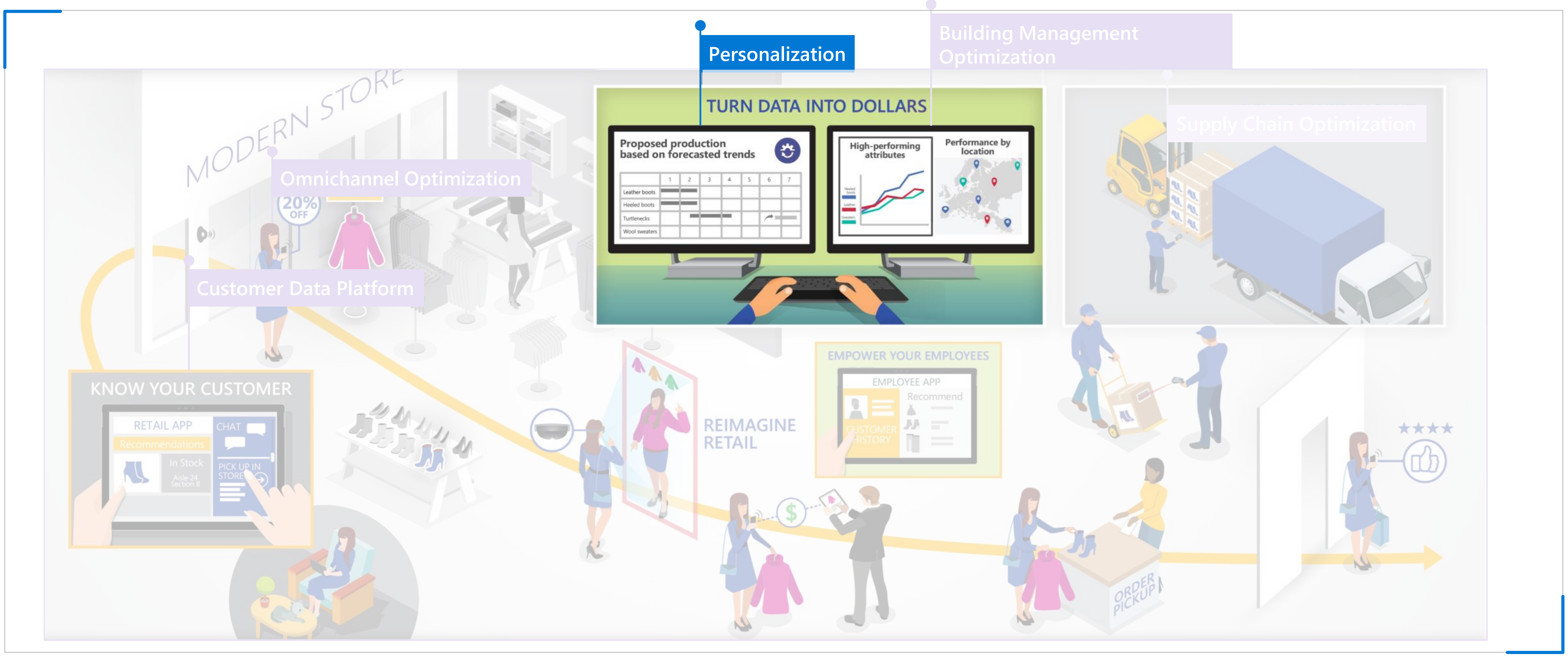
Microsoft Azure

Retail Recommender Solution Accelerator

POWERED BY: EARLY ACCESS ENGINEERING

Customer expectations intensify as online retail accelerates



Today's customers demand personalized retail experiences

Empower your employees

62% of consumers say friendly and/or **knowledgeable employees are most important aspect** of in-store customer service.[1]

Embrace the new normal

75% of consumers have tried a **new shopping experience** or approach since COVID and intend to continue.[3]

DATA

Reimagine retail

650M AI-native shoppers (ages 5-9) are expected to give up product research in favor of **relying on AI to provide best offerings** preselected for them.[2]

Know your customer

78% of consumers are more likely to shop at retailers and brands providing a **personalized experience.**[4]

