Chapter 16
Machine Learning-Based Mobile Applications Using Python and Scikit-Learn

Santhosh Kumar Rajamani
https://orcid.org/0000-0001-6552-5578
Maharashtra Institute of Medical Education and Research, India & Dr. Bhausaheb Sardesai Talegoan Rural Hospital, India

Radha Srinivasan Iyer
https://orcid.org/0000-0001-7387-4401
SEC Centre for Independent Living, India

ABSTRACT

This chapter gives a broad outline of machine learning on Android mobile phones using the Scikit-learn module. The first section introduces the reader to Python language; next, Python on Android is introduced with a brief historical note on implementations of Python on Android mobile phones. Pydroid3 is introduced in the subsequent section. This is followed by instructions on setting up an Android phone for machine learning. This is followed by a description of supportive modules for machine learning that are available for Pydroid3, and some example codes, namely: os, pathlib, Pandas, NumPy, SciPy, Matplotlib, Seaborn, PySimpleGUI, NetworkX, Biopython, WordCloud, Kivy, and Jupyter Notebook. The last section of this compilation describes the Scikit-learn library, basic concepts of the Scikit-learn module, and algorithms available with this module, namely: Linear Regression, Logistic Regression, Principal Component Analysis (PCA), XGBoost, K-nearest neighbors, and support vector machine.

DOI: 10.4018/978-1-6684-8582-8.ch016

INTRODUCTION

A robust and user-friendly toolkit for the Python programming language, *Scikit-learn* offers several machine learning methods. Because of its extensive toolkit, which includes modules for data pre-processing, model creation, model evaluation, and model selection, it is one of the most popular libraries among developers. Along with factor analysis, ability to extricate features, and scale dimensions on complex datasets. *Scikit-learn* also supports many supervised and unsupervised learning techniques like random forests, k-nearest neighbors, and support vector machine SVM. It is based on the well-known Python libraries *NumPy* and *SciPy*. Additionally, its algorithms are simple to use because of the detailed documentation and tutorials that provide step-by-step explanations, making it the perfect tool for anyone perusing machine learning. Finally, *Scikit-learn* module is the best choice for many machine learning applications thanks to its user-friendly API and strong emphasis on automation (Lutz, 2013).

PYTHON

Python is a popular, easy to learn, simple, object-oriented, interpreted, high-level programming language, based on more abstract, difficult, and symbolic C language. Python was created by Guido van Rossum in 1990s. It has become known for its ease of use, readability, low-level syntax, and ability to rapid develop ideas. Python also has a wealth of robust machine learning libraries that make it easy to implement complex algorithms and platforms in Python. This makes Python an attractive choice when working with machine learning problems. Python and its source code are released under liberal GNU General Public License (GPL) which makes it attractive option even for commercial projects (Lutz, 2006).

Currently, Python is the preferred programming language of more than 80% of the developers. One key advantage of using Python is that it is generally easier to get started with than many other languages promoted as suitable for machine learning. This is partly due to Python's readability and partly due to the abundance of free resources available online. Additionally, Python has been widely adopted by data scientists, making it a familiar toolkit that can be easily integrated into existing neural networks and business applications. For example, Flutter (Dart)is currently the de facto language of Android development is a bracket hell, brackets to enclose chunks of code, brackets to enclose those brackets, *ad infinitum*., making code confusing and incomprehensible to the reader (Lutz, 2013 and https://neurodeb.pirsquared.org)

THE POWER OF PYTHON FROM THE SMALLEST TO THE LARGEST

In contrast despite its obvious simplicity Python is a powerful language capable of implementing many complex ML algorithms using vast collection of additional libraries. On the other hand Python can also be used to automate common menial tasks like for instance, document conversions (*docx2pdf, pdfplumber,PyPDF2*), creating PowerPoints (*pptx*), taking screenshot, copy and paste text from the system clipboard (*pyperclip*), merging word documents (*docx*), manipulate Excel sheets (*openpyxl*), image background removal (*rembg*)or lookup a stock price(*yfinance)* or lookup currency exchange rate (*forex_python*) or changing colour images to greyscale (*pillow*) or spell-check a text (*textblob*) or download a file (*urlib*). Office automation task to complex Machine learning algorithms like for instance, Distributed Evolutionary Algorithms in Python (DEAP) (*deap* module) and Symbolic mathematics (*sympy*). DEAP gives the power of mathematical modelling, optimization, genetic programing, particle swarm optimization, and evolution strategies. Many of these modules are also available via pip on Pydroid3 and code conveniently tested on Android itself. Even the complex module *deap* is obtainable on Android. Python is indeed the Swiss army knife of programming language.

LIMITATION OF PYTHON

Python interpreter itself is written in C and converted into computer-executable machine code. When Python code is written and executed, it is first parsed and converted by the interpreter into bytecode. The interpreter then runs this bytecode, carrying out the operations dictated by the code. Thus, every instruction in Python must be interpreted in symbolic byte language, this makes Python much slower than other languages (Kane, 2017).

MACHINE LEARNING PARADIGMS IN PYTHON

Machine learning implementation in Python is the most popular tool used by data scientists to solve complex problems and improve existing models (http://github.com). By leveraging powerful libraries such as *Scikit-learn*, *NumPy*, and *Tensor Flow*, users can build sophisticated models that can make decisions based on large amounts of data and can even improve the accuracy of predictions over time. With python's ease of use and versatile syntax, developers of all levels can quickly access powerful machine learning algorithms when developing new applications or

looking to apply existing ML models. Python also offers a robust suite of graphing libraries that allow users to visualize their data to better inform model building or interpret results from existing ML models. The first part of this section focuses on adjunct software on available Android for Machine learning along with some usage examples (Apeltsin, 2021).

PYTHON ON ANDROID

Python on Android is an increasingly popular platform for developers who want to create applications on mobile devices. It works by providing a virtual machine (VM) layer which sits between the operating system and the application, allowing users to develop apps with Python code. The VM optimizes the performance of an app, as well as simplifying development tasks such as porting existing apps from other platforms. Additionally, it also allows developers to use common libraries like TensorFlow which makes it easier to create machine learning models. With the rise of 5G networks making streaming video more feasible and enabling faster file transfers, Python on Android could become even more popular with developers looking for a powerful platform for their projects.

