AI-900 WhizCard

Quick Exam Reference - Hand-Picked for You

WHIZLABS

Describe Artificial Intelligence workloads and considerations

Identify features of common AI workloads

Artificial Intelligence(AI) is the software that imitates human cognitive abilities and functions.

There are five key elements of Microsoft Artificial Intelligence.

- **Machine Learning (ML)** — tools and services to produce predictions based on the patterns in input data. ML is the foundation of AI systems.
- **Anomaly Detection** — tools and services for the identification of unusual activities.
- **Natural Language Processing (NLP)** — tools and services for the understanding of written and spoken language.
- **Computer Vision** — tools and services for detection, recognition and analysis of objects, faces and text in images and videos.
- **Conversational AI** — tools and services for intelligent conversation.

Identify guiding principles for responsible AI

The principle of **Fairness** guides AI solutions to treat everybody fairly with no bias.
The principle of **Reliability and safety** guides AI solutions to be reliable in operation, to resist harmful manipulations, and be safe for the users.
The principle of **Privacy and security** guides AI solutions to be secure and follow privacy rules.
The principle of **Inclusiveness** guides AI solutions to to provide the benefits of the solutions to everyone with no barriers and limitations. Solution should follow three Inclusive Design Principles: *Recognize exclusion*; *Solve for one, extend to many*, and *Learn from diversity*.
The principle of **Transparency** guides AI solutions to provide full information on their operations, behavior, and limitations.
The principle of **Accountability** guides AI solutions to follow governance and organizational norms.

Source: Microsoft Learn

Describe fundamental principles of machine learning on Azure

Identify common machine learning types

Machine Learning (ML) — software or system that builds and trains models based on the input data to predict the outcome.

Microsoft Machine Learning is the foundation for Artificial Intelligence service. It includes four features and capabilities:

- **Automated machine learning** — automated creation of ML models for non-experts.
- **Azure Machine Learning designer** — a graphical interface for no-code creation of the ML solutions.
- **Data and Compute management** — cloud-based tools for data science professionals.
- **Pipelines**— visual designer for creating ML tasks workflow.

Machine Learning algorithm is a program. It includes instructions for patterns discovery within the provided data set. ML algorithms are generally grouped by ML techniques. There are three main ML groups:

- The **Supervised** group makes predictions based on the information from the previous outcomes (labeled data).
- An **Unsupervised** group makes predictions without any prior knowledge of the possible outcomes.
- The **Reinforced** group learns from the outcome and decides the next move based on this knowledge.

The **Supervised** model types rely on the structured data where input columns or fields are called features, and the output is the label or labels. There are two **Supervised** model types:

- The **Regression** model produces a numeric value prediction for the label, like a game score or a stock price.
- The **Classification** model predicts a class (dog or cat) or multi-class (dog, cat, or rabbit) of the label based on incoming data (features).

The **Clustering** model is the only one that belongs to the **Unsupervised** model type. The **Clustering** model predicts what data points belong to what cluster or group. There is no prior knowledge about the data clusters or groups that can be used for prediction. The **Clustering** algorithm learns common cluster properties first. Then calculates the cluster “membership” probability for each data point.

Describe features of computer vision workloads on Azure

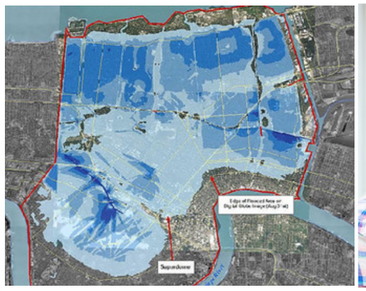
Identify Azure tools and services for computer vision tasks

Computer Vision uses Machine Learning models trained for images and videos.
There are six common Computer Vision tasks.

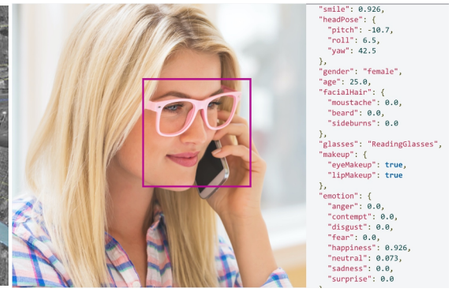
- **Image classification** — analyzes images and videos, detects objects and text, extract descriptions, and create tags.
- **Object detection** — identifies objects and their boundaries within the image.
- **Semantic segmentation** — classifies pixels to the objects they belong to.
- **Image analysis** — extracts information from the images, tag them, and creates a descriptive image summary.
- **Face detection, analysis, and recognition** — detects, analyzes, and recognizes human faces.
- **Optical character recognition** — detects and recognizes text in images and documents.

The **Image classification** model classifies images based on their content. This Computer Vision technique helps doctors find cancer and other medical conditions on X-ray or MRI images. It supports visual product search.Disaster investigation benefits from engineering structures classifications on aerial photos, like bridges.

Object detection helps recognize objects on the images. It places each recognizable object in the bounding box with the class name and probability score.



Semantic segmentation classifies pixels that belong to a particular object, like flooded areas on aerial images, and highlights them.



Azure Face API call returns information about face attributes with a confidence level. Face attributes include age, gender, smile, glasses, emotion, makeup, hair, etc.

Optical character recognition includes two sets of APIs: **Read API** and **OCR API**.

Read API helps "read" texts within predominantly document images. **Read API** is an *asynchronous* service. Microsoft designed it for heavy on text images or documents with many distortions. **Read API** returns page information for each page, including page size and orientation. Then information about each line on the page. And finally, information about each word in each line, including the bounding box of each word.

OCR API extracts small amounts of text within an image. It is *asynchronous* service designed for an immediate result. **OCR API** returns regions on the image with text defined by bounding box coordinates. Then lines of text in each region, bounding box coordinates. And finally, words in each line with bounding box coordinates.

On the picture is the example of the OCR API in action. The service reads the plates of each scooter. The results are in the Preview section.

