Microsoft Azure

AI in Action: Five Case Studies

Examples of innovation with computer vision, intelligent chatbots, personalization, cognitive search, and forecasting, with code samples to kickstart your projects

Contents

Introduction

Companies around the world are using AI to build apps to automate complex tasks and derive new knowledge from large sets of business data. Whether it's using computer vision to improve safety, personalizing content for users, or offering customers more options to interact, machine learning and artificial intelligence technologies can drive your business in new ways. This collection of technical case studies highlights how development teams are using Microsoft Azure AI to create applications that allow companies to deliver information more quickly to customers and enable them to take action.

Before you start

Interested in trying code from these case studies? Sign up for an Azure free account.

1.
Computer Vision in Action

Computer vision uses advanced algorithms to process the information in images, performing such tasks as identifying objects in a picture, tracking movements in a video, or recognizing people in a crowd. This technology can also be used to label the content of an image to, for example, block inappropriate images or prevent the communication of intellectual property.

Shell sees safety in Azure's Computer Vision

Royal Dutch Shell, usually known as Shell, manages everything from oil wells to retail gas stations. The global company is a leading retailer of transportation products and services, serving 30 million customers a day across a network of 44,000 stations in more than 75 countries.

Shell has begun using computer vision to automate safety checks at its service stations. For example, if a customer pulls into a station, starts filling up their car with gas, and then lights a cigarette to smoke, cameras and machine-vision systems can detect the dangerous act and notify the station manager. The manager can disable the pump until the customer puts out the cigarette.

- *Solution and technology*

  With its tens of thousands of stations all over the world, the company relies on innovative technology to detect risks and ensure the safety of its consumers and business. While the company has a network of "service champions" to catch dangerous behavior at its retail sites, Shell aims to minimize the amount of effort needed to monitor the safety of its employees and customers.

  Shell worked with Microsoft to create a project known as Video Analytics for Downstream Retail (VADR), which uses machine vision combined with automated image processing to detect dangerous behavior and alert service champions. The company used the OpenCV (open source computer vision library) and Azure Databricks to implement its solution, underscoring how Azure can handle both generic solutions as well as custom implementations.

  "Things like smoking pose really big risks," says Daniel Jeavons, general manager of data science at Shell. "We want to use digital technology to be able to respond quickly and correctly to prevent dangerous scenarios. We consider that not just good business but part of our responsibility as a member of the communities we're part of."

  Shell is running a pilot deployment of VADR at retail stations in Thailand and Singapore, with plans to expand the project once those installations have proven successful. Its experience with machine vision and AI has sparked a keen interest in further AI projects within Shell, which has more than 280 machine learning and AI projects underway, and the company is starting a residency program for students who want to do AI work at Shell.

AB InBev tracks its supply chain

AB InBev relies heavily on its supply chain in its production of a variety of beverages—from Budweiser to Wäls Brut Belgian ale. The company uses computer vision to help gather information and automate a variety of tasks to keep its supply chain moving and on schedule. In March 2020, the company launched an effort to gather data on crop yields, aiming to help its suppliers predict harvests. In 2019, the company began using autonomous drones to take inventory in stores and warehouses, tracking to avoid the possibility of a lack of inventory of needed goods and supplies.

Solution and technology

AB InBev has built its digital infrastructure on Microsoft Azure, with data analytics and machine learning capabilities provided by Azure AI. For example, Smart Barley—an effort to help farmers improve their yields—uses computer vision as a component to reduce water and fertilizer usage and create a much more sustainable agricultural process. The company also uses AI services throughout its supply chain, from customer-facing chatbots to automating and simplifying everyday back-office tasks to save time. Azure also enables AB InBev to integrate more customized models with services like Databricks and OpenCV.

Implementing Computer Vision with Azure AI

These companies used one or more of the following technologies:

- Azure Computer Vision API

- Azure Custom Vision API

- Azure IoT Edge

- Azure IoT Hub

- Apache Kafka

- Apache Spark

- Azure Databricks

- OpenCV

Technical deliverable: Deploying onsite using Azure IoT Edge

Shell, for example, deployed Azure IoT Edge so that its application logic would be as close as possible to the situations it analyzes, further reducing the time the system needs to make vital safety decisions. "We have a set of containerized flows that include analytic models," says Jeavons. "We push them to the edge using Apache Kafka streaming—which is really good for managing large datasets—and the closed-circuit TV camera footage passes through the models, which extract the frames we need to identify potential safety risks."

To create a Windows container to act as an IoT Edge device, use:

```
az vm create --resource-group IoTEdgeResources --name EdgeVM --image
MicrosoftWindowsDesktop:Windows-10:rs5-pro:latest --admin-username
azureuser --admin-password {password} --size Standard_DS1_v2
```

To create an Ubuntu Linux container to act as an IoT Edge device, use:

```
az vm image accept-terms --urn microsoft_iot_edge:iot_edge_vm_
ubuntu:ubuntu_1604_edgeruntimeonly:latest

az vm create --resource-group IoTEdgeResources --name EdgeVM --image
microsoft_iot_edge:iot_edge_vm_ubuntu:ubuntu_1604_edgeruntimeonly:latest
--admin-username azureuser --generate-ssh-keys
```

Create a hub with:

```
az iot hub create --resource-group IoTEdgeResources --name {hub_name}
--sku F1
```

Then register the IoT Edge device with the hub:

```
az iot hub device-identity create --hub-name {hub_name} --device-id
myEdgeDevice --edge-enabled

az iot hub device-identity show-connection-string --device-id
myEdgeDevice --hub-name {hub_name}
```

The IoT Edge device is now ready to be configured and modules can be pushed down to the device.

