MACHINE LEARNING

Eng Teong Cheah

Malaysia, 27 February 2021

http://bit.ly/taruccloudmlworkshop

CUSTOMER SITUATION

- Trey Research Inc. delivers innovative solutions for manufacturers.
- They specialize in identifying and solving problems for manufacturers that can run the range form automating away mundane but time-intensive processes to delivering cutting edge approaches that provide new opportunities for their manufacturing clients.
- Trey Research is looking to provide the next generation experience for connected car manufacturers by enabling them to utilize AI to decide when to pro-actively reach out to the customer thru alerts delivered directly to the car’s in-dash information and entertainment head unit.
- For their PoC, they would like to focus on two maintenance related scenarios.

CUSTOMER SITUATION

Requirements

- Trey Research recently instituted new regulations defining what parts are compliant or out of compliance. Rather than rely on their technicians to assess compliance, they would like to automatically assess the compliance based on component notes already entered by authorized technicians.

CUSTOMER SITUATION

Requirements

- Trey Research would like to predict the likelihood of batter failure based on the time series-based telemetry data that the car provides. The data contains details about how the battery performs when the vehicle is started, how it is charging while running, and how well it is holding its charge, among other factors. If they detect a battery failure s imminent within the next 30 days, they would like to send an alert. They are, however, concerns about the quality of the battery telemetry data, so Trey Research would like to be sure that, before fed into the Machine Learning process, data is properly cleansed and prepared.

CUSTOMER SITUATION

Requirements

- Trey Research wants to understand how they might use machine learning or deep learning in both scenarios and standardize the platform that would support the data processing, model management and inferencing aspects of each.
- They are also interested to learn what new capabilities Azure provides that might help them to integrate with their existing investments in MLflow for managing machine learning experiments. Furthermore, they would also like to understand how Azure might help them to document and explain the models that are created to non-data scientists or might accelerate their time to creating production ready, performant models.

CUSTOMER SITUATION

Requirements

- They would like to be able to easily create dashboards that summarize the alerts generated so they can observe the solution in operation.

CUSTOMER NEEDS

- We would like to minimize the number of distinct technologies we need to apply across the data science process, from data collection, exploratory data analysis, data preparation, model training, model management and model deployment. We already have existing investments in MLflow. How can Azure helps us in this?
- Our board is concerned about the ability of not being able to justify and explain the function of the models we create. We recognize many models are not easily explained in layman’s terms, but how can we go about documenting how our models make predictions generally or how why they made specific predictions for particular input?

CUSTOMER NEEDS

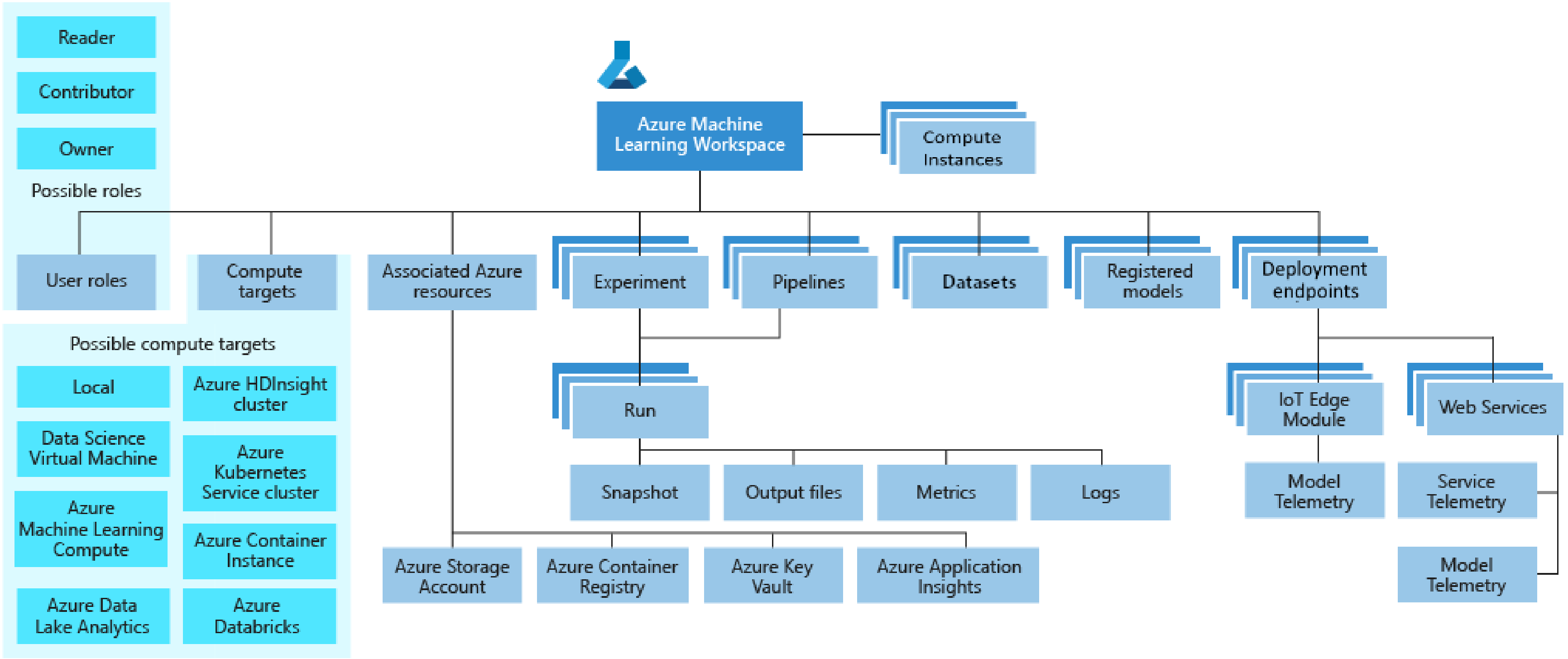
- Understand how they should create the NLP based model for compliance and the battery life-span forecasting model.
- Know the data pipeline they need to build in Azure, from ingesting telemetry, to storing both the compliance text and battery telemetry, to visualizing the result.

