AML for MLOps with Microsoft Azure

As the use of Machine Learning (ML) becomes more widespread, so too does the need for scalable, automated solutions that can support the development, deployment, and management of ML models. This is where MLOps comes in - a set of practices and tools that enable organizations to streamline the development and deployment of ML models while ensuring their quality, security, and compliance.

However, MLOps faces a significant challenge when it comes to Anti-Money Laundering (AML) compliance. AML regulations require companies to identify and report suspicious financial transactions, and failure to comply can result in significant fines and reputational damage. Therefore, any ML model that is used for AML purposes needs to be carefully designed, implemented, and monitored to ensure its accuracy and effectiveness. In this blog, we will explore how Microsoft Azure can be used to implement AML for MLOps.

Problem Statement:

The problem with AML for MLOps is that it requires a complex set of steps to ensure compliance, including data preparation, feature engineering, model training, and model monitoring. Additionally, there are regulatory requirements that must be met to ensure that the model is transparent, interpretable, and auditable. This complexity can make it challenging to implement AML for MLOps effectively, especially if organizations lack the necessary resources and expertise.

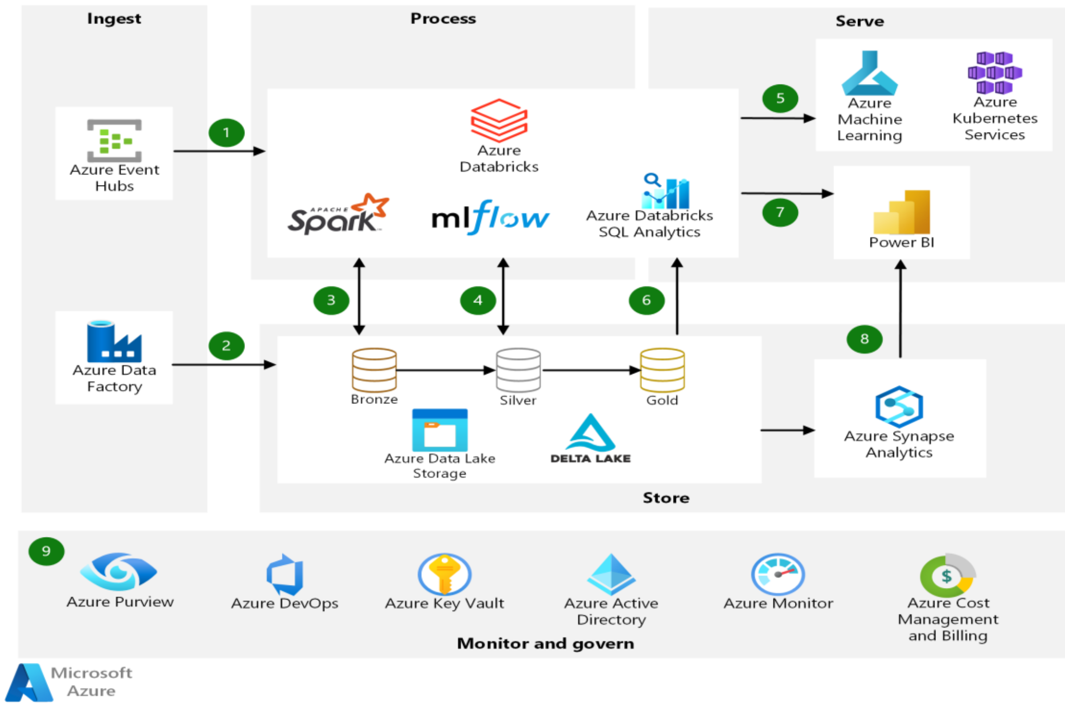
Solution/ Architecture:

To address this challenge, we can use Microsoft Azure to implement a comprehensive AML solution that includes data preparation, model training, and model monitoring. The solution consists of several components, including Azure Machine Learning, Azure Data Factory, Azure Databricks, and Azure Synapse Analytics.

Azure Machine Learning provides a scalable, collaborative environment for building and deploying ML models. It includes a range of tools and services for data preparation, model training, and model management.

Azure Data Factory is a cloud-based data integration service that allows us to create, schedule, and manage data pipelines. It can be used to ingest data from various sources, transform the data, and store it in different destinations.

Azure Databricks is a fast, easy, and collaborative Apache Spark-based analytics platform that allows us to build and deploy ML models at scale. It includes a range of tools for data preparation, model training, and model management.



Azure Synapse Analytics is an integrated analytics service that allows us to analyze data from various sources using a unified experience. It includes a range of tools for data preparation, data warehousing, and big data analytics.

Technical Details and Implementation of Solution :

The first step in implementing the AML solution is to prepare the data. This involves cleaning, transforming, and feature engineering the data to make it suitable for use in ML models. We can use Azure Data Factory to ingest the data from various sources, transform it using Azure Databricks, and store it in Azure Synapse Analytics. The following code shows an example of how to use Azure Data Factory to ingest data from a CSV file and store it in Azure Synapse Analytics.

