A machine learning pipeline is used to help automate machine learning workflows. They operate by enabling a sequence of data to be transformed and correlated together in a model that can be tested and evaluated to achieve an outcome, whether positive or negative.

Machine learning (ML) pipelines consist of several steps to train a model. Machine learning pipelines are iterative as every step is repeated to continuously improve the accuracy of the model and achieve a successful algorithm. To build better machine learning models, and get the most value from them, accessible, scalable and durable storage solutions are imperative, paving the way for on-premises object storage.

Now-a-days Data has become a modern-day currency. Tremendous value and intelligence is being extracted from large, captured datasets (Big data) that has led to actionable insights through today's world. It's not just about storing data any longer, but capturing, preserving, accessing and transforming it to take advantage of its possibilities and the value it can deliver.

1. The main objective of having a proper pipeline for any ML model is to exercise control over it. A well-organised pipeline makes the implementation more flexible. It is like having an exploded view of a computer where you can

pick the faulty pieces and replace it- in our case, replacing a chunk of code.

2. The term ML model refers to the model that is created by the training process.
3. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer to be predicted), and it outputs an ML model that captures these patterns.
4. A model can have many dependencies and to store all the components to make sure all features available both offline and online for deployment, all the information is stored in a central repository.
5. A pipeline consists of a sequence of components which are a compilation of computations. Data is sent through these components and is manipulated with the help of computation.

Pipelines are not one-way flows. They are cyclic in nature and enables iteration to improve the scores of the machine learning algorithms and make the model scalable.

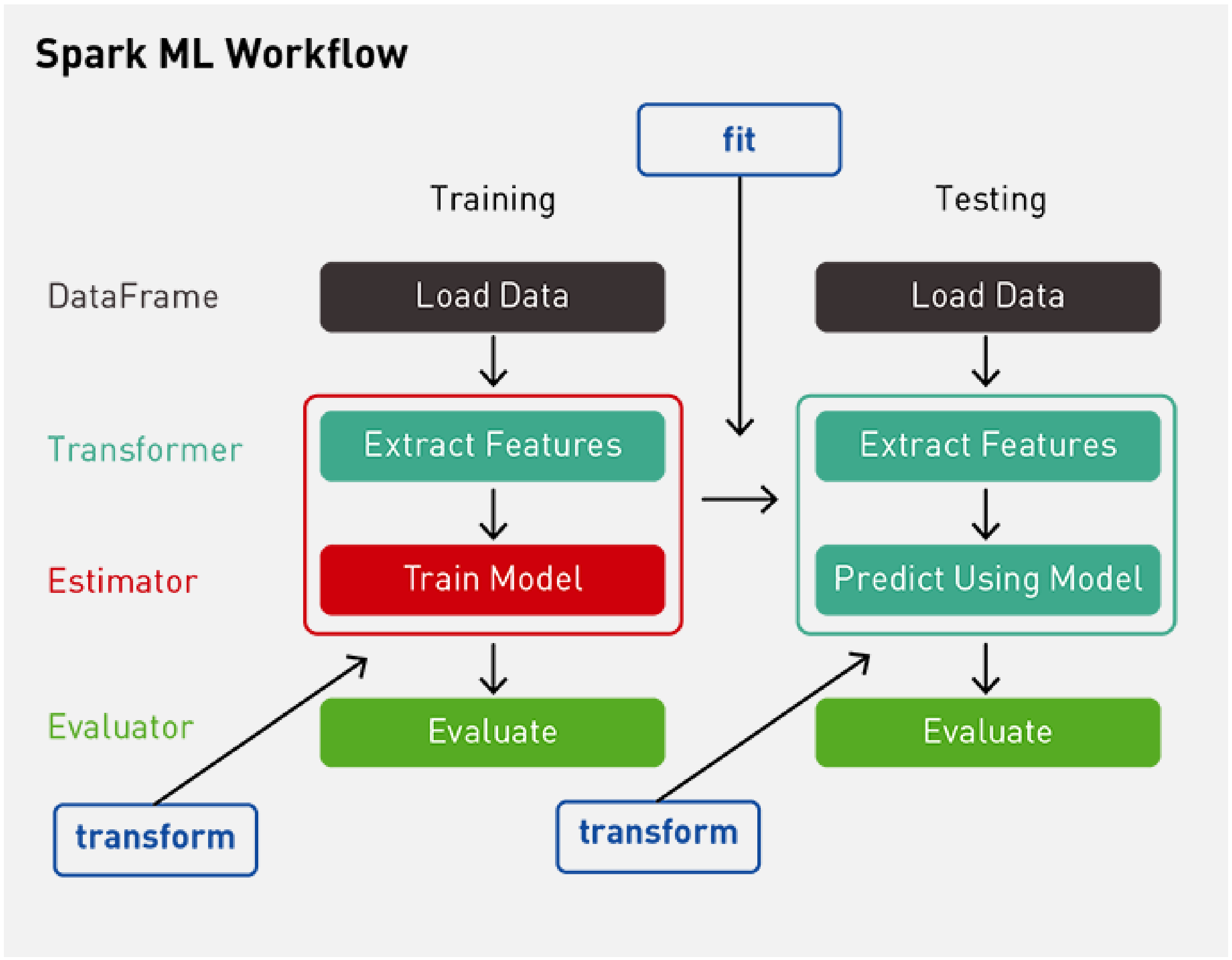
Many of today's ML models are 'trained' neural networks capable of executing a specific task or providing insights derived from 'what happened' to 'what will likely to happen' (predictive analysis). These models are complex and are never completed, but rather, through the repetition of mathematical or computational procedures, are applied to the previous result and improved upon each time to get closer approximations to ‘solving the problem’. Data scientists want more captured data to provide the fuel to train the ML models.