CUSTOMER OBJECTIONS

- Should we use machine learning or deep learning approaches?
- How should we choose between Keras and PyTorch for performing deep learning?
- We have heard Azure Machine Learning supports automated machine learning; can we use automated machine learning to create models using deep learning? Can we really expect a non-data scientist to create performant models using these tools?
- Some of our team has worked with Azure Databricks, and they are confused by the overlap with Azure Machine Learning. How should we be thinking about when to use which?
- We have heard a lot about how complicated and opaque trained deep learning models are. How is it even possible to attempt to explain them?

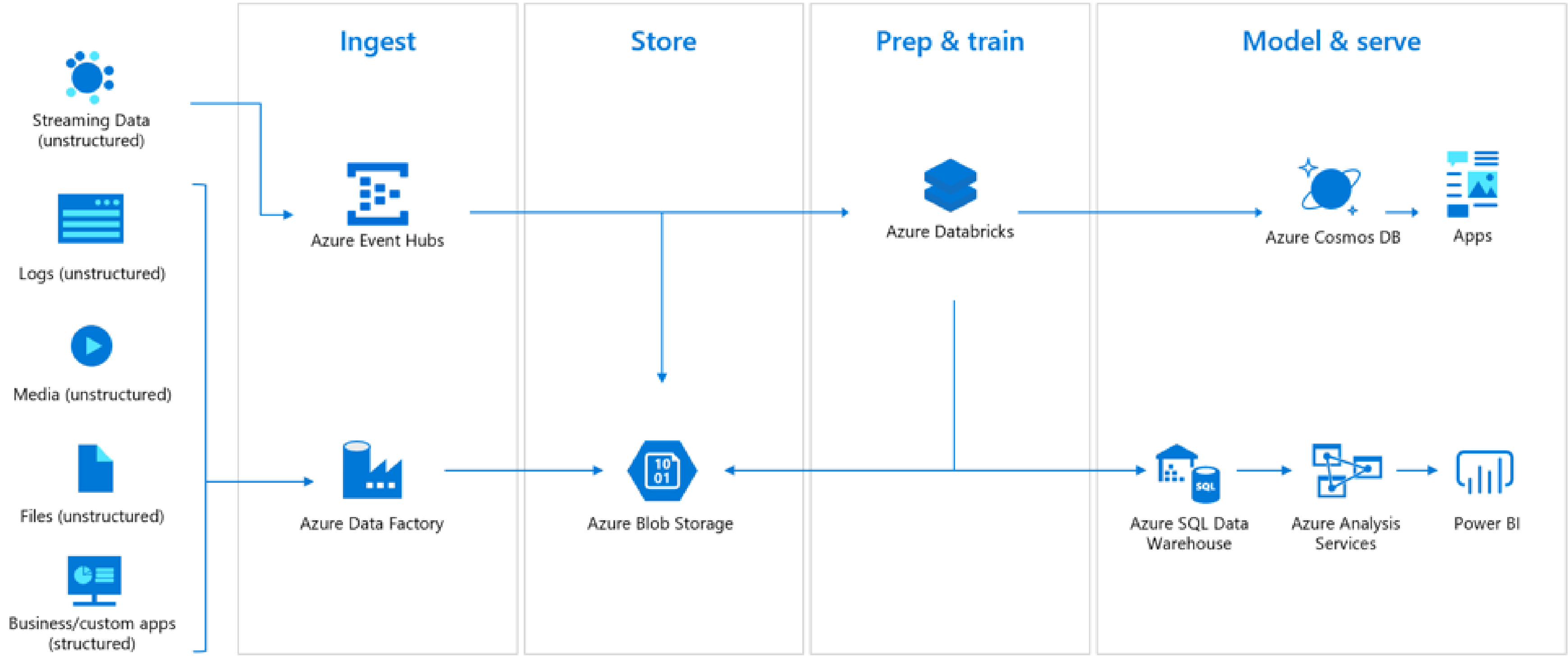
COMMON SCENARIOS

Azure Machine Learning workspace taxonomy



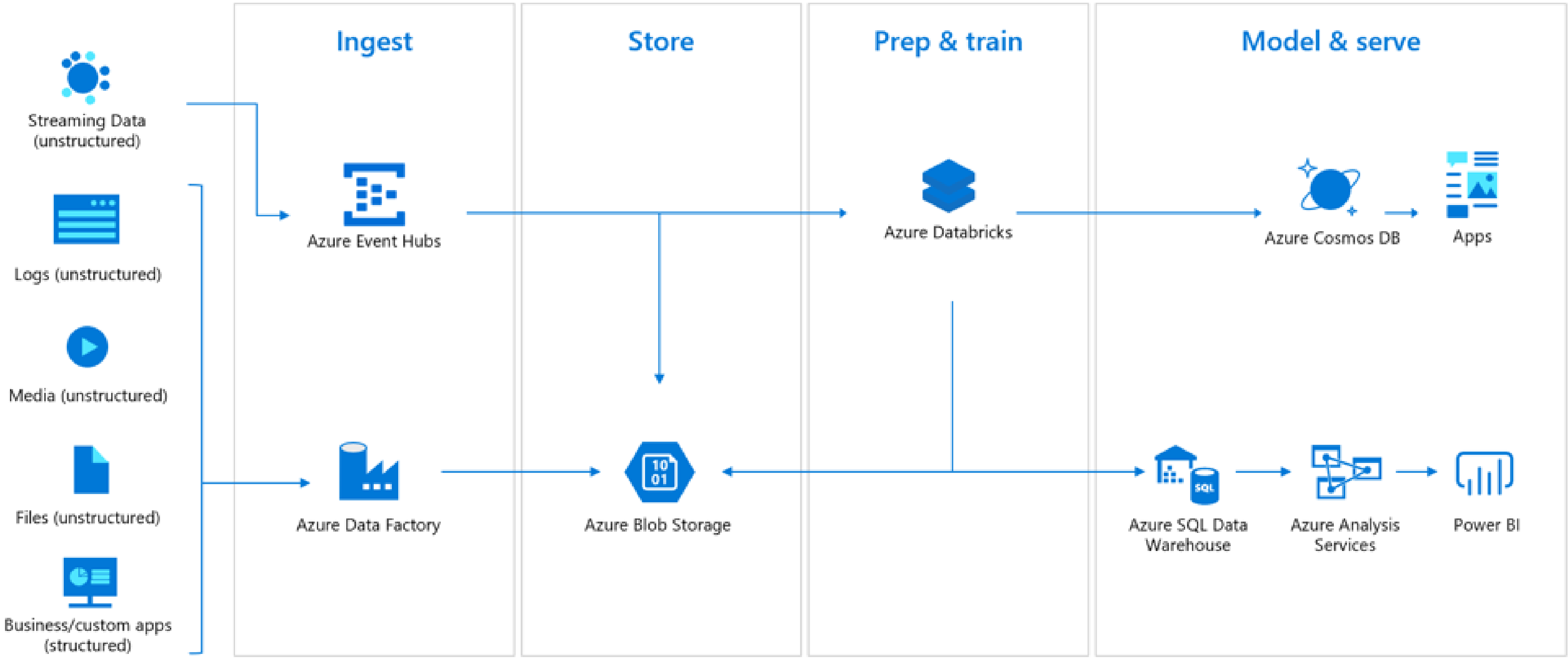
COMMON SCENARIOS

Real-time analytics



COMMON SCENARIOS

Real-time analytics

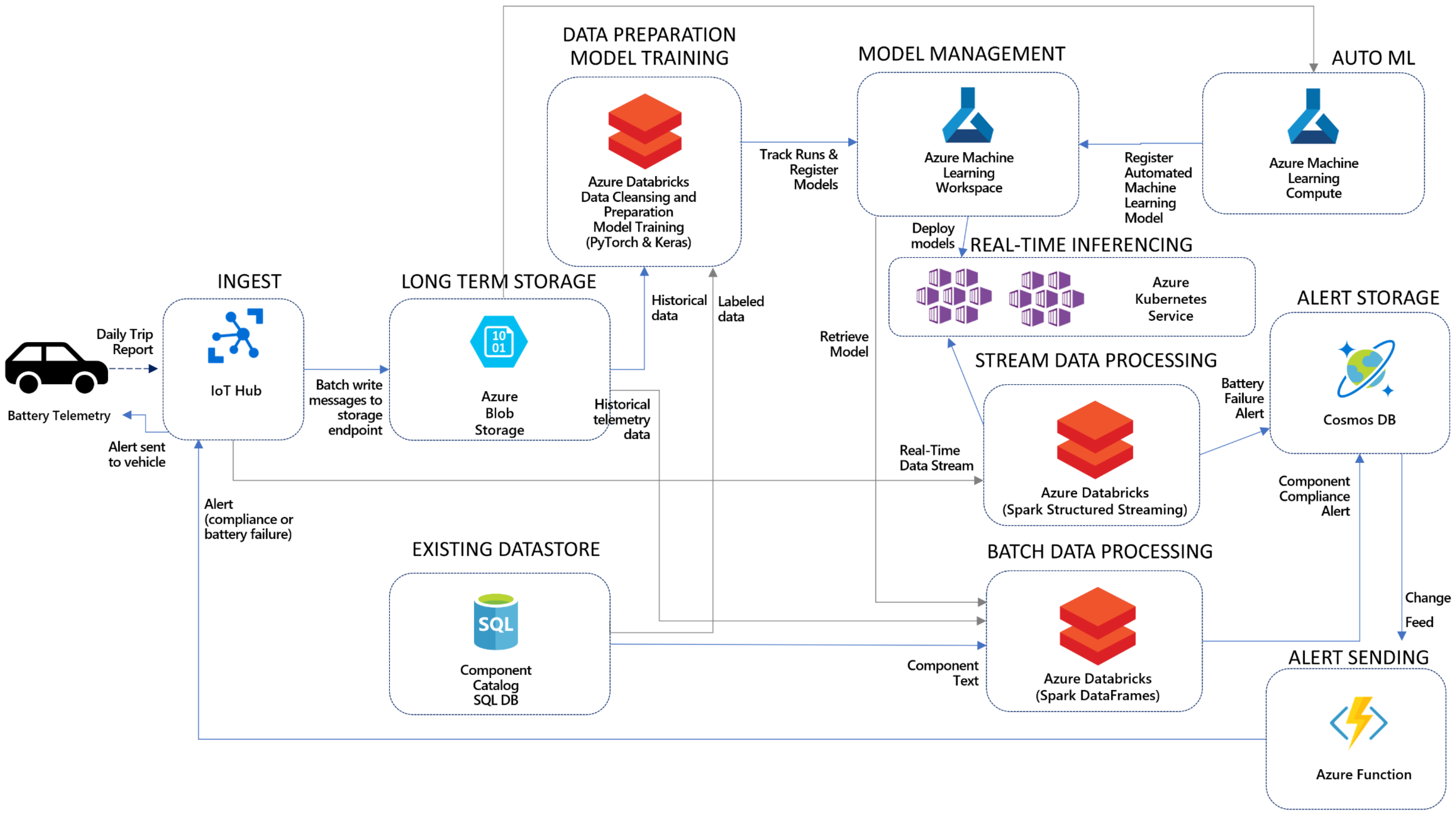


PREFERRED TARGET AUDIENCE

Francine Fisher, CIO of Trey Research

The primary audience is the business decision makers and technology decision makers. From the case study scenario, this would include the Director of Analytics. Usually we talk to the infrastructure managers. Who report to the chief information officers (CIOs), or to application sponsors (like vice president [VP] line of business [LOB], or chief marketing officer [CMO]), or to those that represent the business unit IT or developers that report to application sponsors.