1) [The ICSC Customer Service Survey. ICSC, 2019]
2) [5 predictions for the future of retail | Retail Dive]
3) [McKinsey & Company, The Great Consumer Shift]
4) [Why Personalized Retail is the Future of Brick-and-Mortar Stores | Traf-Sys]

Data is the foundation of successful personalization in retail

Behavior and purchasing data (online and in person) reside in different systems and **don't support personalizing the customer experience**

There are **too many disparate channels** that disrupt and silo my customers' journey

Legacy systems make two-way feedback from customers difficult

With apps in the cloud and in the datacenter, it **takes too long** to make consumer experience improvements

Departments are **operating in silos** making it difficult to respond to customer demand for fast fulfillment, flexible delivery, and easy returns

Competitors are personalizing the consumer relationship and **taking market share**

Deeper insights drive revenue and increase retention

Machine learning helps you evaluate how operational decisions factor into business outcomes.

Gain 360-degree customer insights

$4.5T

Online sales growth expected in 2021 (from $1.3 trillion in 2014)[1]

Enhance customer experience

69%

Percent of internet users prefer to look for reviews online than approach an in-store employee[2]

Improve customer acquisition

97%

Likely to repeat purchase after positive online experience[3]

1) [10 Important Ecommerce Trends to Watch for in 2021 - Oberlo]
2) [10 Mobile Usage Statistics You Should Know in 2021 [Infographic] (oberlo.com)]
3) [Avionos_B2C_Datareportv3.3.pdf]

Transform the customer experience

Retail Recommender leverages the robust analytics capabilities of Azure Synapse to **evaluate past purchasing and browsing habits** to deliver a **tailored browsing experience** with personalized product recommendations.

Reward loyal customers with meaningful product recommendations

Drive analytics end-to-end across the supply chain

Build better in-store and online **experiences**

TSC

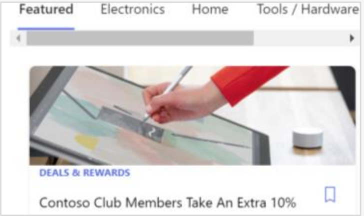
Enable deeper insights and more personalized customer experience

Use machine learning to make personalized recommendations that are meaningful to loyalty customers

Shopping recommendations made in real-time

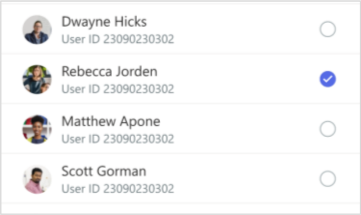
1

Customer lands on page, sees personalized items based on their purchase history.



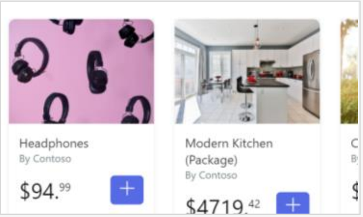
2

Customer selects or switches user.



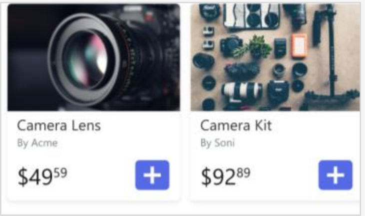
3

Landing page refreshes, customer sees personalized items in real-time based on their purchase history.



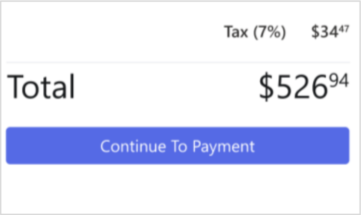
4

Customer enters product detail page, sees suggested products based on purchase and browsing history.



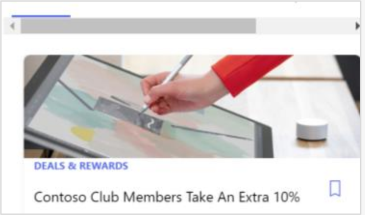
5

Customer puts items in cart, sees additional recommendations based on items they've put in the cart.



