EXPLORE | DIGITAL SKILLS

Setting up Comet

Version control

Learning Objectives

Learning Objectives	Why this is important
Version Control	Project management Reproducibility Collaboration
Comet.ml	Easy version control You choose what to keep Records per experiment

Tools & guides

Have your Comet starter notebook open (found under trains) if you would like to follow along for the “Your first experiment” section.

Comet

Setting up your account

Step 1

- Visit https://www.comet.ml and **create a free account**

Step 2

- Sign up with GitHub or fill in your details

Step 3

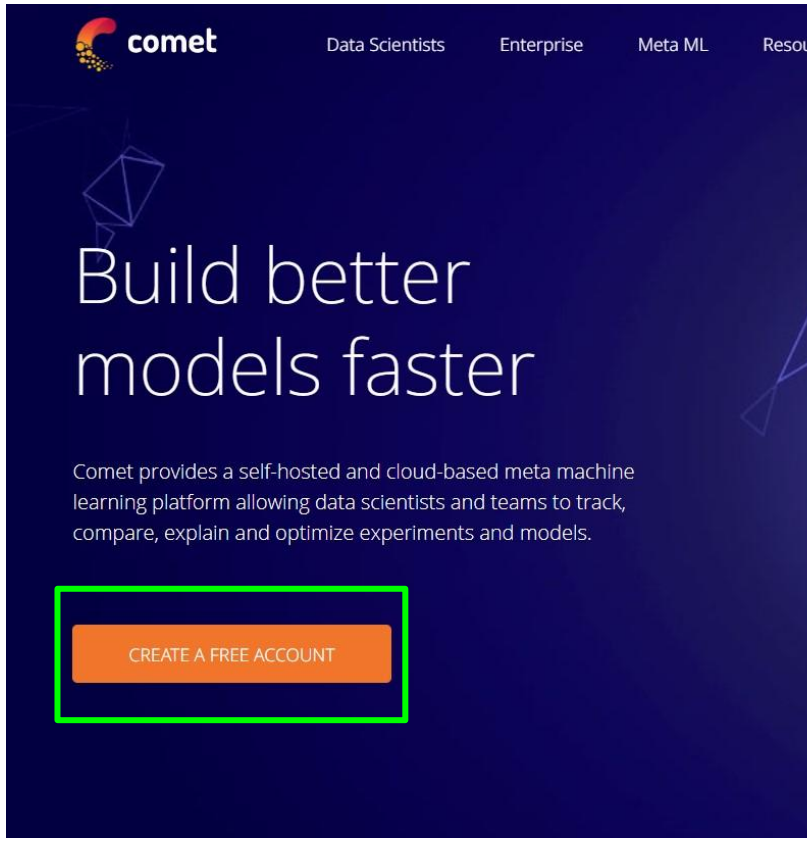
- Choose the **individual** option

Step 4

- On the quick start page, have a look at the sklearn guide

Step 5

- `pip install comet_ml` from your notebook or shell (If you are using colab you will have to do this every time)



