DATA SCIENCE E-BOOK

PYTHON PACKAGES TO LEARN DATA SCIENCE

The packages recommended for your learning

V.2.0

WRITTEN BY CORNELLIUS YUDHA WJAYA

PYTHON PACKAGES TO LEARN DATA SCIENCE

About

Please feel free to share this PDF with anyone for free.
The latest version of this book can be downloaded from:

https://cornelliusyudhawijay.gumroad.com/follow

Visit me at:

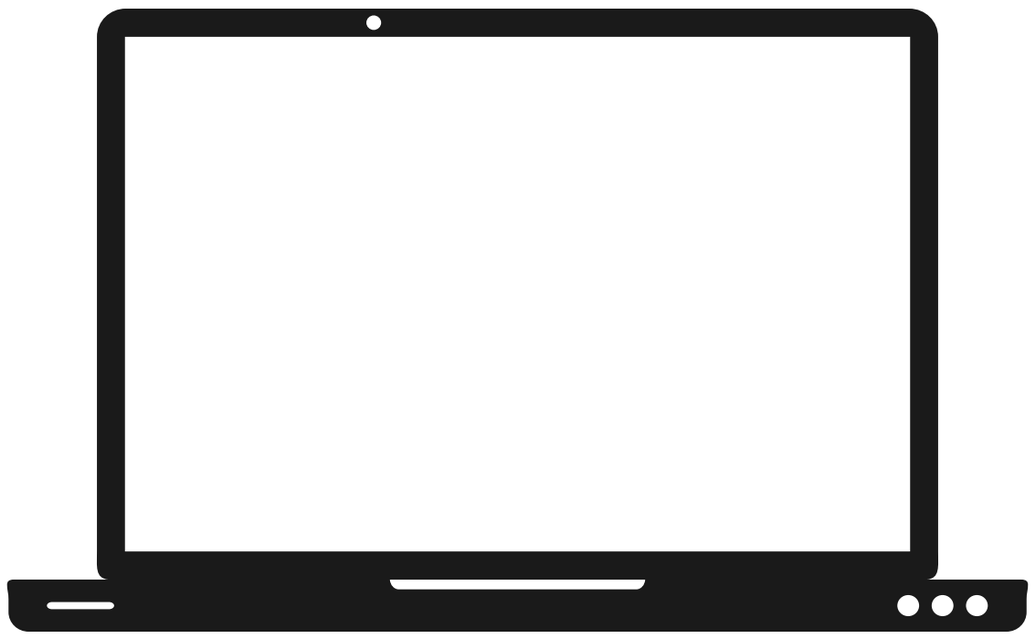
LinkedIn: **Cornellius Yudha Wijaya**

Medium: **@cornelliusyudhawijaya**

Substack: **cornellius.substack.com**

TABLE OF

CONTENTS



I. Introduction

Why this ebook is created?

Data science is a broad subject; that is why learning all the concepts would not be easy. Many have asked me how to learn data science and machine learning concepts properly. For me, the answer would be learning by hands-on with the current technology, and that is Python programming. To effectively learn data science, I want to introduce various Python packages to support your data science learning.

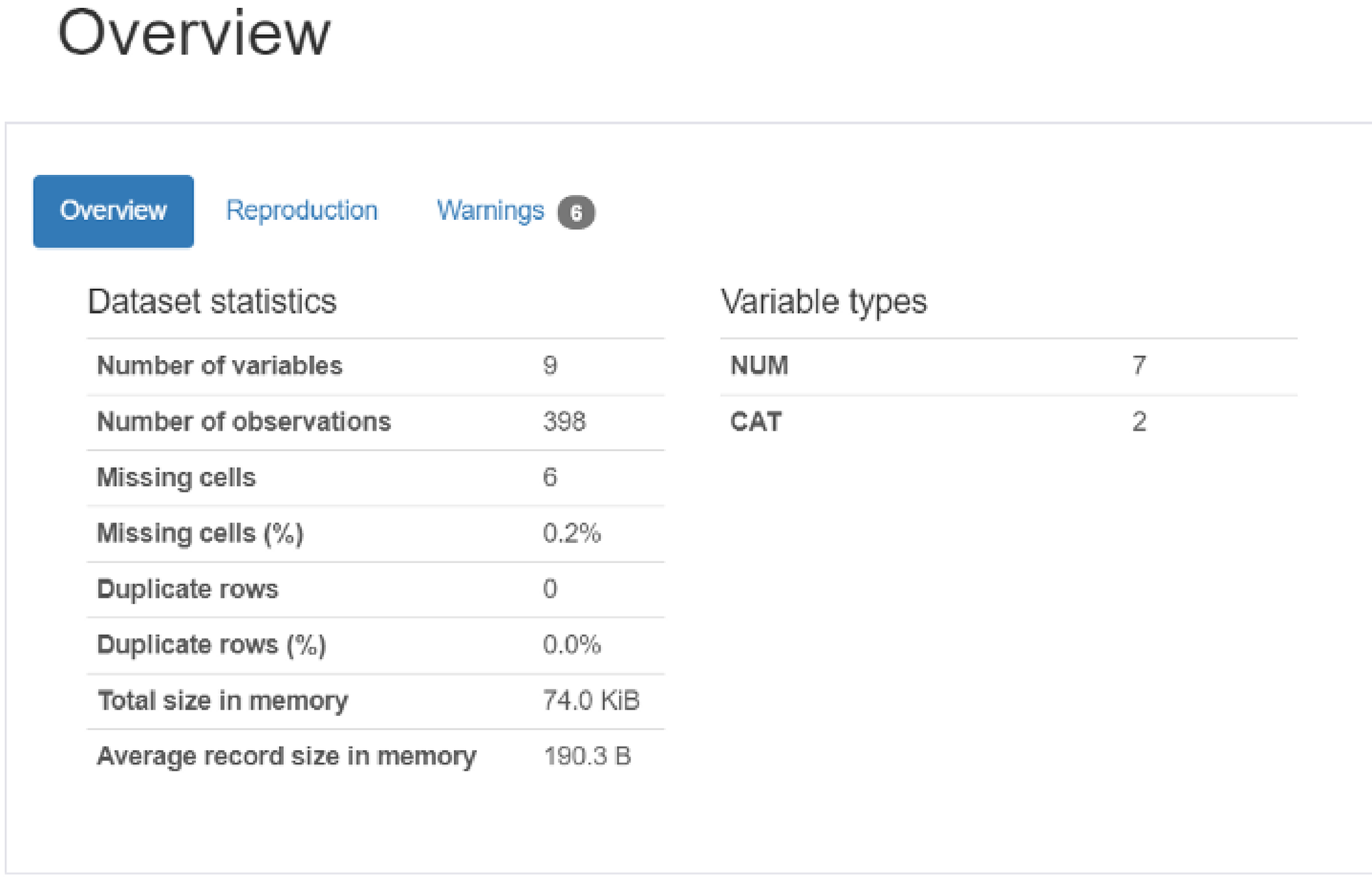
02

EXPLORATORY DATA ANALYSIS

02

y-data Profiling

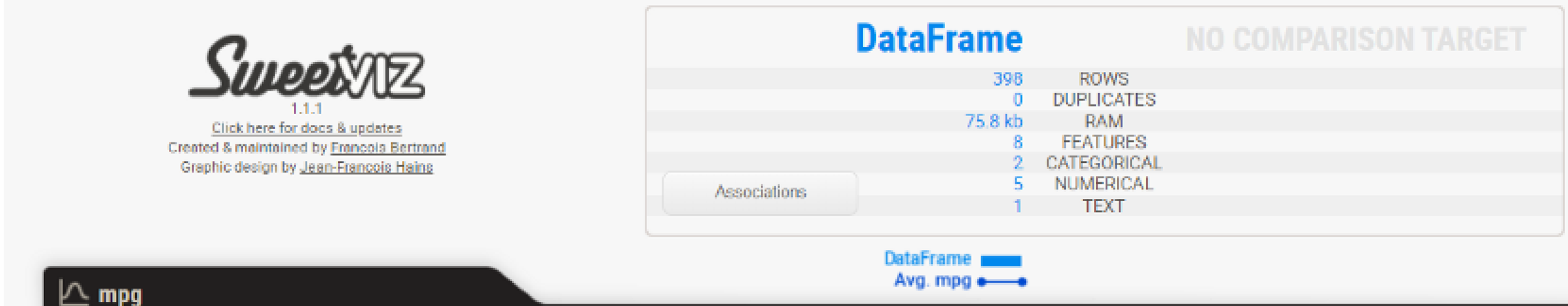
Complete set of report



y-data profiling *is a Python Package to generate data analysis reports with as few lines as possible. It offers a nice quick report of the dataset. This module is the best to work in the Jupyter environment.*

```
 pip install ydata-profiling
[notebook]
#Enable the widget extension in Jupyter
jupyter nbextension enable --py widgetsnbextension
```

Sweetviz

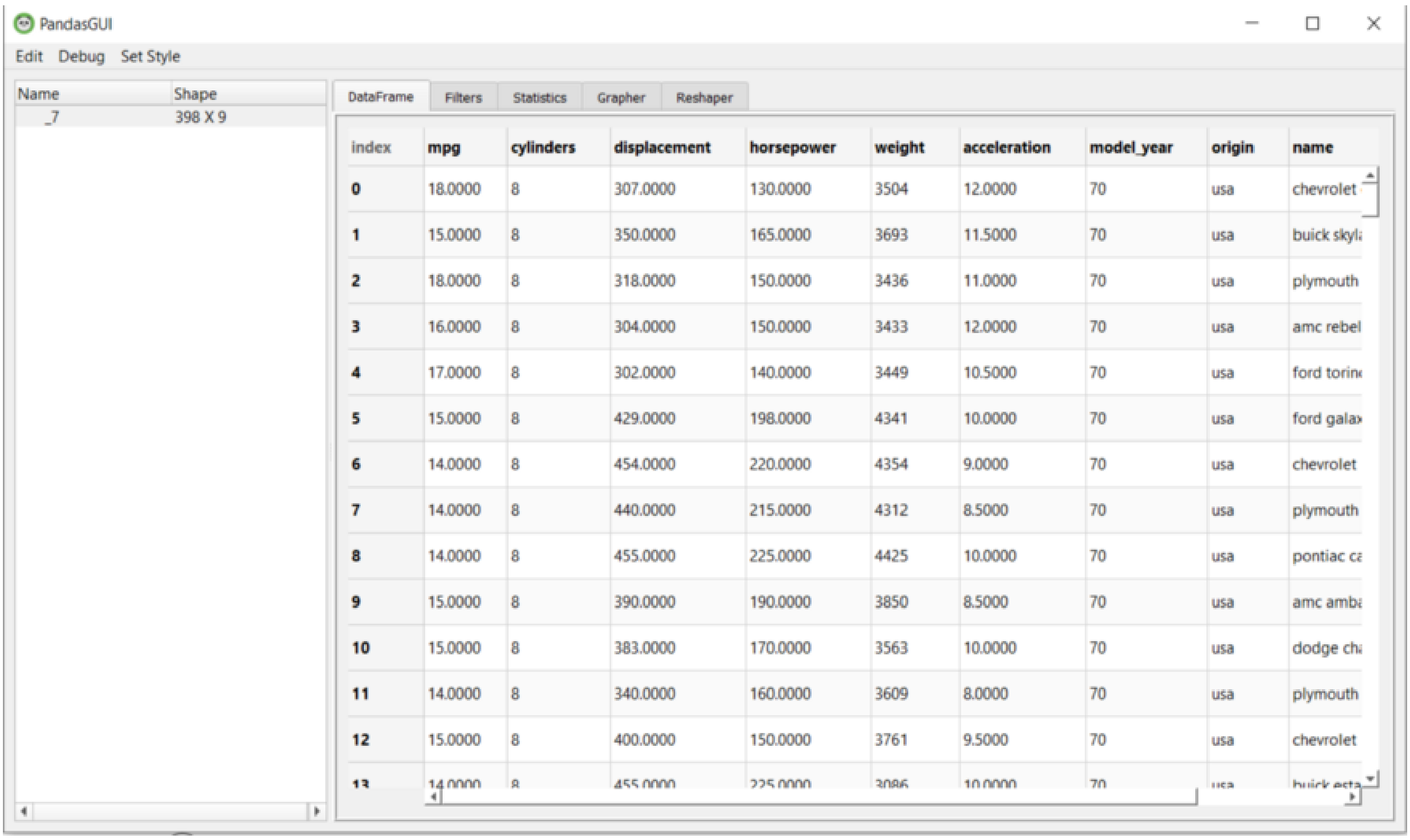


Sweetviz *is another open-source Python package to generate a beautiful EDA report with a single code line. The difference from Pandas Profiling is that the output is a fully self-contained HTML application.*

```
#Installing the sweetviz package via pip

pip install sweetviz
```

PandasGUI



PandasGUI *is different from the previous packages I explained above. Instead of generating a report, PandasGUI generates a* GUI *(Graphical User Interface) data frame we could use to analyze our Pandas Data Frame in more detail.*

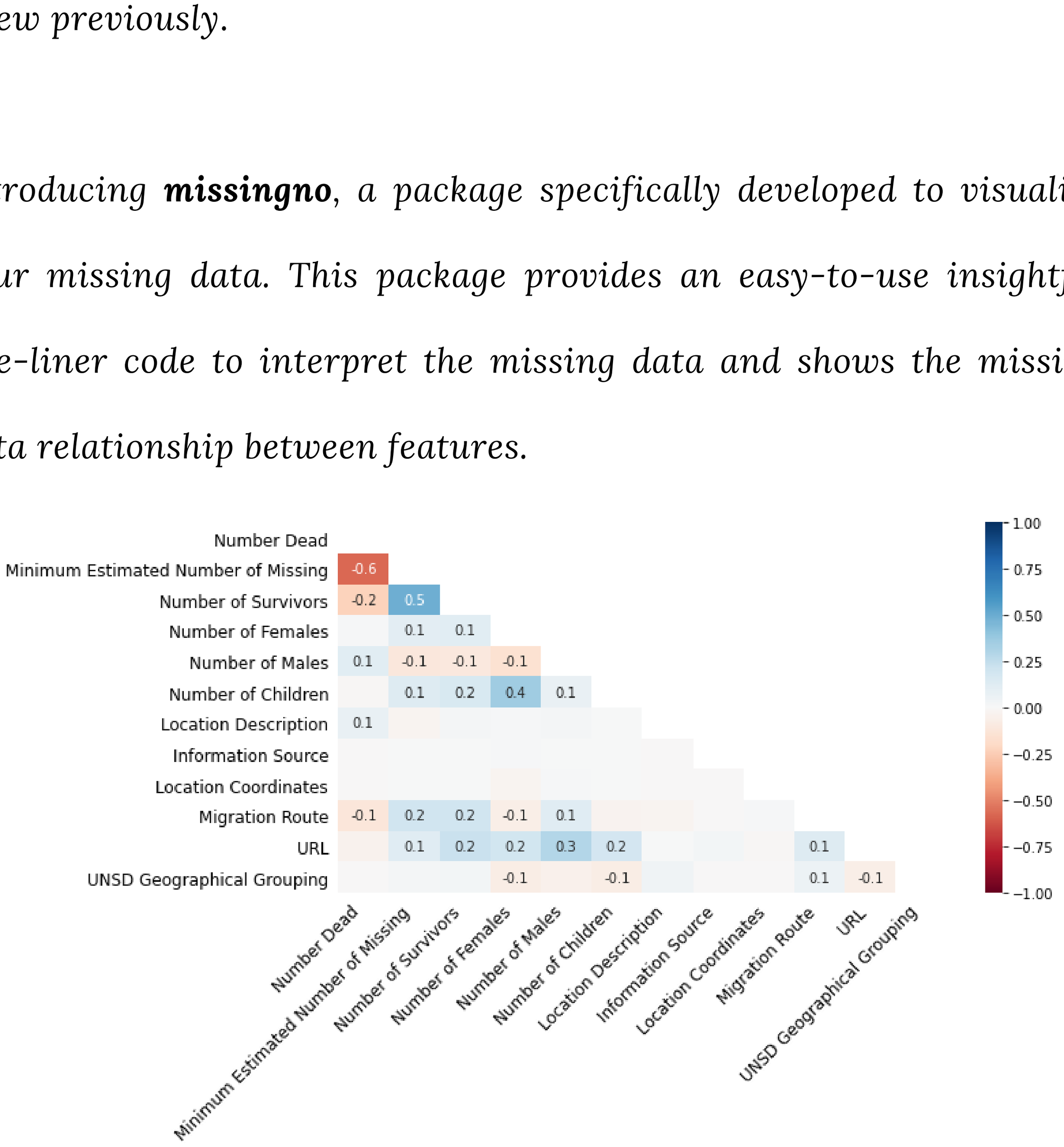
```
#Installing via pip
pip install pandasgui

#or if you prefer directly from the source
pip install git+https://github.com/adamerose/pandasgui.git
```

Missingno

Data exploration is not limited to the data present in the dataset, but it includes the missing data from your dataset. There are cases that missing data happen because of an accident or pure chance, but this is often not true. Missing data might uncover insight that we never knew previously.

Introducing ***missingno****, a package specifically developed to visualize your missing data. This package provides an easy-to-use insightful one-liner code to interpret the missing data and shows the missing data relationship between features.*

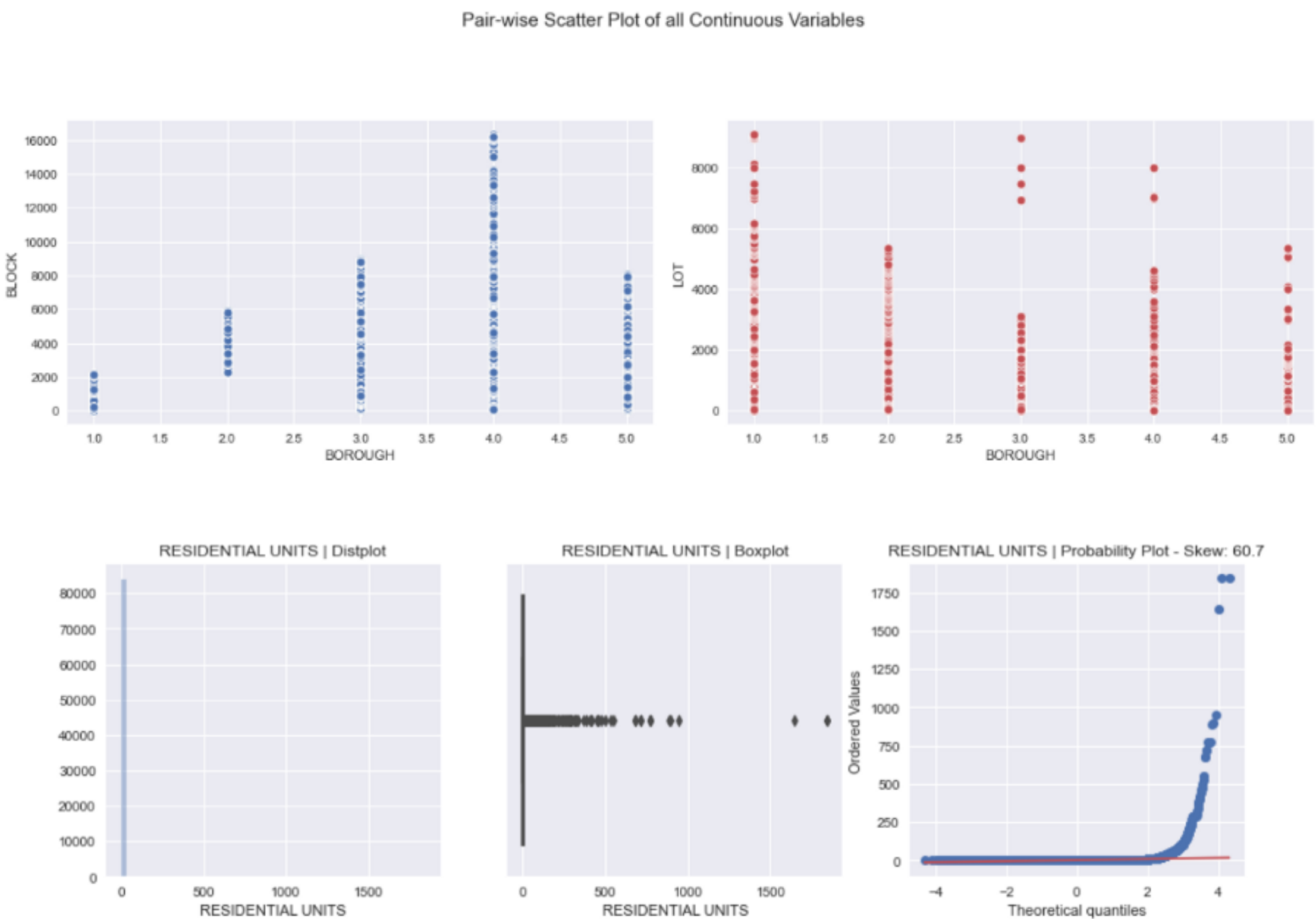



```
pip install missingno
```

AutoViz

AutoViz *is an open-source visualization package under the* AutoViML *package library designed to automate many data scientists' works. Many of the projects were quick and straightforward but undoubtedly helpful, including* AutoViz.