6

Customer returns to the landing page, sees recommended add-on products.



● = Azure Synapse guides consumer within shopping experience

Starbucks success

Everything we do in technology is centered around the customer connection in the store, the human connection, one person, one cup, one neighborhood at a time.

GERRI MARTIN-FLICKING
Executive Vice President/CTO

Read full story here

SITUATION	Starbucks needed to better understand guests using online channels and be able to activate insights to drive appropriate personalized recommendations that incorporate buyer behavior, store location, weather, and other data.
SOLUTION	Starbucks used Azure and machine learning to display recommendations based on reinforcement learning, providing a more personalized experience to Starbucks mobile app users.
IMPACT	16 million active Starbucks Rewards members now receive thoughtful recommendations from the app for food and drinks based on local store inventory, popular selections, weather, time of day, community preferences, and previous orders.

TSC success

...our analytics platform will deliver insights that help us better understand our customers, while offering products and services that truly meet their needs.

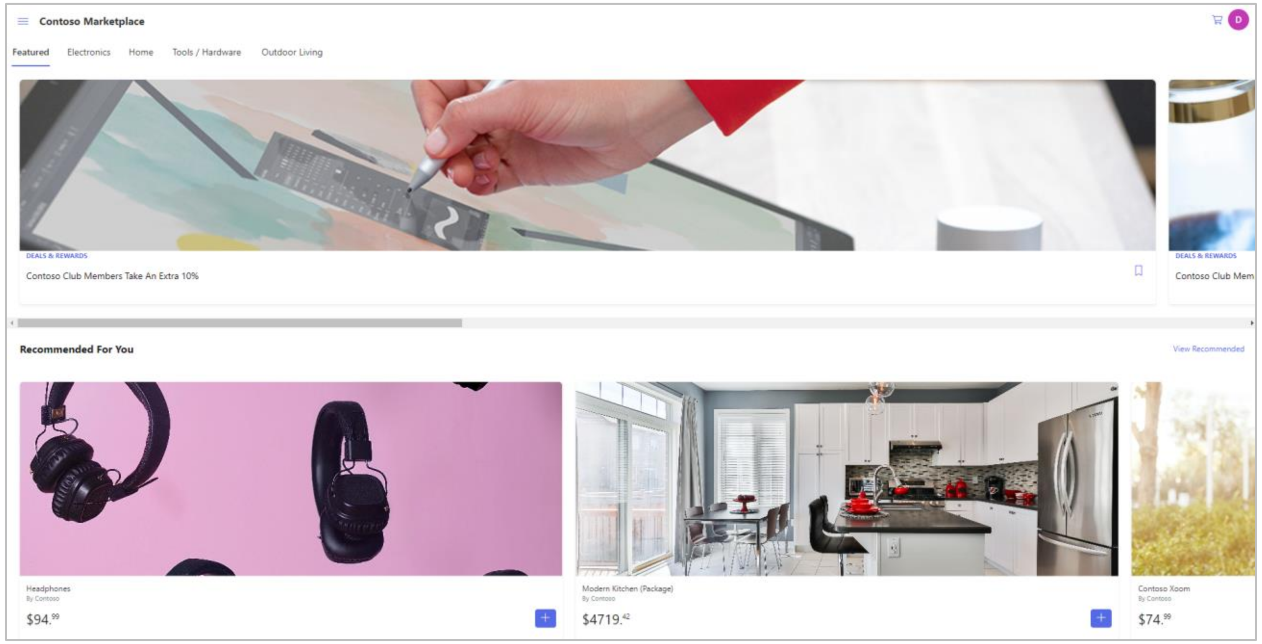
ROB MILLS
Chief Digital Commerce & Strategy Officer
Executive Vice President

TSC TRACTOR SUPPLY CO

Read full story here

SITUATION	Tractor Supply Company (TSC) needed a platform to respond to market changes and evolving customer needs. They wanted to deliver personalized, convenient shopping experiences anytime, anywhere.
SOLUTION	TSC chose Microsoft Azure to upgrade their e-commerce platform and to access insights that will enable deeper personalization.
IMPACT	Azure analytics enabled deeper personalization to tailor the customer's shopping experience and created business intelligence to drive enterprise-wide analytics that support TSC's ONETractor strategy.