Describe features of computer vision workloads on Azure

Identify common types of computer vision solution

Computer Vision is one of the key elements for Artificial Intelligence. It includes the following services:

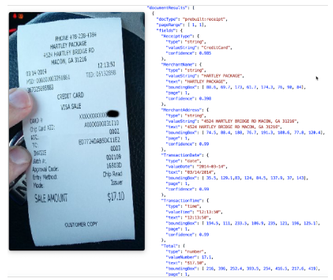
- **Computer vision** — analyzes images and video, detects objects and text, extracts descriptions, and creates tags.
- **Custom vision** — trains custom models for image classification and custom object detection.
- **Face** — detects, analyzes, and recognizes faces.
- **Form Recognizer** — extracts information from scanned forms and invoices.
- **Video Indexer** — analyzes and indexes video and audio content.

Computer vision service works with images. This service brings sense to the image pixels by using them as features for ML models. These predefined models help categorize and classify images, detect and recognize objects, tag, and identify them. **Computer vision** can "read" a text in images in 25 languages and recognize landmarks.

Custom Vision service helps create your own computer vision model. These models are based on image classifications. As for any classification model, it should be a set of images for each known class or category. **Custom Vision** service relies on *deep learning techniques*. These techniques use *convolutional neural networks (CNN)*. *CNN* links pixels to the classes or categories. For a creation of the Custom Vision solution, users can use a general Azure Cognitive Service resource. It includes both resources, for training and prediction. Or they can create separate Custom Vision resources only for training or prediction. Such separation is useful only for resource tracking purposes.
After provisioning the resources, users train the model at the Custom Vision portal: https://www.customvision.ai. Here they can create applications and submit images. It should be enough images with object classes from various angles. When a model is created, the service assesses the model performance based on the following metrics:

- **Precision** — defines the percentage of the class predictions that the model makes correct. For example, if the model predicts ten images are bananas, and there are actually only seven bananas, the model precision is 70%.
- **Recall** — defines the percentage of the class identification that the model makes correct. For example, if there are ten apple images, and the model identifies only eight, the model recall is 80%.
- **Average Precision (AP)** is the combined metrics of both Precision and Recall.

For automated document processing, **Form Recognizer** uses two models:
Custom Model and **A pre-build receipt model**.
With the **Custom model** approach, you train the **Form Recognizer** model based on your own form and data. You just need only 5 samples of your form to start.
A pre-build receipt model is a **Form Recognizer** default model that is trained to work with receipts. It helps recognize receipts and extract data from them.



Form **Recognizer** service uses pre-build receipt models to extract such information from receipts: *date of transaction*, *time of the transaction*, *merchant information*, *taxes paid*, *receipt total*. The service also recognizes all text on the receipt and returns it.

Azure **Cognitive Face** service currently includes the following functionality: **Face detection**, **Face verification**, **Find similar faces**, **Group faces on similarities**, and **Person identification**.

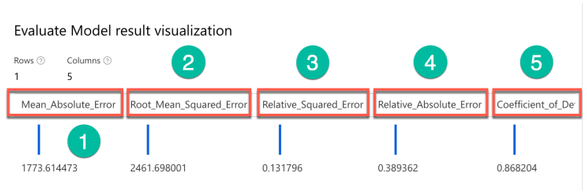
Ver. 01.03.1231-20 Created by Dr. Andrei Sergeev

AI-900 WhizCard

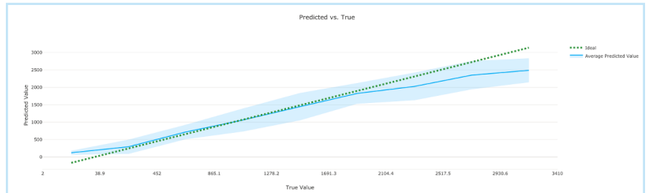
Describe fundamental principles of machine learning on Azure

Describe core machine learning concepts. Regression Model Metrics.

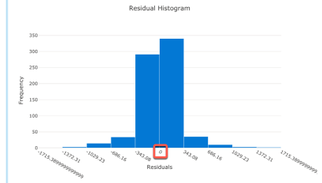
Azure ML uses model evaluation for the measurement of the trained model accuracy. For **Regression** models Evaluate Model module provides the following five metrics:



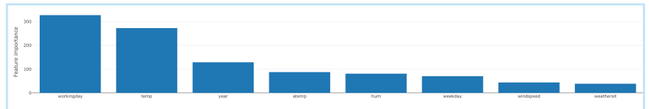
- **Mean absolute error (MAE)** (1) is the regression model evaluation metrics. It produces the score that measures how close the model is to the actual values — the lower score, better the model performance.
- **Root Mean Squared Error (RMSE)** (2) is the regression model evaluation metrics. It represents the square Root from the squared mean of the errors between predicted and actual values.
- **Relative squared error (RSE)** (3) is the regression model evaluation metrics. It is based on the square of the differences between predicted and true values. The value is between 0 and 1. The closer this value is to 0, the better is model performance. Relativity of this metric helps to compare model performances for the labels in different units.
- **Relative absolute error (RAE)** (4) is the regression model evaluation metrics. It is based on absolute differences between predicted and true values. The value is between 0 and 1. The closer this value is to 0, the better is model performance. Relativity of this metric helps to compare model performances for the labels in different units.
- **Coefficient of determination (R^2)** (5) is the regression model evaluation metrics. It reflects the model performance: the closer R^2 to 1 - the better the model fits the data.



The **Predicted vs. True** chart presents the differences between predicted and true values. The dotted line outlines the ideal model performance and the solid line reflects the average model predictions. Closer these lines to each other, better model performance.



The **Residual histogram** presents the frequency of residual values distribution. ***Residual*** is the difference between predicted and actual values. It represents the amount of error in the model. For high-performance models, we should expect that most of the errors are small. They will cluster around 0 on the Residual histogram.

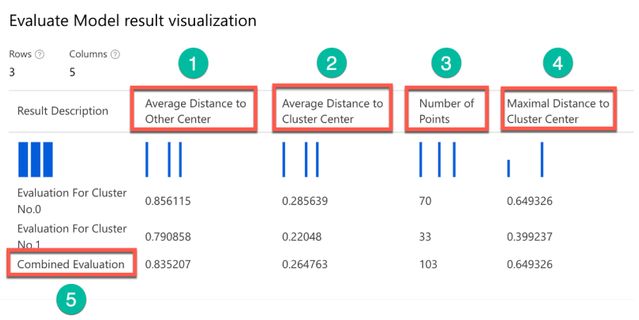


Azure ML Studio provides explanations for the best fitting model. The part of these explanations is the **Global Importance** histogram. It presents the importance of each feature in the label prediction.