AutoViz is a one-liner code visualization package that would automatically produce data visualization.




```
pip install autoviz
```

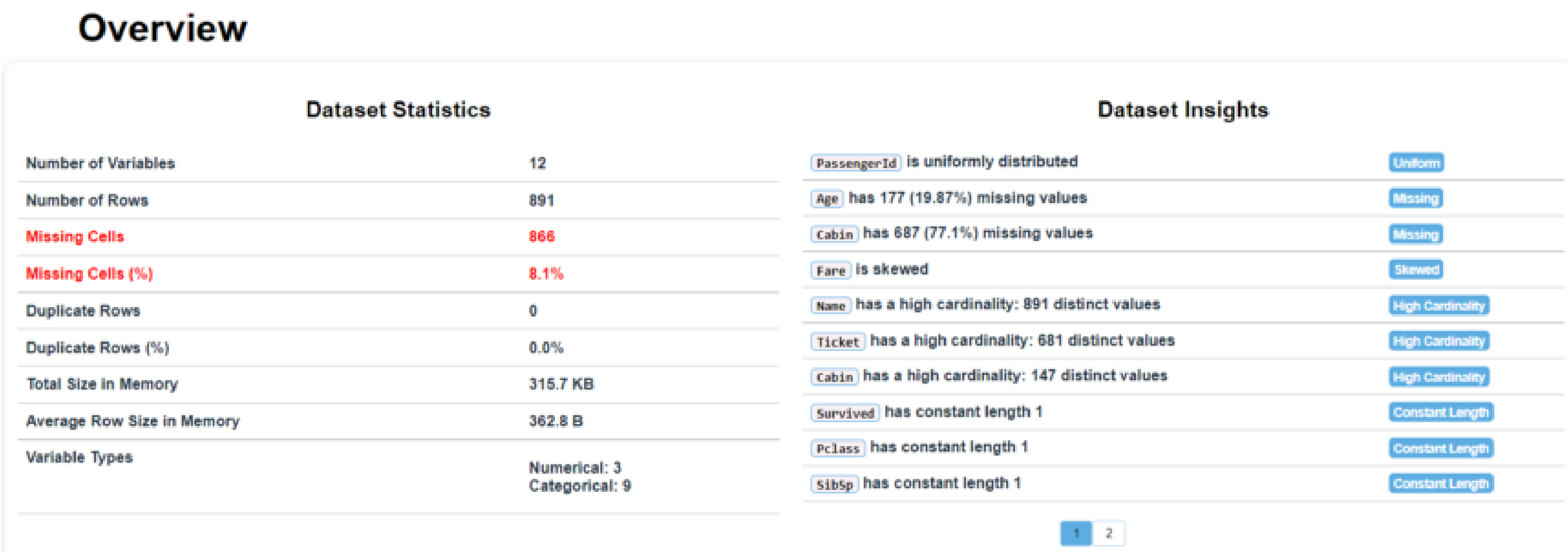
DataPrep

Data preparation is the initial step that any data professional does. Whether you want to analyze the data or preprocess the data for a machine learning model, you need to prepare the data. Preparing data means you need to collect, clean, and explore the data. To do all the activities I have mentioned, there is a Python package developed called **DataPrep**.

DataPrep is a Python Package developed to prepare your data. This package contains three main APIs for us to use, they are:

- Data Exploration (dataprep.eda)
- Data Cleaning(dataprep.clean)
- Data Collection (dataprep.connector)

DataPrep packages are designed to have a fast data exploration and work well with Pandas and Dask DataFrame objects.




```
pip install -U dataprep
```

03

STATISTIC

03

Scipy.Stats

SciPy (*pronounced "Sigh Pie") is an open-source package computing tool for performing a scientific method in the Python environment. The Scipy itself is also a collection of numerical algorithms and domain-specific toolboxes used in many mathematical, engineering, and data research.*

One of the APIs available within Scipy is the statistical API called **Stats**. *According to the Scipy homepage, Scipy.Stats is a module that contains a large number of probability distributions and a growing library of statistical functions, especially for probability function study.*

```
python -m pip install --user numpy scipy matplotlib ipython jupyter pandas sympy nose
```

Pingouin

Pingouin *is an open-source statistical package that is mainly used for statistical. This package gives you many classes and functions to learn basic statistics and hypothesis testing. According to the developer, Pingouin is designed for users who want simple yet exhaustive stats functions.*

Pingouin is simple but exhaustive because the package gives you more explanation regarding the data. On Scipy.Stats, they return only the T-value and the p-value when sometimes we want more explanation regarding the data.

In the Pingouin package, the calculation is taken a few steps above. For example, instead of returning only the T-value and p-value, the t-test from Pingouin also return the degrees of freedom, the effect size (Cohen's d), the 95% confidence intervals of the difference in means, the statistical power, and the Bayes Factor (BF10) of the test.

	Source	ddof1	ddof2	F	p-unc	np2
0	origin	2	395	98.542	1.915486e-35	0.333

```
pip install pingouin
```

Statsmodel

Statsmodels *is a statistical model python package that provides many classes and functions to create a statistical estimation. Statsmodel package used to be a part of the Scipy module, but currently, the statsmodel package is developed separately.*

What is different between Scipy.Stats and statsmodel? The Scipy.Stats module focuses on the statistical theorem such as probabilistic function and distribution, while the statsmodel package focuses on the statistical estimation based on the data.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.741
Model:	OLS	Adj. R-squared:	0.734
Method:	Least Squares	F-statistic:	108.1
Date:	Fri, 14 May 2021	Prob (F-statistic):	6.72e-135
Time:	21:20:00	Log-Likelihood:	-1498.8
No. Observations:	506	AIC:	3026.
Df Residuals:	492	BIC:	3085.
Df Model:	13		
Covariance Type:	nonrobust		

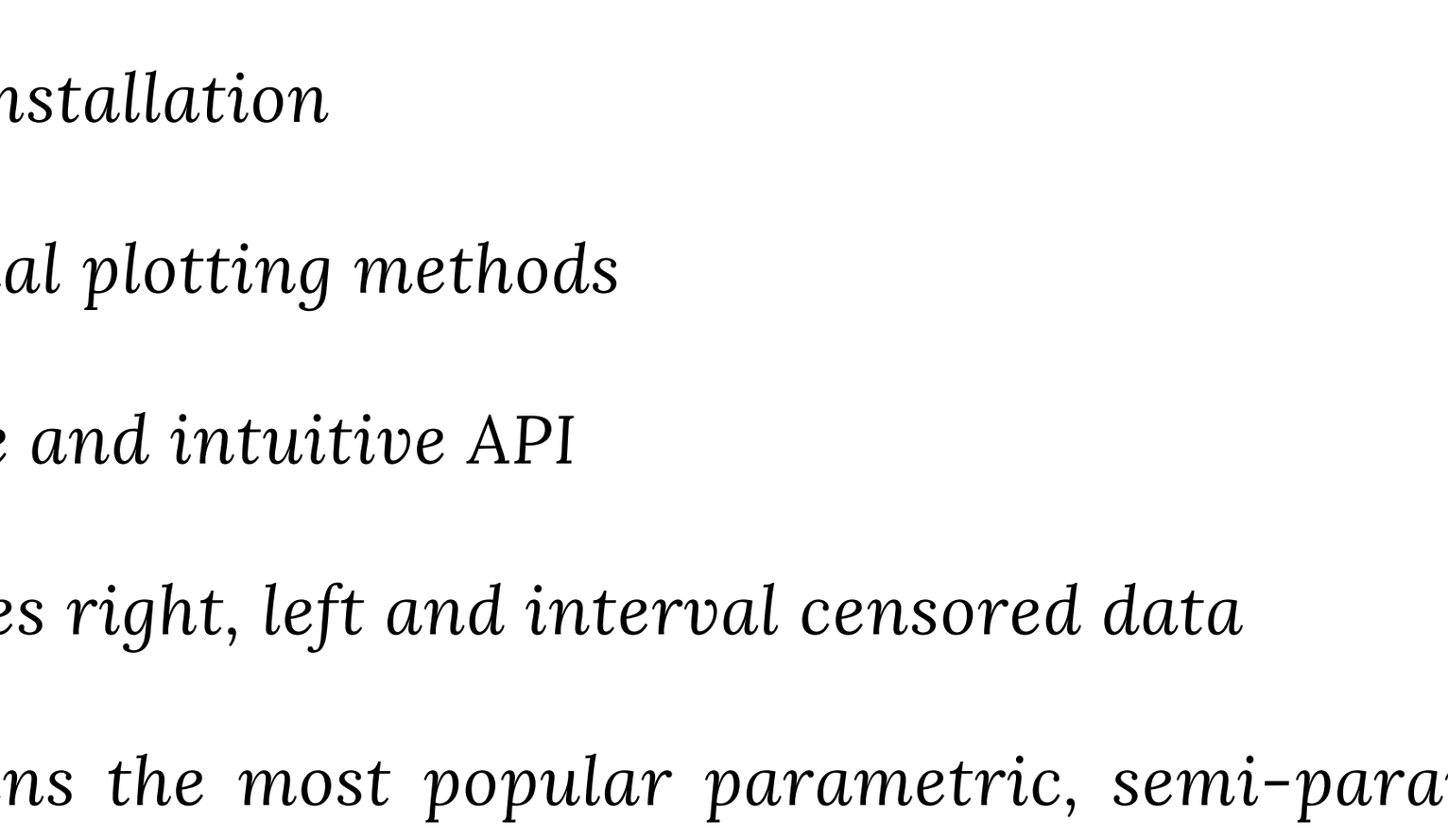
```
pip install statsmodels
```

Lifelines

*The **lifelines** package in Python is a specialized library used for survival analysis, a set of statistical approaches for analyzing the expected duration of time until one or more events happen. This type of analysis is commonly used in fields like biology, engineering, and economics, especially for analyzing the time until events like death, failure, or churn occur.*

Lifelines packages features including:

- *easy installation*
- *internal plotting methods*
- *simple and intuitive API*
- *handles right, left and interval censored data*
- *contains the most popular parametric, semi-parametric and non-parametric models*

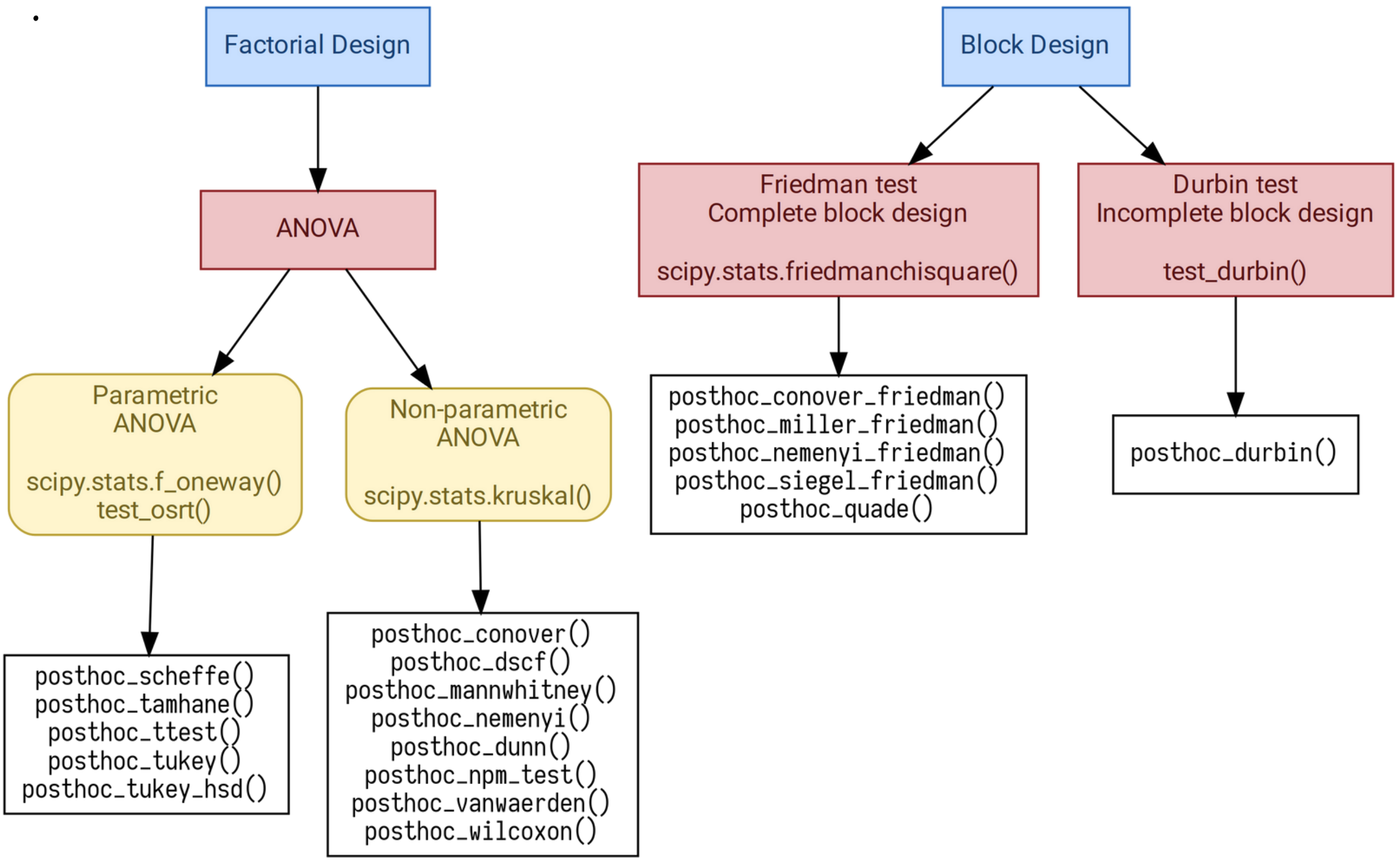



```
pip install lifelines
```

Scikit-posthocs

scikit-posthocs *is a Python package that provides post hoc tests for pairwise multiple comparisons that are usually performed in statistical data analysis to assess the differences between group levels if a statistically significant result of the* ANOVA *test has been obtained.*

scikit-posthocs attempts to improve Python statistical capabilities by offering a lot of parametric and nonparametric post hoc tests along with outliers detection and basic plotting methods.




```
pip install scikit-posthocs
```

linearmodels

linearmodels *is a Python package that Extends stats models with Panel regression, instrumental variable estimators, system estimators, and models for estimating asset prices. The package's main features include:*

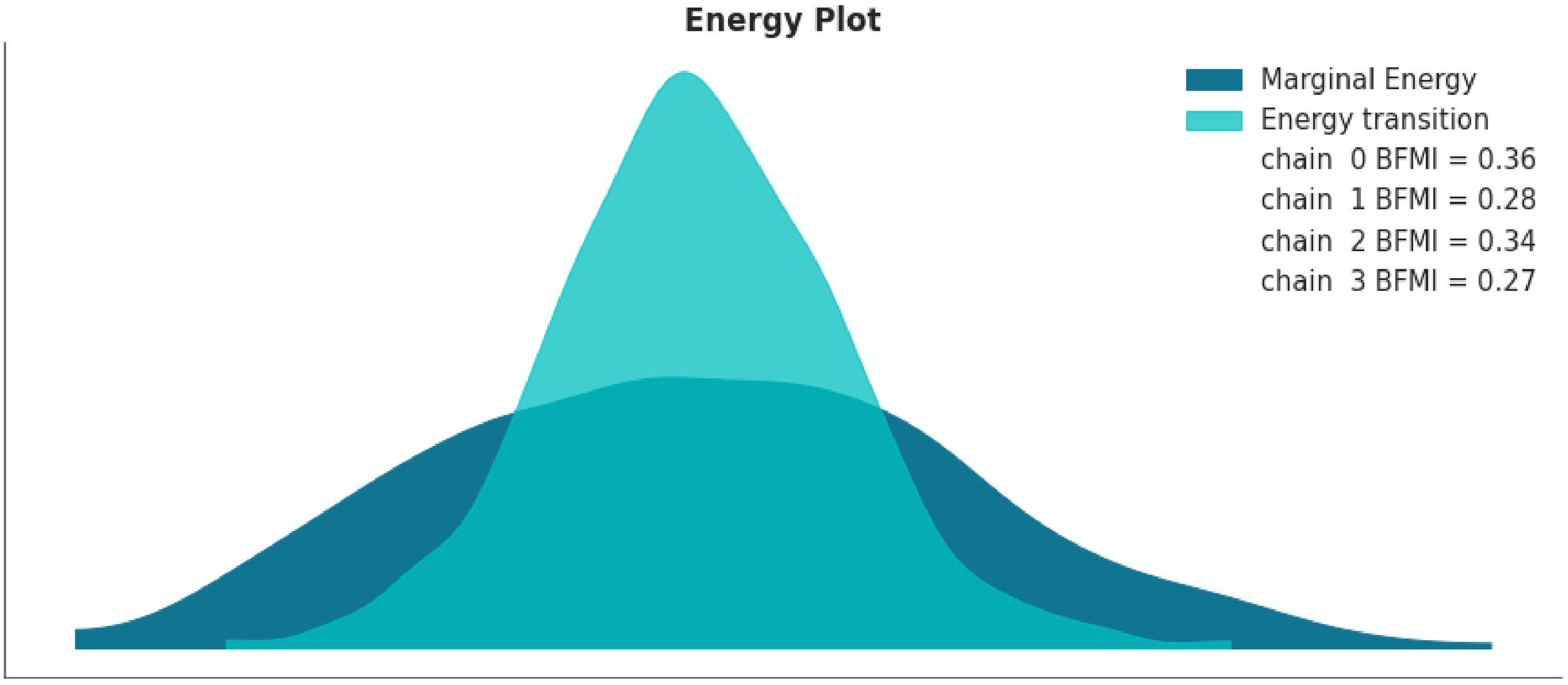
- *Panel Data Models: It supports various models for panel data analysis, such as Panel regression with fixed effects, random effects, and between estimators. It also includes support for first-difference models and pooled models.*
- *Instrumental Variable Estimators: It provide tools for two-stage least squares (2SLS) for instrumental variables regression, which is useful when dealing with endogenous regressors.*
- *Asset Pricing Models: The package includes models specifically designed for asset pricing, such as factor models used in finance.*
- *System Estimators: These are used for estimating simultaneous equations models, which are common in econometrics.*

```
                Model Comparison
=================================================
                          Panel OLS       Panel IV
-------------------------------------------------
Dep. Variable                lscrap         lscrap
Estimator                       OLS         IV-2SLS
No. Observations                 45              45
Cov. Est.                unadjusted      unadjusted
R-squared                    0.0619          0.0159
Adj. R-squared               0.0401         -0.0070
F-statistic                  2.9707          3.3464
P-value (F-stat)             0.0848          0.0674
==================     ============    ============
Intercept                   -0.1035         -0.0327
                          (-1.0208)       (-0.2632)
hrsemp                      -0.0076         -0.0142
                          (-1.7236)       (-1.8293)
=================== ============== ==============
Instruments                                   grant
-------------------------------------------------
```

```
pip install linearmodels
```

ArviZ

ArviZ *is a Python package for exploratory analysis of Bayesian models. It serves as a backend-agnostic tool for diagnosing and visualizing Bayesian inference.*




```
pip install arviz
```

04

MATHEMATIC

04

Numeric and Mathematical Modules

Well, technically, the packages I want to outline in this point is not only one single package, but it ***consists of several packages*** *that intertwined and were called* ***Numeric and Mathematical Modules.***

The modules were documented on the **Python homepage**, *and we are provided with a complete explanation regarding the package. Taken from the Python Documentation, the packages listed in their modules are:*

- **numbers** — Numeric abstract base classes
- **math** — Mathematical functions
- **cmath** — Mathematical functions for complex numbers
- **decimal** — Decimal fixed point and floating-point arithmetic
- **fractions** — Rational numbers
- **random** — Generate pseudo-random numbers
- **statistics** — Mathematical statistics functions

SymPy

What is **SymPy***? It is a Python library for symbolic mathematics. Well, what is Symbolic computation? The tutorial page given in the SymPy documentation explains that Symbolic computation is a computation problem that deals with mathematical objects symbolically. In simpler terms, symbolic mathematics represented the mathematical object precisely and not approximately. If the mathematical expressions are unevaluated variables, they are left in the symbolic form.*

Basic functionality of the module

Introduction

This tutorial tries to give an overview of the functionality concerning polynomials within SymPy. All code examples assume:

Run code block in SymPy Live

```
>>> from sympy import *
>>> x, y, z = symbols('x,y,z')
>>> init_printing(use_unicode=False, wrap_line=False)
```