Retail Recommender demo



[Retail Web Site]

Next steps: Accelerate your journey

Kick-off	Proof of value	MVP & deployment
Learn more about the Demand Forecasting Solution Accelerator and see a demo	Solution code walk-through and prototype creation for customer testing	Minimum Viable Product (MVP) is built and deployed for the customer with support of the technical specialists (CSA) and Partners
30 minutes	3 Hours or less	2-5 weeks

Deliver powerful insights for fast ROI

Azure gives you critical information and data, secured with the most advanced features in the market.

Easily launch using prebuilt code from GitHub to gain quick time to value and focus on other initiatives

Creates a **single pane of glass** for ingestion, orchestration, ML, and Power BI

Gain insights from all your data across all your data warehouses with blazing speed

Reduce project time with unified, end-to-end analytics

Achieve real-time personalization while users browse

Open & Login

The customer opens the app and logs in

Obtain Key Data

Get the user profile and associated transactional data

Personalize the page

Display purchase History, Favorites, Recommendations, etc.

Recommendations

Level 1: Recommendations based on Purchase History

Level 2: Recommendations based on Selected products

Level 3: Recommendations featuring Promotions

Complete Purchase

The customer completes their purchase thanks to the personalized experience

All in real time, while shopping

Next steps in your personalized MVP deployment

To begin creation of your MVP, follow these steps:

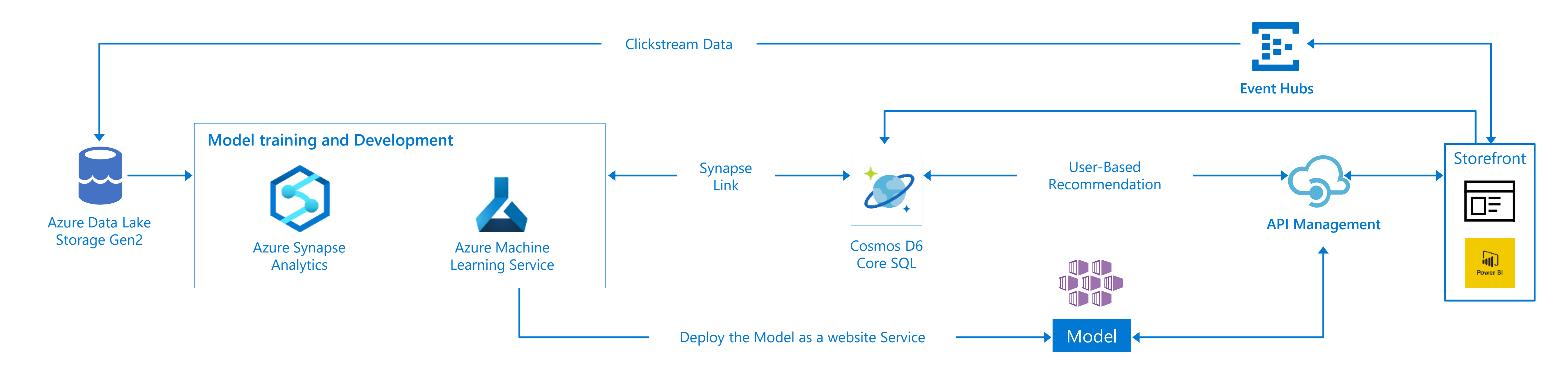
Deploy resources	Prepare data	Train with custom script
Create a resource group you can use as a container to begin deploying the resources to Azure	Configure your development environment for Azure Machine Learning. Notebook VM is a highly-compatible, preconfigured way to get up and running quickly	The scripting process is outlined in a series of Jupyter Notebooks. Review them sequentially to walk you through training, scoring, and making predictions using Azure Machine Learning