Find more examples and code for Linux and Windows.

Technical deliverable: Recognizing dangerous activity using the Computer Vision API

VADR uploads video frames identified as showing potentially dangerous behavior to a custom OpenCV application that Microsoft Azure analyzes via more sophisticated deep learning AI models. The Azure Computer Vision REST API analyzes frames from video images, by themselves or in conjunction with keyframe selection, using the Azure Media Service Video Indexer. Developers need Visual Studio 2015 (or later) and the computer vision capabilities of Azure Cognitive Services (available with an Azure free account).

Developers can send an image for analysis using the Computer Vision API and, in the example below, Python. A more complete tutorial, using other languages, can be found in this quickstart.

```python
computervision_client = ComputerVisionClient(endpoint,
CognitiveServicesCredentials(subscription_key))

'''
Tag an Image - remote
This example returns a tag (key word) for each thing in the image.
'''
print("===== Tag an image - remote =====")
# Call API with remote image
tags_result_remote = computervision_client.tag_image(remote_image_url )

# Print results with confidence score
print("Tags in the remote image: ")
if (len(tags_result_remote.tags) == 0):
    print("No tags detected.")
else:
    for tag in tags_result_remote.tags:
        print("'{}' with confidence {:.2f}%".format(tag.name, tag.
confidence * 100))
# </snippet_tags>
print()
'''
```

Technical deliverable: Recognizing objects in images

Using drones—whether over fields in an agricultural setting or above shelves in a store—enables the automated capture of images and video from a variety of locations and angles. The content captured by drones can be triaged by algorithms on the device itself (much like the Azure IoT scenario used by Shell), with potential matches uploaded to the cloud for more detailed analysis.

As demonstrated by AB InBev, tracking products on shelves or yields from planted fields can both be done using similar technology. Azure's Computer Vision has built-in categories to search for features including faces, colors, and other content classifications, and using the Custom Vision API, custom classification models can also be built for specific applications. The C# code below shows how simple it is to call the Computer Vision API to extract features, assign categories, and tag images. Extracted features could include the number or quantity of product containers or boxes on a shelf or whether a field is fully planted, for example, although customer applications would require specific trained models.

```
public static async Task AnalyzeImageUrl(ComputerVisionClient
client, string imageUrl)
{
    // Creating a list that defines the features to be extracted
from the image.
    List<VisualFeatureTypes> features = new
List<VisualFeatureTypes>()
    {
        VisualFeatureTypes.Categories, VisualFeatureTypes.
Description,
        VisualFeatureTypes.Faces, VisualFeatureTypes.ImageType,
        VisualFeatureTypes.Tags, VisualFeatureTypes.Adult,
        VisualFeatureTypes.Color, VisualFeatureTypes.Brands,
        VisualFeatureTypes.Objects
    }
};
```

Azure Computer Vision API

The Computer Vision service allows any code to easily take advantage of premade and advanced features running on Azure. Using a software development kit (SDK) built on top of REST APIs, developers can make a simple function call to read text, classify an image, determine if a trademark or celebrity is in a photo, and tag an image with automatically generated keywords. More advanced features can be built using the SDKs available in .NET, Python, JavaScript, and Java and provide additional functionality. Extracted features could include the number or quantity of product containers or boxes on a shelf or whether a field is fully planted, for example, although customer applications would require specific trained models.

Conclusion

Computer vision can help companies better automate aspects of their business that, until now, have been resistant to automation and data acquisition.

Most of the activities across Shell operations, for example, involve the management and control of physical assets—retail stations, tankers, lube plants, pipelines, refineries, and oil and gas platforms—so the VADR project and machine vision technologies have a wide range of potential applications within the company. These could include automated alerts for systems on remote offshore platforms in case of dangerous conditions or the monitoring of vital assets, such as pipelines and wells, to prevent corrosion.

AB InBev has focused on the automated management of its supply chain, using computer vision to analyze crop yields and to track the inventory of products on store shelves.

Resources

Create your Azure free account

What is Computer Vision?

Azure Computer Vision API

How AI is building better gas stations and transforming Shell's global energy business

From seed to sip: How Anheuser-Busch InBev uses AI to drive growth

Quickstart: How to build a classifier with Custom Vision

2.
Intelligent Chatbots in Action

Chatbots offer the ability for organizations to interact with customers and users in a natural way across multiple channels and experiences such as company websites, mobile apps and devices, and smart-speaker ecosystems. This technology also allows the added bonus of translation—the ability to communicate the same information to users no matter what language they speak—which is especially helpful in entertainment.

La Liga connects with soccer fans

La Liga, Spain's top men's professional football league, has almost 1.6 billion social media followers around the world. The league's priority is to stay connected with these fans across various channels such as Skype and the Google Assistant ecosystem.