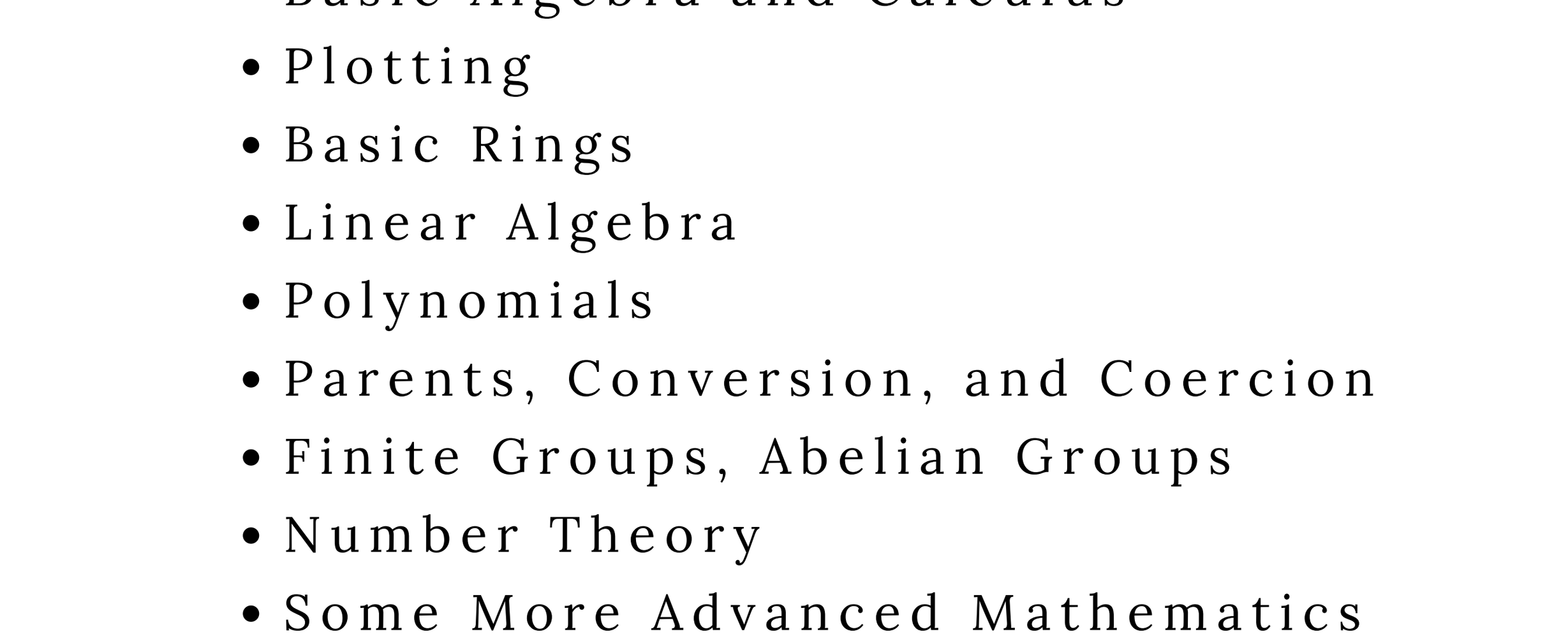
```
pip install sympy
```

Sage

Sage *is open-source mathematic software that runs above the Python programming language. Technically, Sage wasn't Python Package but rather software. The usage is simple if you already know Python, so you would not feel too hard when using the software.*

Sage supports research and teaching in algebra, geometry, number theory, cryptography, numerical computation, and related areas. There are many general and specific topic that was included within Sage, including:

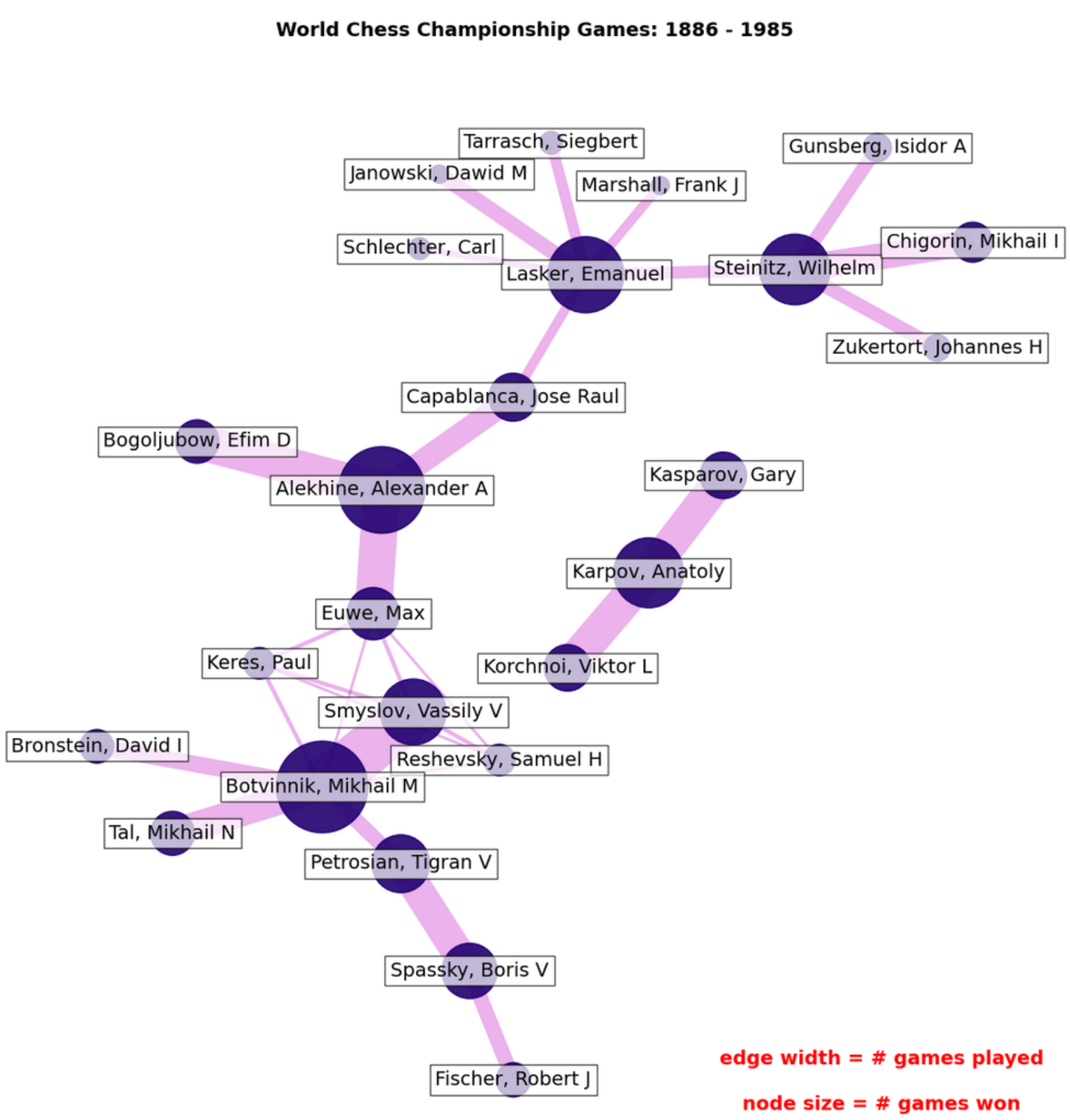
- Basic Algebra and Calculus
- Plotting
- Basic Rings
- Linear Algebra
- Polynomials
- Parents, Conversion, and Coercion
- Finite Groups, Abelian Groups
- Number Theory
- Some More Advanced Mathematics



NetworkX

NetworkX *is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. The package offers various functions, including:*

- *Data structures for graphs, digraphs, and multigraphs*
- *Many standard graph algorithms*
- *Network structure and analysis measures*
- *Generators for classic graphs, random graphs, and synthetic networks*
- *Nodes can be "anything" (e.g., text, images, XML records)*
- *Edges can hold arbitrary data (e.g., weights, time-series)*




```
pip install networkx
```

05

BIG DATA
PROCESSING

05

Polars

Polars *is a DataFrame library designed to processing data with a fast lighting time by implementing Rust Programming language and using Arrow as the foundation. Polars premise is to give the users a swifter experience in comparison to Pandas package. The ideal situation to use the Polars package is when you have data that were too big for Pandas but too small for using Spark.*

For you who familiar with the Pandas workflow, Polars would not be that different — there is some extra functionality, but overall they are pretty similar.

```
[<class 'polars.datatypes.Utf8'>, <class 'polars.datatypes.Int64'>, <class 'polars.datatypes.Int64'>]
['Patient', 'Weight', 'Segment']
shape: (5, 3)
```

Patient	Weight	Segment
str	i64	i64
"Anna"	41	1
"Be"	56	2
"Charlie"	78	1
"Duke"	55	1
"Earth"	80	3

```
pip install polars
```

```
shape: (4, 3)
```

Patient	Weight	Segment
str	i64	i64
"Anna"	41	1
"Be"	56	2
"Duke"	55	1
"Goal"	36	1

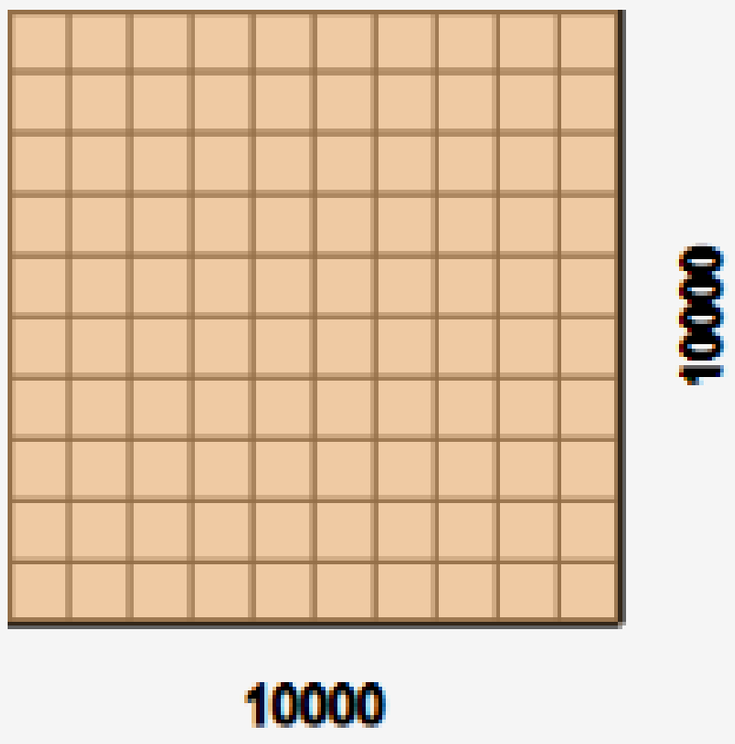
Dask

Dask *is a Python package for parallel computing in Python. There are two main parts in Dask, there are:*

1. *Task Scheduling. Similar to Airflow, it is used to optimized the computation process by automatically executing tasks.*
2. *Big Data Collection. Parallel data frame like Numpy arrays or Pandas data frame object – specific for parallel processing.*

In simpler terms, Dask offers a data frame or array object like you could find in Pandas, but it is processed in parallel for faster execution time, and it offers a task scheduler.

	Array	Chunk
Bytes	800.00 MB	8.00 MB
Shape	(10000, 10000)	(1000, 1000)
Count	100 Tasks	100 Chunks
Type	float64	numpy.ndarray



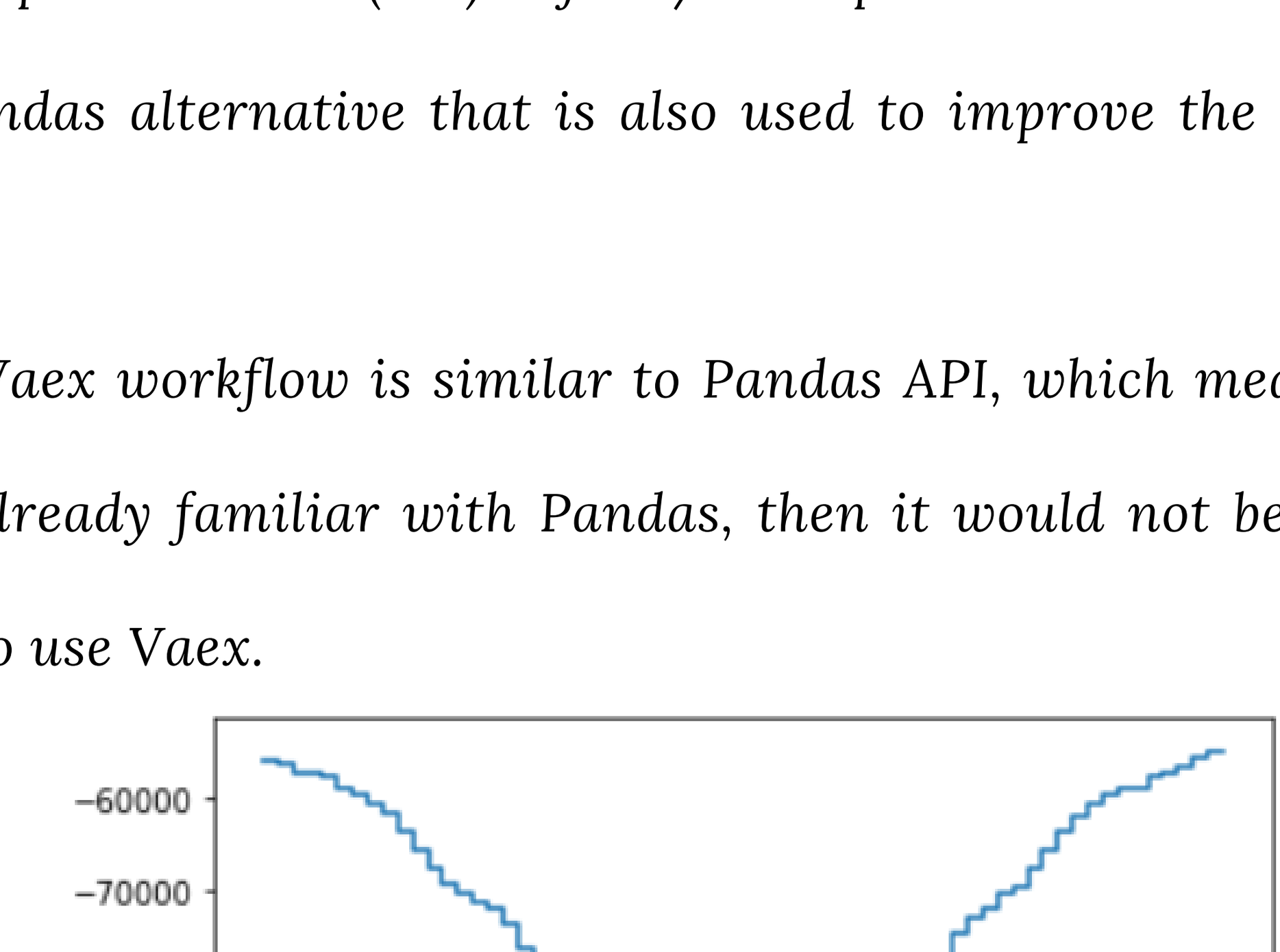

```
#If you want to install dask completely
python -m pip install "dask[complete]"

#If you want to install dask core only
python -m pip install dask
```

Vaex

Vaex *is a Python package used for processing and exploring big tabular datasets with interfaces similar to Pandas. Vaex documentation shows that it can calculate statistics such as mean, sum, count, standard deviation, etc., on an N-dimensional grid up to a billion (109) objects/rows per second. It means Vaex is Pandas alternative that is also used to improve the execution time.*

The Vaex workflow is similar to Pandas API, which means if you are already familiar with Pandas, then it would not be hard for you to use Vaex.




```
pip install vaex
```

06

MACHINE
LEARNING

06

Scikit-Learn

The king of Machine Learning modeling in Python. There is no way I would omit **Scikit-Learn** *in my list as your learning references. If, for some reason, you never heard about Scikit-Learn, this module is an open-source Python library for machine learning built on top of SciPy.*

Scikit-Learn contains all the common Machine Learning models we use in our everyday data science work. According to the homepage, Scikit-learn supports supervised and unsupervised learning modeling. It also provides various tools for model fitting, data preprocessing, selection and evaluation, and many other utilities.

User Guide

1. Supervised learning
2. Unsupervised learning
3. Model selection and evaluation
4. Inspection
5. Visualizations
6. Dataset transformations
7. Dataset loading utilities
8. Computing with scikit-learn
9. Model persistence
10. Common pitfalls and recommended practices

```
pip install -U scikit-learn
```

MLFlow

The current state of Machine Learning education is not limited to the machine learning model, but it is expanded into the automation process of the model. This is what we call MLOps or Machine Learning Operations.

Many open-source Python packages support the MLOps lifecycle, but in my opinion, ***MLflow*** *has a complete MLOps learning material for any beginner.*

According to the MLFlow homepage, MLflow is an open-source platform for managing the end-to-end machine learning lifecycle. This package handles four functions, they are:

- ***Experiments tracking*** *(MLflow Tracking),*
- ***ML code reproducible*** *(MLflow Projects),*
- ***Managing and deploying models*** *(MLflow Models),*
- ***Model central lifecycle*** *(MLflow Model Registry).*

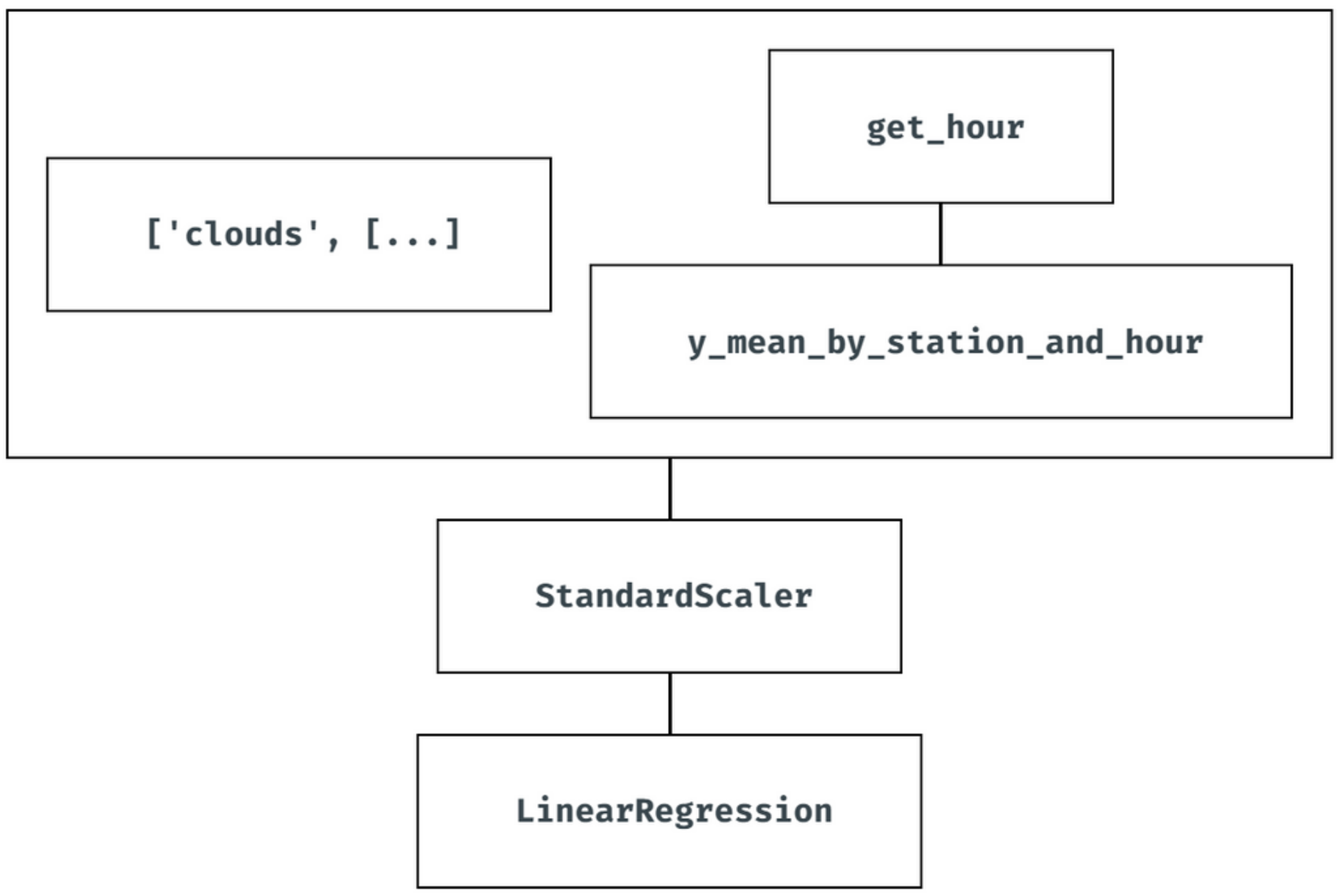
```
pip install mlflow
```

River

River *is a library for building online machine learning models.*

Such models operate on data streams. But a data stream is a bit of a vague concept.

River is not the only library allowing you to do online machine learning. But it might just the simplest one to use in the Python ecosystem. River plays nicely with Python dictionaries, therefore making it easy to use in the context of web applications where JSON *payloads are aplenty.*

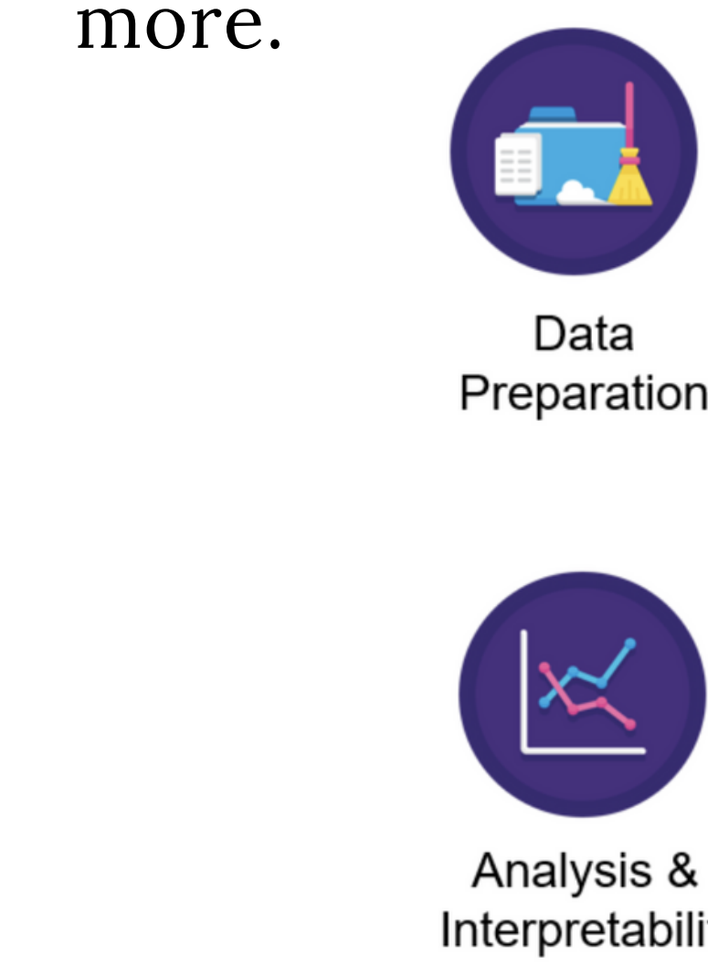



```
pip install river
```

PyCaret

PyCaret *is an open-source, low-code machine learning library in Python that automates machine learning workflows.* It is an end-to-end machine learning and model management tool that exponentially speeds up the experiment cycle and makes you more productive.

