.22.4 → 1.23.5
- opencv-python 4.7.0.72 → 4.8.0.76
- pillow 8.4.0 → 9.4.0
- plotly 5.13.1 → 5.15.0
- prettytable 0.7.2 → 3.8.0
- pytensor 2.10.1 → 2.14.2
- spacy 3.5.4 → 3.6.1
- statsmodels 0.13.5 → 0.14.0
- xarray 2022.12.0 → 2023.7.0
- Python package inclusions
 - PyDrive2 1.6.3

2023-07-21

- Launched auto-plotting for dataframes, available using the chart button that shows up alongside datatables ([post](#))



- Added a menu to the table of contents to support running a section or collapsing/expanding sections ([post](#))

The screenshot shows the Colab interface with the 'Table of contents' sidebar open. The sidebar lists several sections and options:

- Making the Most of your Colab Subscription
- Faster GPUs
- More memory
- Longer runtimes
- Relaxing resource limits in Colab Pro
- Send us feedback!

 On the right side of the sidebar, there are additional buttons for 'Run cells in s', 'Collapse', 'Expand', 'Delete cells', and 'Users upgrade'.

- Added an option to automatically run the first cell or section, available under Edit → Notebook settings ([post](#))

	mpg	cyl	disp	hp	wt	qsec	vs	gear	carb
mpg	1	-0.85	-0.85	-0.78	-0.87	0.42	0.66	0.48	-0.55
cyl	-0.85	1	0.9	0.83	0.78	-0.59	-0.81	-0.49	0.53
disp	-0.85	0.9	1	0.79	0.89	-0.43	-0.71	-0.56	0.39
hp	-0.78	0.83	0.79	1	0.66	-0.71	-0.72	-0.13	0.75
wt	-0.87	0.78	0.89	0.66	1	-0.17	-0.55	-0.58	0.43
qsec	0.42	-0.59	-0.43	-0.71	-0.17	1	0.74	-0.21	-0.66
vs	0.66	-0.81	-0.71	-0.72	-0.55	0.74	1	0.21	-0.57
gear	0.48	-0.49	-0.56	-0.13	-0.58	-0.21	0.21	1	0.27
carb	-0.55	0.53	0.39	0.75	0.43	-0.66	-0.57	0.27	1

Notebook settings

Runtime type
Python 3

Hardware accelerator
None

Automatically run the first cell or section
 Omit code cell output when saving this notebook

- Launched Pro/Pro+ to Algeria, Argentina, Chile, Ecuador, Egypt, Ghana, Kenya, Malaysia, Nepal, Nigeria, Peru, Rwanda, Saudi Arabia, South Africa, Sri Lanka, Tunisia, and Ukraine ([tweet](#))
- Added a command, "Toggle tab moves focus" for toggling tab trapping in the editor (Tools -> Command palette, "Toggle tab moves focus")
- Fixed issue where `files.upload()` was sometimes returning an incorrect filename ([#1550](#))
- Fixed f-string syntax highlighting bug ([#3802](#))
- Disabled ambiguous characters highlighting for commonly used LaTeX characters ([#3648](#))
- Upgraded Ubuntu from 20.04 LTS to [22.04 LTS](#)
- Updated the Colab Marketplace VM image
- Python package upgrades:
 - autograd 1.6.1 -> 1.6.2
 - drivefs 76.0 -> 77.0
 - flax 0.6.11 -> 0.7.0
 - earthengine-api 0.1.357 -> 0.1.358
 - GDAL 3.3.2->3.4.3
 - google-cloud-bigquery-storage 2.20.0 -> 2.22.2
 - gspread-dataframe 3.0.8 -> 3.3.1
 - holidays 0.27.1 -> 0.29
 - jax 0.4.10 -> jax 0.4.13
 - jaxlib 0.4.10 -> jax 0.4.13
 - jupyterlab-widgets 3.0.7 -> 3.0.8
 - nbformat 5.9.0 -> 5.9.1
 - opencv-python-headless 4.7.0.72 -> 4.8.0.74
 - pygame 2.4.0 -> 2.5.0
 - spacy 3.5.3 -> 3.5.4
 - SQLAlchemy 2.0.16 -> 2.0.19
 - tabulate 0.8.10 -> 0.9.0
 - tensorflow-hub 0.13.0 -> 0.14.0

v Applying PCA

```
pca = PCA()
Z = pca.fit_transform(X)

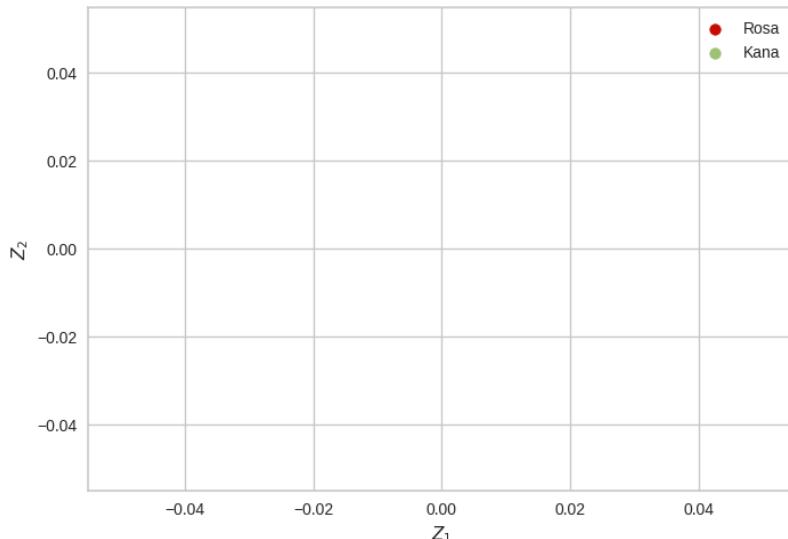
# 1(Rosa)->0, 2(Kama)->1, 3(Canadian)->2
idx_Rosa = np.where(y == 0)
idx_Kana = np.where(y == 1)

plt.figure()
plt.scatter(Z[idx_Rosa,0], Z[idx_Rosa,1], c='r', label=
plt.scatter(Z[idx_Kana,0], Z[idx_Kana,1], c='g', label=

plt.legend()
plt.xlabel('$Z_1$')
plt.ylabel('$Z_2$')
```

2023-06-23

Text(0, 0.5, '\$Z_2\$')



Eigenvectors

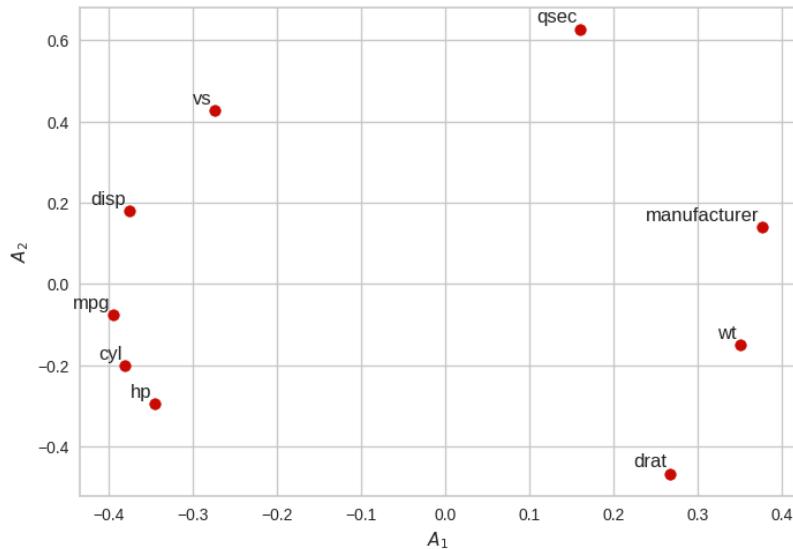
```
A = pca.components_.T

plt.scatter(A[:,0],A[:,1],c='r')
plt.xlabel('$A_1$')
plt.ylabel('$A_2$')
for label, x, y in zip(variables, A[:, 0], A[:, 1]):
    plt.annotate(label, xy=(x, y), xytext=(-2, 2), textcolor='black')
```