With demo data, create your MVP in less than a day

With your custom data and a prep session, create your MVP in 2-5 weeks

Thank you

Architecture

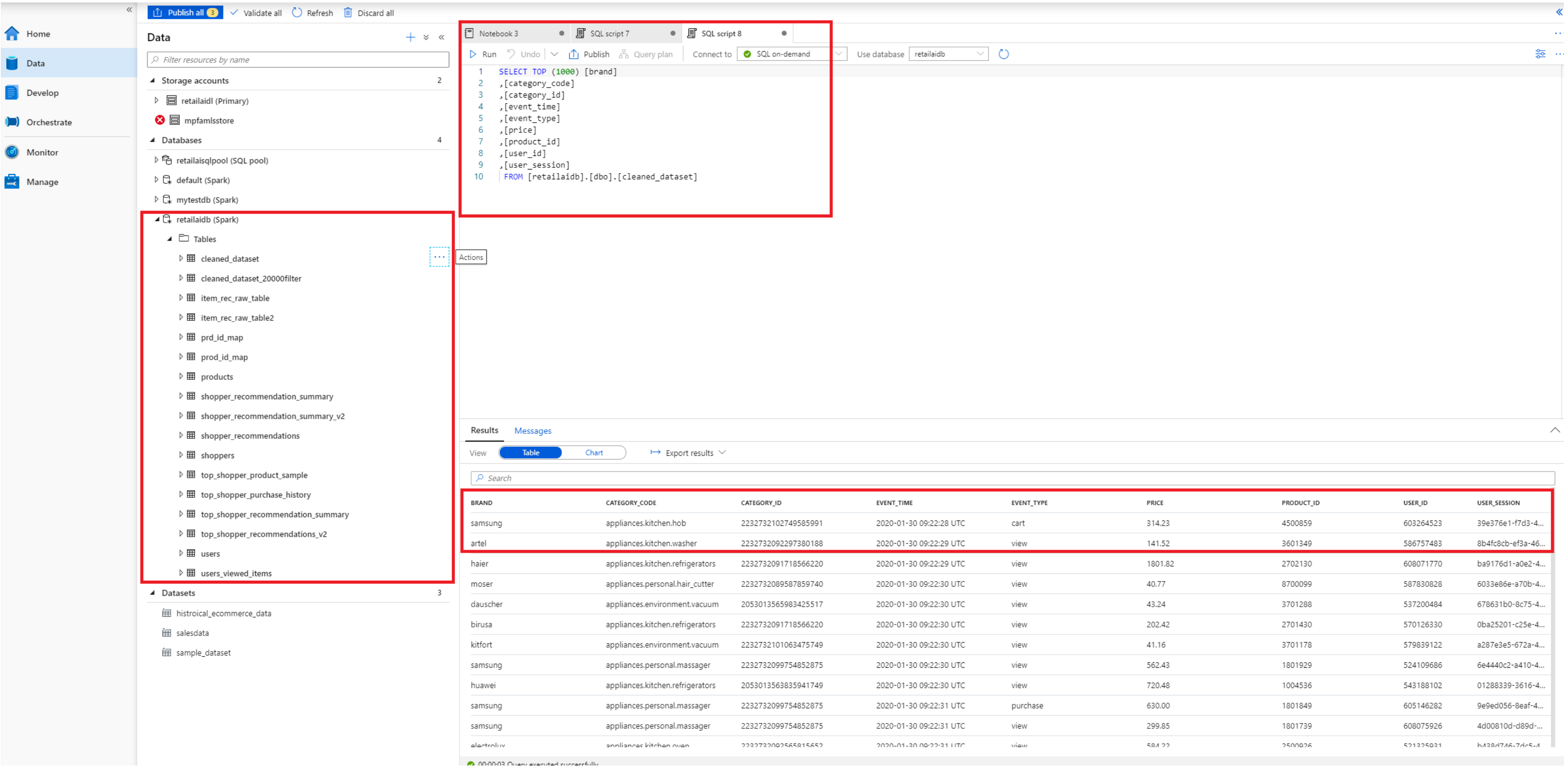


Data Exploration, Model Testing and Training, and other Data Asset Preparation takes place in Azure Synapse Analytics, Features Utilized

