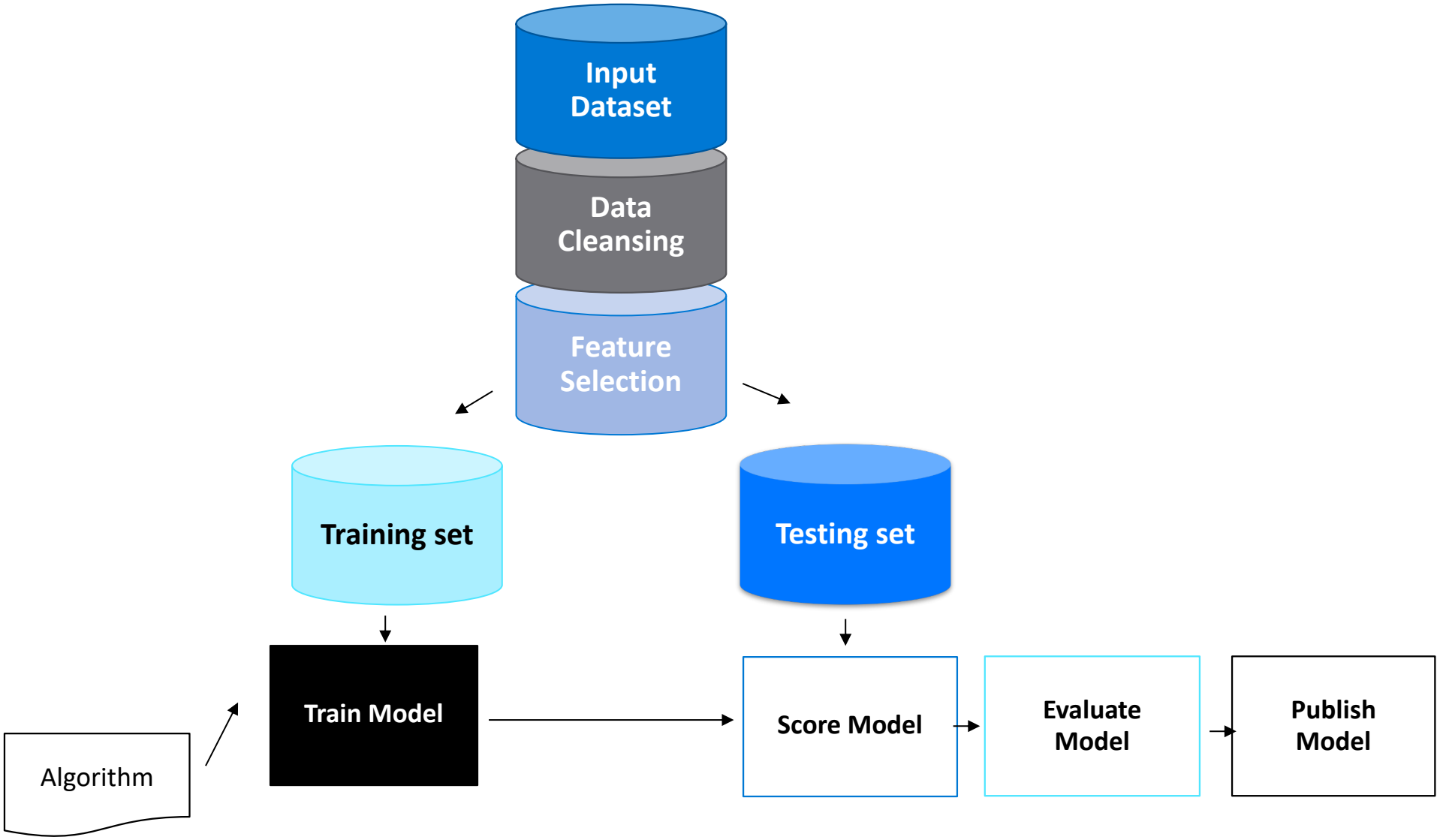

Microsoft
Data Drift Service for Azure ML
mem
proc
Alan Weaver
AI Specialist
Microsoft

What is Data Drift?

Machine Learning – Basic Workflow Recap



What happens to a model in production?

99% accuracy in training is not a guarantee of good performance in production

Monitoring production models is a very manual and tedious task - often based on complaints