- Launched AI coding features to subscribed users starting with Pro+ users in the US ([tweet](#), [post](#))
- Added the Kernel Selector in the Notebook Settings ([tweet](#))
- Fixed double space trimming issue in markdown [#3766](#)
- Fixed run button indicator not always centered [#3609](#)
- Fixed inconsistencies for automatic indentation on multi-line [#3697](#)
- Upgraded Python from 3.10.11 to 3.10.12
- Python package updates:
 - duckdb 0.7.1 -> 0.8.1
 - earthengine-api 0.1.350 -> 0.1.357
 - flax 0.6.9 -> 0.6.11
 - google-cloud-bigquery 3.9.0 -> 3.10.0
 - google-cloud-bigquery-storage 2.19.1 -> 2.20.0
 - grpcio 1.54.0 -> 1.56.0
 - holidays 0.25 -> 0.27.1
 - nbformat 5.8.0 -> 5.9.0
 - prophet 1.1.3 -> 1.1.4
 - pydata-google-auth 1.7.0 -> 1.8.0
 - spacy 3.5.2 -> 3.5.3
 - tensorboard 2.12.2 -> 2.12.3
 - xgboost 1.7.5 -> 1.7.6
- Python package inclusions:
 - gcsfs 2023.6.0
 - geopandas 0.13.2
 - google-cloud-bigquery-connection 1.12.0
 - google-cloud-functions 1.13.0
 - grpc-google-iam-v1 0.12.6
 - multidict 6.0.4
 - tensorboard-data-server 0.7.1

2023-06-02

- Released the new site [colab.google](#)
- Published Colab's Docker runtime image to us-docker.pkg.dev/colab-images/public/runtime ([tweet](#), [instructions](#))
- Launched support for Google children accounts ([tweet](#))
- Launched DagsHub integration ([tweet](#), [post](#))
- Upgraded to Monaco Editor Version 0.37.1
- Fixed various Vim keybinding bugs
- Fixed issue where the N and P letters sometimes couldn't be typed ([#3664](#))
- Fixed rendering support for compositional inputs ([#3660](#), [#3679](#))



```
plt.scatter(A[:, 0], A[:, 1], marker='o', c=A[:, 2], s=100)
plt.xlabel('$A_1$')
plt.ylabel('$A_2$')
for label, x, y in zip(variables, A[:, 0], A[:, 1]):
    plt.annotate(label, xy=(x, y), xytext=(-20, 20),
                textcoords='offset points', ha='right', va='bottom',
                bbox=dict(boxstyle='round,pad=0.5', fc='yellow', ec='black'),
                arrowprops=dict(arrowstyle = '->', connectionstyle='angle,angleB=90,rad=12'))
```

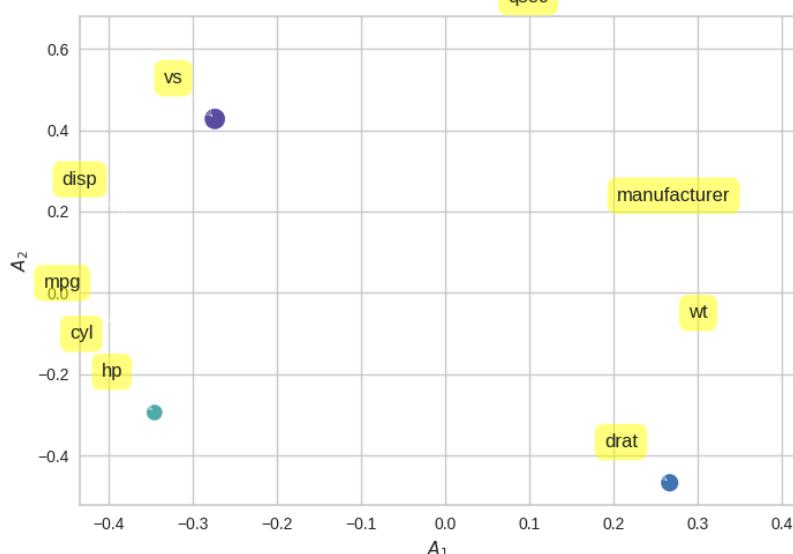
- Fixed lag in notebooks with lots of cells ([#3676](#))
- Improved support for R by adding a Runtime type notebook setting (Edit -> Notebook settings)
- Improved documentation for connecting to a local runtime (Connect -> Connect to a local runtime)
- Python package updates:
 - holidays 0.23 -> 0.25
 - jax 0.4.8 -> 0.4.10
 - jaxlib 0.4.8 -> 0.4.10
 - pip 23.0.1 -> 23.1.2
 - tensorflow-probability 0.19.0 -> 0.20.1
 - torch 2.0.0 -> 2.0.1
 - torchaudio 2.0.1 -> 2.0.2
 - torchdata 0.6.0 -> 0.6.1
 - torchtext 0.15.1 -> 0.15.2
 - torchvision 0.15.1 -> 0.15.2
 - tornado 6.2 -> 6.3.1

2023-05-05

- Released GPU type selection for paid users, allowing them to choose a preferred NVidia GPU
- Upgraded R from 4.2.3 to 4.3.0
- Upgraded Python from 3.9.16 to 3.10.11
- Python package updates:
 - attrs 22.2.0 -> attrs 23.1.0
 - earthengine-api 0.1.349 -> earthengine-api 0.1.350
 - flax 0.6.8 -> 0.6.9
 - grpcio 1.53.0 -> 1.54.0
 - nbclient 0.7.3 -> 0.7.4
 - tensorflow-datasets 4.8.3 -> 4.9.2
 - termcolor 2.2.0 -> 2.3.0
 - zict 2.2.0 -> 3.0.0

2023-04-14

- Python package updates:
 - google-api-python-client 2.70.0 -> 2.84.0
 - google-auth-oauthlib 0.4.6 -> 1.0.0
 - google-cloud-bigquery 3.4.2 -> 3.9.0
 - google-cloud-datastore 2.11.1 -> 2.15.1
 - google-cloud-firebase 2.7.3 -> 2.11.0
 - google-cloud-language 2.6.1 -> 2.9.1
 - google-cloud-storage 2.7.0 -> 2.8.0



Scree plot

```
#Eigenvalues
Lambda = pca.explained_variance_

#Scree plot
x = np.arange(len(Lambda)) + 1
plt.plot(x,Lambda/sum(Lambda), 'ro-', lw=3)
plt.xticks(x, [""+str(i) for i in x], rotation=0)
plt.xlabel('Number of components')
plt.ylabel('Explained variance')
```

- google-cloud-translate 3.8.4 -> 3.11.1
- networkx 3.0 -> 3.1
- notebook 6.3.0 -> 6.4.8
- jax 0.4.7 -> 0.4.8
- pandas 1.4.4 -> 1.5.3
- spacy 3.5.1 -> 3.5.2
- SQLAlchemy 1.4.47 -> 2.0.9
- xgboost 1.7.4 -> 1.7.5

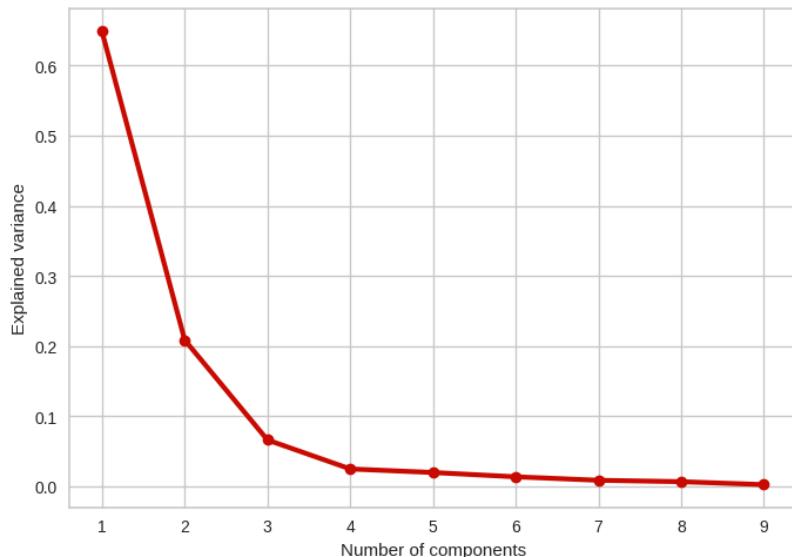
2023-03-31

- Improve bash ! syntax highlighting ([GitHub issue](#))
- Fix bug where VIM keybindings weren't working in the file editor
- Upgraded R from 4.2.2 to 4.2.3
- Python package updates:
 - arviz 0.12.1 --> 0.15.1
 - astropy 4.3.1 --> 5.2.2
 - dopamine-rl 1.0.5 --> 4.0.6
 - gensim 3.6.0 --> 4.3.1
 - ipykernel 5.3.4 -> 5.5.6
 - ipython 7.9.0 -> 7.34.0
 - jax 0.4.4 -> 0.4.7
 - jaxlib 0.4.4 -> 0.4.7
 - jupyter_core 5.2.0 -> 5.3.0
 - keras 2.11.0 -> 2.12.0
 - lightgbm 2.2.3 -> 3.3.5
 - matplotlib 3.5.3 -> 3.7.1
 - nltk 3.7 -> 3.8.1
 - opencv-python 4.6.0.66 -> 4.7.0.72
 - plotly 5.5.0 -> 5.13.1
 - pymc 4.1.4 -> 5.1.2
 - seaborn 0.11.2 -> 0.12.2
 - spacy 3.4.4 -> 3.5.1
 - sympy 1.7.1 -> 1.11.1
 - tensorboard 2.11.2 -> 2.12.0
 - tensorflow 2.11.0 -> 2.12.0
 - tensorflow-estimator 2.11.0 -> 2.12.0
 - tensorflow-hub 0.12.0 -> 0.13.0
 - torch 1.13.1 -> 2.0.0
 - torchaudio 0.13.1 -> 2.0.1
 - torchtext 0.14.1 -> 0.15.1
 - torchvision 0.14.1 -> 0.15.1