To keep the league's supporters engaged, La Liga created a virtual assistant to answer questions from its fans in both English and Spanish, despite the various nuances in terms across the multiple dialects used in Spanish-speaking countries. People can ask questions of the virtual assistant using natural language, and the bot will parse the request to determine intent.

Solution and technology

Developers for La Liga created the assistant using Azure Bot Service, the Microsoft Bot Framework, and a variety of Azure Cognitive Services. The assistant takes advantage of the scalability and flexibility of Azure App Service to both add reliability and streamline development. With the virtual assistant, La Liga deployed to the Azure App Service for the main bot logic and a variety of microservices, such as the image-processing service, Google Assistant connector, Alexa connector, data loaders, cache management, and two Azure functions for live data and proactive messages. It even uses the Azure Content Moderator to issue a yellow card to fans who use inappropriate language with the bot.

The development team for the assistant focused on scalability, and for live queries such as match updates, caches the data to quickly respond to other requests for the same information. The football league also has a developer dedicated to scalability testing, automating millions of simulated queries a day to help prepare for a smooth launch.

### *Implementing an intelligent chatbot with Azure AI*

This company used one or more of the following technologies:

- Azure Bot Service

- Azure Cognitive Services

  - Language Understanding

  - Content Moderator

  - Text Analytics

  - Translator

  - Custom Vision

- Bing Spell Check API

- Azure Cognitive Search

- Azure Cosmos DB

- Bing Images Search API and Custom Vision API

- Azure Functions

- Azure Storage

- Power BI

Technical deliverable: Implementing basic chatbot functionality

Creating a chatbot on Microsoft Azure is simple and can be done in less than an hour through the Azure Bot Service. To create a multilingual bot, developers can use a combination of the Bot Framework, Cognitive Services, and Azure along with translation middleware. Microsoft has posted a sample Multilingual Bot on GitHub.

Developers should have Visual Studio and the Bot Framework SDK installed, as well as an Azure account and a key for the Microsoft Translator Text API. An app can be created using Azure Cognitive Services Language Understanding, a core component of chatbot solutions. Using Language Understanding service, the developer can add intelligence to the conversation by matching a user's intent with a fixed number of actions and extracting the relevant detailed information.
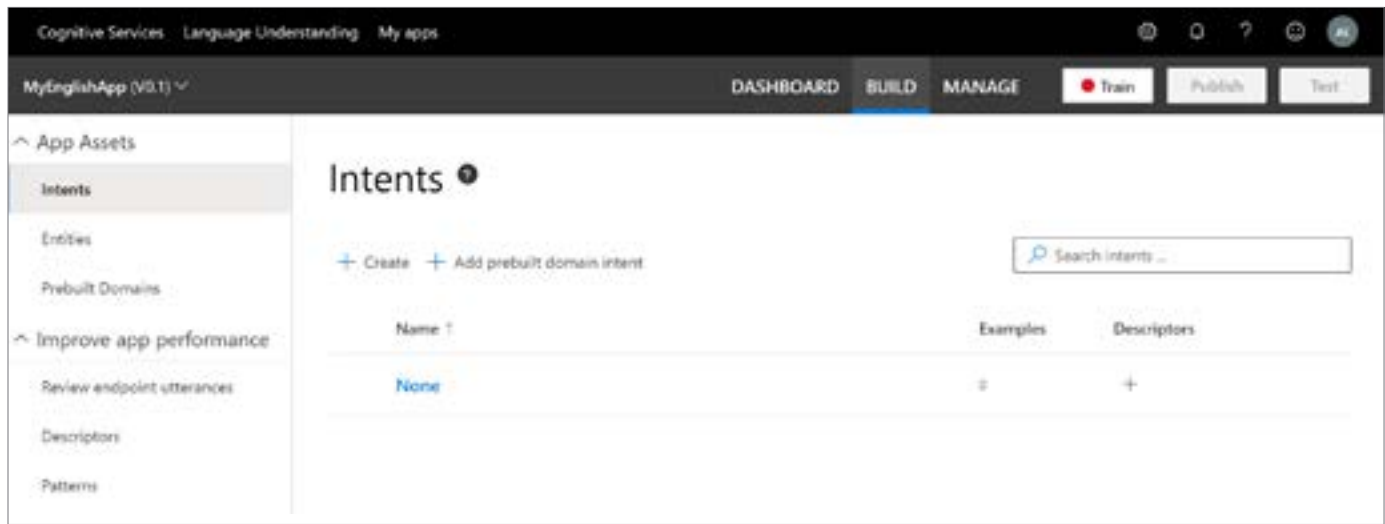
To start, use the Language Understanding service portal to create an app for conversations, assigning a name and a culture, such as English. Next, the developer must add intents—actions that Language Understanding service will use to attempt to map chat requests during use—and entities, which represent the potential objects, such as restaurants, landmarks, or particular food. Examples of utterances—potential commands—are also provided to help Language Understanding service determine mappings.

More information can be found at the Language Understanding (LUIS) authoring client library for Python Quickstart. A software development kit is also available for .NET, Java, JavaScript, and Python, and developers can also use a REST API.