Comet

Setting up your account

Step 1

- Visit https://www.comet.ml and **create a free account**

Step 2

- **Sign up** with GitHub or fill in your details

Step 3

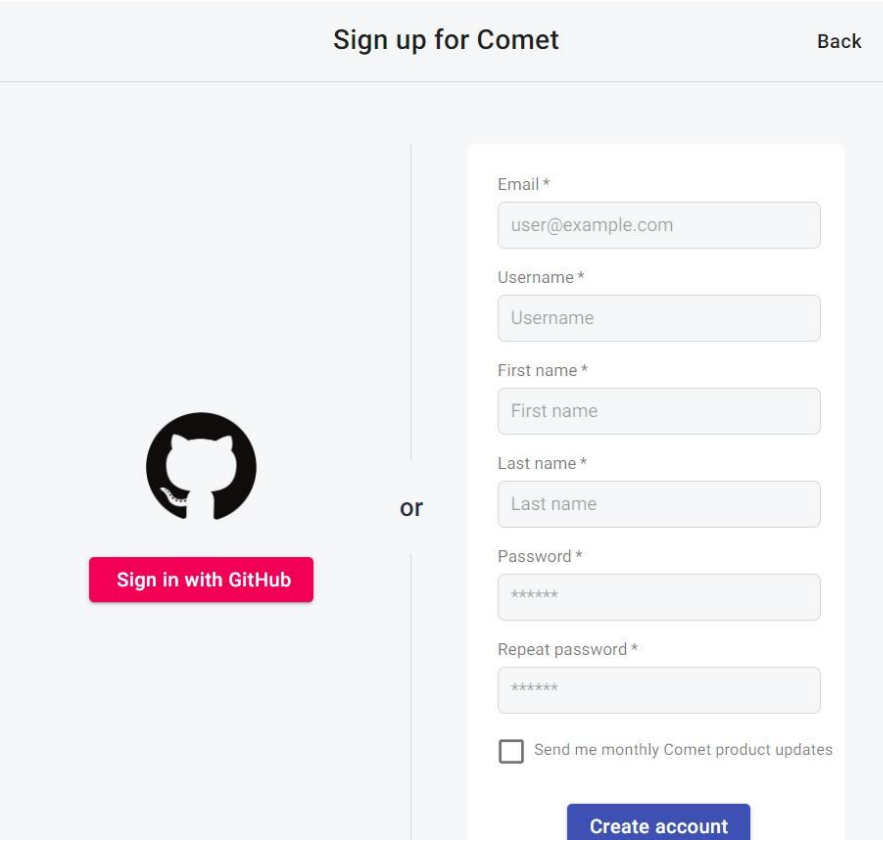
- Choose the **individual** option

Step 4

- On the quick start page, have a look at the sklearn guide

Step 5

- `pip install comet_ml` from your notebook or shell (If you are using colab you will have to do this every time)



Comet

Setting up your account

Step 1

- Visit https://www.comet.ml and **create a free account**

Step 2

- **Sign up** with GitHub or fill in your details

Step 3

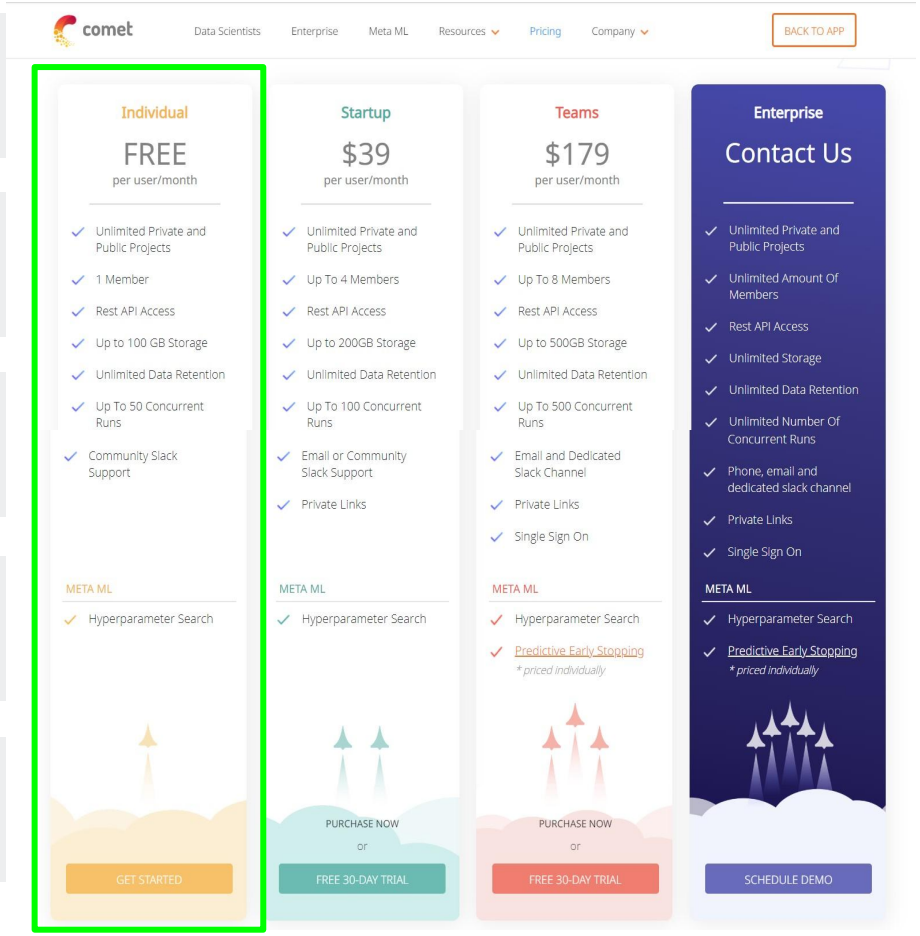
- Choose the **individual** option

Step 4

- On the quick start page, have a look at the sklearn guide

Step 5

- `pip install comet_ml` from your notebook or shell (If you are using colab you will have to do this every time)