Describe fundamental principles of machine learning on Azure

Describe core machine learning concepts. Clustering Model Metrics.

Azure ML uses model evaluation for the measurement of the trained model accuracy. For **Clustering** models, the Evaluate Model module provides the following five metrics:



- **Average Distance to Other Center** (1) is the clustering model evaluation metrics. It reflects how far is the average distance for each data point in a cluster to the centroids of all other clusters.
- **Average Distance to Cluster Center** (2) is the clustering model evaluation metrics. It reflects how far is the average distance for each data point in a cluster to the centroid of the cluster.
- **Number of Points** (3) is the clustering model evaluation metrics. Its value is the number of points assigned to each cluster.
- **Maximal Distance to Cluster Center**(4) is the clustering model evaluation metrics. It reflects the cluster spread.Metric's value is the sum of distances from each point in the cluster to the cluster's centroid.
- **Combined Evaluation** (5)is the clustering model evaluation metrics. It combines the above metrics per cluster into the combined model evaluation metric.

Describe fundamental principles of machine learning on Azure

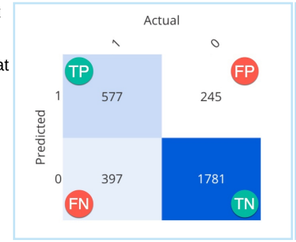
Describe core machine learning concepts. Classification Model Metrics.

The **Confusion** matrix (or error matrix) provides a tabulated view of predicted and actual values for each class. It is usually used as a performance assessment for **Classification** models. But it can also be used for fast visualization of the **Clustering** model results too.

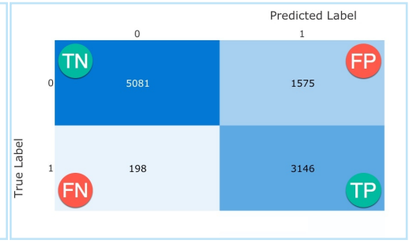
A binary confusion matrix is divided into four squares that represent the following values:

- **True positive (TP)** — the number of positive cases that the model predicted correctly.
- **True negative (TN)** — the number of negative cases that the model predicted correctly.
- **False positive (FP)** — the number of positive cases that the model predicted as negative.
- **False negative (FN)** — the number of negative cases that the model predicted as positive.

Azure ML Studio offers two types of Confusion matrix that have different places for the **TP** and **TN**.

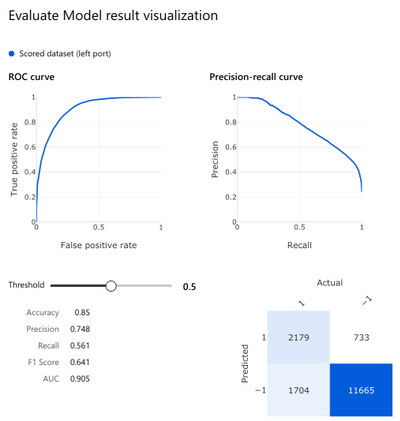


Azure ML Designer Confusion matrix



Azure ML Automated ML Confusion matrix

Azure ML uses model evaluation for the measurement of the trained model accuracy. For **Classification** models, the Evaluate Model module provides the following metrics:



- **Accuracy** metric defines how many predictions (positive and negative) are actually predicted right. To calculate this metric, use the following formula: (TP+TN)/Total number of cases.
- **Precision** metric defines how many positive cases are actually predicted right. To calculate this metric, use the following formula: TP/(TP+FP).
- **Recall** or **True Positive Rate (TPR)** metric defines how many positive cases that model predicted are actually predicted right. To calculate this metric, use the following formula: TP/(TP+FN).
- **F1 Score** metric combines *Precision* and *Recall*. To calculate this metric, use the following formula: 2TP/(2TP+FP+FN).
- **Fall-out** or **False Positive Rate (FPR)** metric defines how many negative cases that model predicted are actual predicted right. To calculate this metric, use the following formula: FP/(FP+TN).

The **Receiver Operator Characteristics** or **ROC Curve** is the relation between **FPR** (**Fall-out**) and **TPR** (**Recall**). **ROC Curve** produces the **Area Under Curve (AUC)**.

Area Under Curve (AUC) is a classification model performance metric reflecting how good the model fits the data. For binary classification models, the **AUC** value of 0.5 represents that a model prediction is the same as randomly selected values of "Yes" or "No." If the **AUC** value is below 0.5, the model performance is worse than random. Ideally, the best-fitted model has a value of 1. Such an ideal model predicts all the values correctly.

Describe fundamental principles of machine learning on Azure

Describe core machine learning concepts

The ***feature*** is a generic name for the input column or field in structured data.
The ***label*** is a generic name for the model output of numeric value or class.

Azure ML Designer provides several **Regression ML algorithm** modules, like **Linear Regression** and **Decision Forest Regression**. **Linear Regression** algorithm based on a linear regression model. **Decision Forest Regression** algorithm is based on a decision forest algorithm.

There are several modules for **Classification** algorithm also, like **Two-class Logistic Regression**, **Multiclass Logistic Regression** or **Two-Class Neural Network**. **Two-class** and **Multiclass Logistic Regression** algorithms are based on logistic regression model. **Two-Class Neural Network** is based on a neural network algorithm.

And there is only one algorithm for **Clustering**: **K-means clustering**.

Regression Algorithm Family has the word "***regression***" in their names without class, like Linear ***Regression*** or Decision forest ***regression.***

All algorithm in the ML **Classification** family includes the word "***class***" in their names, like Two-***class*** logistic regression, Multi***class*** logistic regression, or Multi***class*** forest regression.

Helping to navigate through Azure ML algorithm options, Microsoft provides a guide for selecting the best algorithm for your solution Machine Learning Algorithm Cheat Sheet for Azure Machine Learning designer.

