

AWS Certified Machine Learning – Speciality Examination (MLS-C01)



Curriculum

- Data Engineering (20%)
- Exploratory Data Analysis (24%)
- Modeling (36%)
- Implementation and Operations (20%)

Data Engineering

The background of the slide is a light beige color with several abstract illustrations. On the left, there is a grey silhouette of a person's head and shoulders. In the center, there are two interlocking grey gears. Below the gears, a grey hand is shown holding a large grey cube. To the right of the hand, there is a network diagram consisting of several grey circles connected by thin grey lines, with a dotted line path leading from the bottom left towards the top right.

- Storage Solutions

- S3 Data Lakes
- DynamoDB

- Transformation

- Glue
- Glue ETL



Data Engineering

- Streaming

- Kinesis
- Kinesis Video Streams

- Workflow Management Tools

- Data Pipelines
- AWS Batch
- Step Functions

Exploratory Data Analysis



- Data Science

- scikit-learn
- Data Distributions
- Trends and Seasonality

- Analysis Tools

- Athena
- Quicksight
- Elastic Map Reduce (EMR)
- Apache Spark



Exploratory Data Analysis

- Feature Engineering
 - Imputation methods
 - Outliers
 - Binning/Categorizing Data
 - Log transforms
 - One-hot encoding
 - Scaling and Normalization

Modeling



- Deep Learning

- Multi-layer Perceptrons (MLPs)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)
- ANN – Tuning and Regularization Techniques

- SageMaker

- Architecture
- Built-in Algorithms
- Automatic Model Tuning
- SageMaker Integration with other services - Spark

Modeling



- High-level AI Services

- Comprehend
- Translate
- Polly
- Transcribe
- Lex
- Rekognition
- Additional Services – Personalize, Forecast, Textract etc
- DeepLens

- Evaluating and Tuning

- Confusion Matrix
- RMSE
- Precision and Recall
- F1 Score
- ROC / AUC

Implementation and Operations



- Sagemaker Operations
 - Using containers
 - Security with SageMaker
 - Choosing instance types
 - A/B testing
 - Tensorflow integration
 - SageMaker Neo and GreenGrass
 - SageMaker Pipes
 - Elastic Inference
 - Inference Pipelines

An illustration with a light beige background featuring various icons: a person silhouette in a circle, two interlocking gears, a network of nodes and lines, a hand holding a laptop, a document icon, and a folder icon. The text "Data Engineering" is centered in a dark grey box.

Data Engineering

AWS S3 Overview

- S3 allows for storing objects (files) in buckets (directories)
- Buckets must have a globally unique name
- The full path of the objects is called 'Key'.
Example:
 - `<bucketname>/<filename>.txt`
 - `<bucketname>/<foldername>/<filename>.txt`
- The maximum object size that can be stored: 5TB

AWS S3 for Machine Learning

- Backbone for many AWS ML services (Ex: SageMaker)
- Core service for Data Lake
 - Infinite size, no provisioning
 - 99.999999999% durability
 - S3 allows for decoupling (segregating) storage for all the compute based services.
Examples:
 - EC2, Athena, Redshift, Rekognition, Glue
- Centralized Architecture – all the data at the same place
- Object Storage – supports any file format
- Common formats for ML – CSV, JSON, Parquet, ORC, Avro, Protobuf

AWS S3 Data Partitioning

- Pattern for speeding up range queries (Eg: AWS Athena)
- Partitioning Examples:
 - By Date:
`s3://<bucketname>/<dataset>/year/month/day/hour/<datafile>.csv`
 - By Product: `s3://<bucketname>/<dataset>/product-id/<datafile>.csv`
- We should choose the partitioning type based on use case
- Some tools like Kinesis and Glue can help with partitioning

AWS S3 Storage Tiers

- Amazon S3 Standard – General Purpose (GP)
- Amazon S3 Standard – Infrequent Access (IA)
- Amazon S3 One Zone-Infrequent Access
 - Cheaper IA with diluted availability
- Amazon S3 Intelligent Tiering
 - New – Amazon determines where to put data to save cost
- Amazon Glacier
 - Archival

AWS S3 Storage Tiers

	Standard	Standard - Infrequent Access	One - Infrequent Access	S3 Intelligent-Tiering	Glacier
Durability	99.999999999%	99.999999999%	99.999999999%	99.999999999%	99.999999999%
Availability	99.99%	99.9%	99.5%	99.90%	NA
AZ	≥3	≥3	1	≥3	≥3
Concurrent facility fault tolerance	2	2	0	1	1
<div><div>Frequently accessed</div><div>Infrequently accessed</div><div>Intelligent (new!)</div><div>Archives</div></div>					

S3 Lifecycle Rules

- In order to save on cost, the lifecycle rules help in moving data between different tiers
- Example:
 - General Purpose (GP) -> Infrequent Access (IA) -> Glacier
- Transition actions – Objects are transitioned to another storage class
 - Move objects from:
 - GP to IA, 60 days post creation
 - IA to Glacier 6 months post creation
- Expiration actions – S3 deletes expired objects on our behalf
 - Log files can be set to delete after a specific period of time