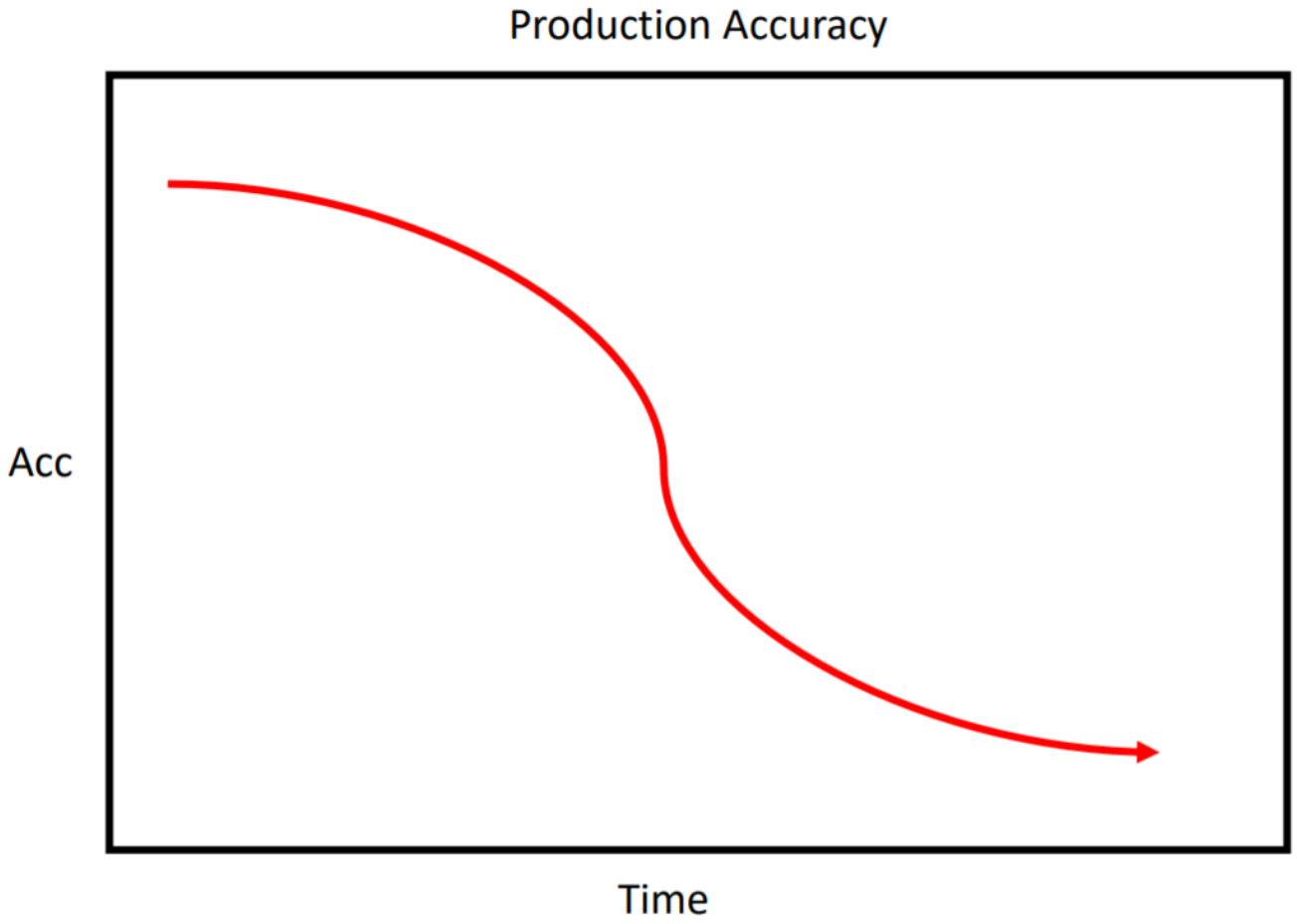
It is often difficult to know how well your model is performing “in the wild”

Models start to degrade due to changes in data, impact of deploying the system, etc.

Why care about Drift?

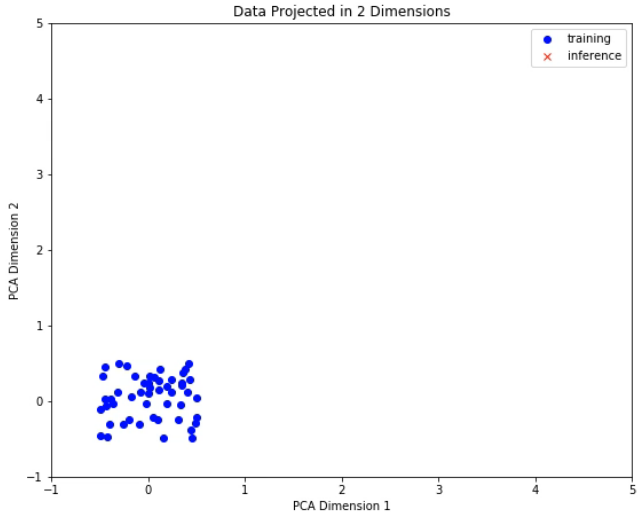
"The greatest model, trained on data inconsistent with the data is actually faces in the real world, will perform at best unreliably and at worst catastrophically".