2023-03-10

- Added the [Colab editor shortcuts](#) example notebook
- Fixed triggering of @-mention and email autocomplete for large comments ([GitHub issue](#))
- Added View Resources to the Runtime menu
- Made file viewer images fit the view by default, resizing to original size on click

Text(0, 0.5, 'Explained variance')



Explained Variance

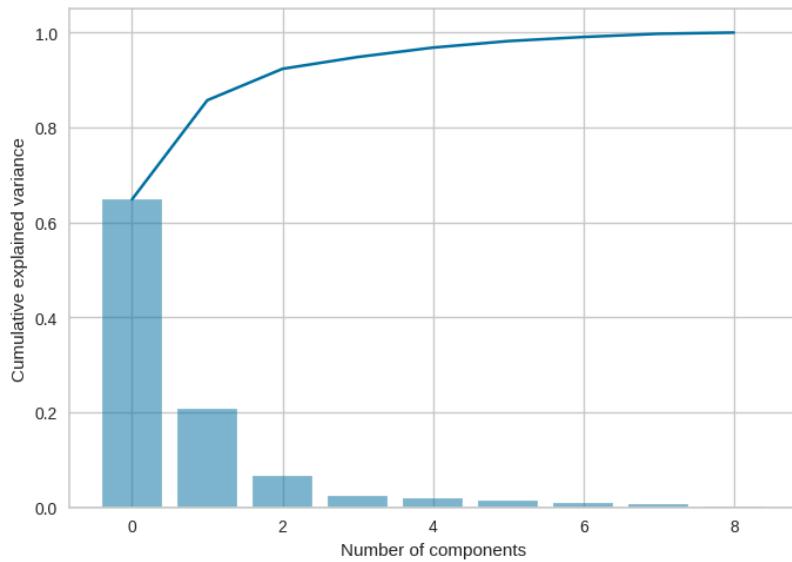
```
ell = pca.explained_variance_ratio_
ind = np.arange(len(ell))
plt.bar(ind, ell, align='center', alpha=0.5)
plt.plot(np.cumsum(ell))
plt.xlabel('Number of components')
plt.ylabel('Cumulative explained variance')
```

- When in VIM mode, enable copy as well as allowing propagation to monaco-vim to escape visual mode ([GitHub issue](#))
- Upgraded CUDA 11.6.2 -> 11.8.0 and cuDNN 8.4.0.27 -> 8.7.0.84
- Upgraded Nvidia drivers 525.78.01 -> 530.30.02
- Upgraded Python 3.8.10 -> 3.9.16
- Python package updates:
 - beautifulsoup4 4.6.3 -> 4.9.3
 - bokeh 2.3.3 -> 2.4.3
 - debugpy 1.0.0 -> 1.6.6
 - Flask 1.1.4 -> 2.2.3
 - jax 0.3.25 -> 0.4.4
 - jaxlib 0.3.25 -> 0.4.4
 - Jinja2 2.11.3 -> 3.1.2
 - matplotlib 3.2.2 -> 3.5.3
 - nbconvert 5.6.1 -> 6.5.4
 - pandas 1.3.5 -> 1.4.4
 - pandas-datareader 0.9.0 -> 0.10.0
 - pandas-profiling 1.4.1 -> 3.2.0
 - Pillow 7.1.2 -> 8.4.0
 - plotnine 0.8.0 -> 0.10.1
 - scikit-image 0.18.3 -> 0.19.3
 - scikit-learn 1.0.2 -> 1.2.2
 - scipy 1.7.3 -> 1.10.1
 - setuptools 57.4.0 -> 63.4.3
 - sklearn-pandas 1.8.0 -> 2.2.0
 - statsmodels 0.12.2 -> 0.13.5
 - urllib3 1.24.3 -> 1.26.14
 - Werkzeug 1.0.1 -> 2.2.3
 - wrapt 1.14.1 -> 1.15.0
 - xgboost 0.90 -> 1.7.4
 - xlrd 1.2.0 -> 2.0.1

2023-02-17

- Show graphs of RAM and disk usage in notebook toolbar
- Copy cell links directly to the clipboard instead of showing a dialog when clicking on the link icon in the cell toolbar
- Updated the [Colab Marketplace VM image](#)
- Upgraded CUDA to 11.6.2 and cuDNN to 8.4.0.27
- Python package updates:
 - tensorflow 2.9.2 -> 2.11.0
 - tensorboard 2.9.1 -> 2.11.2
 - keras 2.9.0 -> 2.11.0
 - tensorflow-estimator 2.9.0 -> 2.11.0
 - tensorflow-probability 0.17.0 -> 0.19.0
 - tensorflow-gcs-config 2.9.0 -> 2.11.0

Text(0, 0.5, 'Cumulative explained variance')



Biplot

```
# 0,1 denote PC1 and PC2; change values for other PCs
A1 = A[:,0]
A2 = A[:,1]
Z1 = Z[:,0]
Z2 = Z[:,1]

plt.figure()
plt.xlabel('$Z_1$')
plt.ylabel('$Z_2$')
for i in range(len(A1)):
    # arrows project features as vectors onto PC axes
    plt.arrow(0, 0, A1[i]*max(Z1), A2[i]*max(Z2), color='k')
    plt.text(A1[i]*max(Z1)*1.2, A2[i]*max(Z2)*1.2, variable)

plt.scatter(Z[idx_Rosa,0], Z[idx_Rosa,1], c='r', label='Rosa')
plt.scatter(Z[idx_Kana,0], Z[idx_Kana,1], c='g', label='Kana')

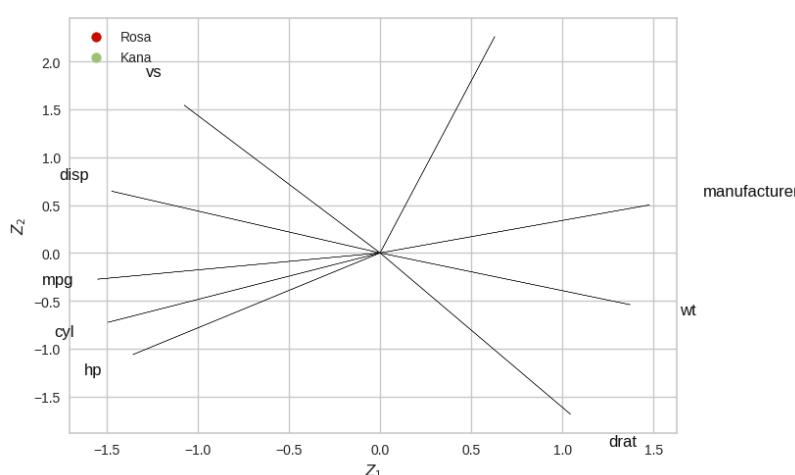
plt.legend(loc='upper left')
```

- earthengine-api 0.1.339 -> 0.1.341
- flatbuffers 1.12 -> 23.1.21
- platformdirs 2.6.2 -> 3.0.0
- pydata-google-auth 1.6.0 -> 1.7.0
- python-utils 3.4.5 -> 3.5.2
- tenacity 8.1.0 -> 8.2.1
- tifffile 2023.1.23.1 -> 2023.2.3
- notebook 5.7.16 -> 6.3.0
- tornado 6.0.4 -> 6.2
- aiohttp 3.8.3 -> 3.8.4
- charset-normalizer 2.1.1 -> 3.0.1
- fastai 2.7.0 -> 2.7.1
- soundfile 0.11.0 -> 0.12.1
- typing-extensions 4.4.0 -> 4.5.0
- widgetsnbextension 3.6.1 -> 3.6.2
- pydantic 1.10.4 -> 1.10.5
- zipp 3.12.0 -> 3.13.0
- numpy 1.21.6 -> 1.22.4
- drivefs 66.0 -> 69.0
- gdal 3.0.4 -> 3.3.2 [GitHub issue](#)
- Added libudunits2-dev for smoother R package installs [GitHub issue](#)

