Technical Deep Dive AI: Data Science Process and Machine Learning

Transform data into actionable insight

Microsoft

Session Objectives

- Understand the Data Science Process
- Discover how you can use Azure Machine Learning to create models
- Learn how to deploy your models to provide your application with predictive analysis capabilities

Agenda

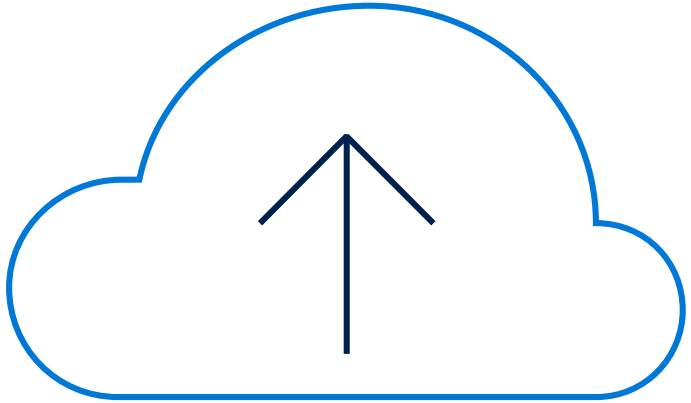
In this Webcast we give you deep insights into the **Data Science Process** and **Azure Machine Learning Platform**, with its fully managed services for **building**, **deploying**, **and managing machine learning and AI models** in the **cloud**. We will show you how to prepare your data, build your Machine Learning model, operationalize your model and consume it within your application. Additionally, we will review the different tools and services you have available for each process.

- Overview
- Data Science Process
- Data Acquisition
- Understanding Data
- Modeling
- Optimization
- Deployment
- Consume

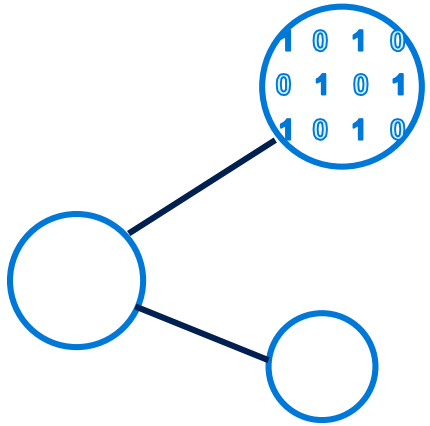
Data, analytics & AI accelerates digital transformation for every organization through data driven insights and action.

Insights is a journey

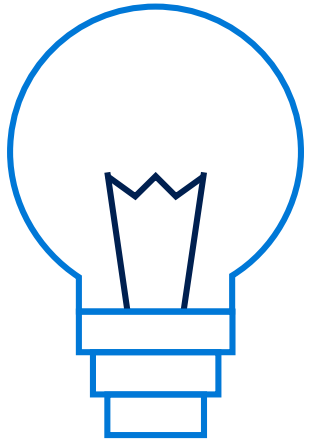
Convergence accelerates digital transformation



Cloud



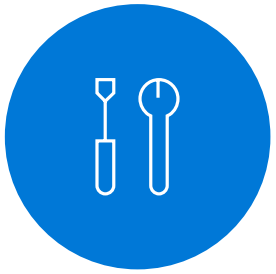
Data



Intelligence

Digital Transformation

Engage customers



Empower employees

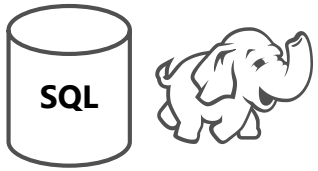
Optimize operations

Transform products

Opportunities exist across functional areas

Marketing	Sales	Service	Finance	Operations	Workforce
Personalization	Lead/Opportunity Scoring	Intelligent Contact Center	Financial Forecasting	Predictive Maintenance	Employee Insights
Customer Insights	Sales Insights	Patient Care & Healthcare Analytics	Fraud Management	Demand Forecasting	HR Insights
Churn Analytics			Risk Management	Operational Efficiency	Resource Matching & Planning
Dynamic Pricing				Inventory Optimization	
Product Innovation				Operations Anomaly Insights	
Marketing Optimization				Quality Assurance	
Product Recommendation				Connected Devices & Smart Buildings	
				Supplier & Spend Insights	

Solution Scenarios

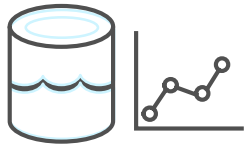


Modern data warehousing

"We want to integrate all our data—including Big Data—with our data warehouse"

Advanced analytics

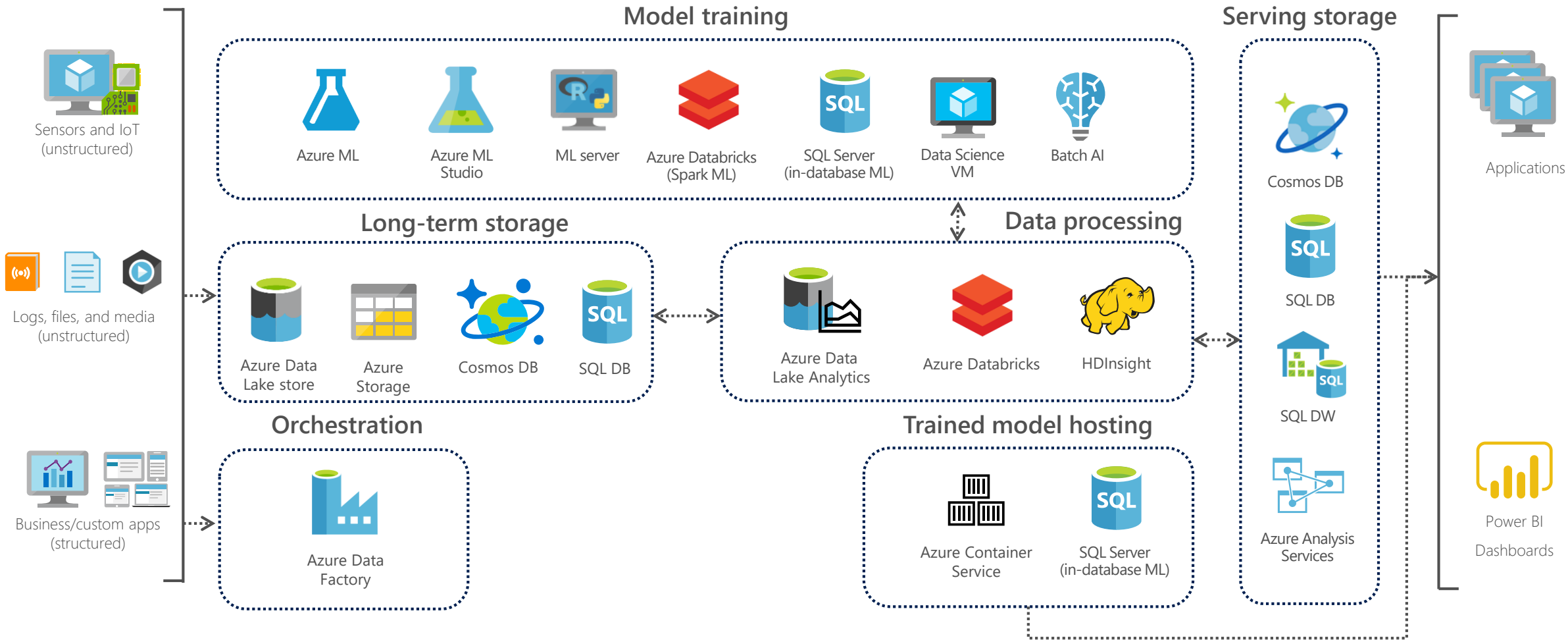
"We're trying to predict when our customers churn"



Real-time analytics

"We're trying to get insights from our devices in real-time"

Advanced analytics pattern in Azure



What is machine learning?

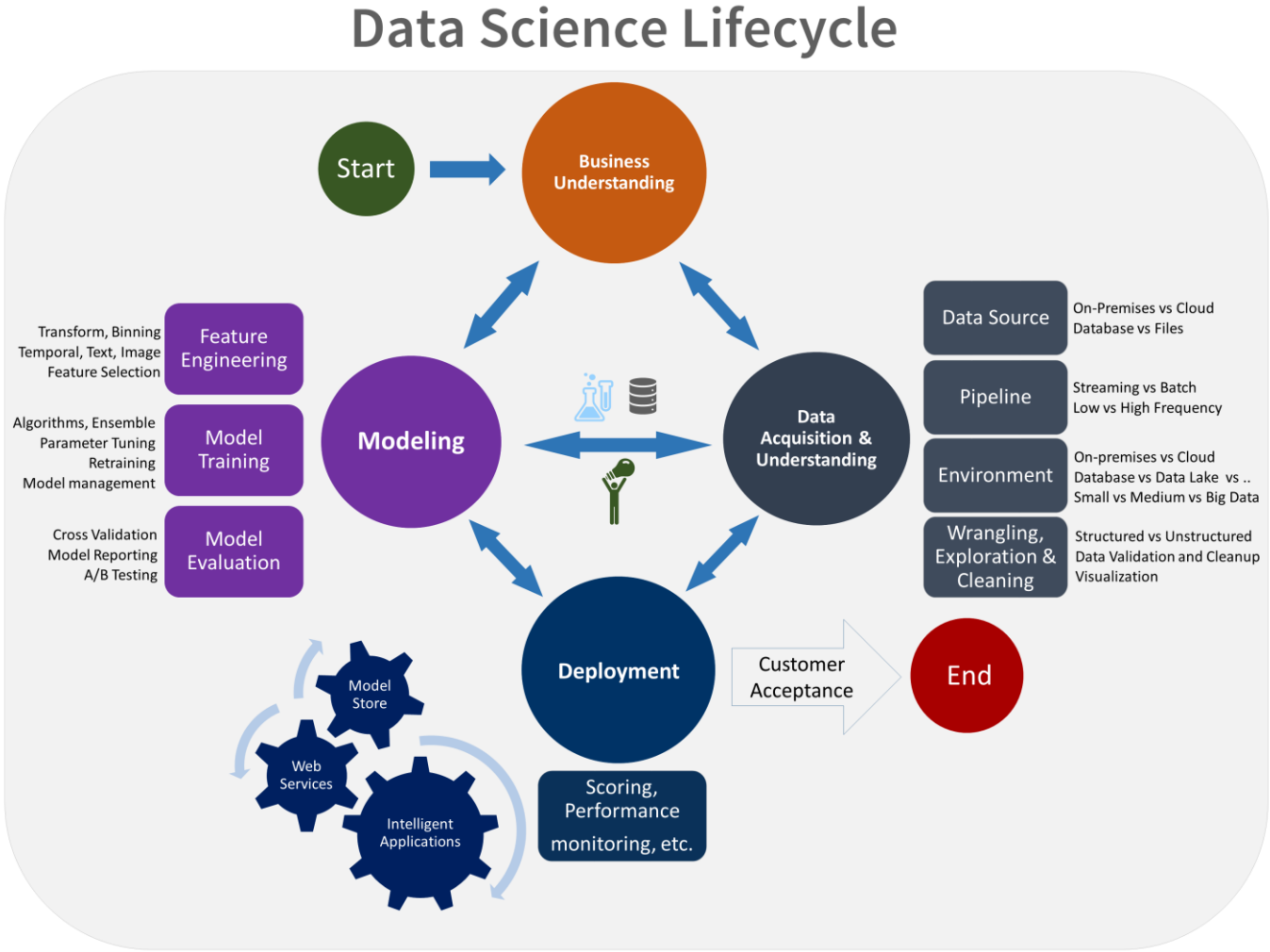
Machine learning is a data science technique that allows computers to use existing data to forecast future behaviors, outcomes, and trends. Using machine learning, computers learn without being explicitly programmed.

Data Science Process

The Team Data Science Process (TDSP) is an agile, iterative data science methodology to deliver predictive analytics solutions and intelligent applications efficiently.

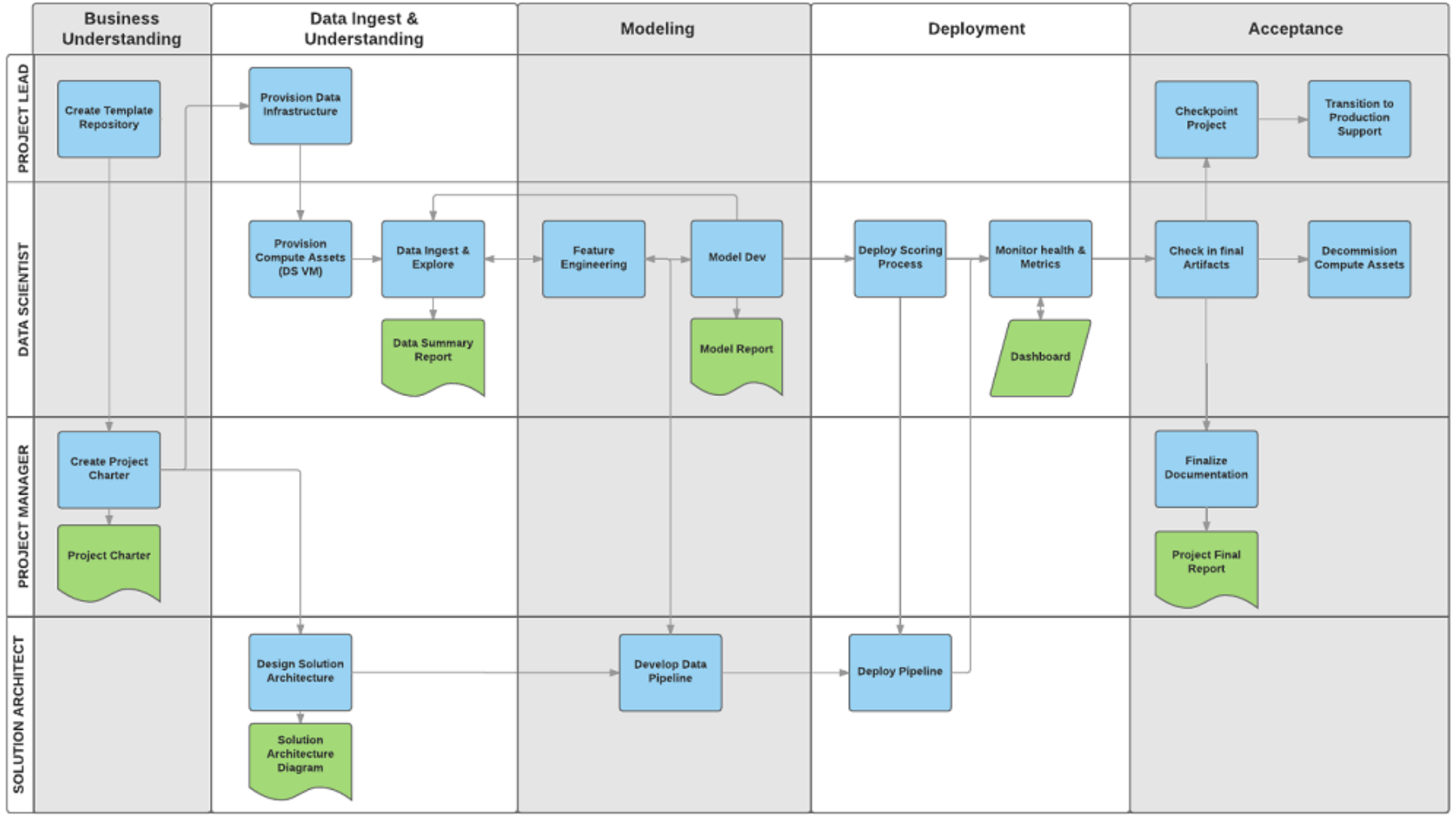
The TDSP lifecycle is composed of five major stages that are executed iteratively. These include:

1. **Business Understanding**
2. **Data Acquisition and Understanding**
3. **Modeling**
4. **Deployment**
5. **Customer Acceptance**



The following diagram provides a grid view of the tasks (in blue) and artifacts (in green) associated with each stage of the lifecycle (on the horizontal axis) for these roles (on the vertical axis).

https://github.com/Azure/Azure-TDSP-ProjectTemplate



1. Business Understanding

Goals

- Specify the key variables that are to serve as the model targets and whose related metrics are used determine the success of the project.
- Identify the relevant data sources that the business has access to or needs to obtain.

How to do it

- Define objectives: Work with your customer and other stakeholders to understand and identify the business problems. Formulate questions that define the business goals that the data science techniques can target.
- Identify data sources: Find the relevant data that helps you answer the questions that define the objectives of the project.

Artifacts

- Charter document
- Data sources
- Data dictionaries

2. Data Acquisition and Understanding

Goals

- Produce a clean, high-quality data set whose relationship to the target variables is understood. Locate the data set in the appropriate analytics environment so you are ready to model.
- Develop a solution architecture of the data pipeline that refreshes and scores the data regularly.

How to do it

- Ingest the data into the target analytic environment.
- Explore the data to determine if the data quality is adequate to answer the question.
- Set up a data pipeline to score new or regularly refreshed data.

Artifacts

- Data Quality Report
- Solution Architecture
- Checkpoint Decision

3. Modeling

Goals

- Determine the optimal data features for the machine-learning model.
- Create an informative machine-learning model that predicts the target most accurately.
- Create a machine-learning model that's suitable for production.

