**How to Upload, Process, and Predict Images to Custom Vision Services (CVS)**

**Objective**: Even though it is possible to upload images using CVS interface, it is cumbersome to tag each image with appropriate tags. The goal here is to upload all images with appropriate tags, conduct training process, and use testing dataset to produce prediction.

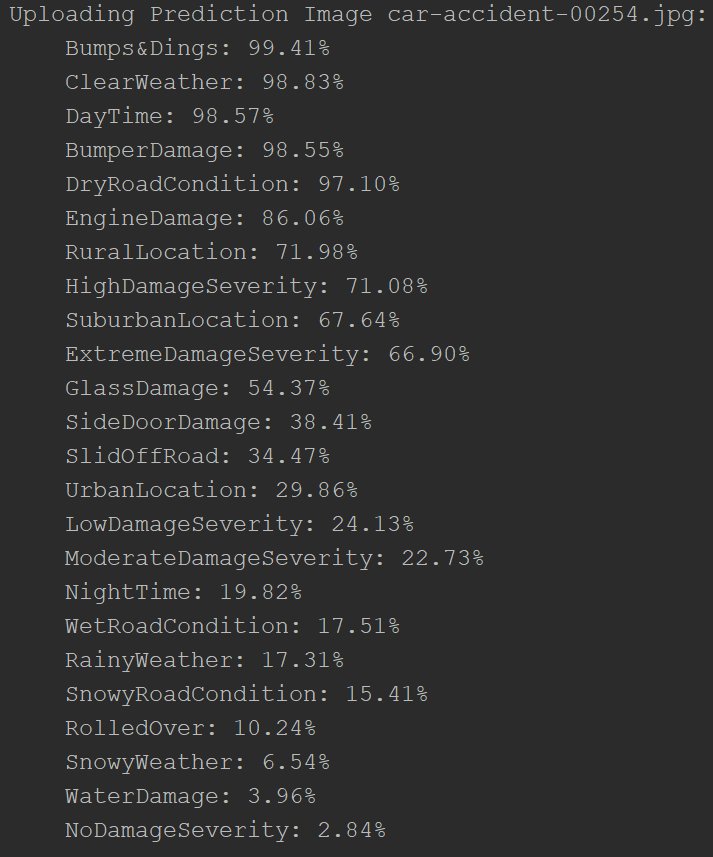
**Location**: Avanade – AI Incubator – FNOL – Image\_Classification – python\_scripts – process\_data.py

**Steps**:

1. Provide all requirements and directory paths

* Project\_id
  + Project\_id can be found in your Custom Vision setting page.
* Training\_key
  + Training\_key can also be found in your Custom Vision setting page
* Prediction\_key
  + Prediction\_key can also be found in your CVS setting page
* Base\_image\_url
  + All images are currently stored in blob storage
* Local\_csv\_location
  + Provide CarDamageLabels.csv file’s directory path. The file should be in “AI Incubator – FNOL\Image\_Classification\CarDamageLabels.csv

1. Run Script
   * When running this script, the script will upload images and create an iteration only when there is no iteration in your project. When there is an iteration, the script skips uploading images and creating an iteration since they already exist, and it starts with uploading testing images to conduct prediction.
   * When there is an iteration, but if user wants to start from scratch, please use “deleteall.py” script to delete all images and iteration. User can upload images and create iteration upon deletion.
   * When there is NO iteration or images, the script will upload images based on the names in the CarDamageLabels.csv. The images being uploaded will be printed in the console screen and uploading takes roughly a few minutes to finish.
   * Upon completing uploading images with appropriate tags to CVS, the scripts begin the training process. The process will be displayed in console, and a message will be displayed once the training process is over.
   * Lastly, the scripts uploads the images and CVS returns prediction percentages for each image.



These data for each image is stored in a dataFrame and gets outputted as an excel csv file at the end.

1. Output – this script output three csv files in the same folder as where script runs
   * **Training\_car.csv** – training dataset, which is 80% of original dataset
   * **Testing\_car.csv** – testing dataset, which is rest of 20% of original dataset
   * **Prediction\_result.csv** – it contains prediction percentage for each testing image