Challenge 1 - Gearing up for Machine Learning

Background

AdventureWorks, a major outdoor and climbing gear retailer, wants to understand customer behavior by learning more about the gear that consumers wear. As a joint project with their Data Science team, they have provided data in the form of product catalog images for your team to get started. The team is tasked with trying several Machine Learning approaches to categorize gear present in the catalog data, a first step towards their goal.

AdventureWorks would, eventually, like to use this knowledge to classify gear in images from Mt. Rainier, a popular climbing destination for mountaineers from around the world, and other destinations as well. This will give a more realistic picture of current needs of existing and future customers worldwide.

Machine Learning solutions are complex and custom models require time, expertise, data preparation, ongoing maintenance and deployment. As a first approach, pre-built solutions offer a way to create a ready-to-use solution quickly before moving on to the work it takes to build custom models.

Prerequisites

* Team has a setup for sharing code and working in Jupyter
* Team has access to the Gear catalog dataset (link and instructions provided in Challenge 0, Data Downloads)
* Team has at least 1 Custom Vision Account. Create [Here](https://customvision.ai/" \t "_blank)

Challenge

Use the team setup and expertise to do the following tasks.

Custom Vision Service is a tool for building custom image classifiers. It takes advantage of Transfer Learning which is when an existing Machine Learning model can be utilized to predict similar classes. This requires less data than training from scratch.

Your challenge is to create a classification model to predict whether an image is a **hardshell jacket** or an **insulated jacket** using a portion of the jacket data in the gear catalog images.

When the model is trained, call the prediction endpoint using Python and a Jupyter Notebook to predict the class of an image not used in training. This can be an image from the catalog data that was not uploaded or an image found online.

Note: Custom Vision has an easy to use User Interface for uploading images, tagging them with their class (e.g.Â insulated\_jacket) and training the model.

Success Criteria

* Each team member can call the team’s Custom Vision prediction endpoint from a Jupyter Notebook to predict the class of a jacket image not used in training and show a successful response

References

**Read me first**

* Machine Learning Demystified [Video](https://youtu.be/k-K3g4FKS_c" \t "_blank)
* Definitions of some common Machine Learning terms [Ref](https://docs.microsoft.com/azure/machine-learning/studio/what-is-machine-learning" \l "key-machine-learning-terms-and-concepts?wt.mc_id=OH-ML-ComputerVision" \t "_blank)
* Classification description [Ref](https://docs.microsoft.com/azure/machine-learning/studio/data-science-for-beginners-the-5-questions-data-science-answers?wt.mc_id=OH-ML-ComputerVision" \l "question-1-is-this-a-or-b-uses-classification-algorithms" \t "_blank)
* Overview diagram of the Machine Learning process [Docs](https://blogs.msdn.microsoft.com/continuous_learning/2014/11/15/end-to-end-predictive-model-in-azureml-using-linear-regression/)

**Jupyter and Managing Packages**

* jupyter [Ref](https://jupyter.readthedocs.io/en/latest/running.html" \t "_blank)
* On using conda or pip to install Python packages [Ref](https://conda.io/docs/user-guide/tasks/manage-pkgs.html" \t "_blank)
* Jupyter notebook usage [Ref](http://jupyter-notebook.readthedocs.io/en/latest/examples/Notebook/Notebook%20Basics.html" \t "_blank)

**Calling the Prediction API**

* Requests is one of the most popular Python libraries to make API calls and is easy to use [Ref](http://docs.python-requests.org/en/master/" \t "_blank) with an example at [Code Sample](https://github.com/michhar/python-jupyter-notebooks/blob/master/cognitive_services/Computer_Vision_API.ipynb)
* Alternatively, the Custom Vision Service gives an example of calling the service with Python’s urllib library and Python3 - [Prediction 2.0 Docs](https://southcentralus.dev.cognitive.microsoft.com/docs/services/450e4ba4d72542e889d93fd7b8e960de/operations/5a6264bc40d86a0ef8b2c290" \t "_blank)

**Custom Vision Service**

* Custom Vision Service [Ref](https://customvision.ai/" \t "_blank)
* Custom Vision Service Docs [Docs](https://docs.microsoft.com/azure/cognitive-services/custom-vision-service/home?wt.mc_id=OH-ML-ComputerVision" \t "_blank)
* Custom Vision Python SDK (Linux) [Ref](https://docs.microsoft.com/azure/cognitive-services/custom-vision-service/python-tutorial?wt.mc_id=OH-ML-ComputerVision" \t "_blank)