How to do it

- Feature engineering: Create data features from the raw data to facilitate model training.
- Model training: Find the model that answers the question most accurately by comparing their success metrics.
- Determine if your model is suitable for production.

Artifacts

- Feature sets
- Model report
- Checkpoint Decision

4. Deployment

Goal

- Deploy models with a data pipeline to a production or production-like environment for final user acceptance.

How to do it

- Operationalize the model: Deploy the model and pipeline to a production or production-like environment for application consumption.

Artifacts

- A status dashboard that displays the system health and key metrics
- A final modeling report with deployment details
- A final solution architecture document

5. Customer acceptance

Goal

- Finalize the project deliverables: Confirm that the pipeline, the model, and their deployment in a production environment satisfy the customer's objectives.

How to do it

- System validation: Confirm that the deployed model and pipeline meet the customer's needs.
- Project hand-off: Hand the project off to the entity that's going to run the system in production.

Artifacts

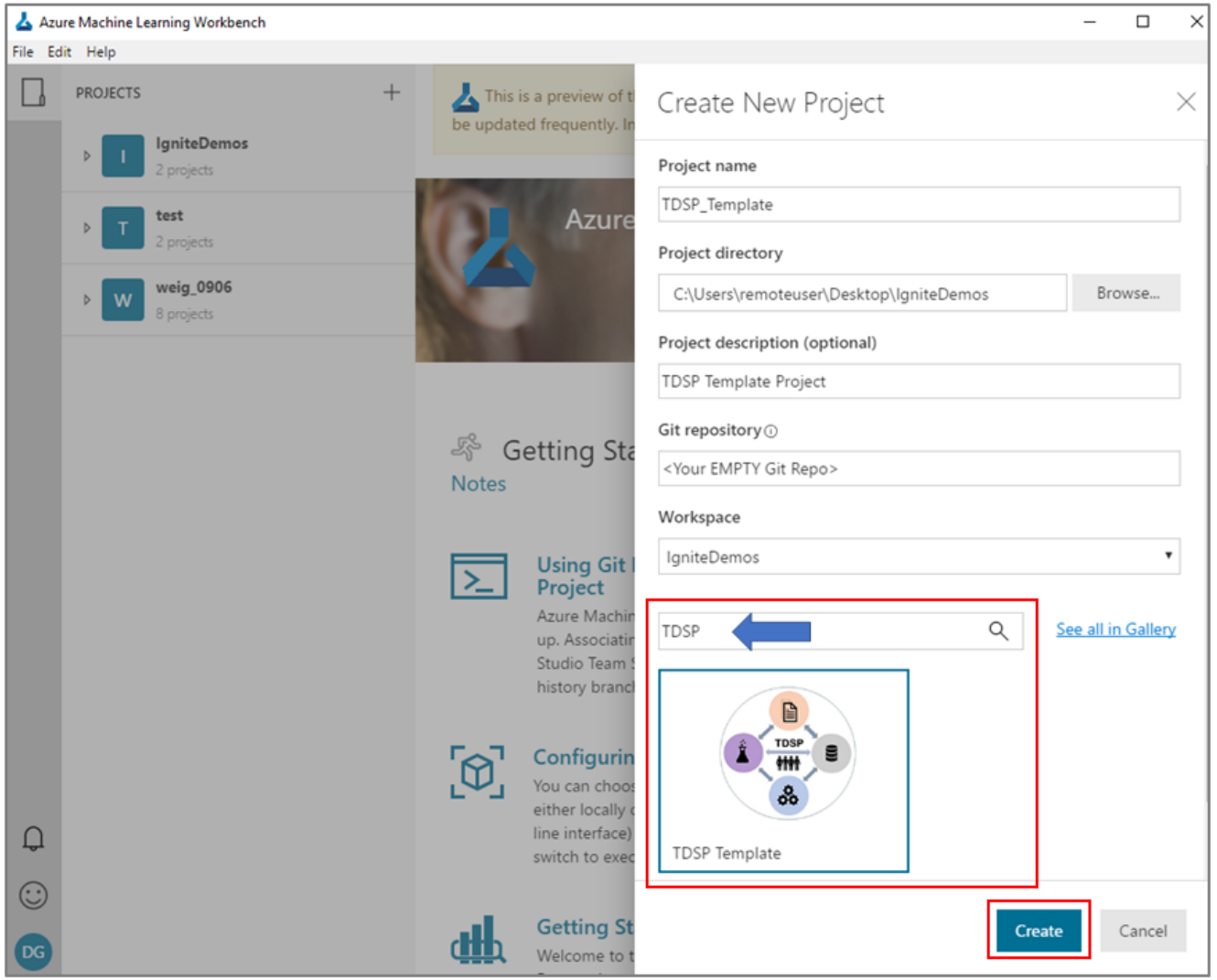
- The main artifact produced in this final stage is the Exit report of the project for the customer.

Structure projects with the Team Data Science Process template

Standardization of the structure, lifecycle, and documentation of data science projects is key to facilitating effective collaboration on data science teams.

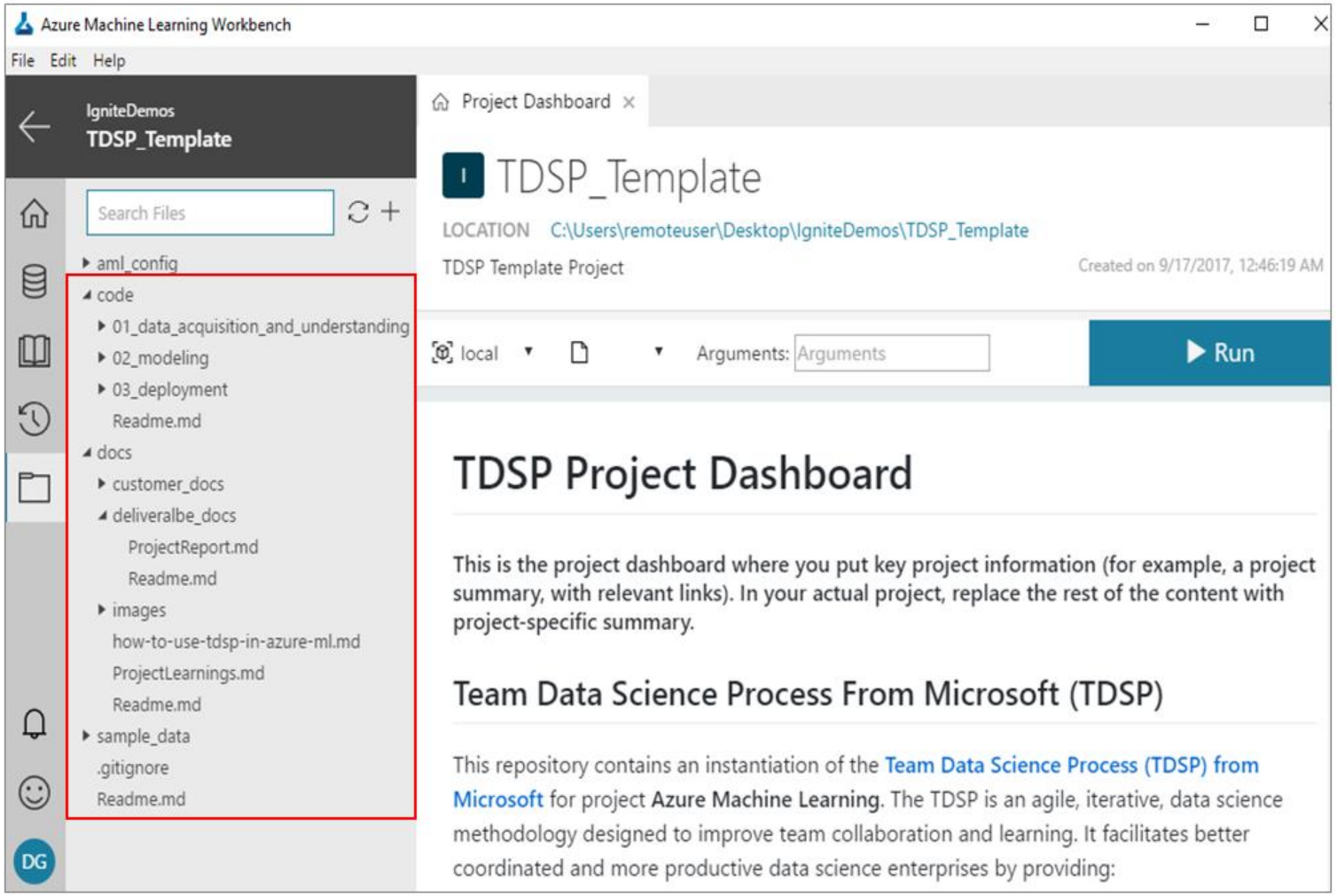
Create a new TDSP-structured project

1. Specify the parameters and information in the relevant box or list:
    1. Project name
    2. Project directory
    3. Project description
    4. An empty Git repository path
    5. Workspace name
2. Then in the **Search** box, enter **TDSP**.
3. When the **Structure a project with TDSP** option appears, select that template.
4. Select the **Create** button to create your new project with a TDSP structure.



Examine the TDSP project structure

- **code**: Contains code.
- **docs**: Contains necessary documentation about the project (for example, markdown files and related media).
- **sample_data**: Contains SAMPLE (small) data that you can use for early development or testing. Typically, these sets are not larger than several (5) MB. This folder is not for full or large data sets.



Platforms and tools for data science team projects

Microsoft provides a full spectrum of data and analytics services and resources for both cloud or on-premises platforms. They can be deployed to make the execution of your data science projects efficient and scalable.

The data and analytics services available to data science teams using the TDSP include:

- **Data Science Virtual Machines (both Windows and Linux CentOS)**
- **HDInsight Spark Clusters**
- **SQL Data Warehouse**
- **Azure Data Lake**
- **HDInsight Hive Clusters**
- **Azure File Storage**
- **SQL Server 2016 R Services**

Data acquisition

How to identify scenarios and plan for advanced analytics data processing

What resources should you plan to include when setting up an environment to do advanced analytics processing on a dataset?

Logistic questions: data locations and movement

- What is your data source? Is it local or in the cloud?
- What is the Azure destination?
- How are you going to move the data?
- Does the data need to be moved on a regular schedule or modified during migration?
- How much of the data is to be moved to Azure?

Data characteristics questions: type, format, and size

- What are the data types?
- How is your data formatted?
- How large is your data?

Data quality questions: exploration and pre-processing

- What do you know about your data?
- Does the data require pre-processing or cleaning?

What languages do you prefer to use for analysis?

- R
- Python
- SQL

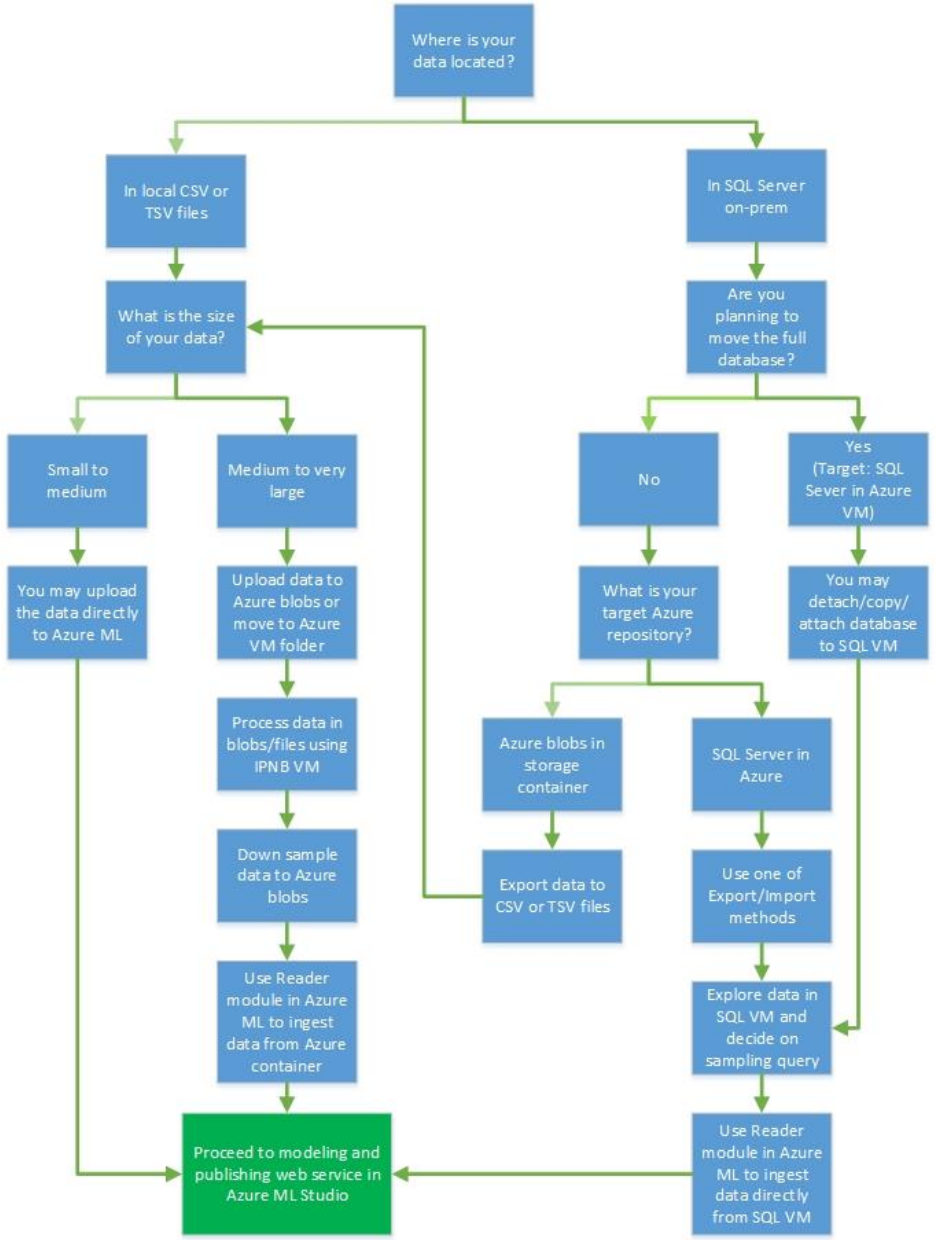
What tools should you use for data analysis?

- Microsoft Azure Powershell •Azure Machine Learning Studio
- Revolution Analytics
- RStudio
- Python Tools for Visual Studio
- Anaconda
- Jupyter notebooks
- Microsoft Power BI

Analyze business needs

Decision tree for scenario selection

The following diagram summarizes the scenarios for the Advanced Analytics Process and Technology choices.



Acquire and understand data

- Before any analytics can be performed, the data must be collected and stored somewhere. For advanced analytics, the choice is between Azure Storage and Azure Data Lake Store
- Alternatively, another service, such as Azure Data Factory, can coordinate pulling data from the source and writing it to long-term storage

	Azure Data Lake Store	Azure Storage
API	WebHDFS-compatible REST API (HTTPS)	REST API over HTTP/HTTPS
HDFS-compatible interface	Yes	Yes
Authentication	Based on Azure Active Directory identities	Based on shared secrets account access keys and shared access signature keys, and role-based access control (RBAC)
Authorization	POSIX access control lists (ACLs). ACLs based on Azure Active Directory identities can be set at file level and folder level	For account-level authorization use account access keys. For account, container, or blob authorization use shared access signature keys.
Encryption	Transparent server side encryption with both service-managed and customer- managed keys by using Azure Key Vault	Transparent server side encryption with both service-managed and customer-managed keys by using Azure Key Vault
SLA	99.9 percent	99.9 percent
Analytics workload performance	Optimized performance for parallel analytics workloads, high throughput and IOPS	Not optimized for analytics workloads
Capacity	No limit on account sizes, file sizes, or number of files	Max 500 TB per account, max 4.75 TB per file
Regional availability	Available in a limited set of regions	Available in all regions

Understanding Data

Is your data ready for data science?

Criteria for data

So, in the case of data science, there are some ingredients that we need to pull together.

We need data that is:

- **Relevant**
- **Connected**
- **Accurate**
- **Enough to work with**

Is your data relevant?

Irrelevant Data

Price of milk ($/gal)	Red Sox batting avg.	Blood alcohol content (%)
3.79	.304	.03
3.45	.320	.09
4.06	.259	.01
3.89	.298	.05
4.12	.332	.13
3.92	.270	.06
3.23	.294	.10

Relevant Data

Body mass (kg)	Margaritas	Blood alcohol content (%)
103	3	.03
67	5	.09
87	1	.01
52	2	.05
73	5	.13
79	3	.06
110	7	.10

Do you have connected data?