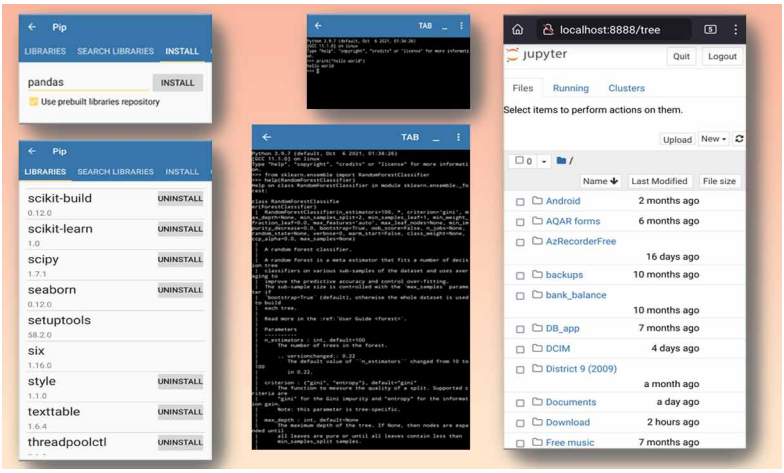
VARIOUS PYTHON DISTRIBUTIONS PORTED TO ANDROID OS

Although Python has not always been supported on the Android operating system, there are currently several choices available. The QPython app, which was published in 2012, was one of the first ways to execute Python on Android. This app included a Python interpreter built-in, as well as several libraries and tools for working with Python on Android, allowing users to run Python scripts and projects on their Android devices. Other Python on Android possibilities, such the *Kivy* framework, which makes it simple to develop cross-platform mobile apps using Python, have developed in recent years. Additionally, Python code may now be run on Android using a web browser thanks to the popularity of Jupyter notebooks. Additionally, due to the popularity of Jupyter notebooks, applications like JupyterLab and Colab may now be used to run Python code on Android online using a web browser (Mirjalili, 2019).

PYDROID V. 3.9.7

Pydroid3 is an Android app that allows users to write and run Python scripts on their Android devices. It provides a Python development environment for Android, allowing users to create and run Python scripts and programs directly on their phones or tablets. Pydroid3 includes a built-in text editor, interpreter, and debugger, making it a powerful and convenient tool for learning and experimenting with Python on Android devices. The current stable version of Pydroid3 available on Google Play store is 3.9.7, that was released on Oct 6, 2021 (running GCC on Linux 11.1.0). GNU Compiler Collection is known as GCC, which is the default compiler on many Linux distributions including Android GCC offers high-quality, effective, and portable code creation for a variety of applications.

Figure 1. Features of Pydroid3 that can be utilized for Machine Learning



Pip is available as a convenient GUI program, B. Installed packages can be seen in a single scroll-view C. Python interpreter mode is also available D. Scikit-Learn installed on phone showing implementation of algorithm Random Forrest Classifier (via. *RandomForrestClassifier*) E. Jupyter notebooks can be installed locally on Android phone (http://localhost:8888/tree) via Pip, does not need an internet connection and data stored on SD card. Users can conveniently execute Python scripts on their Android devices with the Pydroid3 software. It is made for people who wish to learn or use Python on the move. Overall, Python support for Android has developed significantly in recent years, and it is now a viable option for many types of platform development. Users of Pydroid3 may create and execute

Python scripts directly on their Android smartphones, thanks to the inclusion of a built-in Python interpreter, a code editor, and several libraries and modules which are discussed individually in the subsequent sections.

A NOTE ON PACKAGE VERSION SYSTEM MENTIONED IN THIS PAPER

In this compilation, package names are mentioned in Italics to help user recognize these as Python addon packages or modules that can be install via Pip. Major version number of package, along with minor version, is mentioned in this work as these remain stable and unchanged for many years, even decades. This follows Python version identification and dependency specification scheme (PEP 440) for modules, which follows Semantic versioning scheme which is depicted as follows: "Major.Minor.Patch", for example, latest release of *SciPy* package on Pydroid3 is "1.7.1". The patch versions are changed frequently as software is updated and minor version numbers gradually change with timeframe of months and years, for instance Python 3.10 has been released as this work is being compiled. On the other hand, many packages do not follow such clear-cut scheme of release and versioning (7. Releasing and Versioning — Python Packages, n.d.).

SETTING UP PYDROID3 (PYTHON V. 3.9) ON ANDROID MOBILE PHONE FOR MACHINE LEARNING

Setup the Android Permissions for Python Development

Android permissions provide access control for applications to access system services and parts of the operating system. Permissions are broken into two major categories: Normal and Dangerous. Dangerous permissions require user approval, as they could potentially cause malicious or privacy-violating effects on the device if granted. Applications must request permission when they communicate with other data sources; however, users can also modify application permissions in their settings menu to approve or deny requests made by apps or carriers to access resources on their device. Android's permission system provides an additional layer of security that helps protecting data and sensitive information stored on mobile devices from unauthorized access.

Android permissions are important security measures that allow users to control which functions, data and resources third-party apps can access on their devices. When the user downloads a new app, they have the option of granting or denying

individual permissions for specific features or data. For example, a user may be asked to grant access to their camera and contacts list for a photo-sharing app. These settings help protect personal information from being shared or transferred without explicit permission from the user, as well as preventing malicious apps from accessing private data and system resources. The Android permissions system is an important tool for protecting user privacy and security on mobile devices (Cohen et al., 2014).

SETUP ANDROID SD CARD PERMISSIONS FOR DATA ACCESS

Android devices use the SD Card permissions system to control access to certain areas of the internal storage. This means that applications must be granted specific permission from the user before they can read, edit, or delete files stored on external SD cards. The Android manifest must also provide such permissions within its code for an application to fully leverage SD card functionality. Furthermore, this permission system helps ensure data security by preventing apps from accessing sensitive information that is stored on external media and granting users increased control over which app can write data to their SD cards. All in all, SD Card permissions are a simple but effective way of controlling file access and protecting users' data (Cohen et al., 2014).

As a part of setting up ML on Android phone, user must grant SD card read-write permission for Pydroid3 to be able to access the SD card and load the datasets and assets. This is accomplished in ***GUI by Settings ® Apps®Your apps® Pydroid3 ®Permissions® Files and media access***. This only solves half the problem of access as described in the following section.