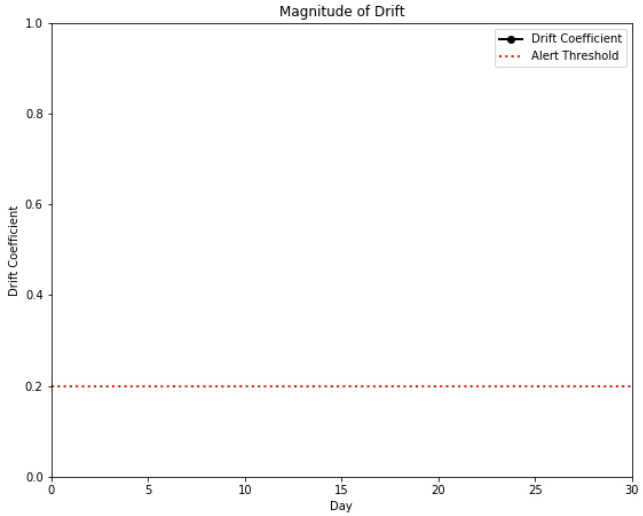
David Talby, CTO Pacific AI

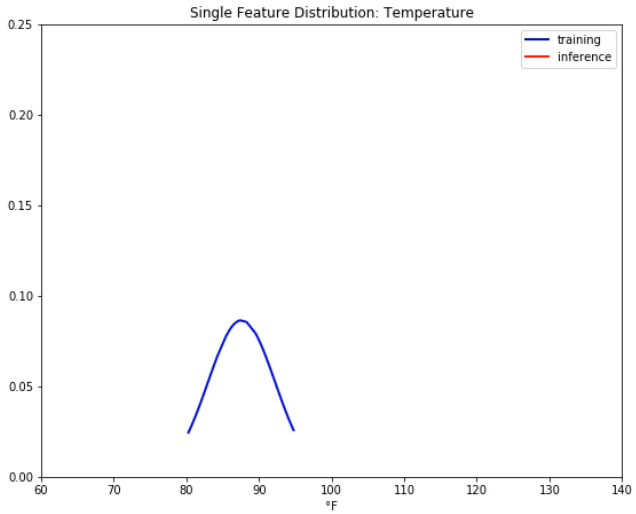


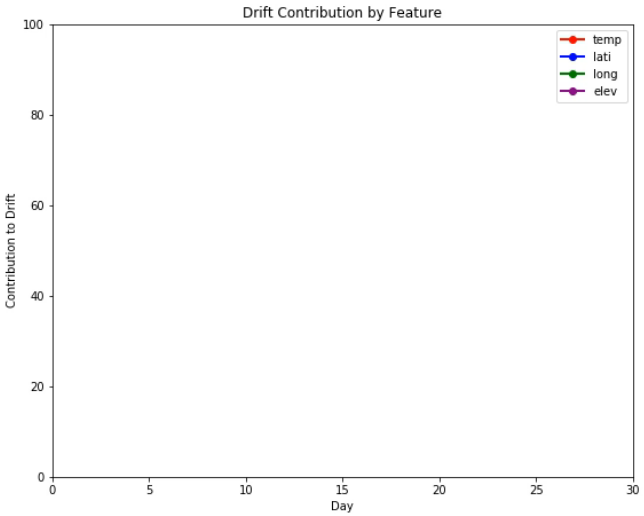
What does Drift look like?

Azure ML Data Drift | Private Preview | http://aka.ms/driftsignup | Day -1









User Flow - What do I do?

Monitoring my Model, Data

$$Age < 20 \; and \; Height < 50 \rightarrow Alert$$

$$Height < 100 \rightarrow Alert$$

Training

Age	Sex	Height	Weight	QRS Duration	…	QRST Angle
75	0	190	80	91	…	-2
56	1	165	64	81	…	31
54	0	172	95	138	…	66
55	0	175	94	100	…	20
75	0	190	80	88	…	3
13	0	169	51	100	…	88
40	1	160	52	77	…	65
49	1	162	54	78	…	51
44	0	168	56	84	…	66
…	…	…	…	…	…	…

Inference

Age	Sex	Height	Weight	QRS Duration	…	QRST Angle
15	1	90	40	139	…	0
23	0	65	32	91	…	63
34	1	72	47	184	…	2
8	1	75	47	97	…	36
27	1	90	40	89	…	38
12	1	69	26	99	…	84
10	0	60	23	103	…	22
43	0	62	22	152	…	53
4	1	68	51	144	…	45
…	…	…	…	…	…	…

Why Detect Concept Drift?

Early detection to Model performance issues

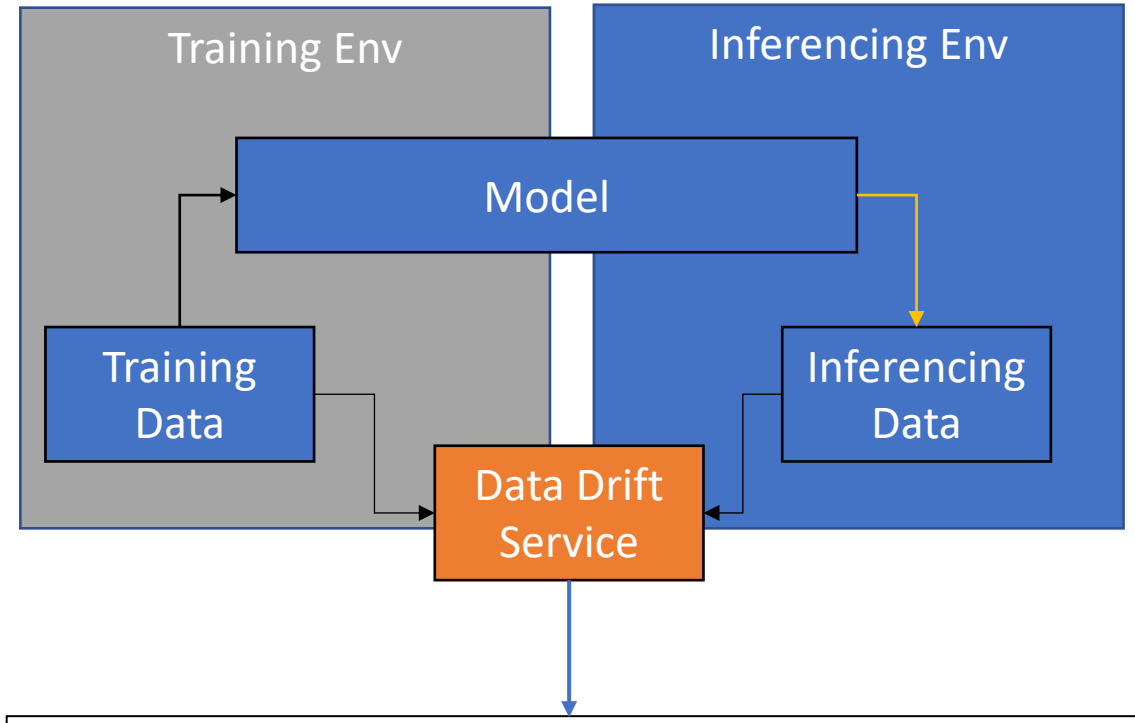
- model performance is increasingly linked to business decisions
- early detection of issues saves money, time, and customer frustration
- a model predicting a customer's purchasing preferences not adapting over time leads to lost revenue
- a model making healthcare decisions not adapting over time can lead to bigger issues