Technical deliverable: Implement Azure Cognitive Search
Although Language Understanding can accurately extract intents from sentences, users often just enter words or short phrases, so developers also have the ability to use Azure Cognitive Search. With the following code snippet, a developer could handle sentences and words and reply with the correct information within the chatbot application.

```
public virtual async Task MessageReceivedAsync(IDialogContext context,
IAwaitable<IMessageActivity> result)
{
    var message = await result;
    try
    {
            SearchResult searchResult = await
                        searchService.SearchByName(message.Text);
            if(searchResult.value.Length != 0)
            {
                    CardUtil.showHeroCard(message, searchResult);
            }
            else{
                    await context.PostAsync(
                        $"No musicians by the name {message.Text} found");
            }
    }
    catch(Exception e)
    {
            Debug.WriteLine(
                $"Error when searching for musician: {e.Message}");
    }
    context.Done<object>(null);
}


            }
            else{
                    await context.PostAsync(
                        $"No musicians by the name {message.Text} found");
            }
    }
    catch(Exception e)
    {
            Debug.WriteLine(
                $"Error when searching for musician: {e.Message}");
    }
    context.Done<object>(null);
}
```

Communicating with images

Azure Cognitive Search can also use other AI services, such as the Computer Vision API (see Computer Vision in Action section above), to allow developers to analyze images, extracting keywords describing the images and potential landmarks that may be pictured in the images. Alternatively, developers can directly provide images to Azure's Bing Image Search to find similar images and "best representative queries." Bing has an enormous knowledge of images, and we can use it to recognize famous places, people, and even foods.

Technical deliverable: Create a custom greeting for the Virtual Assistant

To customize the greeting of the Virtual Assistant, developers use an Adaptive Card, an open framework that lets you describe and customize your content as you see fit. Adaptive Cards work well with bots, allowing developers to author the content for a card once and use it to render information inside multiple apps, such as a website or messaging application.

The following JSON payload can be used as a template. Note, the "url" name does not have a valid value and will have to be updated with actual URLs.

```
{
    "type": "AdaptiveCard",
    "id": "NewUserGreeting",
    "backgroundImage": **Base64 Encoded Image**,
    "body": [
        {
            "type": "Container",
            "items": [
                {
                    "type": "Image",
                    "url": **Base64 Encoded Image**,
                    "size": "Stretch"
                }
            ]
        },
```

```
        {
            "type": "Container",
            "spacing": "None",
            "backgroundImage": **Base64 Encoded Image**,
            "items": [
                {
                    "type": "TextBlock",
                    "id": "title",
                    "spacing": "Medium",
                    "size": "Large",
                    "weight": "Bolder",
                    "color": "Light",
                    "text": "Hi, I'm **your** Virtual Assistant",
                    "wrap": true
                },
                {
                    "type": "TextBlock",
                    "id": "body",
                    "size": "Medium",
                    "color": "Light",
                    "text": "Now that I'm up and running, explore the links
here to learn what I can do.",
                    "wrap": true
                }
            ]
        }
    ],
    "actions": [
        {
            "type": "Action.Submit",
            "title": "Get started",
            "data": {
                "action": "startOnboarding"
            }
        },
        {
            "type": "Action.OpenUrl",
            "title": "Documentation",
            "url": "https://aka.ms/bfvadocs"
        },
        {
            "type": "Action.OpenUrl",
            "title": "Skills",
            "url": "https://aka.ms/bfskills"
        }
    ],
    "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
    "version": "1.0",
    "speak": "Hi, I'm **your** Virtual Assistant. Now that I'm up and
running, explore the links here to learn what I can do."
}
```

Conclusion

Intelligent chatbots give companies additional ways to connect with their customers and expand opportunities for products.

La Liga predicts that its assistant will be popular for fans seeking information from the league; it expects multiple questions from average users, and therefore, scalability of the virtual assistant will be extremely important.

Resources

Create your Azure free account today

LaLiga entertains millions with Azure-based conversational AI

Multilingual Bot

Language Understanding (LUIS) authoring client library for Python Quickstart

Microsoft on the Bing Image Search API

What is a Virtual Assistant?

QnA Maker

3.
Personalizer and Recommenders in Action

Deliver the right experiences and content with Cognitive Services

When companies want to present customers with content that best fits their request, Microsoft Azure Personalizer can be used to select the best item based on up-to-date information about the customer, the context, and the application. The cloud-based service uses reinforcement learning to determine the most suitable content, which can be any unit of information such as text, images, products, or messaging that you want to select from to show to your user. Custom recommender services can also be created using Azure Machine Learning (Azure ML) to find the best matches among a large population of items.

Xbox One raises user engagement

Personalizing content for gamers remains a complex, dynamic problem. Machine learning workflows and costly A/B testing often fail to catch on to the interests of specific gamers. The Microsoft Xbox One group needed a better way to find content suited to each user to keep gamers engaged and interested in coming back.

- *Solution and technology*

  Cognitive Services Personalizer processes content and data and autonomously learns to make optimal decisions. The Xbox One group found that compared to a random personalization policy, they had a 40 percent increase in user engagement with the home page presented to gamers on the Xbox platform.

Technical deliverable: Operationalizing Personalizer

In the beginning, suggestions from the Personalizer will likely only be 20–30 percent successful, but after a number of requests, the system will improve. Every 2,000 to 10,000 requests, the developer can run an offline evaluation and apply a new learning policy to improve the Personalizer.

