Sprint 02: Induction to Data Science

*“Move/Backup Application”*

|  |  |
| --- | --- |
| **Stakeholder** |  |
| **Business contact** |  |
| **Duration** | 1-2 days |

# Overview:

## Key findings:

## Sprint Aim:

To evaluate and better understand Tkinter and how it can be used to make a basic application

## Sprint Objectives:

1. To develop knowledge of Python Tkinter
2. To understand how files are moved and copied
3. To turn a Python file into an Executable file
4. To document the skills you have learnt.

# Results

Working on two image classification approaches allow

Contents

[Overview: 1](#_Toc20210185)

[Key findings: 1](#_Toc20210186)

[Sprint Aim: 1](#_Toc20210187)

[Sprint Objectives: 1](#_Toc20210188)

[Results 1](#_Toc20210189)

[Objective 1: To develop knowledge of Python Tkinter. 2](#_Toc20210190)

[Objective 2: To understand how files are moved and copied. 3](#_Toc20210191)

[Objective 3: To turn a Python file into an Executable 4](#_Toc20210192)

[Objective 4: To document the skills you have learnt. 5](#_Toc20210193)

## Objective 1: To develop knowledge of Python Tkinter

**What is Tkinter?**

Tkinter is a package in Python that allow python to bind to a Tk GUI toolkit. This is a basic GUI uses widgets to add functionality to the window.

Tkinter is a package that is already installed within python, therefore you only have import the modules you want instead of having to install Tkinter first.

**Useful Code**

**From tkinter import <module>** - This installs all modules from Tkinter

**Ttk** – This is a themed text widget, allowing a person to create a widget that has more functionality than what Tkinter’s basic widgets offer

**Filedialog** – This module with open and save functions within the Tkinter GUI.

**Askdirectory** – This asks for a directory and returns the folder name

**Askopenfilename** – This asks for the file and returns its file name

**windowVar = Tk()** - This declares that we are using this as our main window

**windowVar.mainloop()** - The method mainloop() is used to run the window and is an infinite loop. It keeps our window open and continuously waits for an event to occur; it only stops running when the window is closed

**lbl = Label(windowVar, text =”this is sample text”)** - This sets up a label that will display the set text.

**Btn = ttk.Button(windowVar, text = “Button”, command=<put your function name here>)** – This sets up a button on your window, once clicked it will run your command/function

## Objective 2: To understand how files are moved and copied

**Basic Pseudo**

**Moving folders and files**

1. Select the folder/file
2. Save the folder/file path
3. Select the new folder you wish to move the folder/file to
4. Move the folder/file

**Copying files**

1. Select the file
2. Save the file path
3. Select the new folder you wish to copy the folder/file to
4. Copy the file the folder

**Copying Folders**

1. Select the file
2. Save the file path
3. Select the new folder you wish to move the folder/file to
4. Copy the folder/file and rename as a copy
5. Move the new copy file into the selected new folder

**Useful code**

**Shutil.move(sourcePath,destinationPath)** – Moves the file/folder to the set destination path

**Shutil.copy(sourcePath,destinationPath)** – Copies the file (using its path) to the destination path (you can also set a new name for the file)

**Shutil.copytree(sourcePath,destinationPath) –** copies the folder and its contents (hence the folder’s tree) to the destination path

## Objective 3: How to turn a Python file into an Executable file

Python usually runs on its own interprets as long as Python is installed on the set machine. However, sometimes this is not possible, or the user may wish to run the program without having to install python and run the code; hence why making the .py file into a.exe would be beneficial.

The two packages commonly used to turn a Python file into an Executable are

* pyInstaller
* cx\_freeze

In my project I used cx\_freeze due to security limitations with pyInstaller.

Cxfreeze <pythonfile> - This runs a set of steps that turns the file into an executable. However the folder the executable is saved to must remain in that exact folder, otherwise it can’t run.

Pyinstaller <pythonfile> - this runs a set of steps that turns the file into an executable.

## Objective 4: To document the skills you have learnt.

I’ve learnt;

1. How to use Tkinter
2. How files are moved and copied using Python
3. How to turn a Python file into an Executable file

**Featured Requests**

* Show the user what path has been selected in a small box etc.
* Add a date and time onto the file – Shows when it was moved/copied
* Allow the GUI to be more interactive – more colours and add a Menu instead