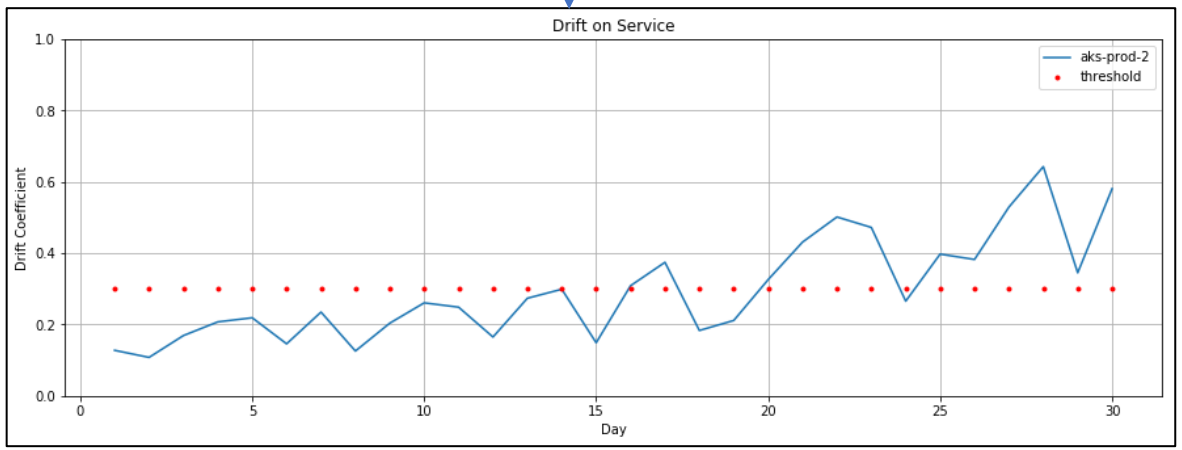
Better understanding of Data and Models used in decisions

- insights into why a model is performing poorly
- improve model on next iteration -> better business results

What is AzureML Data

Data Drift Service in Azure ML Overview





What is Data Drift

- When the inference data varies enough from the training data to cause model degradation, we consider it data drift

Why is Data Drift important

- in absence of labels, data drift can be used as a proxy for model performance metrics

What can I expect from data drift service

- Alerting and Insights
- Drift Coefficient - measures magnitude of drift
- Drift Contribution by Feature - measures feature that caused data drift
- Distance Metrics by Feature - distance metrics on features
- Distribution Visualization - capture and visualize change in distributions over time

Drift Coefficient

The **Drift Coefficient** is a number ranging from 0 to 1 representing the magnitude of drift between two datasets.

0: no drift

1: full confidence in drift

In the absence (or delay) of labels, the **Drift Coefficient** and related metrics give insights into how a model is performing.

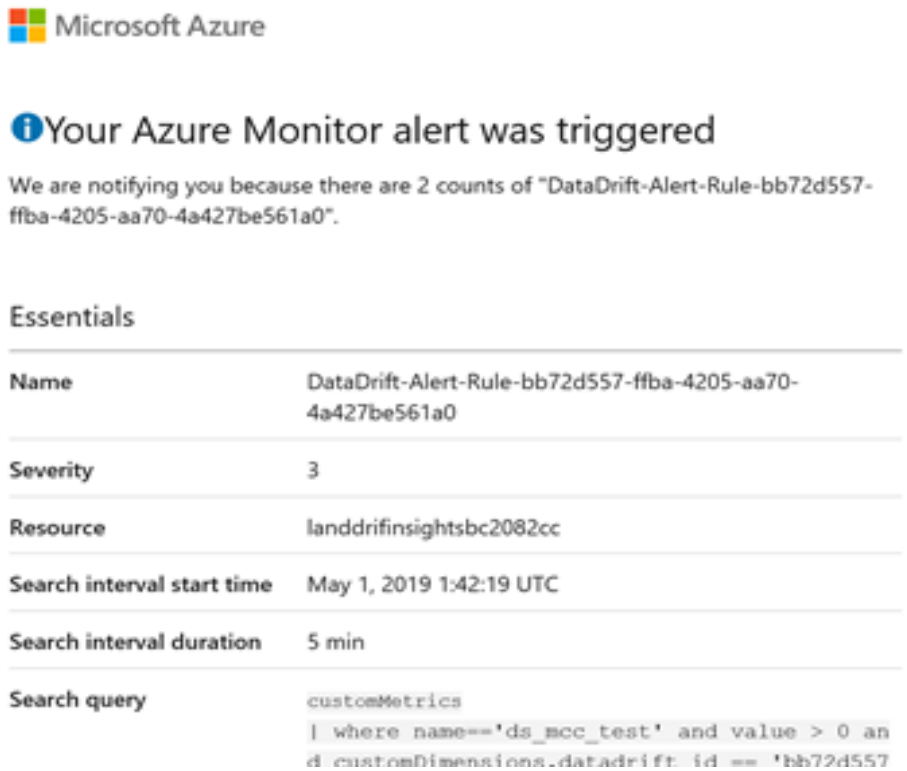
Drift Alerts

Email Alerts

Easily setup email alerting if the **Drift Coefficient** or any Distance Metrics exceed a threshold. All metrics are logged in Azure Monitor to configure alerts, or some can be configured through our Python SDK and in the Azure ML Workspace.

Event Grid

https://docs.microsoft.com/en-us/azure/event-grid/event-schema-machine-learning#microsoftmachinelearningservicesdatasetdriftdetected-event



