Challenge 3: Introduction to Custom Machine Learning

Background

Challenge 2 set the team up for success by providing quality data. Challenge 3 will begin a journey into custom Machine Learning.

AdventureWorks wants their Data Science team, including your team members, to begin learning how and when to perform custom Machine Learning with powerful, more programmatic APIs. Becoming proficient in Machine Learning takes some time, however, beginning with a high-level API, that is stable and has good documentation, is a great start. Some of the reasons to move on to custom Machine Learning include:

* To explore different algorithms to optimize what works best for the data
* To create a workflow with more control over your process
* To deploy a model to a specific architecture or configuration
* Certain limits in bandwidth or data size for a current service have been exceeded

In this challenge, classical or traditional Machine Learning will be used. There are times when it makes sense to use, or at least begin with, traditional Machine Learning:

* No need for expensive GPU
* Used in ML on Edge
* Simple APIs for fast prototyping
* Is used in production systems
* Algorithm variety

Prerequisites

* Team has a setup for sharing code and working in Jupyter
* Preprocessed Gear image data from Challenge 2
* An installation of the Python package called scikit-learn - check if it is installed and update it to the lastest or simply install (see [Hints](https://openhacks.azurewebsites.net/labs/player/microsoft-open-hack-ai#hints)).

Challenge

One of the most popular and well-established Python ML packages, Scikit-Learn, is often the go-to package for starting out and is not uncommon in production systems. It has a simple, intuitive API (often modeled by other packages) and is a great place to start for learning, implementing and programming traditional ML and basic neural networks in Python.

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

Use a non-parametric classification method (see [References](https://openhacks.azurewebsites.net/labs/player/microsoft-open-hack-ai#references)) to create a model to predict the class of a new gear image, training on the preprocessed 128x128x3 gear data from Challenge 2. The algorithm should be chosen from “off-the-shelf” non-parametric algorithms for classification found in the scikit-learn library.

Split the processed data into a train and test data set in order to train a machine learning model and calculate the accuracy of that model. In order to optimize model quality, the training data set is usually much larger than the test set while still leaving enough data to adequately test the model.

Find more information in the [References](https://openhacks.azurewebsites.net/labs/player/microsoft-open-hack-ai#references) and [Hints](https://openhacks.azurewebsites.net/labs/player/microsoft-open-hack-ai#hints).

Perform the following as a team:

* Split the preprocessed image array data from Challenge 2 into train and test sets
* Choose an algorithm from scikit-learn documentation
* Train the model with the training data from the split
* Predict the class of the following piece of gear with the model: [here](https://shop.epictv.com/sites/default/files/ae42ad29e70ba8ce6b67d3bdb6ab5c6e.jpeg)
* Evaluate the model with a confusion matrix to see how individual classes performed (use test data from the split)
* Output the overall accuracy (use test data from the split)

Success Criteria

* The team will run one code cell in a Jupyter notebook for the coach predicting the class successfully of a piece of gear in the provided URL above.
* The team will run one code cell in a Jupyter notebook for the coach showing the accuracy score on the test data from the split. This score should be above 80%.

References

**Read me first**

* scikit-learn algorithm cheatsheet [Ref](http://scikit-learn.org/stable/index.html)
* jupyter [Ref](https://jupyter.readthedocs.io/en/latest/running.html)
* Non-parametric and parametric algorithm differences [Ref](https://sebastianraschka.com/faq/docs/parametric_vs_nonparametric.html)

**ML and Scikit-Learn**

* scikit-learn Machine Learning guide with vocabulary [Ref](http://scikit-learn.org/stable/tutorial/basic/tutorial.html#introduction)
* scikit-learn Supervised Learning [Ref](http://scikit-learn.org/stable/tutorial/statistical_inference/supervised_learning.html)
* scikit-learn General User Guide [Ref](http://scikit-learn.org/stable/user_guide.html)

Hints

* Install packages with pip install package\_name if it is not installed and pip install --upgrade package\_name if it needs to be updated to the latest version.
* It can help to create an “image reader” function if not already built.