Comet

Setting up your account

Step 1

- Visit https://www.comet.ml and **create a free account**

Step 2

- **Sign up** with GitHub or fill in your details

Step 3

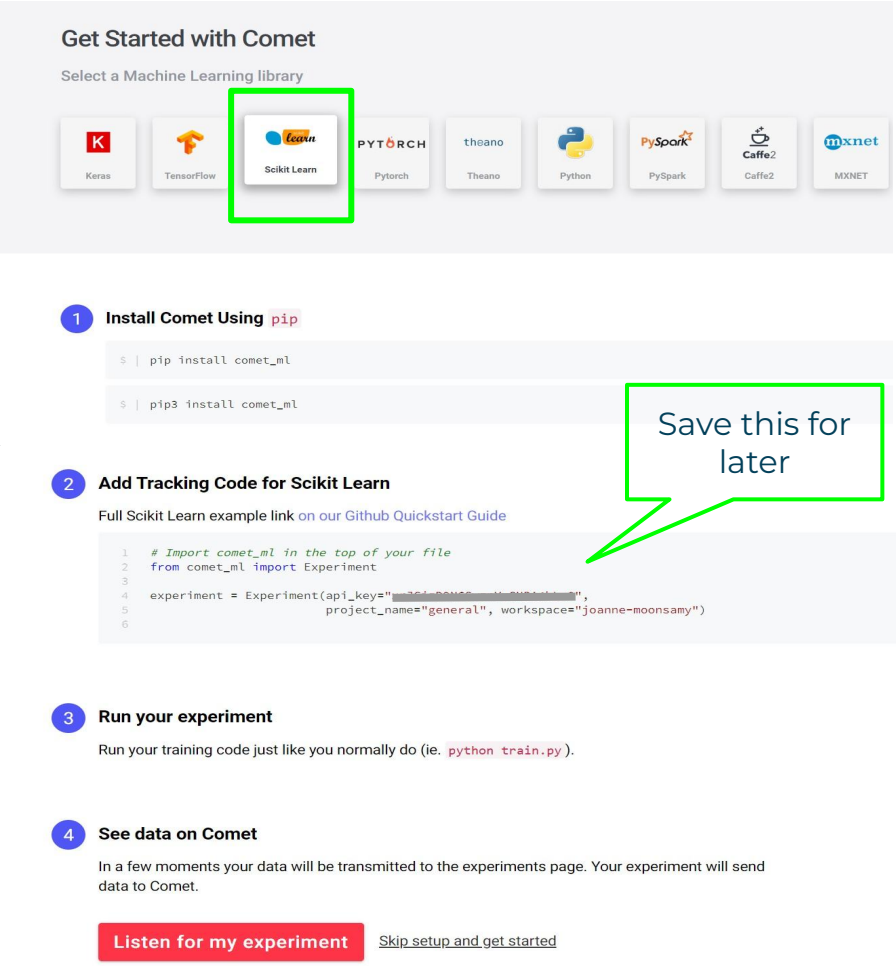
- Choose the **individual** option

Step 4

- On the quick start page, have a look at the **sklearn guide**

Step 5

- `pip install comet_ml` from your notebook or shell (If you are using colab you will have to do this every time)



