Module 1 Research

Wednesday, June 14, 2017 12:01 PM

1. **Setup**
    a. Intro/facilities
        i. Things you need prior to class
            i. Background in data technologies, such as working with relational and non-relational data processing systems
            ii. A general level of predictive and classification statistics
            iii. A general understanding of machine learning
            iv. A subscription to azure
    b. Lab: (optional) Local workstation setup
        i. Visual studio community
            i. A fully featured, extensible, free IDE for creating modern apps for android, iOS, windows, as well as web apps and cloud services
            ii. Everything you need, all in one place
                1) Flexibility
                2) Productivity
                3) Ecosystem
                4) Languages
    c. Learning objectives
        i. Understand the CIS process
        ii. Understand the CIS platform
        iii. Understand DevOps for Data Science projects
        iv. Set up and configure your development environments
    d. Class/Self-Study/Projects
    e. Understanding Cortana intelligence
        i. Cortana intelligence suite
            i. Help your business thrive with a trusted data platform
                1) Take action using business analytics
                2) Innovate using intelligent agents
                3) Stay agile and achieve value quickly
                4) Scale up analytics with piece of mind
            ii. Refine and evolve your business processes, whatever industry
                1) Manufacturing
                    a) Boost productivity with proactive monitoring. Eliminate downtime by enabling better predictive maintenance for your capital assets.
                2) Financial services
                    a) Increase customer satisfaction and scale online. Connect more securely and effectively to your financial data and customers
                3) Retail
                    a) Attract customers and keep them coming back. Offer personalized service across hundreds of thousands of customers and locations
                4) Healthcare
                    a) Generate more positive patient outcomes. Provide comprehensive care by connecting to patient data during every in-person interaction
                5) Government
                    a) Make informed decisions and improve citizen engagement, driving change and understanding emerging public trends
        ii. Learn Analytics portal
            i. Analytics training
            ii. Course catalog
            iii. Data science certifications
            iv. Analytics training partners
            v. Links to blogs and other resources
    f. Cortana intelligence in a sentence

g. The data science process and platform
    i. CRISP-DM methodology
        i. Cross-industry process for data mining
        ii. Provides a structured approach to planning a data mining project
        iii. Stage 1 - business understanding
        iv. Determine business objectives
            1) What is the desired outputs of the project?
                a) Set objectives
                b) Produce project plan
                c) Business success criteria
        v. Assess the current situation
            1) Inventory of resources
            2) Requirements, assumptions and constraints
            3) Risks and contingencies
            4) Terminology
            5) Costs and benefits
        vi. Determine data mining goals
            1) Business success criteria
                a) e.g. increase catalogue sales to existing customers
            2) Data mining success criteria
                a) Predict how many widgets a customer will buy, given their purchases over the past three years, demographic info, and price of the item
        vii. Produce project plan
            1) Project plan
            2) Initial assessment of tools and techniques
        viii. Stage 2 - data understanding
            1) Initial data collection report
        ix. Describe data
            1) Examine the gross or surface properties of the acquired data and report on the results
            2) Data description report, then evaluate if it satisfies your requirements
        x. Explore data
            1) Data exploration report
                a) Distribution of key attributes
                b) Relationships between pairs or small numbers of attributes
                c) Results of simple aggregations
                d) Properties of significant sub-populations
                e) Simple stat analysis
        xi. Verify the data quality
            1) Is the data complete?
            2) Is it correct, or does it contain errors and, if there are errors, how common are they?
            3) Are they missing values in the data? If so, how are they represented, where do they occur and how common are they?
        xii. Data quality report
            1) List the results, suggest problems to solutions. Cycle back if necessary
        xiii. Stage 3 - data preparation
        xiv. Select your data
            1) Rationale for inclusion/exclusion
        xv. Clean your data
            1) Data cleaning report
        xvi. Construct required data
            1) Derived attributes - these are new attributes that are constructed from one or more existing attributes in the same record, for example you might use the variables of length and width to calculate a new variable of area
            2) Generated records - here you describe the creation of any completely new records. For example you might need to create records for customers who made no purchase during the past year. There was no reason to have such records in the raw data, but for modelling purposes it might make sense to explicitly represent the fact that particular customers made zero purchases.
        xvii. Integrate data
            1) Merged data
            2) Aggregations

xviii. Stage 4 - Modelling
xix. Select modelling technique
    1) Modelling technique
    2) Modelling assumptions
xx. Generate test design
    1) e.g. in supervised data mining tasks such as classification, it is common to use error rates as quality measures for data mining models. Therefore, typically split dataset into train and test sets, build the model on the train set, and estimate its quality on the separate test set
    2) Test design
xxi. Build model
    1) Parameter settings, consider rationale for each
    2) Models
    3) Model descriptions and interpretations
xxii. Assess model
    1) Summarize results and quality of model
    2) Revised parameter settings -- tune for next run
xxiii. Stage 5 - evaluation
xxiv. Evaluate your results
    1) The degree to which the model meets your business objectives and seek to determine if there is some business reason why this model is deficient
    2) Assessment of data mining results and select approved models
xxv. Review process
    1) Summarize the process and highlight activities that have been missed and those that should be repeated
xxvi. Determine next steps
xxvii. Stage 6 - deployment
xxviii. Plan deployment
xxix. Plan monitoring and maintenance
xxx. Produce final report
xxxi. Review project

h. The team data science process
    i. Break it down
        i. Team
            1) Not just individuals
        ii. Data science
            1) Focus
        iii. Process
            1) Not just product
    ii. What is a process?
        i. A process specifies a detailed sequence of activities necessary to perform specific business tasks
        ii. It's used to standardize procedures and establish best practices
    iii. Technology changes fast -- but process does not
    iv. Challenges
        i. Global teams
        ii. Team growth
        iii. Varied use cases
        iv. Diverse data science backgrounds
    v. We're going English -> math -> english
    vi. TDSP
        i. Business understanding
            1) Goals:
                a) The key variable are specified that are to serve as the model targets and whose related metrics are used determine the success for the project
                b) The relevant data sources are identified that the business has access to and or from other sources when needed
            2) How to do it
                a) Define objectives
                    i) Work with your customer and other stakeholders to understand and identify the business problems. Formulate questions that define the business goals and that data science techniques can target
                b) Identify data sources

i) Find the relevant data that helps you answer the questions that define the objectives of the project

3) Define success metrics -- makes sure they're SMART (specific, measurable, achievable, relevant, time-bound)

ii. Data acquisition and understanding
   1) Goals
      a) A clean, high quality dataset whose relations to the target variables are understood that are located in the analytics environment, ready to model
      b) A solution architecture of the data pipeline to refresh and score data regularly has been developed
   2) Ingest the data, explore the data, set up a data pipeline
   3) Exploration -- you can use a tool developed by microsoft called IDEAR
      a) Interactive Data Exploratory Analysis and reporting can quickly generate a data quality report

iii. Modeling
   1) Goals
      a) Optimal data features for the ML model
      b) An informative ML model that predicts the target most accurately
      c) An ML model that is suitable for production
   2) How to do it
      a) Feature engineering: create data features from the raw data to facilitate model training
      b) Model training: find the model that answers the question most accurately by comparing their success metrics
      c) Determine if your model is suitable for production
   3) TDSP also has a Azure Modeling and Reporting tool (AMR) that is able to run through multiple algorithms and parameter sweeps to produce a baseline model. Can further drive feature engineering. Balancing act of having enough features and not too many

iv. Deployment
   1) Goal
      a) Models and pipeline are deployed to a production or production-like environment for final user acceptance
   2) How to do it
      a) Operationalize the model: deploy the model and pipeline to a production or production-like environment for application consumption
      b) To be operationalized, the models have to be exposed with an open API interface that is easily consumed from various applications such as online website, spreadsheets, dashboards or line of business and backend apps.

v. Customer acceptance
   1) Goal
      a) Finalize the product deliverables: confirm that the pipeline, the model, and their deployment in a production environment are satisfying customer objectives
   2) How to do it
      a) System validation - confirm it meets customer needs
      b) Project hand off to the entity that will run the system in production

i. The Cortana intelligence platform
   i. Azure
      i. Microsofts cloud - IaaS, PaaS, SaaS
      ii. Platform that allows you to host all other types of services
         1) Infrastructure - networking
         2) Platform - hosting
         3) Software - machine learning

Microsoft Azure

What it is:

Microsoft's Cloud Platform including IaaS, PaaS and SaaS

iii. When to use it:

- Storage and Data
- Networking
- Security
- Services
- Virtual Machines
- On-demand Resources and Services

ii. Azure data catalog
    i. Document metadata around data
    ii. Doesn't hold data
    iii. Any kind of data
    iv. Web, Odata feed, spreadsheet under Don's desk
    v. Doesn't store names and passwords, tells you about it

Azure Data Catalog

What it is:

vi. On-Line Catalog of Meta-Data about your Data Sources, with easy tagging and searching

When to use it:

- Sourcing data
- Data discovery
- Data vetting

iii. Azure data factory
    i. Orchestration and management piece
    ii. Helps things talk to each other
    iii. Tie everything together in a pipeline

Azure Data Factory

What it is:

A pipeline system to move data in, perform activities on data, move data around, and move data out

iv.

When to use it:

- Create solutions using multiple tools as a single process
- Orchestrate processes - Scheduling
- Monitor and manage pipelines
- Call and re-train Azure ML models

iv. Azure event hubs
    i. IoT - can bring in tons of data very rapidly based on some conditions

Event Hub

What it is:

A system to ingest data from the web, IoT, and apps at scale

ii.

When to use it:

- To stream in large amounts of data
- With IoT workloads
- Use with variable or unpredictable large data loads
- Similar to Kafka

v. Platform and storage
vi. Azure data lake
    i. Two features in one
    ii. Large storage layer - a single item can be larger than single storage
    iii. Means you can store large amounts of data and then has a query level above it
    iv. Everything from HIVE to U-SQL

Data Lake

What it is:

Data storage (Web-HDFS) and Distributed Data Processing (HIVE, Spark, HBase, Storm, U-SQL) Engines

v.

When to use it:

- Low-cost, high-throughput data store
- Non-relational data
- Larger storage limits than Blobs

vii. Azure sql data warehouse

i. Relational or semi relational

ii. Structured or semi structured, good to put it here or cosmos db

SQL Data Warehouse

What it is:

A Scaling Data Warehouse Service in the Cloud

iii.

When to use it:

- When you need a large-data BI solution in the cloud
- When you are using lots of relational data
- When you need lower cost relational storage than Blobs
- When you need pause-able scaled compute

viii. SQL Database

SQL Database

What it is:

A SQL Server Database Service in the Cloud

When to use it:

i.

- When you need a relational store
- When you need full transactional support
- When you have familiarity with SQL and T-SQL and SQL Server Objects
- When you need lots of flexible indexing
- When you do not want to manage a SQL Server
- When you have multitenant databases needed

ix. Azure cosmos db

DocumentDB

What it is:

An automatically-indexed, schema-agnostic JSON database

i.

When to use it:

- Query non-relational data
- Schema defined per object
- Document (JSON) – Oriented database
- Ad-hoc queries
- Stored Procedures

x. Cortana/Cognitive service/Bot framework
   i. Intelligent processing
   ii. Can be very useful and you want to tie it into your analytics processes

Cortana and Cognitive Services, Bot Framework

iii.

What it is:

Intelligent assistant available in computing and mobile platforms, integrated into user's ecostructure, speech and vision interaction

When to use it:

- When you want your users to interact with your solution in a natural language format
- When you have an application of your solution lends itself to the user's connected ecostructure

xi. Azure ML

i. ML models and operationalize them

Azure ML

What it is:

A multi-platform environment and engine to create and deploy Machine Learning models and API's

ii.