Compared with the other open-source machine learning libraries, PyCaret is an alternate low-code library that can be used to replace hundreds of lines of code with a few lines only. This makes experiments exponentially fast and efficient. PyCaret is essentially a Python wrapper around several machine-learning libraries and frameworks, such as scikit-learn, XGBoost, LightGBM, CatBoost, spaCy, Optuna, Hyperopt, Ray, and a few more.

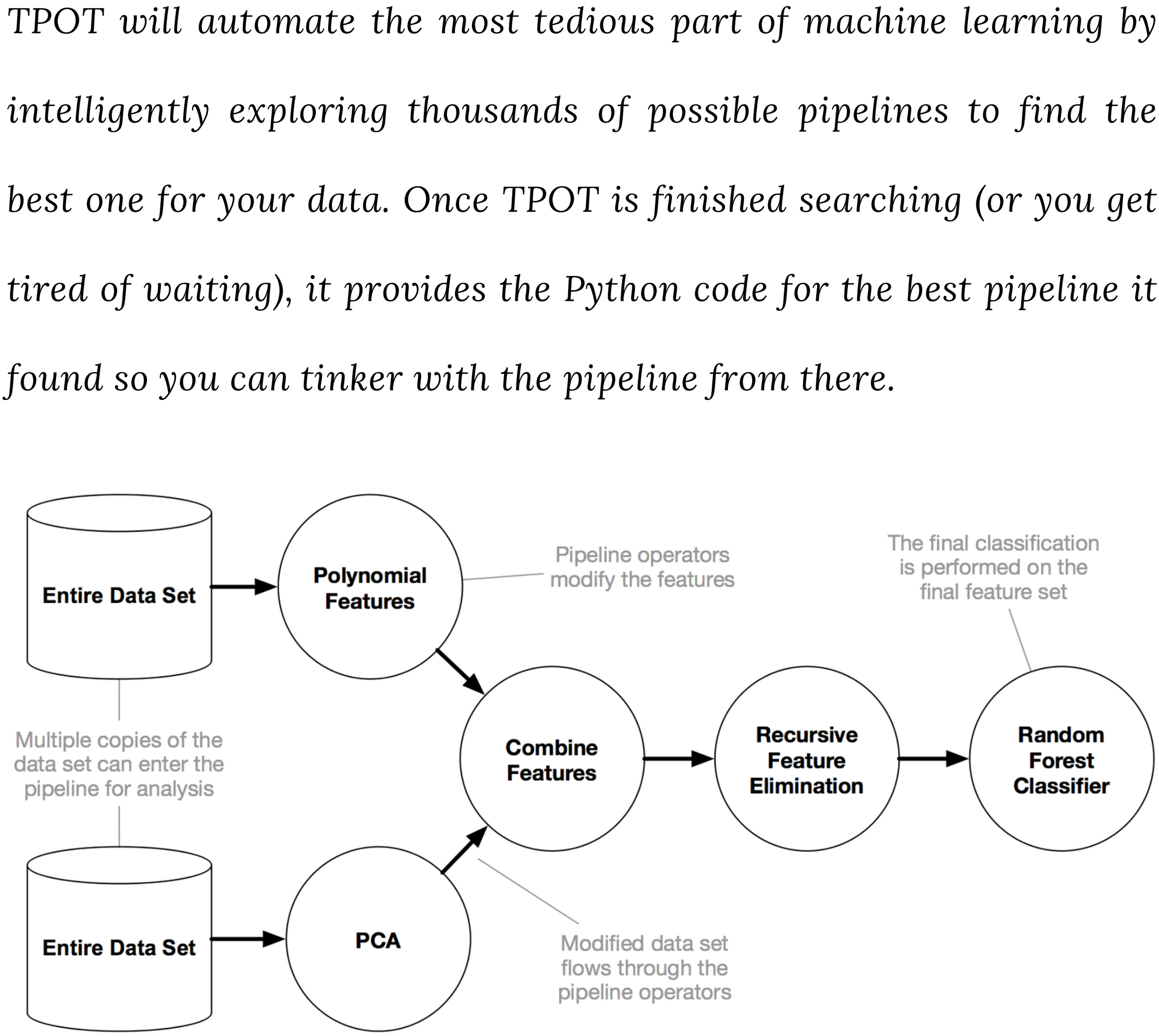



```
pip install pycaret
```

TPOT

TPOT *stands for Tree-based Pipeline Optimization Tool. Consider TPOT as your Data Science Assistant. TPOT is a Python Automated Machine Learning tool that optimizes machine learning pipelines using genetic programming.*

TPOT will automate the most tedious part of machine learning by intelligently exploring thousands of possible pipelines to find the best one for your data. Once TPOT is finished searching (or you get tired of waiting), it provides the Python code for the best pipeline it found so you can tinker with the pipeline from there.




```
pip install tpot
```

EvalML

EvalML *is an AutoML library that builds, optimizes, and evaluates machine learning pipelines using domain-specific objective functions.*

Key Functionality

- *Automation - Makes machine learning easier. Avoid training and tuning models by hand. Includes data quality checks, cross-validation, and more.*
- *Data Checks - Catches and warns of problems with your data and problem setup before modeling.*
- *End-to-end - Constructs and optimizes pipelines that include state-of-the-art preprocessing, feature engineering, feature selection, and a variety of modeling techniques.*
- *Model Understanding - Provides tools to understand and introspect on models, to learn how they'll behave in your problem domain.*
- *Domain-specific - Includes repository of domain-specific objective functions and an interface to define your own.*

```
pip install evalml
```

07

INTERPRETABILITY

07

Eli5

There are many advanced ML Interpretation Python Package out there, but most of them are too specific which devoid of any learning opportunities. In this case, I recommended **Eli5** *for your Machine Learning interpretability study package as it offers all the basic concepts without many complicated concepts.*

Taken from the Eli5 package, the basic usage of this package is to:

1. *inspect model parameters and try to figure out how the model works globally;*
2. *inspect an individual prediction of a model and figure out why the model makes the decision.*

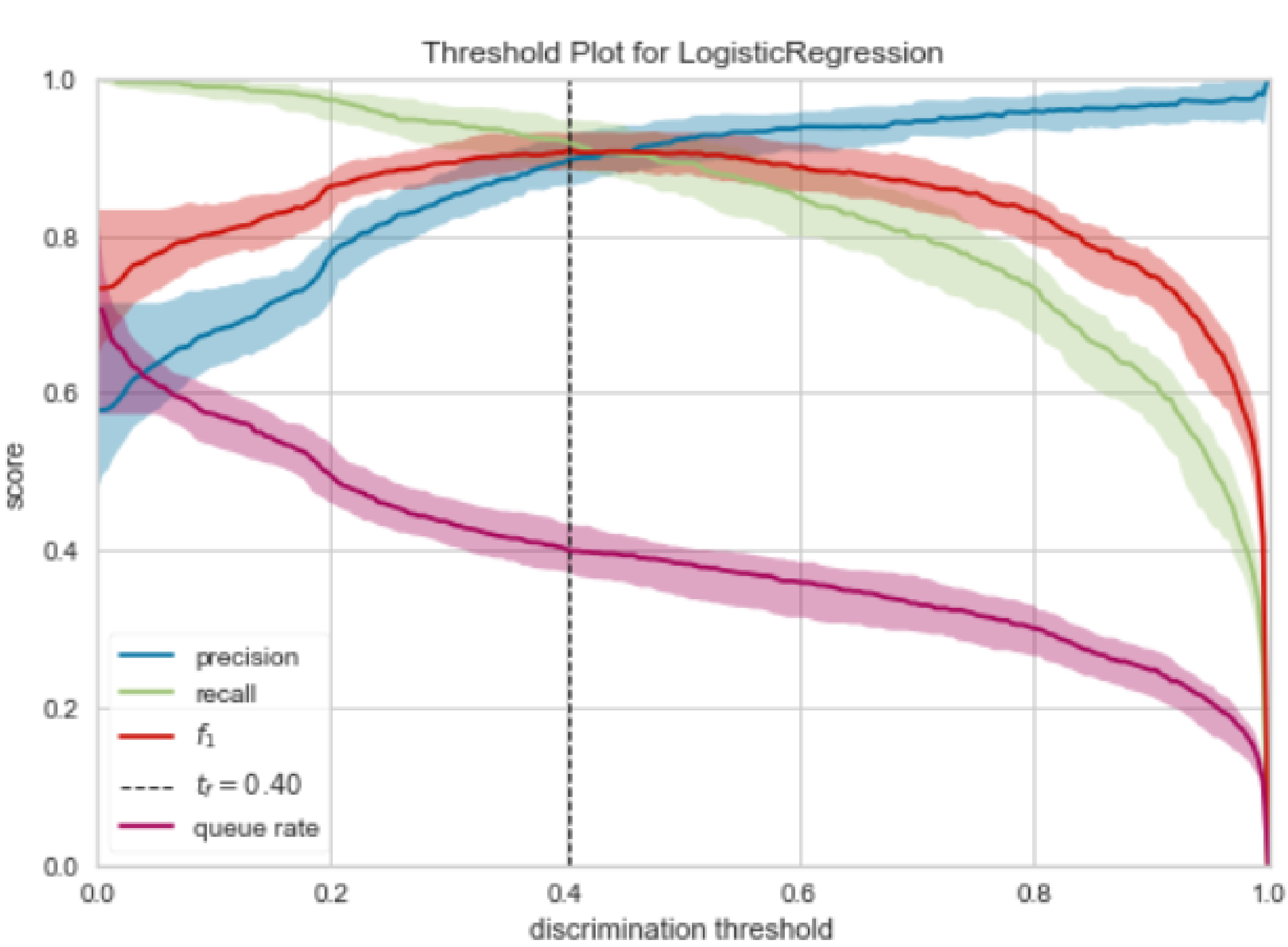
Weight	Feature
0.3797 ± 0.0577	displacement
0.0785 ± 0.0405	weight
0.0329 ± 0.0441	horsepower
0.0127 ± 0.0160	cylinders
0.0101 ± 0.0372	acceleration
-0.0051 ± 0.0124	model_year
-0.0152 ± 0.0405	mpg

```
pip install eli5
```

Yellowbrick

Yellowbrick *is an open-source Python package that extends the scikit-learn API with visual analysis and diagnostic tools. For Data scientists, Yellowbrick is used to evaluate the model performance and visualize the model behavior.*

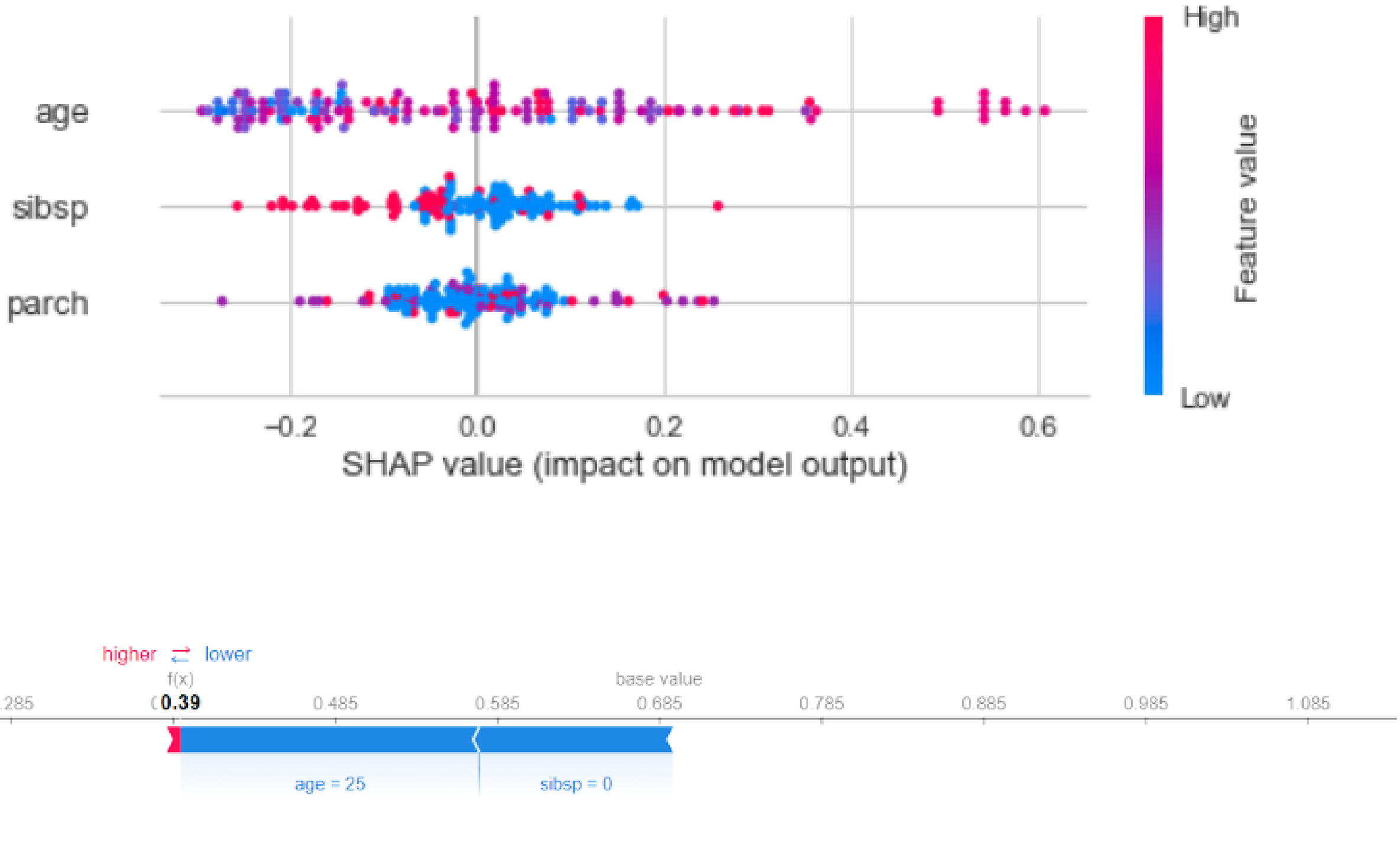
Yellowbrick is a multi-purpose package that you could use in your everyday modeling work. Even though most of the interpretation API from the Yellowbrick is at the basic level, it is still useful for our first modeling steps.




```
pip install yellowbrick
```

SHAP

SHAP *or (SHapley Additive exPlanations) is a game-theoretic approach to explain the output of any machine learning model. In a simpler term, SHAP uses the SHAP values to explain the importance of each feature. SHAP uses the SHAP values difference between the prediction of the model and the null model developed. SHAP is model agnostic, similar to the Permutation Importance, so it is useful for any model.*

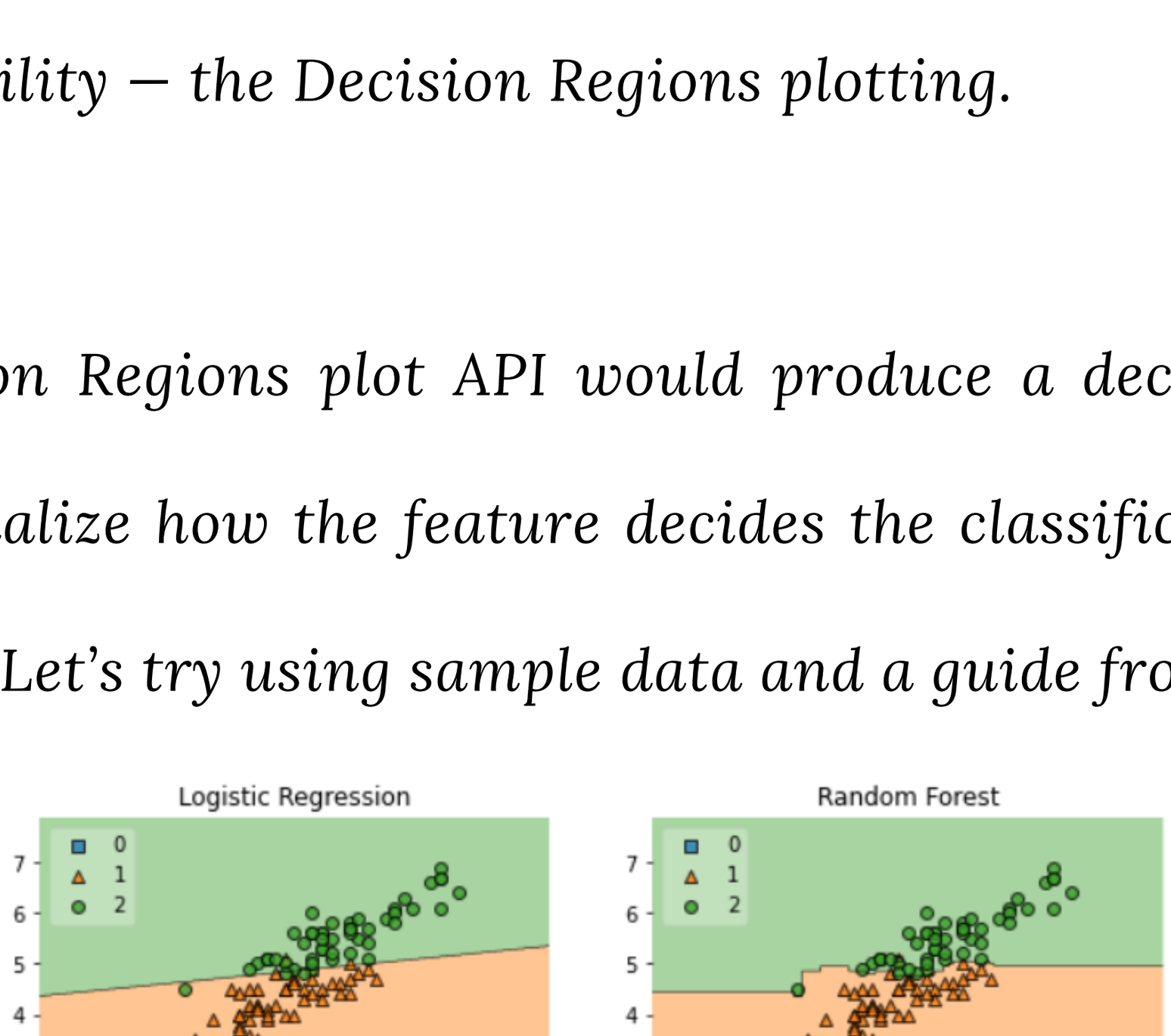



```
pip install shap
```

Mlxtend

Mlxtend *or machine learning extensions is a Python package for data science everyday work life. The APIs within the package are not limited to interpretability but extend to various functions, such as statistical evaluation, Data Pattern, Image Extraction, and many more. However, we would discuss the API for interpretability — the Decision Regions plotting.*

The Decision Regions plot API would produce a decision region plot to visualize how the feature decides the classification model prediction. Let's try using sample data and a guide from Mlxtend.

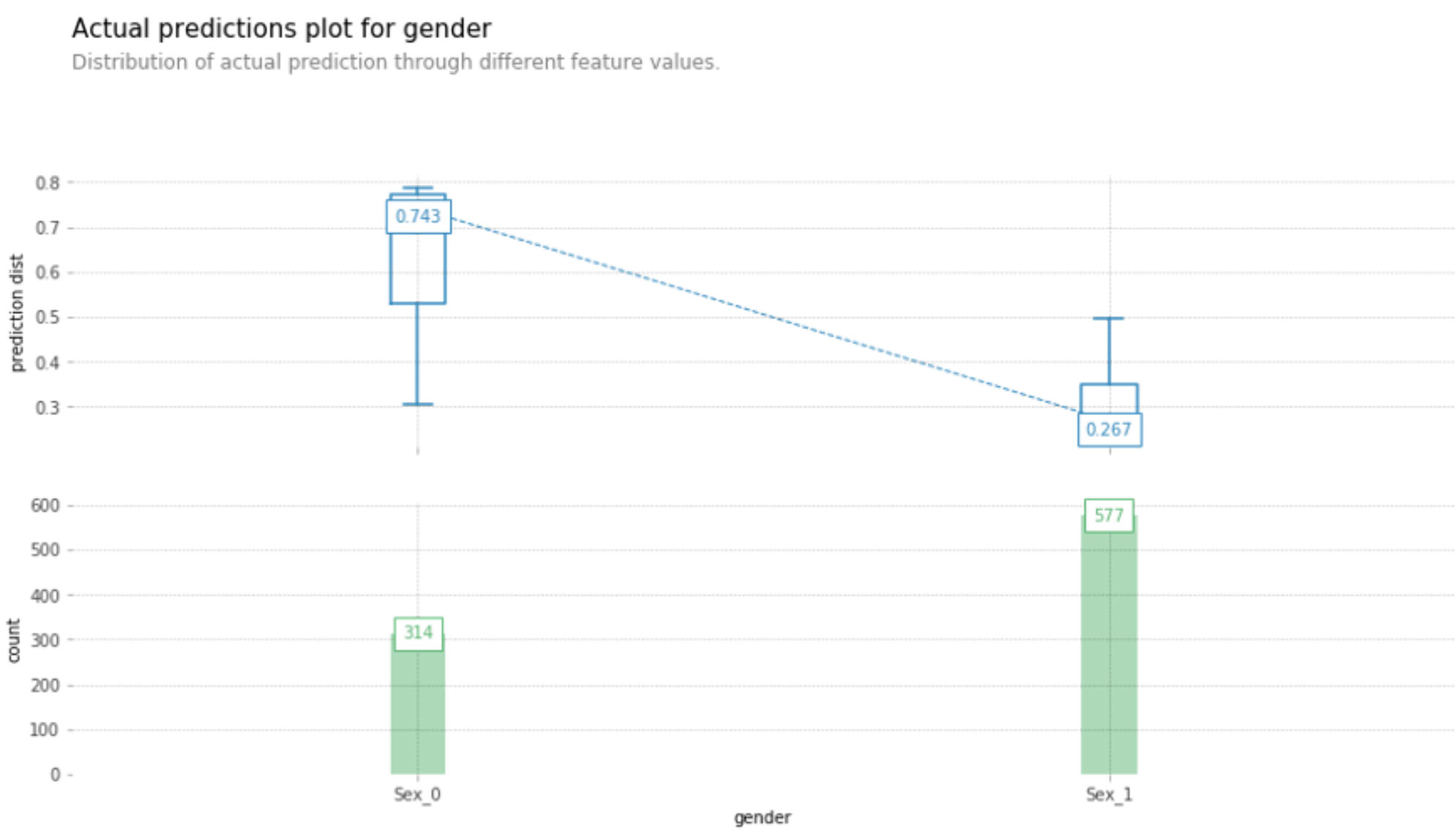



```
pip install Mlxtend
```

PDPBox

PDP *or Partial Dependence Plot is a plot that shows the marginal effect of features on the predicted outcome of the machine learning model. It is used to evaluate whether the correlation between the feature and target is linear, monotonic, or more complex.*

The advantage of interpreting with a Partial Dependence plot is that it is easy to interpret for business people. The calculation for the partial dependence plots has a causal interpretation when we intervene on a feature and measure the changes in the predictions; this is when we could measure the interpretation.