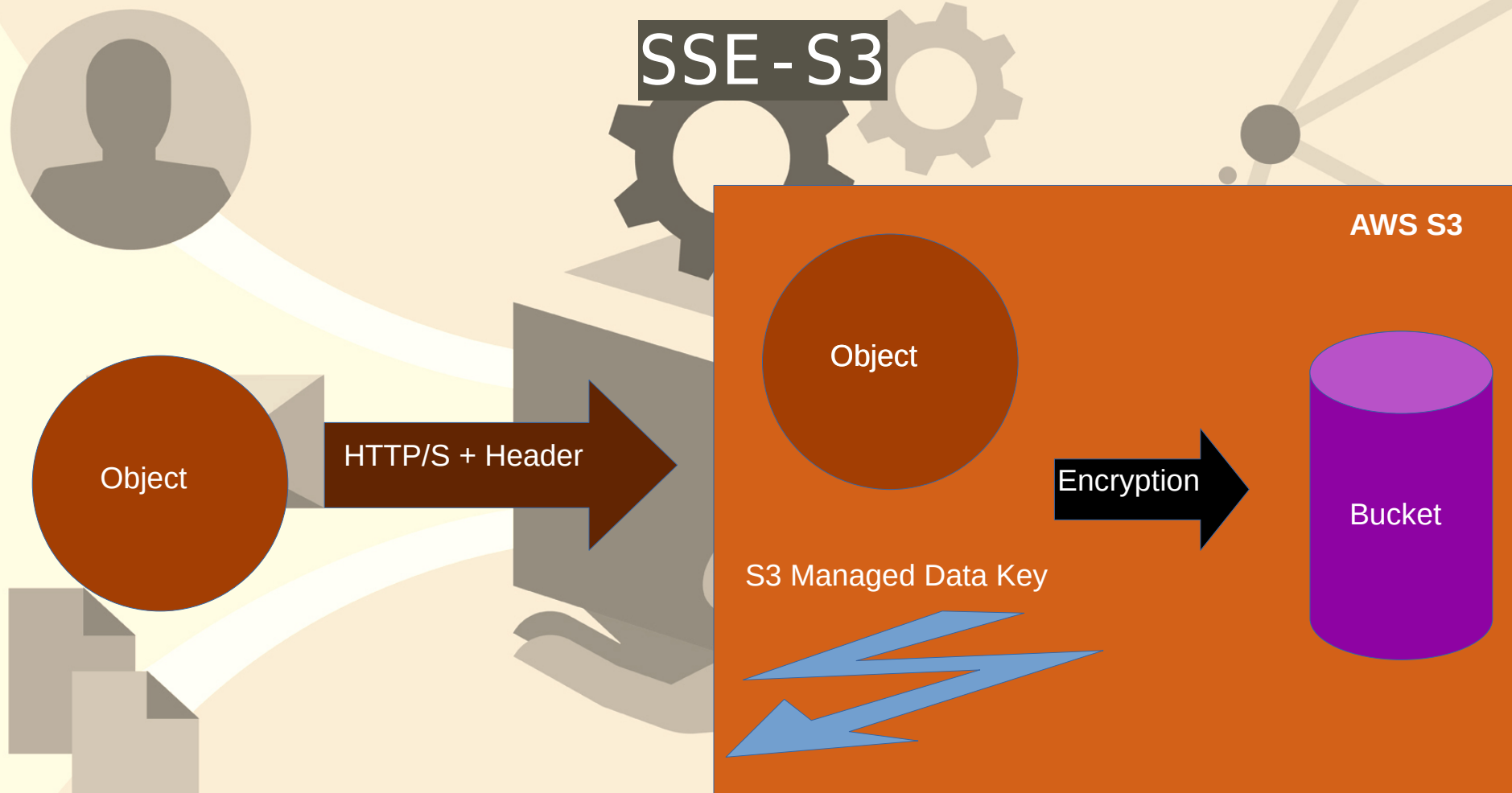
S3 Security - Encryption for Objects

- There are four methods of encrypting objects in S3:
- SSE-S3: Encrypts S3 objects using keys handled and managed by AWS
- SSE-KMS: Use AWS key Management Service to manage encryption keys
 - Additional Security
 - Audit trail for KMS key usage