The basic cycle of interaction with the Personalizer is the rank-and-reward system. To get personalized recommendations based on past data, developers use the Rank API to call out to the model and get a ranked list of options.

```
# choose an action - get prediction from Personalizer
response = requests.post(personalization_rank_url, headers =
headers, params = None, json = jsonFormat)
```

To reinforce the desired responses, the developer can use the Reward API to teach the model suitable outcomes.

```
# rank actions in term of preferences
response = requests.post(personalization_reward_url, headers =
headers, params = None, json = jsonFormat)
```

ASOS recommends clothing styles

London-based ASOS, an online clothing retailer, provides shoppers with targeted recommendations, but it needed a way to consolidate three separate solutions that the company's data scientists used for its data models and allow those data scientists to collaborate more effectively. The company also needed to improve its customer recommendations modeling and overcome inefficiencies common to data modeling projects.

The clothing retailer is all about choice—offering clothing in 30 different sizes and helping customers determine the clothing that is right for their style. By applying data science to its data on customers and their preferences, the company is helping customers develop a picture of their preferred style over time. "We pivot much faster than brick-and-mortar retailers," explains Reda Kechouri, senior agile delivery manager at ASOS. "We add about 5,000 new products every week to a catalog of 85,000 items. And trend cycles are getting shorter."

Solution and technology

As an online retailer, ASOS is inherently data driven. The retailer's engineers and data scientists created a brand recommender during a four-week hackathon with Microsoft, using gigabytes of data on its 19.2 million customers' preferences and styles. The recommender is built using the Azure Machine Learning service, the Deep Learning Virtual Machine service, Azure Cosmos DB for global scalability, and other services.

The Deep Learning Virtual Machine accelerates machine learning by incorporating GPU-based instances of virtual machines for training deep learning models. "Because it's an agnostic platform, we use Azure Machine Learning service with any technologies that best fit our needs," says Naeem Khedarun, principal software engineer (AI) at ASOS.

Implementing a recommender with Azure AI

This company used one or more of the following technologies:

- Microsoft recommenders repository on GitHub

- Azure Cosmos DB

- Azure Machine Learning

- Azure Databricks

- Azure Kubernetes

April
2020

AI in Action: Five Case Studies

Technical deliverable: Data transformation for recommenders

In real-world datasets, such as data collected from users' past clothing interests and activity on the ASOS website, actual ratings may be scarce. Instead, customer preferences need to be inferred from their interactions with clothing items and pages on the website. A user that has visited the page of a clothing item multiple times, for example, may prefer that item's style. Placing the item in the shopping cart but not buying it may indicate a stronger preference, while purchasing the item suggests the strongest preference.

In this example, the data scientist uses website activity to determine a weighted count, or affinity, for the clothing item:

```python
# Create a copy of the data
weighted_data = data.copy()

# Assign a numerical value based on the type of interaction
conditions = [
    weighted_data['Type'] == 'click',
    weighted_data['Type'] == 'add',
    weighted_data['Type'] == 'purchase'
]
choices = [1, 2, 3]
weighted_data['Weight'] = np.select(conditions, choices, default='black')

# Convert to numeric type.
weighted_data['Weight'] = pd.to_numeric(weighted_data['Weight'])

#
weighted_data_count = weighted_data.groupby(['UserId', 'ItemId']).
agg({'Timestamp': 'count'}).reset_index()
weighted_data_count.columns = ['UserId', 'ItemId', 'Affinity']
```

Technical deliverable: Creating a recommender

Sometimes—especially when there are too many options—developers should use a recommendation engine to winnow the field. Microsoft provides a GitHub repository of code for recommenders with options and algorithms that are best for solving certain types of problems. The repository makes it easy to switch between code, allowing data scientists to experiment with different algorithms and then choose the one that best performs for their application.

There are many ML models that can be used for a recommender. The Alternating Least Squares (ALS) model, for example, is one way to minimize the loss function for a particular dataset. To use it, the data scientist first needs to split a dataset into two sets—one for training the data and one for testing the resulting model.

```
train, test = spark_random_split(data, ratio=0.75, seed=123)
print ("N train", train.cache().count())
print ("N test", test.cache().count())
```

A header needs to be created that assigns particular columns from the dataset, `data`, to particular metrics in the model. If the data from ASOS, for example, consisted of users, clothing items, and the user's rating for that item, then the model can be constructed using the following code:

```python
header = {
    "userCol": "UserId",
    "itemCol": "ClothingItem",
    "ratingCol": "Rating",
}
als = ALS(
    rank=10,
    maxIter=15,
    implicitPrefs=False,
    regParam=0.05,
    coldStartStrategy='drop',
    nonnegative=False,
    seed=42,
    **header
)
model = als.fit(train)
```

Conclusion

Personalizer, part of Azure Cognitive Services, is the right choice when you are developing an application for which there are a limited number of choices (less than 50), a well-defined description of the content and actions to take, enough daily events to keep the Personalizer up to date (at least 1,000), and well-described user profiles.

The Microsoft Xbox group found that Azure Cognitive Services Personalizer increased user engagement by 40 percent, delivering content that its core customers wanted to see. The reinforcement learning technology allows real-life gains compared to other previously used technology.