- Spark Pools
- Notebooks for Spark Development
- SQL Pools
- External Tables from ADLS
- sqlanalytics() connector for Spark

Azure Machine Learning Service is used to deploy the item-based recommender to AKS as a Web Service

Dataset



Publish all 3
Validate all
Refresh
Discard all
Home
Data
Develop
Orchestrate
Monitor
Manage
Data
Filter resources by name
Storage accounts 2
retailaidl (Primary)
mpfamlsstore
Databases 4
retailaisqlpool (SQL pool)
default (Spark)
mytestdb (Spark)
retailaidb (Spark)
Tables
cleaned_dataset
cleaned_dataset_20000filter
item_rec_raw_table
item_rec_raw_table2
prd_id_map
prod_id_map
products
shopper_recommendation_summary
shopper_recommendation_summary_v2
shopper_recommendations
shoppers
top_shopper_product_sample
top_shopper_purchase_history
top_shopper_recommendation_summary
top_shopper_recommendations_v2
users
users_viewed_items
Datasets 3
histroical_ecommerce_data
salesdata
sample_dataset
Actions
Notebook 3
SQL script 7
SQL script 8
Run
Undo
Publish
Query plan
Connect to
SQL on-demand
Use database
retailaidb
SELECT TOP (1000) [brand]
,[category_code]
,[category_id]
,[event_time]
,[event_type]
,[price]
,[product_id]
,[user_id]
,[user_session]
FROM [retailaidb].[dbo].[cleaned_dataset]
Results
Messages
View
Table
Chart
Export results
Search
BRAND CATEGORY_CODE CATEGORY_ID EVENT_TIME EVENT_TYPE PRICE PRODUCT_ID USER_ID USER_SESSION
samsung appliances.kitchen.hob 2232732102749585991 2020-01-30 09:22:28 UTC cart 314.23 4500859 603264523 39e376e1-f7d3-4...
artel appliances.kitchen.washer 2232732092297380188 2020-01-30 09:22:29 UTC view 141.52 3601349 586757483 8b4fc8cb-ef3a-46...
haier appliances.kitchen.refrigerators 2232732091718566220 2020-01-30 09:22:29 UTC view 1801.82 2702130 608071770 ba9176d1-a0e2-4...
moser appliances.personal.hair_cutter 2232732089587859740 2020-01-30 09:22:30 UTC view 40.77 8700099 587830828 6033e86e-a70b-4...
dauscher appliances.environment.vacuum 2053013565983425517 2020-01-30 09:22:30 UTC view 43.24 3701288 537200484 678631b0-8c75-4...
birusa appliances.kitchen.refrigerators 2232732091718566220 2020-01-30 09:22:30 UTC view 202.42 2701430 570126330 0ba25201-c25e-4...
kitfort appliances.environment.vacuum 2232732101063475749 2020-01-30 09:22:30 UTC view 41.16 3701178 579839122 a287e3e5-672a-4...
samsung appliances.personal.massager 2232732099754852875 2020-01-30 09:22:30 UTC view 562.43 1801929 524109686 6e4440c2-a410-4...
huawei appliances.kitchen.refrigerators 2053013563835941749 2020-01-30 09:22:30 UTC view 720.48 1004536 543188102 01288339-3616-4...
samsung appliances.personal.massager 2232732099754852875 2020-01-30 09:22:31 UTC purchase 630.00 1801849 605146282 9e9ed056-8eaf-4...
samsung appliances.personal.massager 2232732099754852875 2020-01-30 09:22:31 UTC view 299.85 1801739 608075926 4d00810d-d89d-...

