** **

**A**

**PROJECT REPORT**

**On**

**Image classification using CNN & Keras**

**Submitted by:**

**Palakh jyoti Kakoti (ET16BTHEE018)**

**Partha pratim Saikia(ET16BTHEE016)**

**Mrityunjoy Bora (ET16BTHEE009)**

**Rahul Dibruwal (ET16BTHEE004)**

**Madhurjya kamal Borah(ET16BTHEE002)**

**School of Engineering and Technology**

**Department Of Electrical Engineering**

**The Assam Kaziranga University**

CONTENTS:

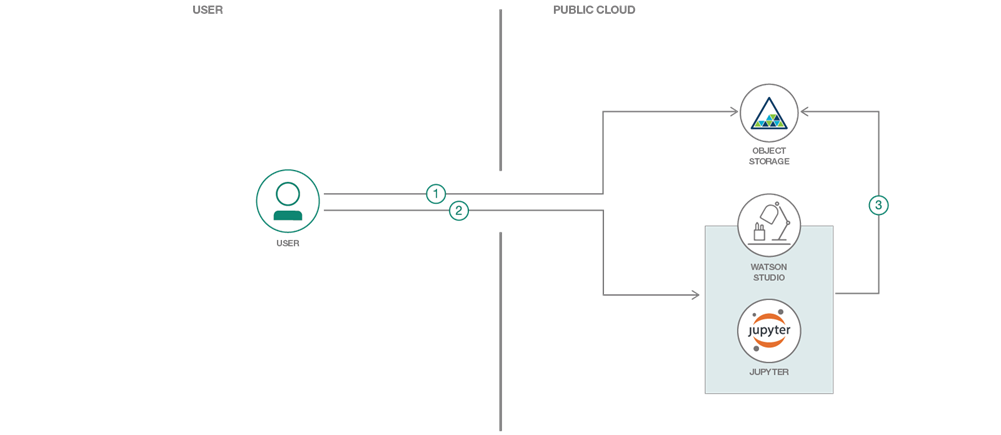
* **INTRODUCTION**
* **FLOW DIAGRAM**
* **INCLUDED COMPONENTS DETAILS**
* **DATASET DETAILS**
* **STEPS**
* **FUTURE SCOPE**
* **CONCLUSION**

INTRODUCTION:

1. This code pattern demonstrates how images, specifically document images like id cards, application forms, cheque leaf etc.
2. It can be classified using Convolutional Neural Network (CNN)

* Many organizations process application forms, such as loan applications, from its customers. Along with the application forms, customers provide supporting documents needed for processing applications. Some of these supporting documents could be an identity proof document, address proof document. Generally, application forms, along with supporting documents, are scanned and captured into the organization's systems for further processing of applications. When the system is fed with a set of all these scanned documents, it needs to identify the form document so that it can process it further. This code pattern shows how to classify images and identify application form document among them

FLOW DIAGRAM:



1. User uploads test images to IBM Cloud Storage.
2. The user adds Cloud Object Storage credentials, in Jupyter notebook, for the test images uploaded. The user runs the notebook.

3.Notebook pulls images from Cloud Object Storage and classifies the images using a trained machine learning model.

INCLUDED COMPONENT DETAILS:

**1.IBM Watson Studio**: Analyze data using RStudio, Jupyter, and Python in a configured, collaborative environment that includes IBM value-adds, such as managed Spark.

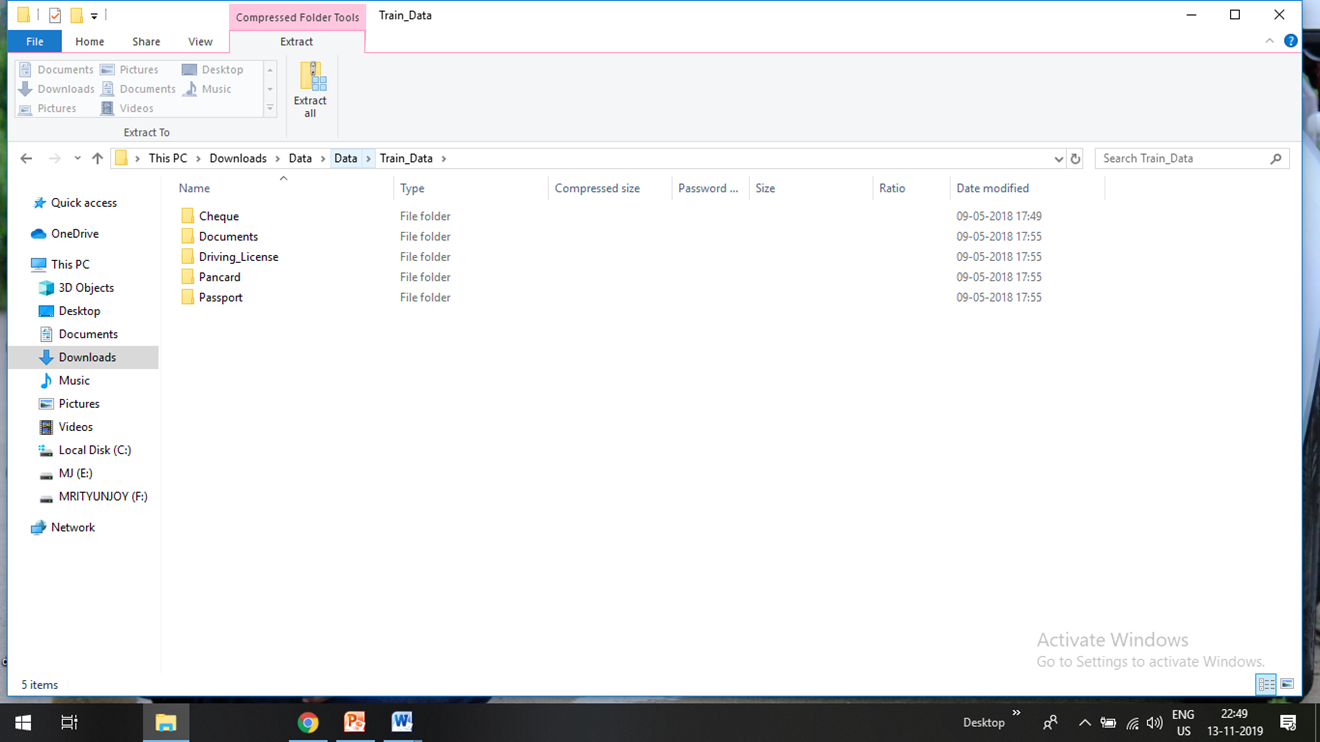
**2.IBM Cloud Object Storage**: An IBM Cloud service that provides an unstructured cloud data store to build and deliver cost-effective apps and services with high reliability and fast speed to market.