The ASOS data science teams wanted tools that would help them do more than just accelerate their modeling process. They wanted to break down barriers for teams and develop best practices for dealing with data science and modeling. Using a single platform for personalizing and recommending content helped them succeed.

Resources

Create your Azure free account today

Online retailer solves challenges with Azure Machine Learning service

Deliver the Right Experiences & Content like Xbox with Cognitive Services

What is Personalizer?

Quickstart: Personalizer client library

Tutorial: Use Personalizer in Azure Notebook

4.
Knowledge Mining in Action

Knowledge mining helps businesses uncover latent insights from vast amounts of data to make better-informed decisions, automate business processes, and identify risks and opportunities. Microsoft estimates that 80 percent of business data is unstructured, making knowledge mining capabilities necessary for digital transformation.

Microsoft Azure Cognitive Search allows companies to ingest a variety of unstructured data from distributed data stores, enrich that data with cognitive functions and machine learning, and present that data in a structured format back to the user.

Azure Cognitive Search is a platform as a service (PaaS) that allows businesses, developers, and researchers to use cognitive skills to enrich their data and create custom skills to perform industry- and application-specific transformations of data into meaningful business information.
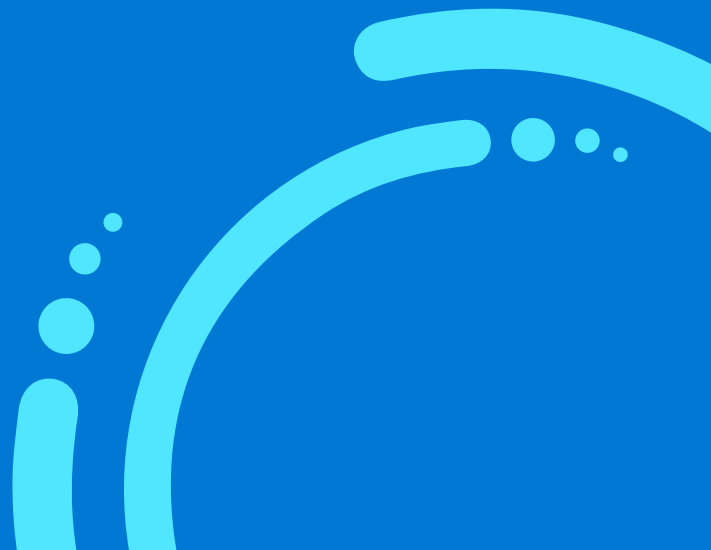
The Met mines collections for new knowledge

The Metropolitan Museum of Art (the Met) houses nearly two million works of art and is continually working to expand its reach worldwide. Three years ago, the museum launched an Open Access program, seeking to make the images and data of public-domain works in its collection available online under an open-data promise.

"The Met welcomes more than seven million physical visitors each year," says Maria Kessler, senior program manager for digital partnerships at the Metropolitan Museum of Art. "However, our goal is to reach the 3.9 billion internet-connected people around the globe."

Such an endeavor requires cataloging and tagging every piece of art in the museum. Historically this was done manually, requiring an immense investment of resources and personnel and resulting in the potential for errors.

Solution and technology

Using Azure Cognitive Services, the Met can automatically survey and classify its entire art collection. The Met is now experimenting not only to catalog and tag each work but to generate new knowledge about the collection at scale. In addition to basic tagging, this knowledge mining can uncover new relationships among the artworks in the collection as well as enrich the information about each piece using publicly available information.

Financial Fabric speeds data acquisition

For large companies, the processing of documents has traditionally required a vast investment in resources and time. Financial Fabric regularly works with clients who need to process thousands of documents—such as financial statements and corporate filings—every day. The company has created technology to help its clients process structured data in, for example, spreadsheets, but it needed to tackle the semi-structured data contained in documents from different financial houses.

Solution and technology

Using a simple REST API call to an Azure Cognitive Service known as Form Recognizer, Financial Fabric created a new document extraction service to quickly find text, fields, and tables in documents. Form Recognizer learns the structure of documents to intelligently extract text and data.

Financial Fabric initiated two pilot projects with Form Recognizer to create data-handling solutions for two clients. One of them was a Fortune 500 company's corporate treasury, which has a large hedge fund. The other pilot solution client was also a Fortune 500 company, whose corporate treasury trust often manages billions of dollars in assets. Using traditional methods, entering in data from tens of thousands of documents can take weeks or months, but Financial Fabric attained the same results in minutes.

"We can extract data from a file and put it on a dashboard with unheard-of speed—latencies of practically zero or within a few minutes," says Subhra Bose, chief executive officer of Financial Fabric. "It normally takes days or sometimes weeks to go from a file to actual analytics."

Implementing knowledge mining with Azure AI
These companies used the following technologies:

- Azure Cognitive Search

- Azure Form Recognizer

- Azure Functions

- Azure Portal

Technical deliverable: Adopting a skill

Azure has more than a dozen ways of transforming content using pretrained machine learning algorithms. These cognitive skills include detecting important phrases, translating text, consolidating text from different sources, recognizing people or places in images, and determining the sentiment of a certain piece of text. Companies can easily integrate these skills into their applications.