Model Development

Azure Synapse Analytics
Spark

Cell 4

```
[5]
model_reload = ALSModel.load("retailai_recommendation_model_v2")
labels = download_file_from_adls("retailaidl", ···
```

Cell 5

```
[29]
get_top_users = spark.read.table("retailaidb.cleaned_dataset").groupBy(["user_id", "user_session"]).count().groupBy(["user_id"]).count().orderBy("count", ascending=False)
```

Cell 6

```
[30]
top_user_ids = get_top_users.select("user_id").limit(10)
```

Cell 7

```
[31]
label_converter = IndexToString(inputCol="product_id_new", outputCol="product_id", labels=labels)
```

Cell 8

```
[32]
preds_alt = model_reload.recommendForUserSubset(top_user_ids, 5).select("user_id", col("recommendations"))
```

Cell 9

```
[33]
convert_labels_df = label_converter.transform(preds_alt.select("user_id",explode("recommendations").alias("recommendations")) \
                    .select("user_id", "recommendations.product_id_new", "recommendations.rating")).drop("product_id_new").orderBy("rating", ascending=False)
```

Cell 10

```
[34]
products = spark.read.format("delta").table("retailaidb.products").withColumnRenamed("product_id", "prd_id")
adj_products = products.where("category_code != 'None'").withColumn("product_name", (element_at(split(col("category_code"), "\."),-1)))
product_columns = adj_products.columns
```

Cell 12

```
[41]
prd_sample = get_prod_data_df.select("product_id", concat("brand", lit(" "), "product_name").alias("product_name")).distinct()
```

Cell 13

```
[42]
prd_sample.write.saveAsTable("retailaidb.top_shopper_product_sample", mode="overwrite")
```

Cell 14

```
[38]
cols_to_collect = get_prod_data_df.columns
cols_to_collect.remove("user_id")
```

Cell 15

```
[39]
final_recommendations = get_prod_data_df.select("user_id", concat_ws(",", *cols_to_collect).alias("recommendations")) \
                        .groupBy("user_id").agg(collect_set("recommendations").alias("recommendations")) \
                        .select("user_id", col("recommendations").cast("string"))
```

Cell 16

```
[40]
final_recommendations.withColumn("updated_ts", lit(current_timestamp())).write.saveAsTable("retailaidb.top_shopper_recommendation_summary", mode="overwrite")
```

User based recommendation model – using Spark's native Collaborative Filtering algorithm

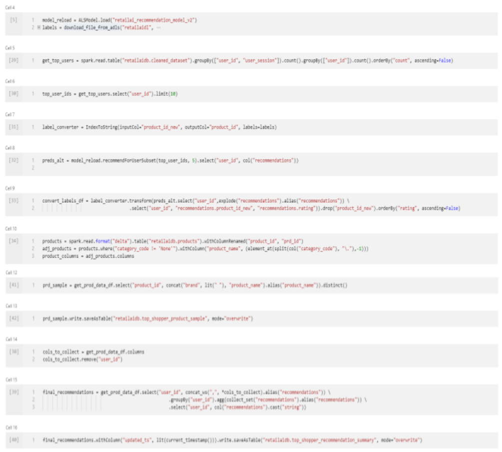
Azure Synapse Analytics
Spark

```
# partly borrowed from https://github.com/akshitvjain/item-based-recommender
from pyspark.sql import Row
from pyspark.sql.functions import col
import numpy as np
from numpy import linalg as LA

from pyspark.ml.recommendation import ALSModel
model_reload = ALSModel.load("retailai_recommendation_model")


class ItemBasedRecommender():

    def __init__(self, model, spark):
        self.model = model
        self.spark = spark
        self.itemFactors = self.model.itemFactors

    def compute_similarity(self, item_id):
        item = self.itemFactors.where(
            col('id') == item_id).select(col('features'))
        item_features = item.rdd.map(lambda x: x.features).first()

        lol = []
        for row in self.itemFactors.rdd.toLocalIterator():
            _id = row.__getattr__('id')
            features = row.__getattr__('features')
            similarity_score = self._cosine_similarity(features, item_features)
            if _id != item_id:
                lol.append([_id, similarity_score])

        R = Row('item_index', 'similarity_score')
        self.similar_items_df = self.spark.createDataFrame(
            [R(col[0], float(col[1])) for col in lol])
        self.similar_items_df = self.similar_items_df.orderBy(
            col('similarity_score').desc()).na.drop()
        return self.similar_items_df


    def _cosine_similarity(self, vector_1, vector_2):
        v1 = np.asarray(vector_1)
        v2 = np.asarray(vector_2)
        cs = v1.dot(v2) / (LA.norm(v1) * LA.norm(v2))
        return(cs)

# item id as input. Note: this is the normalized item id starting from 0.
test_id = 100
item_based_rec = ItemBasedRecommender(model_reload, spark)
ret_df = item_based_rec.compute_similarity(test_id)
ret_df.show(5)
```