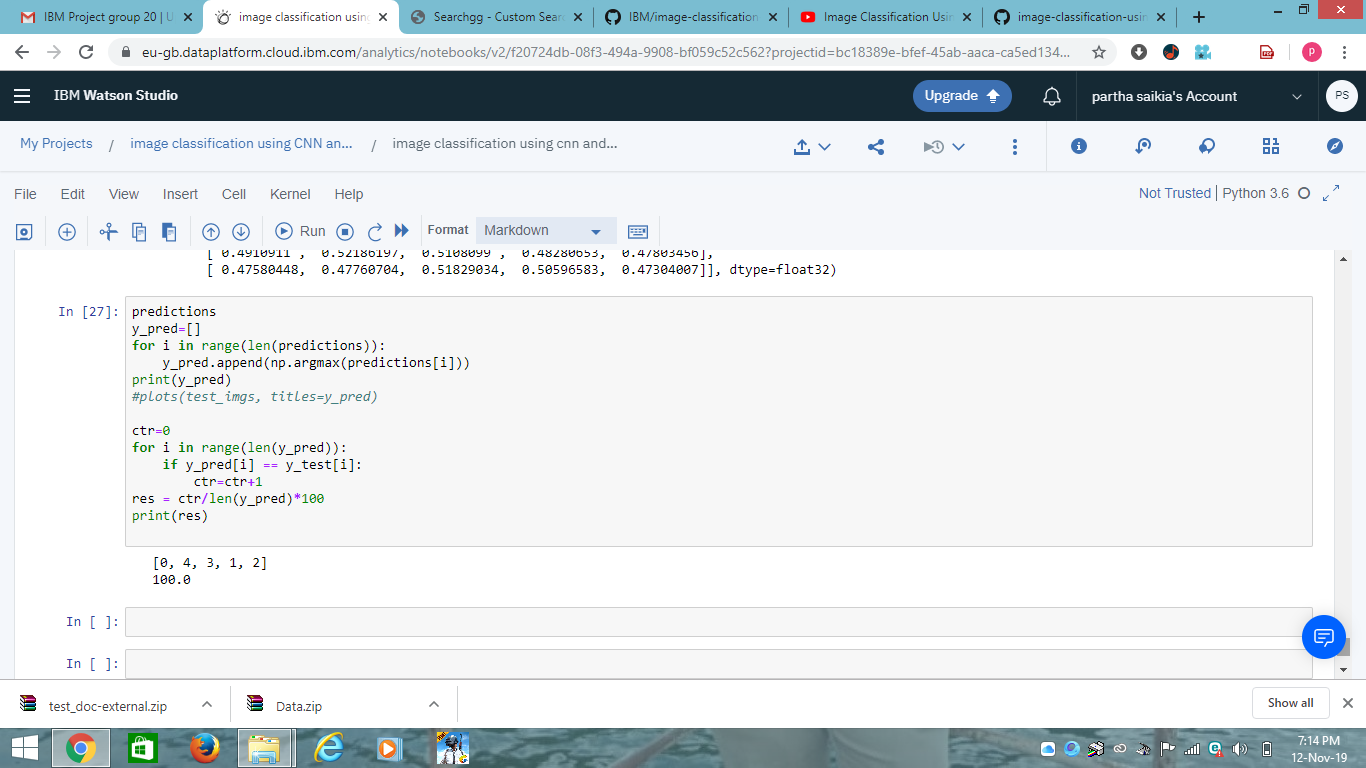
INCLUDED COMPONENT DETAILS:

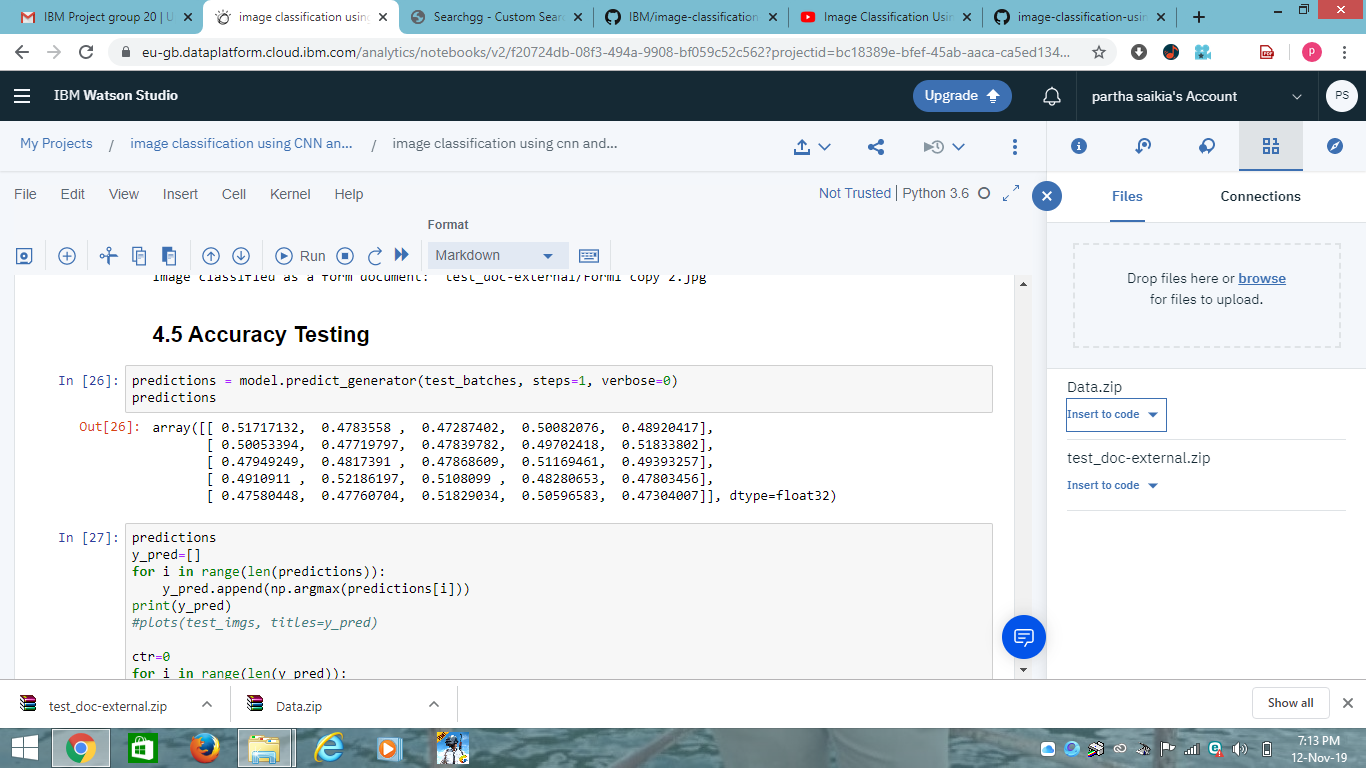
* **Featured technologies:**

1. **Jupyter Notebooks**: An open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and explanatory text.
2. **Artificial Intelligence**: Intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans.
3. **Machine Learning:** Uses statistical techniques to give computer systems the ability to "learn" with data.
4. **Python**: An interpreted high-level programming language for general-purpose programming.

DATASET DETAILS:

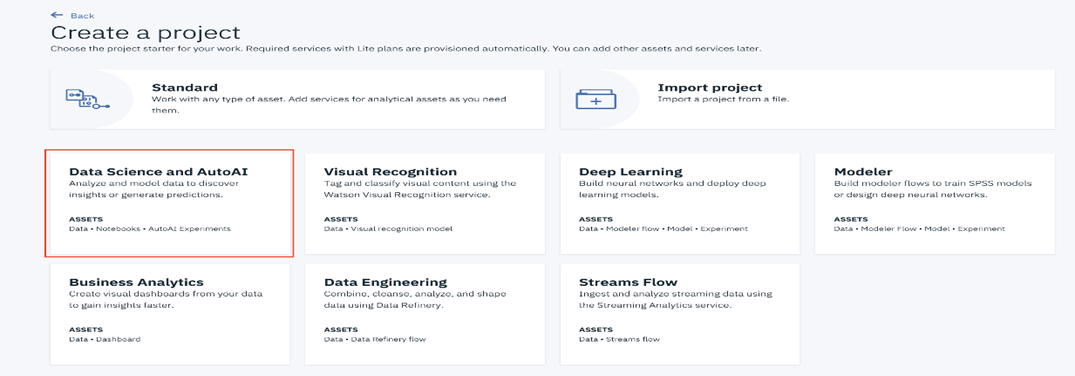






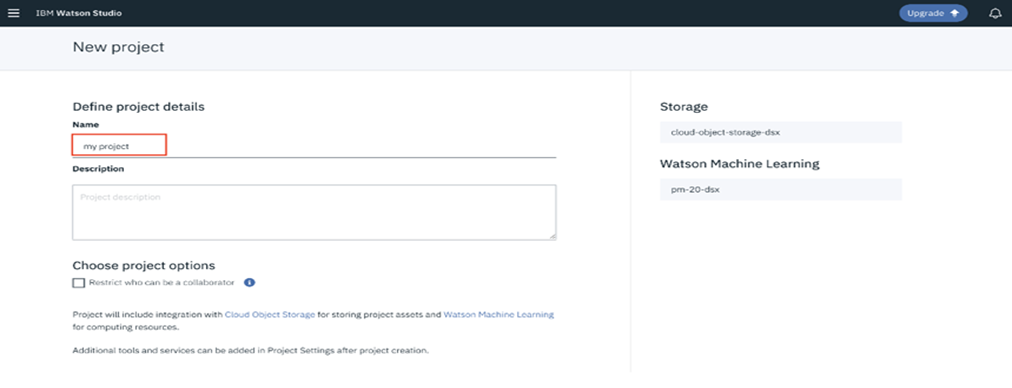
STEPS:

* Create a new watson studio project.
* Create a notebook.
* Add data files.
* Run the notebook.
* **Create a new Watson Studio project**:
* Log into IBM's [Watson Studio](https://dataplatform.cloud.ibm.com/). Once in, you'll land on the dashboard.
* Create a new project by clicking + New project and choosing Data Science:

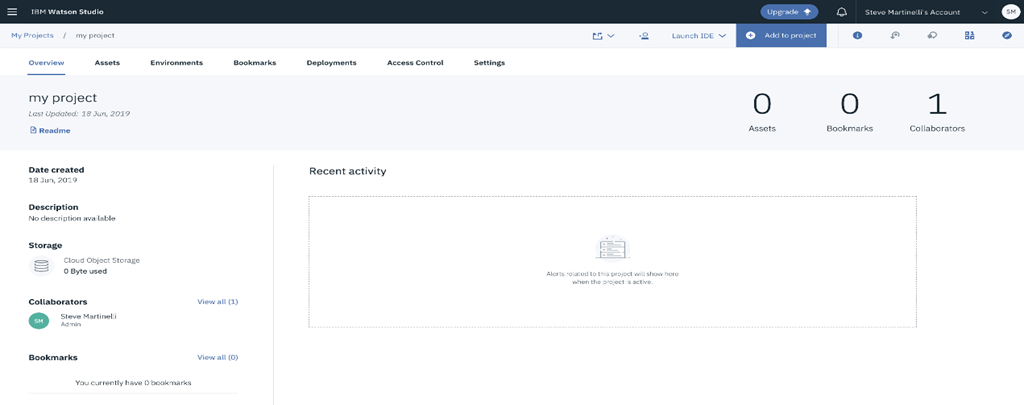


* Enter a name for the project name and click Create:

**NOTE**: By creating a project in Watson Studio a free tier Object Storage service and Watson Machine Learning service will be created in your IBM Cloud account. Select the Free storage type to avoid fees.



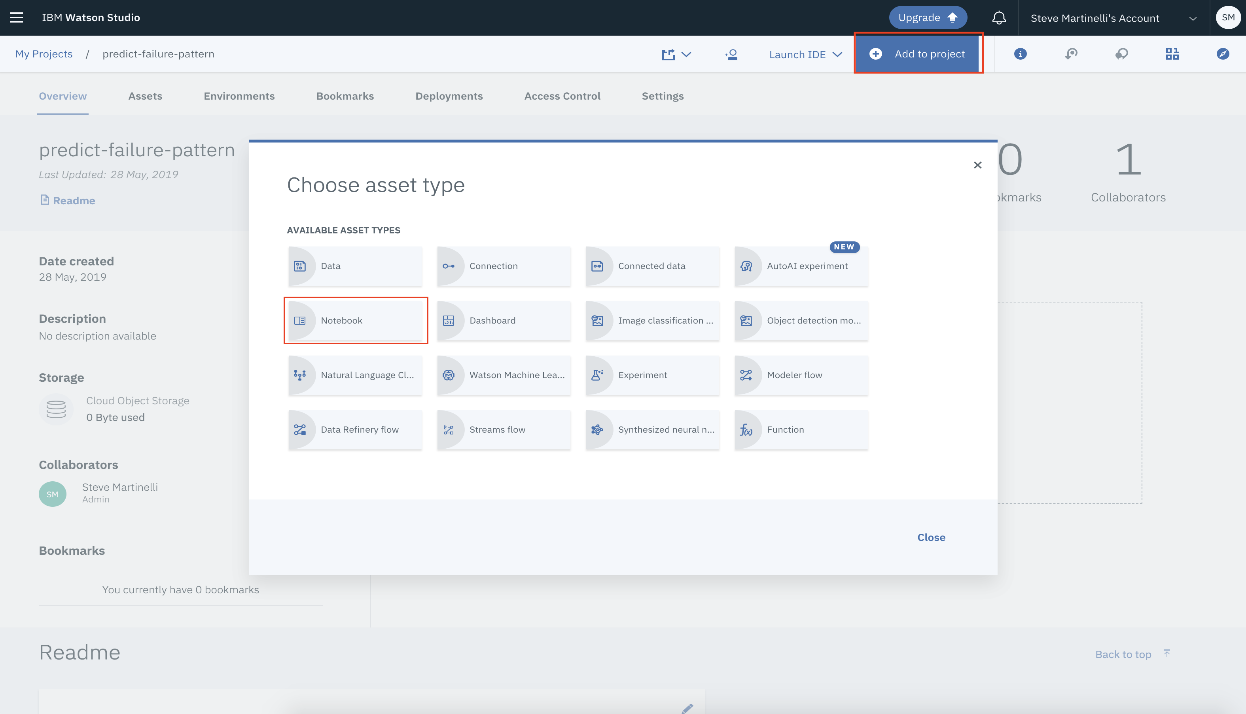
* Upon a successful project creation, you are taken to a dashboard view of your project. Take note of the Assets and Settings tabs, we'll be using them to associate our project with any external assets (datasets and notebooks) and any IBM cloud services.



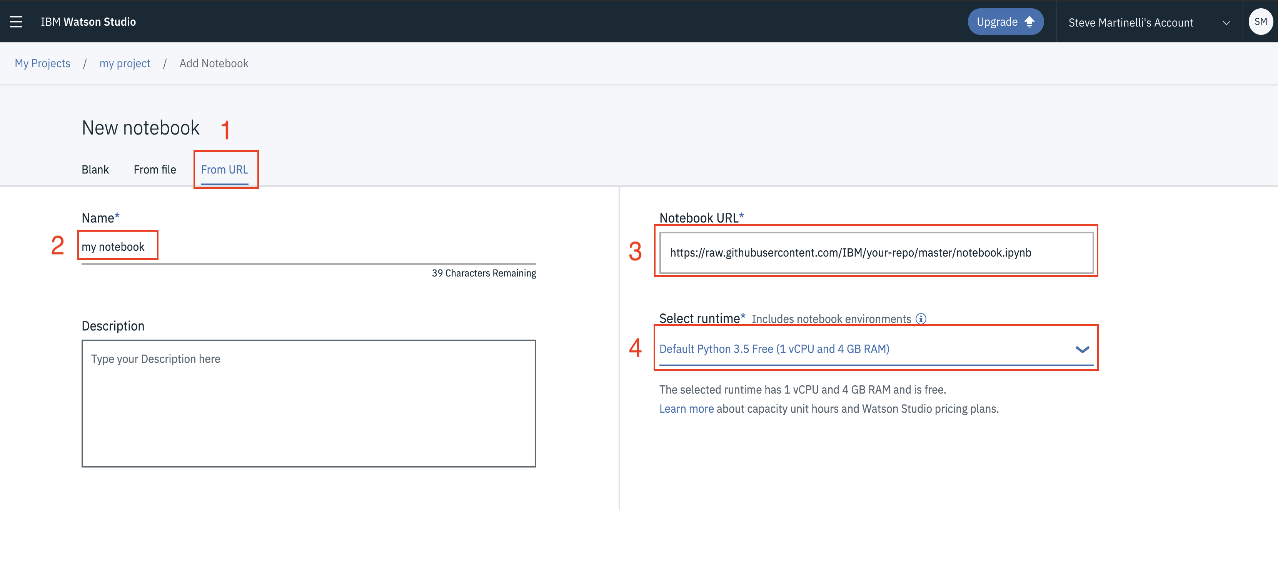
## **Create the Notebook:**

The notebook we'll be using can be viewed in [notebooks/employee-attrition.ipynb](https://github.com/IBM/employee-attrition-aif360/blob/master/notebooks/employee-attrition.ipynb), and a completed version can be found in [examples/employee-attrition.ipynb](https://github.com/IBM/employee-attrition-aif360/blob/master/examples/employee-attrition.ipynb).