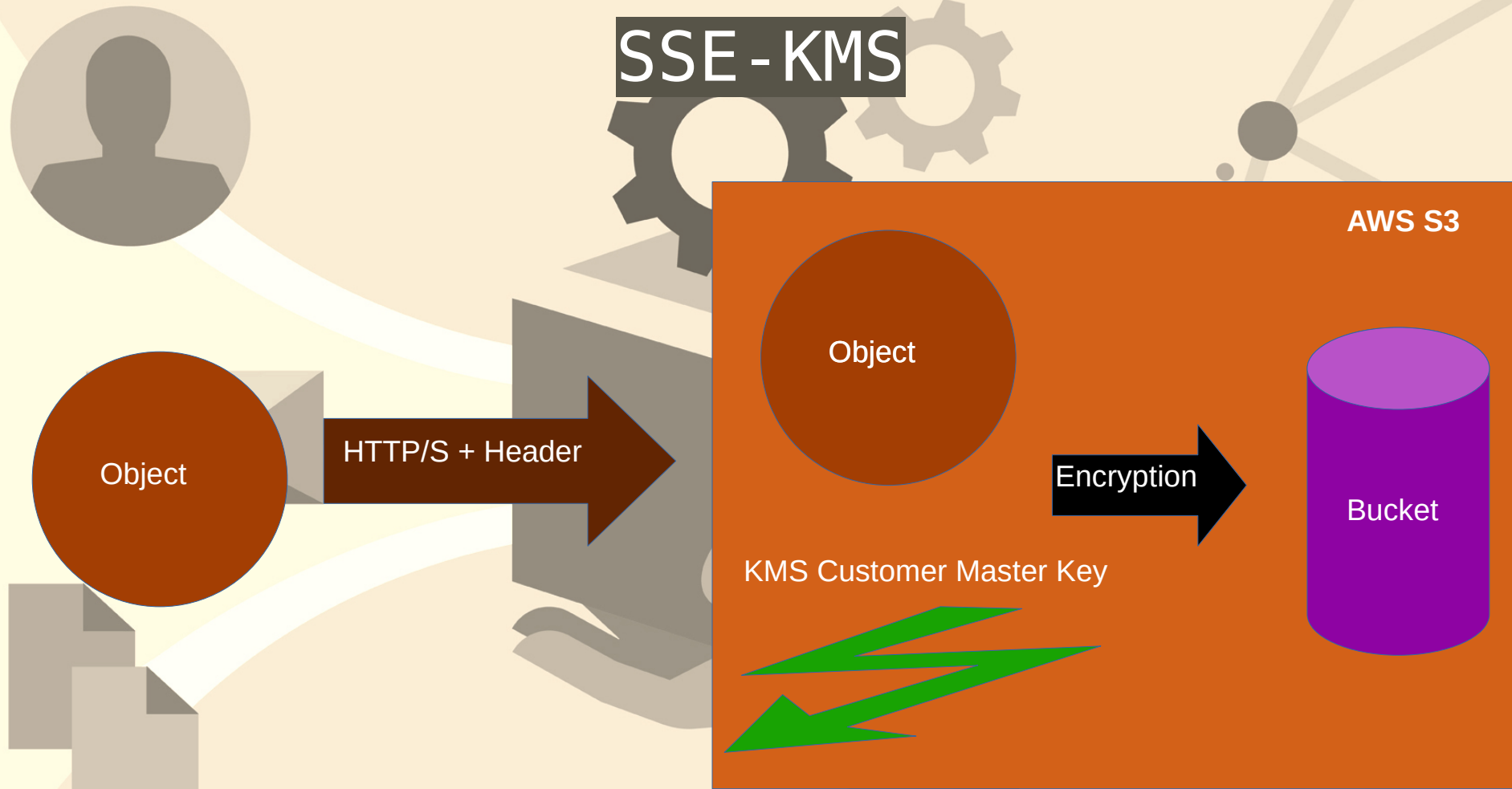
S3 Security - Encryption for Objects

- SSE-C: We need to use our own encryption keys
- Client Side Encryption
- From an ML perspective, SSE-S3 and SSE-KMS will be the most likely used scenarios

SSE - S3



SSE - KMS



S3 Security

- User Based
 - IAM Policies – which API calls the user should be allowed
- Resource Based
 - Bucket Policy – allowing cross account access
 - Object Access Control List (ACL) – more precise control
 - Bucket Access Control List (ACL) – less commonly used

S3 Bucket Policies

- JSON based policies
 - Resources: buckets and objects
 - Actions: Set of API to Allow or Deny
 - Effect: Allow / Deny
 - Principal: The account or user to apply the policy to

S3 Bucket Policies

- Use S3 bucket policies for:
 - Granting public access to the bucket
 - For objects to be encrypted at upload
 - Grant access to another account (Cross Account)

S3 Security – Points to Remember

- Networking – VPC Endpoint Gateway
 - Allow traffic to stay within your VPC
 - Make sure the private services (Eg: SageMaker) can access S3
- Logging and Audit:
 - S3 access logs can be stored in other S3 bucket
 - API calls can be logged in AWS CloudTrail
- Tagged Based (combined with IAM and bucket policies)