 Ver. 01.03.1231-20 Created by Dr. Andrei Sergeev

AI-900 WhizCard

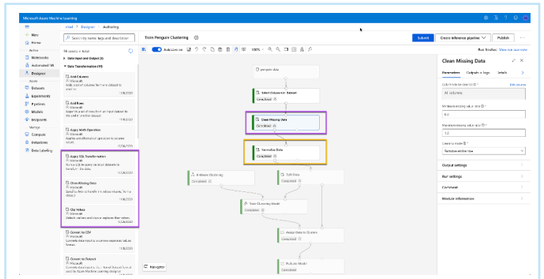
Describe fundamental principles of machine learning on Azure

Identify core tasks in creating a machine learning solution

There are several core tasks for building ML solutions.

1. Data ingestion. Data ingestion is the process of bringing data from different sources into a common repository or storage. After ingestion, data is accessible for various services. There are three general ways to get the data: upload dataset, manual input, and import data.
Azure ML Studio has four options to import data: From local files, From datastore, From web files, and From open Datasets.

2. Data preparation and data transformation. Before we can use the data for ML modeling, we need to prepare or pre-process data: find and correct data errors, remove outliers, impute missing data with appropriate data values. Azure Auto ML executes data preparation during the run. Azure ML Designer provides several modules for these tasks: Clip Values (for outliers), Clean Missing Data (for missing data), Remove Duplicate Rows, Apply SQL Transformation, Python, and R scripts. These prebuilt modules come from Data Transformation and Languages groups.



Azure ML Designer Data transformation Modules

3. Feature selection and engineering. Before model training, we need to review the data, select features that influence the prediction (label), and discard other features from the final dataset. If the dataset has numeric fields/columns on different scales, like one column has all values from 0 to 0.5 and another column — from 100 to 500, we need to bring them to the common scale. This process is named ***data normalization***. If, for better model performance, the ML solution requires to generate a new feature based on the current features, this operation is called ***feature engineering***. Users can use Apply SQL Transformation, Python, and R modules to script these operations. In Azure ML, ***featurization*** is the name for the generic application of all data-preprocessing techniques, like scaling, normalization, or feature engineering.

Clean Missing Data result visualization

Rows 342 Columns 4

CulmenLength	CulmenDepth	FlipperLength	BodyMass
39.1	18.7	181	3750
39.5	17.4	186	3800
40.3	18	195	3250
36.7	19.3	193	3450
39.3	20.6	190	3650

Normalize Data result visualization

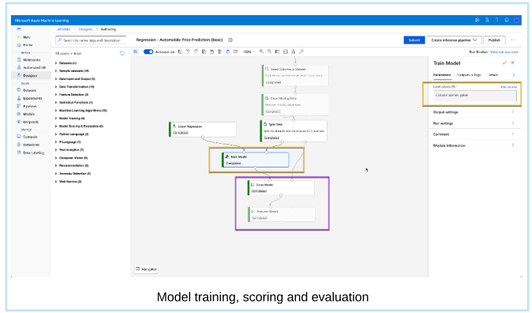
Rows 342 Columns 4

CulmenLength	CulmenDepth	FlipperLength	BodyMass
0.254545	0.666667	0.152542	0.291667
0.269091	0.511905	0.237288	0.305556
0.298182	0.583333	0.389831	0.152778
0.167273	0.738095	0.355932	0.208333
0.261818	0.892857	0.305085	0.263889

Dataset before and after data normalization

4. Model training. After data-preprocessing, data is almost ready for model training. We need to have two sets of data: one for training and one for test or validation. Auto ML splits the original dataset into training and validation sets automatically. However, users have the option to upload a validation set. Azure ML Designer provides a Split Data module for creating training and test datasets. Before the training, we need to connect the left output of a Split Data module to a Training module's right input.We also need to connect the selected for the solution ML Algorithm module to the Training module's left input. For Regression and Classification models, we need to mark a label column in the Training module. And we can run the training.

Important note: Azure ML Studio requires to create a compute resource for model training. We must provision Training Clusters in the Compute section of the Manage blade or within Auto ML or Azure Designer settings.



Model training, scoring and evaluation

5. Evaluation. After training is finished, we need to score (test) a model with the test dataset. Auto ML uses a validation dataset for model validation and cross-validation of the child processes. Azure ML Designer utilizes the Score Model module to score the model predictions using the test dataset. The score results are supplied to the Evaluate Model module. The module evaluates the score results and produces the standard model performance metrics.

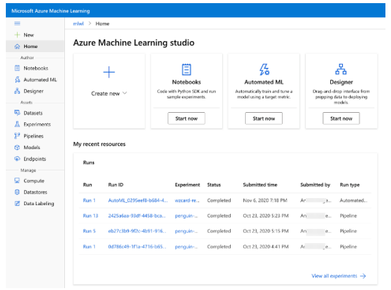
6. Model deployment. If model performance is satisfactory, we can deploy the model to a production environment. For model deployment, we need to create the production version of the training Pipeline — ***inference pipeline***. Azure ML Designer provides an option to create a real-time inference pipeline. After we create the inference pipeline, we need to do the test run. But before that, we want to be sure that our input field selection doesn't include a labeled field. If the inference pipeline test run is successful, we are ready to deploy our solution. For the next step, we must have **Inference Clusters** provisioned. Only then can we push the "Deploy" button in the Azure ML Designer's top right, and the solution will be deployed. We can see the solution's status on the Assets section's Azure ML Studio Endpoints blade after deployment.

7. Model management. Azure ML implements principles from DevOps for model maintenance. This approach is called ***Machine Learning Operations*** or ***MLOps***. It includes reproducible environments, code management, package deployment, monitoring, alerts, notification, and end-to-end automation.

Describe fundamental principles of machine learning on Azure

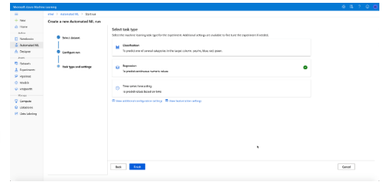
Describe capabilities of no-code machine learning with Azure Machine Learning

The **Azure ML Studio** provides all essential tools to create and work with models. There are three main Author options for creating the experiments: **Automated ML** (or no-code), **Azure ML Designer** (or low-code), and **Notebooks** (or coding).