Intended Usage

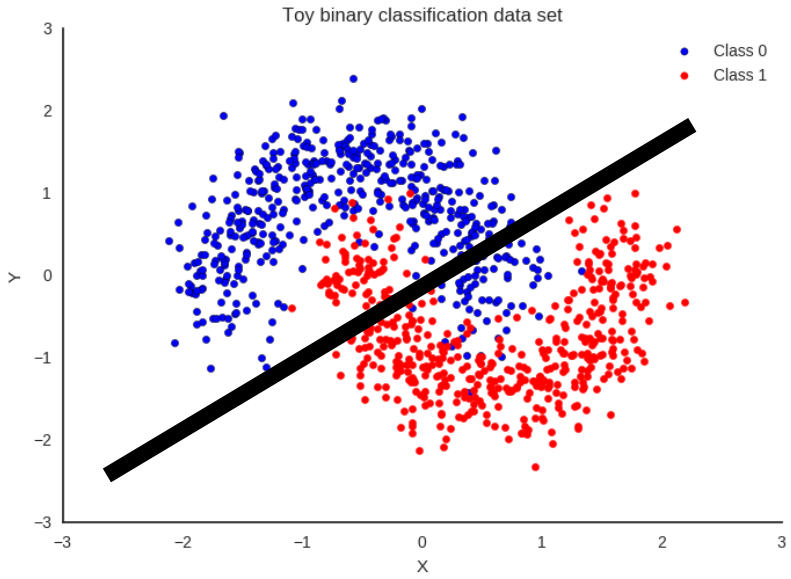
1. Use **Drift Coefficient** to receive alerts and monitor model performance
2. If **Drift Coefficient** exceeds threshold, investigate **Drift Importance by Feature** to determine where and why drift is detected
3. Use Distance Metrics and distribution visualization tools to further understand your data
4. Use raw inference data with Azure ML EDA for deeper analysis
5. Label and retrain based on investigation

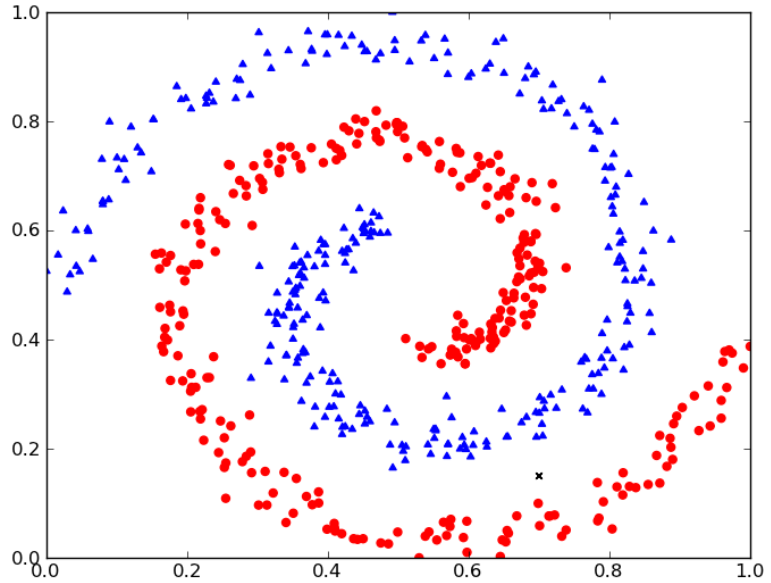
How does AzureML Data Drift Work

Classification using ML

Binary Classification

- Train a model to tell which of two classes a sample came from





Model Training Data

Age	Sex	Height	Weight	QRS Duration	...	QRST Angle	Arrythmia Class (Target)
75	0	190	80	91	...	-2	1
56	1	165	64	81	...	31	1
54	0	172	95	138	...	66	1
55	0	175	94	100	...	20	7
75	0	190	80	88	...	3	1
13	0	169	51	100	...	88	2
40	1	160	52	77	...	65	6
49	1	162	54	78	...	51	1
44	0	168	56	84	...	66	1
...	...	...	...	...	...	...	...

Model Inference Data

Age	Sex	Height	Weight	QRS Duration	...	QRST Angle	Arrythmia Class (Prediction)
75	0	190	80	91	...	-2	1
56	1	165	64	81	...	31	1
54	0	172	95	138	...	66	1
55	0	175	94	100	...	20	7
75	0	190	80	88	...	3	1
13	0	169	51	100	...	88	2
40	1	160	52	77	...	65	6
49	1	162	54	78	...	51	1
44	0	168	56	84	...	66	1
...	...	...	...	...	...	...	...

Dataset A (Baseline)

Age	Sex	Height	Weight	QRS Duration	...	QRST Angle	**Dataset**
75	0	190	80	91	...	-2	**A**
56	1	165	64	81	...	31	**A**
54	0	172	95	138	...	66	**A**
55	0	175	94	100	...	20	**A**
75	0	190	80	88	...	3	**A**
13	0	169	51	100	...	88	**A**
40	1	160	52	77	...	65	**A**
49	1	162	54	78	...	51	**A**
44	0	168	56	84	...	66	**A**
...	...	...	...	...	...	...	**...**

Dataset B (Target)

Age	Sex	Height	Weight	QRS Duration	...	QRST Angle	**Dataset**
75	0	190	80	91	...	-2	**B**
56	1	165	64	81	...	31	**B**
54	0	172	95	138	...	66	**B**
55	0	175	94	100	...	20	**B**
75	0	190	80	88	...	3	**B**
13	0	169	51	100	...	88	**B**
40	1	160	52	77	...	65	**B**
49	1	162	54	78	...	51	**B**
44	0	168	56	84	...	66	**B**
...	...	...	...	...	...	...	**...**

Dataset A

Age	Sex	Height	Weight	QRS Duration	...	QRST Angle	Dataset
75	0	190	80	91	...	-2	**A**
56	1	165	64	81	...	31	**A**
54	0	172	95	138	...	66	**A**
55	0	175	94	100	...	20	**A**
75	0	190	80	88	...	3	**A**
13	0	169	51	100	...	88	**A**
40	1	160	52	77	...	65	**A**
49	1	162	54	78	...	51	**A**
44	0	168	56	84	...	66	**A**
...	...	...	...	...	...	...	**...**