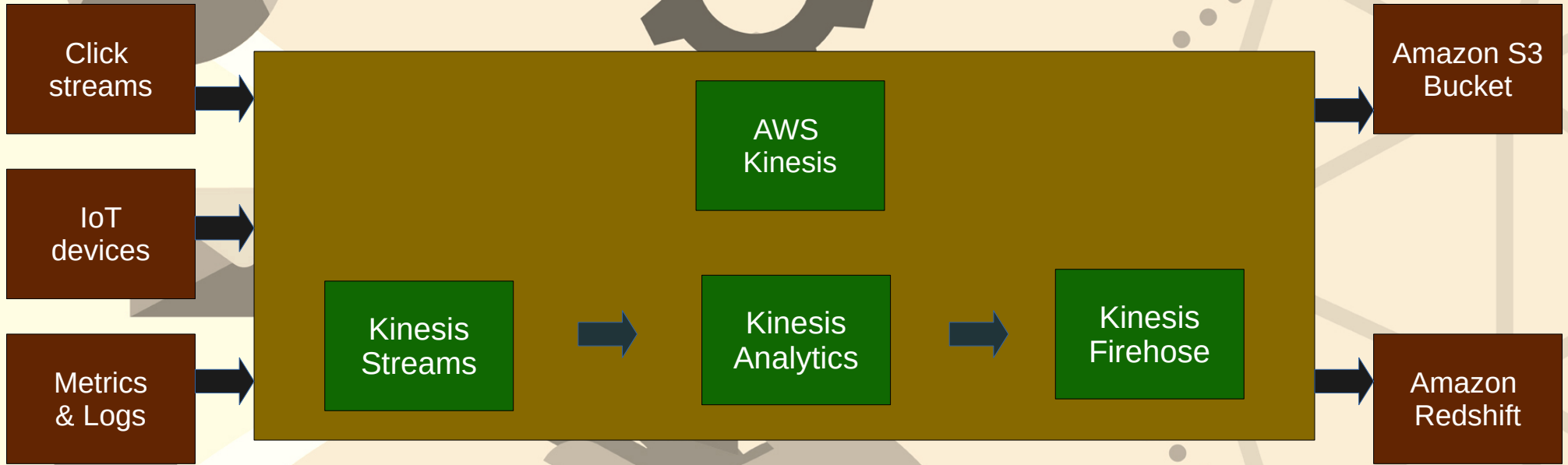
AWS Kinesis - Overview

- Kinesis is a managed alternative to Apache Kafka
- Great for application logs, metrics, IoT, click streams etc
- Any reference to 'real-time' in the exam is an indication of relation to Kinesis
- Great for streaming processing frameworks (Spark, NiFi etc)
- Data is automatically replicated synchronously to 3 AZs

AWS Kinesis – Key Services

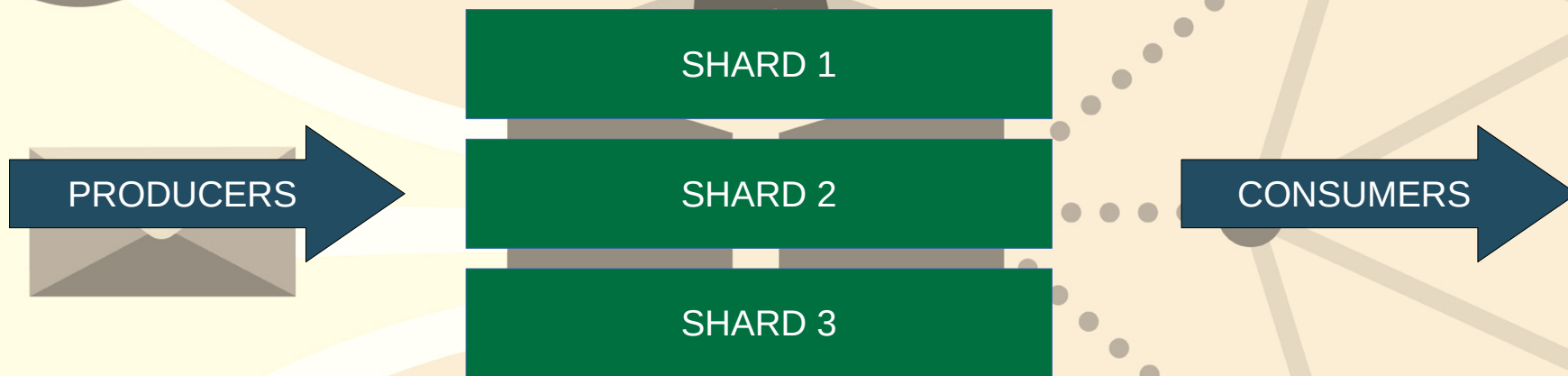
- Kinesis Streams – low latency streaming ingest at scale
- Kinesis Analytics – perform real-time analytics on streams using SQL
- Kinesis Firehose – load streams into S3, Redshift, Elasticsearch and Splunk
- Kinesis Video Streams – meant for streaming video in real-time

AWS Kinesis – Architecture



Kinesis Streams - Overview

- Streams are divided into ordered Shards / Partitions



- Shards have to be provisioned in advance (capacity planning)

Kinesis Streams - Overview

- Data retention is 24 hours by default and can go upto 7 days
- Ability to reprocess/replay data
- Multiple applications can consume the same stream
- Once data is inserted in Kinesis, it cannot be deleted (data immutability)
- Records can be upto 1 MB in size – fine for streaming use cases but not for large data analysis

Kinesis Data Streams - Limits

- **Producers:**
 - 1MB/s or 1000 messages/s write speed per shard
 - If exceeded – 'ProvisionedThroughputException'
- **Consumer (Classic):**
 - 2MB/s read speed per shard across all consumers
 - Max 5 API calls/s/shard across all consumers
- **Data Retention:**
 - By default – 24 hours
 - Can be extended to upto 7 days

