# Title

## Introduction

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 20121. In other words, Anaconda is a package manager, an environment manager, a Python/R data science distribution, and a collection of over 7,500+ open-source packages. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free2. Additionally, a desktop graphical user interface (GUI) is provided through the Navigator. Thankfully, for the purposes of this thesis the Anaconda Individual Edition is more than enough, therefore any paid editions were not necessary.

A conda environment is a directory that contains a specific collection of conda packages that you have installed. For example, you may have one environment with NumPy 1.7 and its dependencies, and another environment with NumPy 1.6 for legacy testing. If one environment is changed, the other environments are not affected. Environments can be easily activated or deactivated, which is how one can switch between them. Environments can also be shared with others by giving them a copy of the environment.yaml file.

A conda package is a compressed tarball file (.tar.bz2) or .conda file that contains system-level libraries, Python or other modules, executable programs and other components, metadata under the info/ directory or a collection of files that are installed directly into an install prefix. Conda keeps track of the dependencies between packages and platforms. The conda package format is identical across platforms and operating systems. Only files, including symbolic links, are part of a conda package. Directories are not included. Directories are created and removed as needed, but you cannot create an empty directory from the tar archive directly.

A conda package format remains the same between different platforms and operational systems. As mentioned, for a conda package to be installed, it is advised to use the <conda install [packagename]> command (but *pip install* might also work). During the process of installation, the files are extracted from the install prefix (apart from the files in the </info> directory).

## Benefits of Anaconda

Over 250 packages are installed automatically upon completion of main installation and there are over 7,500 additional open-source packages (including R) available that can be individually installed from the Anaconda repository with the *conda install* command. Moreover, thousands of packages are available from Anaconda.org while there is also the option to use the original *pip install* command as in some cases they can work together with conda packages (it is advised however to prefer the conda command to install). Finally, there is also the option to create custom packages through the *conda build* command and share them to Anaconda.org, PyPI and other repositories2.

## Conda in the context of the Thesis

Since the operation system used in this thesis was Windows 10, anaconda has proven a vital tool to use packages that otherwise would not have been available. As such it is considered a vital component and essentially the backbone of all future work. Specifically, Conda has proven tremendously useful in creating the virtual environment necessary to run flask, as well as perform Tensorflow oriented tasks.

# Bibliography

[1] "About Anaconda" (<https://web.archive.org/web/20200419034550/https://www.-anaconda.com/media-kit/>)

[2] Anaconda Documentation (https://docs.anaconda.com/)