Comet

Setting up your account

Step 1

- Visit https://www.comet.ml and **create a free account**

Step 2

- **Sign up** with GitHub or fill in your details

Step 3

- Choose the **individual** option

Step 4

- On the quick start page, have a look at the **sklearn guide**

Step 5

- `pip install comet_ml` from your notebook or shell (If you are using colab you will have to do this every time)

```
[1] !pip install comet_ml
  Collecting comet_ml
    Downloading https://files.pythonhosted.org/packages/d3/c3/0e46ea9ccb0a761f005dbde4fe0bd884308251e1a854987ef610659fc09c/comet_ml-3.1.
      |████████████████████████████████| 194kB 1.3MB/s
  Collecting comet-git-pure>=0.19.11
    Downloading https://files.pythonhosted.org/packages/56/2b/c1ca11a237e3fcbe0c1ea53134901f30f3aa657e5ea141dccaae0f46df0e/comet_git_pur
      |████████████████████████████████| 409kB 2.5MB/s
  Requirement already satisfied: jsonschema<3.1.0,>=2.6.0 in /usr/local/lib/python3.6/dist-packages (from comet_ml) (2.6.0)
  Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from comet_ml) (1.12.0)
  Collecting websocket-client>=0.55.0
    Downloading https://files.pythonhosted.org/packages/4c/5f/f61b420143ed1c8dc69f9eaec5ff1ac36109d52c80de49d66e0c36c3dfdf/websocket_cli
      |████████████████████████████████| 204kB 3.2MB/s
  Collecting wurlitzer>=1.0.2
    Downloading https://files.pythonhosted.org/packages/24/5e/f3bd8443bfdf96d2f5d10097d301076a9eb55637b7864e52d2d1a4d8c72a/wurlitzer-2.0
  Collecting netifaces>=0.10.7
    Downloading https://files.pythonhosted.org/packages/0c/9b/c4c7eb09189548d45939a3d3a6b3d53979c67d124459b27a094c365c347f/netifaces-0.1
  Requirement already satisfied: requests>=2.18.4 in /usr/local/lib/python3.6/dist-packages (from comet_ml) (2.21.0)
  Collecting everett[ini]>=1.0.1; python_version >= "3.0"
    Downloading https://files.pythonhosted.org/packages/12/34/de70a3d913411e40ce84966f085b5da0c6df741e28c86721114dd290aaa0/everett-1.0.2
  Requirement already satisfied: nvidia-ml-py3>=7.352.0 in /usr/local/lib/python3.6/dist-packages (from comet_ml) (7.352.0)
  Requirement already satisfied: urllib3>=1.24.1 in /usr/local/lib/python3.6/dist-packages (from comet-git-pure>=0.19.11->comet_ml) (1.2
  Requirement already satisfied: certifi in /usr/local/lib/python3.6/dist-packages (from comet-git-pure>=0.19.11->comet_ml) (2019.11.28)
  Requirement already satisfied: idna<2.9,>=2.5 in /usr/local/lib/python3.6/dist-packages (from requests>=2.18.4->comet_ml) (2.8)
  Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from requests>=2.18.4->comet_ml) (3.0.
  Collecting configobj; extra == "ini"
    Downloading https://files.pythonhosted.org/packages/64/61/079eb60459c44929e684fa7d9e2fdca403f67d64dd9dbac27296be2e0fab/configobj-5.0
  Building wheels for collected packages: configobj
    Building wheel for configobj (setup.py) ... done
    Created wheel for configobj: filename=configobj-5.0.6-cp36-none-any.whl size=34546 sha256=e187785469ed4c714f732dc29e751bfac4377b5dd7
    Stored in directory: /root/.cache/pip/wheels/f1/e4/16/4981ca97c2d65106b49861e0b35e2660695be7219a2d351ee0
  Successfully built configobj
  Installing collected packages: comet-git-pure, websocket-client, wurlitzer, netifaces, configobj, everett, comet-ml
  Successfully installed comet-git-pure-0.19.15 comet-ml-3.1.3 configobj-5.0.6 everett-1.0.2 netifaces-0.10.9 websocket-client-0.57.0 wu
```

Get ready to start recording your own experiments!

Overview

Requirements

Setting up Comet account

Your first experiment

Conclusion

Notebook — *Recording your first experiment*

Step 1

- **Import comet** at the very top of your notebook

```
[ ] from comet_ml import Experiment
```

Step 2

- **Link your workspace** using the code from the starter-guide

You will see an API key button at the top of the page when you click on an experiment-
workspace to comet. (If a project is empty, the code below will autogenerate for you o

```
# Setting the API key (saved as environment variable)
experiment = Experiment(#api_key='your api key here'
                        project_name="general", workspace="jo-moon")
```

```
COMET INFO: Experiment is live on comet.ml https://www.comet.ml/jo-moon/ge
```

Step 3

- Create/run your notebook as you usually would

Import the rest of your necessary libraries as you usually would. For this demonstration we will be using the breast c
classification so we will also import that from sklearn.

```
[ ] import numpy as np
    from sklearn.datasets import load_breast_cancer
    from sklearn.model_selection import train_test_split, GridSearchCV
    from sklearn.linear_model import LogisticRegression
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.preprocessing import StandardScaler
    from sklearn.metrics import f1_score, precision_score, recall_score, confusion_matrix
```