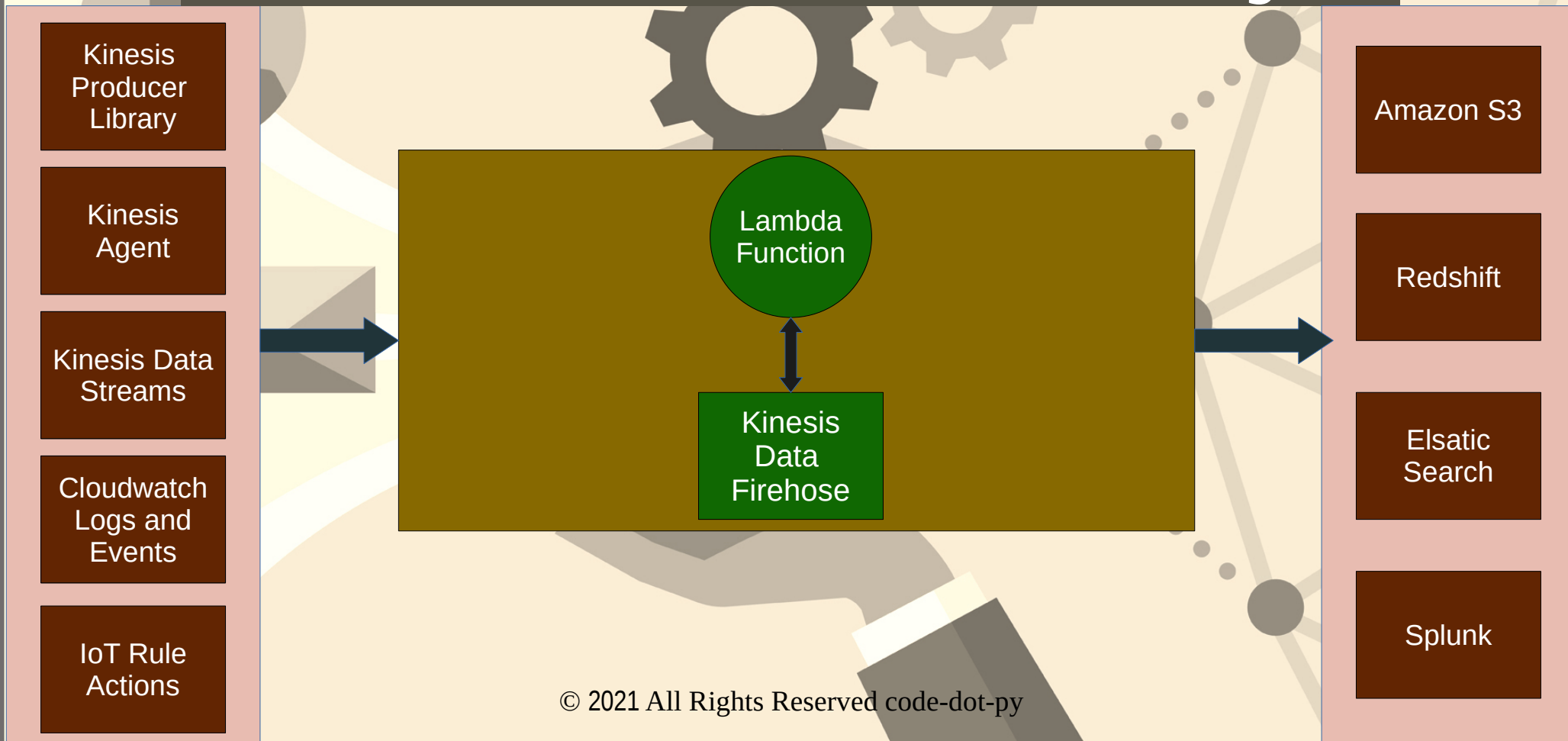
Kinesis Data Firehose

- Fully managed service, no administration
- Near real time (60 seconds latency minm for non full batches)
- Can perform data injestion into the following four services:
 - Redshift
 - Amazon S3
 - ElasticSearch
 - Splunk
- Automatic scaling

Kinesis Data Firehose

- Supports many data formats
- Data conversion from CSV/JSON to Parquet/ORC (only for S3)
- Data transformation through AWS Lambda (CSV => JSON)
- Supports compression when target is Amazon S3 (GZIP, ZIP and SNAPPY)
- Pay for the amount of data going through Firehose

