

# Challenge 2 - Warming Up with some Image Classification

## Background

Adventure Works wants to understand customer behavior by learning more about the gear that consumers wear, and you have collected and pre-processed the product catalog images, ready to start building a machine learning solution.

Machine Learning solutions can be complex and require time, expertise, data preparation, ongoing maintenance and deployment. As an initial approach, the data science team at Adventure Works wants to try a using the Microsoft *Custom Vision* cognitive service as a quick and easy way to build a solution that can categorize different kinds of jacket that a customer might want to buy.

## Prerequisites

- An environment for sharing code and working in Jupyter notebooks.
- The resized versions of the **gear** catalog images that you created in the previous challenge.
- A **Custom Vision** cognitive service account. Create one [here](#), signing in using one of the environment logins provided to your team.

## Challenge

The Custom Vision service is a cloud-based tool that you can use to build custom image classifiers.

Your challenge is to:

1. Use the Custom Vision service to create a classification model that can predict whether an image is a **hardshell jacket** or an **insulated jacket**, using a portion of the resized jacket images to train the model.
2. Call the prediction endpoint for your model using Python code in a Jupyter Notebook to predict the class of an image that was not used in training. This can

be an image from the catalog data that was not uploaded or an image found online.

## Hints

- The Custom Vision service has an easy to use user interface for interactively uploading images, tagging them with their class (e.g. *insulated jacket*), and training the model.
- Alternatively, you can use the Python SDK for the Custom Vision service to write code that creates your project, uploads images, and trains your model. This approach is preferred because it provides a more repeatable solution.
- To install the Custom Vision SDK package using the Python installer (`pip`), you can use the `!` prefix in a notebook cell. If you are using the Data Science Virtual Machine (on which there are multiple Python environments), you may need to use the `sys.executable` module to do this as explained in the [DSVM documentation](#).
- When using the Custom Vision SDK, you will need the following information:
  - The endpoint for your service. By default, this is “<https://southcentralus.api.cognitive.microsoft.com>”.
  - The training and prediction keys for your service. You can find these on the **Settings** page for your account on the [Custom Vision portal](#).
- Try to use the same number of images from each class when training the model.

The **References** section below includes some links to helpful resources.

## Success Criteria

Each team member must call the team’s Custom Vision prediction endpoint from a Jupyter Notebook to predict the class of two jacket images (one of each class) that were **not** used in training and show the predicted class tag, like this:



# References

## The Custom Vision Service

- [Custom Vision portal](#)
- [Custom Vision documentation](#)
- [Custom Vision classification tutorial](#)
- [Custom Vision Service SDK for Python reference](#)