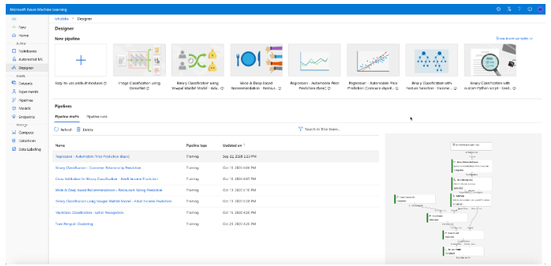
A collection of runs or trials in Azure ML calls an ***experiment***.

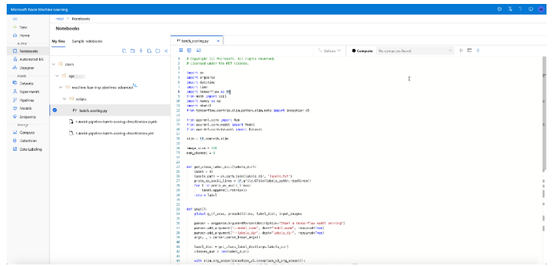
Automated ML is the "no-code" option. It doesn't require any specific data science knowledge. The simple wizard type interface helps users to set up Auto ML run. It includes dataset selection, experiment and compute resource setup, and learning task choice. The wizard presents three task types: Classification, Regression, and Time series forecasting. After task selection, users click "Finish" and start the run. Auto ML run settles on the best algorithm and creates the model suited for users' goals.



Azure ML Designer is the "low-code" option. It helps users to create ML workflow using Pipelines. To create a Pipeline, users drag&drop modules from the library on the Designer's canvas. The library includes Data Transformation, ML Algorithms,Model Scoring & Evaluation, and other prebuilt modules.

Notebooks is the "coding" option. Users can use Python language and Python SDKs for coding their ML solutions.





Describe features of Natural Language Processing (NLP) workloads on Azure

Identify features of common NLP Workload Scenarios

Natural Language Processing (NLP) is one of the key elements for Artificial Intelligence. It includes four services:

- **Text Analytics** — helps analyze text documents, detect document's language, extract key phrases, determine entities, and provide sentiment analysis.
- **Translator Text** — helps translate texts in real time between 70+ languages.
- **Speech** — helps recognize and synthesize speech, recognize and identify speakers, translate live or recorded speech.
- **Language Understanding Intelligent Service (LUIS)** — helps to understand voice or text commands.

Azure Cognitive services provide two types of translation: Text and Speech.

Azure **Translator** service supports multi-language near real-time text translations between 70 languages. The service uses neural network technologies. **Custom Translator** extends **Translator** with custom specific language domains. Custom extended models can benefit both **Translator** and **Speech services** for their translations.

Users can easily integrate **Translator** and **Custom Translator** with their applications. Important to know that the **Translator** doesn't store any user's data. If we need to translate the same text into several languages, we can do it in one request to API. We submit a text for translation in the API call body and a sequence of the language codes for translation to as parameters.

Azure Translator service supports multi-language translations between 70 languages. It uses ***Neural Machine Translation (NMT)*** technology as a service backbone. The significant benefit of ***NMT*** is that it assesses the entire sentence before translating the words. **Custom Translator** customizes ***NMT*** systems for translation of the specific domain terminology.

Translator Text API service has two options for fine-tuning the results:

- **Profanity filtering** controls a translation of the profanity words by marking them as profanity or by omitting them.
- **Selective Translation** allows a user to tag a word or phrase that need not be translated, like a brand name.

 Ver. 01.03.1231-20 Created by Dr. Andrei Sergeev

AI-900 WhizCard

WHIZLABS

Describe features of Natural Language Processing (NLP) workloads on Azure

Identify features of common NLP Workload Scenarios

Azure Text Analytics is a part of **Natural Language Processing (NLP)**. It includes the following four services:

- **Language detection** — helps to identify the language of the text.
- **Sentiment analysis** — helps to analyze text, and returns sentiment scores and labels for each sentence.
- **Key phrase extraction** — helps to extract the key phrases from the unstructured text.
- **Named entity recognition** — helps to identify entities in the text and group them into categories.

All **Text Analytics** services, except Language detection, utilize the same JSON body format for API calls. This format includes three fields for each document in the collection: language name, document id, and text for analysis. ***Language detection*** service accepts only two: document id and text for analysis.

Each text should have less than 5,120 characters. And the documents collection can handle up to 1,000 items (ids). Documents in the collection can be in different languages.

Named Entity Recognition (NER) is a Text Analytics service. It identifies entities in the text. And group them into categories, like a person, organization, location, event, and others. **NER** service has two types of API calls: general entity recognition and ***entity linking*** support. For example, there are 18 meanings for the word "bank." It can be "bank of the river" or "agent bank" or "food bank," etc. **NER** service analyzes the link between entities to resolve the possible meaning ambiguity. But to do this effectively, the service uses Wikipedia as a knowledge base for the entity linkage and identification in different languages.
Currently, ***entity linking*** supports only English and Spanish languages. On the contrary, ***general entity*** recognition **NER** service supports 23 languages.

Here is the output from **NER** service ***entity linking*** call `(entities/linking)` for the following sentence:
"After they met at the bank of the Seine, they walked to the Bank of France building."

The word "***bank***" is used twice: as a bank of the river and as a financial institution. The service recognizes the difference by providing Wikipedia links for the Seine River and Bank of France.


```
{
  "documents": [
    {
      "id": "1",
      "entities": [
        {
          "name": "Seine",
          "matches": [
            {
              "text": "the Seine",
              "offset": 30,
              "length": 9,
              "confidenceScore": 0.02
            }
          ],
          "language": "en",
          "id": "Seine",
          "url": "https://en.wikipedia.org/wiki/Seine",
          "dataSource": "Wikipedia"
        },
        {
          "name": "Bank of France",
          "matches": [
            {
              "text": "Bank of France",
              "offset": 60,
              "length": 14,
              "confidenceScore": 0.78
            }
          ],
          "language": "en",
          "id": "Bank of France",
          "url": "https://en.wikipedia.org/wiki/Bank_of_France",
          "dataSource": "Wikipedia"
        }
      ],
      "warnings": []
    }
  ],
  "errors": [],
  "modelVersion": "2020-02-01"
}
```



```
{
  "documents": [
    {
      "id": "1",
      "entities": [
        {
          "text": "bank of the Seine",
          "category": "Location",
          "offset": 22,
          "length": 17,
          "confidenceScore": 0.68
        },
        {
          "text": "Bank of France building",
          "category": "Location",
          "offset": 60,
          "length": 23,
          "confidenceScore": 0.61
        }
      ],
      "warnings": []
    }
  ],
  "errors": [],
  "modelVersion": "2020-04-01"
}
```


If we call **NER** without ***entity linking*** option (`entities/recognition/general`) we get just two locations.