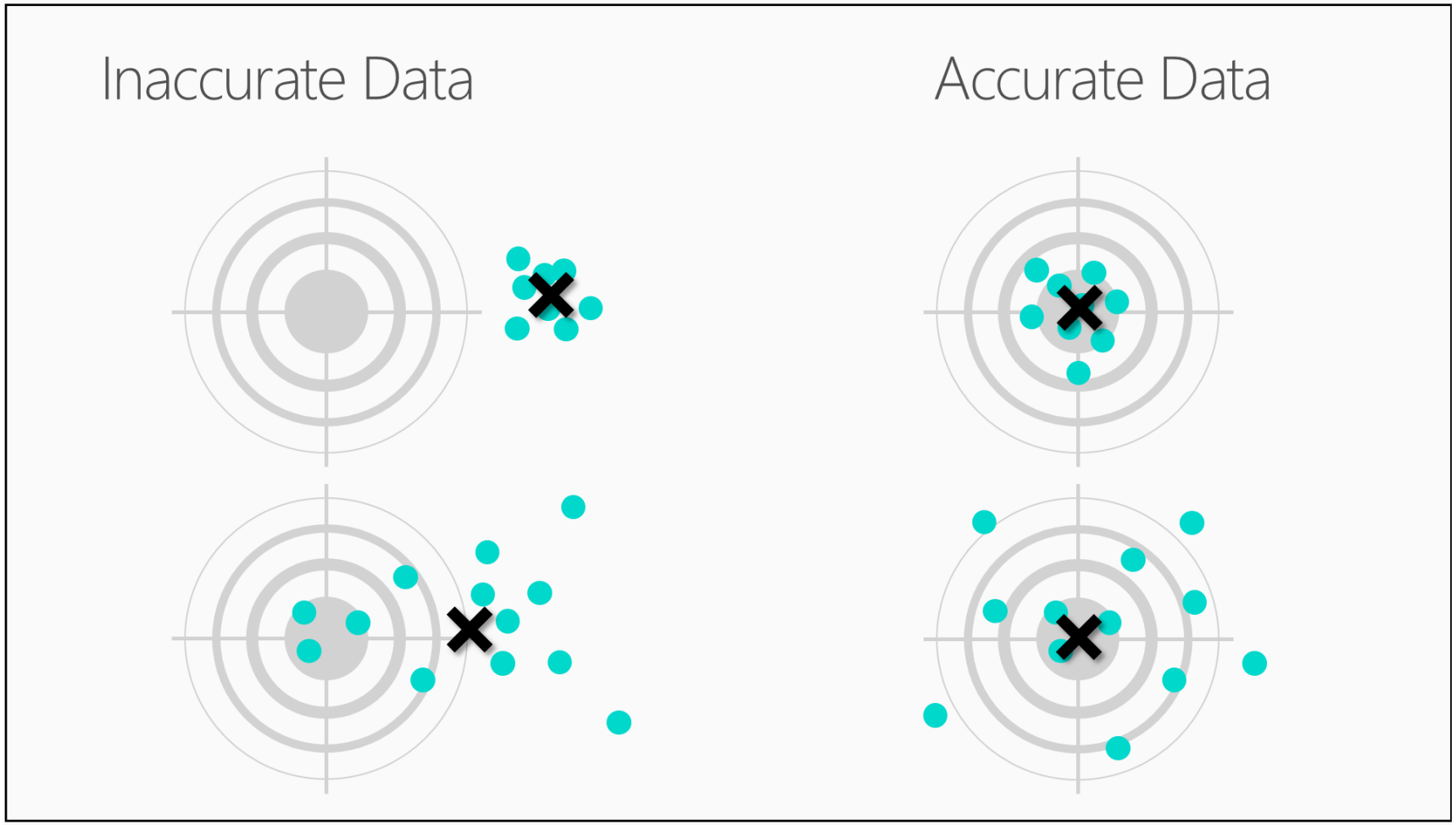
Disconnected Data

Grill temp. (Fahrenheit)	Weight of beef patty (lb)	Burger rating (out of 10)
	.33	8.2
	.24	5.6
550		7.8
725	.45	9.4
600		8.2
625		6.8
	.49	4.2

Connected Data

Grill temp. (Fahrenheit)	Weight of beef patty (lb)	Burger rating (out of 10)
575	.33	8.2
550	.24	5.6
550	.69	7.8
725	.45	9.4
600	.57	8.2
625	.36	6.8
550	.49	4.2

Is your data accurate?



Do you have enough data to work with?

Barely enough data

Prepare Data

Pre-processing and cleaning data are important tasks that typically must be conducted before dataset can be used effectively for machine learning. Raw data is often noisy and unreliable, and may be missing values. Using such data for modeling can produce misleading results

Why pre-process and clean data?

- **Incomplete**: Data lacks attributes or containing missing values.
- **Noisy**: Data contains erroneous records or outliers.
- **Inconsistent**: Data contains conflicting records or discrepancies.

What are some typical data health screens that are employed?

- The number of **records**.
- The number of **attributes** (or **features**).
- The attribute **data types** (nominal, ordinal, or continuous).
- The number of **missing values**.
- **Well-formedness** of the data.
- **Inconsistent data records**. Check the range of values are allowed.

Azure Machine Learning consumes well-formed tabular data. If the data is already in tabular form, data pre-processing can be performed directly with Azure Machine Learning in the Machine Learning Studio. **If data is not in tabular form, say it is in XML, parsing may be required in order to convert the data to tabular form.**

What are some of the major tasks in data pre-processing?

- **Data cleaning**: Fill in or missing values, detect and remove noisy data and outliers.
- **Data transformation**: Normalize data to reduce dimensions and noise.
- **Data reduction**: Sample data records or attributes for easier data handling.
- **Data discretization**: Convert continuous attributes to categorical attributes for ease of use with certain machine learning methods.
- **Text cleaning**: remove embedded characters which may cause data misalignment, for e.g., embedded tabs in a tab-separated data file, embedded new lines which may break records, etc.

Missing Values	Normalize	Discretize	Reduce	Clean Text
•Deletion •Dummy substitution •Mean substitution •Frequent substitution •Regression substitution	•Min-Max Normalization •Z-score Normalization •Decimal scaling	•Equal-Width Binning •Equal-Height Binning	•Record Sampling •Attribute Sampling •Aggregation	•Text Fields in tabular data •Data exploration

Explore data in the Team Data Science Process

Examples using Pandas

dataframe_blobdata.head(10)

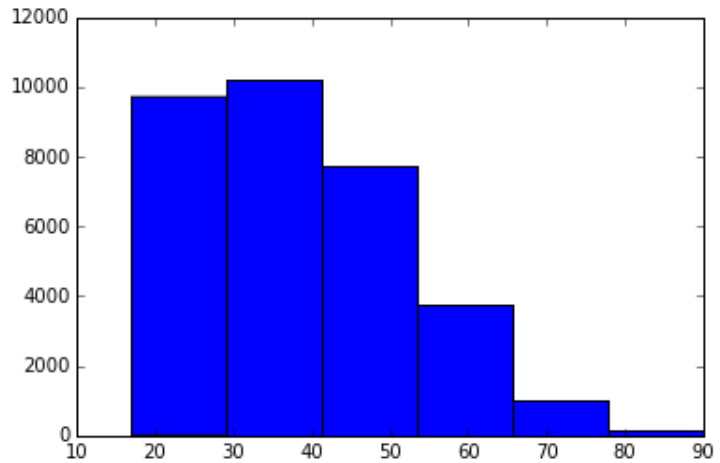
dataframe_blobdata.describe()

dataframe_blobdata['<column_name>'].value_counts().plot(kind='bar')
np.log(dataframe_blobdata['<column_name>']+1).hist(bins=50)
#correlation between column_a and column_b dataframe_blobdata[['<column_a>', '<column_b>']].corr()

```
In [7]: %matplotlib inline

In [8]: frame.age.hist(grid=False, bins=6)

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x9e427d0>
```



Services for data understanding

A side-by-side comparison of the capabilities and features

	Notebooks in Azure ML Workbench	Notebooks in Azure Databricks
Requires software installation	Yes	No
Execution environment	Local, HDInsight, Docker (local machine or remote Linux VM)	Azure Databricks only
Serverless service	No	Yes
Kernels supported	Python, PySpark	Spark
Languages supported	Python, SQL, Bash Shell	Python, Scala, R, SQL, Bash shell
Visualizations	Supports all standard Jupyter Notebook visualizations as well as libraries like Matplotlib	Provides extensive visualization library in addition to supporting third-party libraries
Supports RBAC	Yes, with Azure Active Directory	Yes, with Azure Active Directory
Collaborative workspaces	No	Yes, enables multiple users to work on same notebook in real-time and share artifacts
Run notebooks as scheduled jobs	No	Yes
Source control	Microsoft Team Services (Git repository)	GitHub, Bitbucket

Data understanding and prep

A side-by-side comparison of the capabilities and features

	Azure ML Workbench data prep	Azure Databricks
Requires software installation	Yes	No
Execution environment	Local, HDInsight, and Docker (local machine or remote Linux VM)	Azure Databricks
Supports defining transformations by example	Yes, via PROSE	No
Provides GUI-driven experience for defining data source	Yes	No
Provides ready-made visualizations to quickly assess data quality	Yes	No
Prepared data accessible via notebooks	Yes	Yes
Can be used against large data sets	Yes	Yes
Data movement required	Yes	No
Supported data sources	SQL Server, SQL DB, local file or directory, files in Azure Blob Storage. CSV, JSON, and Parquet formats	Integrated with Azure Blob Storage and Azure Data Lake Store, SQL DW, Cosmos DB and Kafka on HDInsight. Supports all file formats available to Spark (CSV, JSON, Parquet, Orc, and others).

Sample data

Why sample data?

If the dataset you plan to analyze is large, it's usually a good idea to down-sample the data to reduce it to a smaller but representative and more manageable size. This facilitates data understanding, exploration, and feature engineering.

Pandas

```
# A 1 percent sample
sample_ratio = 0.01
sample_size = np.round(dataframe_blobdata.shape[0] * sample_ratio)
sample_rows = np.random.choice(dataframe_blobdata.index.values, sample_size)
dataframe_blobdata_sample = dataframe_blobdata.ix[sample_rows]
```

SQL

```
select  * from <table_name> where <primary_key> in
(select top 10 percent <primary_key> from <table_name> order by newid())
```

Modeling

Flavors of machine learning

Supervised: Supervised learning algorithms make predictions based on a set of examples. There are several specific types of supervised learning that are represented within Azure Machine Learning: classification, regression, and anomaly detection.

Unsupervised: In unsupervised learning, data points have no labels associated with them. Instead, the goal of an unsupervised learning algorithm is to organize the data in some way or to describe its structure. This can mean grouping it into clusters or finding different ways of looking at complex data so that it appears simpler or more organized.

Reinforcement learning: In reinforcement learning, the algorithm gets to choose an action in response to each data point. The learning algorithm also receives a reward signal a short time later, indicating how good the decision was. Based on this, the algorithm modifies its strategy in order to achieve the highest reward.

The 5 questions data science answers

It might surprise you, but *there are only five questions that data science answers*:

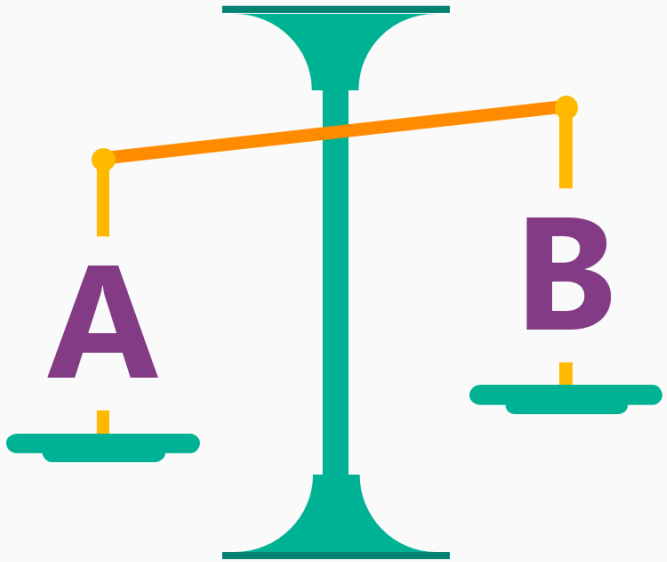
- **Is this A or B? - Classification**
- **Is this weird? –Anomaly Detection**
- **How much – or – How many? Regression**
- **How is this organized? - Clustering**
- **What should I do next? - ML Reinforcement**

Each one of these questions is answered by a separate family of machine learning methods, called algorithms.

Question 1: Is this A or B? uses classification algorithms

Is this A or B?

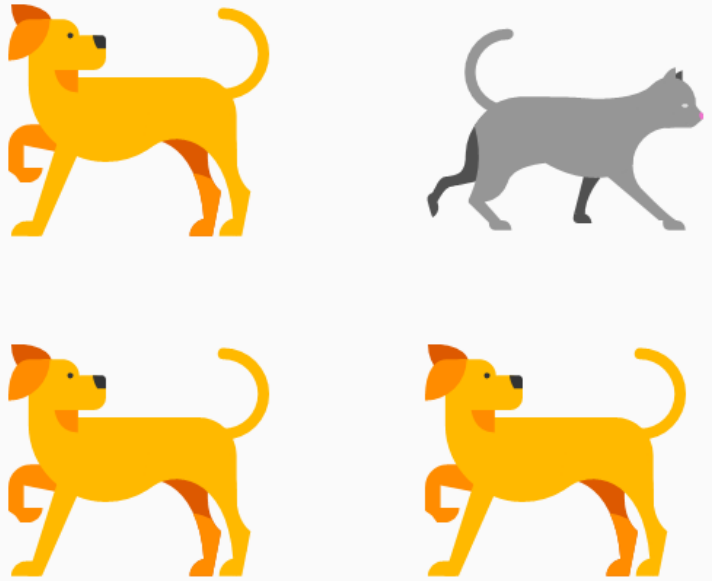
Classification algorithms



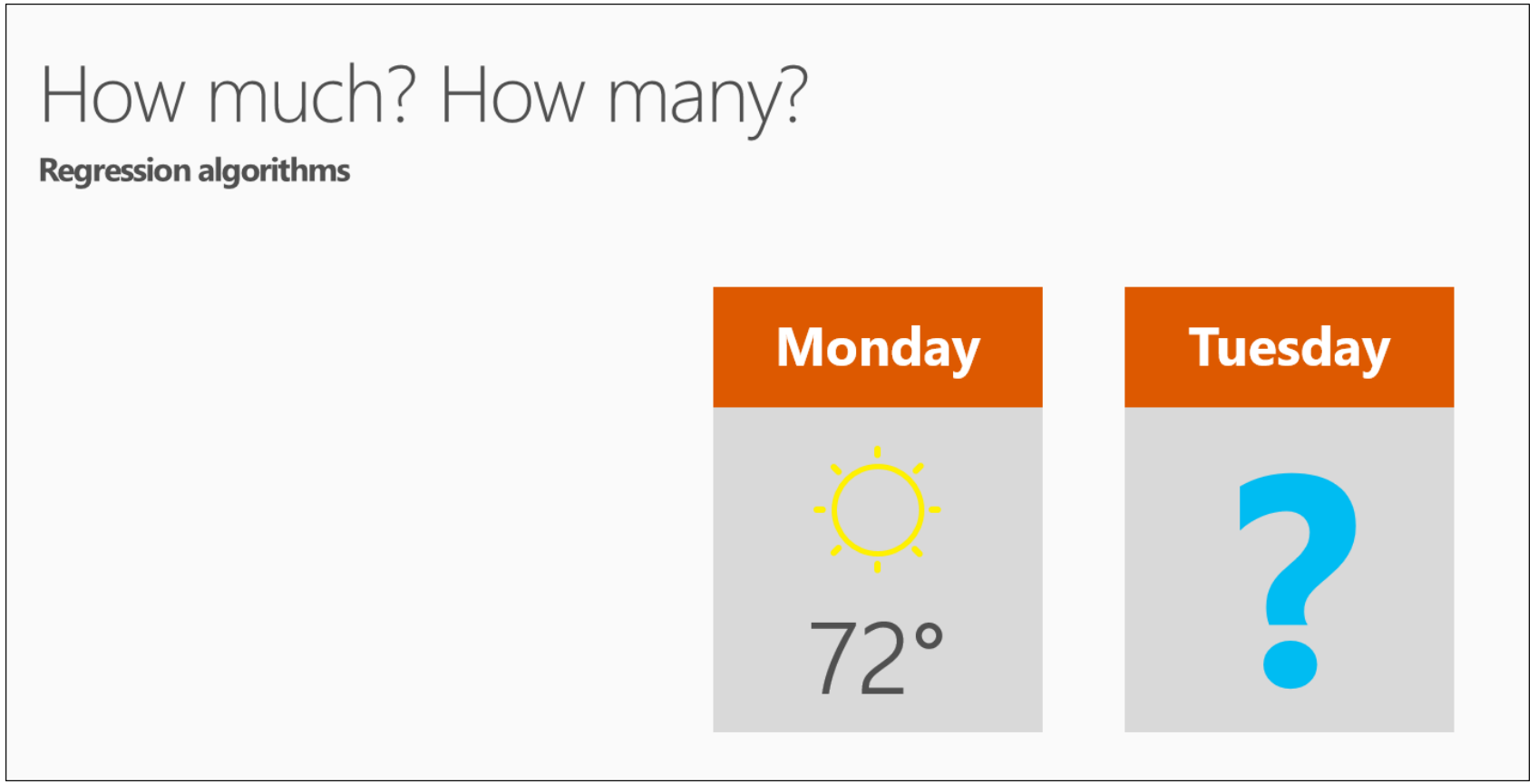
Question 2: Is this weird? uses anomaly detection algorithms

Is this weird?