Due to inherent restriction on the system, user cannot directly access the vital operating system files which reside in the system folder like: ***"/data/system",*** unless the phone is rooted. Download folders at ***"/storage/emulated/0/Download"*** is a fixed path folder which user can both read and write.***"/data/user/0/ru.iiec.pydroid3/app_HOME***" (https://linuxhint.com). is the default folder of Pydroid3 app, is inaccessible by default except when read or written by host Python program. For example, if we save an output plot *matplotlib* via the code:

```
# Save the rendered figure to disk
plot.savefig('trends_in_hospital_admission.png', dpi=300)
plot.show()
# (OR) saving the data frame
dataframe.to_csv(r'my_data.csv', index=False)
```

Please note, that plt.show() which displays the graph is placed after save, logic dictates that plot be shown then saved. This 'trends_in_hospital_admission.png', PNG image file or data_file.csv, CSV data file will be saved in ***"app_HOME"*** folder and no file manager can directly access this folder for editing or visualization. This creates a unique, Android-specific access problem for loading datasets and other assets for Machine learning.

MY WORKAROUND FOR THIS PROBLEM

For instance, the code *df = pd.readcsv("data.csv")* will load the dataset data.csv only from app_HOME folder. User themselves have little access to this directory which is hidden deep in *Android* system directory. The Python program host have access to the folder, so the workaround would be to copy this output file from the '***app_HOME'*** folder to any other folder which we have full access, like the '*Downloads'* folder.

Another simpler solution is to use OS module to change current working directory to one of those folders that the user has full access. This is accomplished by code on my phone (this maybe different on other phones, but will be something in similar lines, use ES-Explorer/MiX file manager from Play store to find out the correct system specific path).

```
# Remove this line in Windows and Linux, causes program to crash by null error
    as no such directory exists on Windows and Linux, use some file manager to
    find the exact path to our media storage folder in Android
import os
os.chdir('/storage/emulated/0')
```

Pathlib purists immediately would frown upon me for my approach of hard coding the path. But this approach gets the work done, keeps the code clean, makes loading, and saving data, much easier and faster. Just load and save directly, no need to bother with the path or access.

The code *pd.read_csv("/storage/emulated/0/data.csv")* assumes that the data.csv file is in path /storage/emulated/0 is fixed, full path location needs to be specified every time one reads or writes the file.

Please note for both the above codes to work without a null error, we must enable Files and media access permission (SD card access) via the Android System GUI.

PIP Python Package v. 21.2 Management Systems on Pydroid3

Pip is a powerful and versatile package management system for easily installation, upgrading, dependency management of additional Python modules. Pip is preinstalled onsite with App installation of Pydrioid3 (Python 3.9) from Google Play Store. In addition to the ability to manage packages, it also provides support for multiple version control systems, can handle complex dependencies, provides a secure authentication system, and more. Furthermore, Pip supports several platforms, such as Linux distributions and Windows PCs as well as Mac OS X devices. Pip simplifies the workflow of software engineers by providing an automated way of keeping track of dependencies, enabling easier development and testing of applications. This makes it one of the most important tools for modern software development teams looking to streamline their process in creating high quality products (Vanderplas, 2017).

PIP is the default package management system which is supplied with every installation of Python. This package management system is used to install additional packages and add-on functionality to the existing Python installation. For example, *Pyinstaller* compiler add-on which can compile Python code to independent single EXE files for porting to other Windows computers. *Kiwi* is a similar compiler add-on available on Linux system, which can be used to create standalone APK files to port Python programs to other Android mobile phones (Vanderplas, 2017).

OS vs. on Pydroid3 for Setting Up Machine Learning on Android Phone

The Python OS module is an incredibly powerful tool that provides numerous ways to interface with the underlying operating system. It offers basic functionality such as manipulating files and directories, managing environment variables, and executing external programs. More advanced features include working with process IDs, creating sockets for network communication, and even interacting with process signals.

The Python OS module provides access to the operating system's functionality such as creating directories, deleting files and directories, interacting with environment variables, and others. It is a built-in library of modules that contains functions related to the operating system's different resources such as hardware and memory. Additionally, it can be used to access command line arguments, set file permissions, stat info (file size, creation time etc.) As its operations may involve dealing with sensitive information or operations that could affect an entire system's stability, it is important for developers to become familiar with proper use of the OS module before performing any tasks related to it.

Table 1. Housekeeping commands of Python PIP package manager, which is the most popular package manager and comes bundled with all distributions of Python, including Pydroid 3 distribution which runs on Android mobile phone where it is a GUI application

Command (Windows, Linux, Mac)	Purpose	Equivalent command on Android Pydroid3 PIP
>pip install package_name Example: >pip install pandas	This command installs package_ name	Installation is via GUI under left drawer menu- PIP-Install- Enter library name- Click on install button
>pip uninstall pandas	This command uninstalls pandas package	List of installed packages can be accessed via GUI under left drawer menu PIP-Scroll header LIBRARIES
>pip install –upgrade pip	This command upgrades self, PIP	PIP is upgraded automatically with App upgrade from Google Play Store
> pip install – upgradepackage_name	This command upgrades the package_name	None, need to install and reinstall hoping that they upgraded for Android
>pip show package_name	This command displays information on package	Installed List can be accessed via GUI under left drawer menu- PIP-Scroll header LIBRARIES
>pip check	This command verifies that there are no broken packages and dependency issues.	None
>pip list	This command lists all the packages installed on the system	Installed List can be accessed via GUI under left drawer menu PIP-Scroll header LIBRARIES

Additionally, this module also provides a wide variety of convenience functions for accessing operating system specific data such as user information, current date & time information, and more. For these reasons, the OS module can prove invaluable for developers who frequently deal with system internals or need to perform more complex tasks than typical everyday operations. OS module is preinstalled onsite with every installation of Pydroid3 (Python 3.9) like Pip. This module is very important in setting up our phone for machine learning.