2023-02-03

- Improved tooltips for pandas series to show common statistics about the series object
- Made the forms dropdown behave like an autocomplete box when it allows input
- Updated the nvidia driver from 460.32.03 to 510.47.03
- Python package updates:
 - absl-py 1.3.0 -> 1.4.0
 - bleach 5.0.1 -> 6.0.0
 - cachetools 5.2.1 -> 5.3.0
 - cmdstanpy 1.0.8 -> 1.1.0
 - dnspython 2.2.1 -> 2.3.0
 - fsspec 2022.11.0 -> 2023.1.0
 - google-cloud-bigquery-storage 2.17.0 -> 2.18.1
 - holidays 0.18 -> 0.19
 - jupyter-core 5.1.3 -> 5.2.0
 - packaging 21.3 -> 23.0
 - prometheus-client 0.15.0 -> 0.16.0
 - pyct 0.4.8 -> 0.5.0
 - pydata-google-auth 1.5.0 -> 1.6.0
 - python-slugify 7.0.0 -> 8.0.0
 - sqlalchemy 1.4.46 -> 2.0.0
 - tensorflow-io-gcs-filesystem 0.29.0 -> 0.30.0
 - tifffile 2022.10.10 -> 2023.1.23.1
 - zipp 3.11.0 -> 3.12.0

→ <matplotlib.legend.Legend at 0x7e35b45c1f50>



Using PCA Library

```
!pip install pca
```

→ Requirement already satisfied: pca in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: datasets in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: statsmodels in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: matplotlib in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: numpy in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: scipy in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: colourmap>=1.1.19 in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: pandas in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: scatterd>=1.3.7 in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: adjusttext in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: seaborn in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: requests in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.8/site-packages
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.8/site-packages

- Pinned sqlalchemy to version 1.4.46

2023-01-12

- Added support for @-mention and email autocomplete in comments
- Improved errors when GitHub notebooks can't be loaded
- Increased color contrast for colors used for syntax highlighting in the code editor
- Added terminal access for custom GCE VM runtimes
- Upgraded Ubuntu from 18.04 LTS to 20.04 LTS ([GitHub issue](#))
- Python package updates:
 - GDAL 2.2.2 -> 2.2.3.
 - NumPy from 1.21.5 to 1.21.6.
 - attrs 22.1.0 -> 22.2.0
 - chardet 3.0.4 -> 4.0.0
 - cloudpickle 1.6.0 -> 2.2.0
 - filelock 3.8.2 -> 3.9.0
 - google-api-core 2.8.2 -> 2.11.0
 - google-api-python-client 1.12.11 -> 2.70.0
 - google-auth-httplib2 0.0.3 -> 0.1.0
 - google-cloud-bigquery 3.3.5 -> 3.4.1
 - google-cloud-datastore 2.9.0 -> 2.11.0
 - google-cloud-firebase 2.7.2 -> 2.7.3
 - google-cloud-storage 2.5.0 -> 2.7.0
 - holidays 0.17.2 -> holidays 0.18
 - importlib-metadata 5.2.0 -> 6.0.0
 - networkx 2.8.8 -> 3.0
 - opencv-python-headless 4.6.0.66 -> 4.7.0.68
 - pip 21.1.3 -> 22.0.4
 - pip-tools 6.2.0 -> 6.6.2
 - prettytable 3.5.0 -> 3.6.0
 - requests 2.23.0 -> 2.25.1
 - termcolor 2.1.1 -> 2.2.0
 - torch 1.13.0 -> 1.13.1
 - torchaudio 0.13.0 -> 0.13.1
 - torchtext 0.14.0 -> 0.14.1
 - torchvision 0.14.0 -> 0.14.1

2022-12-06

- Made fallback runtime version available until mid-December ([GitHub issue](#))
- Upgraded to Python 3.8 ([GitHub issue](#))
- Python package updates:

```
Requirement already satisfied: tzdata>=2022.1 in /usr
Requirement already satisfied: joblib>=1.2.0 in /usr
Requirement already satisfied: threadpoolctl>=2.0.0
Requirement already satisfied: patsy>=0.5.6 in /usr/
Requirement already satisfied: six>=1.5 in /usr/locat
Requirement already satisfied: charset-normalizer<4,
Requirement already satisfied: idna<4,>=2.5 in /usr/
Requirement already satisfied: urllib3<3,>=1.21.1 in
Requirement already satisfied: certifi>=2017.4.17 in
```

```
from pca import pca
# Initialize and keep all PCs
model = pca(n_components=0.85)
# Fit transform
out = model.fit_transform(X)
```

```
[pca] >Extracting column labels from dataframe.
[pca] >Extracting row labels from dataframe.
[pca] >The PCA reduction is performed to capture [85
[pca] >Fit using PCA.
[pca] >Compute loadings and PCs.
[pca] >Compute explained variance.
[pca] >Number of components is [2] that covers the [
[pca] >The PCA reduction is performed on the [9] col
[pca] >Fit using PCA.
[pca] >Compute loadings and PCs.
[pca] >Outlier detection using Hotelling T2 test wit
[pca] >Multiple test correction applied for Hotellin
[pca] >Outlier detection using SPE/DmodX with n_std=
```

Principal Components

```
out['PC']
```

- jax from 0.3.23 to 0.3.25, jaxlib from 0.3.22 to 0.3.25
- pyarrow from 6.0.1 to 9.0.0
- torch from 1.12.1 to 1.13.0
- torchaudio from 0.12.1 to 0.13.0
- torchvision from 0.13.1 to 0.14.0
- torchtext from 0.13.1 to 0.14.0
- xlrd from 1.1.0 to 1.2.0
- DriveFS from 62.0.1 to 66.0.3
- Made styling of markdown tables in outputs match markdown tables in text cells
- Improved formatting for empty interactive table rows
- Fixed syntax highlighting for variables with names that contain Python keywords ([GitHub issue](#))

2022-11-11

- Added more dark editor themes for Monaco (when in dark mode, "Editor colorization" appears as an option in the Editor tab of the Tools → Settings dialog)
- Fixed bug where collapsed forms were deleted on mobile ([GitHub issue](#))
- Python package updates:
 - rpy2 from 3.4.0 to 3.5.5 ([GitHub issue](#))
 - notebook from 5.5.0 to 5.7.16
 - tornado from 5.1.1 to 6.0.4
 - tensorflow_probability from 0.16.0 to 0.17.0
 - pandas-gbq from 0.13.3 to 0.17.9
 - protobuf from 3.17.3 to 3.19.6
 - google-api-core[grpc] from 1.31.5 to 2.8.2
 - google-cloud-bigquery from 1.21.0 to 3.3.5
 - google-cloud-core from 1.0.1 to 2.3.2
 - google-cloud-datastore from 1.8.0 to 2.9.0
 - google-cloud-firebase from 1.7.0 to 2.7.2
 - google-cloud-language from 1.2.0 to 2.6.1
 - google-cloud-storage from 1.18.0 to 2.5.0
 - google-cloud-translate from 1.5.0 to 3.8.4

2022-10-21

- Launched a single-click way to get from BigQuery to Colab to further

	PC1	PC2	
0	0.084886	1.321054	
1	0.078150	1.094447	
2	2.563973	-0.097727	
3	1.044496	-1.775744	
4	-1.584346	-0.776611	
5	0.995837	-2.093867	
6	-2.824979	0.117404	
7	2.443099	-0.572019	
8	2.630059	-1.270134	
9	0.564303	0.259885	
10	0.566279	0.067798	
11	-1.836838	-0.702951	
12	-1.627074	-0.631099	
13	-1.717906	-0.801747	
14	-3.479195	-1.345825	
15	-3.584249	-1.310257	
16	-3.364238	-1.006577	
17	3.589880	-0.087944	
18	3.443758	0.533082	
19	3.910968	-0.045611	
20	2.360222	-1.416411	
21	-1.568809	-0.833202	
22	-1.448393	-0.905373	
23	-3.019585	0.142137	
24	-1.818518	-0.962487	
25	3.273493	0.024487	
26	1.858097	1.947506	
27	3.055423	1.986919	
28	-2.424407	2.421461	
29	-0.636557	3.097457	
30	-3.533605	3.600372	
31	2.005774	0.021577	

explore query results
([announcement](#))

- Launched [Pro, Pro+, and Pay As You Go](#) to 19 additional countries: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, Finland, Greece, Hungary, Latvia, Lithuania, Norway, Portugal, Romania, Slovakia, Slovenia, and Sweden ([tweet](#))
- Updated jax from 0.3.17 to 0.3.23, jaxlib from 0.3.15 to 0.3.22, TensorFlow from 2.8.2 to 2.9.2, CUDA from 11.1 to 11.2, and cuDNN from 8.0 to 8.1 ([backend-info](#))
- Added a readonly option to [drive.mount](#)
- Fixed bug where Xarray was not working ([GitHub issue](#))
- Modified Markdown parsing to ignore block quote symbol within MathJax ([GitHub issue](#))