Step 4

- After your model has run, **save all your metrics** as separate variables and **create dictionaries** for the data you want to log.

```
[ ] # Have a look at your dataset
    cancer = load_breast_cancer()
    print("cancer.keys(): {}".format(cancer.keys()))
    print("Shape of cancer data: {}\n".format(cancer.data.shape))
    print("Sample counts per class:\n{}".format(
          {n: v for n, v in zip(cancer.target_names, np.bincount(cancer.target))}))
    print("\nFeature names:\n{}".format(cancer.feature_names))
```

```
cancer.keys(): dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'filename'])
Shape of cancer data: (569, 30)

Sample counts per class:
{'malignant': 212, 'benign': 357}

Feature names:
['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
```

Step 5

- Log the necessary items using the commands : **`experiment.log_metrics()`** and **`experiment.log_parameters()`**

Step 6

- Remember to **`experiment.end()`** if you're using a **notebook**

Notebook — *Recording your first experiment*

Step 1

- **Import comet** at the very top of your notebook

```
[ ] from comet_ml import Experiment
```

Step 2

- **Link your workspace** using the code from the starter-guide

You will see an API key button at the top of the page when you click on an experiment-
workspace to comet. (If a project is empty, the code below will autogenerate for you o

```
# Setting the API key (saved as environment variable)
experiment = Experiment(#api_key='your api key here'
                        project_name="general", workspace="jo-moon")
```

```
COMET INFO: Experiment is live on comet.ml https://www.comet.ml/jo-moon/ge
```

Step 3

- Create/run your notebook as you usually would

Import the rest of your necessary libraries as you usually would. For this demonstration we will be using the breast c
classification so we will also import that from sklearn.

```
[ ] import numpy as np
    from sklearn.datasets import load_breast_cancer
    from sklearn.model_selection import train_test_split, GridSearchCV
    from sklearn.linear_model import LogisticRegression
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.preprocessing import StandardScaler
    from sklearn.metrics import f1_score, precision_score, recall_score, confusion_matrix
```

Step 4

- After your model has run, **save all your metrics** as separate variables and **create dictionaries** for the data you want to log.

```
[ ] # Have a look at your dataset
    cancer = load_breast_cancer()
    print("cancer.keys(): {}".format(cancer.keys()))
    print("Shape of cancer data: {}\n".format(cancer.data.shape))
    print("Sample counts per class:\n{}".format(
          {n: v for n, v in zip(cancer.target_names, np.bincount(cancer.target))}))
    print("\nFeature names:\n{}".format(cancer.feature_names))
```

Step 5

- Log the necessary items using the commands : **`experiment.log_metrics()`** and **`experiment.log_parameters()`**

```
cancer.keys(): dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'filename'])
Shape of cancer data: (569, 30)

Sample counts per class:
{'malignant': 212, 'benign': 357}

Feature names:
['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
```

Step 6

- Remember to **`experiment.end()`** if you're using a **notebook**

Notebook — *Recording your first experiment*

Step 1

- **Import comet** at the very top of your notebook

```
[ ] from comet_ml import Experiment
```

Step 2

- **Link your workspace** using the code from the starter-guide

You will see an API key button at the top of the page when you click on an experiment-
workspace to comet. (If a project is empty, the code below will autogenerate for you o

```
# Setting the API key (saved as environment variable)
experiment = Experiment(#api_key='your api key here'
                        project_name="general", workspace="jo-moon")
```

```
COMET INFO: Experiment is live on comet.ml https://www.comet.ml/jo-moon/ge
```

Step 3

- Create/run your notebook as you usually would

Import the rest of your necessary libraries as you usually would. For this demonstration we will be using the breast c
classification so we will also import that from sklearn.

```
[ ] import numpy as np
    from sklearn.datasets import load_breast_cancer
    from sklearn.model_selection import train_test_split, GridSearchCV
    from sklearn.linear_model import LogisticRegression
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.preprocessing import StandardScaler
    from sklearn.metrics import f1_score, precision_score, recall_score, confusion_matrix
```

```
[ ] # Have a look at your dataset
    cancer = load_breast_cancer()
    print("cancer.keys(): {}".format(cancer.keys()))
    print("Shape of cancer data: {}\n".format(cancer.data.shape))
    print("Sample counts per class:\n{}".format(
          {n: v for n, v in zip(cancer.target_names, np.bincount(cancer.target))}))
    print("\nFeature names:\n{}".format(cancer.feature_names))
```

```
cancer.keys(): dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'filename'])
Shape of cancer data: (569, 30)

Sample counts per class:
{'malignant': 212, 'benign': 357}

Feature names:
['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
```

Step 4

- After your model has run, **save all your metrics** as separate variables and **create dictionaries** for the data you want to log.