Sentiment analysis is a Text Analytics service. It analyzes text and returns sentiment scores (between 0 and 1) and labels ("positive," "neutral," and "negative") for each sentence. A score close to 0 means a negative sentiment, and close to 1 - positive. And in cases with a neutral or undefined sentiment, the score is 0.5.

Currently, the **Sentiment Analysis** service supports 20 languages. The service helps analyze social media, customer reviews, or other media that provide people's opinions.

An example of the **Sentiment Analysis** output for the phrase:

"Peter was surprised and very happy to meet Sara in Paris."


```
{
  "documents": [
    {
      "id": "1",
      "sentiment": "positive",
      "confidenceScores": {
        "positive": 1.0,
        "neutral": 0.0,
        "negative": 0.0
      },
      "sentences": [
        {
          "sentiment": "positive",
          "confidenceScores": {
            "positive": 1.0,
            "neutral": 0.0,
            "negative": 0.0
          },
          "offset": 0,
          "length": 57,
          "text": "Peter was surprised and v
        }
      ],
      "warnings": []
    }
  ],
  "errors": [],
  "modelVersion": "2020-04-01"
}
```


Key Phrase Extraction service is a part of **Azure Text Analytics**. It helps to extract the key phrases from the unstructured text. This functionality is beneficial when you need to create a summary or a catalog from the document content or understand the customer reviews' key points.

Key Phrase extraction works best with long documents. Using Text Analytics APIs, we can submit the text of the documents in a simple JSON format. This format includes three fields for each document in the collection: language name, document id, and text for key phrase extraction. Each text should have less than 5,120 characters. And the documents collection can handle up to 1,000 items (ids). Documents in the collection can be from different languages.
Currently, **Key phrase extraction** service supports 15 languages.


```
{
  "documents": [
    {
      "language": "en",
      "id": "1",
      "text": "Data ingestion is the process of bringing data from different sources into a common repository or storage. After ingestion, data is accessible for various services."
    },
    {
      "language": "fr",
      "id": "2",
      "text": "Bonjour tout le monde"
    },
    {
      "language": "es",
      "id": "3",
      "text": "La carretera estaba atascada. Había mucho tráfico el día de ayer."
    }
  ]
}
```


The body of the API call


```
{
  "documents": [
    {
      "id": "1",
      "keyPhrases": [
        "Data ingestion",
        "different sources",
        "process",
        "common repository",
        "various services",
        "storage"
      ],
      "warnings": []
    },
    {
      "id": "2",
      "keyPhrases": [
        "monde"
      ],
      "warnings": []
    },
    {
      "id": "3",
      "keyPhrases": [
        "carretera",
        "tráfico",
        "día"
      ],
      "warnings": []
    }
  ],
  "errors": [],
  "modelVersion": "2020-07-01"
}
```


The Key Phrase Extraction Service output.

Azure **Speech-to-Text** service uses the Universal language model. This Microsoft's proprietary model is optimized for conversational and dictation scenarios. Users can use the service for **real-time** or **batch** transcription of the audio data into the text format.
Real-time Speech-To-Text transcribes or translates the audio streams or files into text.
Batch Speech-To-Text transcription is an asynchronous service. It works with large audio data stored in Azure Blob Storage.

Azure Text-to-Speech service gives users an option to select between ***standard*** and ***neural*** voice generation. **Neural voices** sound very close to human reproducing stress and intonation of spoken language. Users also can create custom voices.

Speech Translation is a part of the Speech services. It is powered by a **Translator** and combines **Translator Speech APIs** with **Custom Speech** services. It provides real-time multi-language translation functionality for user's applications. Users can use this service for **Speech-to-Speech** and **Speech-to-Text** translations.

Azure Speech service provides two APIs: **Speech-to-Text** and **Text-to-Speech**. These APIs also include speech recognition and speech synthesis functionalities.

Describe features of Natural Language Processing (NLP) workloads on Azure

Identify Azure tools and services for NLP workloads

The biggest challenge for processing the language is to understand the meaning of the text or speech. The language understanding models resolve this issue. **Azure Language Understanding service**, or **LUIS**, helps users to work with language models.
The primary goal of **LUIS** based applications is to understand the user's ***intention***. **LUIS** examines the user's input, or ***utterance***, and extracts the keywords, or ***entities***. It then uses a compiled list of ***entities*** linked to ***intent*** and outputs the probable action or tasks that the user wants to execute.

Azure provides a **LUIS** portal (https://www.luis.ai) for creating solutions based on language models.There are two stages in this process: ***authoring*** and ***prediction***.

Authoring is the process of language understanding model creation and training. To train these models, we need to supply the following key elements:

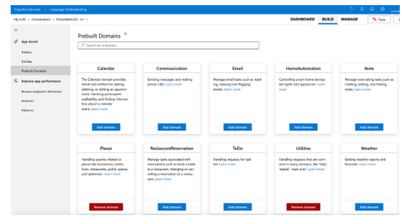
- **Entity** is the word or phrase that is the focus of the ***utterance***, as the word "light" in the utterance "`Turn the lights on`."
- **Intent** is the action or task that the user wants to execute. It reflects in ***utterance*** as a goal or purpose. We can define **intent** as "`TurnOn`" in the ***utterance*** "`Turn the lights on`."
- **Utterance** is the user's input that your model needs to interpret, like "`Turn the lights on`" or "`Turn on the lights`."

After we define the intents and entities, we can iteratively train our model by using sample utterances. When we are satisfied with the model performance, we publish the LUIS application.

Every LUIS model has the default "***None***" intent. It is empty by default, and it can't be deleted. This intent matches utterances outside of the application domain.

There are four types of **entities** that we can create: *Machine-Learned*, *List*, *RegEx*, and *Pattern.any*.

We can use pre-built LUIS collections of intents and entities for the common domains, like Calendar, Places, Utilities, etc, for our model.