Item based recommendation, based on cosine distance and the trained user-based recommendation model

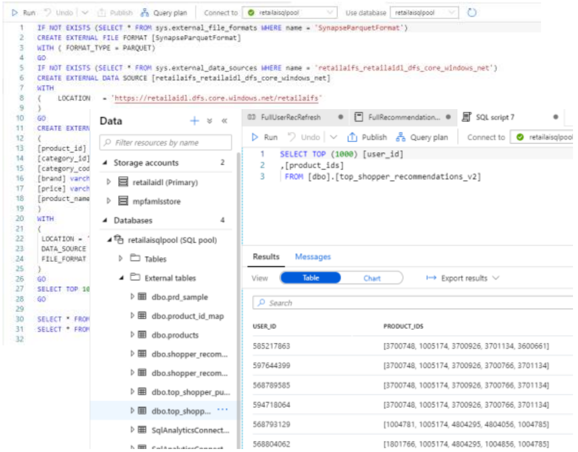
Model Deployment (User-Based Recommendations)

Azure Synapse Analytics
Spark



Using the Spark Pools on Azure Synapse Analytics, the model trained by our Data Scientists was used to generate user-based recommendations

Azure Synapse Analytics
SQL Analytics

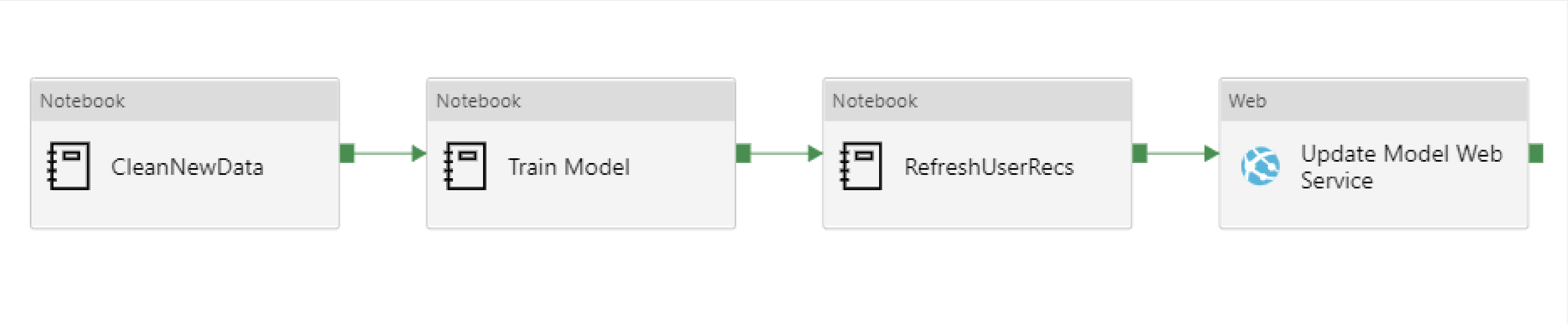


To serve these recommendations, they were stored in the attached Data Lake Storage and served in an External Table using the SQL Pool in our Synapse workspace

Azure Synapse Analytics
SQL On-Demand

Using the integration between Azure Synapse Analytics and Azure Cognitive Search, the user recommendations were made accessible to the front-end e-commerce platform

Retraining Models in Synapse



1. Another important piece of any ML Lifecycle is retraining the models to adapt to new data and to improve the accuracy of the models over time.

2. We accomplish this in Azure Synapse Analytics by using a Pipeline to orchestrate the execution of three notebooks that clean new data, retrain the model, and then refresh the user-based recommendations.

3. Lastly, we trigger a process to rollout the new model object to the web service so our live recommendation service can utilize the updated model.