Step 5

- Log the necessary items using the commands : **`experiment.log_metrics()`** and **`experiment.log_parameters()`**

Step 6

- Remember to **`experiment.end()`** if you're using a **notebook**

Notebook

Recording your first experiment

Step 1

- **Import comet** at the very top of your notebook

Step 2

- **Link your workspace** using the code from the starter-guide

Step 3

- Create/run your notebook as you usually would

Step 4

- After your model has run, **save all your metrics** as separate variables and **create dictionaries** for the data you want to log.

Step 5

- Log the necessary items using the commands : **`experiment.log_metrics()`** and **`experiment.log_parameters()`**

Step 6

- Remember to **`experiment.end()`** if you're using a **notebook**

```
# Saving each metric to add to a dictionary for logging

f1 = f1_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
```

```
[ ] # Create dictionaries for the data we want to log

params = {"random_state": 7,
          "model_type": "logreg",
          "scaler": "standard scaler",
          "param_grid": str(param_grid),
          "stratify": True
          }
metrics = {"f1": f1,
           "recall": recall,
           "precision": precision
           }
```

```
[ ] # Log our parameters and results
    experiment.log_parameters(params)
    experiment.log_metrics(metrics)
```

Notebook

Recording your first experiment

Step 1

- **Import comet** at the very top of your notebook

Step 2

- **Link your workspace** using the code from the starter-guide

Step 3

- Create/run your notebook as you usually would

Step 4

- After your model has run, **save all your metrics** as separate variables and **create dictionaries** for the data you want to log.

Step 5

- Log the necessary items using the commands : **`experiment.log_metrics()`** and **`experiment.log_parameters()`**

Step 6

- Remember to **`experiment.end()`** if you're using a **notebook!**

You can find additional information on experiments and what data you can record at the link below:
https://www.comet.ml/docs/python-sdk/Experiment/

```
# Log our parameters and results
experiment.log_parameters(params)
experiment.log_metrics(metrics)
```

Notebook

Recording your first experiment

Step 1

- **Import comet** at the very top of your notebook

Step 2

- **Link your workspace** using the code from the starter-guide

Step 3

- Create/run your notebook as you usually would

Step 4

- After your model has run, **save all your metrics** as separate variables and **create dictionaries** for the data you want to log.

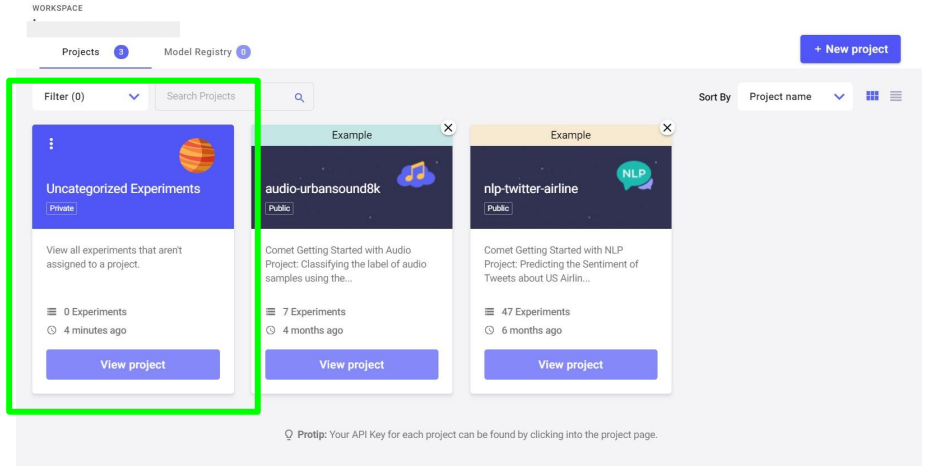
Step 5

- Log the necessary items using the commands : **`experiment.log_metrics()`** and **`experiment.log_parameters()`**

