**Program 1.Python Packages for Machine Learning and Deep Learning .**

**NumPy:**

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions.

**Keras:**

It provides many inbuilt methods for groping, combining and filtering data. Keras is a very popular Machine Learning library for Python. It is a high-level neural networks API .

**Pandas**

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. It provides many inbuilt methods for grouping, combining and filtering data.

**Matplotlib**

Matplotlib is a very popular Python library for data visualization. It is a 2D plotting library used for creating 2D graphs and plots. It provides various kinds of graphs and plots for data visualization, viz., histogram, bar chats, etc.

**Scikit-learn**

Scikit-learn is one of the most popular ML libraries for classical ML algorithms. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with ML. 

Install Anaconda software (<https://repo.anaconda.com/archive/Anaconda3-2022.05-Windows-x86_64.exe>)

Open anaconda prompt and set run as administartor

Type

pip install **numPy**

pip install **pandas**

pip install **keras**

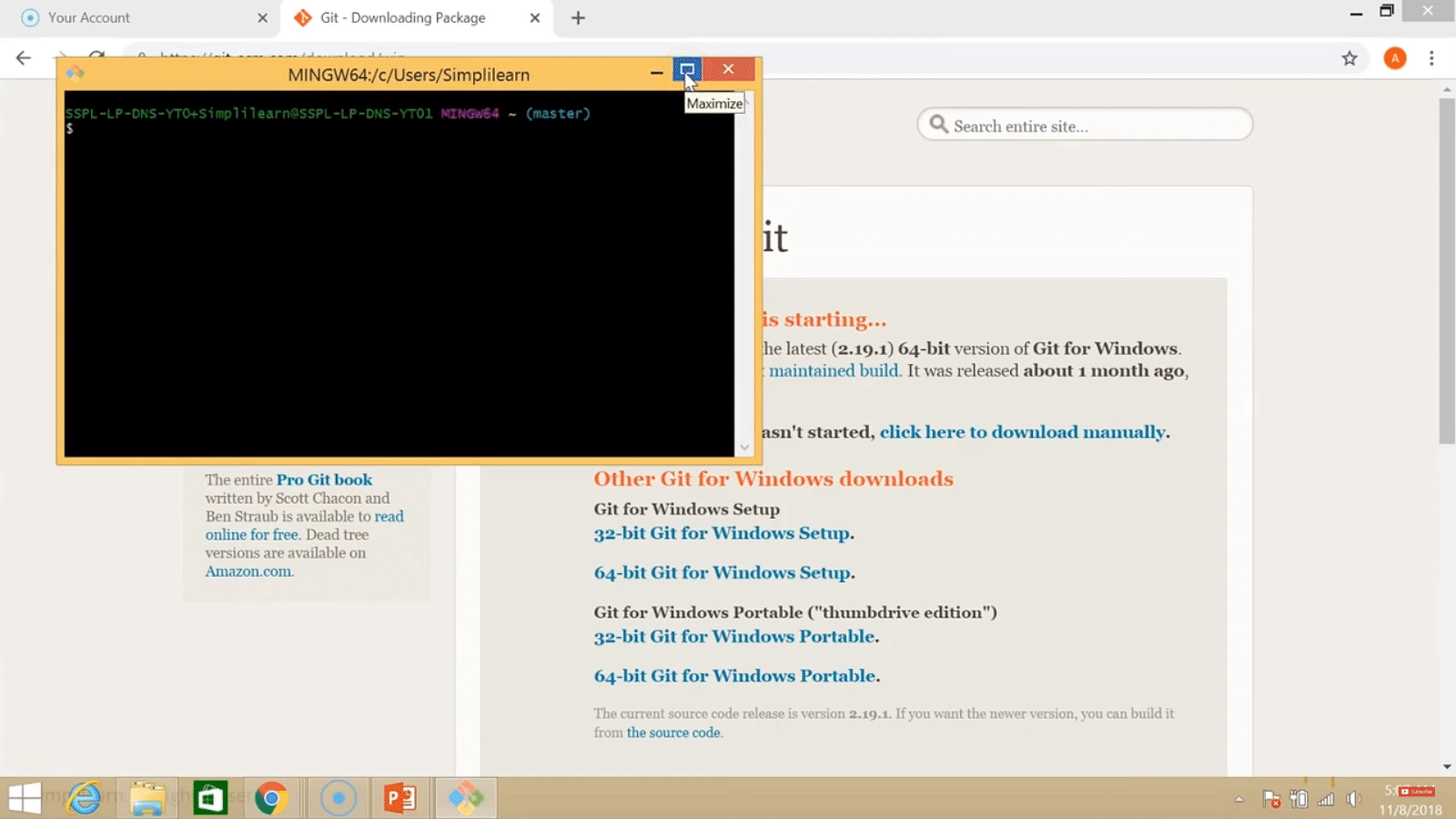
pip install **matplotlib**

pip install **scikit-learn**

Program 2.Git Installation on Windows

### **Step 1:**

Download the [latest version of Git](https://git-scm.com/downloads) and choose the 64/32 bit version. After the file is downloaded, install it in the system. Once installed, select Launch the Git Bash, then click on finish. The Git Bash is now launched.



### **Step 2:**

Check the Git version:

|  |
| --- |
| $ git --version |

### **Step 3:**

For any help, use the following command:

|  |
| --- |
| $ git help config |

This command will lead you to a browser of [config commands](https://www.simplilearn.com/tutorials/git-tutorial/git-commands). Basically, the help the command provides a manual from the help page for the command just following it (here, it's config).