Dataset B

Age	Sex	Height	Weight	QRS Duration	...	QRST Angle	Dataset
75	0	190	80	91	...	-2	**B**
56	1	165	64	81	...	31	**B**
54	0	172	95	138	...	66	**B**
55	0	175	94	100	...	20	**B**
75	0	190	80	88	...	3	**B**
13	0	169	51	100	...	88	**B**
40	1	160	52	77	...	65	**B**
49	1	162	54	78	...	51	**B**
44	0	168	56	84	...	66	**B**
...	...	...	...	...	...	...	**...**

Drift Model

Data are Similar

- ML model trains and performs poorly on being able to tell A from B
- no alert needed
- stores drift metrics for you for use
- provide visualizations of drift metrics

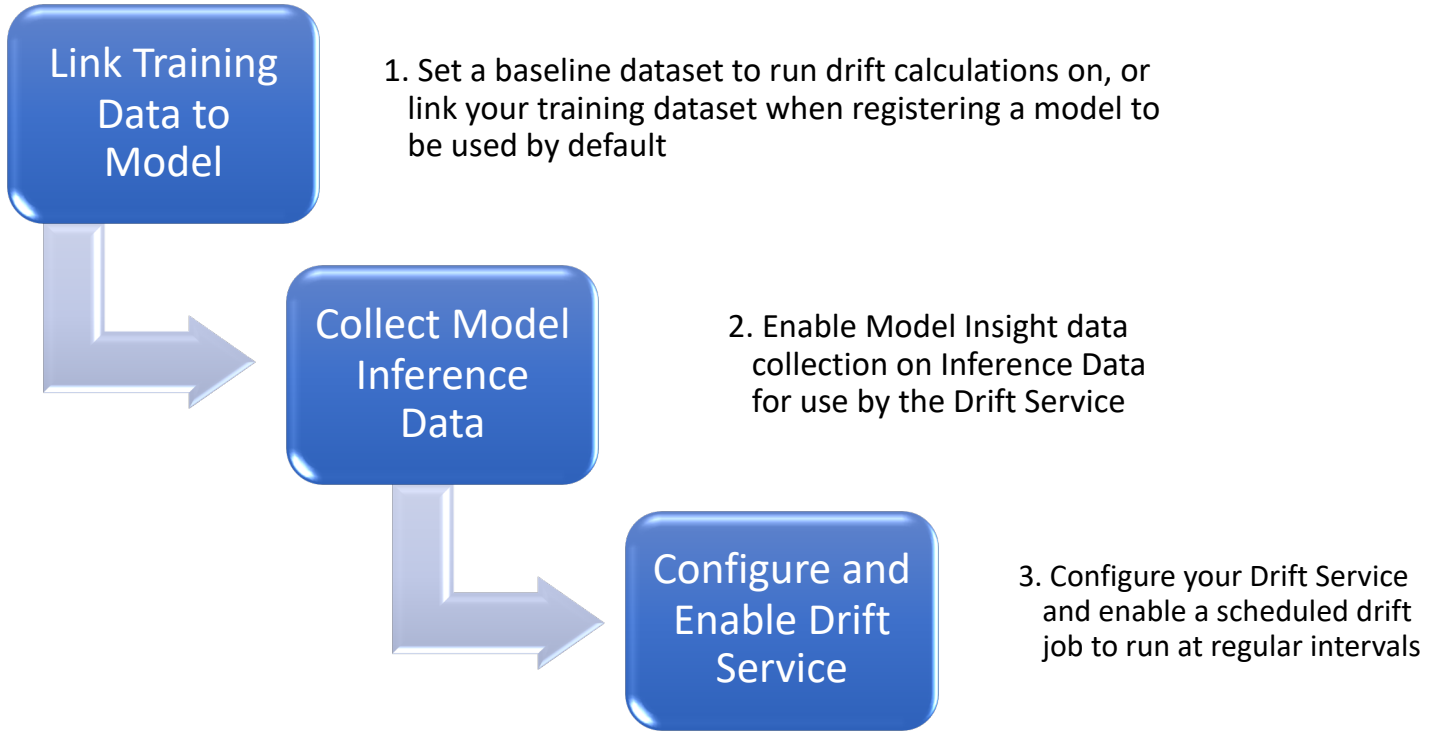
Data are Different

- ML model trains and performs well on being able to tell A from B
- can now alert you to the issue earlier
- identifies the age column has drifted
- gives insights, visualizations to help explain root cause