* From the new project Overview panel, click + Add to project on the top right and choose the Notebook asset type.



* Fill in the following information:
* Select the From URL tab. [1]
* Enter a Name for the notebook and optionally a description. [2]
* Under Notebook URL provide the following url: <https://github.com/IBM/employee-attrition-aif360/blob/master/notebooks/employee-attrition.ipynb> [3]
* For Runtime select the Python 3.5 option. [4]



* **Add Data Files**.
* **Create grithub account.**

**Run the Notebook:**

When running the notebook, you will come to the cell that requires you to enter your Watson Machine Learning instance credentials. These will be required to complete the notebook. Refer to step #1 above for more details.

When a notebook is executed, what is actually happening is that each code cell in the notebook is executed, in order, from top to bottom.

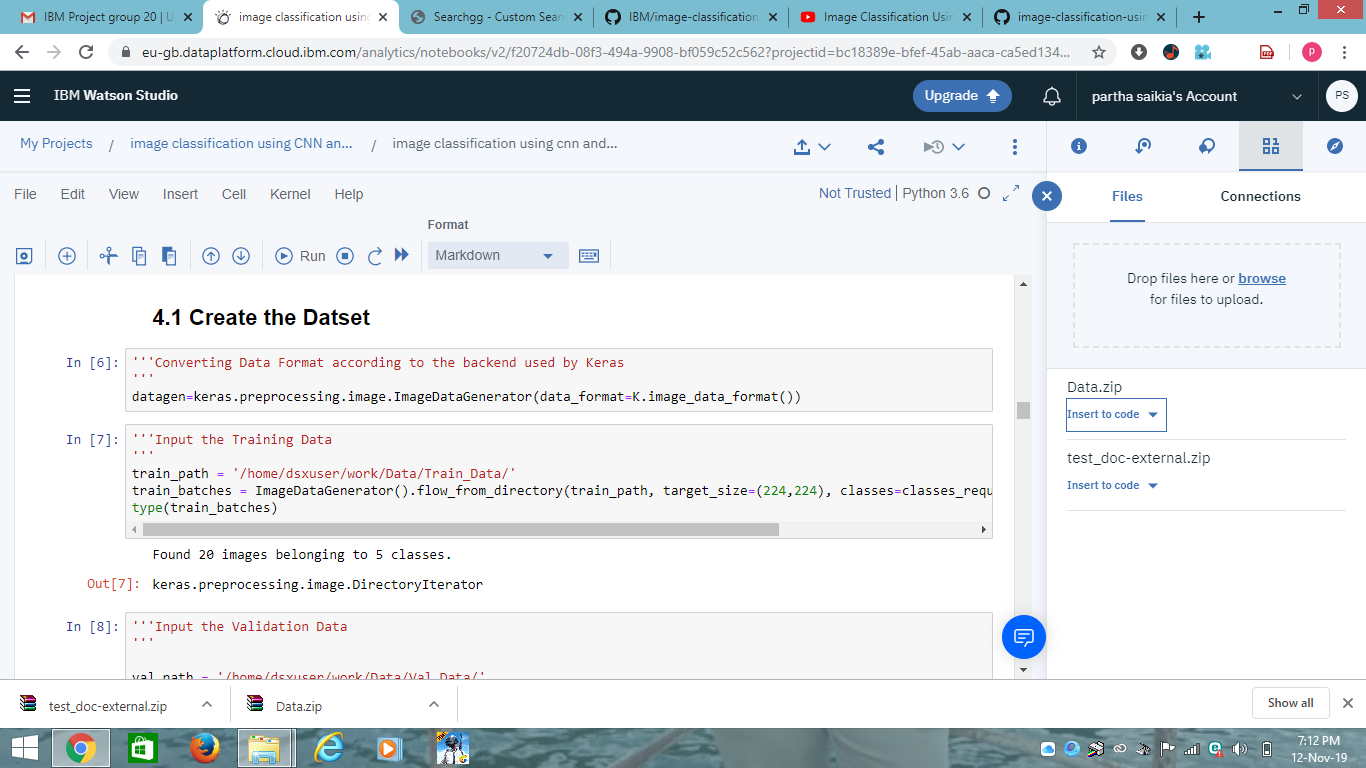
Each code cell is selectable and is preceded by a tag in the left margin. The tag format is In [x]: Depending on the state of the notebook, the x can be:

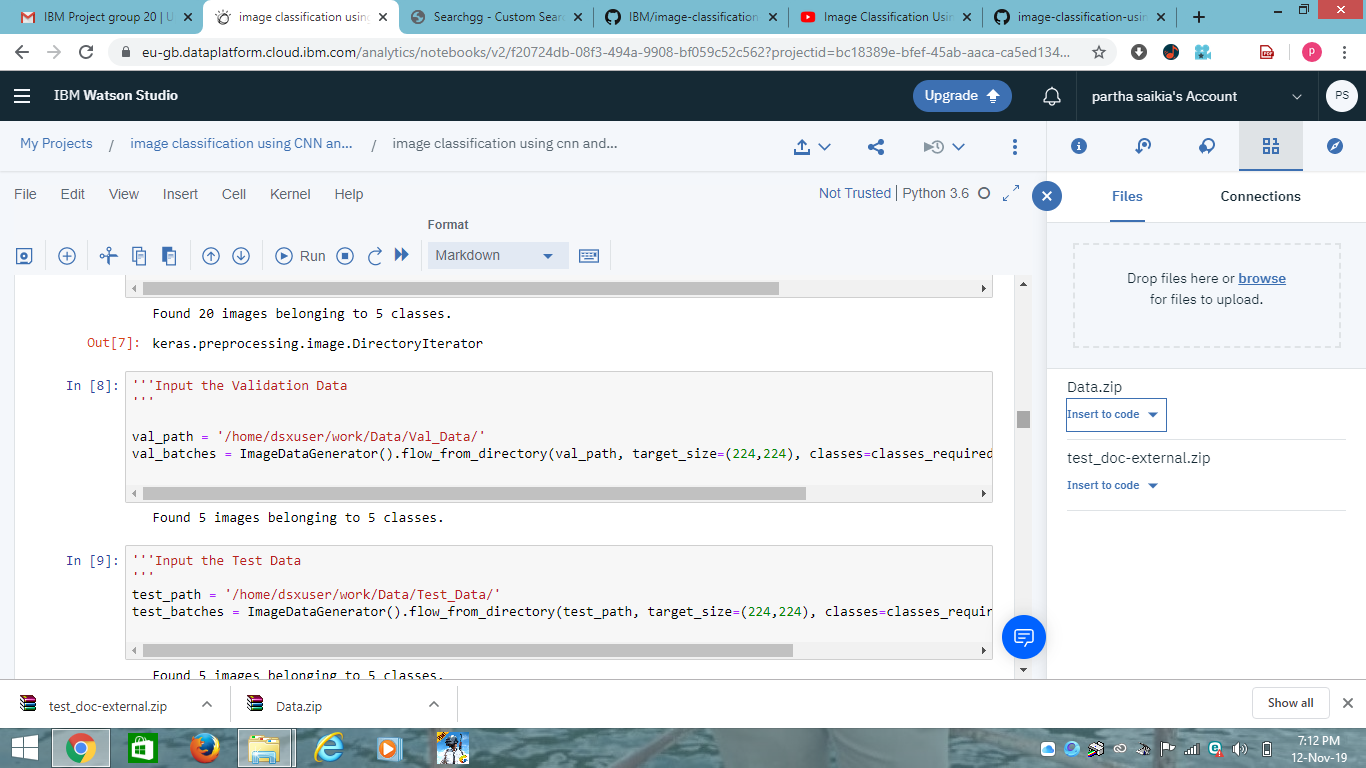
* A blank, this indicates that the cell has never been executed.
* A number, this number represents the relative order this code step was executed.
* A \*, this indicates that the cell is currently executing.