Using the Cognitive Skillset REST API and JSON, a series of skills can be performed to transform content and enrich the data. In the example below, entity recognition is used to find the organizations mentioned in a document.

```
{
      "@odata.type": "#Microsoft.Skills.Text.EntityRecognitionSkill",
      "context": "/document",
      "categories": [ "Organization" ],
      "defaultLanguageCode": "en",
      "inputs": [
        {
          "name": "text",
          "source": "/document/content"
        }
      ],
      "outputs": [
        {
          "name": "organizations",
          "targetName": "organizations"
        }
      ]
}
```

Technical deliverable: Defining a custom skill

The Met created a custom classifier to determine specific classifications and tags for the artwork. Custom skills based on machine learning can use the same JSON schema as above but reference a uniform resource identifier (URI) that points to the skill's location. When a machine learning model is not needed, an Azure Function can be used for the transformation. As long as the developer follows the input and output schema, they can use any language and any framework to build a skill.

```
{
    "description": "Extract image analysis.",
    "@odata.type": "#Microsoft.Skills.Vision.ImageAnalysisSkill",
    "context": "/document/normalized_images/*",
    "defaultLanguageCode": "en",
    "visualFeatures": [
        "Tags",
        "Categories",
        "Description",
        "Faces"
    ],
    "inputs": [
        {
            "name": "image",
            "source": "/document/normalized_images/*"
        }
    ],
    "outputs": [
        {
            "name": "categories"
        },
        {
            "name": "tags"
        },
        {
            "name": "description"
        },
        {
            "name": "faces"
        }
    ]
}
```

Technical deliverable: Integrating skills with apps

The Azure Portal can be used to bring together data, predefined skills, and custom functions into an application for searching and enriching data. After creating a data store and importing data, users can link built-in and custom skills through the Azure Portal. By selecting "Import Data" from the portal, users can then add enrichments, such as predefined and custom skills.

Conclusion

Azure Cognitive Search allows businesses to search through unstructured data, enrich the information, and create a custom search application to mine information from their operations data. The Met's use of AI helps the museum encourage the exploration of art in immersive new ways. The metadata generated by Cognitive Search could allow new search and navigation options, giving patrons the ability to see where a work was created or to compare pictured items to actual artifacts. Future analysis could also result in the presentation of visually similar pieces and allow the arrangement of the collections in different ways, such as by art medium or time period.

Financial Fabric expects its improved data handling will meet the growing needs of its clients, investors who want more transparency of their risk. The ability to complete fast and accurate data analysis at high volume will make it significantly easier for financial organizations to provide this transparency.

Resources

Create your Azure free account today

The Metropolitan Museum of Art is exploring the use of AI to build a deeper understanding of its encyclopedic collection

Data services company uses Form Recognizer to cut document-processing time from months to minutes

Knowledge Mining with Azure Cognitive Search

Create a custom skill using the Bing Entity Search API

5.
Forecasting
in Action

A key promise of machine learning is the prediction and forecasting of a variety of events based on limited information, from the performance of a company's stock to the on-time departure of a bus. Microsoft Azure Machine Learning service gives companies the tools they need.

TransLink predicts bus departures

In the Vancouver metropolitan area, TransLink serves more than 400 million riders every year over a 1,800 square kilometer area. Customers want to know whether the bus is at capacity and if it will be on time.

Solution and technology

The transportation company deployed 18,000 different sets of machine learning models to better predict bus departure times, taking into account factors such as traffic, bad weather, and at-capacity buses. Using Machine Learning Operations (MLOps) with Azure Machine Learning, the company trained models and deployed the best predictors to production. Ultimately, TransLink customers saw an improvement in predicted bus departure times of 74 percent.

Another project focused on creating a platform that could determine bus "crowdedness" using machine learning. With Microsoft support, a group of interns used TransLink data and Azure Machine Learning to forecast how full buses would be. Using their model, the interns built a mobile app to allow people to plan their trips based on how at-capacity a bus might be at different times of day, with an eye toward changing riders' habits.

Johnson Controls makes smart cities cooler

Johnson Controls has produced HVAC, fire suppression, and security equipment for buildings for more than 130 years. The company is now targeting the smart city revolution, applying machine learning to predict maintenance needs for its equipment. The company currently runs thousands of chillers with 70 different types of sensors each, streaming terabytes of data.

Solution and technology

Using Microsoft Machine Learning Operations (MLOps) helped put models into production in a timely fashion, with a repeatable process, to deliver real-time insights on maintenance routines. As a result, chiller shutdowns could be predicted days in advance and mitigated effectively, delivering cost savings and increasing customer satisfaction.