When to use it:

- When you need to create predictive analytics
- When you need to share Data Science experiments across teams
- When you need to create call-able API's for ML functions
- When you also have R and Python experience o Data Science team

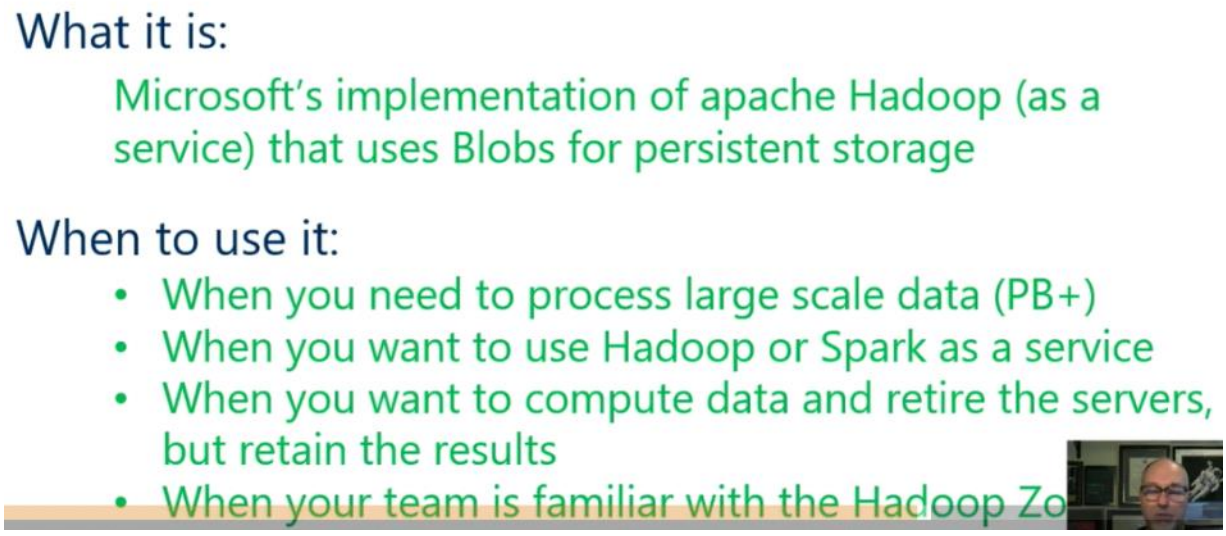
xii. HDInsight

i. Microsofts implementation of Hadoop (open source)
ii. Allows more features, easy to set up and tear down but retain the storage
iii. Allows azure hdinsight to work across the azure storage layer
iv. Keep results and get rid of cluster
v. Hadoop and MapReduce -- Mapreduce does the following:
    1) Limit the size of the data that needs to be process by selecting out of the data store only the data you actually need to analyze. E.g., you want to know the makeup of your user base by birth year, so you select only birth years out of your user profile data store
    2) Break down the data into parts and send them to different computers for processing. Computer A calculates the number of people with 1950-1959 dates, computer B does 1960-1969, etc. This group of computer is called a Hadoop cluster
    3) Put the results of each part back together after the processing on the parts is

done. You now have a relatively short list of how many people for each birth year and the task of calculating percentages in this overall list is manageable.

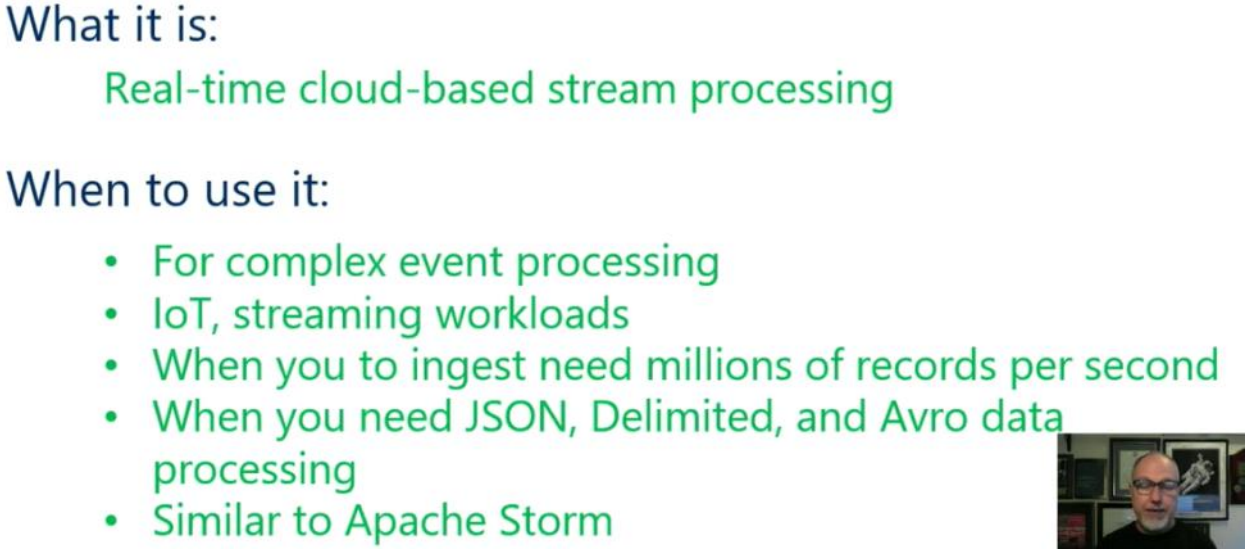
vi. HDInsight enables you to process, analyze and gain new insights from big data using the power of Hadoop

vii.



xiii. Stream Analytics
   i. e.g. thermostat is a simple event processor
   ii. Complex event processing system. Watches many attributes and does something based on that

iii.



xiv. Analysis services
   i. Built on SQL Server Analysis services
   ii. Provides enterprise-grade data modeling in the cloud

xv. Power BI
   i. Active visualization tool
   ii. As you click on things data changes and you can interact with the data

Power BI

What it is:

Interactive Report and Visualization creation for computing and mobile platforms

iii. When to use it:

- When you need to create and view interactive reports that combine multiple datasets
- When you need to embed reporting into an application
- When you need customizable visualizations
- When you need to create shared datasets, repo dashboards that you publish to your team

xvi. MRS

Microsoft R Server (MRS)

What it is:

i. A scalable, highly-performing R engine used in on-prem, in-cloud, and in-service areas

When to use it:

- When you need to use the R language and environment for data processing at scale

j. DevOps for Data Science
   i. See the DevOps guide in LearnAnalytics-ATHOMAS folder -- this was a good read
k. Azure DevOps for advanced analytics
   i. This is Azure DevOps and some of the things involved
   ii. This course focuses on Deployment and Tools
   iii. Deployment
      i. Service creation and set up
      ii. Authentication --
         1) Different ways of handling this, but often there needs to be a connection between the Azure directory (aka AAD) and the on premises directory
      iii. Tools
         1) The azure portal
            a) A comprehensive marketplace that lets you browse through thousands of items from Microsoft and other vendors that can be purchased and/or provisioned
            b) A unified and scalable browse experience that makes it easy to find the resources you care about and perform various management operations
            c) Consistent management pages (or blades) that let you manage Azure's wide variety of services through a consistent way of exposing settings, actions, billing info, health monitoring and usage data, and much more
            d) A personal experience that lets you create a customized start screen that shows the information that you want to see whenever you log in. You can

also customize any of the management blades that contain tiles
    e) Walkthrough how to create a resource, where they are, how to delete them, how to get help.
2) Powershell
    a) Provides a set of cmdlets that use the azure resource manager model for managing your azure resources
3) Command line interface
    a) Show Cloud Shell, from the top navigation in the Azure portal
    b) Choose the subscription to create a storage account
    c) OR install it locally and run from command line
4) Others - VSO, System Center SCCM, etc
5) Azure Automation
    a) Powershell and CLI are great resources for automation
    b) Save time and lower overhead cost
    c) Use out powershell workflows or create your own
    d) Integrate with the services you depend on
    e) Deliver more reliable services, faster
    f) Get easier configuration management in the cloud

l. Lab: Account activation, resource groups, DSVM setup
m. Setting up your dev environment
    i. Team foundation server
    ii. Microsoft has various solution architectures depending you're trying to accomplish
        i. For testing PaaS solutions
        ii. For testing IaaS solutions
        iii. For testing microservice solutions
    iii. Dev and test
        i. You're delivering more features faster -- keep up with a comprehensive set of dev and testing tools for your team to collaborate and deliver at cloud speed. Quickly create consistent dev and test environments on your teams through a scalable, on-demand infrastructure
    iv. Get more time to create better apps
        i. Let developers dev and build great apps. Using dev test solutions, you'll significantly reduce the time and hassle of managing dev efforts so your team can maintain its focus on app dev
        ii. "With VSTS and .NET, I save at least 30% of the time it would otherwise take to manage software dev. That's hours every week that I can spend on building better solutions
    v. Build for all the platforms you use
        i. Heterogeneous environments are the norm rather than the exception
        ii. Bring cross-platform functionality. From Linux to Windows to iOS and Android
    vi. Create dev-test environments in seconds, not weeks
        i. Simplify and speed the process of running a dev-test environment. Provision VMs in seconds, instead of days or weeks. And unlike other cloud providers, you only pay by the minute. Spin up as many as you need, network them, and allocate to your developers . Manage your environment with agility, whether you support self-provisioning by your developers, or maintain centralized control
    vii. Scalable, high-fidelity test environments
        i. Can test at production scale before moving to production
    viii. Minimize waste, maximize control
        i. Get visibility and tight control on usage of computing resources. Access real-time util data to eliminate wastage and implement chargebacks to internal customers. Advanced automation helps reduce errors, unified management balances access and control, and enterprise-ready governance capabilities enable you to set limits and control costs
    ix. Related -- Azure DevTest Labs, VSTS, App Insights
n. Primary Dev Tools
    i. Operations Management Suite (OMS)
        i. Minimal cost and complexity of deployment
        ii. Scale to cloud levels
        iii. Take advantage of the latest features
        iv. Integrated services
        v. Global knowledge
        vi. Access from anywhere

vii. Services
    1) Log analytics
    2) Automation
    3) Backup and restore critical data
    4) Site recovery

ii. Azure SDKs
    i. Visual studio tools for azure, a kit
    ii. Lots of kits. Install the azure SDK for additional set of templates and tools that help you access even more cloud resources and services to improve your azure dev experience directly from visual studio. Use these tools to deploy infinitely-scalable applications and APIs, configure diagnostics, create and manage app service resources, and integrate your data

iii. Command line tools
    i. Azure powershell - cmdlets to create, test, deploy and manage solutions and services delivered through the azure platform
    ii. Azure CLI - a lightweight cross-platform command line tool to manage services and accomplish common tasks
    iii. Powershell tools for Vis Studio - brings the VSO dev experience to powershell: edit, run and debug powershell scripts locally and remotely, leveraging VSO's locals, watch, call stack for your scripts and modules

iv. Storage explorer
    i. Standalone app that allows you to easily work with azure storage data - from any platform, anywhere. Create and manage blobs, tables, queues, generate SAS keys and more

v. VSO code
    i. Build and deploy multi-platform apps to get the most from azure services. Use any of hundreds of cool extensions and themes that help integrate your apps with azure services and author templates for ARM. Your apps and templates can be deployed to azure with simple multi-platform scripts

vi. Docker tools
    i. Build and debug your apps in a locally hosted or azure hosted Docker container. Use a variety of tools and extensions to work with Docker containers. Included in the .NET Core cross platform workload for VSO 2017

vii. Azure service fabric tools
    i. Get started building microservices using azure service fabric and VSO. These tools create new service fabric apps by using a variety of service templates, allowing you to easily debug, deploy, version and upgrade them