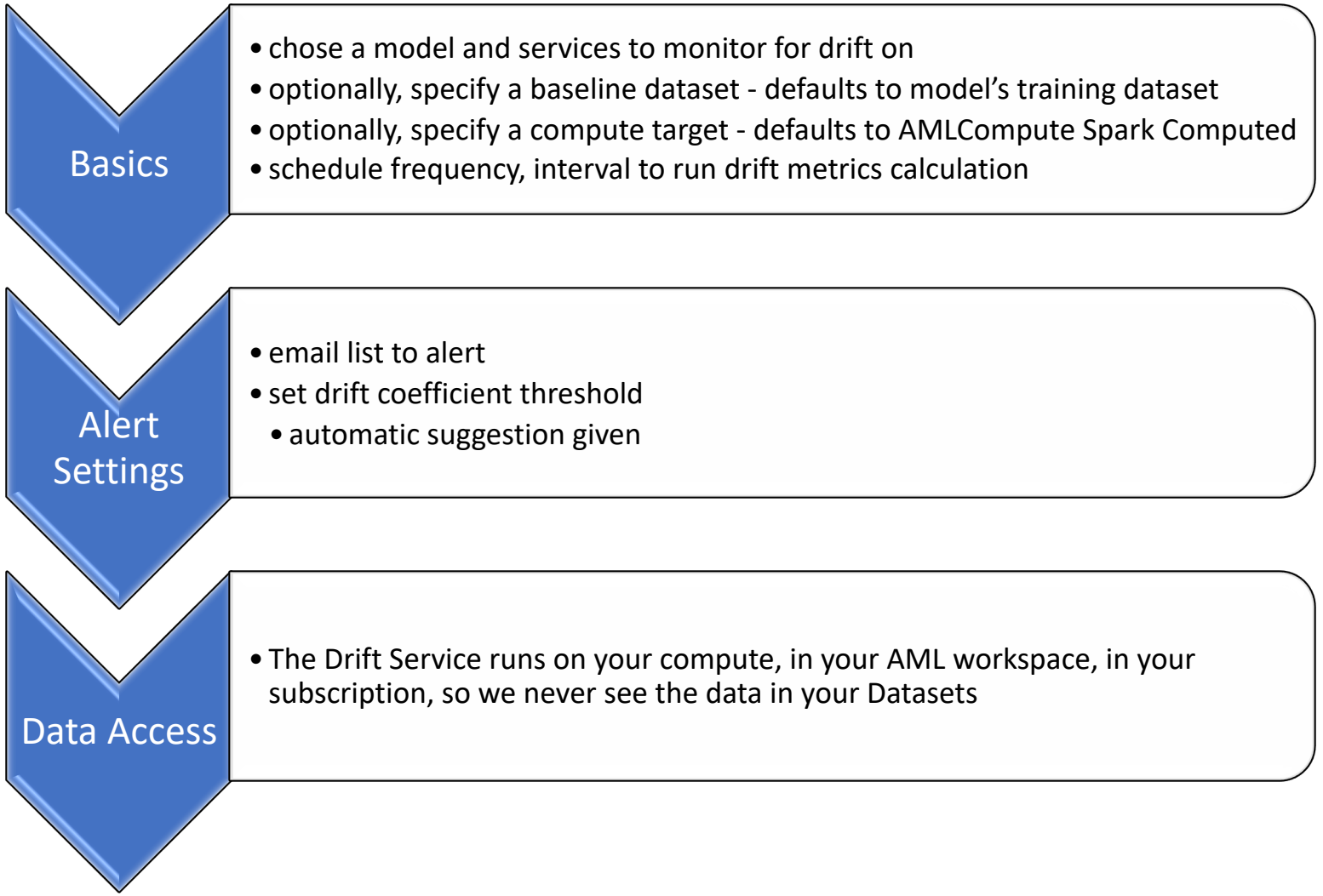
How do I use Data Drift in Azure ML?

Python SDK, UI

- there are 3 steps in either the Azure ML Python SDK or UI needed to schedule drift monitoring, alerts, and metrics



Configuration Options



Why Use Drift Service in Azure ML?

Model Monitoring

- drift provides a key insight into a model's performance
- easily collect inference data
- email alerting to any potential issues for quick investigation

Explainability of Drift

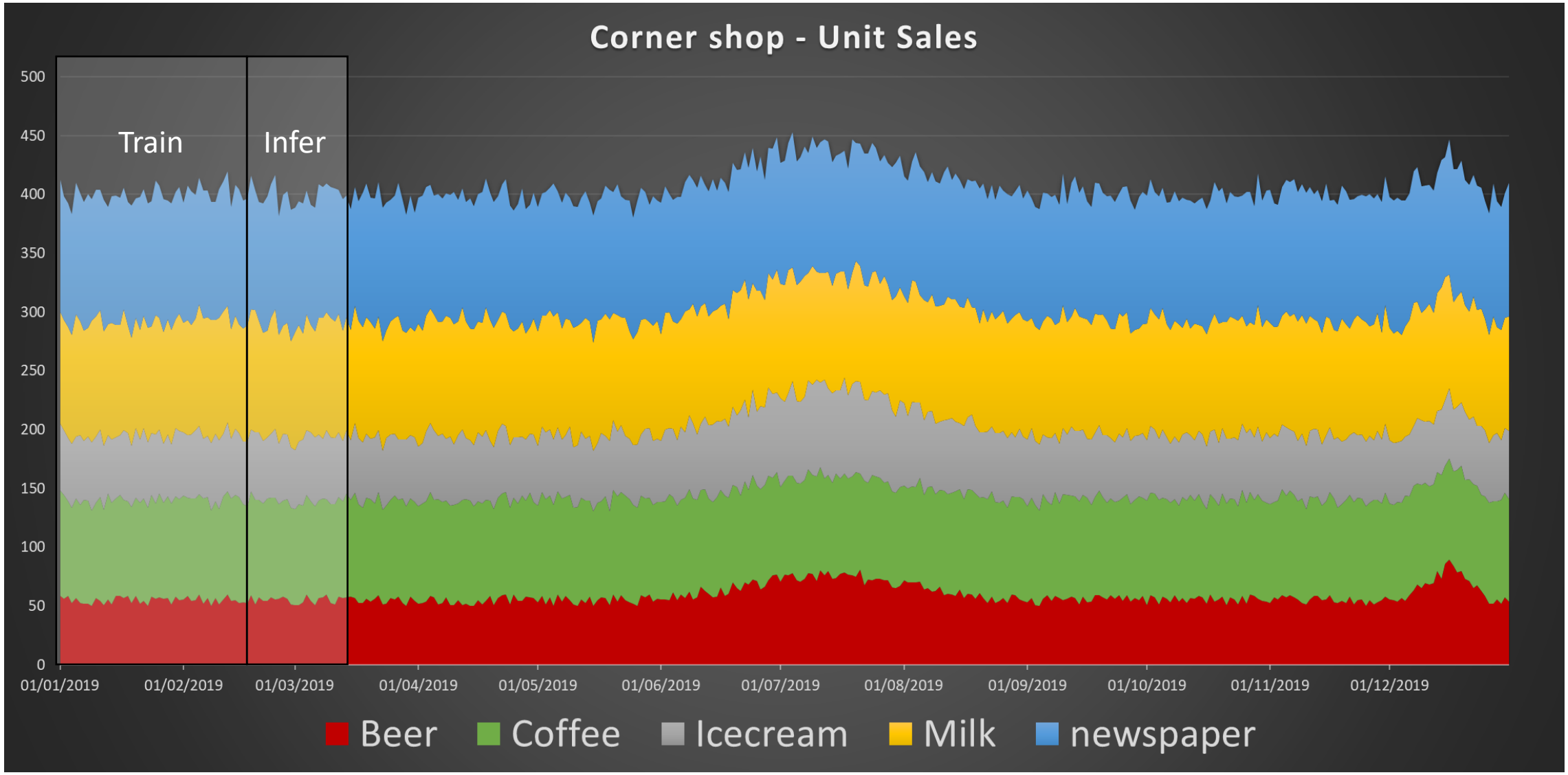
- which inference samples caused drift
- which features caused drift