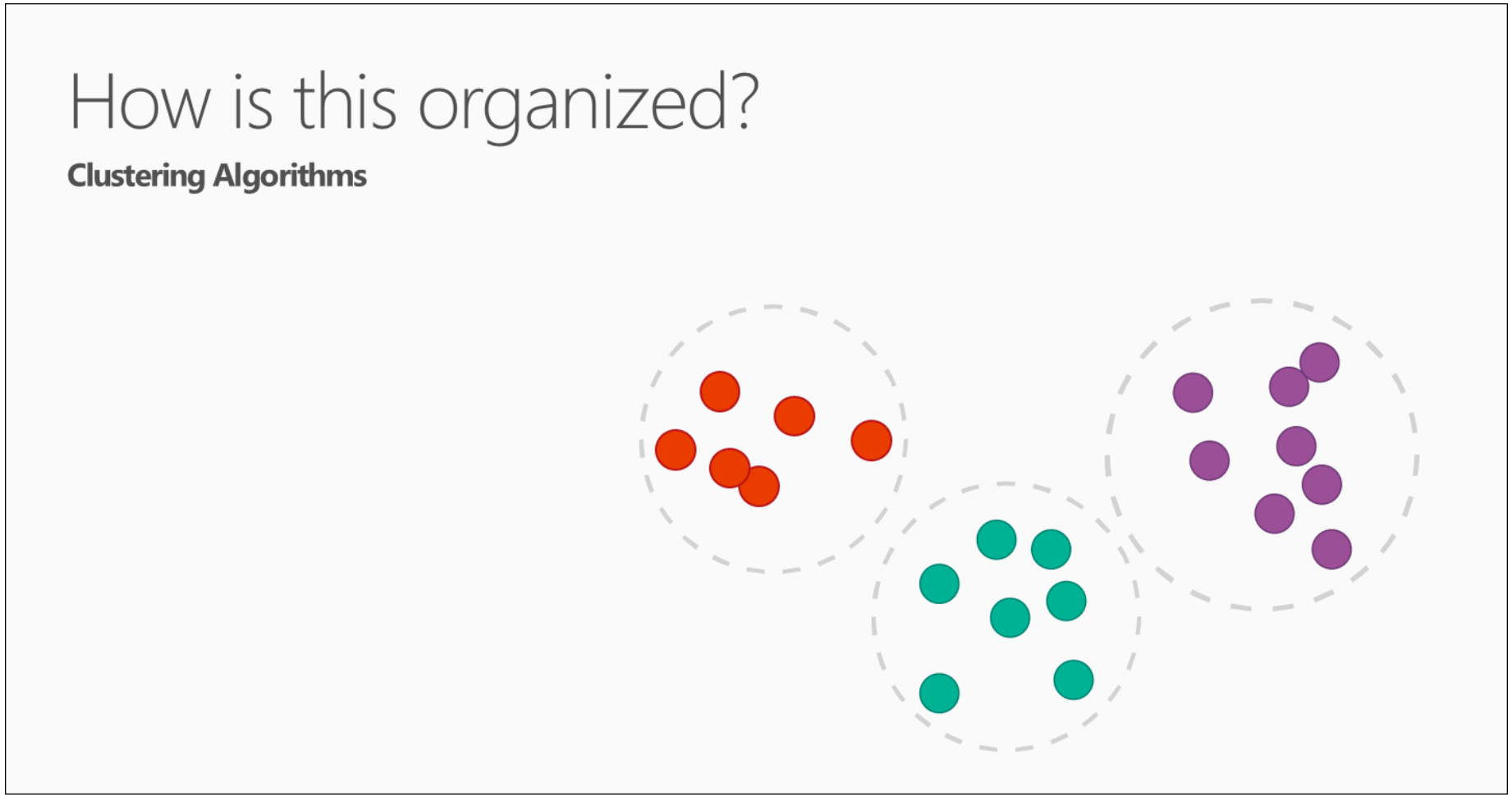
Anomaly detection algorithms



Question 3: How much? or How many? uses regression algorithms



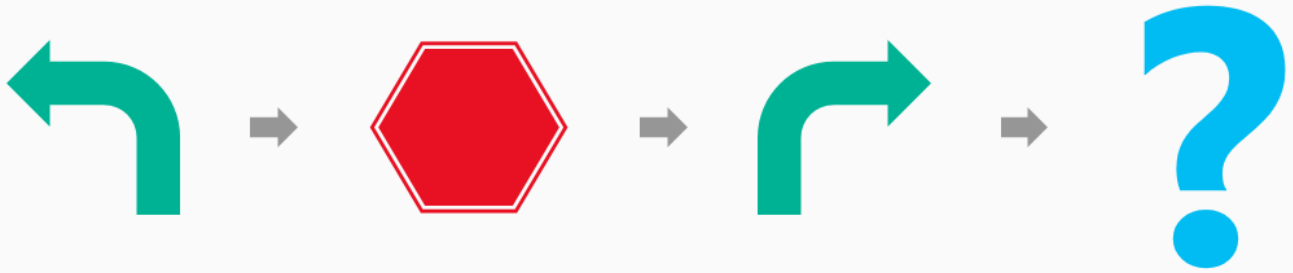
Question 4: How is this organized? uses clustering algorithms



Question 5: What should I do now? uses reinforcement learning algorithms

What should I do now?

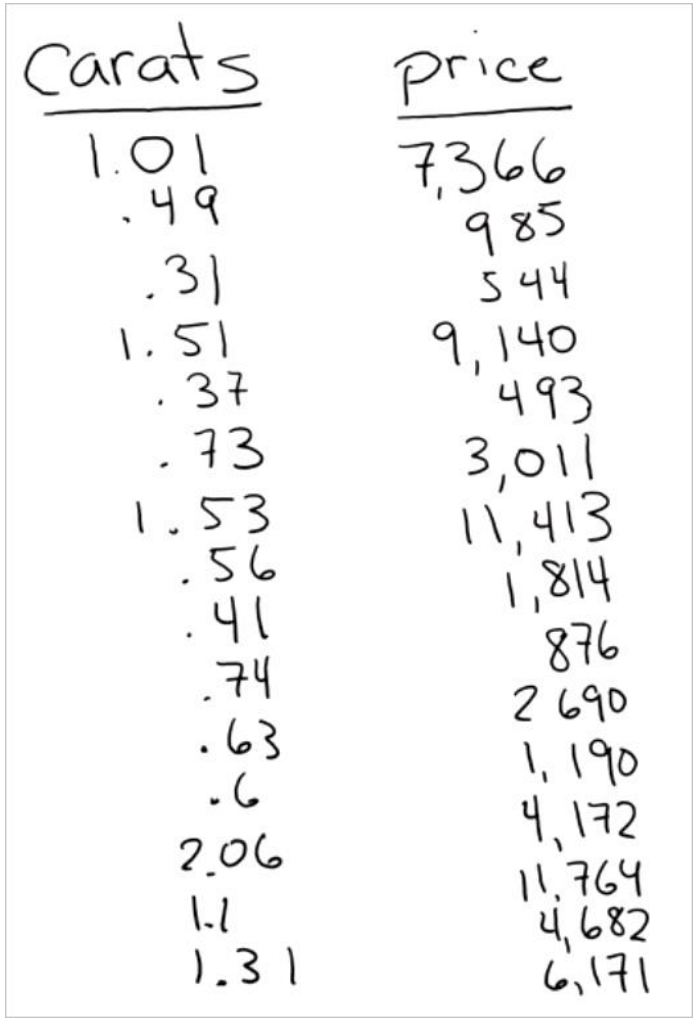
Reinforcement Learning Algorithms



Predict an answer with a simple model

Collect relevant, accurate, connected, enough data

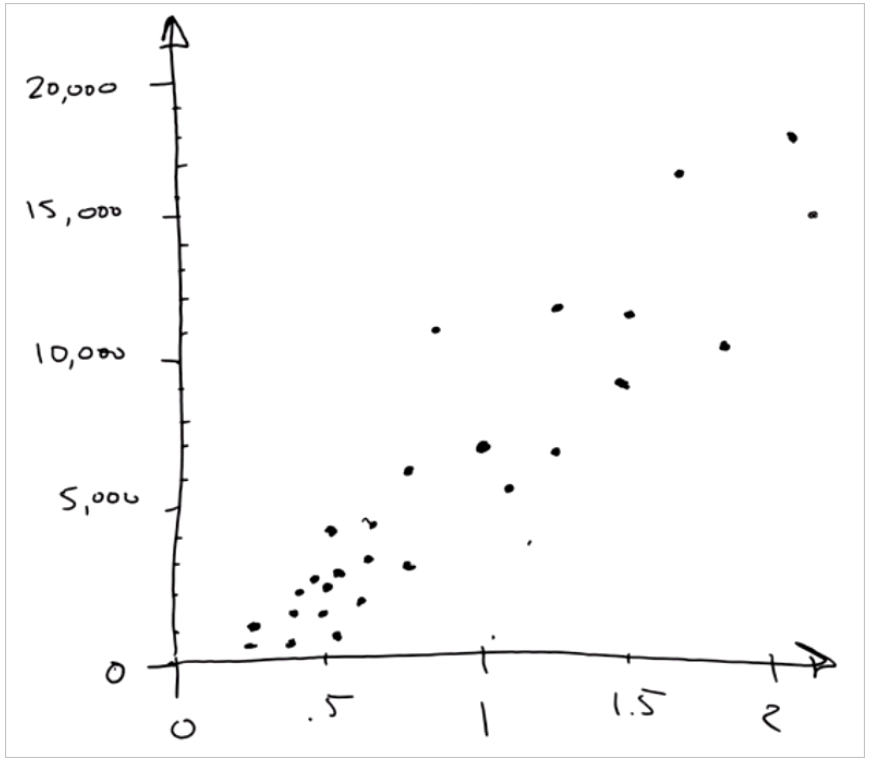
Say I want to shop for a diamond. I have a ring that belonged to my grandmother with a setting for a 1.35 carat diamond, and I want to get an idea of how much it will cost. I take a notepad and pen into the jewelry store, and I write down the price of all of the diamonds in the case and how much they weigh in carats. Starting with the first diamond - it's 1.01 carats and $7,366.

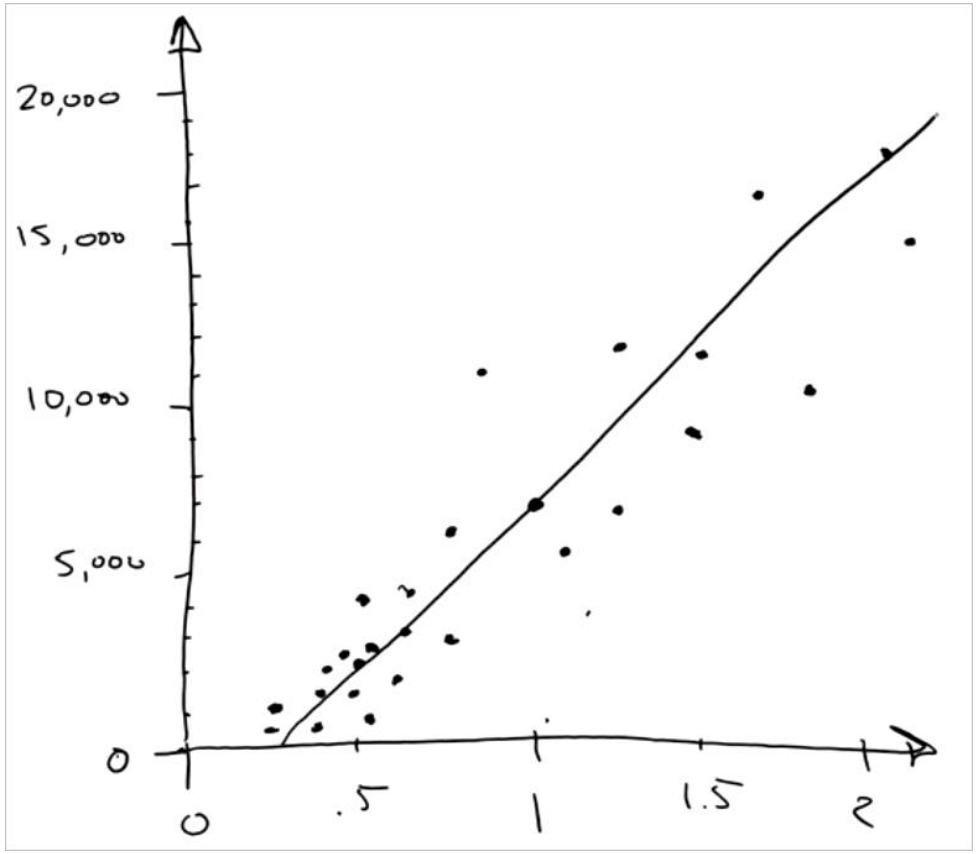


Carats	Price
1.01	7,366
.49	985
.31	544
1.51	9,140
.37	493
.73	3,011
1.53	11,413
.56	1,814
.41	876
.74	2690
.63	1,190
.6	4,172
2.06	11,764
1.1	4,682
1.31	6,171

- The data is **relevant** - weight is definitely related to price
- It's **accurate** - we double-checked the prices that we write down
- It's **connected** - there are no blank spaces in either of these columns
- And, as we'll see, it's **enough** data to answer our question

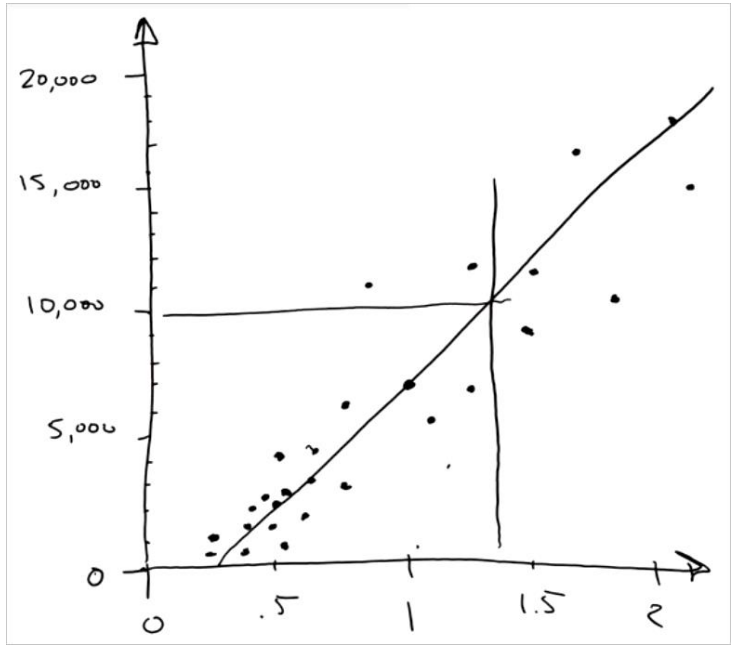
Now we'll pose our question in a sharp way: "How much will it cost to buy a 1.35 carat diamond?"+
Our list doesn't have a 1.35 carat diamond in it, so we'll have to use the rest of our data to get an answer to the question.