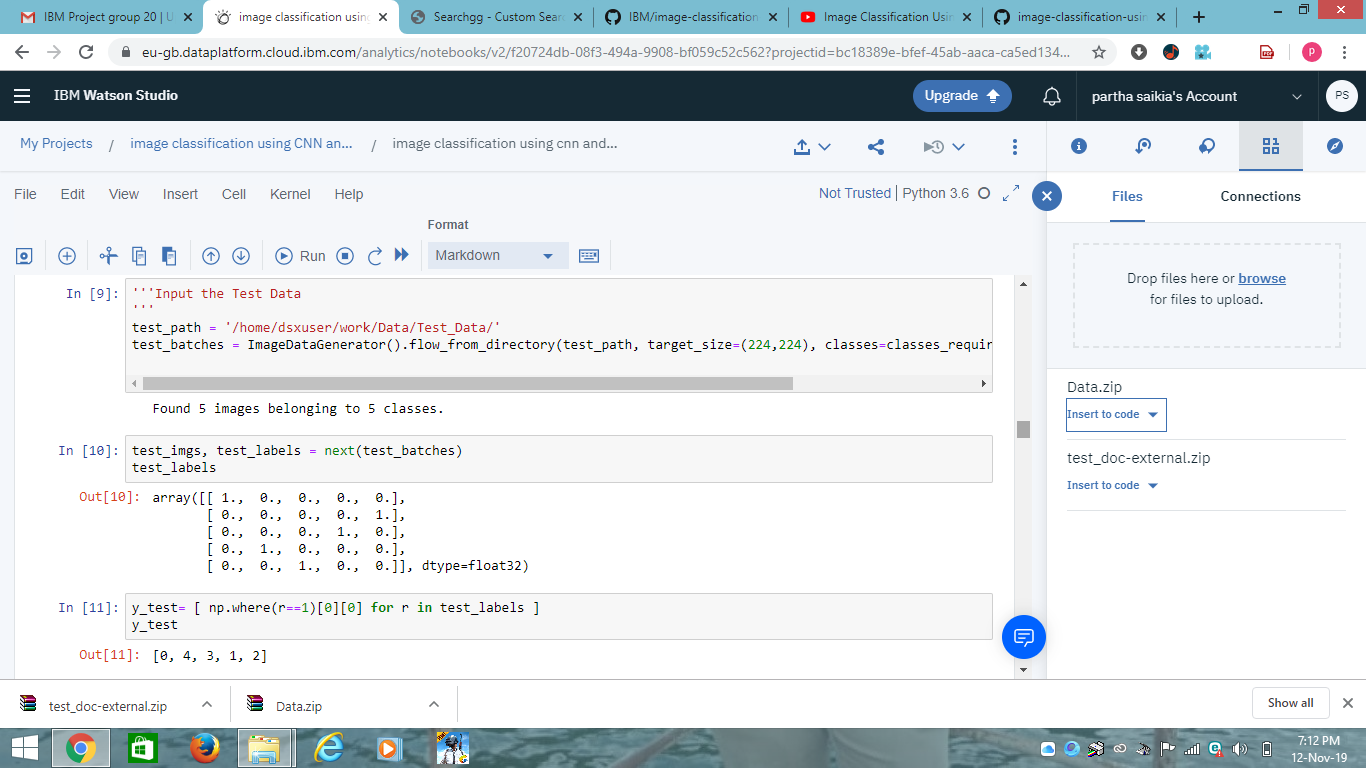
There are several ways to execute the code cells in your notebook:

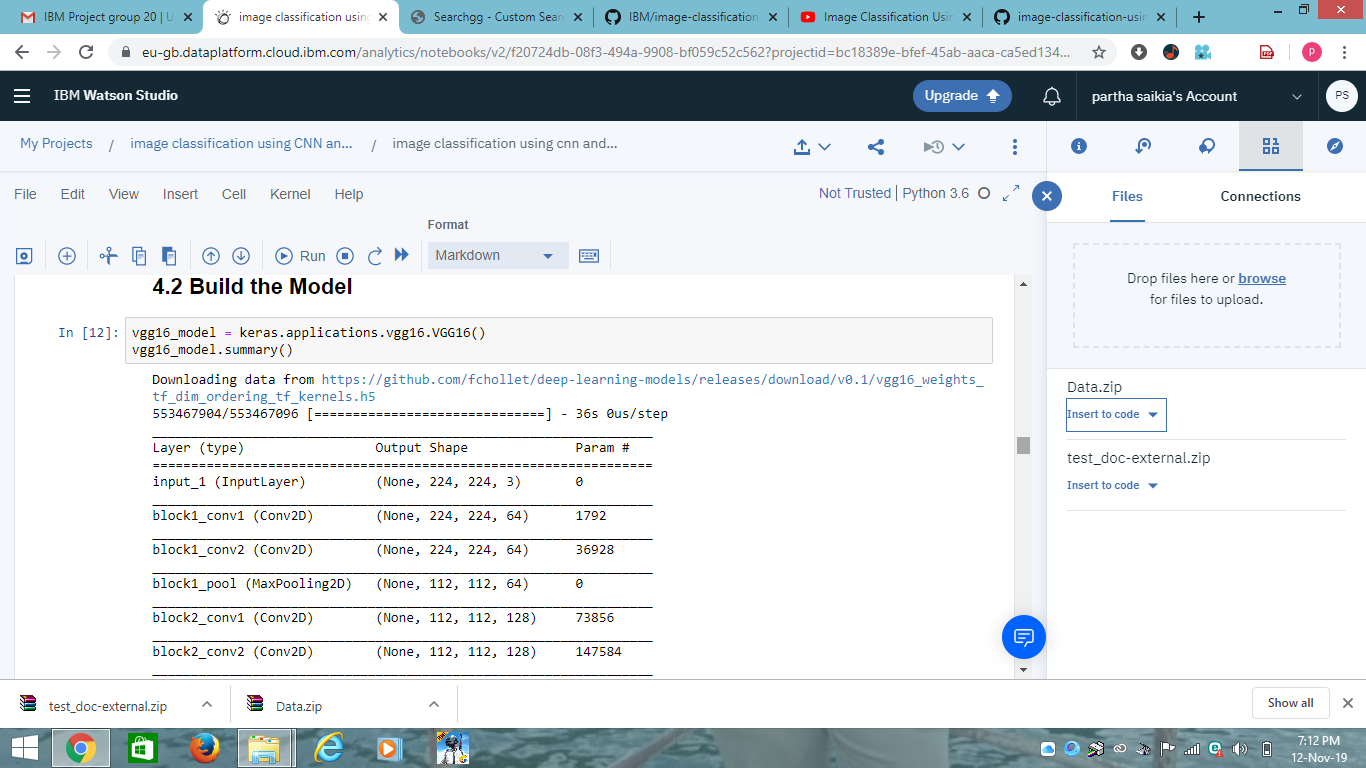
* One cell at a time.
  + Select the cell, and then press the Play button in the toolbar.
* Batch mode, in sequential order.
  + From the Cell menu bar, there are several options available. For example, you can Run All cells in your notebook, or you can Run All Below, that will start executing from the first cell under the currently selected cell, and then continue executing all cells that follow.
* At a scheduled time.
  + Press the Schedule button located in the top right section of your notebook panel. Here you can schedule your notebook to be executed once at some future time, or repeatedly at your specified interval.