Time Savings

- abstract manual labor of setting up data collection, cleaning data, comparing against training day, setting up alerting, etc.
- generalizable to all models without need for complex setup

Demo

Corner shop sales



Further reading

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-monitor-datasets
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-monitor-data-drift
- https://techcommunity.microsoft.com/t5/data-architecture-blog/data-drift-in-azure-machine-learning/ba-p/1201319
- https://docs.microsoft.com/en-us/python/api/azureml-datadrift/azureml.datadrift?view=azure-ml-py
- https://www.youtube.com/watch?v=qz1S-tv6iZs

Questions

Drift Setup - Private Preview

Setup Drift from Azure ML Python SDK or UI

1. Register training dataset
2. Link training dataset to model
3. Collect inference data
4. Configure and enable Data Drift service

Python Setup - Training Data to Model

```
# Register the training dataset to the model object
df = pd.read_csv("training.csv")
dataset = Dataset("trainingdataset", ws, PandasDataSource(df), datastore=ds)
dataset.save()

model = Model.register(model_path="model2-dd.pkl",
                       model_name = "sklearn_regression_model", # this is the name the model is registered as
                       tags = {'area': "store", 'type': "regression"},
                       description = "Store prediction",
                       workspace = ws,
                       dataset=dataset)
```

Python Setup - Collect Inference Data

Setup - Update Score.py File

```
%%writefile score.py
import pickle
import json
import numpy
from sklearn.externals import joblib
from sklearn.linear_model import Ridge
from azureml.core.model import Model
from azureml.monitoring import ModelDataCollector
import time

def init():
    global model
    print ("model initialized" + time.strftime("%H:%M:%S"))
    # note here "sklearn_regression_model.pkl" is the name of the model registered under the workspace
    # this call should return the path to the model.pkl file on the local disk.
    model_path = Model.get_model_path(model_name = 'sklearn_regression_model')
    # deserialize the model file back into a sklearn model
    model = joblib.load(model_path)
    global inputs_dc, prediction_dc
    # this setup will help us save our inputs under the "inputs" path in our Azure Blob
    inputs_dc = ModelDataCollector(model_name="sklearn_regression_model", identifier="inputs", feature_names=["Store_Latitude", "Sto
    # this setup will help us save our ipredictions under the "predictions" path in our Azure Blob
    prediction_dc = ModelDataCollector("sklearn_regression_model", identifier="predictions", feature_names=["kwh"])

# note you can pass in multiple rows for scoring
def run(raw_data):
    global inputs_dc, prediction_dc
    try:
        data = json.loads(raw_data)['data']
        data = numpy.array(data)
        result = model.predict(data)
        print ("saving input data" + time.strftime("%H:%M:%S"))
        inputs_dc.collect(data) #this call is saving our input data into our blob
        prediction_dc.collect(result)#this call is saving our prediction data into our blob
        print ("saving prediction data" + time.strftime("%H:%M:%S"))
        # you can return any data type as long as it is JSON-serializable
        return result.tolist()
    except Exception as e:
        error = str(e)
        print (error + time.strftime("%H:%M:%S"))
        return error
```

Python Setup - Drift Configuration, Enable

Cmd 3

```
# Construct datadrift object

services = ["aks-dd-3"]
start = datetime(year=2019, month=2, day=21)
end = datetime(year=2020, month=1, day=22, hour=15, minute=16)

datadrift = DataDrift(ws, model, services, frequency="Day", start_time=start)
```

```
found existing compute target.
```

Command took 0.28 seconds -- by rafarmah@microsoft.com at 3/5/2019, 4:26:36 PM on RamandeepCluster

Cmd 6

```
# Enable the schedule based on the frequency and schedule_start (defaults to current time if not specified)
datadrift.enable_schedule()

print("Datadrift is enabled: {}".format(datadrift.enabled))

# Disable schedule
datadrift.disable_schedule()

print("Datadrift is disabled: {}".format(datadrift.enabled))
```

Python Setup - Configuration, Enable w/ Alerting

```
# Alert if any of these features exceed threshold on measures
feature_alerts = {'temperature':
                      {'wasserstein_distance': 50,
                       'energy_distance': 100,
                       },
                  }

# Add alert config

alerts = {'emails': ['user1@contoso.com', 'user2@contoso.com'],
          'Drift Coefficient': .3,
          'Feature_measures': feature_alerts
          }

datadrift.update(alert_config=alerts)

datadrift.enable_schedule()
```

```
found existing compute target.
WARNING:root:Schedule.activate() is being deprecated. Use Schedule.enable() instead.
⊞ErrorResponseException: (BadRequest) Cannot update an entity while it's in Provisioning state.
```

Command took 1.77 seconds -- by rafarmah@microsoft.com at 3/5/2019, 4:25:45 PM on RamandeepCluster

Cmd 8

```
# Print list of datadrift schedules attached to the model
datadrifts = DataDrift.get(ws, model)

for dd in datadrifts:
  print(dd.workspace.name, dd.model.name, dd.services)
```