Pathlib v. 1.0 on Pydroid3

Python *pathlib* ver. 1.0 module is a new, succinct, and clearer implementation of OS module is also available for Pydroid3. Pathlib code is easier to understand and debug than OS module. The main concept behind pathlib package is to do away with hard-coded fixed path and keep a flexible path independent implementation of the program. Example flexible path code sample is *my_home=pathlib.Path.home*(). This *home()* method always points to home path irrespective of the system, in Windows

it is user folder and ~ in Linux/Posix systems. The equivalent in Android system is the default path of execution of Pydroid3, which is mostly *"/data/user/0/ru.iiec.pydroid3/app_HOME*"(https://linuxhint.com). Unfortunately, due to the restrictive director permissions and fixed path in Android, this flexibility is of little utility in Android OS.

Pandas v. 1.3 on Pydroid3

Pandas is an essential tool for manipulating, organizing, and analyzing data in the Python programming language. It provides powerful capabilities for working with tabular data, such as intuitive manipulation, sorting and filtering functions, grouping operations, and efficient hierarchical indexing. Additionally, *Pandas* makes robust use of the *NumPy* library for optimized numerical computing and statistical operations. This combination makes it a suitable choice for dealing with complex datasets requiring sophisticated operations at scale. With features that allow forward-filling missing values in temporal series, merging multiple data sources into one coherent dataset, automatic inference, and detection of data types across columns, as well as customized access commands allowing users to quickly pivot and reshape data to suit their needs. *Pandas* is the most popular choice amongst analysts and data scientists (Chen, 2018).

NumPy v. 1.2 on Pydroid3 (Python 3.9)

NumPy is a powerful open-source numerical computing library for Python. It is used by scientists, engineers, and data analysts alike to perform array operations such as calculations, manipulations, and transformations. *NumPy* features a powerful array object which makes it suitable for high-performance operations on multi-dimensional arrays and provides a versatile set of linear algebra functions. Furthermore, *NumPy* provides tools to read and write n-dimensional arrays from/to external files as well as integrate with existing data analysis tools like *SciPy*, Pandas, and TensorFlow. Through these capabilities, *NumPy* allows smooth interoperability between different libraries in Python's scientific computational ecosystem enabling users to quickly implement complex algorithms in their applications (Bressert, 2013).

NumPy has support for large, multi-dimensional arrays and matrices along with a collection of high-level mathematical functions to operate on these objects. *NumPy*'s fundamental data structure is a multidimensional array, which enables it to efficiently manipulate large datasets. In addition, it provides an interface to many other popular libraries such as *Pandas* and *Matplotlib*. With *NumPy*, linear algebra operations such as matrix inversion and singular value decomposition can be performed quickly and efficiently. Furthermore, its built-in random number

generator makes it the ideal choice for random simulations or stochastic optimization problems. For many machine learning projects involving artificial intelligence or data mining tasks, *NumPy* is key to achieving reliable results in terms of performance and accuracy (Idris, 2011).

SciPy v. 1.7 on Pydroid3 (Python 3.9)

SciPy is a powerful open-source scientific computing library for Python, built on the *NumPy* extension. *SciPy* version 1.71 is latest module that is available for Pydroid3.*SciPy* is designed to provide fast access to a collection of specialized mathematical routines such as integration, optimization, statistics, and linear algebra (Hans Petter Langtangen, 2016). It also provides an extensive set of capabilities such as visualization and plotting functions that allow developers to demonstrate their results in real time. *SciPy* simplifies the development process by allowing users to create complex models quickly and efficiently. Furthermore, it has strong external linking ability which allows researchers to easily link with other software packages for data analysis tasks or numerical simulation. This makes *SciPy* highly efficient for collaborative projects as it enables multiple scientists to work together without significant overlap or risk. All in all, *SciPy* is an ideal toolset for both newcomers and experts alike due to its broad feature support and wide range of usage options (Blanco-Silva, 2015).

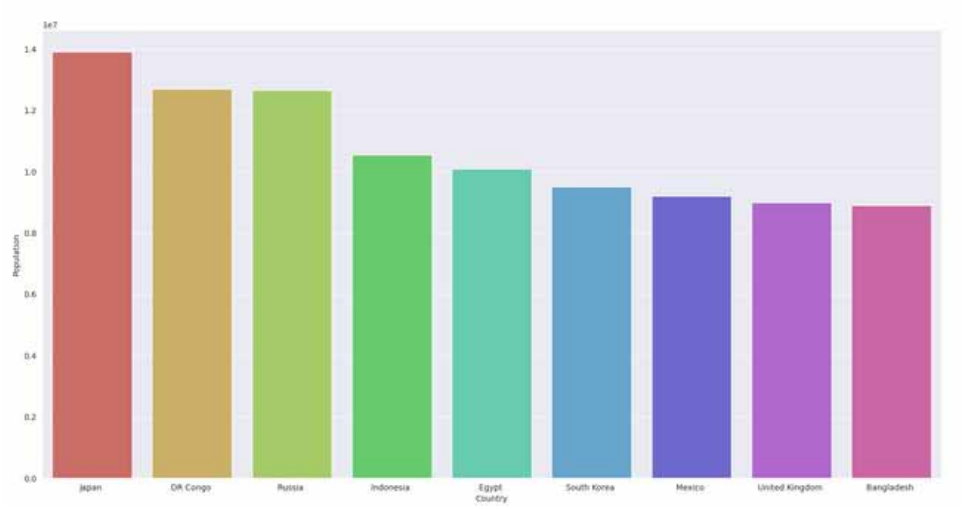
SciPy is a powerful open-source scientific computing library, based on *NumPy* and providing a wide variety of advanced mathematics solutions for industry standard applications. It features functions for data analysis, image processing, linear algebra and probability statistics offering comprehensive performance in these areas. *SciPy* has a high level of compatibility with other scientific computing libraries such as MATLAB or R, so users can easily leverage previously established code in the existing environment. The library also provides an easy-to-use interface for developing self-contained proprietary functionality (Bressert, 2013).

Additionally, it is designed to efficiently combine multiple existing scientific libraries in complex setups across multiple hardware platforms. By creating both simple and complex software solutions together with free access to updates and community support, *SciPy* makes scientific development more accessible than ever before. *SciPy* is a collection of algorithms that can be utilized in various machine learning tasks. It provides easy-to-use and efficient implementations of standard routines in supervised learning, unsupervised learning, and semi-supervised learning. It provides tools for data mining and data analysis. It also has various command line tools to perform specific tasks such as preprocessing data, transforming it into a different representation, performing model selection, or making predictions (Bruce & Bruce, 2017).