2022-09-30

- Launched [Pay As You Go](#), allowing premium GPU access without requiring a subscription
- Added vim and tcllib to our runtime image
- Fixed bug where open files were closed on kernel disconnect ([GitHub issue](#))
- Fixed bug where the play button/execution indicator was not clickable when scrolled into the cell output ([GitHub issue](#))
- Updated the styling for form titles so that they avoid obscuring the code editor
- Created a GitHub repo, [backend-info](#), with the latest apt-list.txt and pip-freeze.txt files for the Colab runtime ([GitHub issue](#))
- Added [files.upload_file\(filename\)](#) to upload a file from the browser to the runtime with a specified filename

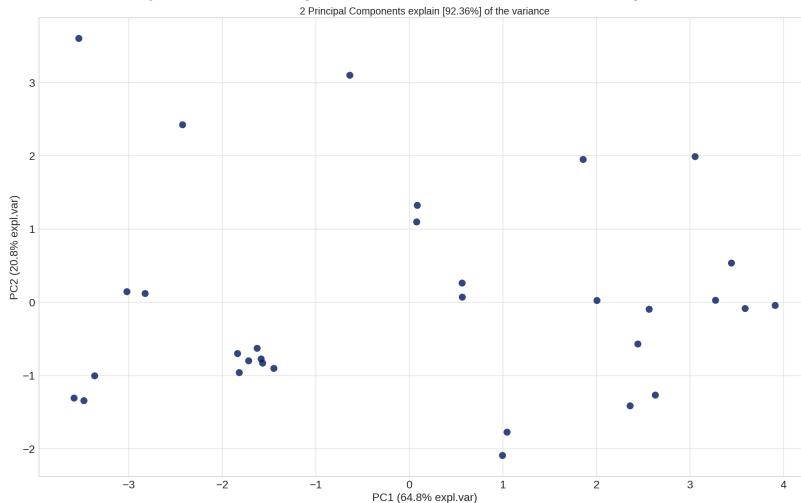
2022-09-16

- Upgraded pymc from 3.11.0 to 4.1.4, jax from 0.3.14 to 0.3.17, jaxlib from 0.3.14 to 0.3.15, fsspec from 2022.8.1 to 2022.8.2
- Modified our save flow to avoid persisting Drive filenames as titles in notebook JSON
- Updated our [Terms of Service](#)
- Modified the [Jump to Cell](#) command to locate the cursor at the end of the command palette input ([Jump to cell](#) in Tools →

Scatter plot

```
model.scatter(label=True, legend=False)
```

→ [scatterd] >INFO> Create scatterplot
 [pca]> [WARNING]: De parameter <label> is deprecated
 [scatterd]> WARNING use the standardized verbose sta
 (<Figure size 2500x1500 with 1 Axes>,
 <Axes: title={'center': '2 Principal Components
 explain [92.36%] of the variance'}, xlabel='PC1
 (64.8% expl.var)', ylabel='PC2 (20.8% expl.var)'>)



Eigenvectors

```
A = out['loadings'].T
```

```
sns.scatterplot(data=A, x="PC1", y="PC2")
plt.xlabel('$A_1$')
plt.ylabel('$A_2$')
for i in range(A.shape[0]):
    plt.text(x=A.PC1[i]+0.02,y=A.PC2[i]+0.02, s=variables[i]
              fontdict=dict(color='red',size=10),
              bbox=dict(facecolor='yellow',alpha=0.5))
```

Command palette in a notebook with section headings)

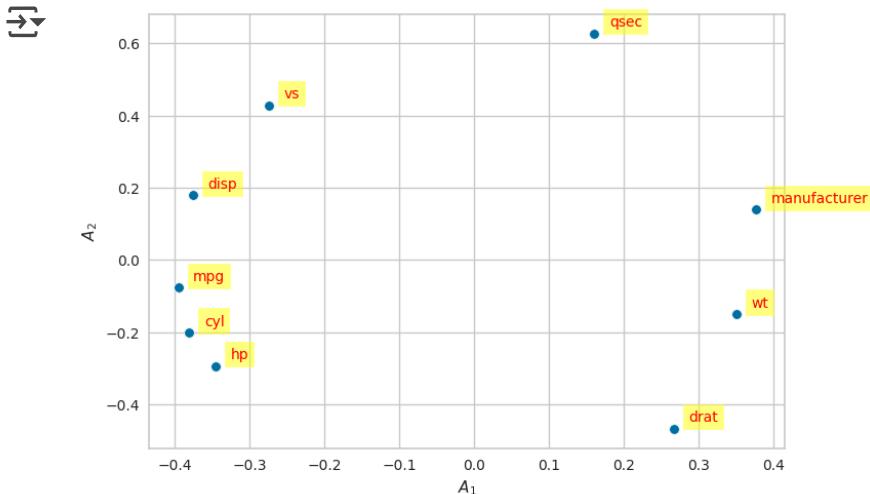
- Updated the styling of the Drive notebook comment UI
- Added support for terminating your runtime from code: `python from google.colab import runtime runtime.unassign()`
- Added regex filter support to the Recent notebooks dialog
- Inline `google.colab.files.upload` JS to fix `files.upload()` not working ([GitHub issue](#))

2022-08-26

- Upgraded PyYAML from 3.13 to 6.0 ([GitHub issue](#)), drivefs from 61.0.3 to 62.0.1
- Upgraded TensorFlow from 2.8.2 to 2.9.1 and ipywidgets from 7.7.1 to 8.0.1 but rolled both back due to a number of user reports ([GitHub issue](#), [GitHub issue](#))
- Stop persisting inferred titles in notebook JSON ([GitHub issue](#))
- Fix bug in background execution which affected some Pro+ users ([GitHub issue](#))
- Fix bug where Download as .py incorrectly handled text cells ending in a double quote
- Fix bug for Pro and Pro+ users where we weren't honoring the preference (Tools → Settings) to use a temporary scratch notebook as the default landing page
- Provide undo/redo for scratch cells
- When writing ipynb files, serialize empty multiline strings as [] for better consistency with JupyterLab

2022-08-11

- Upgraded ipython from 5.5.0 to 7.9.0, fbprophet 0.7 to prophet 1.1, tensorflow-datasets from 4.0.1 to 4.6.0, drivefs from 60.0.2 to 61.0.3, pytorch from 1.12.0 to 1.12.1, numba from 0.51 to 0.56, and lxml from 4.2.0 to 4.9.1
- Loosened our requests version requirement ([GitHub issue](#))
- Removed support for TensorFlow 1
- Added Help → Report Drive abuse for Drive notebooks
- Fixed indentation for Python lines ending in [
- Modified styling of tables in Markdown to left-align them rather than centering them



Scree Plot

```
VR = out['variance_ratio']
x = np.arange(len(VR)) + 1
plt.plot(x, VR, 'ro-', lw=3)
plt.xticks(x, [""+str(i) for i in x], rotation=0)
plt.xlabel('Number of components')
plt.ylabel('Explained variance')
plt.show()
```

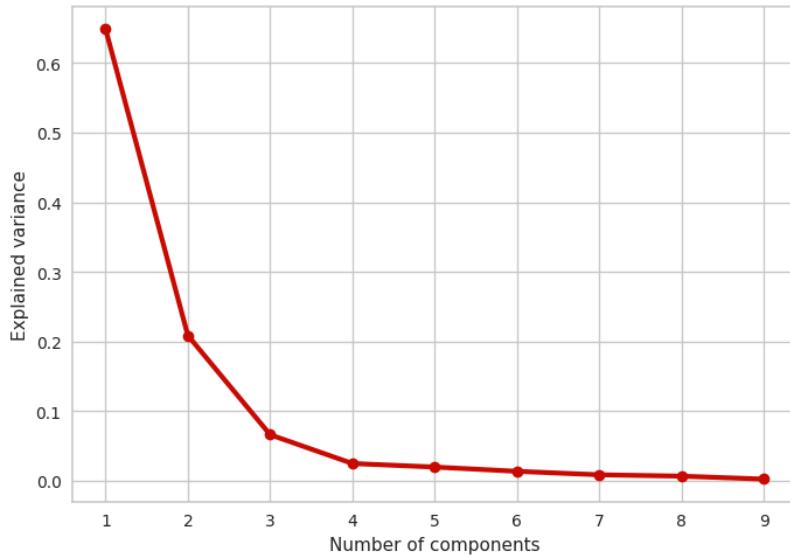
- Fixed special character replacement when copying interactive tables as Markdown
- Fixed ansi 8-bit color parsing ([GitHub issue](#))
- Configured logging to preempt transitive imports and other loading from implicitly configuring the root logger
- Modified forms to use a value of None instead of causing a parse error when clearing raw and numeric-typed form fields