*The goal for ML is simple: “ **Make faster and better predictions**”*

Challenges Associated with ML Pipelines

A typical machine learning pipeline would consist of the following processes:

- Data collection
- Data cleaning
- Feature extraction (labelling and dimensionality reduction)
- Model validation
- Visualisation



Google image

Data collection and cleaning are the primary tasks of any machine learning engineer who wants to make meaning out of data. But getting data and especially getting the right data is an uphill task in itself.

Data quality and its accessibility are two main challenges one will come across in the initial stages of building a pipeline.

The captured data should be pulled and put together and the benefits of collection should outweigh the costs of collection and analysis.

For this purpose, a data lake is recommended for every organisation. A data lake is a centralised repository that allows the user to store both structured and unstructured data at any scale. It also enables ad-hoc analysis by applying schemas to read, not write. In this way, the user can apply multiple analytics and processing frameworks to the same data.

Since every case has its own bargain for the amount of data, usually in an unsupervised setting, things can go out of hand if the quantity of data available for training is less.

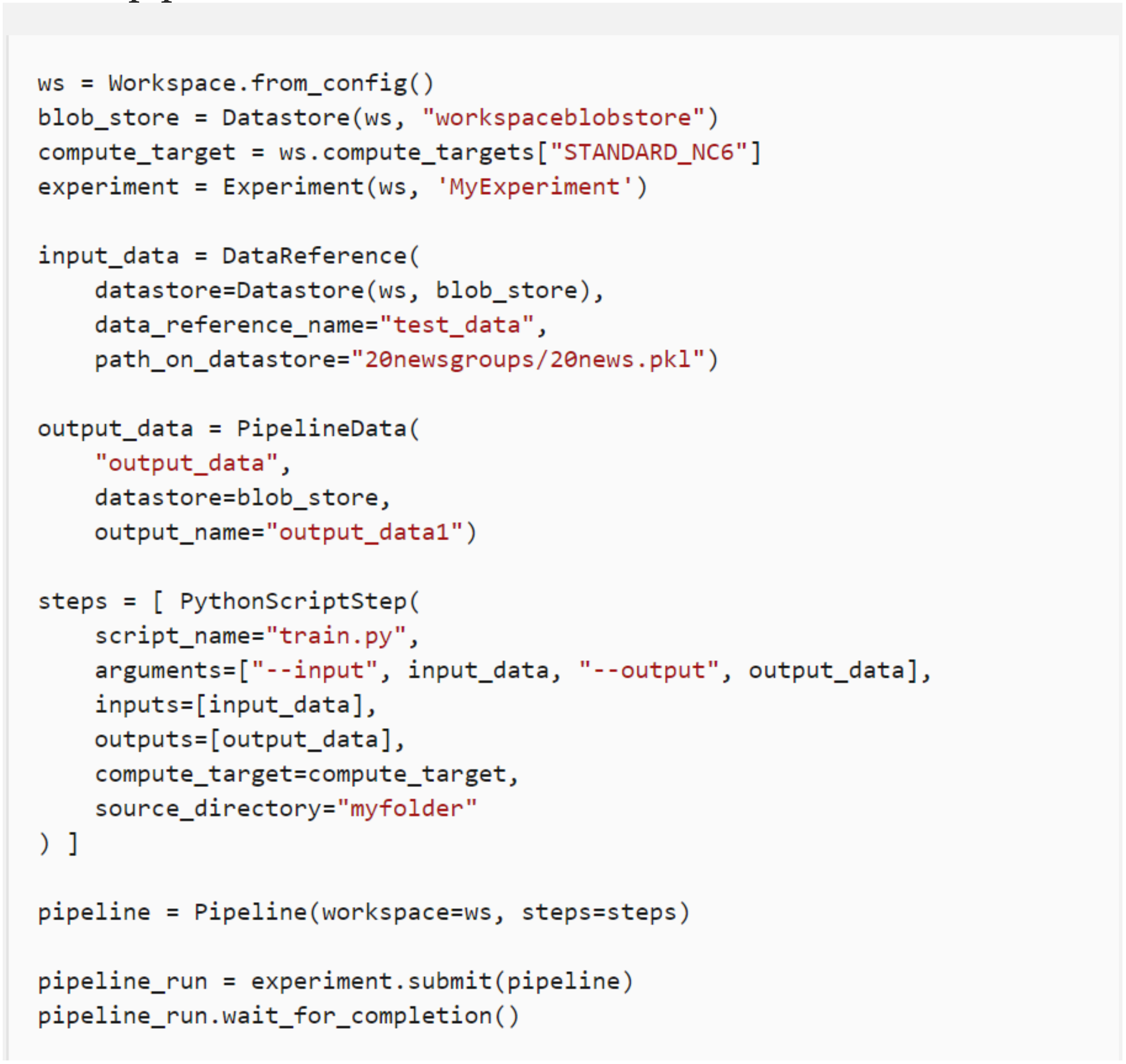
Use Cases

A machine learning model's life cycle needs to be more adaptable to model tuning and monitoring. With new data coming in frequently, there can be significant changes in the outcomes.

Currently, improvements are being made to the existing neural networks to make them run even when the data is vague and when there is a lack of labelled training data.

A simple Python Pipeline

This snippet shows the objects and calls needed to create and run a basic pipeline:



```
ws = Workspace.from_config()
blob_store = Datastore(ws, "workspaceblobstore")
compute_target = ws.compute_targets["STANDARD_NC6"]
experiment = Experiment(ws, 'MyExperiment')

input_data = DataReference(
    datastore=Datastore(ws, blob_store),
    data_reference_name="test_data",
    path_on_datastore="20newsgroups/20news.pkl")

output_data = PipelineData(
    "output_data",
    datastore=blob_store,
    output_name="output_data1")

steps = [ PythonScriptStep(
    script_name="train.py",
    arguments=["--input", input_data, "--output", output_data],
    inputs=[input_data],
    outputs=[output_data],
    compute_target=compute_target,
    source_directory="myfolder"
) ]

pipeline = Pipeline(workspace=ws, steps=steps)

pipeline_run = experiment.submit(pipeline)
pipeline_run.wait_for_completion()
```

Image by Shashanka Manyam

The snippet starts with common Azure Machine Learning objects, a *workspace*, a *Datastore*, a *Compute_Target* and an Experiment. Then, the code creates the objects to hold *input_data* and *output_data*. The array steps holds a single element, a *PythonScriptStep* that will use the data objects and run on the *Compute_Target*. Then, the code instantiates the *Pipeline* object itself, passing in the workspace and steps array. The call to *experiment.submit(pipeline)* begins the Azure ML pipeline run. The call to *wait_for_completion()* blocks until the pipeline is finished.

Until a project gets large or near to deployment, your pipelines should be coarser rather than fine-grained. If you think of your ML project as involving *stages* and a pipeline as providing a complete workflow to move you through a particular stage, you're on the right path.

So happy tinkering!!

References:

https://docs.microsoft.com/en-us/azure/machine-learning/service/concept-ml-pipelines