Microsoft

Module 10: Monitoring Models

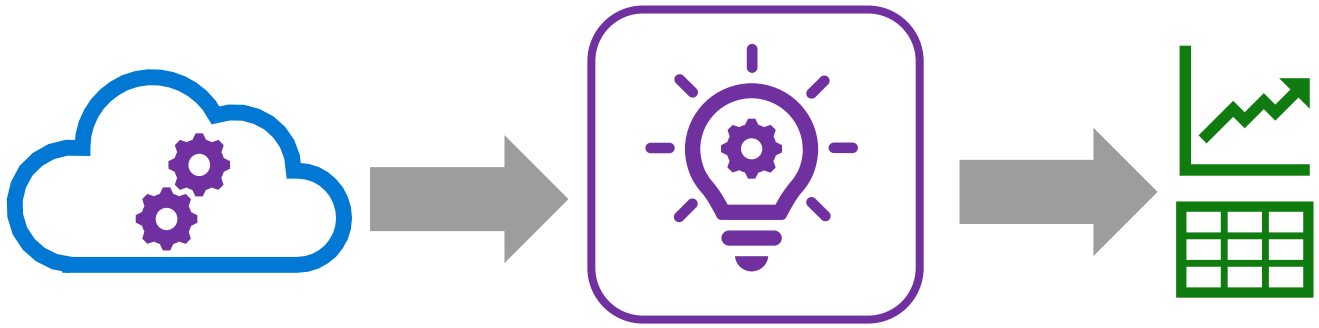
Agenda

- Monitoring Models with Application Insights
- Monitoring Data Drift

Lesson 1
Monitoring Models with Application Insights

What is Application Insights?

- An Application Performance Management service in Azure
- Enables capture, storage, and analysis of telemetry data



Enabling Application Insights

- Determine the Application Insights resource for your workspace

```
ws.get_details()['applicationInsights']
```

- Enable in a new service deployment configuration using the SDK:

```
deploy_config = Webservice.deploy_configuration(enable_app_insights=True)
```

- Enable for existing deployed services:
  - Configure AKS Deployment in the Azure portal
  - Update deployed service using the SDK

```
service.update(enable_app_insights=True)
```

Capturing and Viewing Application Insights Data

- Print log data in the scoring script

```
def init():
    model = joblib.load(Model.get_model_path('my_model'))
def run(raw_data):
    data = json.loads(raw_data)['data']
    predictions = model.predict(data)
    log_txt = 'Data:' + str(data) + ' - Predictions:' + str(predictions)
    print(log_txt)
```

- Query Logs in Application Insights

```
traces
|where  message == "STDOUT" and customDimensions.["Service Name"] = "my-svc"
| project  timestamp, customDimensions.Content
```

timestamp	customDimensions_Content
01/02/2020, 9:11:57.846 PM	Data:[[1, 2, 2.5, 3.1], [0, 1, 1,7, 2.1]] - Predictions:[0 1]

Microsoft

Lab 10A

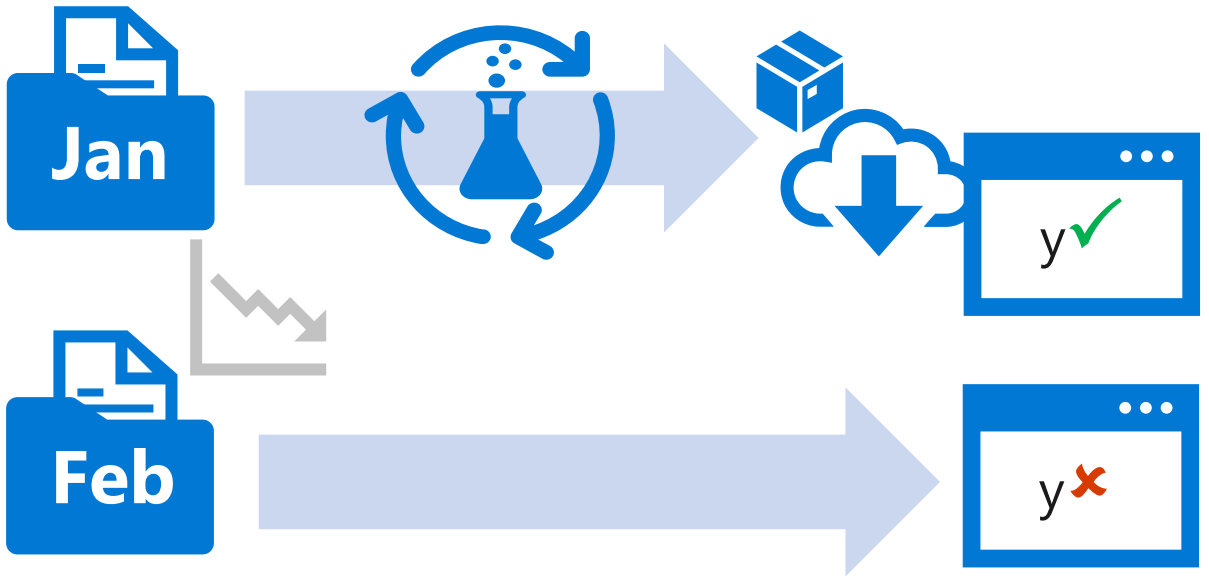
Monitoring a Model with Application Insights

https://aka.ms/msl-dp100

Lesson 2
Monitoring Data Drift

What is Data Drift?