2022-07-22

- Update scipy from 1.4.1 to 1.7.3, drivefs from 59.0.3 to 60.0.2, pytorch from 1.11 to 1.12, jax & jaxlib from 0.3.8 to 0.3.14, opencv-python from 4.1.2.30 to 4.6.0.66, spaCy from 3.3.1 to 3.4.0, and dlib from 19.18.0 to 19.24.0
- Fix Open in tab doc link which was rendering incorrectly ([GitHub issue](#))
- Add a preference for the default tab orientation to the Site section of the settings menu under Tools → Settings
- Show a warning for USE_AUTH_EPHEM usage when running authenticate_user on a TPU runtime ([code](#))

2022-07-01

- Add a preference for code font to the settings menu under Tools → Settings
- Update drivefs from 58.0.3 to 59.0.3 and spacy from 2.2.4 to 3.3.1
- Allow [display_data](#) and [execute_result](#) text outputs to wrap, matching behavior of JupyterLab (does not affect stream outputs/print statements).
- Improve LSP handling of some magics, esp. %%writefile ([GitHub issue](#)).
- Add a [FAQ entry](#) about the mount Drive button behavior and include link buttons for each FAQ entry.
- Fix bug where the notebook was sometimes hidden behind other tabs on load when in single pane view.
- Fix issue with inconsistent scrolling when an editor is in multi-select mode.
- Fix bug where clicking on a link in a form would navigate away from the notebook



Explained Variance Plot

```
model.plot();
```

- Show a confirmation dialog before performing Replace all from the Find and replace pane.

2022-06-10

- Update drivefs from 57.0.5 to 58.0.3 and tensorflow from 2.8.0 to 2.8.2
- Support more than 100 repos in the GitHub repo selector shown in the open dialog and the clone to GitHub dialog
- Show full notebook names on hover in the open dialog
- Improve the color contrast for links, buttons, and the `ipywidgets.Accordion` widget in dark mode

2022-05-20

- Support URL params for linking to some common pref settings: `force_theme=dark`, `force_corgi_mode=1`, `force_font_size=14`. Params forced by URL are not persisted unless saved using Tools → Settings.
- Add a class `markdown-google-sans` to allow Markdown to render in Google Sans
- Update monaco-vim from 0.1.19 to 0.3.4
- Update drivefs from 55.0.3 to 57.0.5, jax from 0.3.4 to 0.3.8, and jaxlib from 0.3.2 to 0.3.7

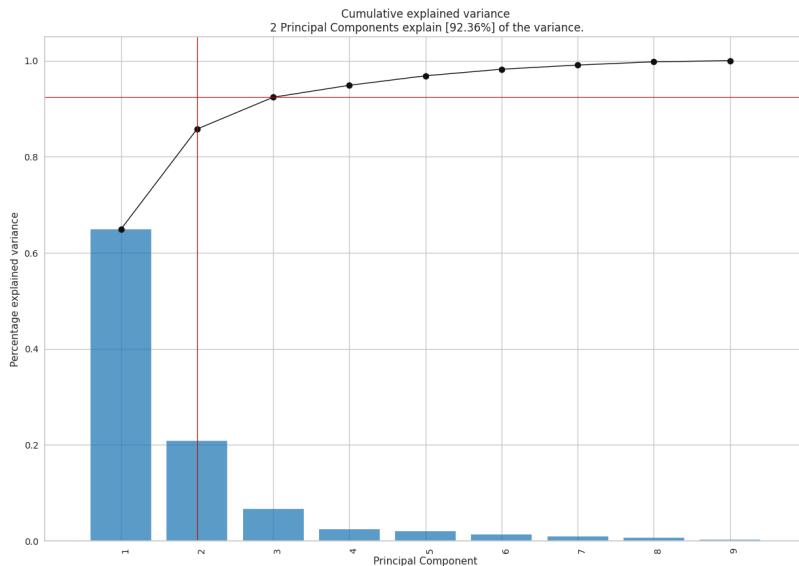
2022-04-29

- Added 🦀 mode (under Miscellaneous in Tools → Settings)
- Added "Disconnect and delete runtime" option to the menu next to the Connect button
- Improved rendering of filter options in an interactive table
- Added git-lfs to the base image
- Updated torch from 1.10.0 to 1.11.0, jupyter-core from 4.9.2 to 4.10.0, and cmake from 3.12.0 to 3.22.3
- Added more details to our [FAQ](#) about unsupported uses (using proxies, downloading torrents, etc.)
- Fixed [issue](#) with apt-get dependencies

2022-04-15

- Add an option in the file browser to show hidden files.
- Upgrade gdown from 4.2.0 to 4.4.0, google-api-core[grpc] from 1.26.0 to

1.31.5, and pytz from 2018.4 to 2022.1



2022-03-25

- Launched [Pro/Pro+](#) to 12 additional countries: Australia, Bangladesh, Colombia, Hong Kong, Indonesia, Mexico, New Zealand, Pakistan, Philippines, Singapore, Taiwan, and Vietnam
- Added [google.colab.auth.authenticate](#) to support using [Service Account keys](#)
- Update jax from 0.3.1 to 0.3.4 & jaxlib from 0.3.0 to 0.3.2
- Fixed an issue with Twitter previews of notebooks shared as GitHub Gists

2022-03-10

- Launched [Pro/Pro+](#) to 10 new countries: Ireland, Israel, Italy, Morocco, the Netherlands, Poland, Spain, Switzerland, Turkey, and the United Arab Emirates
- Launched support for [scheduling notebooks for Pro+ users](#)
- Fixed bug in interactive datatables where filtering by number did not work
- Finished removing the python2 kernelspec

2022-02-25

- Made various accessibility improvements to the header
- Fix bug with [forms run:auto](#) where a form field change would trigger multiple runs
- Minor updates to the [bigquery example notebook](#) and snippet
- Include background execution setting in the sessions dialog for Pro+ users
- Update tensorflow-probability from 0.15 to 0.16
- Update jax from 0.2.25 to 0.3.1 & jaxlib from 0.1.71 to 0.3.0

2022-02-11

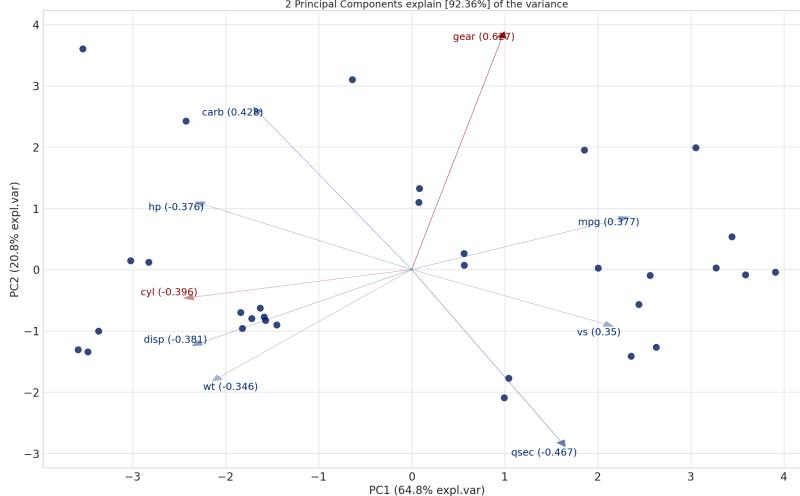
- Improve keyboard navigation for the open dialog
- Fix issue where nvidia-smi stopped reporting resource utilization for some users who were modifying the version of nvidia used
- Update tensorflow from 2.7 to 2.8, keras from 2.7 to 2.8, numpy from 1.19.5 to 1.21.5, tables from 3.4.4 to 3.7.0

Biplot

A biplot is a graphical representation of multivariate data that displays both observations and variables simultaneously in a single plot. Each observation is represented as a point in a two-dimensional plot, while the variables are represented as arrows or vectors that point in the direction of the greatest variation in the data. The length of the vector indicates the magnitude of the variable, while the angle between two vectors reflects their correlation.

```
model.biplot(label=False, legend=False)
```

→ [scatterd] >INFO> Create scatterplot
 [pca]> [WARNING]: De parameter <label> is deprecated
 [pca] >Plot PC1 vs PC2 with loadings.
 [scatterd]> WARNING use the standardized verbose sta
 (<Figure size 2500x1500 with 1 Axes>,
 <Axes: title={'center': '2 Principal Components
 explain [92.36%] of the variance'}, xlabel='PC1
 (64.8% expl.var)', ylabel='PC2 (20.8% expl.var)'>)



2022-02-04

- Improve UX for opening content alongside your notebook, such as files opened from the file browser. This includes a multi-pane view and drag-drop support
- Better Twitter previews when sharing example Colab notebooks and notebooks opened from GitHub Gists
- Update pandas from 1.1.5 to 1.3.5
- Update openpyxl from 2.5.9 to 3.0.0 and pyarrow from 3.0.0 to 6.0.0
- Link to the release notes from the Help menu

2022-01-28

- Add a copy button to [data tables](#)
- Python LSP support for better completions and code diagnostics. This can be configured in the Editor Settings (Tools → Settings)
- Update [gspread examples](#) in our documentation
- Update gdown from 3.6 to 4.2

2022-01-21

- New documentation for the [google.colab package](#)
- Show GPU RAM in the resource usage tab
- Improved security for mounting Google Drive which disallows mounting Drive from accounts other than the one currently executing the notebook

2022-01-14

- Add a preference (Tools → Settings) to use a temporary scratch notebook as the default landing page
- Fix bug where / and : weren't working in VIM mode
- Update gspread from 3.0 to 3.4
- Update the [Colab Marketplace VM image](#)

Multiclass Classification with PyCaret

Multiclass classification is a supervised machine learning technique where the goal is to classify instances into one of three or more classes. (Classifying instances into one of two classes is called Binary Classification).