PREFERRED SOLUTION



PREFERRED OBJECTIONS HANDLING

1. **Should we use machine learning or deep learning approaches?**

For Trey's two scenarios, they could actually use either approach. They would likely want to try both approaches and determine which yields the best performance in terms of model training time, inferencing time, and inferencing performance (e.g., accuracy).

PREFERRED OBJECTIONS HANDLING

2. **How should we choose between Keras and PyTorch for performing deep learning?**

This is the subject of much discussion in the community, however the guidance for selecting between Keras and PyTorch boils down to: Keras may be easier to start with and easier to build production grade models, while PyTorch has a steeper initial learning, as it is lower level than Keras, but it offers greater flexibility, faster inferencing and improved debuggability in the balance.

For a comprehensive comparison, see https://deepsense.ai/keras-or-pytorch/.

PREFERRED OBJECTIONS HANDLING

3. **We have heard Azure Machine Learning supports automated machine learning, can we use automated machine learning to create models using deep learning? Can we really expect a non-data scientist to create performant models using these tools?**

Automated machine learning in Azure Machine Learning helps to simplify and expedite the process of producing a performant model. It does this by trying many combinations of best practice data preparation (automated pre-processing and featurization), algorithm selection and algorithm parameters (hyper-parameter tuning) while asking the user only for some relatively simple configuration information (such as the type of prediction problem, the input training data set, the feature to predict and the compute resources on which to experiment) to perform the job.

Azure Machine Learning provides access to the automated machine learning capabilities via a Python SDK and via visual interface in the Azure Machine Learning studio. The latter user experience can simplify the setup enough such that a non-data scientist who has an understanding of the fundamentals of training a model can use it to create a model.

PREFERRED OBJECTIONS HANDLING

4. **Some of our team has worked with Azure Databricks, and they are confused by the overlap with Azure Machine Learning. How should we be thinking about when to use which?**

Consider using both. The best way to think about the relationship between Azure Databricks and Azure Machine Learning is that Azure Databricks provides the tools for data engineers and data scientists to author their data and machine learning pipelines as well as the compute that powers these, and Azure Machine Learning provides the platform that formalizes the modeling process by capturing data about training runs, versioning pipelines and models and assisting with the deployment of models as web services.

PREFERRED OBJECTIONS HANDLING

5. **We have heard a lot about how complicated and opaque trained deep learning models are. How is it even possible to attempt to explain them?**

The approach taken under the covers by Azure Machine Learning Python SDK is effectively black box testing of a model- it takes an input sample, uses the model to make the prediction and evaluates the outcome using a variety of techniques (called explainers). As such it is agnostic to whether the model is machine learning based or deep learning based. An example of such an explainer is the Mimic Explainer (also called a Global Surrogate). The Mimic Explainer is based on the idea of training global surrogate models (https://christophm.github.io/interpretable-ml-book/global.html) to mimic blackbox models. A global surrogate model is an intrinsically interpretable model that is trained to approximate the predictions of any black box model as accurately as possible. Data scientists can interpret the surrogate model to draw conclusions about the black box model. You can use one of the following interpretable models as your surrogate model: LightGBM (LGBMExplainableModel), Linear Regression (LinearExplainableModel), Stochastic Gradient Descent explainable model (SGDExplainableModel), and Decision Tree (DecisionTreeExplainableModel).

Using the classes and methods in the SDK, you can:

Explain model prediction by generating feature importance values for the entire model and/or individual datapoints.
Achieve model interpretability on real-world datasets at scale, during training and inference.
Use an interactive visualization dashboard to discover patterns in data and explanations at training time
The azureml-interpret SDK module uses the interpretability techniques developed in Interpret-Community, an open source python package for training interpretable models and helping to explain blackbox AI systems.

CUSTOMER QUOTE

"We are excited by how Azure enables us with a comprehensive platform for performing machine learning and deep learning without complicating big data analytics."

- Francine Fischer, CIO of Trey Research

HANDS-ON LAB


COFFEE
BREAK
12:00-1.00PM

THANK YOU

CONTACT

Eng Teong Cheah
etcheah@outlook.com