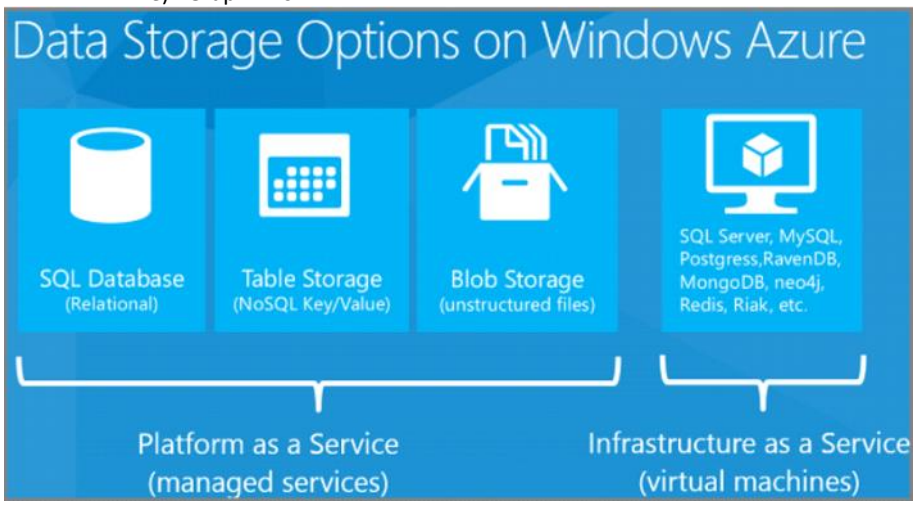
viii. Azure Resource Manager overview:
    i. Resource: a manageable item that is available through azure. Some common resources are a VM, storage account, web app, database, virtual network, but there are many more
    ii. Resource group: a container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources you want to manage as a group. You decide how to want to allocate resources and RGs based on what makes sense for your org
    iii. Resource provider: a service that supplies the resources you can deploy and manage through resource manager. Each resource provider offers operations for working with the resources that are deployed. Some common resource providers are Microsoft.Compute, which supplies the VM resource, Microsoft.Storage, which supplies the storage account resource, and Microsoft.Web, which supplies resources related to web apps
    iv. Resource manager template: a JavaScript Object Notation (JSON) file that defines one or more resources to deploy to a RG. It also defines the dependencies between the deployed resources. The template can be used to deploy the resources consistently and repeatedly.
    v. Declarative syntax: syntax that lets you state "Here is what I intend to create" wihtout having to write the sequence of programming commands. The RM template is an example of declarative syntax. In the file, you define properties for the infrastructure to deploy to azure.
    vi. https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-overview

ix. R tools for visual studio
    i. R interactive window

1) Prototype quickly
ii. Intellisense
1) Understand your code -- useful and syntactically-correct completions
iii. Plots
1) A visual gateway to your data. Ggplot2 and ggviz to make graphs and understand data
iv. Variable explorer
1) Understand your environment - keep track of objects and support drilling down
v. Debug
1) Pinpoint problems quickly with the debugger.
vi. History -- recall quickly
vii. Enhanced interpreters - turn on the afterburners
1) Enhanced multi threaded math libs, cluster scale computing, and high performance CRAN repo with checkpoint capabilities.

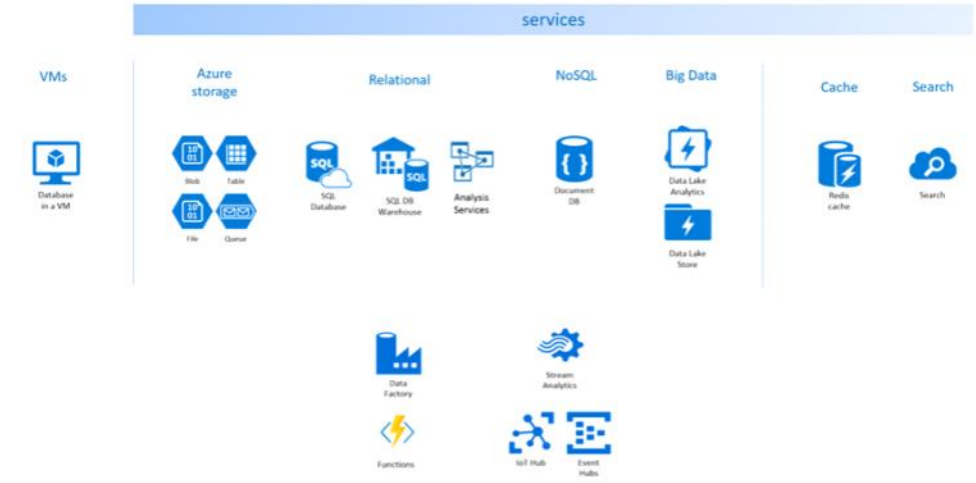
o. Azure storage for advanced analytics
i. Data storage options
i. Relational
1) Azure SQL DB
2) SQL Server
3) Oracle
4) MySQL
5) SQL compact
6) SQLite
7) Postgres
ii. Key/Value - store a single serialized object for each key value. They're good for storing large volumes of data where you want to get one item for a given key value and you don't have to query based on other properties of the item
1) Azure blob storage - key/value database that functions like file storage in the cloud, with key values that correspond to folder and file names. You retrieve a file by its folder and file name, not by searching for values in the file contents
2) Azure table storage - key/value database. Each value is called an entity (similar to a row, identified by a partition key and row key) and contains multiple properties (similar to columns, but not all entities in a tables have to share the same columns). Querying on columns other than the key is extremely inefficient and should be avoided. For example, you can store user profile data, with one partition storing information about a single user. You could store data such as username, password hash, birthdate, and so forth, in separate properties of one entity or in separate entities in the same partition. But you wouldn't want to query for all users with a given range of birthdays, and you can't execute a join query between your profile table and another table. Table storage is more scalable and less expensive than a relational database but it doesn't enable complex queries or joins
3) Azure cache
4) Redis
5) Memcached
6) Riak
iii. Document DBs - key/value databases in which the values are documents. Document here isn't used in the sense of a Word or Excel document but means a collection of named fields and values, any of which could be a child document. For example, in an order history table an order document might have order #, order date, and customer fields; and the customer field might have name and address fields. The DB encodes field data in a format such as XML, YAML, JSON, or BSON; or it can use plain text. One feature that sets document DBs apart from key/values DBs is the ability to query on non-key fields and define secondary indexes to make querying more efficient. This ability makes a document database more suitable for applications that need to retrieve data based on criteria more complex than the value of the document key. For example, in a sales order history document DB you could query on various fields such as product ID, customer ID, customer name, etc. MongoDB is a popular DB
1) Mongo DB
2) Raven DB
3) Couch DB
4) Cosmos DB
iv. Graph - store info as a collection of objects and relationships. The purpose of a graph

database is to enable an app to efficiently perform queries that traverse the network of objects and the relationships between them. For example, the objects might be employees in a HR DB, and you might want to facilitate queries such as "find all employees who directly or indirectly work for Scott."
        1) Neo4J
    v. Column family - are key/value data stores that enable you to structure data storage into collections of related columns called column families. For example, a census DB might have one group of columns for a person's name (first, middle, last), one group for the person's address, and one for profile info. The DB can then store each column family in a separate partition while keeping all of the data for one person related to the same key. You can then read all profile info without having to read through all of the name and address info as well. Cassandra is popular
        1) Cassandra
        2) Hbase
ii. NoSQL options
    i. Offer greater scalability and cost-effectiveness for storage and analysis of unstructured data
    ii. Tradeoff: don't provide rich queryability and robust data integrity of relational DBs.
    iii. Works well for IIS log data -- which involves high volume with no need for join queries
    iv. Not work well for banking transactions, which requires absolute data integrity and involves many relationships to other account-related data
iii. NewSQL - combines scalability of a NoSQL DB with the queryability and transactional integrity or a relational DB.
    i. Designed for distributed storage and query processing, which is often hard to implement in "OldSQL" DBs. e.g NuoDB

p. Storage architecture
  i. PaaS vs IaaS
    i. PaaS - we manage the hardware and software and you set up the service
        1) SQL DB
        2) Table storage
        3) Blob storage
    ii. IaaS - you set up, configure and manage VMs that run in our data center infrastructure, and you put whatever you want on them
        1) Anything you can load onto a VM
            a) SQL
            b) Key/value data
            c) Column data
            d) DocDB
            e) Graph DBs
  ii. 

q. Storage scenarios
  i. Two types of storage accounts
    i. General purpose storage accounts -- blob/table/queue/file/disk
    ii. Blob storage accounts -- blob
  ii. Blob/table/queue/file

iii.



iv.

Storage Mechanism	Interaction	Starts at*	Min GB	Max TB	Use Case
'Cool' Block Blob storage	REST (Blob), SDKs	£0.01/GB	0	500	*Archive, nearline unstructured https storage*
'Hot' Block Blob storage	REST (Blob), SDKs	£0.02/GB	0	500	Online unstructured *https* storage
Virtual Machine Disk / Page Blob Storage	VM Only	£0.05/GB	0	500 (Not per VM)	High Performance storage, up to 500 IOPS per disk volume (up to max volumes per VM type).
Table Storage	REST (Blob), SDKs	£0.07/GB	0	500	Tabular, non-relational (NoSQL) Mass-scale dictionary *https* lookup service, partitioned by default, no secondary indexes allowed.
Premium Virtual Machine Disk/ Page Blob Storage	VM Only	£0.13/GB	128	35	GUARANTEED High Performance storage, up to 5000 IOPS per disk volume (up to max volumes per VM type > 80,000 IOPS & 2TB / Second).
SQL Database Basic	*T-SQL (TDS)*	£2.33/GB *	*2*	*1*	*Tabular, Scalable, Classic Relational DBMS, Always On*
SQL Data Warehouse	*T-SQL (TDS)*	£1.03/GB *	*1024*	*~250 Compressed (~1PB)*	Scalable, *Tabular*, Parallelized, *Relational DBMS, can be paused.*
DocumentDB	REST (Blob), SDKs	£4.62/GB *	1	0.01 (10GB)	JSON Indexed document storage, can be partitioned.
Data Lake Store	WebHDFS	£0.06/GB *	*1024*	1PB per file (!)	Hadoop based unstructured data storage layer. -- Preview price includes 50% discount off US pricing, not available in Europe yet.

v. Blob storage
    i. Unstructured object data and can be any type of text or binary data, such as a doc, media file, etc
    ii. 60MB/sec or 500 requests/sec target throughput
    iii. Hot storage tier optimized for accessed frequently, cool storage infrequently accessed and long-lived
    iv. e.g. data sharing, big data, backups
    v. Disks -- page blobs
vi. Table storage
    i. Structured datasets. NoSQL key-attribute data store, which allows for rapid dev and fast access to large quantities of data
    ii. 1 MB max size of entity with a max 255 properties
    iii. Single RowKey with additional PartitionKey, 1 KB max size
    iv. e.g. store large amounts of metadata, Odata protocol (JSON)
    v. Entities
        1) The partition key for an entity is account name + table name + partition key, where the partition key is the value of the required user-defined PartitionKey property for the entity. All entities with the same partition key value are grouped into the same partition and are served by the same partition server.
    vi. The key advantage to grouping a set of entities into a single partition is that it's possible to perform atomic batch operations across entities in the same partition, since a partition exists on a single server.
    vii. On the other hand, entities that are in the same table but have different partition keys can be load balanced across different servers, making it possible to have greater scalability
vii. Queue
    i. Provides reliable messaging for workflow processing and for communication between components of cloud services

ii. 2000 messages/sec target throughput
iii. e.g. decouple components/roles
    1) Web role to worker role communication
    2) Allows roles to scale independently
iv. Implement scheduling of asynchronous tasks
v. Building process/work flows
vi. Messages
    1) The partition key for a message is the account name + queue name, so all messages in a queue are grouped into a single partition and are served by a single server. Different queues may be process by different servers to balance the load for however many queues a storage account may have

viii. File
    i. Shared storage for traditional applications using SMB protocol (Server Message Block protocol)
    ii. VMs can share file data via mounted shares
    iii. 5 TB max size of file share, 1 TB max size of a file in a share
    iv. Supported by REST and SMB

r. Redundancy and location
    i. https://docs.microsoft.com/en-us/azure/storage/storage-redundancy
    - Locally redundant storage (LRS). Locally redundant storage maintains three copies of your data. LRS is replicated three times within a single data center in a single region. LRS protects your data from normal hardware failures, but not from the failure of a single data center.
    - Zone-redundant storage (ZRS). Zone-redundant storage maintains three copies of your data. ZRS is replicated three times across two to three facilities, either within a single region or across two regions, providing higher durability than LRS. ZRS ensures that your data is durable within a single region.
    - Geo-redundant storage (GRS). GRS maintains six copies of your data. With GRS, your data is replicated three times within the primary region, and is also replicated three times in a secondary region hundreds of miles away from the primary region, providing the highest level of durability. In the event of a failure at the primary region, Azure Storage will failover to the secondary region. GRS ensures that your data is durable in two separate regions.
    - Read-access geo-redundant storage (RA-GRS). Read-access geo-redundant storage replicates your data to a secondary geographic location, and also provides read access to your data in the secondary location. Read-access geo-redundant storage allows you to access your data from either the primary or the secondary location, in the event that one location becomes unavailable. Read-access geo-redundant storage is the default option for your storage account by default when you create it.