Matplotlib v. 3.4 and Seaborn v. 3.4.3 on Pydroid3 (Python 3.9)

Matplotlib provides comprehensive 2D plotting and high-quality publication-ready graphics. It supports both interactive graphical options as well as scripting with the powerful *PyLab* interface. Specifically, *Matplotlib* includes several unique features to aid in creating graphs through a user-friendly interface, including a variety of different color palettes that can be used to customize plots; customization of legend labels, tick marks, line styles and other details; built-in support for plotting arbitrary datasets alongside one another; integration with *NumPy* and *SciPy* packages for statistical datasets; resolution aware display formats plus panning and zooming tools. With its ability to generate high quality vector graphics pertaining to publications or web service applications, it also serves as an excellent choice to represent data-intensive scientific problems (Chi et al., 2017).

Figure 2. Output plot of matplotlib and seaborn 10 countries with their population, generated on Pydroid3. Such good quality data visualization is possible on mobile devices.



Seaborn is a graphing module used to visualize data, Seaborn a wrapper module of higher API function calls to *Matplotlib* which allow users to quickly generate attractive, articulated plots. By incorporating features like color-coded matrices and hierarchical clustering, Seaborn takes advantage of intuitive and visually appealing concepts to improve the understanding of complex datasets. These capabilities make

Seaborn ideal for exploratory data analysis, offering visualizations such as line graphs, bar plots, heat maps, violin plots and joint plots that all aim to improve the workflow in a quantitative analysis environment (Chi et al., 2017).

PySimpleGUI v. 4.6 on Pydroid3 (Python 3.9)

The *PySimpleGUI* module for Python is an easy-to-use, robust graphical user interface (GUI) library optimized around the *Tkinter* framework. It provides a platform-independent wrapper that makes creating custom GUI programs in Python both simple and easy. This module makes developing cross-platform applications with a wide range of functionality a breeze, it works on all common computer systems including Mac OS X, as well as popular Android OS. Additionally, its drag-and-drop feature allows one to quickly create widgets without having to deal with any complicated coding. The module also features built-in themes for further customization and offers several tools such as row/column creation wizards, text boxes and labels that are sure to provide users with the perfect GUI choice for their application.

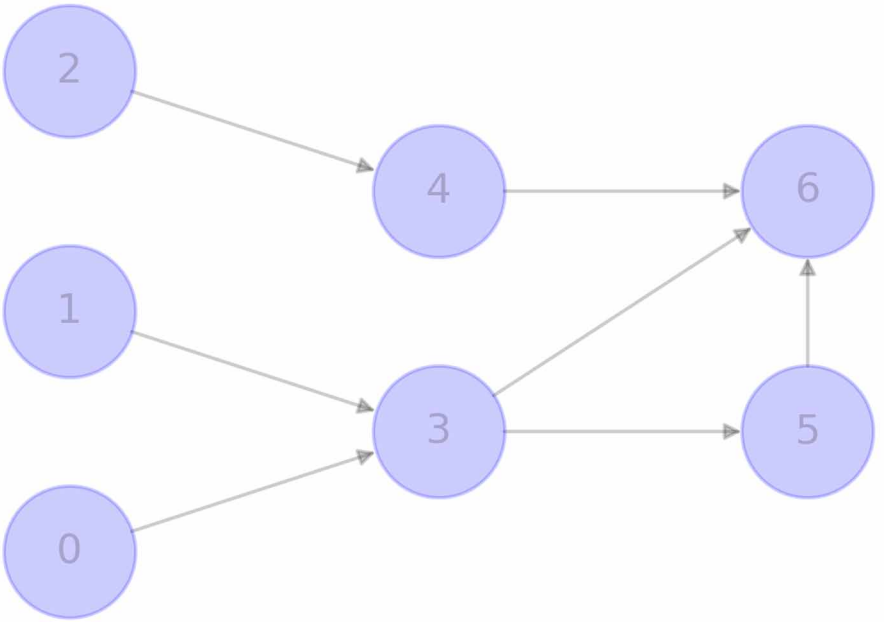
NetworkX v. 3.0 (Released on Jan 08 2023) on Pydroid3 (Python 3.9)

Networkx allows users to deploy traditional network analytical tools as well as generate graphical visualizations of networks. Networkx offers several tools that make it an excellent choice for network applications including structured graph types such as directed graphs, convenience methods for node and edge manipulations.

There is support for network algorithms for structural characteristics like clustering coefficient, betweenness centrality, closeness centrality etc., writers that support exporting to standard file formats such as GML, GraphML and Pajek (Platt, 2019).

The package is easily extendable with user-defined plugins which makes it highly versatile and adaptable to various domain tasks. Additionally, its features strong interoperability with other Python packages such as *NumPy/NumPy* or *Matplotlib* which makes it ideal for performing sophisticated analysis tasks within data science pipelines (Goerzen, 2004). NetworkX library has been ported for Machine learning and Network analysis on Android mobile phones. The graph above shows output of NetworkX generated using sample code on Pydroid3.

Figure 3. Output graph produced by NetworkX on Pydroid3 (Py3.9) using following options: "node_size": 1500, "alpha": 0.2 and "node_color": "blue" in NetworkX to render the graph on Mobile device



Biopython v. on Pydroid3 (Python 3.9) for Bioinformatics Research

Biopython (*biopython*) is an open source, collaborative Python project that provides a comprehensive set of tools for biological computation. It offers modules for a variety of bioinformatics tasks and supports Python 2, 3 as well as Java and C code. *Biopython* provides interfaces to useful bioinformatics programs and libraries, allowing users to leverage the benefits of both third-party code and existing user expertise without having to learn all the details. Furthermore, the project includes a large collection of modules for dealing with the different types of data generated by high-throughput biology experiments. This makes it easy for bioinformaticians to quickly develop powerful analysis pipelines in their chosen programming environment. *Biopython* is also available on Pydroid3, bioinformatics research can be carried out on Android Mobile phone with ease and portability (Zhang & Rajapakse, 2009).