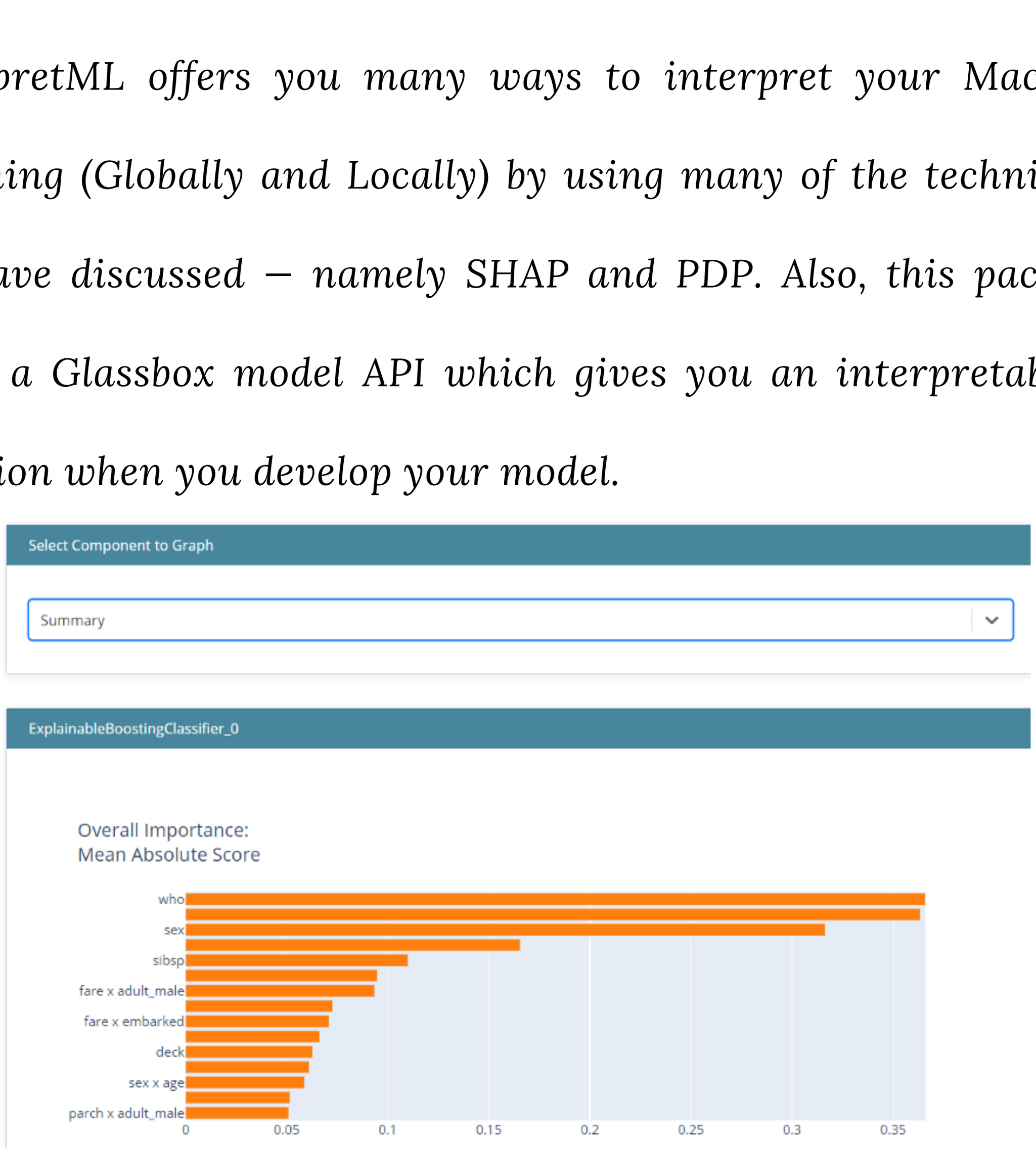



```
pip install pdpbox
```

InterpretML

InterpretML *is a Python Package that includes many Machine Learning Interpretability APIs. The purpose of this package is to give you an interactive plot based on plotly to understand your prediction result.*

InterpretML offers you many ways to interpret your Machine Learning (Globally and Locally) by using many of the techniques we have discussed — namely SHAP and PDP. Also, this package owns a Glassbox model API which gives you an interpretability function when you develop your model.




```
pip install interpret
```

08

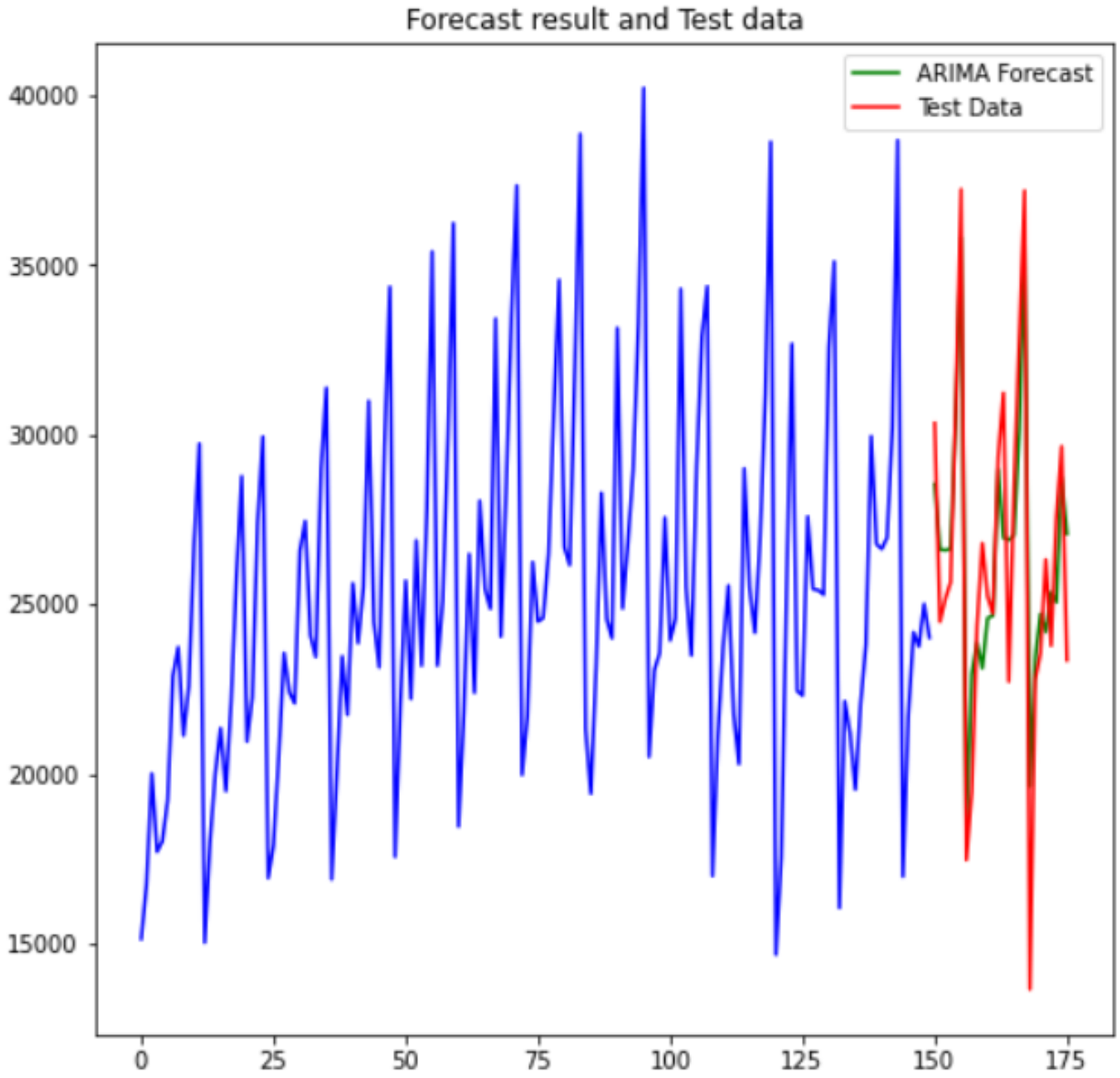
TIME SERIES

08

pmdarima

One of the forecasting models often used in the time-series analysis is ARIMA (*AutoRegressive Integrated Moving Average*). ARIMA *is a forecasting algorithm where we could predict future values based on the past values of the time series without any additional information.*

Pmdarima *is a statistical Python package that provides the* ARIMA API *and all the basic time-series analysis* API, *but we only try the Auto* ARIMA.

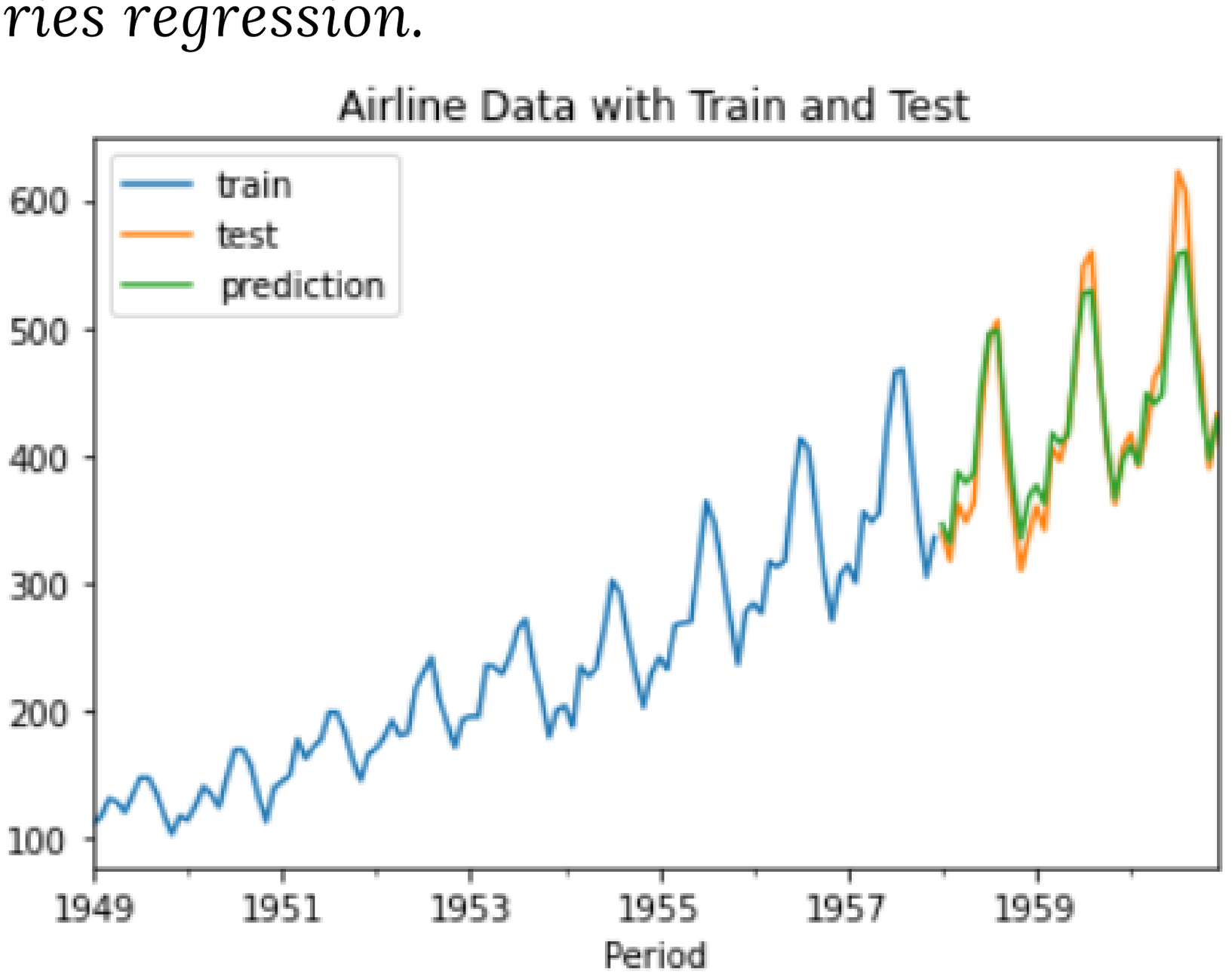



```
pip install pmdarima
```

sktime

Many people who learned machine learning with Python would use Sklearn as their starter point. The problem with Sklearn is that the package provides no time-series analysis module; this is why ***sktime*** *packages are developed.* According to the homepage, *sktime is specialized in time series algorithms and scikit-learn compatible tools, including:*

- *Forecasting,*
- *Time series classification,*
- *Time series regression.*

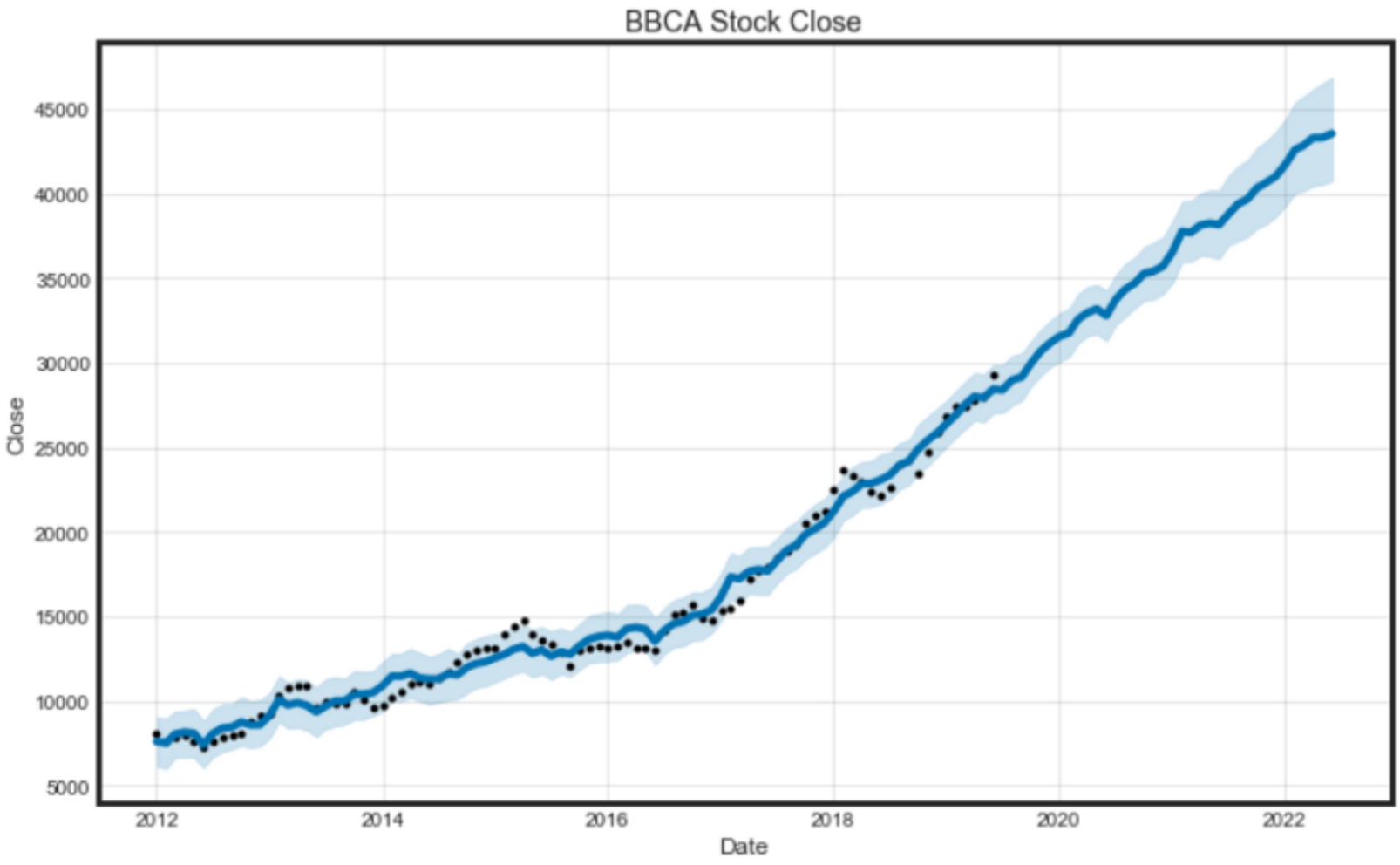



```
pip install sktime
```

fbprophet

*The **fbprophet** or prophet is a time-series analysis developed by the Facebook group. According to the homepage, fbprophet is a package to develop forecasting time series data based on an additive model where non-linear trends fit time seasonality with holiday effects.*

Fbprophet mentions that it works best with time series data with strong seasonal effects and several seasons of historical data. Also, fbprophet notes that it is robust to missing data and could handle outliers well. From the explanation, we could infer that fbprophet is a good package to model time data with high seasonality

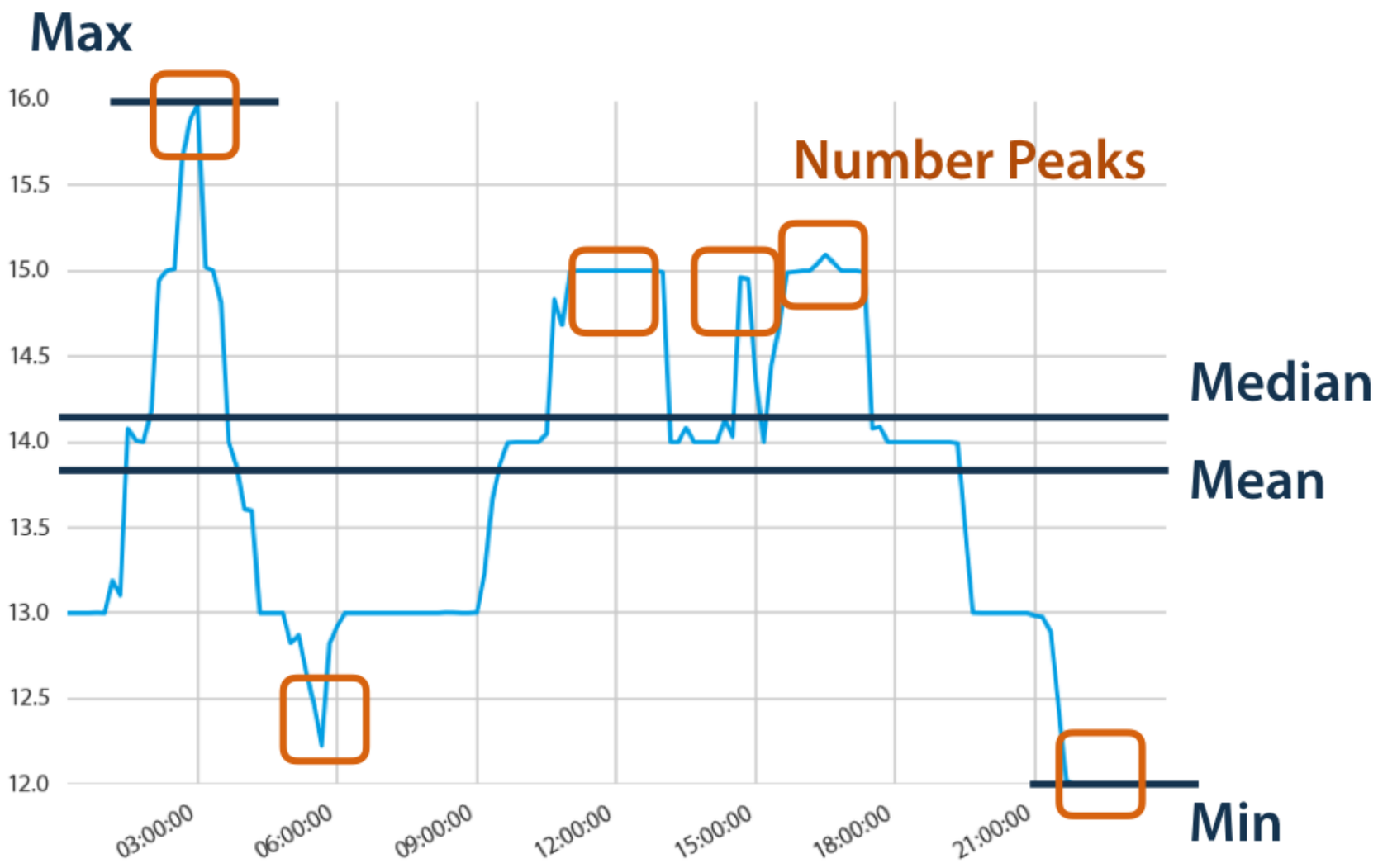



```
pip install pystan==2.19.1.1
pip install prophet
```

tsfresh

tsfresh *is a python package that automatically calculates a large number of time series characteristics, the so-called features. Further, the package contains methods to evaluate the power and importance of such characteristics for regression or classification tasks.*

The package provides systematic time-series feature extraction by combining established algorithms from statistics, time-series analysis, signal processing, and nonlinear dynamics with a robust feature selection algorithm. In this context, the term time-series is interpreted in the broadest possible sense, such that any types of sampled data or even event sequences can be characterised.