s. Creating and managing Azure storage
    i. AZCOPY
        i. CLI utility designed for copying data to and from Azure Blob/file/table storage using simple commands with optimal performance. You can copy data from one object to another within your storage account, or between storage accounts
    ii. Service Management REST API
        i. Provides programmatic access to much of the functionality available through the management portal
        ii. REST API -- http://www.restapitutorial.com/lessons/whatisrest.html watch this video
            1) Representational State Transfer (REST)
    iii. Storage resource provider REST API
        i. The storage resource provider enables you to manage your storage account and keys programmatically
        ii. Storage accounts
            1) Operations to manage your storage accounts, such as create, update, delete, etc
        iii. Usage
            1) Operation to retrieve the current usage count and limit for the subscription's resources

t. Lab: copy and view data on your storage account using AZCOPY

Module 2 Research

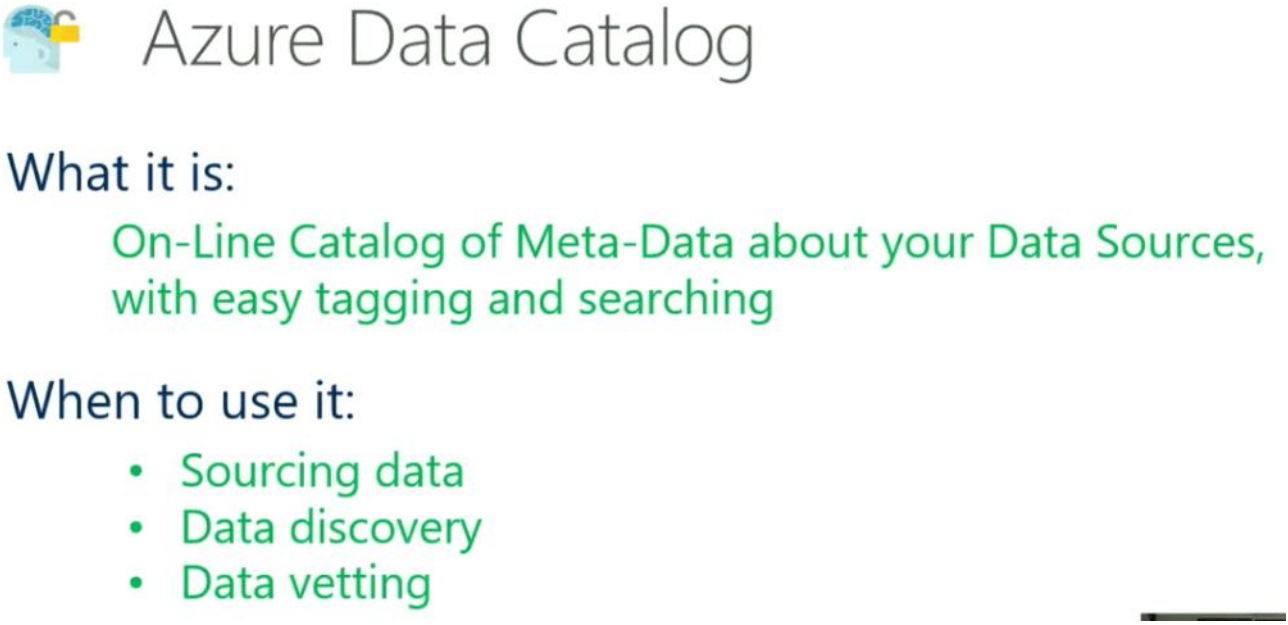
Friday, June 16, 2017 12:21 PM

1. **Business Understanding**

a. Learning objectives
   i. Determine questions from business requirements
   ii. Locate and document data sources for advanced analytics
   iii. Use patterns to create solution frameworks

b. The data science process and platform

c. The team data science process
   i. Business understanding
      1) Goals:
         a) The key variables are specified that are to serve as the model targets and whose related metrics are used determine the success for the project
         b) The relevant data sources are identified that the business has access to and or from other sources when needed
      2) How to do it
         a) Define objectives
            i) Work with your customer and other stakeholders to understand and identify the business problems. Formulate questions that define the business goals and that data science techniques can target
         b) Identify data sources
            i) Find the relevant data that helps you answer the questions that define the objectives of the project
      3) Define success metrics -- makes sure they're SMART (specific, measurable, achievable, relevant, time-bound)

d. The Cortana intelligence platform
   i. We're going to talk about Data catalog and the cortana intelligence solutions gallery
   ii. Azure data catalog
      i. Document metadata around data
      ii. Doesn't hold data
      iii. Any kind of data
      iv. Web, Odata feed, spreadsheet under Don's desk
      v. Doesn't store names and passwords, tells you about it
      vi. 

e. Defining your objectives

i. Work with your customer and other stakeholders to understand and identify the business problems. Formulate questions that define the business goals and that data science techniques can target
ii. A central objective of this step is to identify key business variables that the analysis needs to predict. These variables are referred to as the model targets and the metrics associated with them are used to determine the success of the project. Sales forecast or the probability of an order being fraudulent are examples of such targets
iii. Define project goals by asking and refining the "sharp" questions that are relevant and specific and unambiguous. Data science is the process of using names and numbers to answer such questions. "When choosing your question, imagine that you are approaching an oracle that can tell you anything in the universe, as long as the answer is a number or a name". Data science /ML is typically used to answer five types of questions:
    a. How much or how many - (regression)
    b. Which category (classification)
    c. Which group? (clustering)
    d. Is this weird? (anomaly detection)
    e. Which option should be taken (recommendation)
iv. Define the project team by specifying the roles and responsibilities of its members. Develop a high-level milestone plan that you iterate on as more info is discovered
v. Define success metrics. For example: Achieve customer churn prediction accuracy of X% by the end of this 3-month project, so that we can offer promotions to reduce churn. The metrics must be SMART (specific, measurable, achievable, relevant, time-bound)
vi. If you're building a cognitive solution, it should be:
    a. Interactive
    b. Adaptive
    c. Iterative with state
    d. Contextual

f. Business Case
    i. https://gallery.cortanaintelligence.com/Solution/Anomaly-Detection-in-Real-time-Data-Streams
    ii. Multiple countries
    iii. Several thousand employees
    iv. Current IT has a significant cost
    v. Lot of private data, financial info, and target with high security profile
    vi. External and internal attacks
    vii. Remote locations, ships and challenging environments
    viii. Taking in a significant amount of semi-structured data -- table or blob but probably table
    ix. Real-time
    x. Determine anomalies within data streams -- azure stream analytics and event hubs then some sort of model that predicts anomalies
    xi. Observe the anomalies in a dashboard -- PBI
    xii. See changes quickly
g. Design Statements
h. Data storage technologies
    i. SQL DB
        a. When you need a relational data store
        b. When you need a transactional workload
        c. When you need high volumes of reads and inserts
        d. High concurrency
        e. When you're looking for a single scale-up database
    ii. Storage blob
        a. When you need a low cost, high throughput data store
        b. When you have non-relational (No-SQL) data
        c. No Ad-hoc Query support
        d. Cheap storage option
        e. Works great with Hadoop & U-SQL
    iii. Document DB (Now Cosmos DB)
        a. When you need a query-able non-relational data store

b. When you have non-relational (No-SQL) data
c. Highly scalable
d. Costly storage (as compared to blob storage). So, use when you need to query data from an external application
e. Great for ad-hoc queries
f. Use stored procedures to transform your data

iv. SQL Data warehouse
a. When you need a relational Big Data store
b. When you have square (relational) data
c. Highly scalable
d. Analytical data store -- with amazing compute power
e. Low cost storage (but costlier than Blob storage)
f. Pause-able compute (elastic scale)

v. Data lake store
a. When you need a low-cost, high throughput data store
b. When you have non-relational (No-SQL) data
c. No ad-hoc query support
d. Works great with Hadoop & U-SQL
e. Does not have the same storage limits as blob storage (great for Big Data and IoT)

i. Data Processing technologies - for compute
i. Event hubs
a. High throughput data ingestion system
b. Great for IoT & Streaming solutions
c. Ingest up to 1 million messages per second
d. Fully managed and very easy to set up
e. Similar to Apache Kafka
f. IoT Hub uses Event Hubs underneath the covers

ii. Stream Analytics
a. Complex event processing (CEP) engine
b. When you want to query your real-time data, IoT, notification and alerting scenarios
c. SQL-life language constructs
d. Support for JSON, delimited (eg CSV) and Avro
e. Open source equivalent is Apache Storm

iii. Search
a. Simple to build a great search experience into web and mobile app
b. User with azure SQL DB, DocDB, Table and blob storage
c. Leverage advanced linguistic capabilities with deep understanding of 56 languages
d. Simple to scale and manage
e. Uses ML APIs to understand past user behavior and make recommendations

iv. HDInsight
a. Fully managed spark and hadoop on azure
b. Handles both No-SQL and rectangular (relational) data
c. Use it when you're trying to solve big data problems (handles >1 PB data)
d. Use data lake store as your storage layer
e. Delete and recreate your cluster without losing any data

v. ML
a. Use it to perform predictive analytics
b. SSIS-like drag and drop features
c. Built in ML algorithms for classification, regression, clustering and anomaly detection
d. Support for r and python to build custom models
e. One click operationalization

vi. Data factory
a. Orchestrator of your data pipelines
b. Does not do any processing (except for copy activity)
c. Use it to schedule your batch style workloads
d. Support for pulling data from various on prem systems
e. Not to be used for real time scenarios
f. Great for calling and retraining ML APIs