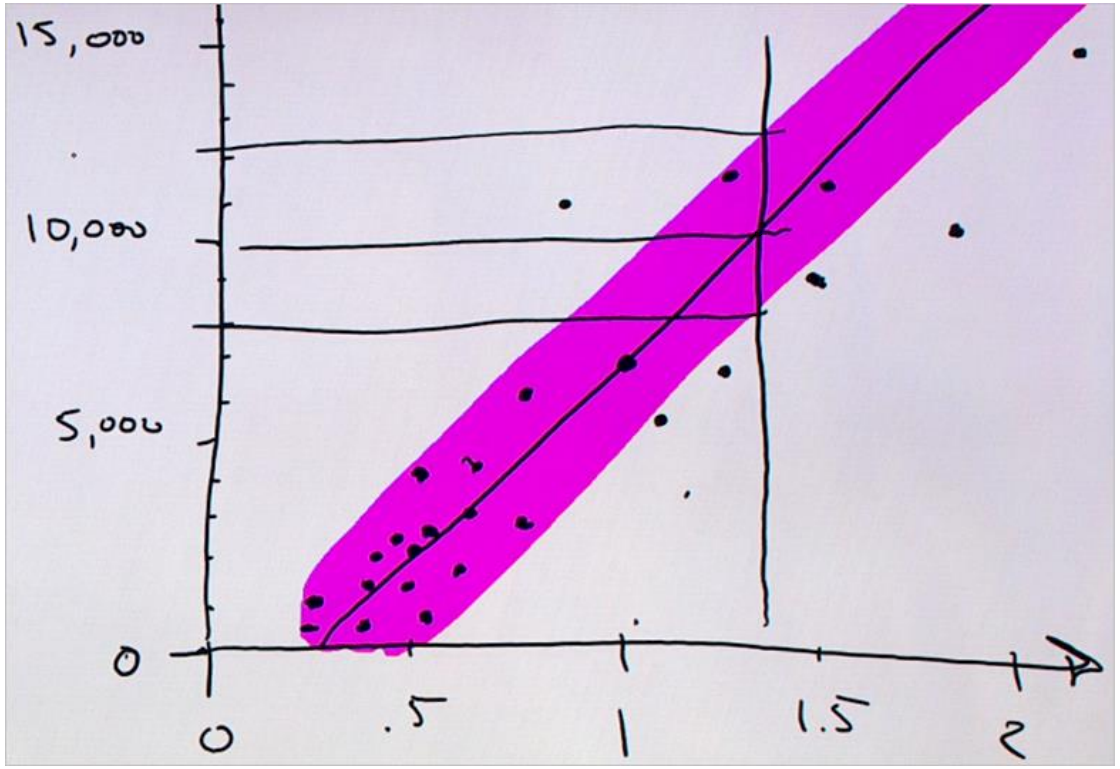




Use the model to find the answer

To answer our question, we eyeball 1.35 carats and draw a vertical line. Where it crosses the model line, we eyeball a horizontal line to the dollar axis. It hits right at 10,000. Boom! That's the answer: A 1.35 carat diamond costs about $10,000



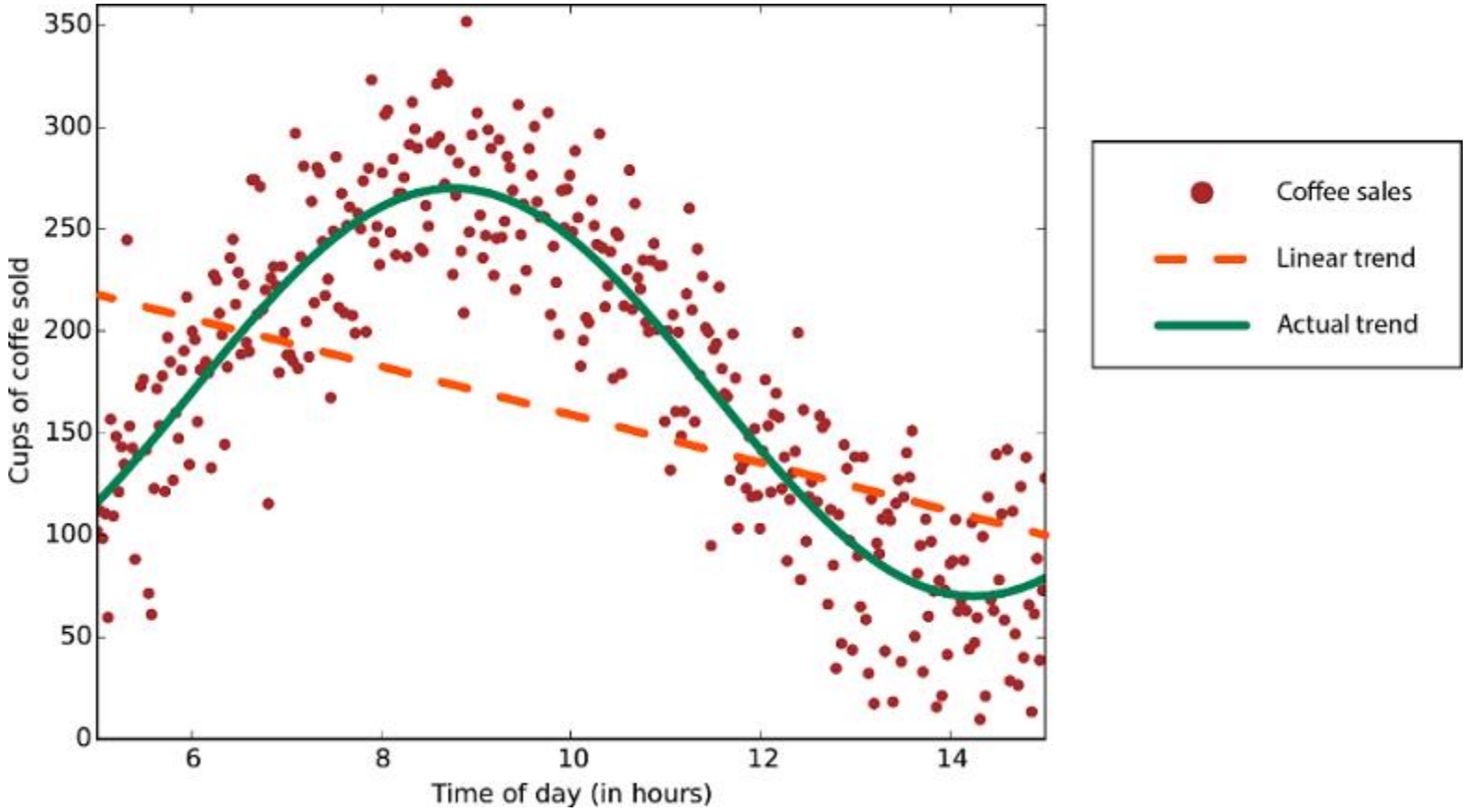


Now we can say something about our confidence interval: We can say confidently that the price of a 1.35 carat diamond is about $10,000 - but it might be as low as $8,000 and it might be as high as $12,000.

How to choose algorithms for Microsoft Azure Machine Learning

Considerations when choosing an algorithm

- **Accuracy**
- **Training Time**
- **Linearity**
- **Number of Parameters**
- **Number of Features**



Data with a nonlinear trend *- using a linear regression method would generate much larger errors than necessary*

Algorithm	Accuracy	Training time	Linearity	Parameters	Notes
Two-class classification					
logistic regression		●	●	5	
decision forest	●	○		6	
decision jungle	●	○		6	Low memory footprint
boosted decision tree	●	○		6	Large memory footprint
neural network	●			9	Additional customization is possible
averaged perceptron	○	○	●	4	
support vector machine		○	●	5	Good for large feature sets
locally deep support vector machine	○			8	Good for large feature sets
Bayes' point machine		○	●	3	

Multi-class classification					
logistic regression		●	●	5	
decision forest	●	○		6	
decision jungle	●	○		6	Low memory footprint
neural network	●			9	Additional customization is possible
one-v-all	-	-	-	-	See properties of the two-class method selected

Algorithm properties:

● - shows excellent accuracy, fast training times, and the use of linearity

○ - shows good accuracy and moderate training times

Regression					
linear		●	●	4	
Bayesian linear		○	●	2	
decision forest	●	○		6	
boosted decision tree	●	○		5	Large memory footprint
fast forest quantile	●	○		9	Distributions rather than point predictions
neural network	●			9	Additional customization is possible
Poisson			●	5	Technically log-linear. For predicting counts
ordinal				0	For predicting rank-ordering

Anomaly detection					
support vector machine	○	○		2	Especially good for large feature sets
PCA-based anomaly detection		○	●	3	
K-means		○	●	4	A clustering algorithm

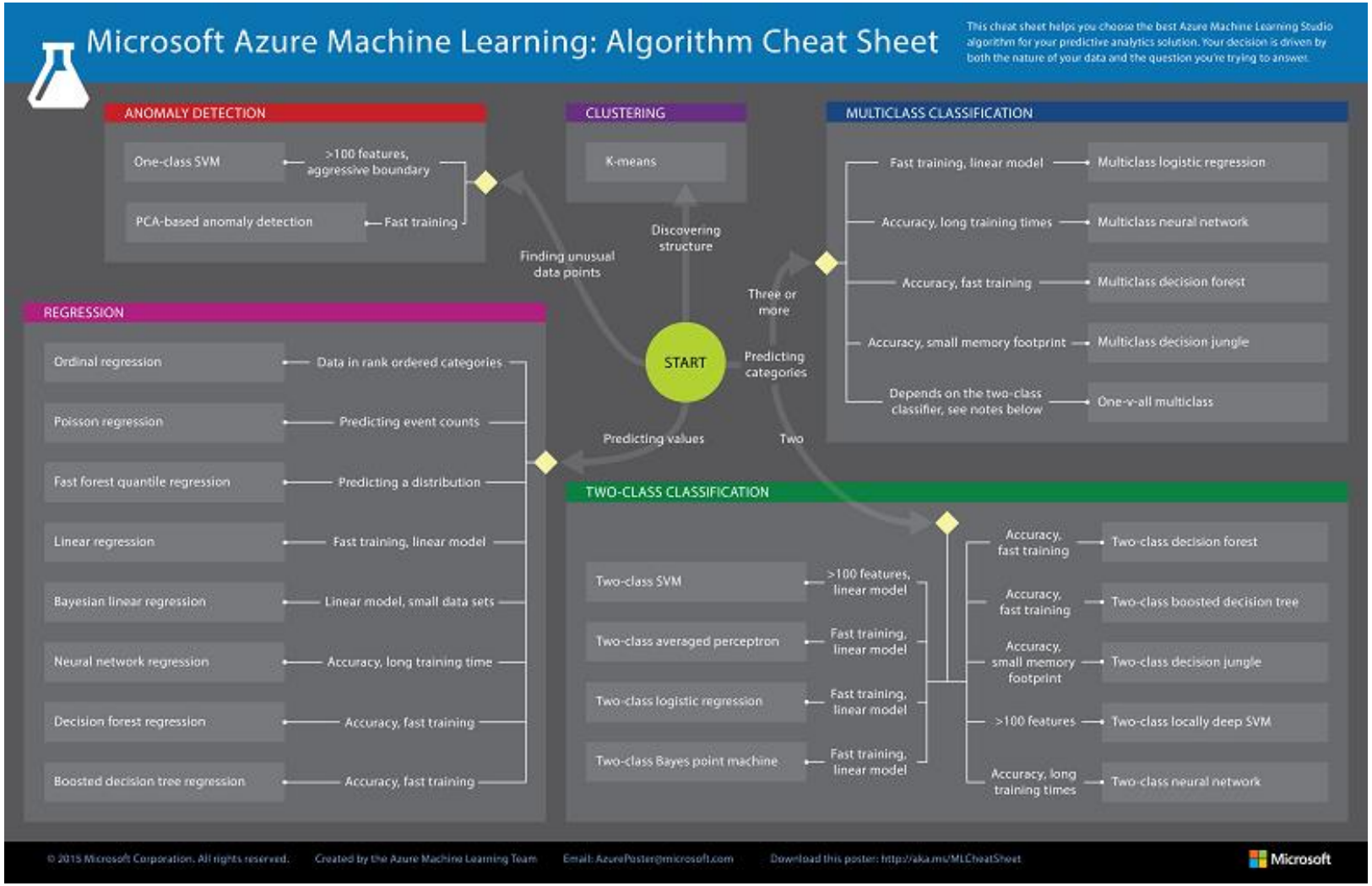
Algorithm properties:

● - shows excellent accuracy, fast training times, and the use of linearity

○ - shows good accuracy and moderate training times

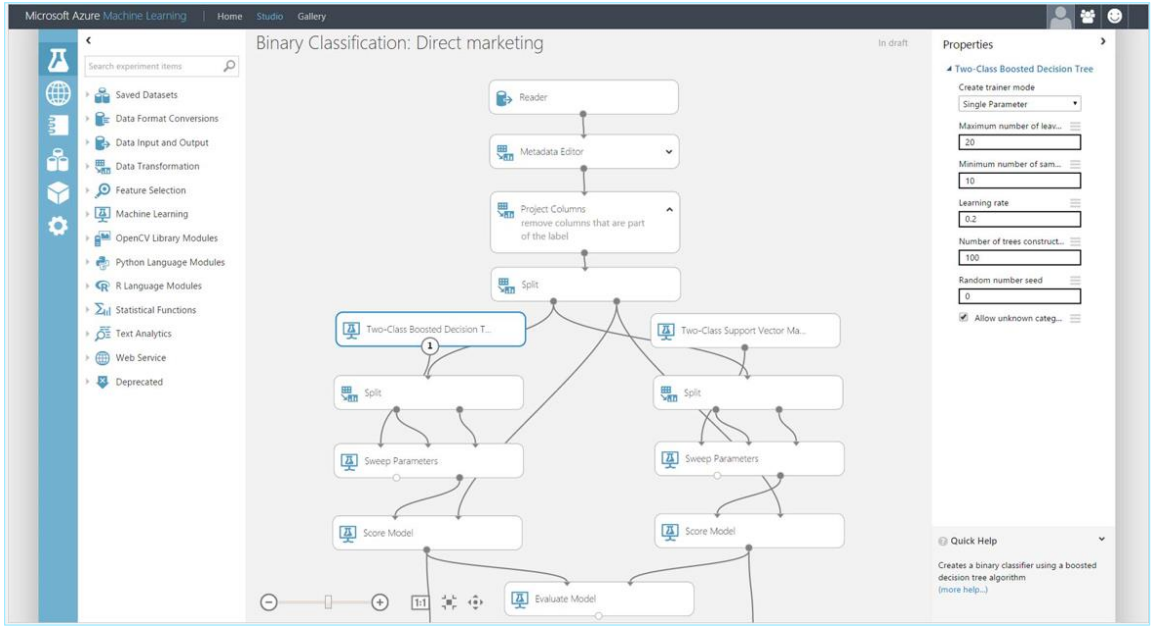
Microsoft Azure Machine Learning Algorithm Cheat Sheet

The Microsoft Azure Machine Learning Algorithm Cheat Sheet helps you choose the right algorithm for a predictive analytics model.



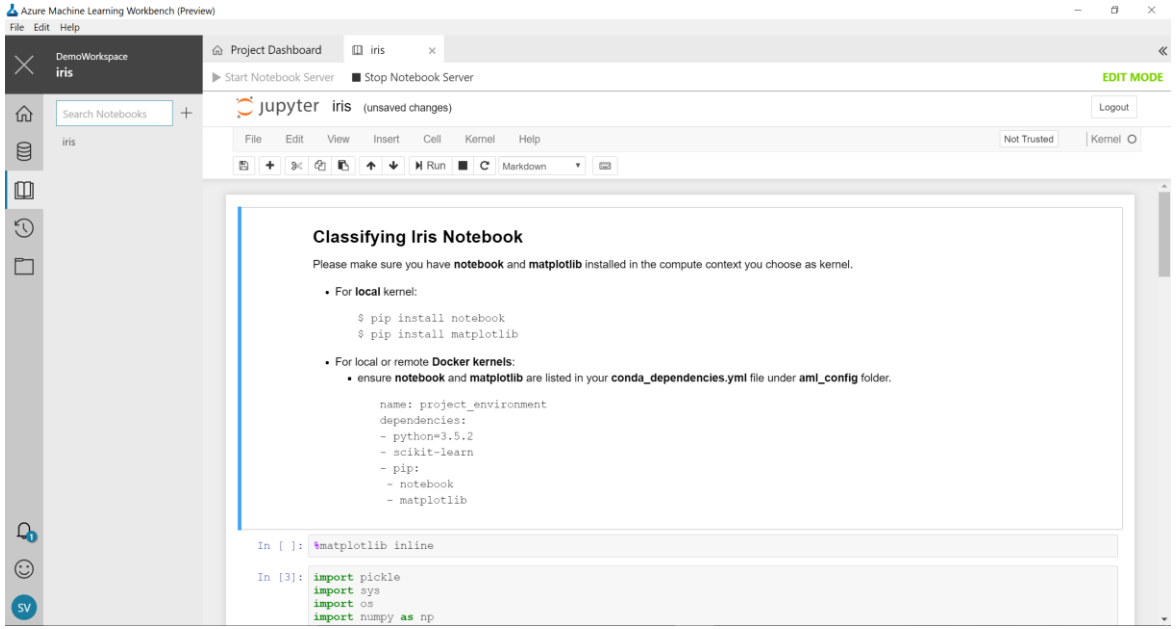
Build as you like

Azure machine learning studio



VISUAL DRAG-AND-DROP

Azure machine learning



CODE-FIRST

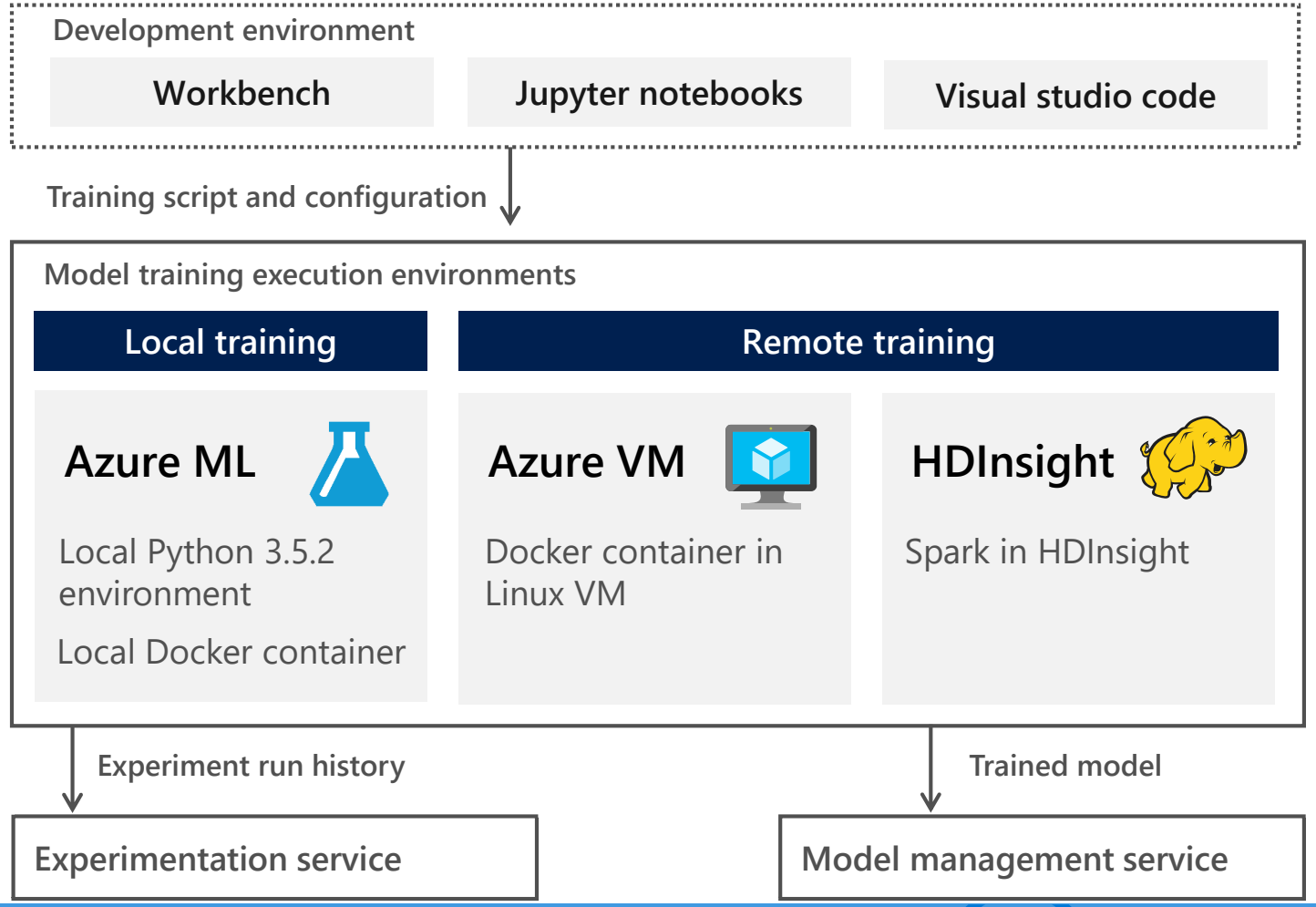
Modeling with Azure machine learning

Author Python training scripts using Jupyter Notebooks provided within the Azure Machine Learning Workbench or with Visual Studio Code.