Prediction is the process of publishing and using the model. Clients can connect to the predicting resource's endpoint by providing an authentication key. Before creating a LUIS application, users need to choose what type of Azure resources they want to provision for their solution. There are two types of resources: the dedicated LUIS resources (authoring or prediction or both) and general Azure Cognitive services resources (only for prediction). This flexibility helps the user manage resources and access to different Cognitive services. But it has some overhead for the developers.

Azure Speech includes the following services:

- **Speech-to-Text** — transcribes audio data into text
- **Text-to-Speech** — synthesizes human-like voice audio data from the input text.
- **Speech Translation** — provides a real-time multi-language translation of the spoken language audio data into speech or text
- **Voice Recognition** — recognizes and authenticates a speaker by the specific voice characteristics.

Azure Voice Recognition service helps to identify and verify the speakers by unique characteristics of their voice. Speakers train the service by using their voice and service creates an enrollment profile. Based on this profile system can identify the speaker or user by his/her voice. The **Speaker Recognition** APIs can identify speakers in voice recordings, real-time chats, and video streams.

Speech recognition and **Speech synthesis** are parts of Azure **Speech** Services. These services help determine the spoken language content and generate the audio content by the synthetic voice.
Speech Recognition uses many models, but two are essential: **The Acoustic** and **The Language**.
The Acoustic model converts audio into *phonemes*. **The Language** model matches phonemes with words. There are several examples of **Speech recognition** applications. Closed captions, transcripts of the phone calls or meetings, or text dictation are some of them.
Speech Synthesis is the "opposite" service to **Speech Recognition**. It requires text content and the voice to vocalize the content. It is working in reverse to the recognition. First, **Speech Synthesis** tokenizes the text into individual words and matches them with phonetic sounds. Then it puts together the sounds into *prosodic* units, like phrases or sentences, and creates *phonemes* from them. After that, the service converts *phonemes* into an audio sequence. Voice synthesizer outputs audio sequence. We can control voice output options by **Speech Synthesis Markup Language (SSML). SSML**, XML-based language, can change the voice speed and pitch or how the text or the text's parts should be read. We use **Speech Synthesis** service in many areas, like personal voice assistants, phone voice menus, or public announcements in airports and train stations.

Describe features of conversational AI workloads on Azure

Identify common use cases for conversational AI

Azure **Conversation AI** supports agents, or bots, that can keep a conversation in turns with the users. Examples of such systems are Web chat AI agents, or bots, and Smart home devices that can answer your questions and act on your commands.

Every organization is trying to keep its costs low. Usually, customer service is one of their expensive operations. So, the challenge is how to lower customer support costs without lowering service quality. Conversation AI agents became a trendy solution for this problem as alternative and/or addition to human customer service.

A simple example of a Conversation AI agent is a **WebChat Bot**. **WebChat Bot** can conversationally answer customer's questions in real-time. Customers can interact with bot by many channels, like a web browser, chat application, phone calls, emails, text messages, social media, and others.

To create a **WebChat Bot**, we need two components: ***Knowledge base*** and ***Bot Service***. ***Knowledge base*** stores information that bot is accessing and providing answers from. We can build a ***Knowledge base*** from website information, FAQ documents, chit-chat lists, etc. Usually, the ***Knowledge base*** is a list of question-and-answer pairs. ***Bot Service*** provides an interface for users to interact with a Knowledge base by communication channels.

Ver. 01.03.1231-20 Created by Dr. Andrei Sergeev

AI-900 WhizCard

WHIZLABS

Describe features of conversational AI workloads on Azure

Identify common use cases for conversational AI

Another popular customer service solution is **telephone voice menus.** The **telephone voice menus** functionality is a good example of a **Speech Synthesis** service.

Speech Synthesis service is a part of Azure **Text-to-Speech** services. Text-To-Speech provides two options for the voice: ***Standard*** and ***Neural***.

The ***Neural*** voice uses deep neural networks for Speech Synthesis and makes output sounds very close to humans. It reduces listening fatigue when people interact with automated attendants.

Users can create their own Bots using **Microsoft Bot Framework**. Bot Framework provides conversation control and integrates with **QnA Maker**.

Microsoft **Azure Bot Service** based on the **Bot Framework**. Users can provision Web App Bot in Azure. During the creation of the Web Bot, users have to choose between two pre-defined templates:

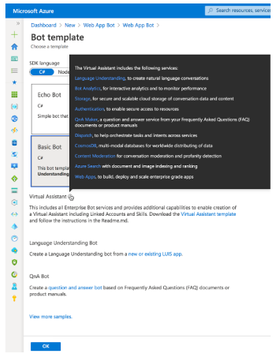
- **Echo Bot** — a simple bot that echos the customer's messages.
- **Basic Bot** — includes "out of the box" integration with LUIS and Bot Analytics services.

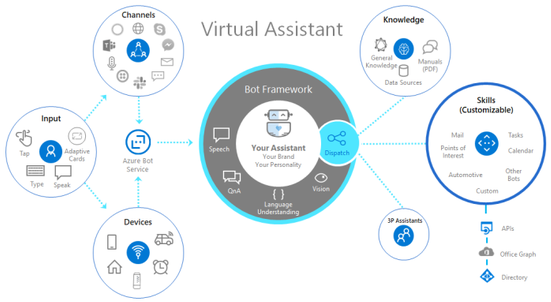
Using **Bot Framework Skills**, users can extend the Bots capabilities. **Skills** are like standalone bots that focus on specific functions: Calendar, To Do, Point of Interest, etc.

When users want to embed Web Bot within their website, they are using Web Chat control. This control requires a secret key for bot access. The secret key is a master key that allows access to all bot's conversations. Its free exposure in a production environment creates a significant security risk. To limit this risk, users need to generate a token based on the secret key. A token gives control access only to a single conversation and has an expiration term.

A **Personal Digital Assistant** is a **Bot Framework** solution. It is based on three major components: ***Azure Bot Service***, ***Bot Framework*** and ***Knowledge base***.

During the creation of the Azure Web App Bot, users can select **Basic Bot** template. This basic Bot can be extended to the Virtual Assistant using Enterprise Bot Services. Digital Assistant incorporates many services, like LUIS, QnA Maker, Cosmos DB, Content Moderation, and others.