Kinesis Data Firehose - Diagram



Kinesis Data Streams vs Firehose

- Streams

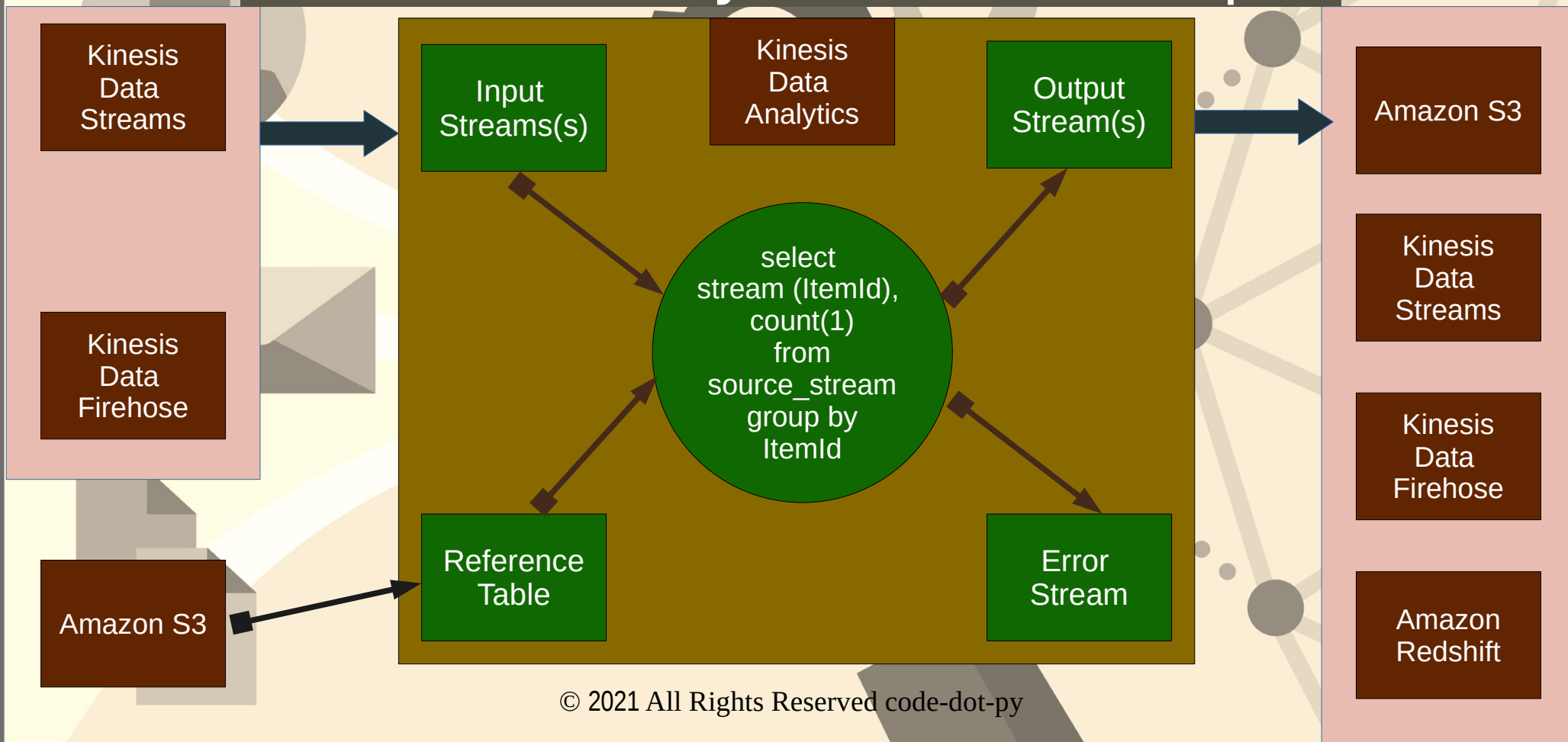
- Supports custom code writing for producer/consumer
- Real-time applications (latency ~200ms for classic and ~70ms for enhanced fan-out)
- Must manage scaling ourselves (shard splitting/merging)
- Data Storage for 1 to 7 days, replay capability, multi consumers

Kinesis Data Streams vs Firehose

- Firehose

- Deliver or ingestion service
- Fully managed. Can send to S3, Splunk, Redshift, ElasticSearch
- Serverless data transformations with Lambda
- Near real time (lowest buffer time is 60 seconds)
- Automated scaling
- No data storage – no replay capability

Kinesis Analytics – In Depth



Kinesis Data Analytics – Use Cases

The background of the slide is a light beige color with several abstract illustrations. On the left, there is a dark grey silhouette of a person's head and shoulders. In the center, there are two interlocking gears, one dark grey and one light grey. On the right, there is a network diagram consisting of several dark grey circular nodes connected by thin grey lines. A dotted line also connects some of these nodes.

- Streaming ETL
 - select columns, make simple transformation on streaming data
- Real-time metric generation
 - live leaderboard for a mobile game
- Responsive analytics
 - look for certain criteria and build alerting

Kinesis Data Analytics - Features

- Pay only for the resources consumed – not cheap
- Serverless and scales automatically
- Use IAM permissions to access streaming source and destination(s)
- SQL or Flink to write the computations
- Schema discovery
- Lambda can be used for preprocessing

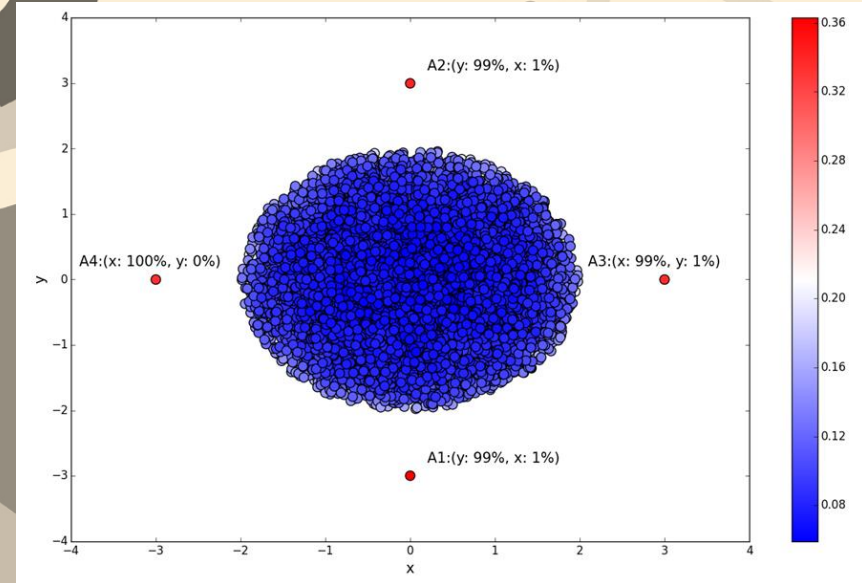
Machine Learning on Kinesis Data Analytics