Execute the training script on-premises or on a remote VM machine or HDInsight cluster.

Experimentation Service handles execution of ML experiments across environments, Git integration, access control, project roaming and sharing, and records run history information.

Model Management Service tracks model versions and lineage across training runs. Models are stored, registered, and managed in the cloud.



Azure machine learning studio

Browser based, no coding, graphical tool to build predictive analytics apps

Serverless Azure service

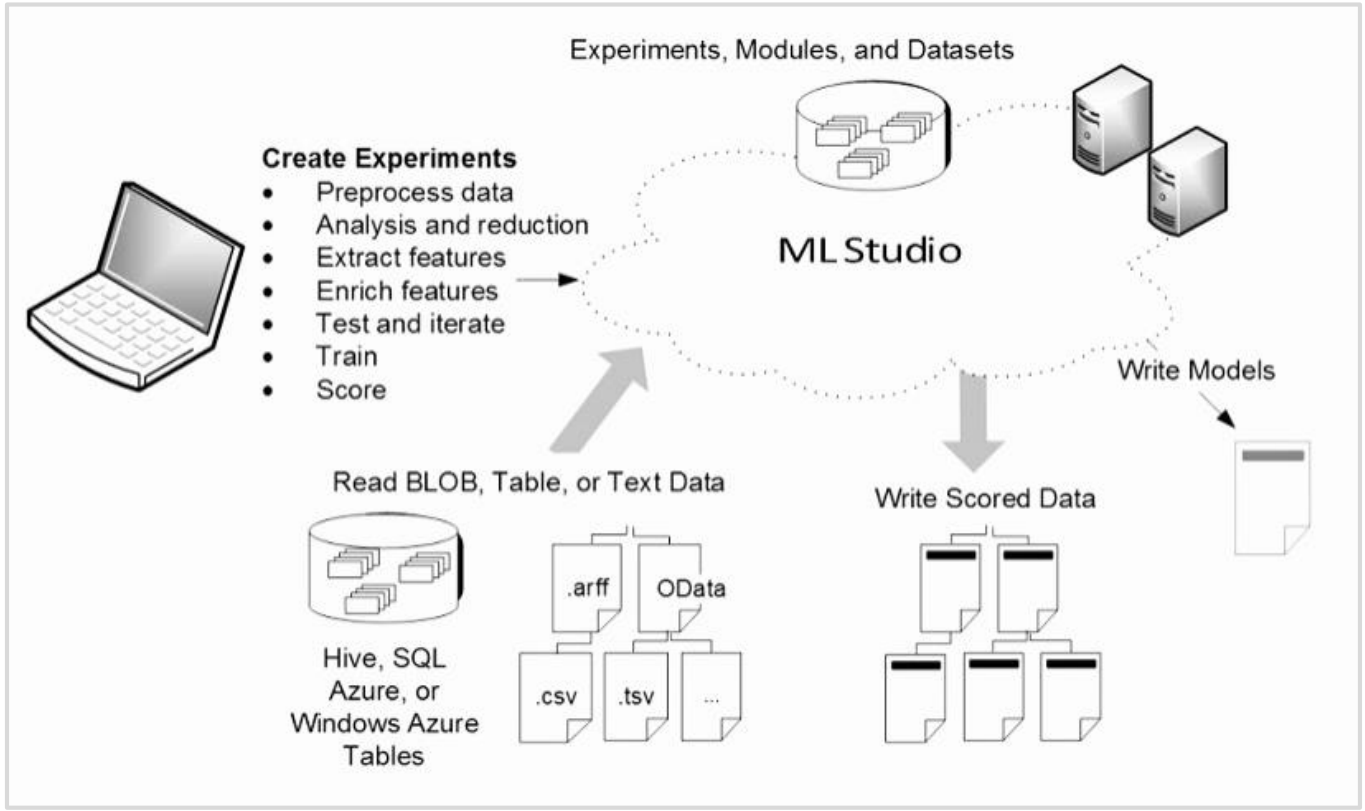
99.95 percent SLA

Supports multiple algorithms, data sources, and data formats

Trained models can be deployed as Request-Response Service (RRS) or Batch Execution Service (BES)

Gallery includes community contributed solutions

Microsoft Azure Machine Learning Studio Capabilities Overview



Other machine learning products and services from Microsoft

In addition to Azure Machine Learning, there are a variety of options at Microsoft to build, deploy, and manage your machine learning models.

- SQL Server Machine Learning Services: enables you to run, train, and deploy machine learning models using R or Python. You can use data located on-premises and in SQL Server databases.
- Microsoft Machine Learning Server: is an enterprise server for hosting and managing parallel and distributed workloads of R and Python processes.
- Azure Data Science Virtual Machine
- Spark MLLib in HDInsight
- Batch AI Training Service
- Microsoft Cognitive Toolkit (CNTK)
- Azure Cognitive Services

Choosing Azure machine learning

When Azure Machine Learning can be a good option for modeling

When you want...	Description
Choice of code-first or drag-and-drop training environments	Azure ML can be used to train models using Python scripts or by designing graphical flows within Azure ML Studio.
To train models against any scale of data	Azure ML training can be performed on small datasets where the training runs a single machine and scale to large datasets stored in HDFS and trained using an HDInsight Spark cluster. Models trained using Azure ML Studio require datasets smaller than 10 GB.
Choice in performing training on-premises or in the cloud	Azure ML doesn't require that model training occur in Azure. Azure ML supports executing training experiments directly on a local machines, in any Linux Docker container (on-premises or running in a VM in Azure), or using an HDInsight Spark cluster. Models trained using Azure ML Studio can execute only in Azure.
To train models using the libraries, frameworks, and tools you already use	Azure ML provides first class support for authoring training scripts in Python and using Spark MLlib, Cognitive Toolkit, TensorFlow, and Scikit-Learn. Visual Studio Code and Jupyter Notebooks are supported environments for authoring training scripts.
Versioning and reproducibility	Azure ML experiments are automatically versioned in a Git repository backed by Team Services. Model configuration, parameters and performance statistics are automatically captured to the Azure ML Experimentation Service, so you have both a reproducible history of experiments and the performance of each experiment run.

Choosing Azure machine learning

When Azure Databricks can be a good option for modeling

When you want...	Description
A frictionless experience for Spark on Azure	Databricks was built by the team who originally created Spark and worked with Microsoft to deliver it as a first-party service on Azure. It includes native and performance-optimized integration with other Azure services such as SQL DW, Cosmos DB, Azure Storage, and Azure Data Lake Store.
To perform modeling using notebooks and Spark	Best-in-class notebooks experience for optimal productivity and collaboration
A solution that simplifies resource management	Azure Databricks provides auto-scaling, cluster termination capabilities, and the ability to leverage serverless pools.
A unified platform that includes deep learning	Azure Databricks provides a single engine for batch, streaming, machine learning, and graph processing. Deep learning libraries/frameworks like the Microsoft Cognitive Toolkit (CNTK), TensorFlow, and BigDL are supported. Azure Databricks clusters don't have GPUs at this time.
Simplified authentication and access control	Azure Databricks provides native integration with Azure Active Directory for authentication and provides granular role-based access control (RBAC) for workspaces, notebooks, jobs, clusters and REST APIs.

Choosing machine learning services (in-database)

When SQL Server Machine Learning Services (in-database) can be a good option for modeling

When you want…	Description
To train a model using data that already exists in SQL Server	Performing the model training on the same database server that contains the training data has the benefits of removing any need to shuffle data between the source and the server executing the training. This simplifies setup and can speed model training. With Machine Learning Services—in databases—for SQL Server, you can perform model training in an on-premises SQL Server or one running in an Azure VM.
To train on data that might not fit in memory of a single SQL server	The SQL Server Enterprise Edition provides optimized performance through parallelization and streaming—where the input data doesn't all need to fit into memory, but is streamed.
To leverage features from the Microsoft Machine Learning Library	In addition to providing fast, parallelized algorithms and deep learning tools, the Microsoft Machine Learning Library included with Machine Learning Server provides a set of pre-trained sentiment models and image featurizers.
To author parallel training scripts in R or Python	Machine Learning Server includes a rich set of highly-scalable, parallelized algorithms such as RevoscaleR for modeling in R, revoscalepy for modeling in Python, and MicrosoftML for modeling in both Python and R within the context of a T-SQL stored procedure.

Choosing Machine Learning Server

When Machine Learning Server can be a good option for modeling

When you want...	Description
To perform modeling exclusively on-premises, without a cloud requirement	Machine Learning Server enables you to train against on-premises clusters on Hadoop/Spark running Linux, or with dedicated servers running Windows Server or Linux. No resources are needed in Azure in this situation.
To perform analytics in a multi-threaded or distributed fashion	Machine Learning Server enables training on a single server to be multi-threaded so that multiple cores can be used with parallelized algorithms. When running in a cluster, training can be spread across nodes in the cluster.
To leverage features from the Microsoft Machine Learning Library	In addition to providing fast, parallelized, and distributed algorithms and deep learning tools, the Microsoft Machine Learning Library included with Machine Learning Server provides a set of pre-trained sentiment models and image featurizers.
To author parallel and distributed training scripts in R or Python	Machine Learning Server includes a rich set of highly-scalable, distributed set of algorithms such as RevoscaleR for modeling in R, revoscalepy for modeling in Python, and MicrosoftML for modeling in both Python and R.
Active Directory (LDAP) integration	Machine Learning Server integrates seamlessly with Active Directory and includes role-based access control to satisfy the security and compliance needs of your enterprise.

Choosing Data Science Virtual Machine

When the Data Science Virtual Machine can be a good option for modeling

When you want...	Description
An easy to deploy, comprehensive data science environment	The goal of the Data Science Virtual Machine is to provide data professionals at all skill levels and roles with a friction-free data science environment. This VM saves you considerable time that you would spend if you had rolled out a comparable environment on your own.
Short-term experimentation and evaluation	The Data Science Virtual Machine can be used to evaluate or learn tools such as Microsoft ML Server, SQL Server, Visual Studio tools, Jupyter, deep learning and machine learning toolkits, and new tools popular in the community with minimal setup effort.
Elastic capacity	Data science hackathons and competitions or large-scale data modeling and exploration require scaled-out hardware capacity, typically for short duration. The Data Science Virtual Machine can help replicate the data science environment quickly on demand, on scaled-out servers that allow experiments that need high-powered computing resources to be run.
GPU-powered deep learning	Data Science Virtual Machine can be used for training models using deep learning algorithms on GPU (graphics processing units)-based hardware. Utilizing VM scaling capabilities of Azure, Data Science Virtual Machine helps you use GPU-based hardware on the cloud. You can switch to a GPU-based VM when training large models or need high-speed computations while using the same OS disk.

Choosing Azure Batch AI

When Azure Batch AI can be a good option for modeling

When you want…	Description
A managed service for scale-out machine learning and deep learning	Batch AI is a managed service that enables data scientists and AI researchers to train AI and other machine learning models on clusters of Azure virtual machines, including VMs with GPU support. You describe the requirements of your job, where to find the inputs and store the outputs, and Batch AI handles the rest.
A solution that simplifies resource management	Batch AI supports automatic and manual scaling of VM clusters using GPUs or CPUs. It provides job status and handles restarting tasks in case of VM failures.
Lower training costs using low-priority VMs	Batch AI supports pools of VMs that can be a blend of dedicated VMs and low-priority VMs.
A scale-out solution for experimentation and training	Batch AI supports running long-running batch jobs, iterative experimentation, and interactive training.
A solution optimized for popular toolkits	Batch AI supports any deep learning or machine learning framework, and provides optimized configuration for popular toolkits such as Microsoft Cognitive Toolkit (CNTK), TensorFlow, and Chainer.

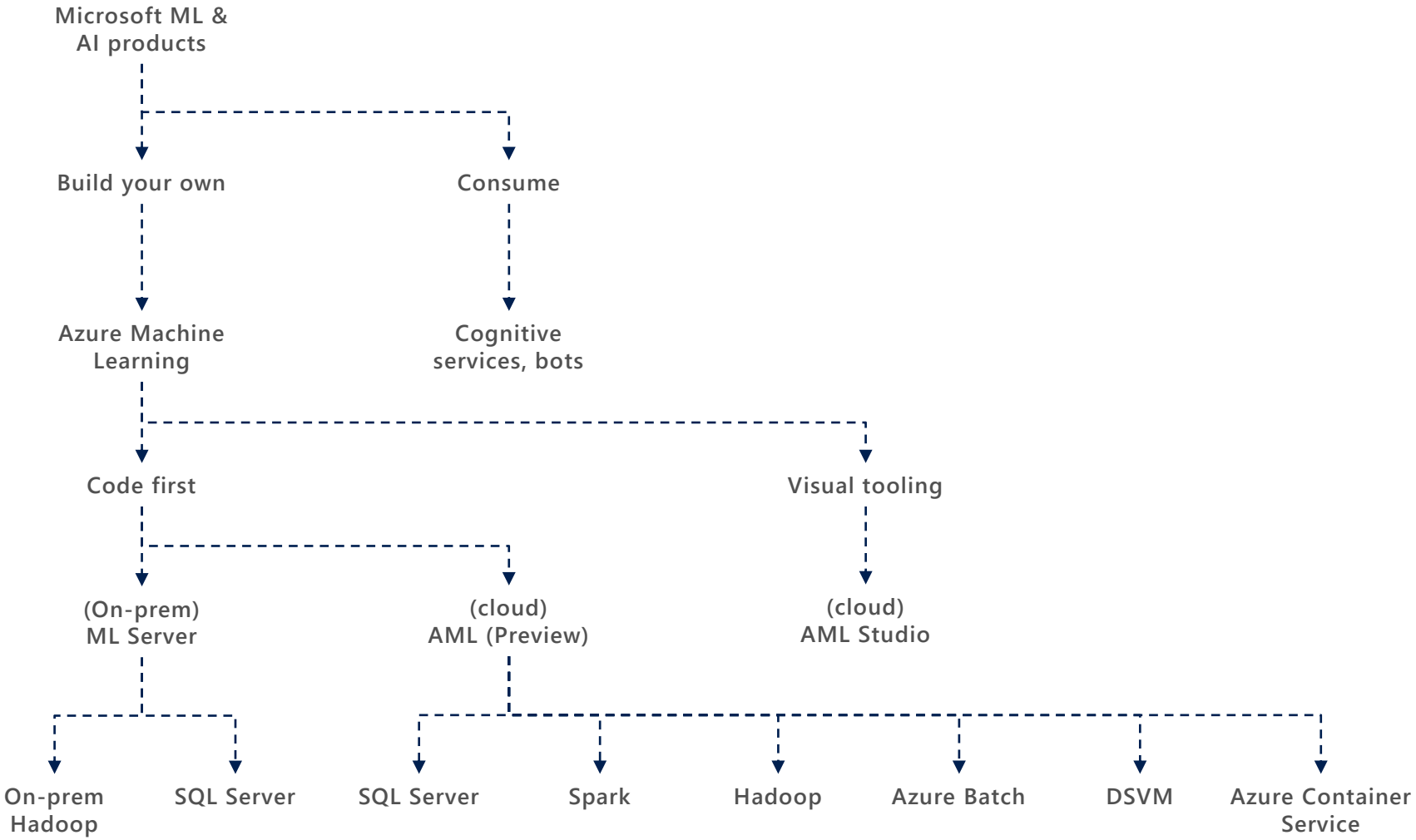
Machine Learning & AI Portfolio
When to use what?