- Changing data trends that can affect the accuracy of trained models



Creating a Data Drift Monitor

- Monitor by Comparing Datasets
  - Baseline dataset (original training data)
  - Target dataset for comparison over time (requires timestamp column)
  - Backfill to populate a data drift profile from target dataset

```
monitor = DataDriftDetector.create_from_datasets(ws, 'dataset-drift-detector',
                                                 baseline_data_set, target_data_set, ...)
backfill = monitor.backfill(dt.datetime.now() - dt.timedelta(days=30), dt.datetime.now())
```

- Monitor Service Inference Data
  - Include training dataset in model registration as baseline
  - Enable data collection for deployed model service
  - Run monitor to get data drift profile from scoring data

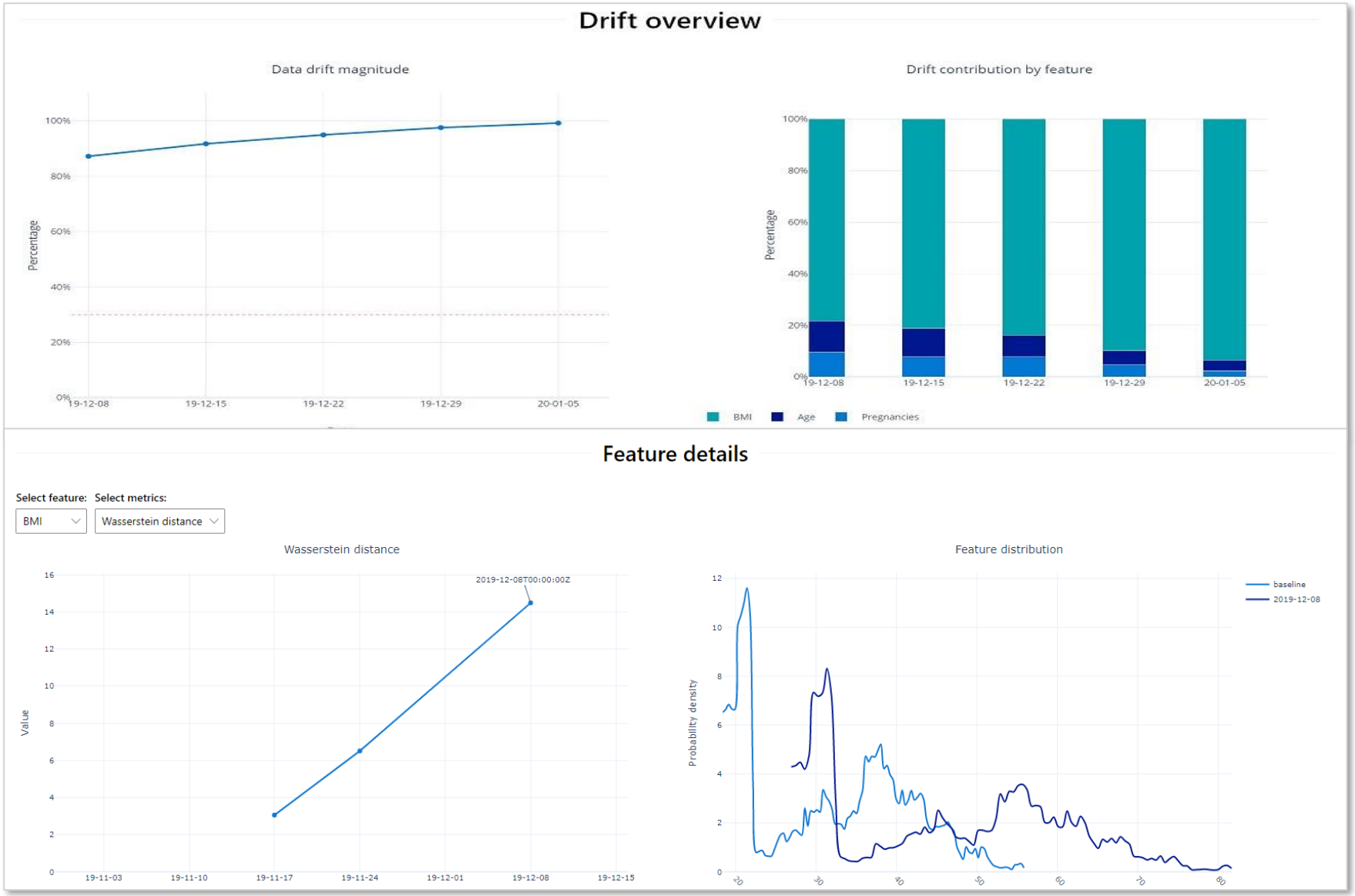
```
model = Model.register(workspace=ws,model_path='./model/model.pkl', model_name='my_model',
                       datasets=[(Dataset.Scenario.TRAINING, train_ds)])
monitor = DataDriftDetector.create_from_model(ws, model.name, model.version, ...)
run = monitor.run(dt.datetime.now(), services, ...)
```

Data Drift Schedules and Alerts

- On creation, specify:
  - Frequency
  - Drift threshold for alerting
  - Alert configuration
  - Schedule start (for model data drift monitors)
  - Data latency (for dataset data drift monitors)

```
alert_email AlertConfiguration('data_scientists@contoso.com')
monitor = DataDriftDetector.create_from_datasets(ws, 'dataset-drift-detector',
                                                 baseline_data_set, target_data_set,
                                                 compute_target=cpu_cluster,
                                                 frequency='Week', latency=2,
                                                 drift_threshold=.3,
                                                 alert_configuration=alert_email)
```

Reviewing Data Drift



Microsoft

Lab 10B

Monitoring Data Drift

https://aka.ms/msl-dp100