Word Cloud in Pydroid Using "wordcloud" v. 1.8 Module

Word clouds are a key tool used in visualization techniques, allowing data and text to be shown graphically. Word clouds provide an excellent way to quickly identify words that are most relevant to a set of data. This type of analysis is helpful for visually summarizing topics from surveys, interviews, or documents, as the size, color, orientation, and font can all be adjusted to emphasize the keywords or phrases within the collection of text. Additionally, word clouds give users an opportunity to explore sentiment analyses, trends and correlations that can be discovered by analyzing different variations within word clouds. Thus, word clouds have become commonplace in market research as well as various other forms of visualization.

The *wordcloud* module for Python is a versatile and powerful library for creating word clouds. It enables users to produce visually attractive diagrams, comprised of words from text-based data sources. The frequency of each word used in the diagram is represented by its size in the drawing. This module facilitates a range of options for representing different fonts and color palettes, as well as adjusting image sizes and orientations. In addition, it supports rendering of complex shapes in the final drawing through support for custom masks (David Paper, 2019).ig4. Demonstration of "*wordcloud*" module of Python for generating Word clouds which show words that are most relevant to a set of data. This Word Cloud is generated for main keywords for this article pertaining to Machine Learning.

Figure 4. Showing the sample output of wordcloud module, this example uses few common Machine learning terms, more complex examples can be implemented by this module

KIVY FOR COMPILING PYTHON TO ANDROID APPS APKS

A free Python framework library for creating cross-platform graphical user interfaces is called Kivy (GUIs). It can make use of multitouch events to make apps that are more interactive and user-friendly, and it is designed for use across all platforms, including mobile ones. Applications for Linux, Windows, MacOS, Android, and iOS can be made using the Python and Cython programming language known as Kivy. Kivy apps maybe deployed on common platforms, like Mac, Android OS. It's made to make it simple for users to develop cross-platform applications that work on a range of gadgets, including smartphones, tablets, PCs, and Raspberry Pi. Python programs can be converted into APK files that can be installed and used on Android devices thanks to Kivy's provision of the required tools and features for creating and packaging Android applications. Kivy's packaging tools maybe utilized to turn the program into an Android device installable APK file. Kivy documentation can provide further details on building Android applications (Cohen et al., 2014).

Jupyter Notebook v. 1.0.0 on Pydroid3

Users can integrate text, visuals, and programming code in a single document using Jupyter notebooks, which are web-based interactive computing platforms. Jupyter notebook ver. 1.0 is available for Pydroid3 which users can utilize to harness the full power of Jupyter notebooks. They are a common tool among data scientists and academics and are frequently used for data analysis and scientific computing (Gupta, 2021).

For instance, Jupyter notebooks offers a more user-friendly and intuitive design, complete with tab completion and syntax highlighting. Additionally, it has sophisticated capabilities like magic commands that make it simple to complete complicated tasks without having to write a lot of code. An improved interactive Python interpreter is called IPython. Jupyter notebooks, allows one to produce, share and collaborate on notebooks containing python programs, latex code, pictures, graphs, interactive maps, and formatted textual matter (like headings, justification), are just one of the tools that IPython offers for collaborating with different programming languages and technologies. Overall, IPython is a useful tool for anyone who works with Python and can significantly increase your productivity and efficiency (Gupta, 2021).

SCIKIT-LEARN LIBRARY ONPYDROID 3

There are several popular Python libraries for machine learning including RapidMiner, MXNet, TensorFlow and *Scikit-Learn.* Each library has its own set of features and

advantages, but they all share a common goal: to enable us to develop efficient M.L models.

Scikit-learn machine learning module implemented for Python and distributed under the 3-clause BSD license. The module is open-source, and anyone can suggest improvements, corrections, and even new algorithms. The liberal BSD license makes it free to use for both commercial and non-commercial purposes. It is a powerful tool that facilitates the development of classification, regression, and clustering algorithms using pre-made decision tree implementations. With *Scikit-learn*, it is possible to build complex models with minimal effort and fine tune them as needed. The library can generate more advanced prediction results by incorporating additional features into existing decision trees to make more accurate predictions than individual decision trees alone (Mirjalili, 2019).

In addition, *Scikit-learn* provides tools for feature selection, feature extraction, cross validation, model evaluation metrics (e.g., accuracy score), and visualization capabilities. By leveraging these capabilities, data scientists can customize their analyses and target specific goals like prediction accuracy or computational efficiency, opening new possibilities for sophisticated data analysis techniques with decision tree methods (Mirjalili, 2019).

Scikit-learn has several utilities for preprocessing data, model selection, and validation. *Scikit-learn* makes it easy to quickly prototype and develop an initial model with minimal code. Its built-in cross-validation tools simplify assessing the accuracy of various models while its efficient components can be combined into complex pipelines that leverage the power of parallel computing architectures. In addition, detailed online documentation combined with an active community of users make *Scikit-learn* an ideal platform for professional machine learning tasks (Collins, 2018).

Scikit-learn is a powerful and versatile machine learning library which allows users to create, test and apply regression models. Specifically, *Scikit-learn* includes various linear and non-linear regression algorithms such as Linear Regression, Ridge Regression, Lasso Regularization and Elastic Net Regularization. These methods allow us to fit a predictive model according to given parameters thereby obtaining an optimal set of coefficients that best explain the relationship between the target variable and its predictors. Moreover, *Scikit-learn* also offers various sorting methods like random forests that are best suited for datasets with multiple correlated parameters. It also provides cross validation methods which enable us to assess the quality of our model's predictions by keeping a separate test set aside from training data. Therefore, beginners and experts alike highly appreciate this Python library for its clear interface and ease of use in doing simple or advanced regression tasks. *Scikit-learn* is widely used in academic and commercial research, as well as in data science competitions and hackathons (Collins, 2018).

BASIC SCIKIT-LEARN CONCEPTS OF ESTIMATORS, PREDICTORS, AND TRANSFORMERS

An estimator is an object that is used to fit a model to data in the context of machine learning. Any object that implements the fit () function, which trains the model on a training data, and the predict outcomes using the predict () method, which uses the trained model to make predictions on new data, is stated to be an *estimator*. Linear Regression and Logistic Regression classes for modelling continuous and binary data, respectively, as well as the Random Forest Classifier for training decision tree ensembles, are a few examples of estimators in *Scikit-learn*. *Scikit-learn*'s estimators are implemented in Python and made to be quick and effective while also offering a robust and adaptable user interface for working with data (Kane, 2017).