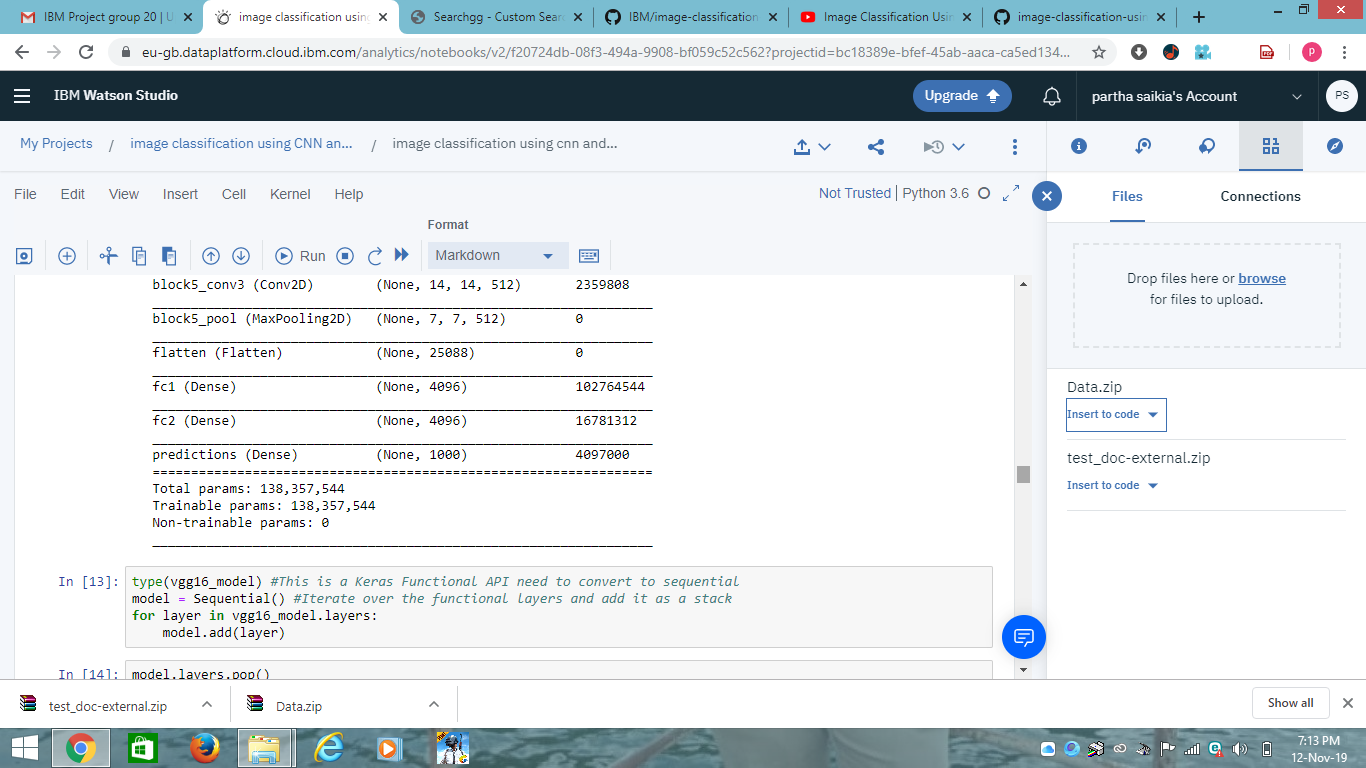
OUTPUT:

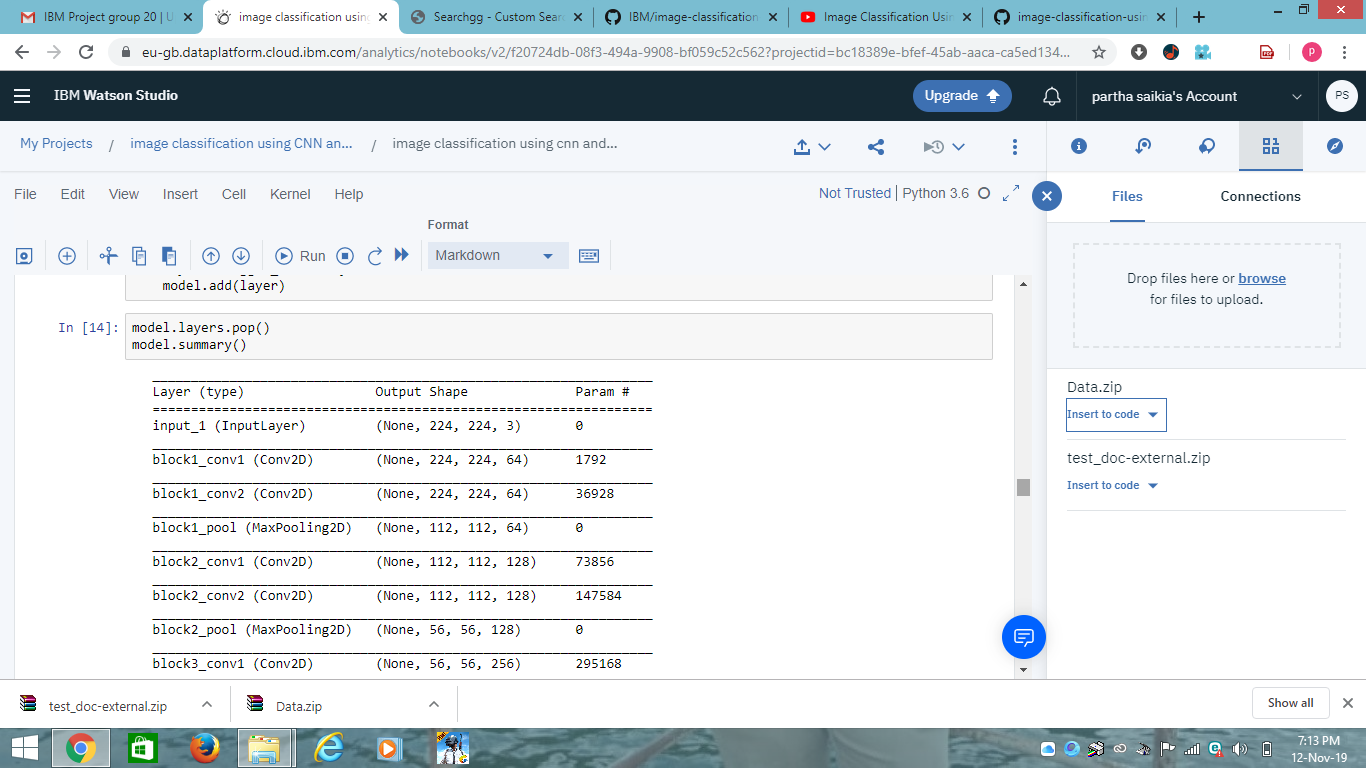


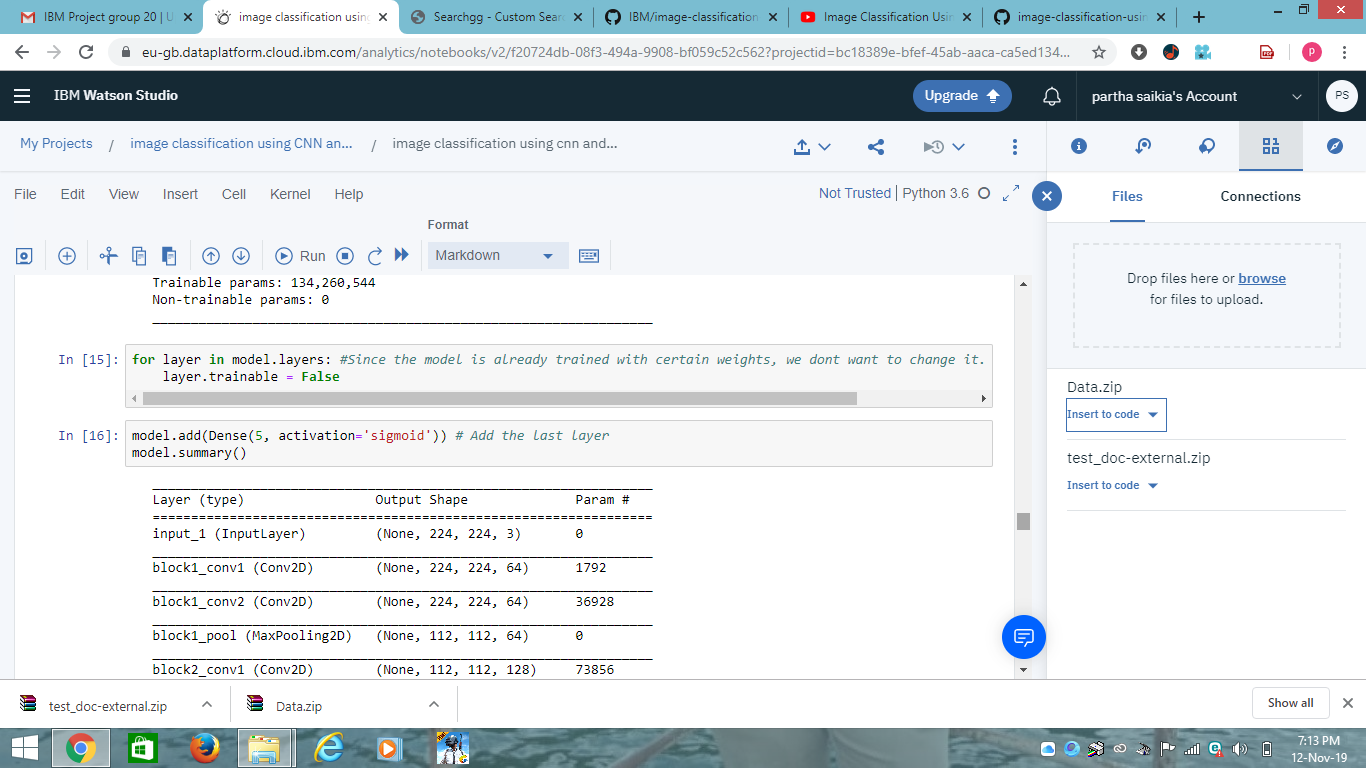


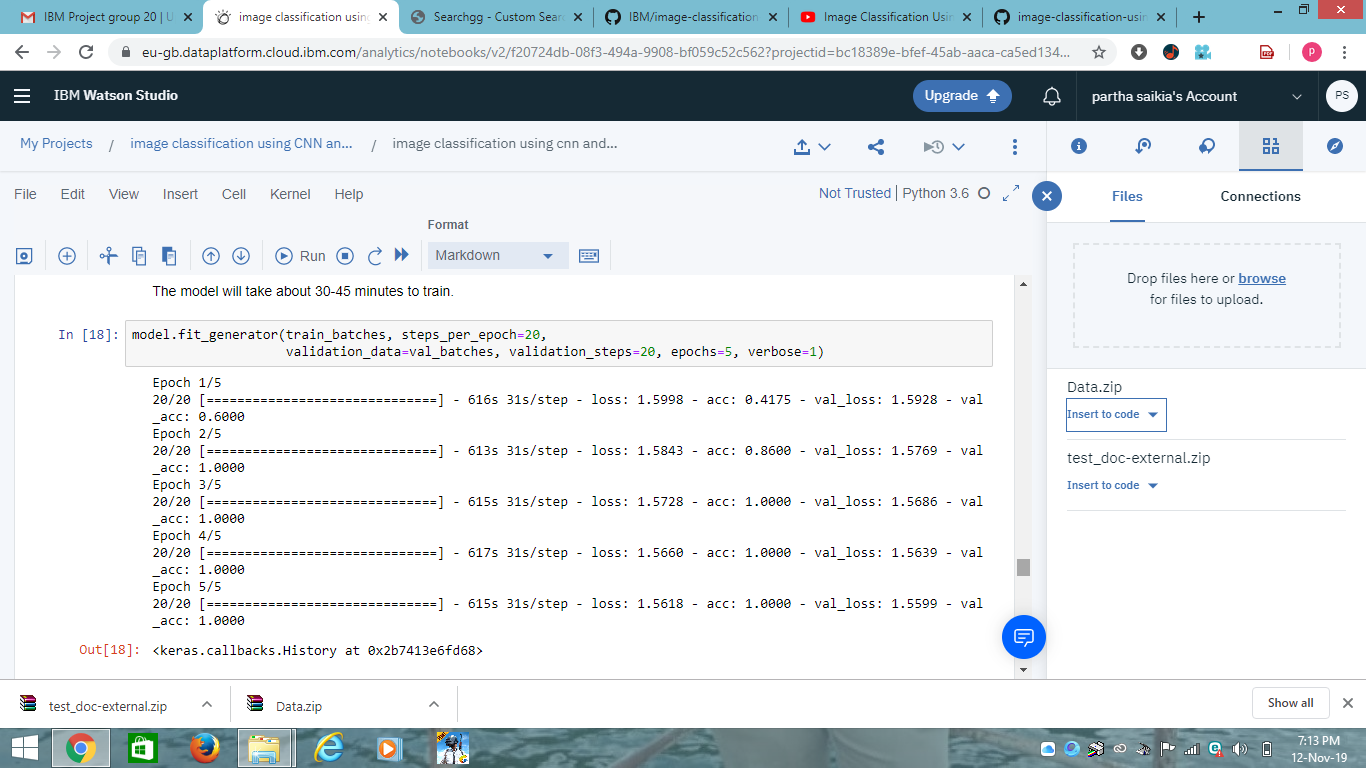


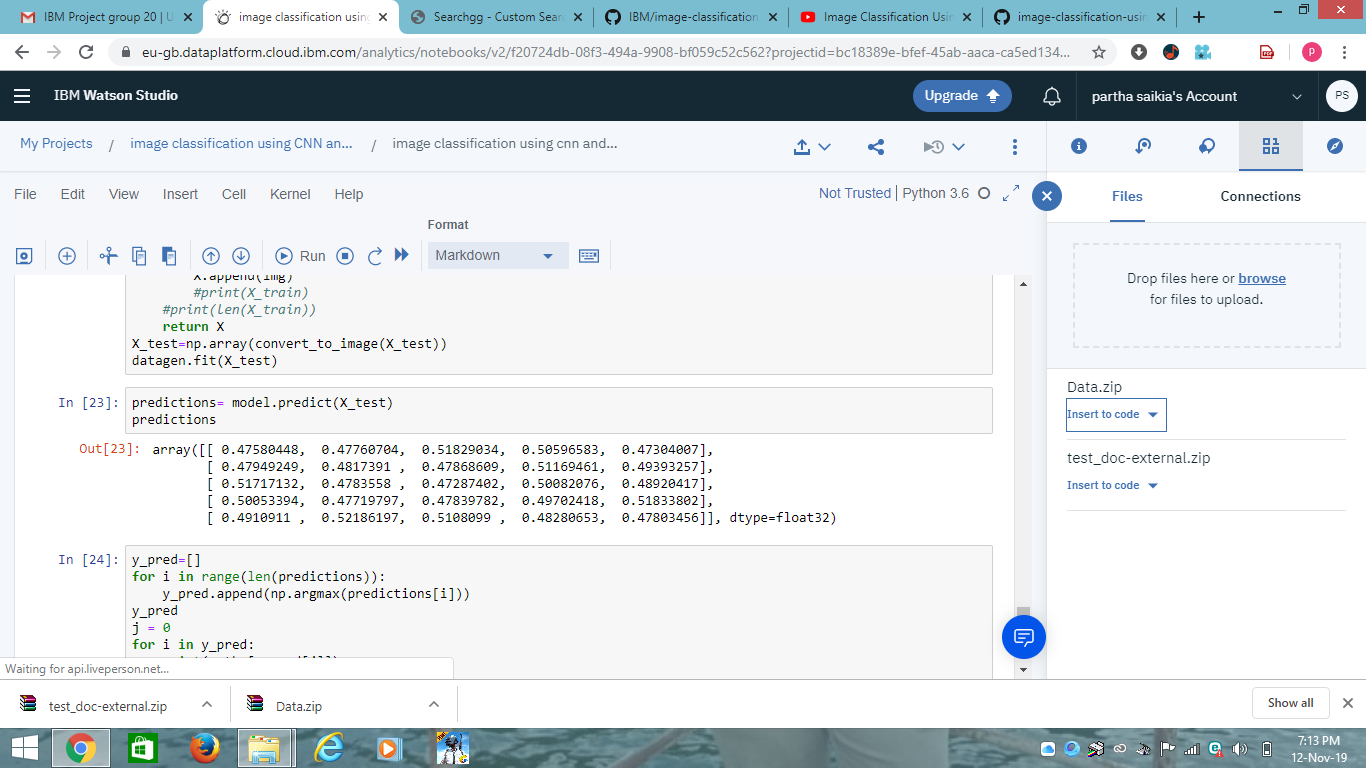


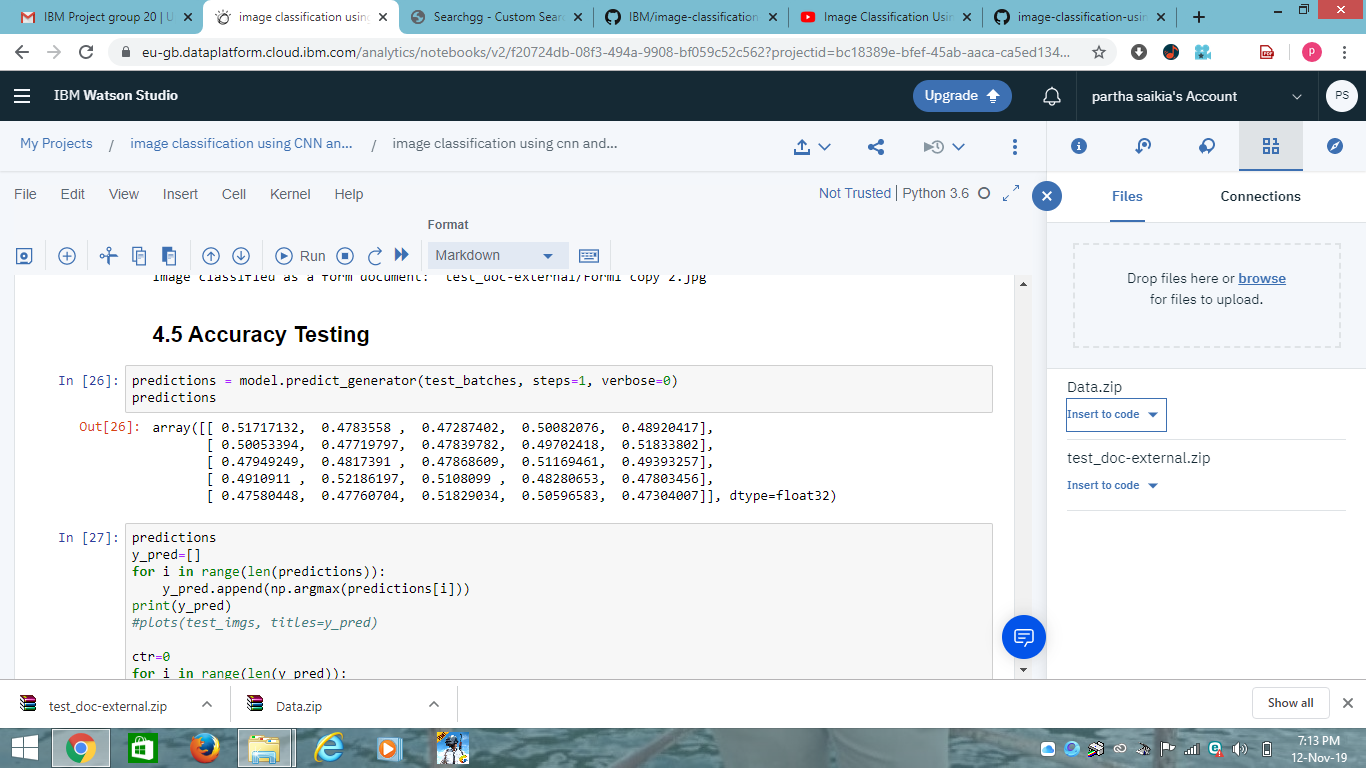


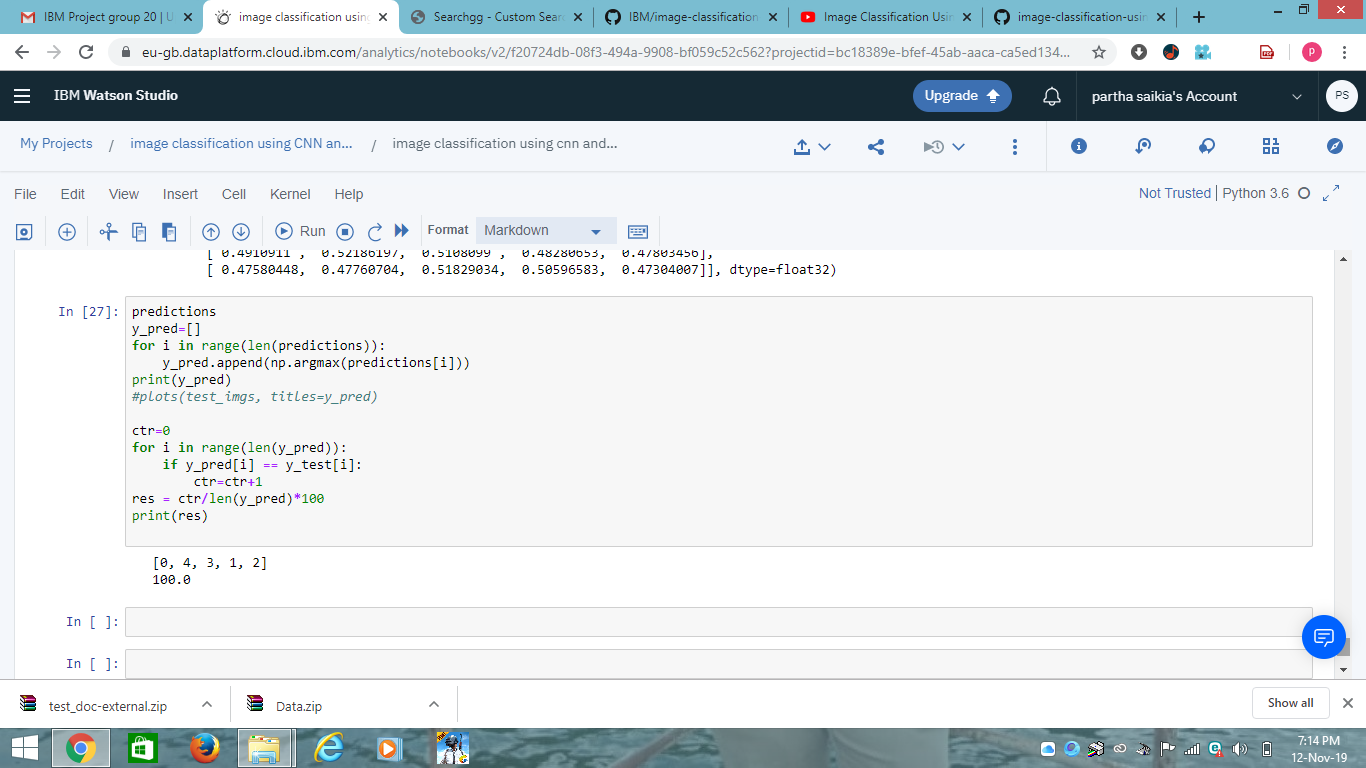












**FUTURE SCOPE:**

* Facial key point detection.
* Analysis of satellite image for disaster detection
* Real Time Criminal Detection through video analysis.

**CONCLUSION:**

* CNN is mostly used for classification of Spatial data.
* CNN does not store or remember past sequence of patterns that can be used for future predictions.
* The Proposed architecture require more research and exploration to solve problems related to speech recognition, language modelling, language translation , and audio/video processing, where the past context and sequences are important for the future predictions.