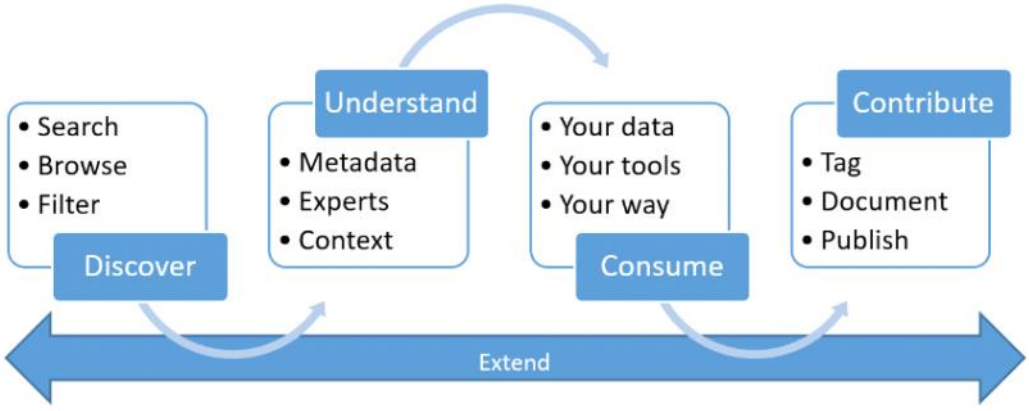
j. Decision matrix
    i. http://www.businessnewsdaily.com/6146-decision-matrix.html
    ii. DM can clear up confusion about the options and highlight points that may factor into the final call
    iii. Quantitative method removes emotion
    iv. "a tool to help you make good decisions when you must weigh difficult-to-compare factors"
    v. Creating a decision matrix
        a. List your decision alternatives as rows, and the relevant factors affecting the decisions, such as cost, ease and effectiveness, as the columns. Then, establish a ratings scale to assess the value of each alternative/factor combo. Be sure the rankings are consistent. E.g., if you're looking at pain points, be sure each issue is worded so it gets more points the worse it is
        b. Next, multiply your original ratings by the weighted rankings to get a score. Highest score wins/first item to work on.
    vi. For a decision where you have multiple options and seemingly diverse features to consider, a decision matrix can shed light on the best choice. - See more at: http://www.businessnewsdaily.com/6146-decision-matrix.html#sthash.sIeTuvZd.dpuf
k. Create a solution workflow
    i. IT Anomaly Insights Solution
        a. Provides a solution with a low barrier of entry based on CIS and AML Anomaly Detection API, making it easy for a business decision maker to evaluate and realize value within minutes, also allowing customers to bring their own data, customize and extend the solution in order to adapt it to their particular scenarios via quick proof of concepts
        b. With this solution, organizations will be able to:
            i. Leverage AML Anomaly detection API to learn and react to anomalies from both historical and real-time data. This eliminates human-in-the-loop, otherwise needed for recalibrating threshold for detecting missing anomalies and minimize false positives
            ii. Quickly realize the potential of the solution by trying it out with their own data without any upfront investment. The "try it now" experience also provides users the ability to determine the right set of sensitivity parameters for the use case in hand
            iii. Deploy an end-to-end pipeline into their subscription to ingest data from on-prem and cloud data sources and report anomalous events to downstream monitoring and ticketing systems in a plug and play manner within a matter of minutes
        c. Solution architectures
            i. Real-time metric stream originating from both on-prem based or cloud based systems can be pumped into Azure Event Hub queue. These events (or time series data points) are processed by Azure Stream Analytics where they are aggregated at five minute intervals. Each time series is sent to Azure Anomaly Detection API for evaluation at 15 minutes cadence. The results from the API along with their dimensions provided during input are then stored in Azure SQL DB. The detected anomalies are also published in Azure Service Bus so that they can be consumed by the downstream ticketing systems. The solution also provides directions to set up PBI.
        d. Anomaly detection API can detected the following types of anomalies in time series data
            i. Spikes and Dips: when monitoring login failures, unusual spikes or dips could indicate security attacks or service disruptions
            ii. Positing and negative trends: when monitoring memory usage in computing, for instance, shrinking free memory size is indicative of a potential memory leak; when monitoring service queue length, a persistent upward trend may indicate an underlying software issue
            iii. Level changes and changes in dynamic range of values: e.g., level changes in latencies of a service after service upgrade or lower levels of exceptions after upgrade
    ii. Architecture and explanation of components
        a. Data Source and Ingestion: Azure event hub
            i. Highly scalable publish-subscribe service that can ingest millions of events per second and stream them into multiple applications. Here, they are set up to ingest raw time series data from a variety of sources such as cloud gateways, monitoring agents, sensors, etc. For the quick start, the solution provides a sample data generator that can read time series data from CSV files and send it to event hubs

b. Data prep and analysis: ASA
    i. Used to aggregate the raw incoming data from the event hubs at 5 min intervals and store it to Azure Storage for later processing by ADF. This job also pushes the time series data to SQL DB to visualize in Power BI
c. Data prep and analysis: ADF
    i. Orchestrates the movement and processing of data. The data factory is made up of pipelines and activities for preparing, analyzing and publishing results. It uses custom activities to read raw data from the input storage tables, prepare individual time series datasets, make calls to Anomaly detection web services deployed in your solution for detecting anomalous events and then publishes the results
d. Data prep and analysis: AML Web Service
    i. Deployed the Anomaly Detection API into your subscription as part of the deployed solution. The solution will have both non-seasonal and seasonal anomaly detection web services. The default end to end solution uses non-seasonal web service. More details on customizing the solution to use seasonal web service can be found below. You can manage and monitor the web services via the new AML web services portal
e. Data publishing: Azure SQL DB Service and Service Bus Topics
    i. Azure SQL DB service is used to store the raw time series and the anomaly scores received from the Anomaly Detection API so that they can be visualized in PBI. The ADF also publishes any anomalies detected in the current slice to Service Bus Topics. These anomaly messages can be subscribed to and consumed by a variety of apps such as ticketing systems, chat clients, mobile apps, etc.
    ii. Service bus messaging: https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messaging-overview
f. Data consumption: PBI
    i. The solutions offer two options for visualizing the data and insights in PBI
        1) An embedded PBI hosted on a website and deployed as part of the solution
        2) A prebuilt PSI template which can be linked to the deployed solution and loaded with real time data from the deployment. User this for customizing.
    ii. These dashboards show views over the raw time series that are monitored by the pipeline and any anomalies detected by the Anomaly Detection API. The data for the dashboard is sourced from Azure SQL DB that is deployed with the solution.

l. Identify data sources
    i. Find the relevant data that helps you answer the questions that define the objectives of the project
    ii. Identify data sources that contain known examples of answers to your sharp questions. Look for the following data
        a. Relevant -- do we have measures of the target and features that are related to the target?
        b. Accurate measure - data that is an accurate measure of our model target and the features of interest
    iii. It is not uncommon, for example, to find that existing systems need to collect and log additional kinds of data to address the problem and achieve the project goals. In this case, you may want to look for external data sources or update your systems to collect newer data
    iv. Data validation
        a. DV guarantees to your app that every data value is correct and accurate. You can design DV into your app with several differing approaches: user interface code, app code, or database constraints. There are several types of data validation:
            i. Data type validation - i.e. answer simple questions "is the number numeric?"
            ii. Range checking - ensures values are within allowable mins and maxs
            iii. Code checking - typically requires a lookup table. E.g. maybe your app calculates sales tax for only certain state codes. You would need to create a validation table to hold the authorized, taxable state codes.
            iv. Complex validation - e.g. a health care claim which has a billed amount of $123, but the allowable amount may depend on a YTD rolling accumulation that is capped at $1500 (not to exceed a lifetime policy max of $100k)

m. Locating and inspecting data
    i. Keys to quality source data
        a. Authority

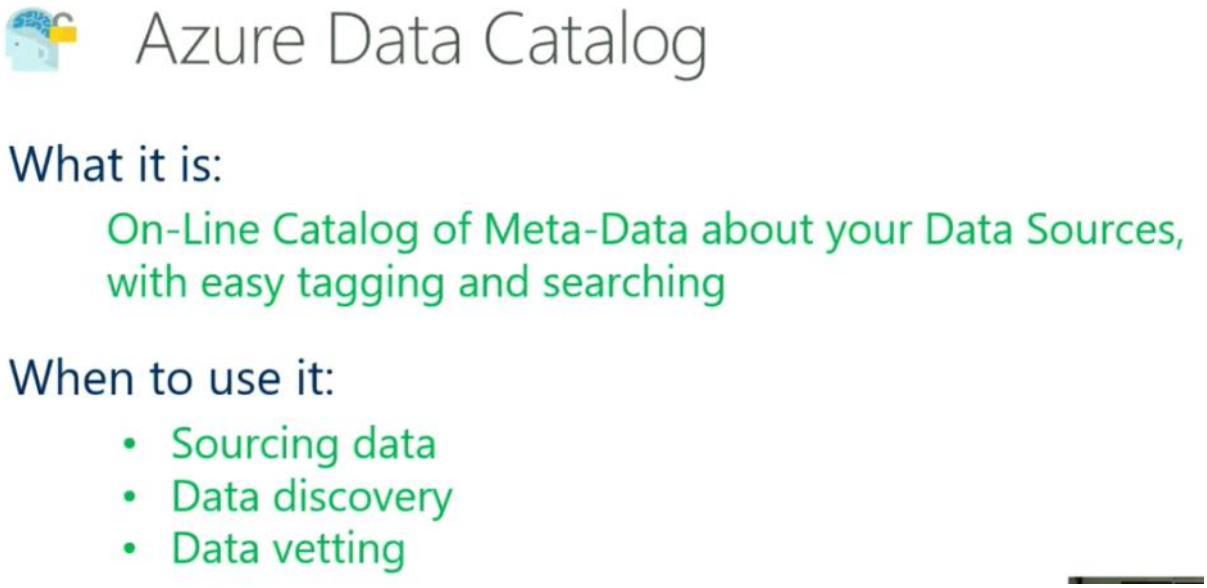
b. Spread
c. Consistency
d. Types and units
e. Representation

n. ADC
   i. Process
      a. Register data sources
      b. Tag data
      c. Document data
      d. Search and connect
   ii. With ADC, any user (analyst, data scientist or developer) can discover, understand and consume data sources. ADC includes a crowdsourcing model of metadata and annotations. It is a single, central place for all of an organization's users to contribute their knowledge and build a community and culture of data
   iii. 

o. Lab: adding and searching data on the ADC

Module 3 Research

Friday, June 16, 2017 12:32 PM

1. **Data Acquisition and Understanding**
    a. Learning objectives
        i. Ingest data into the Azure platform
        ii. Explore data using various tools
        iii. Update data documentation
        iv. Create a mechanism to orchestrate and manage data flows through a solution
        v. The CIS process
    b. The data science process and platform
    c. The team data science process
        i. Data acquisition and understanding
            1) Goals
                a) A clean, high quality dataset whose relations to the target variables are understood that are located in the analytics environment, ready to model
                b) A solution architecture of the data pipeline to refresh and score data regularly has been developed
            2) Ingest the data, explore the data, set up a data pipeline
            3) Exploration -- you can use a tool developed by Microsoft called IDEAR
                a) Interactive Data Exploratory Analysis and reporting can quickly generate a data quality report
    d. The Cortana intelligence platform
        i. Azure data catalog
            i. Document metadata around data
            ii. Doesn't hold data
            iii. Any kind of data
            iv. Web, Odata feed, spreadsheet under Don's desk
            v. Doesn't store names and passwords, tells you about it
            vi. 

        ii. Azure data factory
            i. Orchestration and management piece
            ii. Helps things talk to each other
            iii. Tie everything together in a pipeline

Azure Data Factory

What it is:

A pipeline system to move data in, perform activities on data, move data around, and move data out

iv.

When to use it:

- Create solutions using multiple tools as a single process
- Orchestrate processes - Scheduling
- Monitor and manage pipelines
- Call and re-train Azure ML models

iii. Azure event hubs

i. IoT - can bring in tons of data very rapidly based on some conditions

Event Hub

What it is:

A system to ingest data from the web, IoT, and apps at scale

ii.