Step 6

- Remember to **`experiment.end()`** if you're using a **notebook!** You can check out your results immediately after by clicking on your project.

```
] experiment.end()
COMET INFO: ----------------------------
COMET INFO: Comet.ml Experiment Summary:
COMET INFO:   Data:
COMET INFO:     url: https://www.comet.ml/jo-moon/general/96b60794fd8747a084b2a1c0cc015a33
COMET INFO:   Metrics [count] (min, max):
COMET INFO:     f1                     : (0.9832402234636872, 0.9832402234636872)
COMET INFO:     precision              : (0.9887640449438202, 0.9887640449438202)
COMET INFO:     recall                 : (0.9777777777777777, 0.9777777777777777)
COMET INFO:     sys.cpu.percent.01 [31] : (0.9, 10.3)
COMET INFO:     sys.cpu.percent.02 [31] : (0.9, 7.5)
COMET INFO:     sys.cpu.percent.avg [31]: (0.95, 8.9)
COMET INFO:     sys.ram.total [31]      : (13653561344.0, 13653561344.0)
COMET INFO:     sys.ram.used [31]       : (592932864.0, 735928320.0)
COMET INFO: ----------------------------
COMET INFO: Uploading stats to Comet before program termination (may take several seconds)
```



Bonus Panel!

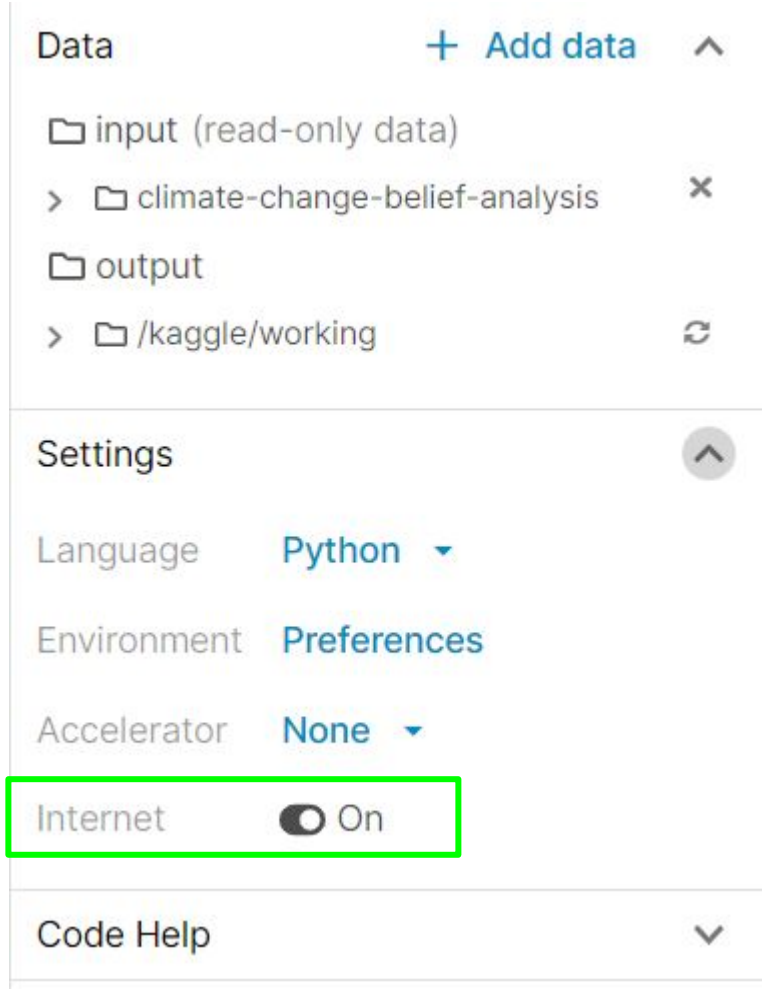
Using Comet in a Kaggle Kernel

You will need to use Comet inside of your Kaggle kernel - This can be done in exactly the same way that we outlined above, Just remember to switch on the internet connectivity in your kernel to install the package. You can do this as illustrated on the right.

You may be prompted to verify you account, and will then be able to proceed :)

```
!pip install comet_ml
```

```
Collecting comet_ml
  Downloading comet_ml-3.1.8-py2.py3-none-any.whl (200 kB)
     |████████████████████████████████| 200 kB 4.8 MB/s eta 0:00:01
```



Conclusion

More info at comet.ml

If you would like to dig deeper, go and explore the tutorials on the comet website. They are robust, and insightful, and will give you a lot to play with.

This is not the only method of model version control- there are many amazing new technologies springing up specifically for Data Science applications, whereas in the past we were adapting tools created for software engineers. So go and explore, and do amazing things!