Build your own or consume pre-trained models?

Which experience do you want?

Deployment target

What engine(s) do you want to use?

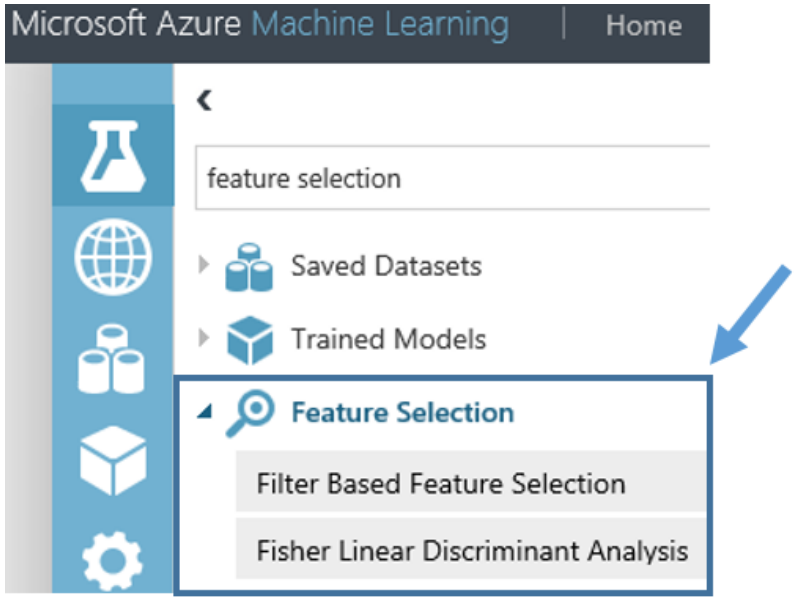


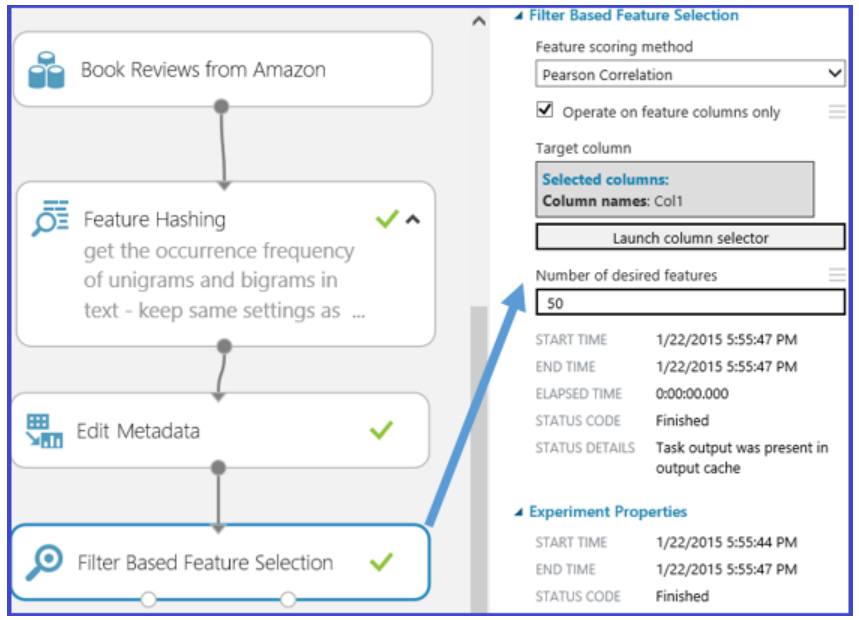
Optimization

Feature engineering in data science

- **feature engineering**: This process attempts to create additional relevant features from the existing raw features in the data, and to increase the predictive power of the learning algorithm.
- **feature selection**: This process selects the key subset of original data features in an attempt to reduce the dimensionality of the training problem.

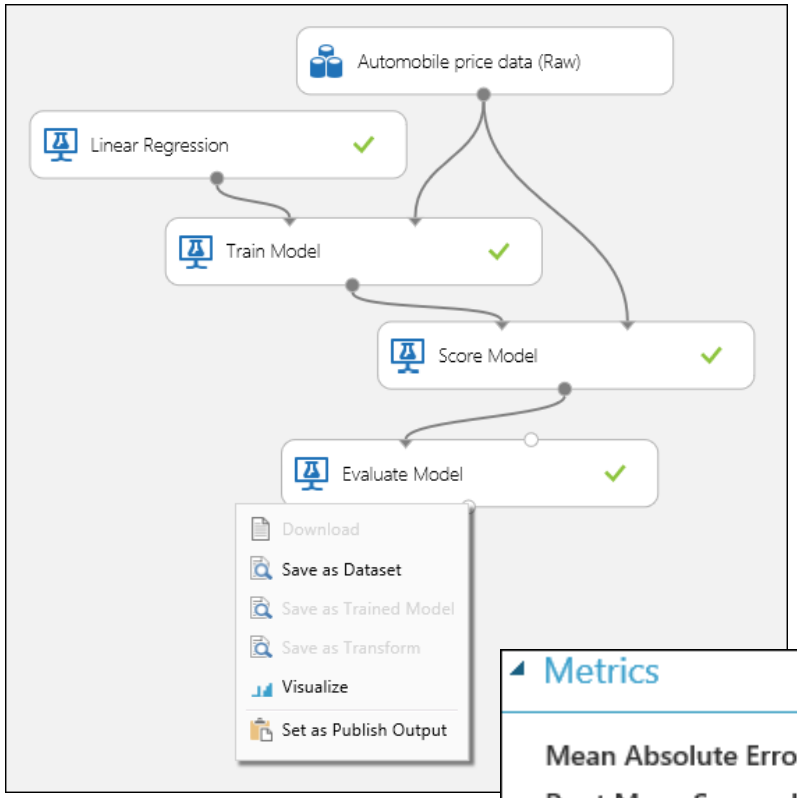
Normally **feature engineering** is applied first to generate additional features, and then the **feature selection** step is performed to eliminate irrelevant, redundant, or highly correlated features.





How to evaluate model performance in Azure Machine Learning Studio

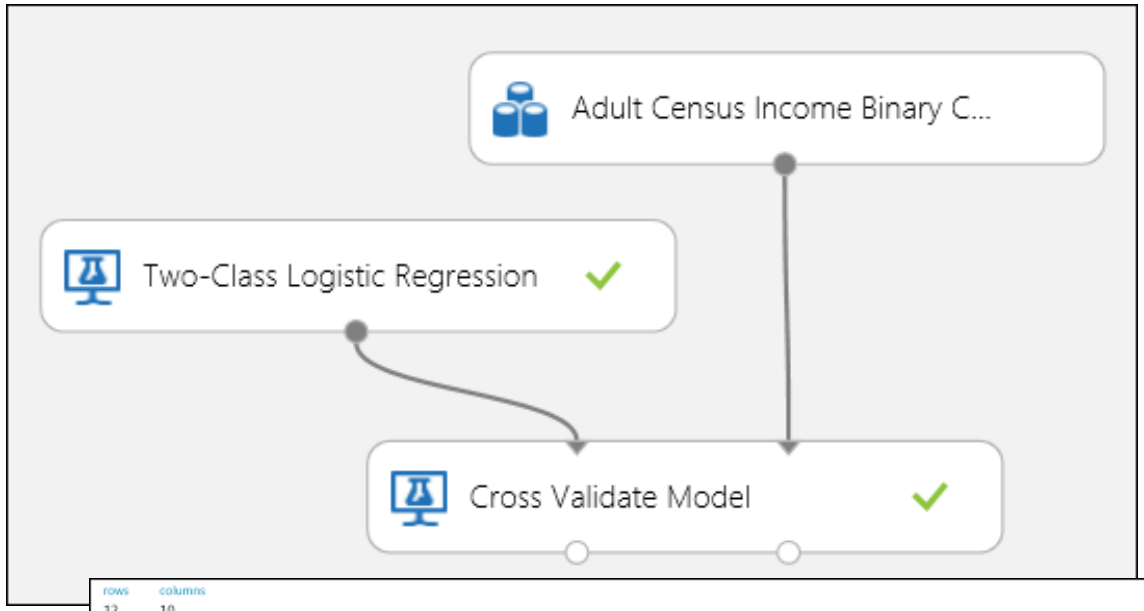
Evaluate



Metrics

Mean Absolute Error	747.975254
Root Mean Squared Error	955.587783
Relative Absolute Error	0.163528
Relative Squared Error	0.026598
Coefficient of Determination	0.973402

Cross-Validate



rows 12 columns 10

Fold Number	Number of examples in fold	Model	Accuracy	Precision	Recall	F-Score	AUC	Average Log Loss	Training Log Loss
0	3256	Logistic Regression	0.850399	0.737342	0.621333	0.674385	0.905809	0.325788	41.989626
1	3256	Logistic Regression	0.842245	0.707358	0.585062	0.640424	0.901871	0.32527	40.990733
2	3256	Logistic Regression	0.852583	0.767007	0.591864	0.668148	0.905203	0.32936	41.51439
3	3256	Logistic Regression	0.847333	0.729685	0.598639	0.657698	0.898848	0.333642	40.075913
4	3256	Logistic Regression	0.844305	0.741194	0.614213	0.671756	0.90569	0.332448	41.921163
5	3256	Logistic Regression	0.834601	0.716393	0.57199	0.636099	0.897829	0.339869	39.879498
6	3256	Logistic Regression	0.85	0.728188	0.598621	0.657078	0.902695	0.327404	40.596835
7	3257	Logistic Regression	0.846128	0.743464	0.599473	0.663749	0.90208	0.333052	41.153979
8	3256	Logistic Regression	0.846973	0.734861	0.601071	0.661267	0.902736	0.329349	41.259562
9	3256	Logistic Regression	0.849388	0.742718	0.607947	0.668609	0.903525	0.328219	41.623392
Mean	32561	Logistic Regression	0.846396	0.734821	0.599021	0.659921	0.902629	0.33044	41.100509
Standard Deviation	32561	Logistic Regression	0.005145	0.016276	0.014102	0.012778	0.00269	0.004408	0.725968

Choose parameters to optimize your algorithms in Azure Machine Learning

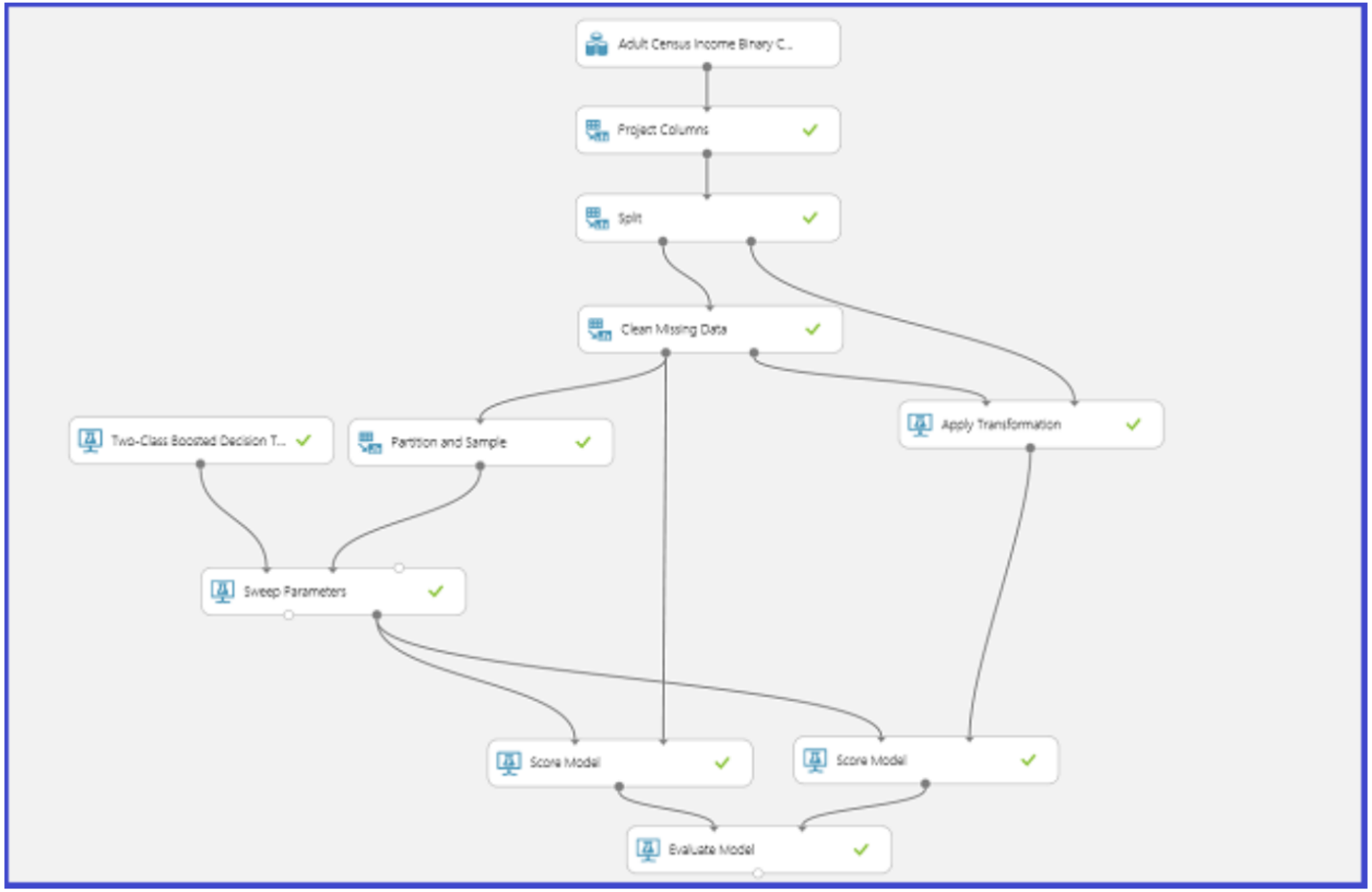
There are four steps in the process of finding the best parameter set:

Define the parameter space: For the algorithm, first decide the exact parameter values you want to consider.

Define the cross-validation settings: Decide how to choose cross-validation folds for the dataset.

Define the metric: Decide what metric to use for determining the best set of parameters, such as accuracy, root mean squared error, precision, recall, or f-score.

Train, evaluate, and compare: For each unique combination of the parameter values, cross-validation is carried out by and based on the error metric you define. After evaluation and comparison, you can choose the best-performing model.


Adult Census Income Binary C...
Project Columns
Split
Clean Missing Data
Two-Class Boosted Decision T...
Partition and Sample
Apply Transformation
Sweep Parameters
Score Model
Score Model
Evaluate Model

Deployment

Model deployment options

A side-by-side comparison of the capabilities and features

	Azure ML	SQL Database	Azure Databricks
Scoring interface provided	Web service	T-SQL stored procedure	Notebook or Job
Deployment environments	Scoring with the trained model can run locally, such as in a Jupyter Notebook. Containerized model can run anywhere that can run a Linux Docker Container. For example, Data Science VM, Azure Container Service, or IoT edge device. For Azure ML Studio web services, these must be hosted in Azure.	SQL Server 2017 database instance on-premises or in Azure VM	Azure Databricks instance
Scalability of scoring interface	When deployed in Azure to Azure Container Service, scales by deploying more instances	Limited to capacity of single server	Can scale across cluster resources
Scoring requirements	For Azure ML, requires creation of Docker image that contains scoring service, model and dependencies. Docker container must be pushed to Azure Container Registry for subsequent deployment by Azure Container Service. For Azure ML Studio, the experiment is published as a predictive web service thru the UI.	Need to author Python or R code within a T-SQL stored procedure that loads the trained model from a table where it is stored and applies it in scoring	Load the trained model from storage and apply to scoring in notebook in Python, Scala, R, or SQL
Model packaging	Docker image	Serialized to table	Serialized to storage

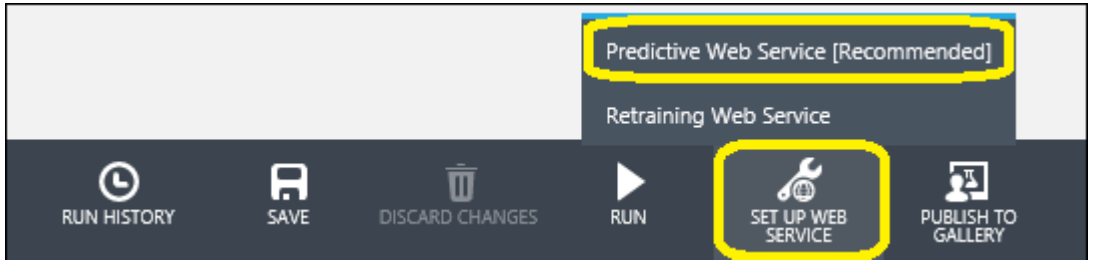
Deploy an Azure Machine Learning web service

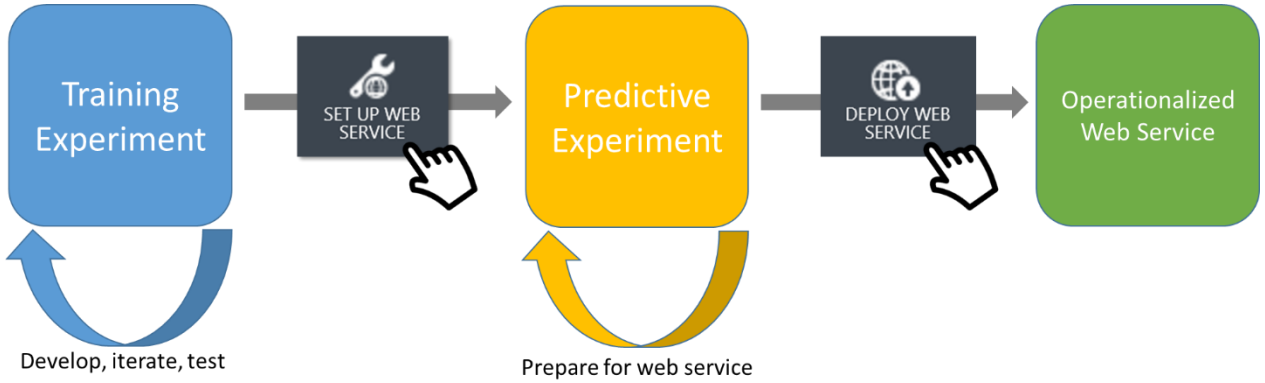
From a high-level point-of-view, this is done in three steps:

Create a training experiment - Azure Machine Learning Studio is a collaborative visual development environment that you use to train and test a predictive analytics model using training data that you supply.