When to use it:

- To stream in large amounts of data
- With IoT workloads
- Use with variable or unpredictable large data loads
- Similar to Kafka

iv. MRS

Microsoft R Server (MRS)

What it is:

i. A scalable, highly-performing R engine used in on-prem, in-cloud, and in-service areas

When to use it:

- When you need to use the R language and environment for data processing at scale

v. Power BI
    i. Active visualization tool
    ii. As you click on things data changes and you can interact with the data

Power BI

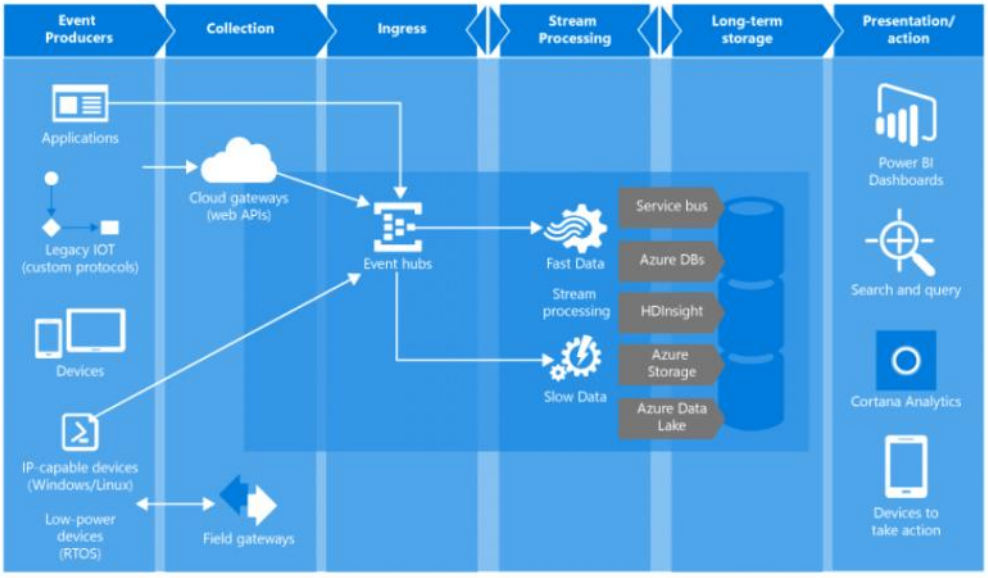
What it is:

Interactive Report and Visualization creation for computing and mobile platforms

iii. When to use it:

- When you need to create and view interactive reports that combine multiple datasets
- When you need to embed reporting into an application
- When you need customizable visualizations
- When you need to create shared datasets, repo dashboards that you publish to your team

    vi. Azure storage options as well
e. Data ingestion
f. Azure event hubs
    i. What is it?
        1) A highly scalable data streaming platform and event ingestion service capable of receiving and processing millions of events per second. Event hubs can process and store events, data or telemetry produced by distributed software and devices. Data sent to an EH can be transformed and stored using any real-time analytics provider or bathing/storage adapters. With the ability to provide publish-subscribe capabilities with low latency and at massive scale, EH serves as the "on ramp" for Big Data
    ii. Why use it?
        1) Application instrumentation
        2) User experience or workflow processing
        3) IoT scenarios
    iii. Overview
        1) "front door" for an event pipeline, often called an event ingestor (a component or service that sits between event publishers and event consumers to decouple the production of an event stream from the consumption of those events)

2) 

iv. Event hubs security and auth
   1) Only clients that present valid credentials can send data to an event hub
   2) A client cannot impersonate another client
   3) A rogue client can be blocked from sending data to an event hub

g. Options for data ingestion
   i. PowerShell in Azure Storage
   ii. Azure Data Factory - Copy activity
      1) Can use to copy data between on prem and cloud data stores. After the data is copied, it can be further transformed and analyzed. You can also use Copy Activity to publish transformation and analysis results for BI and application consumption
   iii. Azure automation
      1) Automate the manual, long-running, error-prone and frequently repeated tasks that are commonly performed in a cloud and enterprise environment. It saves time and increases the reliability of admin tasks, and even schedules them to be automatically performed at regular intervals. You can automate using runbooks or configuration management
      2) Runbooks are based on powershell
   iv. Azure storage SDKs
   v. Azure storage explorer
      1) Standalone app from MSFT that allows you to easily work with Azure storage data on windows, macOS and Linux
   vi. AZCopy
      1) Command line utility designed for copying data to and from Blob/File/Table storage using simple commands. Can copy within your storage account or between storage accounts
   vii. Import/Export service
      1) Securely transfer large amounts of data to azure Blob by shipping hard disk drives to an azure data center. You can also use this service to transfer data from blob to hard disk drives and ship to your on prem site. This service is suitable in situations where you want to transfer several TB of data

      You can use this service in scenarios such as:
      - Migrating data to the cloud: Move large amounts of data to Azure quickly and cost effectively.
      - Content distribution: Quickly send data to your customer sites.
      - Backup: Take backups of your on-premises data to store in Azure blob storage.
      - Data recovery: Recover large amount of data stored in blob storage and have it delivered to your on-premises location.

      From <https://docs.microsoft.com/en-us/azure/storage/storage-import-export-service>

h. Connect on prem to anything
   i. VPN gateway - a type of virtual network gateway that sends encrypted traffic across a public connection to an on prem location. You can also use VPN gateways to send encrypted traffic between azure virtual networks over the Msft network. To send encrypted network traffic between your azure virtual network and your on-prem site, you must create a VPN gateway for your virtual network

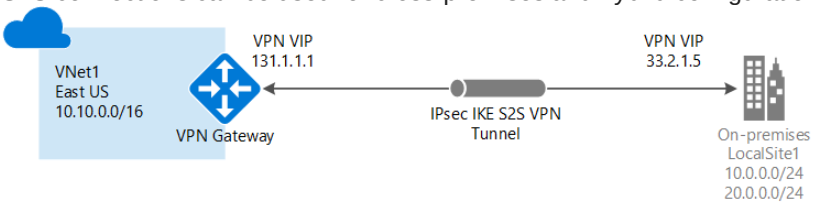
ii. Each virtual network can have only one VPN gateway, but you can create multiple connections to the same VPN gateway. E.g. a multi-site connection configuration. When you create multiple connections to the same VPN gateway, all VPN tunnels, including point-to-site VPNs, share the bandwidth that is available for the gateway

What is a virtual network gateway?

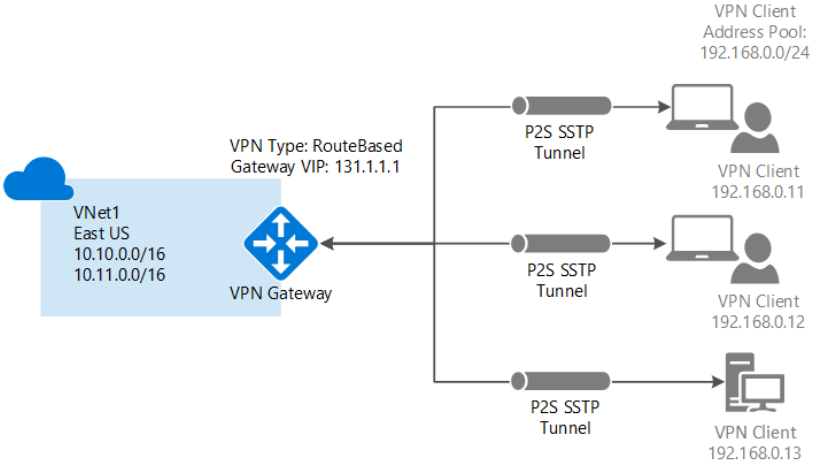
A virtual network gateway is composed of two or more virtual machines that are deployed to a specific subnet called the GatewaySubnet. The VMs that are located in the GatewaySubnet are created when you create the virtual network gateway. Virtual network gateway VMs are configured to contain routing tables and gateway services specific to the gateway. You can't directly configure the VMs that are part of the virtual network gateway and you should never deploy additional resources to the GatewaySubnet.
When you create a virtual network gateway using the gateway type 'Vpn', it creates a specific type of virtual network gateway that encrypts traffic; a VPN gateway. The Gateway SKU that you select when you create your virtual network gateway determines the VMs that are created and configured in the GatewaySubnet.

From <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpngateways>

iii. Site to site vs Point to site
A Site-to-Site (S2S) VPN gateway connection is a connection over IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has a public IP address assigned to it and is not located behind a NAT. S2S connections can be used for cross-premises and hybrid configurations. +



From <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpngateways>
A Point-to-Site (P2S) VPN gateway connection allows you to create a secure connection to your virtual network from an individual client computer. P2S is a VPN connection over SSTP (Secure Socket Tunneling Protocol). P2S connections do not require a VPN device or a public-facing IP address to work. You establish the VPN connection by starting it from the client computer. This solution is useful when you want to connect to your VNet from a remote location, such as from home or a conference, or when you only have a few clients that need to connect to a VNet. P2S connections can be used with S2S connections through the same VPN gateway, as long as all the configuration requirements for both connections are compatible.+



From <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpngateways>

Microsoft Azure ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a dedicated private connection facilitated by a connectivity

provider. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure, Office 365, and CRM Online. Connectivity can be from an any-to-any (IP VPN) network, a point-to-point Ethernet network, or a virtual cross-connection through a connectivity provider at a co-location facility.+
ExpressRoute connections do not go over the public Internet. This allows ExpressRoute connections to offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet.+
An ExpressRoute connection does not use a VPN gateway, although it does use a virtual network gateway as part of its required configuration. In an ExpressRoute connection, the virtual network gateway is configured with the gateway type 'ExpressRoute', rather than 'Vpn'. For more information about ExpressRoute, see the ExpressRoute technical overview.

From <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpngateways>

i. Lab: work with Table Storage
j. Data exploration
k. Exploring data
    i. You can use R, Azure ML, Excel, other tools.
    As data scientists, we ask many questions when graphically exploring a data set including:

    - Which features appear to track the behavior of the labels in some way? We are always on the lookout for features which will improve model performance. Conversely, features which show only random behavior with respect to the labels, will likely only add noise if used for training an ML model. Be careful, correlation should never be confused with causation! Causation can only be determined with domain knowledge.

    - Which features show independent behavior? Conversely, we must eliminate features which are nearly collinear to other features or otherwise redundant. These features will only add noise when used to train an ML model.

    - Do the features contain outliers? Can we validate that these data are indeed outliers and not interesting, unusual cases? If there are outliers, what treatment should we apply?

    - Are there degenerate features which only add noise if used for training an ML model? For example, a feature may have mostly a single value, say zero, and few other values. In many cases, such a feature conveys no information. But, be careful, perhaps those few different values represent important special cases of interest.

    - Are there trends or biases in the data which we must account for before we can build an ML model? This is particularly important in time series or forecasting models.