A *predictor* in machine learning is a model that has been trained to forecast upcoming events or results based on data in the setting of *Scikit-learn*. For creating and training predictors, the *Scikit-learn* library has several methods, including Linear regression modelling for the forecasting of continuous values, Logistic regression modelling that forecast binary outcomes as zero and one, models for making predictions via a succession of decisions based on decision trees, and Support vector machines for applications involving regression and classification (Kane, 2017).

A transformer is a sort of estimator that is employed in the setting preprocess the data by changing its representation. Any object in the *Scikit-learn* package that employed the fit and transform methods, is referred to as a transformer. Examples of transformers in *Scikit-learn* include the StandardScaler, which normalizes data by scaling to unit variance and subtracting the mean, and the PCA (principal component analysis) transformer, which maps data onto a lower-dimensional space by exploiting directions of maximum variance. In a machine learning pipeline, transformers are frequently employed to prepare the data for modelling (Kane, 2017).

LINEAR REGRESSION ALGORITHM ON SCIKIT-LEARN LIBRARY OF PYDROID3

A statistical technique for simulating the relationship between a dependent variable and one or more independent variables is linear regression. This means that the change in the dependent variable is inversely proportionate to the changes in the independent variables because it is assumed that the dependent variable and the independent variables have a linear connection. In other words, linear regression seeks to identify the line of greatest fit between the independent and dependent variables. Based on the known values of the independent variables, this line can then be used to forecast the value of the dependent variable(Weisberg, 2013). The coefficient

of determination (R-squared) and other metrics are used to assess the accuracy of the predictions(Bruce & Bruce, 2017). Logistic regression is employed to predict a variable's outcome using single or more independent parameter that can only have one of two possible values, such as "success" or "failure" (Hackeling, 2017).

LOGISTIC REGRESSION ALGORITHM ON SCIKIT-LEARN LIBRARY OF PYDROID3

Logistic regression uses a non-linear function to represent the non-linear association between the dependent parameter and an autonomous variable as opposed to linear regression, which assumes a linear relationship between the two parameters.

Performance of a classification model is pictorially assessed using AUC-ROC curve. The true positive rate are plotted on ordinant against the false positives are enlisted on the abscissae, resulting in score which can range from 0 to 1. A score nearer to 1 indicates better predictive accuracy of the model. Additionally, it is useful for establishing thresholds at which models adjust their decision boundaries between classes; these points may be determined by comparing different ROC curves. As with all statistical tests, however, results should be interpreted with high caution and multiple cross-validation experiments should be conducted to establish reliable evaluation metrics (Collins, 2018).

POLYNOMIAL REGRESSION

Polynomial regression is a method of modeling nonlinear relationships between predictors and outcomes in linear regression. It applies a nonlinear transformation to the features before calculating a linear relationship between them, allowing for the creation of predictive models with greater complexity than those described by standard linear equations. Polynomial regression can be used to address more complex issues, such as multiple interactions and higher order terms, as well as situations where a nonlinear effect appears in data; however, it can also result in overfitting if not applied correctly. This practice should be avoided by using cross-validation techniques whenever possible to ensure accurate prediction models are created (Giancarlo Zaccone, 2017).

PRINCIPAL COMPONENT ANALYSIS ALGORITHM ON SCIKIT-LEARN LIBRARY OF PYDROID3

Principal Component Analysis (PCA) scales the dimensionality of large data sets, by uncovering patterns and correlations within the data. *Scikit-learn*, provides an effective application of PCA to analyze large datasets searching for meaningful patterns. *Scikit-learn*'s PCA functionality utilizes a flexible API which allows users to control parameters such as the count of components, to retain and nature of transformation applied on the data (whether linear or kernel). Furthermore, it provides users with visualization tools such as scree plots and biplots to gain further insight into their data. Ultimately, *Scikit-learn*'s PCA module is robust enough to be used on large datasets while still yielding high performance results, making it a valuable tool in the exploratory phase of any effective Machine Learning workflow (Cross, 2015).

XGBOOST ALGORITHM ON SCIKIT-LEARN LIBRARY OF PYDROID3

XGBoost is a powerful and efficient machine learning algorithm. It is highly scalable implementation of the gradient boosting algorithm. It has been used in many different domains to solve many problems. XGBoost has been developed by Tianqi Chen and company. XGBoost has gained much popularity in applied machine learning field due to its remarkable performance gains over traditional algorithms, ability to scale across multi-dimensional data sets, and easy integration with existing frameworks such as *Scikit-learn* (Wade, 2020).

XGBoost library offers fast, accurate and efficient solution for many problems including regression and classification tasks. XGBoost leverages several techniques including regularization techniques such as L1 and L2 penalty, tree pruning techniques which helps reduce overfitting and various optimization techniques such as coordinate descent making it easier to tune hyperparameters. XGBoost is increasingly being used by data scientists in real world applications thanks to its speed, scalability, and accuracy (Wade, 2020).

K-MEANS CLUSTERING ALGORITHM ON SCIKIT-LEARN LIBRARY OF PYDROID3

K-nearest neighbors, or KNN machine learning algorithm is usually deployed for cataloguing and regression tasks on datasets. This technique searches the training

dataset to find the most similar instances to each test instance and classifies it based on its neighbor's class labels. It works by finding the Euclidean distance between each point and its nearest neighbors, who are then assigned votes equal to their relative distance from the query data point. The label of the majority group of data points is then assigned as the same label for that query data point. This ensures robust accuracy, limited computational time and an easy-to-interpret output since it relies less on mathematical functions compared to other algorithms. Overall, KNN can be effectively applied when working with large datasets because of its ability to quickly group similar features together in unseen datasets (Vanderplas, 2017).

KNN is a form of unsupervised type of algorithm that divides the fed dataset into collections or clusters. The procedure divides the data points into clusters so that the totality of distance square between each datapoint and the center of the cluster is effectively minimalized. This clustering method is popular because it's easy to understand and implement, and it can be parallelized to speed up computation time(Vanderplas, 2017).