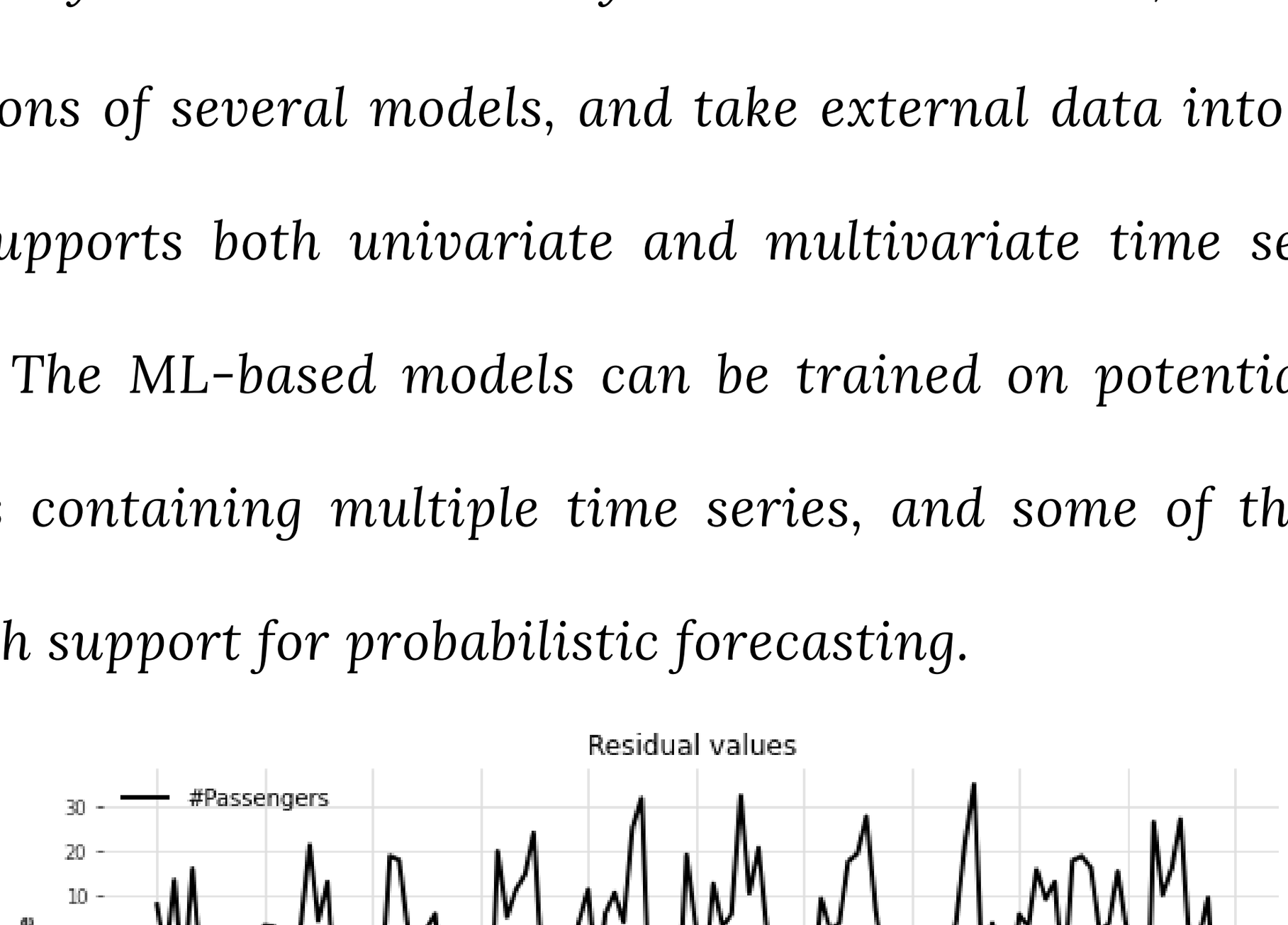



```
pip install tsfresh
```

darts

Darts *is a Python library for user-friendly forecasting and anomaly detection on time series. It contains a variety of models, from classics such as* ARIMA *to deep neural networks. The forecasting models can all be used similarly, using fit() and predict() functions, similar to scikit-learn.*

The library also makes it easy to backtest models, combine the predictions of several models, and take external data into account. Darts supports both univariate and multivariate time series and models. The ML-based models can be trained on potentially large datasets containing multiple time series, and some of the models offer rich support for probabilistic forecasting.




```
pip install darts
```

09

NLP

09

NLTK

Natural Language Toolkit, or **NLTK**, *is an open-source Python package developed specifically for human language. It is arguably the most-used package for beginners and professionals in the* NLP *field as* NLTK *offers many useful* APIs *in* NLP *research. According to the homepage,* NLTK *is suitable for any profession — linguist, data scientist, researcher, student, and many more.*

NLTK contains all the common APIs *we use in our everyday* NLP *everyday activities work. If we explore the homepage, we will find that* NLTK *provides various tools for parsing, stemming, tokenization, and many more. They also include* API *to read data from sources such as Twitter.*

```
pip install --user -U nltk
```

```
[('the', 62713),
 (',', 58334),
 ('.', 49346),
 ('of', 36080),
 ('and', 27915),
 ('to', 25732),
 ('a', 21881),
 ('in', 19536),
 ('that', 10237),
 ('is', 10011)]
```

Pattern

Pattern *package is an open-source Python package developed for text processing and web-mining data. The API provides many functions, such as:*

- *Data Mining API from various sources (Google, Twitter, and Wikipedia)*
- *NLP Processing*
- *Machine Learning Modelling*
- *Network Analysis*

If we compared the Pattern package to the NLTK package, the functions for the text processing within the Pattern are incomplete than NLTK. However, Pattern contains the web-mining data, which NLTK doesn't have. This is because Pattern packages were developed with a focus on data mining.

```
Computer Vision with Tensorflow (towardsdatascienc...)

Journey to Spirit Island.Image by author This will be a continuation from the introduction t...

Add your highlights:
https://t.co/1Fhvkbv297
#data #science #dat...
The 5 Biggest Data Science Trends In 2022 #MachineLearning #learning https://t.co/ngnE4GyJ8c
RT @AI4ESG: Celebrating World Red Cross Day 2021 with the new Australian Red Cross Volunteer Data Science team. Together, we ar
e #unstoppable!

https://t.co/ngy747vKD7
RT @commonslibrary: 📢We're recruiting a data science lead! This role will play a vital part in building our #datascience capab
ility and improving our services.

Salary: £50,870
Closing date: 17/10/21

Apply here: https://t.co/6EQXnmVYKi

Find out more about working with us: https://t.co/bMALsppEEf
RT @AI4ESG: COVID-19 has become a global challenge and data science became one of the strategies to leverage the fight against
this global crisis.
```

```
pip install pattern
```

TextBlob

TextBlob *is a Python text processing package that provides many APIs to easing the NLP project tasks. TextBlob was built on top of the NLTK and Pattern packages, which means you would find many of the familiar APIs.*

TextBlob stands out for how beginner-friendly this package is to do NLP activity with their simple API. TextBlob package was developed specifically for NLP tasks such as tagging, translation, sentiment analysis, and more in a simple way.

	Google Search	Polarity	Subjectivity
0	Data science is an interdisciplinary field tha...	0.000000	0.000000
1	Offered by Johns Hopkins University. Launch Yo...	0.100000	0.300000
2	Data science continues to evolve as one of the...	0.400000	0.500000
3	Data science has become a necessary leading te...	0.000000	0.500000
4	Insight Data Science Fellows Program · An inte...	0.000000	0.000000
5	... Data science combines the scientific metho...	0.400000	0.600000
6	This book will teach you how to do data scienc...	0.400000	0.250000
7	The simplest definition of data science is the...	-0.230769	0.461538
8	Data science is the process of using advanced ...	0.137500	0.487500
9	With cross-disciplinary research and innovativ...	0.346591	0.551136

```
pip install -U textblob
python -m textblob.download_corpora
```

SpaCy

SpaCy *is an Open-source Python package with a tagline for Industrial-Strength Natural Language Processing. It means that SpaCy is developed for production environment and industrial activity than for academic purposes.*

Although it is developed for Industrial purposes, the tutorial and documentation they have are quite complete. The pages offer you the guide, lesson, and online video to learn NLP from the beginning and use SpaCy. For example, the SpaCy 101 Guide would let you know many subjects such as Linguistic annotations, Tokenization, POS tags and dependencies, Vocab, hashes, and lexemes, and many more.

	TEXT	LEMMA	POS	TAG	DEP	SHAPE	ALPHA	STOP
0	Data	datum	NOUN	NNS	compound	Xxxx	True	False
1	science	science	NOUN	NN	nsubj	xxxx	True	False
2	is	be	AUX	VBZ	ROOT	xx	True	True
3	an	an	DET	DT	det	xx	True	True
4	interdisciplinary	interdisciplinary	ADJ	JJ	amod	xxxx	True	False
5	field	field	NOUN	NN	attr	xxxx	True	False
6	that	that	DET	WDT	nsubj	xxxx	True	True
7	uses	use	VERB	VBZ	relcl	xxxx	True	False
8	scientific	scientific	ADJ	JJ	amod	xxxx	True	False
9	methods	method	NOUN	NNS	dobj	xxxx	True	False
10	,	,	PUNCT	,	punct	,	False	False

```
pip install -U pip setuptools wheel
pip install -U spacy
python -m spacy download en_core_web_sm
```

FastText

FastText fastText is a library for efficient learning of word representations and sentence classification. The package is developed by Facebook Researcher,.

The features including:

- Recent state-of-the-art English word vectors.
- Word vectors for 157 languages trained on Wikipedia and Crawl.
- Models for language identification and various supervised tasks.

Table of models

Each entry describes the test accuracy and size of the model. You can click on a table cell to download the corresponding model.

dataset	ag news	amazon review full	amazon review polarity	dbpedia
regular	0.924 / 387MB	0.603 / 462MB	0.946 / 471MB	0.986 / 427MB
compressed	0.92 / 1.6MB	0.599 / 1.6MB	0.93 / 1.6MB	0.984 / 1.7MB

dataset	sogou news	yahoo answers	yelp review polarity	yelp review full
regular	0.969 / 402MB	0.724 / 494MB	0.957 / 409MB	0.639 / 412MB
compressed	0.968 / 1.4MB	0.717 / 1.6MB	0.957 / 1.5MB	0.636 / 1.5MB

```
pip install darts
```

10

RECOMMENDATION SYSTEM

10

Surprise

Surprise *is an open-source Python package for building a recommendation system based on the rating data. The name SurPRISE is an abbreviation for the Simple Python RecommendatIon System Engine. The package provides all the necessary tools for building the recommendation system — from loading the dataset, choosing the prediction algorithm, and evaluating the model.*

Using prediction algorithms

Surprise provides a bunch of built-in algorithms. All algorithms derive from the `AlgoBase` base class, where are implemented some key methods (e.g. `predict`, `fit` and `test`). The list and details of the available prediction algorithms can be found in the `prediction_algorithms` package documentation.

Every algorithm is part of the global Surprise namespace, so you only need to import their names from the Surprise package, for example:

```
from surprise import KNNBasic
algo = KNNBasic()
```

Some of these algorithms may use baseline estimates, some may use a similarity measure. We will here review how to configure the way baselines and similarities are computed.

Baselines estimates configuration

Note

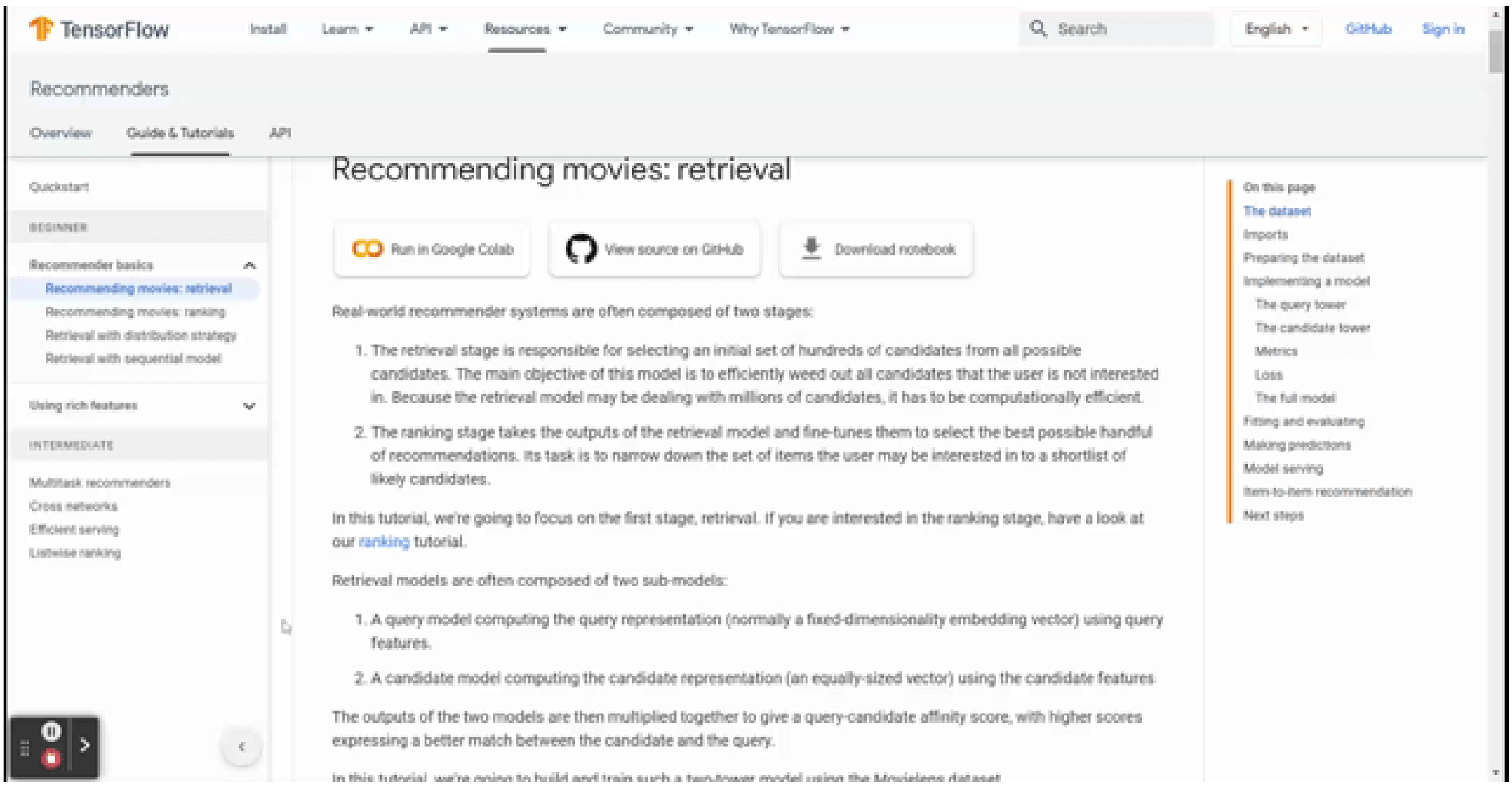
This section only applies to algorithms (or similarity measures) that try to minimize the following regularized squared error (or equivalent):

$$\sum_{r_{ui} \in R_{train}} \left(r_{ui} - (\mu + b_u + b_i)\right)^2 + \lambda \left(b_u^2 + b_i^2\right) .$$

```
pip install scikit-surprise
```

TensorFlow Recommenders

The ***TensorFlow framework*** *contains a library to build the recommendation system called TensorFlow Recommenders. Like the other package, the TensorFlow Recommenders contains dataset examples, recommender algorithms, model evaluations, and deployment.* TensorFlow Recommenders would allow us to build a recommendation system based only on the TensorFlow framework.



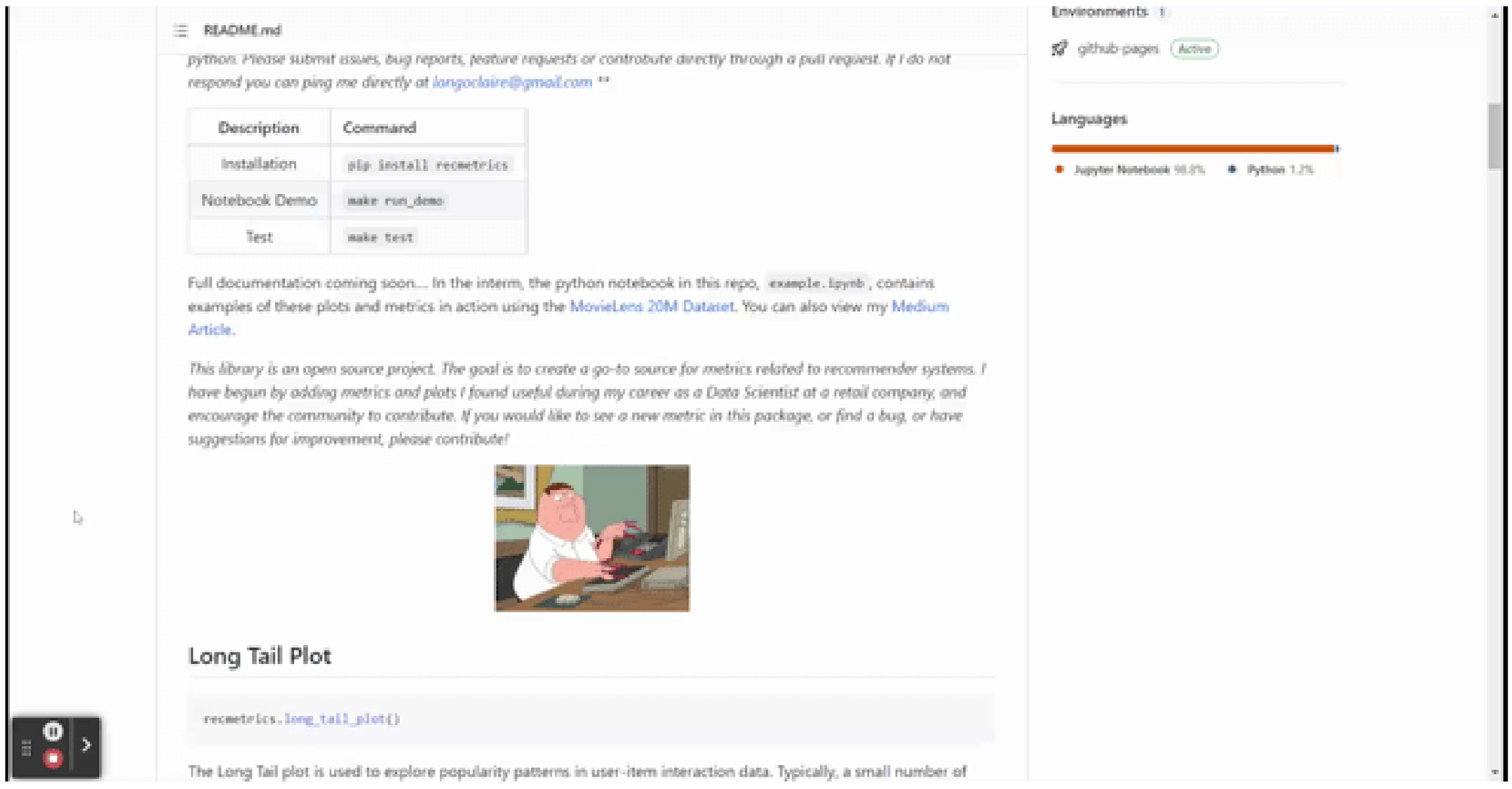

```
pip install tensorflow-recommenders
```

Recmetrics

Learning about the recommendation system algorithm would not be complete without the evaluation metrics. The previous pages I mentioned have taught us some basic recommendation evaluation metrics, but a Python package focuses on the metrics — **Recmetrics.**

The package contains many evaluation metrics for the recommendation system, such as:

- *Long Tail Plot*
- *Coverage*
- *Novelty*
- *Personalize*
- *Intra-List Similarity*




```
pip install recmetrics
```