Data Splitting

In order to demonstrate the predict_model() function on unseen data, a sample of 21 observations has been withheld from the original dataset to be used for predictions. This should not be confused with a train/test split as this particular split is

performed to simulate a real life scenario. Another way to think about this is that these 21 records were not available at the time when the machine learning experiment was performed.

```
data = df.sample(frac=0.9, random_state=786)
data_unseen = df.drop(data.index)

data.reset_index(drop=True, inplace=True)
data_unseen.reset_index(drop=True, inplace=True)

print('Data for Modeling: ' + str(data.shape))
print('Unseen Data For Predictions: ' + str(data_unseen.s
```

→ Data for Modeling: (29, 12)
Unseen Data For Predictions: (3, 12)

Setting up the Environment in PyCaret

The **setup()** function initializes the environment in pycaret and creates the transformation pipeline to prepare the data for modeling and deployment. **setup()** must be called before executing any other function in pycaret. It takes two mandatory parameters: a pandas dataframe and the name of the target column. All other parameters are optional and are used to customize the pre-processing pipeline.

When **setup()** is executed, PyCaret's inference algorithm will automatically infer the data types for all features based on certain properties. The data type should be inferred correctly but this is not always the case. To account for this, PyCaret displays a table containing the features and their inferred data types after **setup()** is executed. If all of the data types are correctly identified enter can be pressed to continue or quit can be typed to end the experiment. Ensuring that the data types are correct is of fundamental importance in PyCaret as it automatically performs a few pre-processing tasks which are imperative to any machine learning experiment. These tasks are performed differently for each data type which means it is very important for them to be correctly configured.

```
from pycaret.classification import *
clf = setup(data=data, target='am', train_size=0.7, sessi
```

	Description	Value
0	Session id	123
1	Target	am
2	Target type	Binary
3	Original data shape	(29, 12)
4	Transformed data shape	(29, 31)
5	Transformed train set shape	(20, 31)
6	Transformed test set shape	(9, 31)
7	Numeric features	10
8	Categorical features	1
9	Preprocess	True
10	Imputation type	simple
11	Numeric imputation	mean
12	Categorical imputation	mode
13	Maximum one-hot encoding	25
14	Encoding method	None
15	Fold Generator	StratifiedKFold
16	Fold Number	10
17	CPU Jobs	-1
18	Use GPU	False
19	Log Experiment	False
20	Experiment Name	clf-default-name
21	USI	81b0

Once the setup has been successfully executed it prints the information grid which contains several important pieces of information. Most of the information is related to the pre-processing pipeline which is constructed when setup() is executed. The majority of these features are out of scope for the purposes of this tutorial however a few important things to note at this stage include:

- **session_id** : A pseudo-random number distributed as a seed in all functions for later reproducibility. If no session_id is passed, a random number is automatically generated that is

distributed to all functions. In this experiment, the session_id is set as 123 for later reproducibility.

- **Target Type** : Binary or Multiclass. The Target type is automatically detected and shown. There is no difference in how the experiment is performed for Binary or Multiclass problems. All functionalities are identical.
- **Label Encoded** : When the Target variable is of type string (i.e. 'Yes' or 'No') instead of 1 or 0, it automatically encodes the label into 1 and 0 and displays the mapping (0 : No, 1 : Yes) for reference.
- **Original Data** : Displays the original shape of the dataset. In this experiment (189, 8) means 189 samples and 8 features including the class column.
- **Missing Values** : When there are missing values in the original data this will show as True. For this experiment there are no missing values in the dataset.
- **Numeric Features** : The number of features inferred as numeric. In this dataset, 7 out of 8 features are inferred as numeric.
- **Categorical Features** : The number of features inferred as categorical. In this dataset, there are no categorical features.
- **Transformed Train Set** : Displays the shape of the transformed training set. Notice that the original shape of (189, 8) is transformed into (132, 7) for the transformed train set.
- **Transformed Test Set** : Displays the shape of the transformed test/hold-out set. There are 57 samples in test/hold-out set. This split is based on the default value of 70/30 that can be changed using the train_size parameter in setup.

Notice how a few tasks that are imperative to perform modeling are automatically handled such as missing value imputation, categorical encoding etc. Most of the parameters in setup() are optional and used for customizing the pre-processing pipeline.

Comparing all Machine Learning Models

```
#show the best model and their statistics
best_model = compare_models()
```

→

	Model	Accuracy	AUC	Recall	Prec.	F1	MCC
ada	Ada Boost Classifier	1.0000	0.8000	0.8000	0.8000	0.8000	0.8000
gbc	Gradient Boosting Classifier	1.0000	0.8000	0.8000	0.8000	0.8000	0.8000
dt	Decision Tree Classifier	0.9500	0.8000	0.8000	0.8000	0.8000	0.8000
ridge	Ridge Classifier	0.9500	0.7000	0.8000	0.7500	0.7500	0.7500
rf	Random Forest Classifier	0.9500	0.8000	0.8000	0.8000	0.8000	0.8000
xgboost	Extreme Gradient Boosting	0.9500	0.8000	0.8000	0.8000	0.8000	0.8000
lr	Logistic Regression	0.9000	0.8000	0.8000	0.7500	0.7500	0.7500
knn	K Neighbors Classifier	0.9000	0.8000	0.8000	0.7500	0.7500	0.7500
-	Naive	~ 0.8000	~ 0.8000	~ 0.8000	~ 0.7500	~ 0.7500	~ 0.7500

```
best_model
```

→ ▾

```
AdaBoostClassifier
AdaBoostClassifier(algorithm='SAMME.R', estimator=NaiveBayes,
n_estimators=50, random_state=12)
```

Create a Model

`create_model` is the most granular function in PyCaret and is often the foundation behind most of the PyCaret functionalities. As the name suggests this function trains and evaluates a model using cross validation that can be set with `fold` parameter. The output prints a score grid that shows Accuracy, Recall, Precision, F1, Kappa and MCC by fold.

For the remaining part of this tutorial, we will work with the below models as our candidate models. The selections are for

illustration purposes only and do not necessarily mean they are the top performing or ideal for this type of data.

- Decision Tree Classifier ('dt')
- K Neighbors Classifier ('knn')
- Logistic Regression ('lr')

There are many classifiers available in the model library of PyCaret. Please view the `create_model()` docstring for the list of all available models.

Create Decision Tree Classifier

```
dt = create_model('dt')
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	C
Fold							
0	1.0000	0.0000	0.0000	0.0000	0.0000	nan	C
1	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	C
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
6	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
Mean	0.9500	0.8000	0.8000	0.8000	0.8000	nan	C
Std	0.1500	0.4000	0.4000	0.4000	0.4000	nan	C

```
#trained model object is stored in the variable 'dt'.
dt
```

```
DecisionTreeClassifier
DecisionTreeClassifier(ccp_alpha=0.0, class_weight='balanced',
max_depth=None, max_features='auto',
min_impurity_decrease=0.0, min_samples_leaf=1,
min_samples_split=2, min_weight_fraction_leaf=0.0,
monotonic_cst=None, random_state=42)
```

Tune a Model: How to automatically tune the hyper-parameters of a multiclass model. When a model is created using the `create_model()` function it uses the default hyperparameters. In order to tune hyperparameters, the `tune_model()` function is used. The `tune_model()` function is a random grid search of hyperparameters over a pre-defined search space. By default, it is set to optimize Accuracy but this can be changed using `optimize` parameter. This function automatically tunes the hyperparameters of a model on a pre-defined search space and scores it using stratified cross validation. The output prints a score grid that shows Accuracy, AUC, Recall, Precision, F1 and Kappa by fold.

Tune Decision Tree Model

```
tuned_dt = tune_model(dt)
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	
Fold							
0	1.0000	0.0000	0.0000	0.0000	0.0000	nan	C
1	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	C
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
6	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
Mean	0.9500	0.8000	0.8000	0.8000	0.8000	nan	C
Std	0.1500	0.4000	0.4000	0.4000	0.4000	nan	C