Another way to use the same command is as follows:

|  |
| --- |
| $ git config --help |

### **Step 4:**

Create a local directory using the following command:

|  |
| --- |
| $ mkdir test  $ cd test |

### **Step 5:**

The next step is to initialize the directory:

|  |
| --- |
| $ git init |

### **Step 6:**

Go to the folder where "test" is created and create a text document named "demo." Open "demo" and put any content, like "Hello Simplilearn." Save and close the file.

### **Step 7:**

Enter the Git bash interface and type in the following command to check the status:

|  |
| --- |
| $ git status |

### **Step 8:**

Add the "demo" to the current directory using the following command:

|  |
| --- |
| $ git add demo.txt |

### **Step 9:**

Next, make a commit using the following command:

|  |
| --- |
| $ git commit -m "committing a text file" |

### **Step 10:**

Link the Git to a [Github](https://www.simplilearn.com/tutorials/git-tutorial/what-is-github" \o "Github" \t "_blank) Account:

|  |
| --- |
| $ git config --global user.username |

Note: simplilearn-github is the username on the Github account.

### **Step 11:**

Open your Github account and create a new repository with the name "test\_demo" and click on "Create repository." This is the remote repository. Next, copy the link of "test\_demo."

### **Step 12:**

Go back to Git bash and link the remote and local repository using the following command:

|  |
| --- |
| $ git remote add origin <link> |

Here, <link> is the link copied in the previous step.

### **Step 13:**

Push the local file onto the remote repository using the following command:

|  |
| --- |
| $ git push origin master |

### **Step 14:**

Move back to Github and click on "test\_demo" and check if the local file "demo.txt" is pushed to this repository.

How to Launch Git in Windows?

There are two methods to launch git in windows. One is launching git using a bash scripting shell with the help of the command line and another is launching git using a graphical user interface.

1. To launch git via bash scripting shell,  
   First, open the window and search for git bash and open it.
2. To launch git via graphical user interface(GUI), similarly, first open the window and search for git GUI and click on the application icon and open it.

Configure GitHub Credentials

You can configure your local GitHub installation with credentials by using the following commands. Also, don't forget to add your own GitHub credentials for username and email address.

1. git config –global user.n  
   ame "github\_username"
2. git config –global user.e  
   mail "email\_address"

# Program 3.How to Push an Existing Project to GitHub

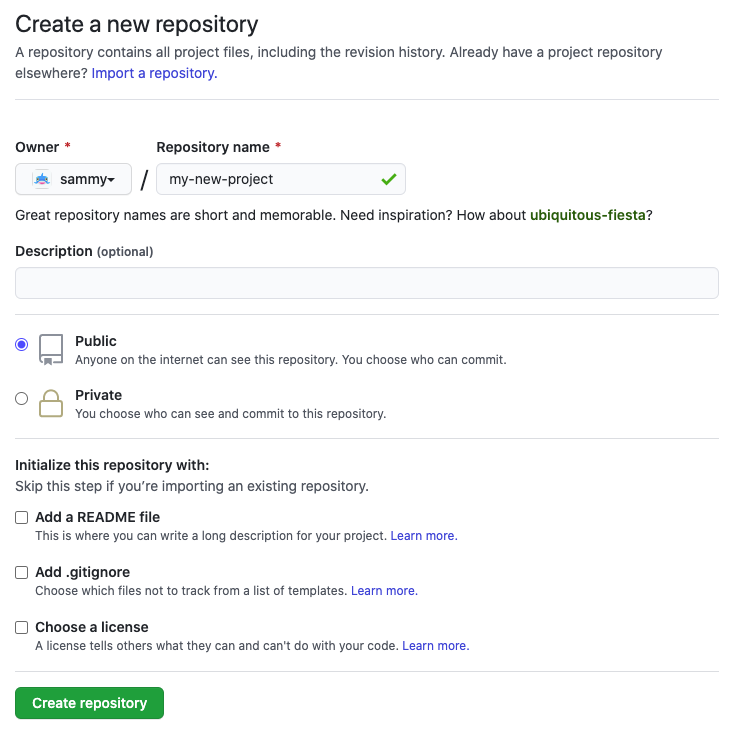
## Prerequisites

To initialize the repo and push it to GitHub you’ll need:

1. [A free GitHub Account](https://github.com/)
2. [git installed on your local machine](https://www.digitalocean.com/community/tutorials/how-to-contribute-to-open-source-getting-started-with-git#check-if-git-is-installed)

## Step 1 — Create a new GitHub Repo

Sign in to GitHub and [create a new empty repo](https://github.com/new). You can choose to either initialize a README or not. It doesn’t really matter because we’re just going to override everything in this remote repository anyways.



## Step 2 — Initialize Git in the project folder

From your terminal, run the following commands after navigating to the folder you would like to add.

### Initialize the Git Repo

Make sure you are in the root directory of the project you want to push to GitHub and run:

**Note:** If you already have an initialized Git repository, you can skip this command.

1. git init

This step creates a hidden .git directory in your project folder, which the git software recognizes and uses to store all the metadata and version history for the project.

### Add the files to Git index

1. git add -A

The git add command is used to tell git which files to include in a commit, and the -A (or --all)

The git add command is used to tell git which files to include in a commit, and the -A (or --all) argument means “include all”.

### Commit Added Files

1. git commit -m 'Added my project'

The git commit command creates a new commit with all files that have been “added”. The -m (or --message) sets the message that will be included alongside the commit, used for future reference to understand the commit. In this case, the message is: 'Added my project'.

### Add a new remote origin

1. git remote add origin git@github.com:sammy/my-new-project.git

### Push to GitHub

1. git push -u -f origin main

### All together

1. git init
2. git add -A
3. git commit -m 'Added my project'
4. git remote add origin git@github.com:sammy/my-new-project.git
5. git push -u -f origin main