    - From <https://blogs.technet.microsoft.com/machinelearning/2015/09/24/data-exploration-with-azure-ml/>

l. Update the ADC
    i. Search
    ii. Add tags
    iii. Add experts
    iv. Thoroughly document any data
    v. ADC is really useful if people buy into it.
m. Lab: Exploring your data
    i. Need to walk through the R code in the GitHub materials
n. Update data
o. Options (how to choose which product to use)
    i. Remember this? We're going to talk about ASA and ADF
p. Decision matrix
    i. Fast multiple sources of data
        1) We can use event hubs as discussed earlier
        2) We can use ASA to have fast ingestion of massive data sets
q. Azure stream analytics
    i. How to
        1) Set up the environment for ASA

2) Provision the azure resources
3) Create ASA job(s)
    a) Define input sources
    b) Define output
4) Set up the azure stream analytics query
5) Start the stream analytics job
6) Check results
7) Monitor

ii. What is Stream Analytics
1) Fully managed, cost effective real time event processing engine that unlocks deep insights from data.

r. ADF
i. Compose and manage data services at scale
1) Create, schedule and manage data pipelines
2) Visualize data lineage
3) Connect to on prem and cloud data sources
4) Monitor data pipeline health
5) Automate cloud resource management

ii. Ingest and prepare
1) Use ADG to ingest from multiple on prem and cloud sources. Connect to on prem sources with a data management gateway, and use ADF to get your data where it needs to go. Prepare and partition your data as you ingest It or apply preprocessing steps

iii. Transform and analyze
i. Schedule and manage you data transformation and analysis process. Choose from a wide range of processing services, and put them into managed data pipelines to use the best tool for the ob. E.g. add a Hadoop processing step for big or semi-structured data, a stored procedure invocation step for structured data, a ML step for analytics, or insert your own custom code as a processing step in any pipeline

iv. Publish and consume
i. Use data pipelines to transform raw data into finished or shaped data that's ready for consumption by BI tools or applications. User data factory to get your valuable data where it needs to go for consumption by your on prem or cloud apps and services.

v. Monitor and manage
i. At a glance to identify issues and take action. Easily understand when data arrives, where it comes from, and how and when it's ready for processing. Set up alerts to monitor ADF service health. Saves you time and money by automating your data pipelines with on demand cloud resource management

s. ADF
i. Create, orchestrate, and manage data movement and enrichment through the cloud

t. ADF Components
i. Datasets
i. An activity takes zero or more datasets as inputs and one or more datasets as outputs. Datasets represent data structures within the data stores, which simply point or reference the data you want to use in your activities as inputs or outputs. For example, an azure blob dataset specifies the blob container and folder in the azure blob storage from which the pipeline should read the data. Or, an azure SQL table dataset specifies the table to which the output data is written by the activity

ii. Linked services
i. Much like connection strings, which define the connection information needed for data factory to connect to external resources. A linked service defines the connection to the data source and a dataset represents the structure of the data. e.g., an azure storage linked services specifies connection strings to connect to the account. And, an azure blob dataset specifies the blob container and folder that contain the data
ii. Linked services are useful for two purposes in data factory
    a) To represent a data store including, but no limited to, an on prem SQL Server, Oracle database, file share, or Azure Blob Storage account.
    b) To represent a compute resource that can host the execution of an activity. For example, the HDInsight Hive activity runs on HDInsight Hadoop cluster

iii. Activity
i. A pipeline may have one or more activities. Activities define the actions to perform on your data. e.g. you may use a copy activity to copy data from one data store to another data store. Similarly, you may use a Hive activity, which runs a Hive query on

Azure HDInsight cluster to transform or analyze your data. ADF supports data movement activities and data transformation activities
        ii. Data movement - copy data from a source to sink
        iii. Data transformation - different compute activities e.g. ML batch execution, Hive, MapReduce, etc.
    iv. Pipeline
        i. Pipeline is a group of activities. Together, the activities in a pipeline perform a task. E.g., a pipeline could contain a group of activities that ingests data from an Azure blob, and then run a Hive query on an HDInsight cluster to partition the data. The benefit of this is that the pipeline allows you to manage the activities as a set instead of each one independently
u. ADF logical flow
v. ADF process
    i. Define architecture: set up objectives and flow
    ii. Create the data factory: portal, powershell, VS
    iii. Created linked services: connections to data and services
    iv. Create datasets: input and output
    v. Create pipeline: define activities
    vi. Monitor and manage: portal or powershell, alerts and metrics
w. Design process
    i. Define data sources, processing requirements, and outputs -- also management and monitoring
    ii. Define the architecture by setting up objectives and flow
x. Simple ADF
    i. Input, transformation activity, output
y. Create the data factory
    i. Using the portal
        i. We're going to use the portal for our lab, but it's possible to use powershell and visual studio, see references.
        ii. Use in Non-MS clients
        iii. Use for exploration
        iv. Use when teaching or in a demo
    ii. Using Powershell
        i. Use in MS clients
        ii. Use for automation
        iii. Use for a quick set up and tear down
    iii. Using visual studio
        i. Use in mature dev environments
        ii. Use when integrated into larger development process
z. Creating linked services
    i. A connection to data or connection to compute resource -- also termed Data Store
aa. Data options
    i. Different source and sink options for ADF
bb. Activity options
    i. Several different transformation activities that can be performed.
cc. Gateway for on prem
    i. Data management gateway for on prem
        i. You must install DMG on your on prem machine to enable moving data to/from an on prem data store. The gateway can be installed on the same machine as the data store or on a different machine as long as the gateway can connect to the data store.
dd. Create datasets
    i. Dataset concepts
ee. Create pipelines
    i. Pipeline JSON
ff. Manage and monitor
    i. Locating failures within a pipeline
gg. Lab: create an ADF project

Module 4 Research

Friday, June 16, 2017 12:33 PM

1. **Modeling**
    a. Learning objectives
        i. Understand ML
        ii. Be able to use the ML environment
        iii. Create and deploy a ML solution
    b. The data science process and platform
    c. The team data science process
        i. Modeling
            i. Goals
                1) Optimal data features for the ML model
                2) An informative ML model that predicts the target most accurately
                3) An ML model that is suitable for production
            ii. How to do it
                1) Feature engineering: create data features from the raw data to facilitate model training
                2) Model training: find the model that answers the question most accurately by comparing their success metrics
                3) Determine if your model is suitable for production
            iii. TDSP also has a Azure Modeling and Reporting tool (AMR) that is able to run through multiple algorithms and parameter sweeps to produce a baseline model. Can further drive feature engineering. Balancing act of having enough features and not too many
    d. The Cortana intelligence platform
        i. Azure ML
            i. ML models and operationalize them

Azure ML

What it is:

A multi-platform environment and engine to create and deploy Machine Learning models and API's

ii. When to use it:

- When you need to create predictive analytics
- When you need to share Data Science experiments across teams
- When you need to create call-able API's for ML functions
- When you also have R and Python experience o Data Science team

    e. Azure ML
        i. ML in 5 minutes
        ii. ML capabilities
        iii. ML algorithms
            i. Supervised
                1) Predicting the future
                2) Learn from known past example to predict future
                3) Labels provided
                4) Classification, regression, anomaly detection
            ii. Unsupervised learning
                1) Making sense of data

2) Understanding the past
3) Learning the structure of data
4) Labels not provided

iv. Azure ML environment
   i. Dev environment - creating experiments and sharing a workspace
   ii. Deployment environment - publishing the model, using the API, consuming in various tools
   iii. ML studio
      1) Collaborative visual dev environment that helps you build, test and deploy predictive analytics solutions in the cloud. You upload data or connect to data already in the cloud, choose an algorithm from a ready to use library of algorithms, and build an end to end predictive workflow. You can then quickly deploy the model and integrate the workflow in apps by calling a we service
   iv. Modules
      1) Each module represents a set of code that can run independently and perform a ML task, given the required inputs. A module might contain a particular algorithm, or perform a task that is important in ML, such as missing value replacement, or statistical analysis
         a) ML algorithms
         b) Data input and output modules
         c) Data transformation modules
         d) Text analysis modules
         e) Support for external languages by Python and R modules
         f) Statistical functions
v. Basic Azure ML elements
   i. Import data
   ii. Preprocess
   iii. Split data
   iv. Algorithm
   v. Train model
   vi. Score model
vi. Creating an experiment
vii. Measuring effectiveness and efficiency in ML
viii. Azure ML Score
   i. Apply a training model to
      1) A list of recommended items
      2) Forecasts for time series models
      3) Estimates of projected demand, volume, or other numeric quantity, for regression models
      4) Cluster assignments
      5) A predicted class or outcome, for classification models
      6) Probability scores associated with these outputs
ix. Azure ML Evaluate and Metrics for regression models
   i. Metrics for classification models
      1) Accuracy, recall, precision F1 score
         a) Accuracy: measures the goodness of a classification model as the proportion of true results to total cases
         b) Precision: the proportion of true results over all positive results
         c) Recall: fraction of all correct results returned by the model
         d) F-score: computed as the weighted average of precision and recall between 0 and 1, where the ideal F-score is 1
      2) AUC: measures the area under the curve plotted with true positives on the y axis and false positives on the x axis. This metric is useful because it provides a single number that lets you compare models of different types
      3) Average Log loss: a single score used to express the penalty for wrong results. It is calculated as the difference between two probability distribution - the true one, and the one in the model
      4) Training Log loss: a single score that represents the advantage of the classifier over a random prediction
   ii. Metrics for regression models
      1) Mean absolute error (MAE): how close the predictions are to the actual outcomes; thus, a lower score is better
      2) Root mean squared error (RMSE): a single value that summarizes the error in

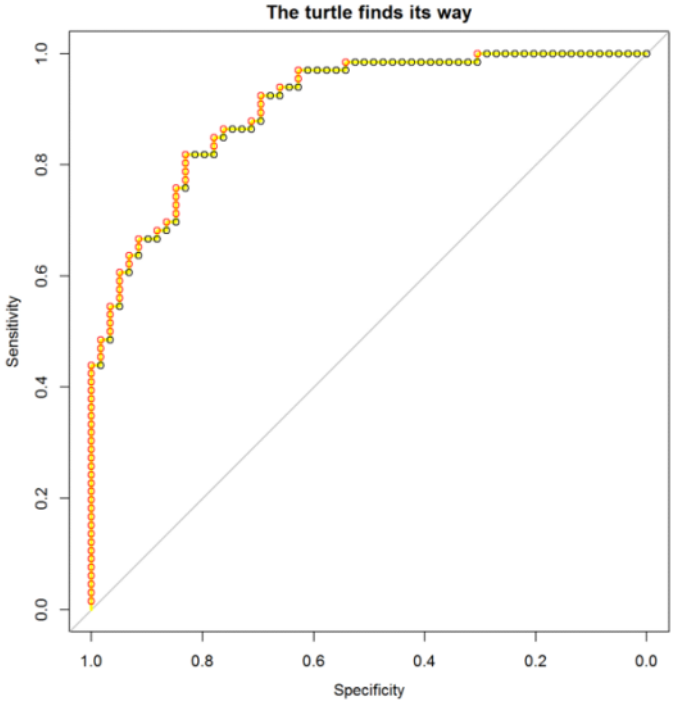
the model. By squaring the difference, the metric disregards the difference between over-prediction and under-prediction

3) Relative absolute error (RAE): the relative absolute difference between expected and actual values; relative because the mean difference is divided by the arithmetic mean
4) Relative squared error (RSE): normalizes the total squared error of the predicted values by dividing by the total squared error of the actual values
5) Coefficient of determination: often referred to as R^2, represents the predictive power of the model as a value between 0 and 1. Zero means the model is random (explains nothing); 1 means there is a perfect fit. Caution should be used in interpreting R^2 values, as low values can be entirely normal and high values can be suspect

iii. ROC Curves

1) Commonly used to characterize the sensitivity/specificity tradeoffs for a binary classifier. The calculation has two steps:
   a) Sort the observed outcomes by their predicted scores with the highest scores first
   b) Calculate the cumulate True Positive Rate (TPR) and True Negative Rate (TNR) for the ordered observed outcomes
2) The function takes in the labels and the scores