Fitting 10 folds for each of 10 candidates, totallir
Original model was better than the tuned model, hence

```
#tuned model object is stored in the variable 'tuned_dt'
tuned_dt
```



DecisionTreeClassifier

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None,  
                      max_depth=None, max_features=None,  
                      min_impurity_decrease=0.0, min_samples_leaf=1,  
                      min_samples_split=2, min_weight_fraction_leaf=0.0,  
                      monotonic_cst=None, random_state=42)
```

Evaluate Decision Tree Model

How to analyze model performance using various plots

```
evaluate_model(tuned_dt)
```



Plot Type:

Pipeline Plot

Hyperparameters

AUC

Confusion Matrix

Threshold

Precision Recall

Prediction Error

Class Report

Feature Selection

Learning Curve

Manifold Learning

Calibration Curve

Validation Curve

Dimensions

Create K Neighbors Model

```
knn = create_model('knn')
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	
Fold							
0	1.0000	0.0000	0.0000	0.0000	0.0000	nan	C
1	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	C
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
3	0.5000	1.0000	1.0000	0.5000	0.6667	0.0000	C
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
6	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
Mean	0.9000	0.8000	0.8000	0.7500	0.7667	nan	C
Std	0.2000	0.4000	0.4000	0.4031	0.3958	nan	C

Tune K Neighbors Model

```
tuned_knn = tune_model(knn, custom_grid = {'n_neighbors':
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	C
Fold							
0	1.0000	0.0000	0.0000	0.0000	0.0000	nan	C
1	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	C
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
3	0.5000	0.5000	1.0000	0.5000	0.6667	0.0000	C
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
5	0.5000	1.0000	0.0000	0.0000	0.0000	0.0000	C
6	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
8	0.5000	1.0000	0.0000	0.0000	0.0000	0.0000	C
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
Mean	0.8000	0.7500	0.6000	0.5500	0.5667	nan	C
Std	0.2449	0.4031	0.4899	0.4717	0.4726	nan	C

Fitting 10 folds for each of 10 candidates, totallir
Original model was better than the tuned model, henc

Evaluate K Neighbors Model

```
evaluate_model(tuned_knn)
```

Plot Type:	
Pipeline Plot	Hyperparameters
AUC	Confusion Matrix
Threshold	Precision Recall
Prediction Error	Class Report
Feature Selection	Learning Curve
Manifold Learning	Calibration Curve
Validation Curve	Dimensions

Create Logistic Regression Model

```
lr = create_model('lr')
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	
Fold							
0	1.0000	0.0000	0.0000	0.0000	0.0000	nan	C
1	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	C
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
3	0.5000	1.0000	1.0000	0.5000	0.6667	0.0000	C
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
6	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
Mean	0.9000	0.8000	0.8000	0.7500	0.7667	nan	C
Std	0.2000	0.4000	0.4000	0.4031	0.3958	nan	C

lr

	LogisticRegression
	LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, multi_class='auto', n_jobs=None, random_state=123, solver='lbfgs', warm_start=False)

Tune Logistic Regression Model

```
tuned_lr = tune_model(lr)
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	C
Fold							
0	1.0000	0.0000	0.0000	0.0000	0.0000	nan	C
1	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	C
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
3	0.5000	1.0000	1.0000	0.5000	0.6667	0.0000	C
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
6	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
Mean	0.9000	0.8000	0.8000	0.7500	0.7667	nan	C
Std	0.2000	0.4000	0.4000	0.4031	0.3958	nan	C

Fitting 10 folds for each of 10 candidates, totallir
Original model was better than the tuned model, hence

Evaluate Logistic Regression Model

tuned_lr

▼	LogisticRegression
	LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, multi_class='auto', n_jobs=None, random_state=123, solver='lbfgs', warm_start=False)

evaluate_model(tuned_lr)

Plot Type:	
<input checked="" type="radio"/> Pipeline Plot	Hyperparameters
<input type="radio"/> AUC	Confusion Matrix
<input type="radio"/> Threshold	Precision Recall
<input type="radio"/> Prediction Error	Class Report
<input type="radio"/> Feature Selection	Learning Curve
<input type="radio"/> Manifold Learning	Calibration Curve
<input type="radio"/> Validation Curve	Dimensions

Create Random Forest Model

```
rf = create_model('rf')
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	C
0	0.6667	1.0000	1.0000	0.5000	0.6667	0.4000	0
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
2	1.0000	0.0000	0.0000	0.0000	0.0000	nan	0
3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
6	0.5000	1.0000	0.0000	0.0000	0.0000	0.0000	0
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
Mean	0.9167	0.9000	0.8000	0.7500	0.7667	nan	0
Std	0.1708	0.3000	0.4000	0.4031	0.3958	nan	0

Tune Random Forest Model

```
tuned_rf = tune_model(rf)
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	C
Fold							C
0	0.3333	0.7500	1.0000	0.3333	0.5000	0.0000	C
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
2	1.0000	0.0000	0.0000	0.0000	0.0000	nan	C
3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
6	0.5000	0.5000	1.0000	0.5000	0.6667	0.0000	C
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
Mean	0.8833	0.8250	0.9000	0.7833	0.8167	nan	C
Std	0.2363	0.3172	0.3000	0.3500	0.3202	nan	C

Fitting 10 folds for each of 10 candidates, totallir
Original model was better than the tuned model, hence

Evaluate Random Forest Model

```
evaluate_model(tuned_rf)
```

Plot Type:	Pipeline Plot	Hyperparameters
	AUC	Confusion Matrix
	Threshold	Precision Recall
	Prediction Error	Class Report
	Feature Selection	Learning Curve
	Manifold Learning	Calibration Curve
	Validation Curve	Dimensions

▼ Tune the Best Model

```
# Tune hyperparameters with scikit-learn (default)
tuned_best_model = tune_model(best_model)
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	C
Fold							C
0	0.3333	0.5000	1.0000	0.3333	0.5000	0.0000	C
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
2	1.0000	0.0000	0.0000	0.0000	0.0000	nan	C
3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
6	0.5000	0.0000	1.0000	0.5000	0.6667	0.0000	C
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1
Mean	0.8833	0.7500	0.9000	0.7833	0.8167	nan	C
Std	0.2363	0.4031	0.3000	0.3500	0.3202	nan	C

Fitting 10 folds for each of 10 candidates, totallir
Original model was better than the tuned model, henc

```
tuned_best_model
```

```
AdaBoostClassifier
AdaBoostClassifier(algorithm='SAMME.R', estimator=None,
n_estimators=50, random_state=12)
```

Evaluate the Best Model

One way to analyze the performance of models is to use the `evaluate_model()` function which displays a user interface for all of the available plots for a given model. It internally uses the `plot_model()` function.

```
evaluate_model(tuned_best_model)
```

Plot Type:

Pipeline Plot	Hyperparameters
AUC	Confusion Matrix
Threshold	Precision Recall
Prediction Error	Class Report
Feature Selection	Learning Curve
Manifold Learning	Calibration Curve
Validation Curve	Dimensions

Classification + PCA

```
clf_pca = setup(data=df, target='am', train_size=0.7, ses
```

	Description	Value
0	Session id	123
1	Target	am
2	Target type	Binary
3	Original data shape	(32, 12)
4	Transformed data shape	(32, 4)
5	Transformed train set shape	(32, 4)
#show the best model and their statistics		
best_model_pca = compare_models()		
7	Numeric features	10
8	Model	Accuracy AUC Recall Prec. F1
xgboost	Extreme Gradient Boosting	0.9667 0.8750 0.9000 0.8500 0.9000
ridge	Numeric imputation Ridge Classifier	0.9500 0.9000 0.9000 0.9000 0.9000
12	Categorical imputation	
lida	Linear Discriminant Analysis	0.9500 0.9000 0.9000 0.9000 0.9000
nb	Naive Bayes	0.9167 0.8500 0.9000 0.8500 0.9000
rf	Random Forest Classifier	0.9167 0.9000 0.8000 0.7500 0.8000
lr	Logistic Regression	0.9000 0.9000 0.8000 0.8000 0.8000
dt	Decision Tree Classifier	0.8833 0.8000 0.9000 0.7833 0.8000
qda	Quadratic Discriminant Analysis	0.8833 0.9000 0.8000 0.7500 0.8000
adaboost	Ada Boost	0.8000 0.8000 0.8000 0.7000 0.8000

best_model_pca

```
XGBClassifier
XGBClassifier(base_score=None, booster='gbtree', ca
  colsample_bylevel=None, colsample_byne
  colsample_bytree=None, device='cpu',
  enable_categorical=False, eval_metric
  gamma=None, grow_policy=None, importa
  interaction_constraints=None, learnin
  max_cat_threshold=None, max_cat_to_on
```

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.