Implementing forecasting with Azure AI

These companies used the following technologies:

- Azure Machine Learning

- Azure Functions

- Azure Portal

- Azure Blob Storage

Technical deliverable: Ingesting data from Azure Blob

To predict future events, whether bus delays or maintenance shutdowns, the machine learning model needs to be trained. The first step in such training is to ingest data from Azure Blob, the Microsoft cloud storage service. The following code, from a tutorial on GitHub, loads the data to train a supervised machine learning network, the data to test the model, and the ground-truth data, which are the actual results the model should predict.

```
    # The raw train data is stored on Azure Blob here:
    basedataurl = "https://{user-account}.blob.core.windows.net/{data-
    location}"

    train_file_path = os.path.join(basedataurl,'PM_train.txt')
    test_file_path =  os.path.join(basedataurl,'PM_test.txt')
    truth_file_path  = os.path.join(basedataurl,'PM_truth.txt')

    # Load raw training data from Azure blob
    cols2drop = [{any columns to drop}]

    # read training data
    train_df = pd.read_csv(train_file_path, sep=" ", header=None)
    train_df.drop(train_df.columns[cols2drop], axis=1, inplace=True)
    train_df.columns = [{clean versions of columns headers}]
    train_df.rename(columns={'id':'bus_route'}, inplace=True)

    # read test data
    test_df = pd.read_csv(test_file_path, sep=" ", header=None)
    test_df.drop(test_df.columns[cols2drop], axis=1, inplace=True)
    test_df.columns = train_df.columns
    test_df.rename(columns={'id': 'bus_route'}, inplace=True)

    # read ground truth data
    truth_df = pd.read_csv(truth_file_path, sep=" ", header=None)
    truth_df.drop(truth_df.columns[[1]], axis=1, inplace=True)
```

Technical deliverable: Creating the compute cluster to run the model

Microsoft Azure offers a number of ways to run machine learning models. In this example, we provision a four-node Azure ML Compute resource to execute the training model. Azure also provides options for accelerating deep neural networks using field-programmable gate arrays (FPGAs) and with graphics processing units (GPUs). The compute instance can also be configured with extra features, such as libraries to help explain detailed feature importance in automated model tuning.

```python
from azureml.core.compute import AmlCompute
from azureml.core.compute import ComputeTarget
import os

# choose a name for your cluster
compute_name = os.environ.get("AML_COMPUTE_CLUSTER_NAME", "cpu-cluster")
compute_min_nodes = os.environ.get("AML_COMPUTE_CLUSTER_MIN_NODES", 0)
compute_max_nodes = os.environ.get("AML_COMPUTE_CLUSTER_MAX_NODES", 4)

# This example uses CPU VM. For using GPU VM, set SKU to STANDARD_NC6
vm_size = os.environ.get("AML_COMPUTE_CLUSTER_SKU", "STANDARD_D2_V2")


if compute_name in ws.compute_targets:
    compute_target = ws.compute_targets[compute_name]
    if compute_target and type(compute_target) is AmlCompute:
        print('found compute target. just use it. ' + compute_name)
else:
    print('creating a new compute target...')
    provisioning_config = AmlCompute.provisioning_configuration(vm_size = vm_
size,
                                                               min_nodes =
compute_min_nodes,
                                                               max_nodes =
compute_max_nodes)

    # create the cluster
    compute_target = ComputeTarget.create(ws, compute_name, provisioning_
config)

    # can poll for a minimum number of nodes and for a specific timeout.
    # if no min node count is provided it will use the scale settings for the
cluster
    compute_target.wait_for_completion(show_output=True, min_node_count=None,
timeout_in_minutes=20)

     # For a more detailed view of current AmlCompute status, use get_status()
    print(compute_target.get_status().serialize())
```

Technical deliverable: Deploy model and monitor

Machine learning models can be deployed to the Azure Kubernetes Service by creating a deployment configuration that describes the compute resources needed, in this case the cluster we created previously. As part of deployment configuration, you should create an inference configuration to describe the environment needed to host the model and web service.

```python
from azureml.core.webservice import AksWebservice, Webservice
from azureml.core.model import Model

aks_target = AksCompute(ws,"myaks")
# If deploying to a cluster configured for dev/test, ensure
that it was created with enough
# cores and memory to handle this deployment configuration.
Note that memory is also used by
# things such as dependencies and AML components.
deployment_config = AksWebservice.deploy_configuration(cpu_cores
= 1, memory_gb = 1)
service = Model.deploy(ws, "myservice", [model], inference_
config, deployment_config, aks_target)
service.wait_for_deployment(show_output = True)
print(service.state)
print(service.get_logs())
```

Conclusion

A common application of machine learning is to predict user behavior or forecast events. Azure Machine Learning has many features that support forecasting based on time, user behavior, or other factors.

Resources

Create your Azure free account today

Garage interns bring machine learning to TransLink

Azure Machine Learning—ML for all skill levels

MLOps—the path to building a competitive edge

Deep learning for predictive maintenance

Auto-train a time-series forecast model

Conclusion

Whether it's chatbots, computer vision, predictive engines, or customized recommendations, machine learning and artificial intelligence empower developers and data scientists to digitally transform their business. Azure's suite of AI services gives organizations access to the most advanced machine learning capabilities with no coding, while enabling developers to customize their applications and empowering data scientists to create new, deeper learning models that enhance their data.

The result is a better experience for customers, deeper insights for businesses, and a faster time-to-market for products. Microsoft Azure provides access to AI out of the box, with an unmatched ease-of-use to get developers and data professionals up and running in minutes.
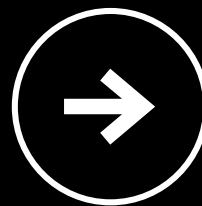
Sign up for an Azure free account today →

Request a call from an Azure specialist for help with pricing, starting your project, migrating workloads, and more. →