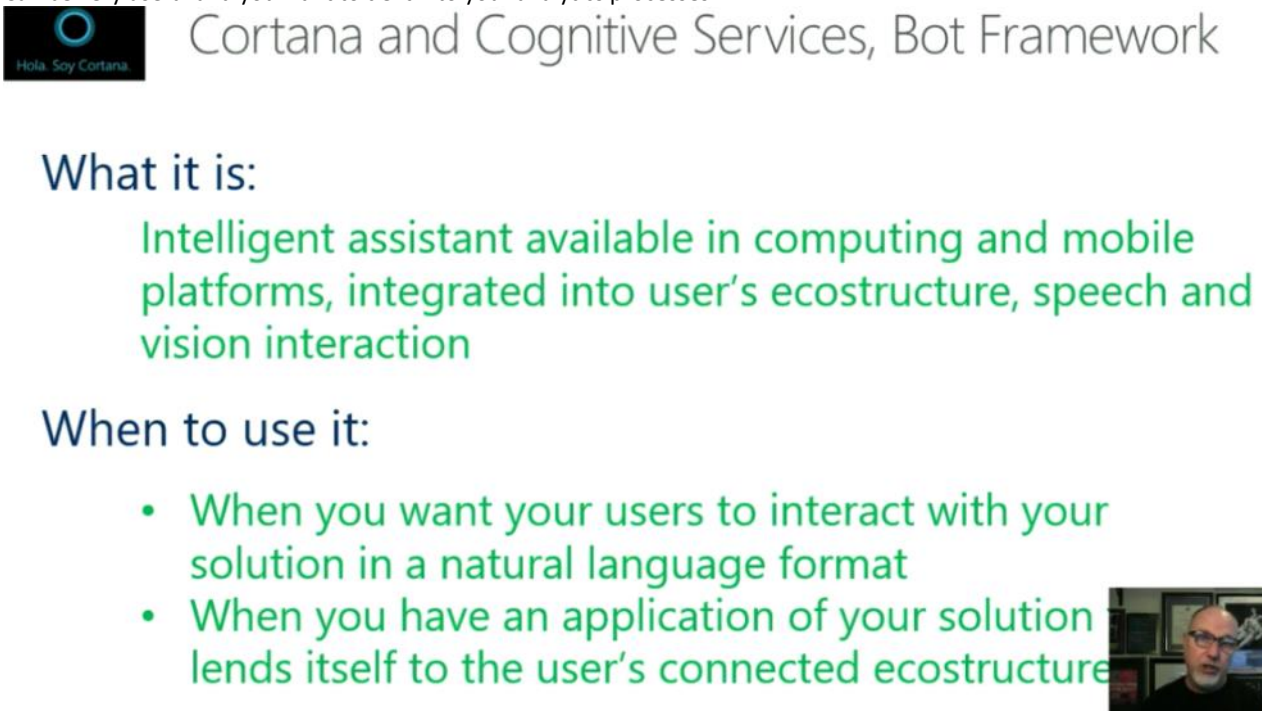
iv.



f. Lab: create and run an experiment in Azure ML
g. MRS Platform
   i. Discover insights and make better decisions with R server
      i. Transform your business with scalable, enterprise-grade R-based data analytics using your data and existing investments
   ii. Analyze data where it lives
   iii. Innovate using AI
   iv. Build mission-critical apps faster

Module 5 Research

Friday, June 16, 2017 12:33 PM

1. **Deployment**
    a. Learning objectives
        i. Deploy a solution using storage
        ii. Deploy a solution using an API
        iii. Deploy a solution using code
        iv. Deploy a solution using the service bus
    b. The data science process and platform
    c. The team data science process
        i. Deployment
            1) Goal
                a) Models and pipeline are deployed to a production or production-like environment for final user acceptance
            2) How to do it
                a) Operationalize the model: deploy the model and pipeline to a production or production-like environment for application consumption
                b) To be operationalized, the models have to be exposed with an open API interface that is easily consumed from various applications such as online website, spreadsheets, dashboards or line of business and backend apps.
    d. The Cortana intelligence platform
        i. Cortana/Cognitive service/Bot framework
            i. Intelligent processing
            ii. Can be very useful and you want to tie it into your analytics processes
            iii. 

        ii. MRS

Microsoft R Server (MRS)

i.

What it is:

A scalable, highly-performing R engine used in on-prem, in-cloud, and in-service areas

When to use it:

- When you need to use the R language and environment for data processing at scale

iii. Power BI

i. Active visualization tool

Power BI

What it is:

Interactive Report and Visualization creation for computing and mobile platforms

ii.

When to use it:

- When you need to create and view interactive reports that combine multiple datasets
- When you need to embed reporting into an application
- When you need customizable visualizations
- When you need to create shared datasets, repo
  dashboards that you publish to your team

iii. As you click on things data changes and you can interact with the data

e. Options for operationalization
   i. Azure storage services REST API
      1) The REST APIs offer programmatic access to the blob/queue/table/file services in azure or in the dev environment, via the storage emulator. All storage services are accessible
   ii. How are you going to access it? How are you going to publish it? How will you monitor/update it as needed?
f. Azure ML - publish as a web service
   i. Narrow it down to just one model
   ii. Convert the training experiment to a predictive experiment
   iii. Deploy as classic web service
   iv. Test the web service
   v. Manage the web service -- in azure portal or azure ML webservices portal
   vi. From the slides:
      1) Build the model
      2) Run the experiment
      3) Create a predictive experiment
         a) Click on web service set up button
         b) Click on create a predictive experiment
      4) Modify the experiment

        a) Connect the web service input to score
        b) Add a project columns after score and only allow scored labels and probabilities
        c) Move the web service output to after project columns module
    5) Run, then deploy the web service
    6) Test the web service using the test dialogue
    7) Review the sample code on the RRS help page
g. Lab: create an azure ML API
h. Cortana and the bot framework
    i. Cortana is getting better and better every day
    ii. Cortana is set up on Windows 10 (and now on Android phones it can be configured) to be your personal assistant. Even an app for iPhones now
    iii. Bots
        1) Have been out a little over a year
        2) Huge growth
        3) Bot framework is a bunch of libraries of services that help you build conversational interfaces for your apps
        4) A way to create conversational things that connect to channels
        5) Building dialogues
            a) Talking to a bot
            b) Typing to a bot
            c) Touching things through adaptive cards
        6) Bot-relative analytics
i. Power BI
    i. A reporting system for multiple data sources -- a data visualization tool
    ii. Available in
        1) Web portal
        2) Power bi desktop
        3) Microsoft excel
        4) Mobile apps (iOS, Android, Windows)
    iii. Author
        1) Connect to data
        2) Shape data
        3) Model data
        4) Report on the data
    iv. Publish
        1) Local
        2) To service
j. Creating a useful report
    i. Find and verify your source data
        1) Locate the most authoritative data
        2) Get permission where required
    ii. Shape and model the data
        1) Find the main message
        2) Remove extraneous data
        3) Change the types to be more effective in interactive layouts
    iii. Select the right graphic
        1) Scale, increments, axes
        2) Simple is better
        3) Tell a story
k. Lab: analyzing data in Power BI
l. Code and storage
    i. When you're deploying your solution and handing it off -- need to think about how you will deploy and manage it. Are you going to use code or the portal or some third party solution/provider?
    ii. The second link tells you more about how to create a web role to display messages
m. Lab: connecting to an Azure SQL Database
n. Azure service bus
    i. Whether an app or service runs in the cloud or on prem, it often needs to interact with other apps or services. Service Bus provides a broadly useful way to do this
    ii. Options
        1) Letting apps send and receive messages through a simple queue
        2) A queue with a publish-and-subscribe mechanism
        3) Just a connection, no queue required

iii. Multi-tenant cloud service - shared by multiple users. Each user creates a namespace and then defines communication mechanisms needed within that namespace
    - *Queues*, which allow one-directional communication. Each queue acts as an intermediary (sometimes called a *broker*) that stores sent messages until they are received. Each message is received by a single recipient.
    - *Topics*, which provide one-directional communication using *subscriptions*-a single topic can have multiple subscriptions. Like a queue, a topic acts as a broker, but each subscription can optionally use a filter to receive only messages that match specific criteria.
    - *Relays*, which provide bi-directional communication. Unlike queues and topics, a relay doesn't store in-flight messages; it's not a broker. Instead, it just passes them on to the destination application.

    From <https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-fundamentals-hybrid-solutions>
iv. https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-fundamentals-hybrid-solutions

o. Lab: work with the azure service bus

Module 6 Research

Friday, June 16, 2017 12:33 PM

1. **Customer Acceptance**
    a. Learning objectives
        i. Alter your solution
        ii. Work with your customer to use the solution
        iii. Hand over the solution to the customer
    b. The data science process and platform
    c. The team data science process
        i. Customer acceptance
            1) Goal
                a) Finalize the product deliverables: confirm that the pipeline, the model, and their deployment in a production environment are satisfying customer objectives
            2) How to do it
                a) System validation - confirm it meets customer needs
                b) Project hand off to the entity that will run the system in production
    d. The Cortana intelligence platform
        i. Azure ML
            i. ML models and operationalize them

What it is:

A multi-platform environment and engine to create and deploy Machine Learning models and API's

ii. When to use it:

- When you need to create predictive analytics
- When you need to share Data Science experiments across teams
- When you need to create call-able API's for ML functions
- When you also have R and Python experience o Data Science team

    e. Customer handoff and acceptance
        i. The customer should validate that the system meets their business needs and answers the question with acceptable accuracy to deploy the system to production for use by their client application. All the documentation is finalized and reviewed. A hand-off of the project to the entity responsible for operations is completed. This could be, for example, a IT or customer data science team or an agent of the customer that is responsible for running the system in production
    f. Altering and maintaining the solution
    g. Creating and maintaining a custom solution
    h. Monitoring and reporting on the solution - Application Insights
        i. Extensible Application Performance Management (APM) service for web developers on multiple platforms. Use it to monitor your live app. It will automatically detect performance anomalies. It includes powerful analytics tools to help you diagnose issues and to understand what user actually do with your app. It's designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .Net, node.js, and J2EE, hosted on prem or in the cloud. It integrates with your

devOps process and has connection points to a variety of dev tools.

ii. Install a small instrumentation package in your app, and set up an app insights resource in the portal. The instrumentation monitors your app and sends telemetry data to the portal.

iii. What does it monitor?
    1) Request rates, response times, and failure rates
    2) Dependency rates, response times, and failures rates
    3) Exceptions
    4) Page views and load performance
    5) AJAX calls
    6) User and session counts
    7) Performance counters
    8) Host diagnostics
    9) Diagnostic trace logs
    10) Custom events and metrics

iv. How do I use app insights?
    1) Monitor
        a) Install App Insights on your app and
            a) Set up a dashboard
            b) Discover which are the slowest and most failing requests
            c) Watch live stream when you deploy a new release, to know immediately about any degradation
    2) Detect, diagnose
        a) When you receive an alert or discover a problem
            a) Assess how many users are affected
            b) Correlate failures with exceptions, dependency calls and traces
            c) Examine profiler, snapshots, stack dumps and trace logs
    3) Build, measure, learn
        a) Measure the effectiveness of each new feature that you deploy
            a) Plan to measure how customers use new UX or business features
            b) Write custom telemetry into your code
            c) Base the next dev cycle on hard evidence from telemetry

i. Re-train an Azure ML Experiment

i. As trends and variables that influence the model's parameters change over time, ideally this pipeline should also support recurring automated retraining and updates to the model with latest training data.

> With Azure ML you typically first setup your scoring and training experiments, then two separate web service endpoints for each experiment. Next, you can use the AzureMLBatchExecution activity with Data Factory to do both scoring of incoming data against the latest model hosted by the scoring web service and scheduled retraining with latest training data. The scoring web service endpoint also exposes an Update Resource method that can be used to update the model used by the scoring web service. This is where the new AzureMLUpdateResource activity comes into picture. You can use this activity now to take the model generated by the training activity and provide it to the scoring web service to update the model for scoring, on a schedule, all automated with your existing data factory pipeline.
>
> From <https://azure.microsoft.com/en-us/blog/retraining-and-updating-azure-machine-learning-models-with-azure-data-factory/?v=17.23h>

j. Project close-out document

i. Exit report of project for customer
    1) Overview
    2) Business domain
    3) Business problem
    4) Data processing
    5) Modeling, validation
    6) Solution architecture
    7) Benefits
        a) Company benefit (internal only)
        b) Customer benefit
    8) Learnings
        a) Project execution
        b) Data science/engineering
        c) Domain
        d) Product

        e) What's unique about this project, specific challenges
    9) Links
    10) Next steps
    11) Appendix
k. Lab: monitoring your solution