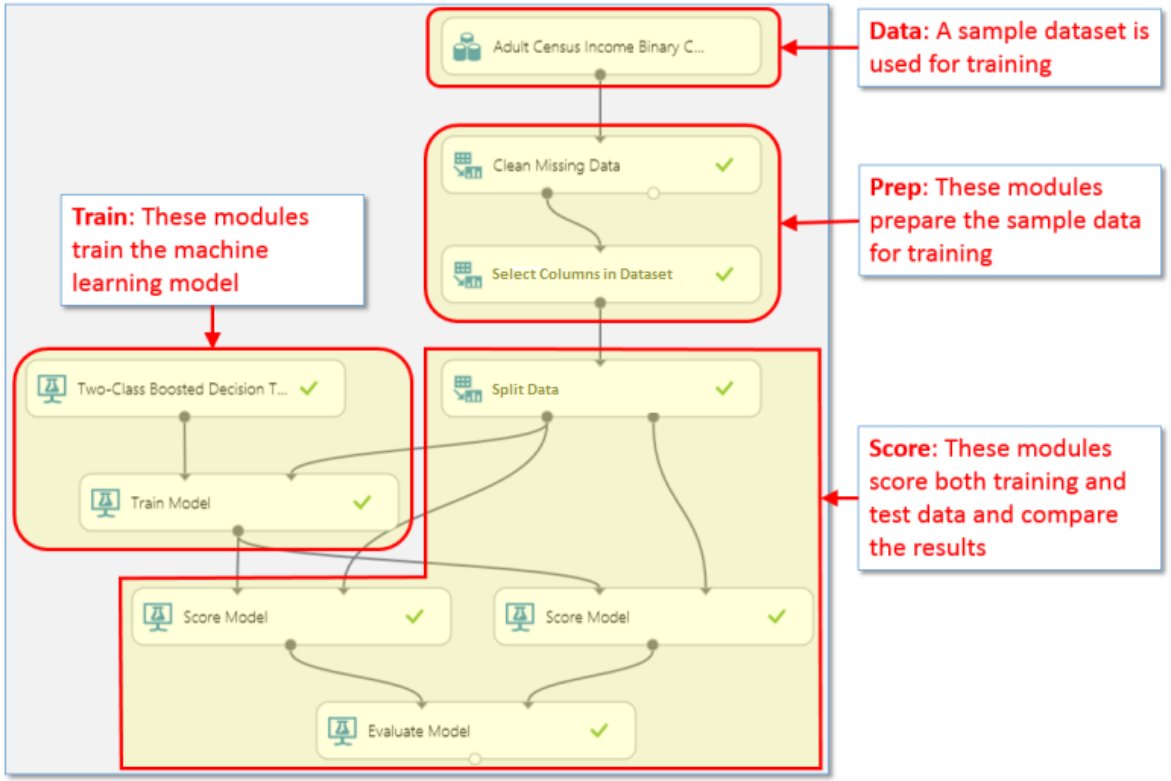
Convert it to a predictive experiment - Once your model has been trained with existing data and you're ready to use it to score new data, you prepare and streamline your experiment for predictions.

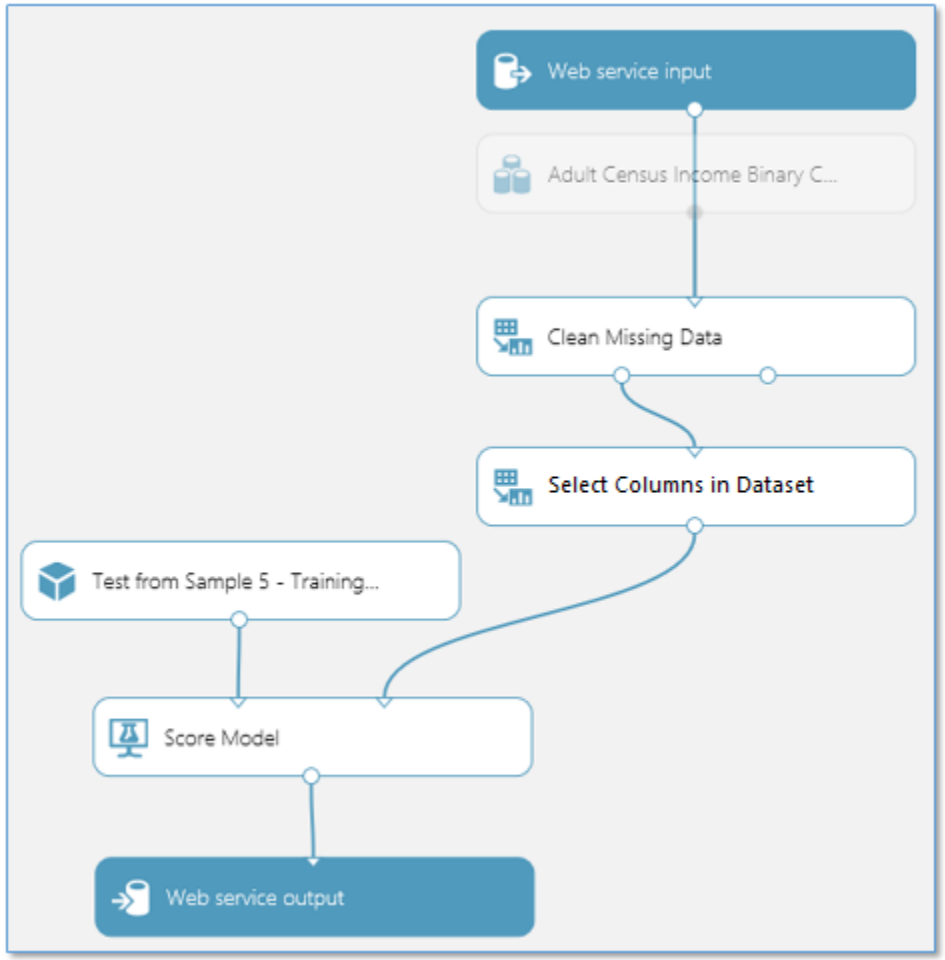
Deploy it as a web service - You can deploy your predictive experiment as a new or classic Azure web service. Users can send data to your model and receive your model's predictions.





How to prepare your model for deployment in Azure Machine Learning Studio





How to consume an Azure Machine Learning Web service

With the Azure Machine Learning Web service, an external application communicates with a Machine Learning workflow scoring model in real time. A Machine Learning Web service call returns prediction results to an external application.

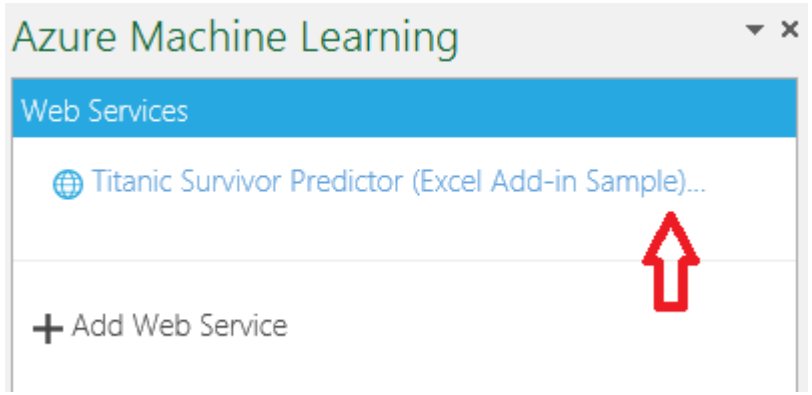
Azure Machine Learning has two types of services:

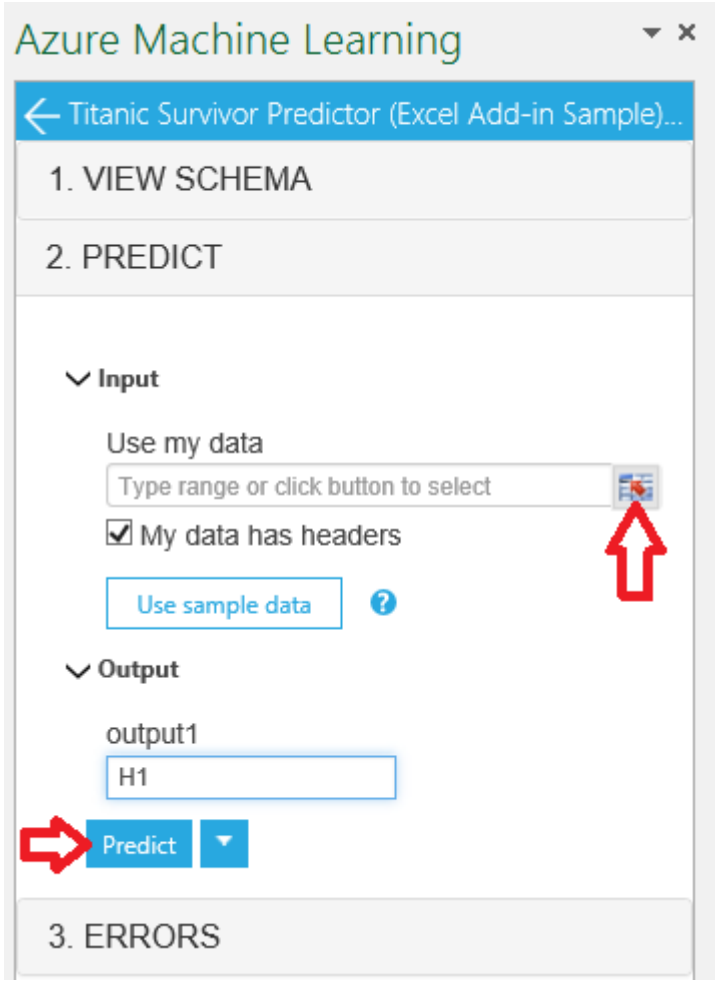
Request-Response Service (RRS) – A low latency, highly scalable service that provides an interface to the stateless models created and deployed from the Machine Learning Studio.

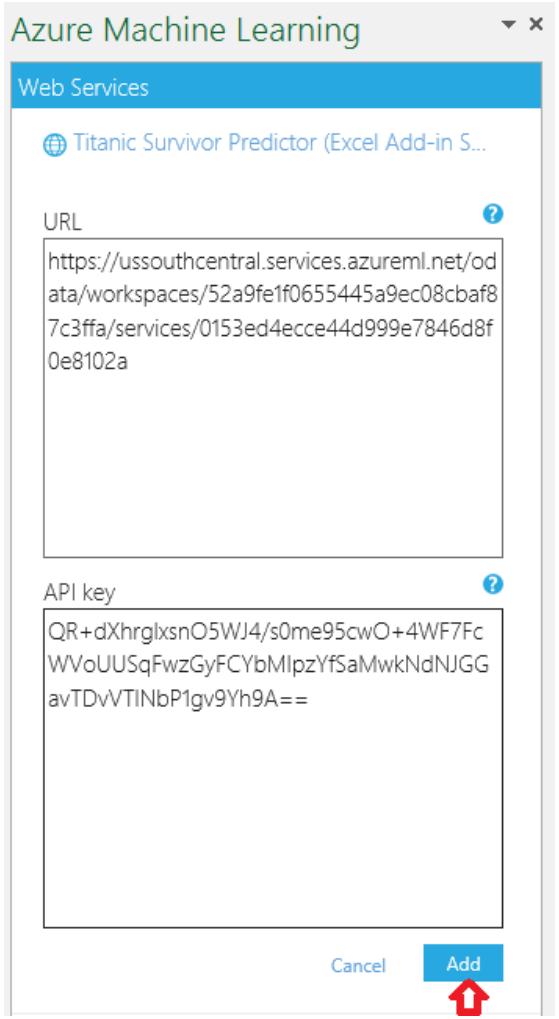
Batch Execution Service (BES) – An asynchronous service that scores a batch for data records.

Consume

Excel Add-in for Azure Machine Learning web services



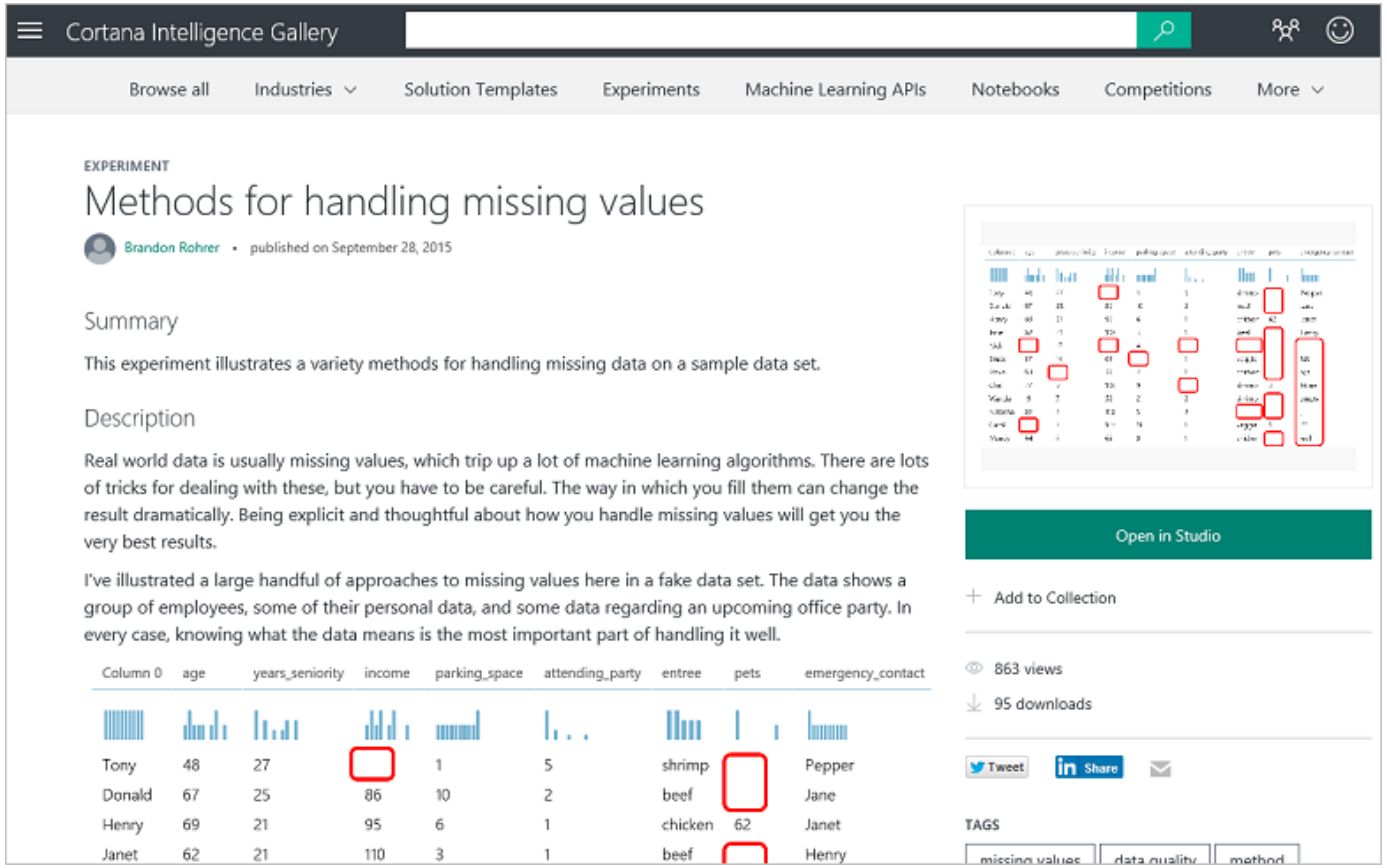




Consume an Azure Machine Learning web service with a web app template

Find experiments that demonstrate machine learning techniques

There are other experiments in the [Cortana Intelligence Gallery] that were contributed specifically to provide how-to examples for people new to data science.



Q&A

Appendix