- There are two algorithms that can be used:
 - Random Cut Forest
 - Hotspots

ML on Kinesis Data Analytics

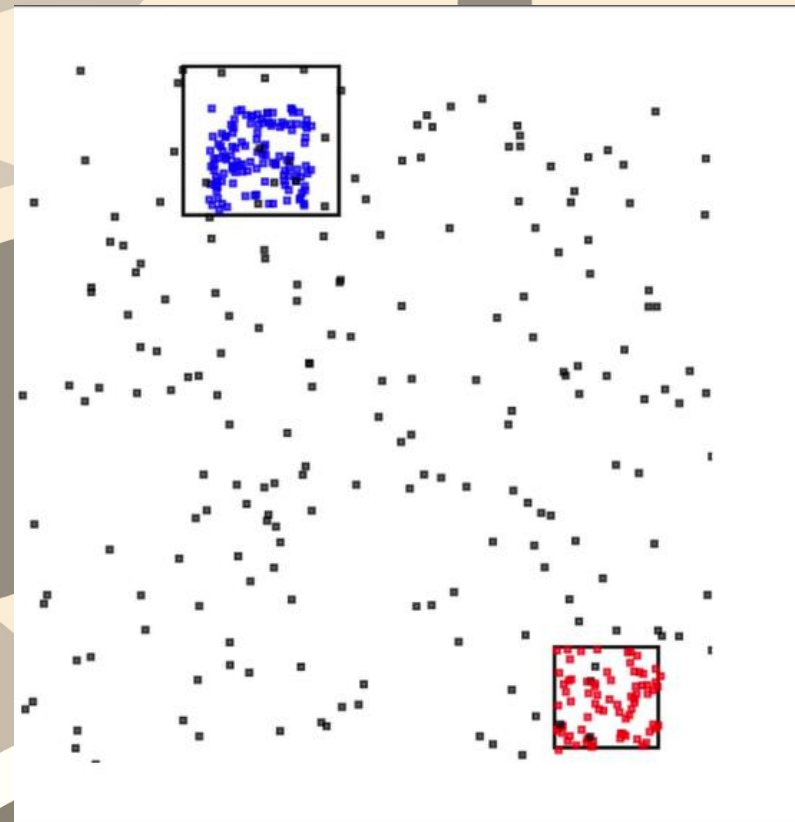
- Random Cut Forest

- Used for anomaly detection in a numerical column data in a stream
- Provided as a SQL like function to use
- Uses recent history to compute the model



ML on Kinesis Data Analytics

- **Hotspot**
 - Locate and return information about relatively dense regions in the data
 - Provided as a SQL like function to use



Kinesis Video Stream

- **Producers**

- Cameras – security, body-worn, action, smartphone
- AWS DeepLens
- Audio and Image feeds
- RADAR data
- One producer per video stream

