Data Mining and Decision Systems Workshop 1
Jupyter Notebook and PIP environment

Aims of the workshop

The aim of this session is to introduce you to the Jupyter Notebook environment, showcased in this week's lectures. For some this will be the first introduction to Python as a programming language. It is important to take time to understand these basic concepts, and explore what they can do.

Please see 'Useful Information' below on how to lookup certain Python functionality. The concept behind this workshop is about discovery, and experimentation surrounding topics covered so far.

Feel free to discuss the work with peers, or with any member of the teaching staff.

Useful Information

Throughout this workshop you may find the following useful.

Python Documentation

https://docs.python.org/3.7/

This allows you to lookup core language features of Python 3.7 as well as tangential information about the Python Language.

Jupyter Notebook Basics

https://jupyter.org/try
The above may be useful as an introduction to the Jupyter Notebook platform, and how to use it. Go to “Try Classic Notebook”. Once the Notebook opens, there should be a link on the page called “Notebook basics”.

Pandas Documentation

https://pandas.pydata.org/pandas-docs/stable/index.html

Similarly to the above, the pandas library itself is well-documented outlining function signatures, alongside example use-cases.

Functions of note: read_csv, head, describe, sample

Reminder

We encourage you to discuss the contents of the workshop with the delivery team, and any findings you gather from the session.

Workshops are not isolated, if you have questions from previous weeks, or lecture content, please come and talk to us.

The contents of this workshop are not intended to be 100% complete within the 2 hours; as such it's expected that some of this work be completed outside of the session. Exercises herein represent an example of what to do; feel free to expand upon this.

Running A Jupyter/IPython Notebook

A jupyter notebook is a server which runs on the local machine. It will use whichever directory it is invoked within as the root folder. It is then able to see all sub-folders within that. This is currently installed on the lab machines; however, if you are wishing to install this at home, you may need to follow pip instructions for installing jupyter.

Jupyter Notebook can be launched from the command line by invoking:

```
jupyter notebook
```

This will start a server, typically on http://localhost:8888/tree, and should automatically open a new tab in the browser.

Jupyter notebook server uses a token, generated at launch, to authenticate access. By default, the notebook server is accessible to anybody with network access to your machine over the port, with the correct token.

If you are asked for a password or token, you can always invoke:

```
jupyter notebook list
```

This will provide you a list of currently active Notebook servers (you can run multiple, on different ports, for different folders), along with their token, all within a single URL.

PIP and packages

PIP is python's packaging tool, it enables the installation of libraries. If you tried to import pandas when starting your notebook, you may have noticed that it gives you an error. This is because, by default, pandas is not installed within the super lab.

Python has a rich community backing, with several libraries available over at PyPI which users can install. Users can even make their own libraries and host them online for others to use. These are similar to standard libraries you may be familiar with from other languages such as C#, or Javascript.

As the super lab image does not provide administrative access, installing packages to the system level is not permitted. However, pip provides a --user flag for installing packages to the user-profile instead.

Most packages can be installed by using

```
pip install packageName
```

Remember, you can always use `pip help install` for specific help information on installing.

Python 3.7

Python is a powerful scripting language, it follows similar conventions to programming languages you may already know. E.g Variable Assignment. The main difference with Python is how it scopes.

In C# you may use {} to denote a code block for an ‘if’ or ‘for’ loop. In Python you would simply indent the code. Notice how we use : to indicate that we’re about to go into a new block.

```
nine_am_lecture = True
if nine_am_lecture:
        print("Why!")
        tired = True

for i in range(10):
        print(i)

def my_function(param1, param2):
        return param1 + param2

print( my_function( 5, 8 ) )
print( "My Number: " + str(my_function( 5, 8 )) )
print( tired )
```

Exercise 1: Launch a Jupyter Notebook, and explore the interface and controls. Look at the useful information section above under Jupyter Notebook Basics. Try executing the python examples above, and noting their output.