Source: Microsoft Bot Framework Documentation

Identify Azure services for conversational AI

Conversation AI is one of the key elements of Artificial Intelligence. It includes two services:

- **QnA Maker** — helps to create a knowledge base - a foundation for a conversation between humans and AI agents.
- **Azure Bot Service** — helps to create, publish, and manage Conversation AI agents, or bots.

Azure **QnA Maker** service transforms the semi-structured text information into structured question-and-answers pairs. Azure stores these pairs as **Knowledge base (KB)**. When the service receives a customer's question, it matches the question with answers from the **Knowledge base**. And then outputs the most appropriate answer with a confidence score.

Before users create a new **Knowledge base**, they need to provision the **QnA Maker** resource in Azure subscription. On the contrary, to other Cognitive Services, **QnA Maker** depends on several Azure services:

- **QnA Maker Management service** — the model training and publishing service.
- **Azure Search** — stores data that submitted to QnA Maker.
- **Azure App Service** — hosts QnA Maker's endpoint.

Azure creates all these services with a provision of the **QnA Maker** instance. After resource deployment, users can create and connect **Knowledge base** to the instance using the Azure QnA portal (https://www.qnamaker.ai/). The portal helps populate **KB** with information from online FAQs, product manuals, different file documents, etc.

If users want to attach a personality to the conversations, they can add ***Chit-chat*** — lists of small talk pairs. Portal provides several pre-built ***Chit-chat*** lists: professional, friendly, witty, caring, and enthusiastic. Azure adds the selected ***Chit-chat*** to the user's **Knowledge base**.

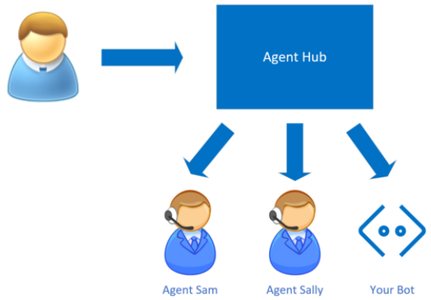
QnA Maker resource can be connected to several **Knowledge bases**. But the language of the first **KB** defines the language for the rest of the bases within the **QnA Maker** instance.

After creating **Knowledge base**, users need to train and test their models. Then published for the clients to use over the REST API. Applications accessing published **Knowledge base** must provide the *KB id*, *KB endpoint address*, and *KB authorization key*. Users can deliver **KB** by creating a Bot. They also can use the **QnA Maker** functionality of bot generation for their **Knowledge base**.

Describe features of conversational AI workloads on Azure

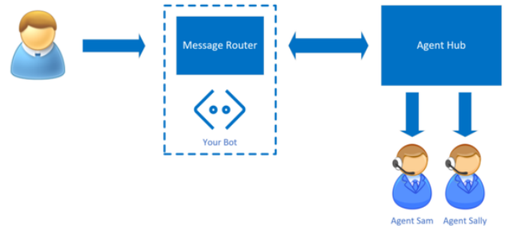
Identify Azure services for conversational AI

Microsoft Bot Framework supports two models of bot integration with agent engagement platforms, like Customer support service. These two models are **Bot as agent** and **Bot as a proxy**.



Source: Microsoft Documentation

Bot as agent model integrates bot on the same level as live agents: the bot is engaged in interactions the same way as Customer support personnel. Handoff protocol regulates bot's disengagement and a transfer of user's communication to a live person. This is the most straightforward model to implement.



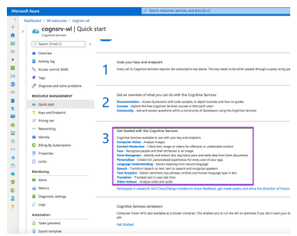
Source: Microsoft Documentation

Bot as proxy model integrates bot as the primary filter before the user interacts with a live agent.

Bot's logic decides when to transfer a conversation and where to route it. This model is more complicated to implement.

AZURE COGNITIVE SERVICES

Azure Cognitive Services resource includes access to the list of Cognitive Services. Customers can use a common endpoint and authentication key to access Computer Vision, Content Moderator, Face, Language Understanding, Speech, Text Analysis, Translator, and other services. The endpoint address consists of the Azure Cognitive service resource name and `cognitiveservices.azure.com` domain, like `https://cognsrv-wl.cognitiveservices.azure.com/`, where "cognsrv-wl" is the resource name. User can utilize the REST API or SDK to call services.



Some services are not on this list and require a separate resource, like Custom Vision, QnA Maker, or Web Chat Bot.

REST API

The REST API HTTP protocol helps users to access Azure AI applications. REST API is a set of rules for Web services. It stands for "**RE**presentational **S**tate **T**ransfer **A**pplication **P**rogramming **I**nterface."

HTTP REST API protocol comprises two parts:**request** and **response**. A user makes a **request** to an AI endpoint to process the input data. After ML service processes the **request**, it sends back a **response** with the results.

The request includes four key parts: service ***endpoint*** URL, a ***method***, ***header*** and a ***body***. Every ***endpoint*** has a service root URL, service path, and query parameters (optional). The HTTP protocol defines the five main action ***methods*** for the service: GET, POST, PUT, PATCH, and DELETE. The ***header*** contains the authentication key for the service. And the ***body*** includes data that the user wants to process by the AI service, like the text to analyze. The **request** can be sent using Postman or CURL command.

The **response** contains the service output in JSON format.

About Whizlabs and WhizCard

Whizlabs is the premier provider of educational and training content. For the last 20 years, millions of IT specialists have been using **Whizlabs**. **Whizlabs** practice tests, video courses, and labs help them prepare and pass the toughest certification exams from companies like Microsoft, AWS, Google, Oracle, IBM, and others.
WhizCard summarizes the required knowledge for the certification and serves as guidance for the exam preparation. Every section header and sub-headers reflect the Microsoft Skills Measured document for the exam. They are also linked to the appropriate Microsoft documents for in-dept study. Please let us know if you like our ***WhizCard*** or how we can improve it. Visit us at www.whizlabs.com or connect with us on Twitter, LinkedIn, Facebook or Slack.

Good luck on your learning journey, and may the power of knowledge be with you.