Json

```
{
    "name": "Copy data from CSV to Synapse",
    "type": "Copy",
    "inputs": [
        {
            "name": "CsvDataset"
        }
    ],
    "outputs": [
        {
            "name": "SynapseDataset"
        }
    ],
    "activities": [
        {
            "type": "DataFlow",
            "name": "Transform data",
            "inputs": [
                {
                    "name": "CsvDataset"
                }
            ],
            "outputs": [
                {
                    "name": "SynapseDataset"
                }
            ],
            "dataFlow": {
                "name":dataflow",
                "type": "MappingDataFlow",
                "source": {
                    "type": "DelimitedTextSource",
                    "formatSettings": {
                    "separator": ","
                    }
                    },
                "sink": {
                    "type": "SqlSink",
                    "sqlWriterCleanupScript": "",
                    "writeBatchSize": 100000,
                    "writeBatchTimeout": "PT1H"
                    }
                }
            }
        ],
    "linkedServices": [
        {
            "name": "CsvLinkedService",
            "type": "AzureBlobStorage",
            "typeProperties": {
                "connectionString": "<CSV Connection String>"
                }
            },
        {
            "name": "SynapseLinkedService",
            "type": "AzureSqlDatabase",
            "typeProperties": {
                "connectionString": "<Synapse Connection String>"
                }
            }
        ]
}
```

Once the data is prepared, we can use Azure Machine Learning to train and deploy the ML model. This involves selecting an appropriate algorithm, defining the features, and tuning the hyperparameters. We can use Azure Databricks to train the model and Azure Machine Learning to deploy it. The following code shows an example of how to train and deploy an ML model using Azure Databricks and Azure Machine Learning.

```
from azureml.core import Workspace, Experiment
from azureml.train.automl import AutoMLConfig

ws = Workspace.from_config()

experiment_name = 'my-experiment'
experiment = Experiment(ws, experiment_name)

automl_config = AutoMLConfig(task='classification',
training_data=<Training Data>,
label_column_name=<Label Column>,
iterations=10,
primary_metric='AUC_weighted')

run = experiment.submit(automl_config)
run.wait_for_completion(show_output=True)

best_run = run.get_best_child()
model = best_run.register_model(model_name='my-model', model_path='./outputs/model.pkl')

Deploy the model as a web service
from azureml.core.webservice import AciWebservice, Webservice
from azureml.core.model import InferenceConfig
from azureml.core.environment import Environment

env = Environment.from_conda_specification('my-env', './environment.yml')
inference_config = InferenceConfig(entry_script='score.py', environment=env)

aci_config = AciWebservice.deploy_configuration(cpu_cores=1, memory_gb=1)

service_name = 'my-service'
service = Model.deploy(ws, service_name, [model], inference_config, aci_config)
service.wait_for_deployment(show_output=True)

print(service.scoring_uri)
```

Challenges in Implementing the Solution:

One of the biggest challenges in implementing an AML solution for MLOps is data preparation. Data must be properly cleaned, normalized, and formatted before it can be used to train an ML model. This can be a time-consuming and complex process, especially if the data is coming from multiple sources or is very large.

Another challenge is selecting the right ML algorithm and tuning the hyperparameters. ML algorithms can be complex and there are many different options to choose from. Additionally, the performance of the ML model can depend heavily on the hyperparameters, which can be difficult to optimize.

Finally, monitoring the model's performance can also be a challenge. The ML model may degrade over time as the data changes or as new data is introduced. It is important to continually monitor the model's performance and update it as necessary to maintain accuracy and effectiveness.

Business Benefit:

Implementing an AML solution for MLOps can provide significant benefits for organizations. By automating many of the processes involved in ML development, organizations can reduce the time and cost associated with developing and deploying ML models. This can lead to faster time to market and improved business outcomes.

Additionally, by monitoring the performance of ML models, organizations can identify issues early and take action to address them before they impact business operations. This can help to ensure that ML models are effective and accurate over time.

Conclusion:

In this blog post, we have discussed how to implement an AML solution for MLOps using the Microsoft Azure platform. We have covered the problem statement, solution architecture with code, technical details and implementation of the solution with code, challenges in implementing the solution, and business benefits.

Implementing an AML solution for MLOps can help organizations to accelerate ML development and deployment, monitor model performance, and ensure that ML models remain accurate and effective over time. With the powerful tools and capabilities offered by the Microsoft Azure platform, organizations can quickly and easily implement an AML solution that meets their specific needs and requirements.

References:

https://learn.microsoft.com/en-us/azure/architecture/solution-ideas/articles/azure-databricks-modern-analytics-architecture

-by Dinesh Chauhan (Technical Head)

dinesh1chauhan123@gmail.com