11

AUDIO PROJECT

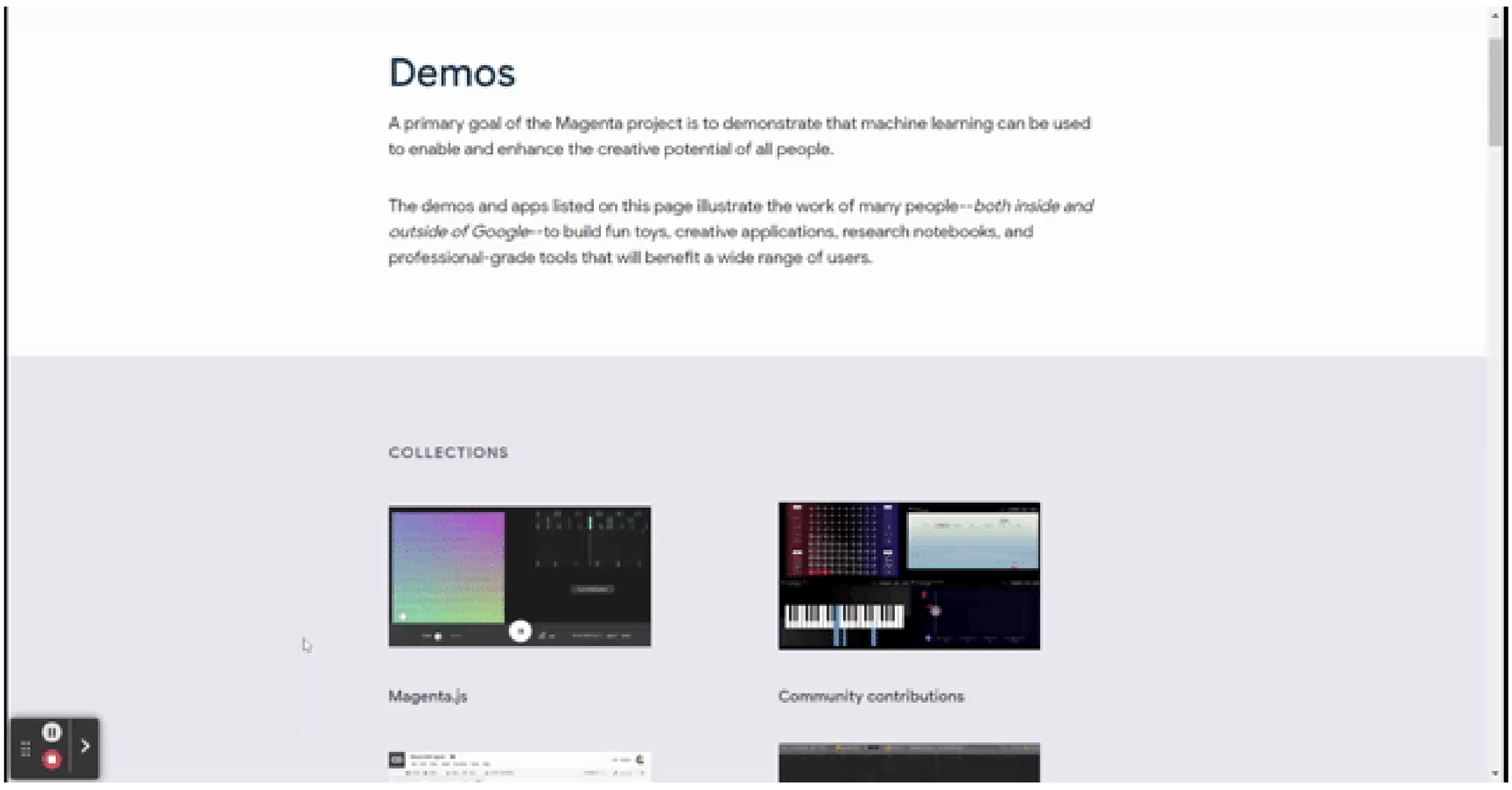
11

Magenta

Magenta *is an open-source Python package built on top of TensorFlow to manipulate image and music data to train a machine learning model with the generative model as the output.*

Magenta does not provide clear API references for us to learn; instead, they give a lot of research demos and collaborator notebooks we could try on our own.

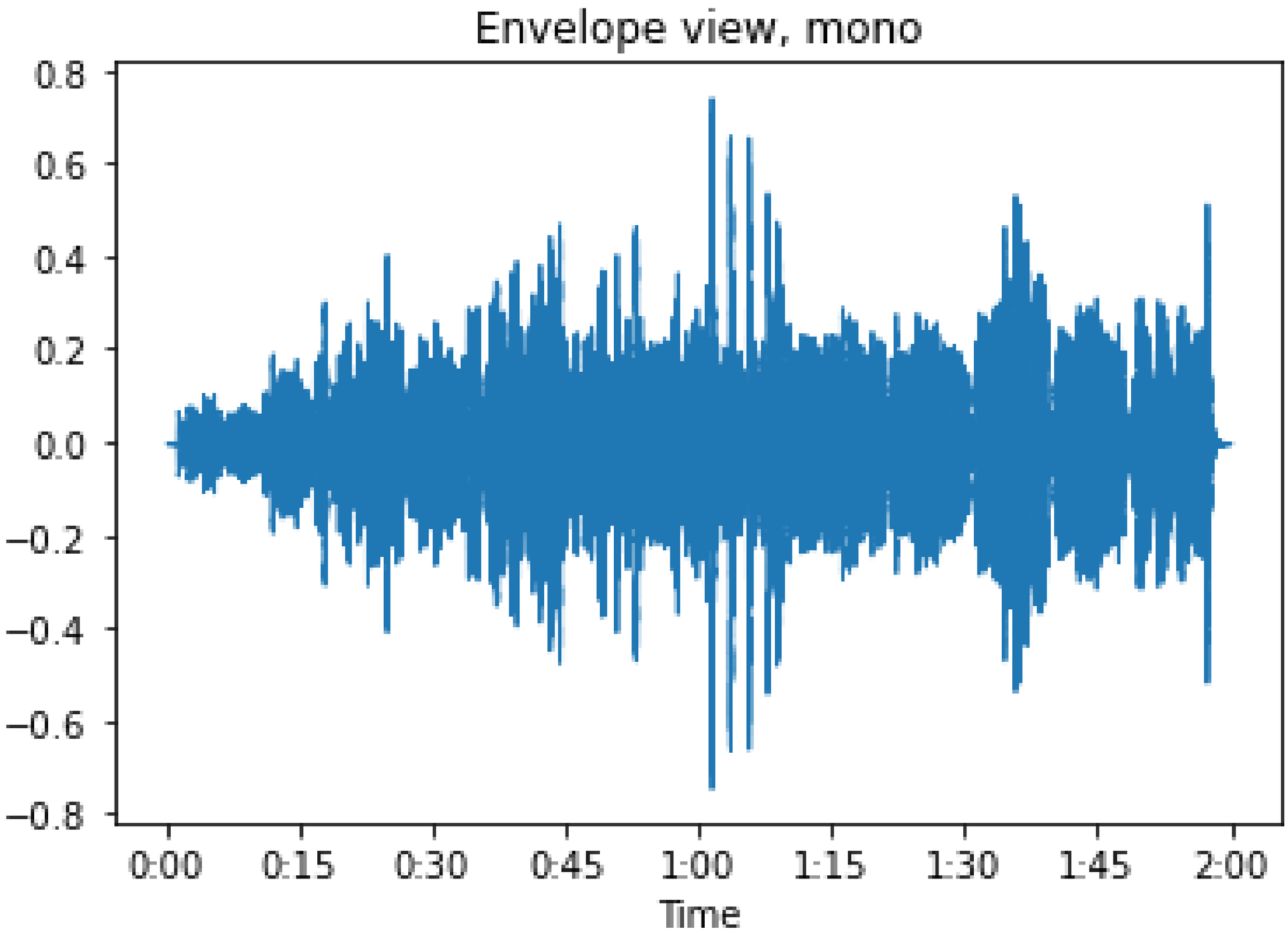
For a first-timer in the Audio Data Science project, I suggest visiting their Hello World notebook for music creation. I learned a lot from their Notebook, especially the part of generative machine learning where you could test various tones to produce your music.




```
pip install magenta
```

Librosa

Librosa *is a Python package developed for music and audio analysis. It is specific on capturing the audio information to be transformed into a data block. However, the documentation and example are good to understand how to work with audio data science projects*



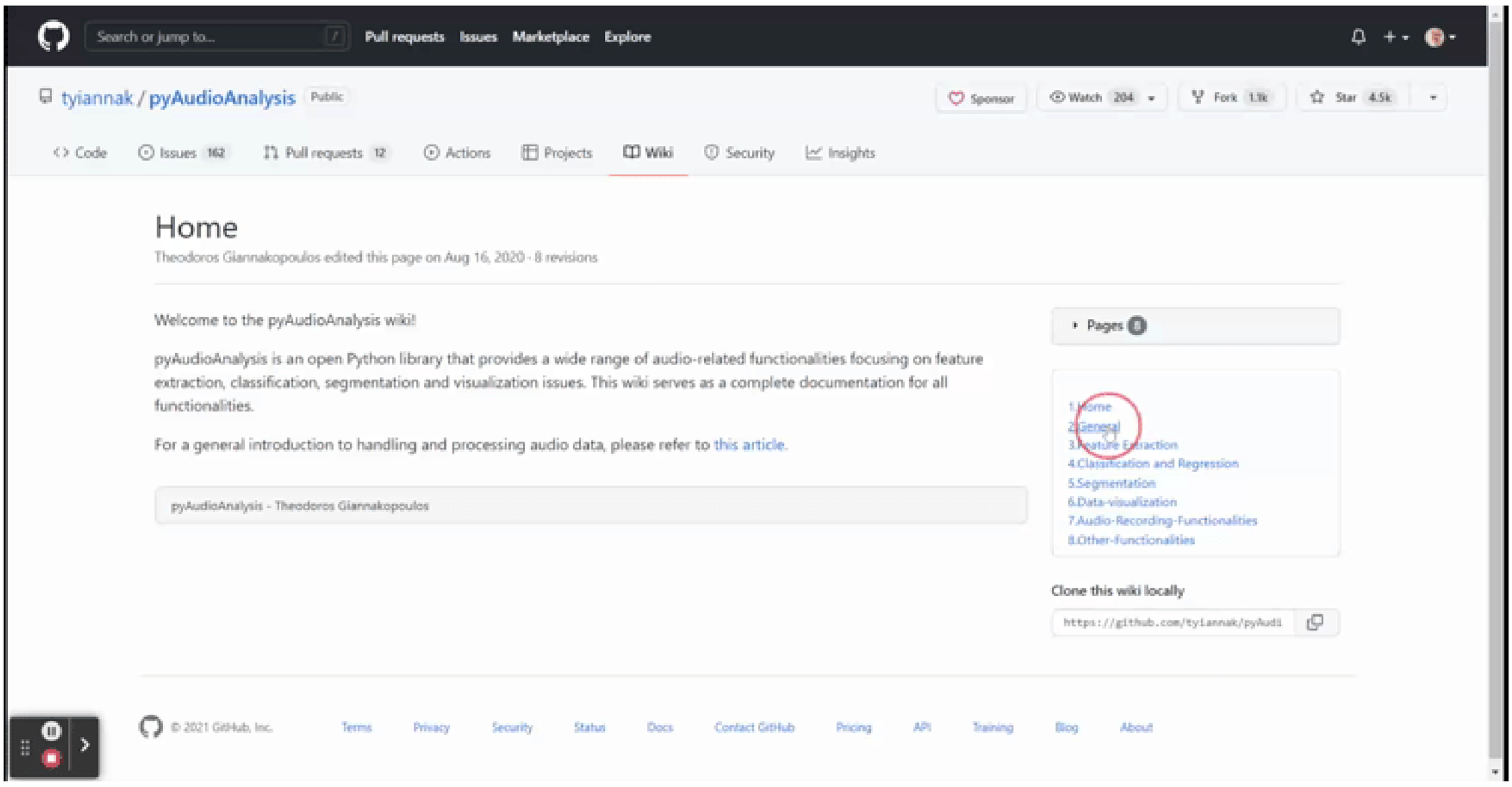

```
pip install librosa
```

pyAudioAnalysis

pyAudioAnalysis *is a Python package for audio analysis tasks. It is designed to do various analyses, such as:*

- *Extract Audio Features*
- *Train machine learning model for audio segmentation*
- *Classification of unknown audio*
- *Emotion recognition with a Regression model*
- *Dimensional Reduction for audio data visualization*

and many more. You could do many things with this package, especially if you are new to audio data science projects.




```
git clone https://github.com/tyiannak/pyAudioAnalysis.git
pip install -r ./requirements.txt
pip install -e .
```

12

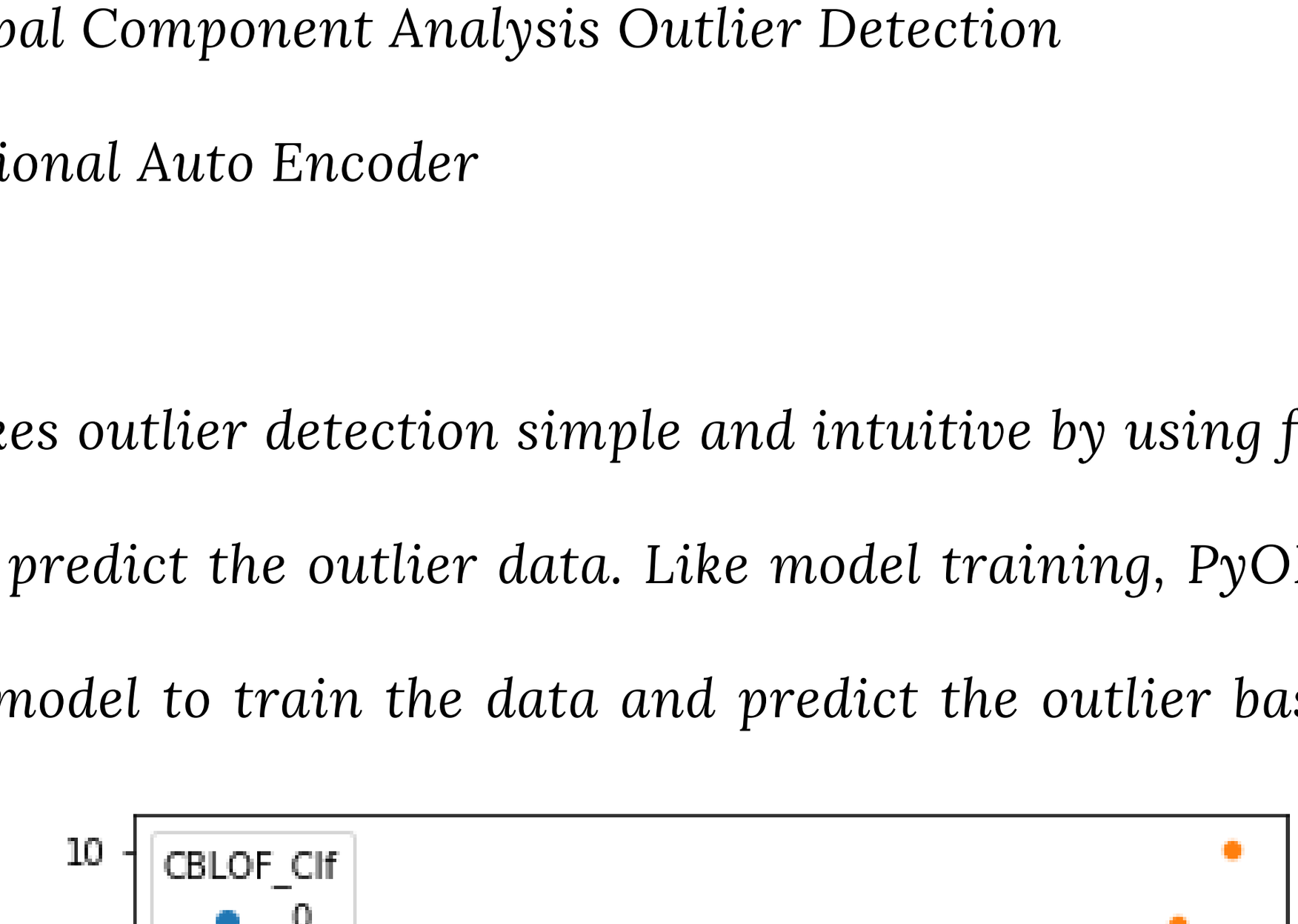
12

PyOD

PyOD *or Python Outlier Detection is a python package toolkit for detecting outlier data. PyOD package boasts 30 outlier detection algorithms, ranging from the classic to the most latest—proof PyOD package is well maintained. Examples of the outlier detection model include:*

- *Angle-Based Outlier Detection*
- *Cluster-Based Local Outlier Factor*
- *Principal Component Analysis Outlier Detection*
- *Variational Auto Encoder*

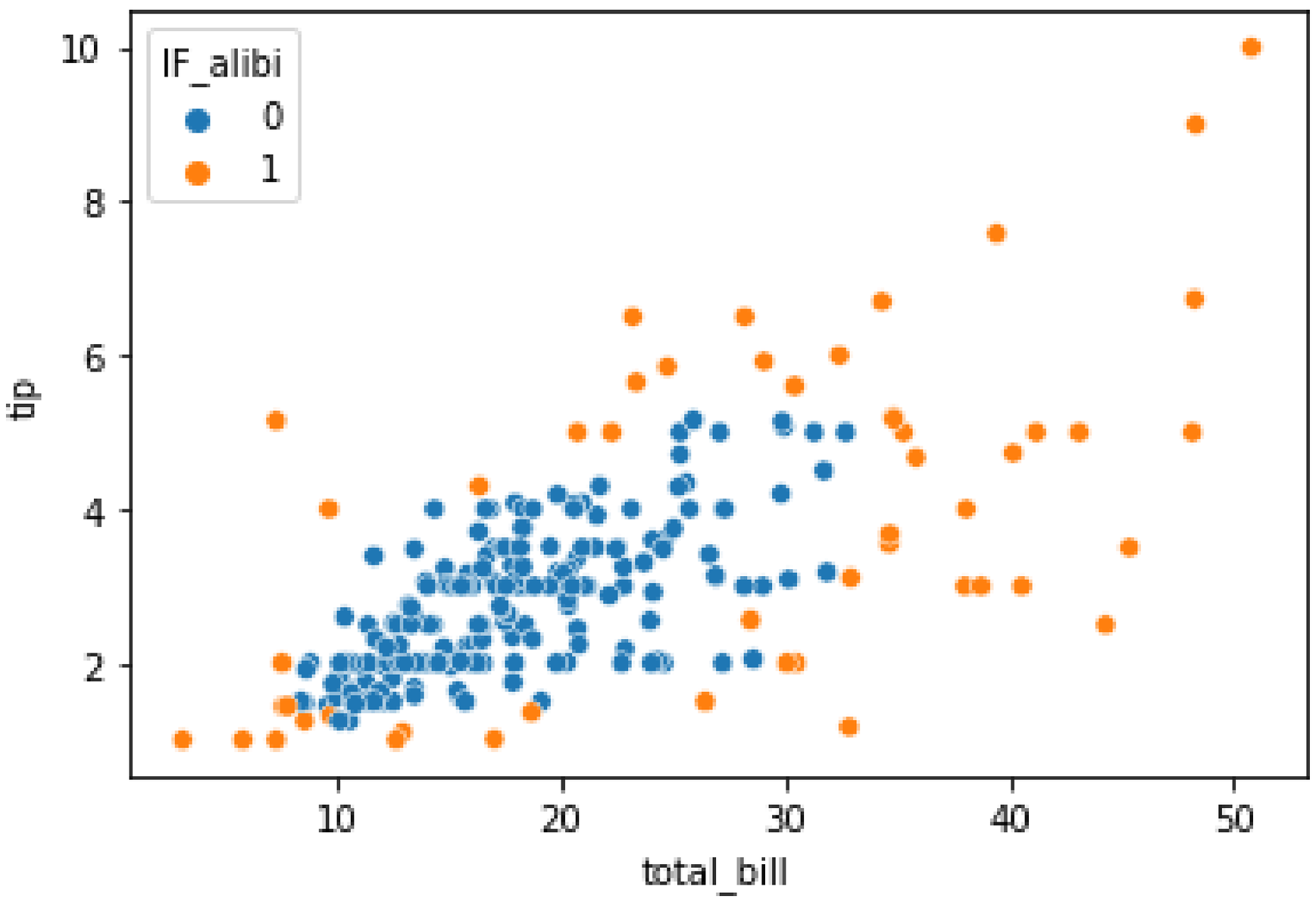
PyOD makes outlier detection simple and intuitive by using fewer lines of code to predict the outlier data. Like model training, PyOD uses the classifier model to train the data and predict the outlier based on the model.




```
pip install pyod
```

alibi-detect

*The **alibi-detect** python package is an open-source package that focuses on outlier, adversarial, and drift detection. This package could be used for tabular and unstructured data such as images or text.* The alibi-detect package offers 10 methods for outlier detection.