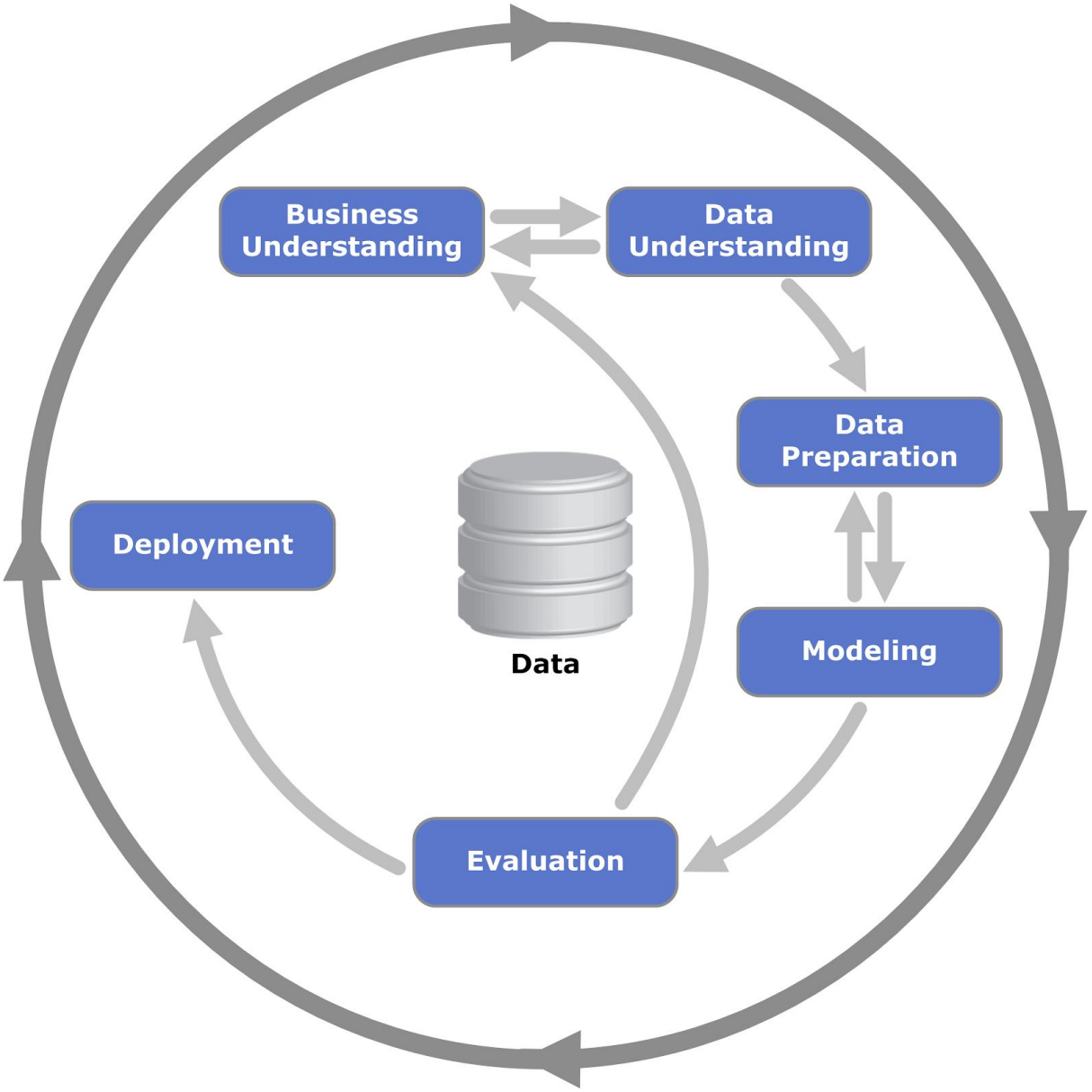
Exercise 2: Using pip, and the --user flag, install the pandas library to your machine. You may need to rerun the Jupyter Notebook session.

Exercise 3: Create a new notebook, import pandas. Create the following DataFrame:

```
data = {        'Country': ['Belgium',  'India',  'Brazil'],
                'Capital': ['Brussels',  'New Delhi',  'Brasilia'],
                'Population': [11190846, 1303171035, 207847528]
        }
df = pd.DataFrame(data, columns=['Country',  'Capital',  'Population'])
```

This makes use of a Python Data Structure called a Dictionary. It is a string-value store. Display this DataFrame within the Notebook.

Exercise 4: Read in the train.csv Titanic Dataset available from Canvas (under Modules Week 2). With the resultant DataFrame, use describe, head, and sample. Remember to look up the documentation for these commands to see what else they can do. What do these commands do?



Exercise 5: Consider the CRISP-DM Methodology. Which phase would the above tasks belong to? Why?

Exercise 6: In this week's lecture, we looked at calculating a subset of the number of patterns for the Titanic Dataset. Using the Full Description, and the data you have, calculate what the total number of patterns might be. You should be able to explain the choices you make for each attribute. Remember to consider the relationship between an attribute and what it stands for. Try researching the domain of the Titanic to see if it can help solve any questions for a given attribute (E.g Cabin, Ticket Number).

The Extended Exercises are optional, and are offered as an advanced supplement for those who have completed the existing work and wish to expand on their knowledge and challenge themselves further.

Extended Exercise 1: Look up Python Virtual environments. These enable more than a singular repository to use for python projects. If you have two different projects, with vastly different library dependencies, it makes sense that you may want to keep them separate. This is especially useful if one project needs a specific older version of a library; impossible to adhere to with only a system-level pool of installed packages. These encapsulated environments can then be sourced by your local shell/commandline so that python knows where to go for its packages.