Inputs comprises of the Data set, Number of clusters denoted by K, and a distance function to measure the distance between two observations (e.g., Euclidean distance). The K-mean algorithm produces output inform of clusters of input data, where each cluster contains observations with similar characteristics, and each observation belongs to only one cluster

SVM OR SUPPORT VECTOR MACHINE

Support Vector Machine is machine learning procedure that is used for organization of data and can be employed to solve any data problem which comprises of two categories. The algorithm, in general, finds the hyperplane that divides the data into two classes such that the distance of each point from this plane is as small as possible. This approach is rooted in statistical learning theory (Taweh Beysolow, 2018).

Finding the optimum hyperplane that can distinguish between the various classes in the data is the aim of an SVM. The nearest points to this hyperplane are referred to as support vectors, and this boundary is known as the decision boundary. The objective is to clearly differentiate the various classes by placing the decision border as far from any class's closest data points as is practical. Due to its ability to frequently locate a decision boundary that is more effective than those found by other algorithms, SVM is excellent when dealing with high-dimensional data (Pardalos et al., 2016) Hao & Ho, 2019.

CONCLUSION

Scikit-learn is a powerful and user-friendly library for the Python programming language. It is one of the most used libraries among developers because of its comprehensive set of tools, which includes modules for data pre-processing, model building, model evaluation and selection. *Scikit-learn* also supports supervised and unsupervised learning methods like SVM, as well as factor analysis, feature extraction and reduction of dimensionality techniques such as PCA. Furthermore, its algorithms are easy to use thanks to clear documentation and tutorials providing step by step explanations which makes it ideal for learning machine learning.

The benefits of implementing Machine learning mobile devices are now abundantly evident as an increasingly mobile workforce changes its focus to remote work. With the aid of smartphones and tablets, people can perform data cleansing, model training, model fitting, optimization, and prediction—all while on the go. This promises greater independence and higher productivity because they can better manage their time, keep organized, manage projects digitally, and start activities more quickly. Developers have unprecedented access to sample code and the opportunity for real-time cooperation when linked via a mobile device.

REFERENCES

Apeltsin, L. (2021). *Data Science Bookcamp*. Simon and Schuster.

Beysolow, T. (2018). *Applied natural language processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing*. Ca Apress. doi:10.1007/978-1-4842-3733-5

Blanco-Silva, F. J. (2015). *Mastering SciPy*. Packt Publishing Ltd.

Bressert, E. (2013). *SciPy and NumPy*. O'reilly.

Bruce, P., & Bruce, A. (2017). *Practical Statistics for Data Scientists*. O'Reilly Media, Inc.

Chen, D. Y. (2018). *Pandas for everyone: Python data analysis*. Addison-Wesley.

Chi, A., Yik, C., & Yuen, K. (2017). *Matplotlib 2.x by example: multi-dimensional charts, graphs, and plots*. Packt Publishing Ltd.

Cohen, R., Tao Wang, & Springerlink. (2014). *Android Application Development for the Intel Platform*. Apress.

Collins, R. (2018). *Scikit-Learn in Details*. Independently Published.

Cross, R. (2015). *Principal component analysis*. Clanrye Intl.

Garreta, R. & Moncecchi, G. (2013). *Learning scikit-learn: Machine learning in Python: Experience the benefits of machine learning techniques by applying them to real-world problems using Python and the open source scikit-learn library*. Packt Publishing Ltd.

GitHub. (n.d.). [Code Sharing]. GitHub. http://github.com

Goerzen, J. (2004). *Foundations of Python Network Programming*. Apress. doi:10.1007/978-1-4302-0752-8

Gupta, P. (2021). *Practical data science with Jupyter: Explore data cleaning, pre-processing, data wrangling, feature engineering and machine learning using Python and Jupyter*. Bpb Publications.

Hackeling, G. (2017). *Mastering machine learning with scikit-learn: Learning to implement and evaluate machine learning solutions with scikit-learn*. Packt Publishing Ltd.

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring Network Structure, Dynamics, and Function using NetworkX. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th Python in Science Conference* (pp. 11–15).

Hao, J., & Ho, T. K. (2019). Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language. *Journal of Educational and Behavioral Statistics*, *107699861983224*(3), 348–361. doi:10.3102/1076998619832248

Idris, I. (2011). *NumPy 1.5: Beginner's guide*. Packt Publishing.

Kane, F. (2017). *Hands-On Data Science and Python Machine Learning*. Birmingham Packt Publishing.

Langtangen, H. P. (2016). *A Primer on Scientific Programming with Python*. Springer.

LinuxHint. (n.d.). [Programming]. Linux Hints. https://linuxhint.com

Lutz, M. (2006). *Programming Python*. O'Reilly Media, Inc.

Lutz, M. (2013). *Learning Python*. O'Reilly Media, Inc.

Mirjalili, S. (2019). *PYTHON MACHINE LEARNING - THIRD EDITION: machine learning and deep learning with python, scikit... -Learn, and tensorflow 2*. Packt Publishing Limited.

Pardalos, P. M., Piero Conca, Giuffrida, G., Nicosia, G., & Springerlink. (2016). Machine Learning, Optimization, and Big Data. *Second International Workshop, MOD 2016*. Springer International Publishing.

Platt, E. L. (2019). *Network Science with Python and NetworkX Quick Start Guide*. Packt Publishing Ltd.

Rajamani, S. K., & Iyer, R. (2022). Development Of an Android Mobile Phone Application for Finding ClosedLoop, Analytical Solutions to Dense Linear, Algebraic Equations for The Purpose of Mathematical Modelling in Healthcare and Neuroscience Research. *NeuroQuantology: An Interdisciplinary Journal of Neuroscience and Quantum Physics*, *20*, 4959–4973. doi:10.6084/m9.figshare.c.6156024.v1

Vanderplas, J. T. (2017). *Python data science handbook: Essential tools for working with data*. O'reilly, Cop.

Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., & Mueller, A. (2015). Scikit-learn. *GetMobile: Mobile Computing and Communications*, *19*(1), 29–33. doi:10.1145/2786984.2786995

Wade, C. (2020). *Hands-on gradient boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with python*. Packt Publishing.

Weisberg, S. (2013). *Applied Linear Regression*. John Wiley & Sons.

Zaccone, G. (2017). *Deep learning with TensorFlow*. Uk Packt Publishing.

Zhang, Y., & Rajapakse, J. C. (2009). *Machine Learning in Bioinformatics*. John Wiley & Sons.