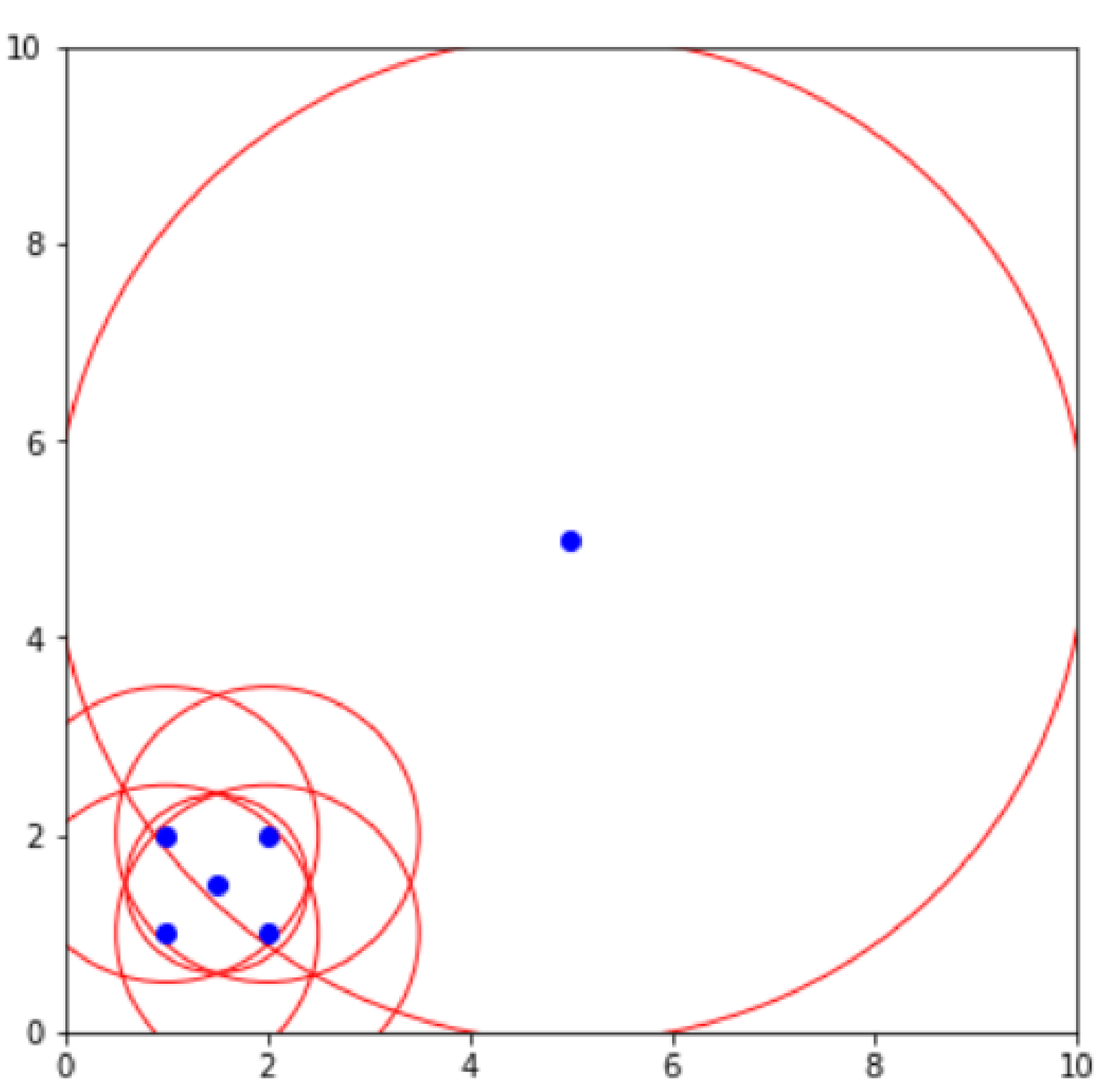



```
pip install alibi-detect
```

PyNomaly

PyNomaly *is a python package to detect outliers based on the* LoOP (*Local Outlier Probabilities*). *The* LoOP *is based on the Local Outlier Factor* (LOF), *but the scores are normalized to the range* [0–1].

Local Outlier Factor or LOF *is an algorithm proposed by Breunig et al.* (2000). *The concept is simple; the algorithm tries to find anomalous data points by measuring the local deviation of a given data point with respect to its neighbors. In this algorithm,* LOF *would yield a score that tells if our data is an outlier or not.*




```
pip install PyNomaly
```

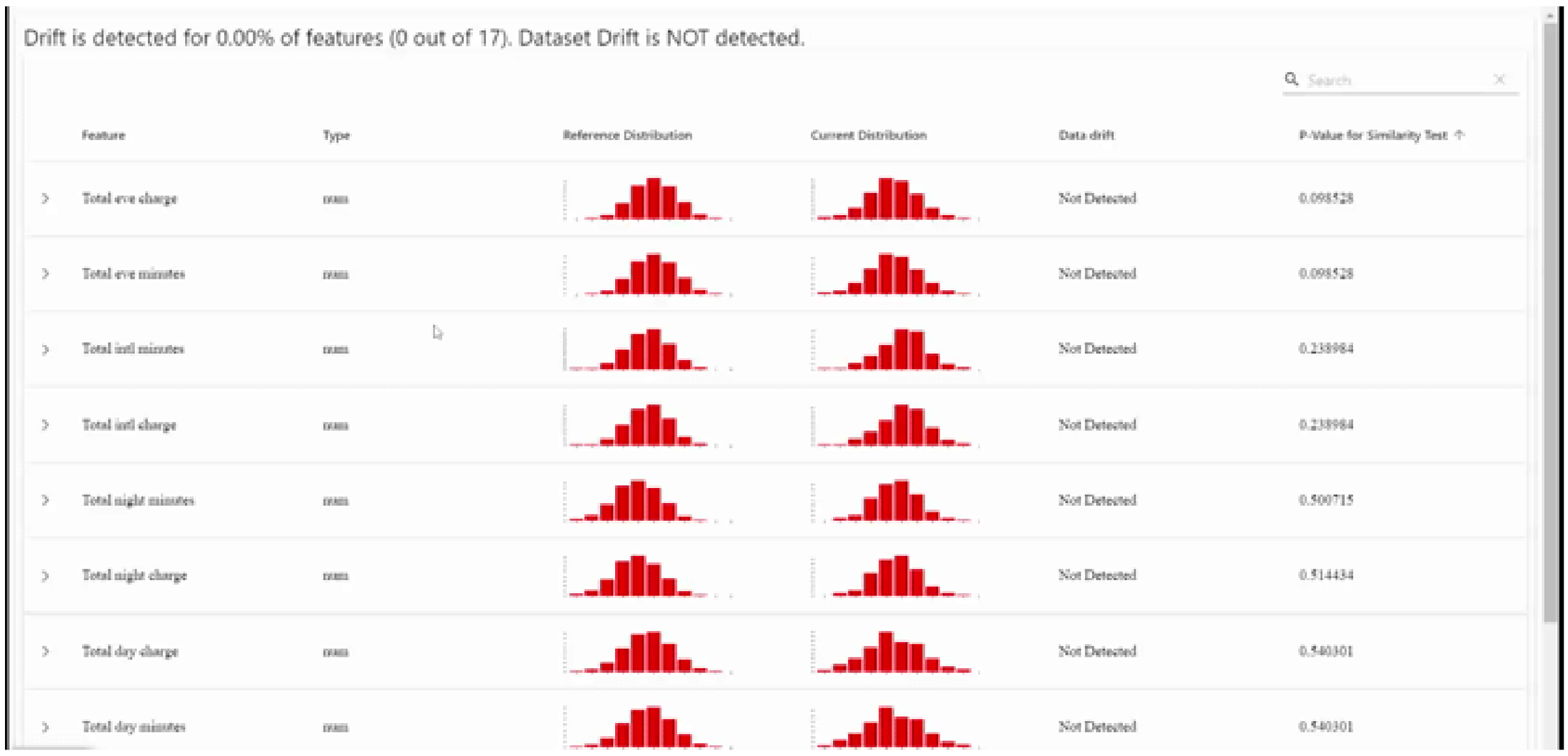
13

13

Evidently

Evidently *is an open-source python package to analyze and monitor machine learning models. The package is explicitly developed to establish an easy-to-monitor machine learning dashboard and detect drift in the data. It's specifically designed with production in mind, so it's better used when a data pipeline is there. However, you could still use it even in the development phase.*

We could monitor our machine learning model metrics as a whole and per feature prediction. The detail is good enough to know if there is a difference when incoming new data.

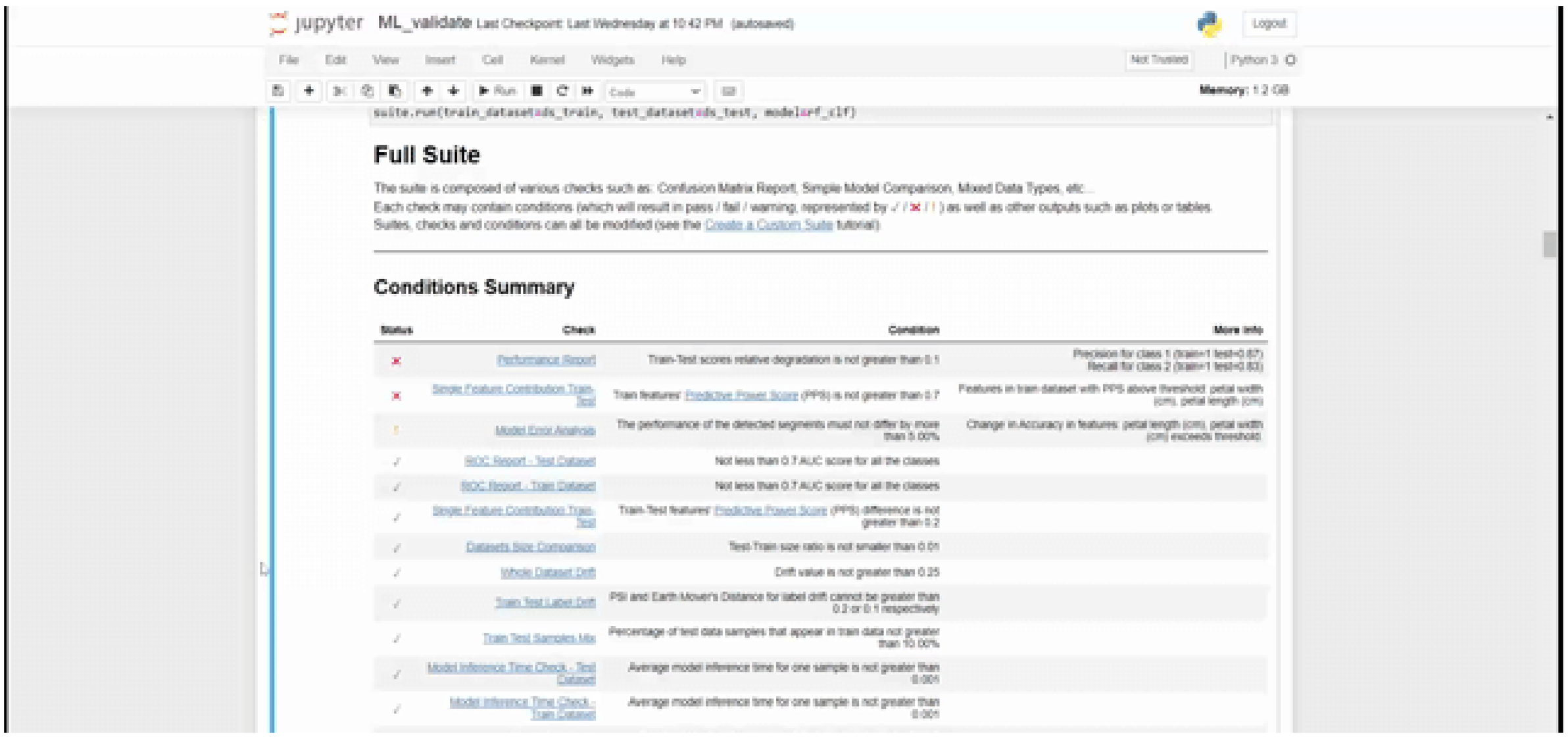



```
pip install evidently
```

Deepchecks

Deepchecks *is a python package to validate our machine learning model with a few lines. Many APIs are available for detecting data drift, label drift, train-test comparison, evaluating models, and many more. Deepchecks are perfect to use in the research phase and before your model goes into production.*

Deepchecks produce full suites reports that would contain much information such as Confusion Matrix Report, Simple Model Comparison, Mixed Data Types, Data Drift, etc. All the information you need to check the machine learning model is available in a single code run.

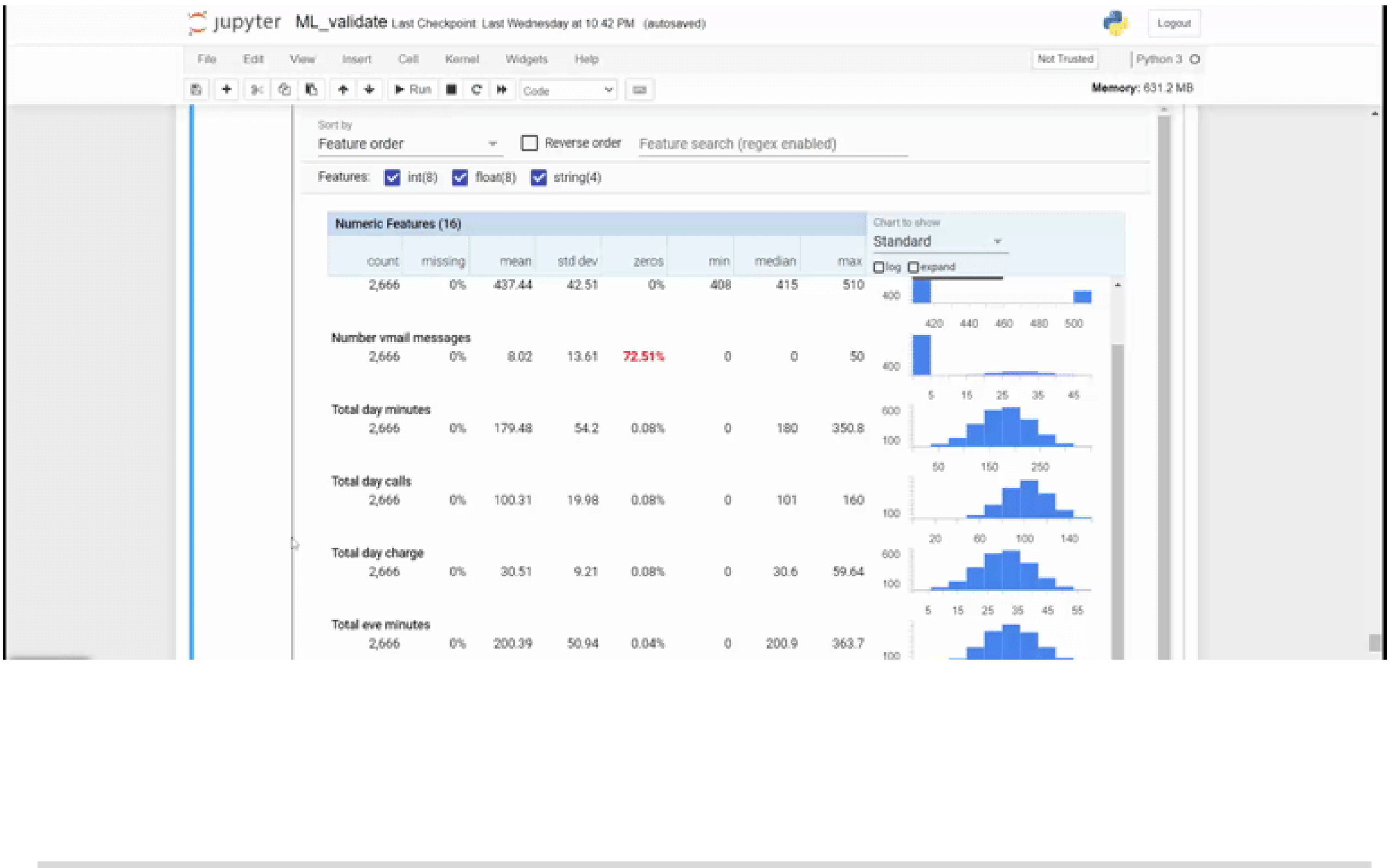



```
pip install deepchecks
```

TensorFlow-Data-Validation

TensorFlow Data Validation *or TFDV is a python package developed by TensorFlow developers to manage data quality issues. It is used to automatically describe the data statistic, infer the data schema, and detect any anomalies in the incoming data.*

The TFDV package is not limited only to generating statistical visualization but is also helpful in detecting any change in the incoming data. We need to infer the original or reference data schema to do this.




```
pip install tensorflow-data-validation
```

14

SYNTHETIC DATA

14

Faker

Faker *is a Python package developed to simplify generating synthetic data. Many subsequent data synthetic generator python packages are based on the Faker package. People love how simple and intuitive this package was.*

With Faker we could generate various synthetic data. For example, we would create a synthetic data name. The result each time we ran Faker is different data than our previous iteration. The randomization process is important in generating synthetic data because we want a variation in our dataset.

```
pip install faker
```

SDV

SDV *or Synthetic Data Vault is a Python package to generate synthetic data based on the dataset provided. The generated data could be single-table, multi-table, or time-series, depending on the scheme you provided in the environment. Also, the generated would have the same format properties and statistics as the provided dataset.*

SDV generates synthetic data by applying mathematical techniques and machine learning models such as the deep learning model. Even if the data contain multiple data types and missing data, SDV will handle it, so we only need to provide the data (and the metadata when required).

metric	name	raw_score	normalized_score	min_value	max_value	goal	error
CSTest	Chi-Squared	0.885318	0.885318	0.0	1.0	MAXIMIZE	None
KSTest	Inverted Kolmogorov-Smirnov D statistic	0.687920	0.687920	0.0	1.0	MAXIMIZE	None

```
pip install sdv
```

Gretel

Gretel *or Gretel Synthetics is an open-source Python package based on Recurrent Neural Network (RNN) to generate structured and unstructured data. The python package approach treats the dataset as text data and trains the model based on this text data. The model would then produce synthetic data with text data (we need to transform the data to our intended result).*

Gretel required a little bit of heavy computational power because it is based on the RNN, so I recommend using free google colab notebook or Kaggle notebook if your computer is not powerful enough.

```
pip install gretel
```

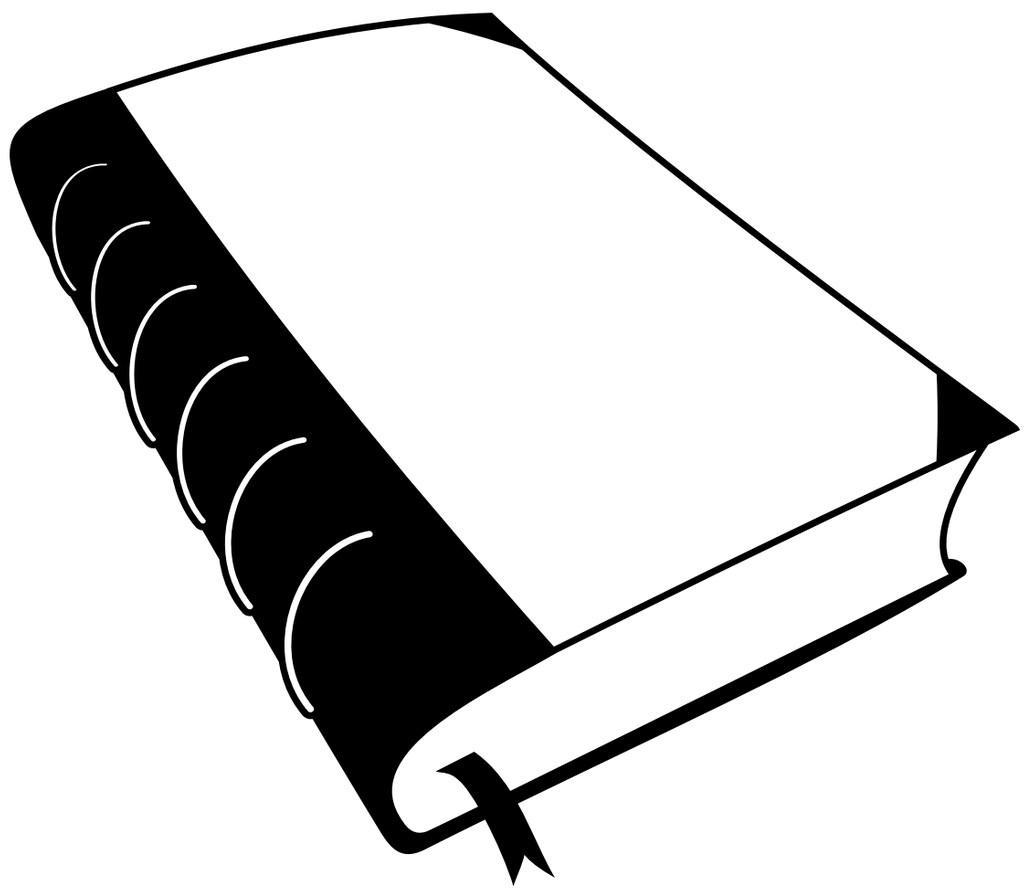
Mimesis

Mimesis is a robust data generator for Python that can produce a wide range of fake data in various languages. This tool is useful for populating testing databases, creating fake API endpoints, filling pandas DataFrames, generating JSON and XML files with custom structures, and anonymizing production data, among other purposes.

The features include:

- Multilingual: Supports multiple languages.
- Extensibility: Supports custom data providers.
- Easy: Offers a simple design and clear documentation for easy data generation.
- Performance: Widely recognized as the fastest data generator among Python solutions.
- Data variety: Includes a variety of data providers designed for different use cases.
- Schema-based generators: Offers schema-based data generators to produce data of any complexity effortlessly.

```
pip install mimesis
```



XV. Closing Remarks

All the packages written in this e-book are purely my opinion and tested personally.

The Python packages listed here might undergo name changes or be completely disabled by the respective developer after I write this e-book, so be cautious with that.

Overall, credits are given to the developer for all these amazing Python packages.

I hope this e-book can help you in the data science learning journey.