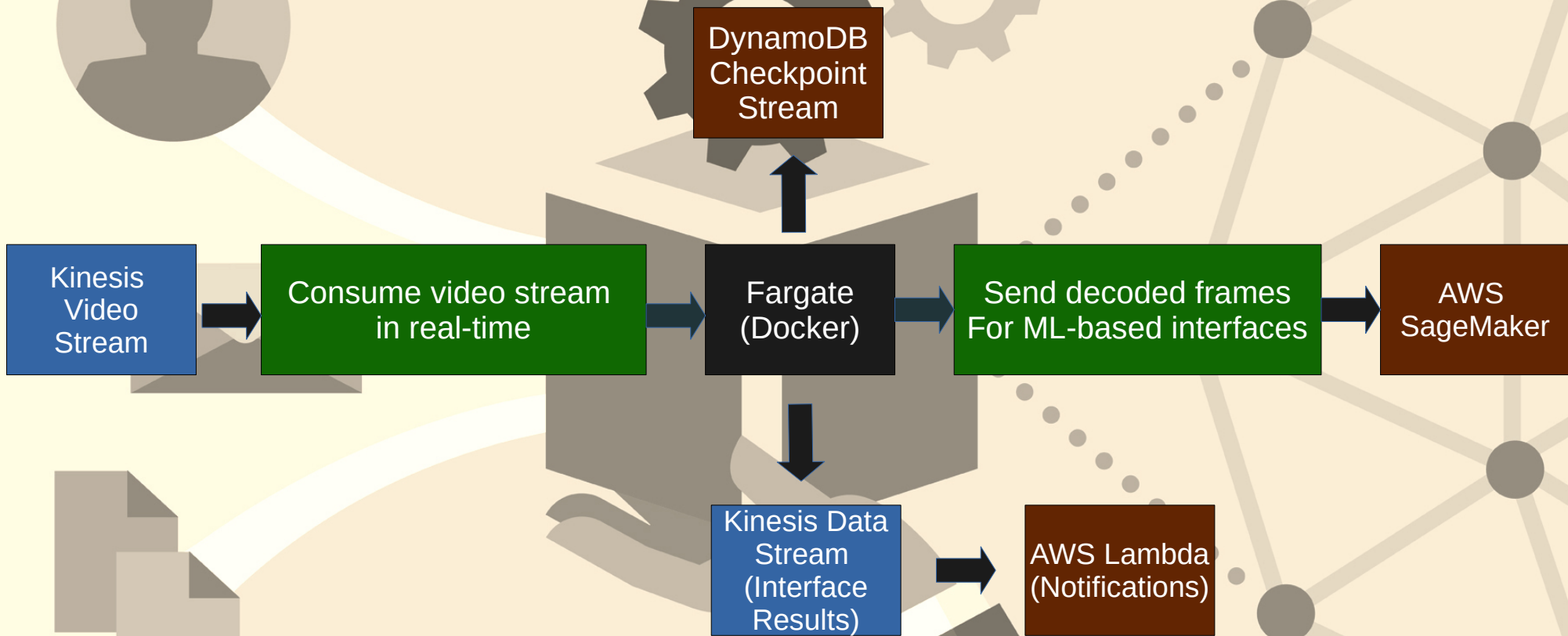
- **Consumers**

- Custom – MXNet, Tensorflow
- AWS SageMaker
- Amazon Rekognition Video

- Data can be retained for 1 hour to upto 10 years

- Video playback capability

Kinesis Video Stream – Use Case



Kinesis Summary

- Kinesis Data Stream – read real-time data and create real-time ML applications on top
- Kinesis Data Firehose – ingest massive data with near-real-time scenario
- Kinesis Data Analytics – real-time ETL/ML algorithms on stream
- Kinesis Video Stream – real-time video stream to create ML applications

AWS Glue

- Glue Data Catalog
 - Metadata repository for all the tables within the account
 - Automated Schema Inferences
 - Schema Versioning
 - Integrates with AWS Athena or redshift Spectrum (schema and data discovery)
 - 'Glue Crawlers' can help building the Glue Data Catalog

AWS Glue

- Glue Data Catalog – Crawlers
 - Crawlers go through the data to infer schemas and partitions
 - They work with – JSON, Parquet, CSV, Relational data etc
 - They work with services like – AWS S3, Redshift, RDS
 - Crawlers can be run on a schedule or on-demand
 - Need IAM role/credentials to access the data stores

AWS Glue and S3 Partitions

- Glue crawler will extract partitions based on how the S3 data is organized
- It has to be thought-of upfront as to how the data will be queried
- Example:
 - A device is sending sensor data every hour
 - Query primarily by time-range -> Organize the bucket like:
 - `s3://my-bucket/dataset/yyyy/mm/dd/device`
 - Query primarily by device -> Organize the bucket like:
 - `s3://my-bucket/dataset/device/yyyy/mm/dd/`

AWS Glue ETL

- Extract, Transform and Load
- Transform Data, Clean Data, Enrich Data before performing any analysis
- Glue ETL Features:
 - Generate ETL code in Python or Scala, the code can also be updated/modified
 - Can also work with custom Spark or PySpark scripts
 - Targets can be S3, JDBC (RDS, Redshift etc) or in Glue Data Catalog
- Fully managed, cost effective – pay only for the resources consumed
- Jobs are executed on a serverless Spark platform
- Glue Scheduler to schedule the jobs
- Glue Triggers to automate job runs based on 'events'

AWS Glue ETL - Transformations

- **Bundled Transformations:**

- DropFields, DropNullFields – remove (null) fields
- Filter -specify a function to filter records
- Join – to enrich data
- Map – add fields, delete fields, perform external lookups

- **Machine Learning Transformations:**

- FindMatches ML: Identify duplicate or matching records in your dataset, even when the records do not have a common unique identifier and no fields match exactly. Useful for data de-duplication

- **Format conversions: CSV, JSON, Avro, Parquet, ORC, XML**

- **Apache Spark Transformations (Eg: k-means)**

AWS Data Store for Machine Learning

- Redshift

- Data Warehousing, SQL Analytics, OLAP
- Load data from S3 to Redshift
- Use Redshift Spectrum to query data directly in S3

- RDS, Aurora

- Relational Store, SQL, OLTP
- Must provision servers in advance

AWS Data Store for Machine Learning

- **DynamoDB**
 - NoSQL data store
 - Serverless – provision read/write capacity
 - Useful to store a machine learning model served by your application
- **S3**
 - Object storage
 - Serverless, infinite storage
 - Integration with most AWS Services

AWS Data Store for Machine Learning

The background of the slide is a light beige color with a subtle illustration. On the left, a hand in a dark suit sleeve holds a laptop. In the center, there are several interlocking gears of different sizes. On the right, there is a network diagram consisting of several dark grey circular nodes connected by thin grey lines, with a dotted line path winding through them.

- ElasticSearch

- Indexing of data
- Search amongst data points
- Clickstream Analytics

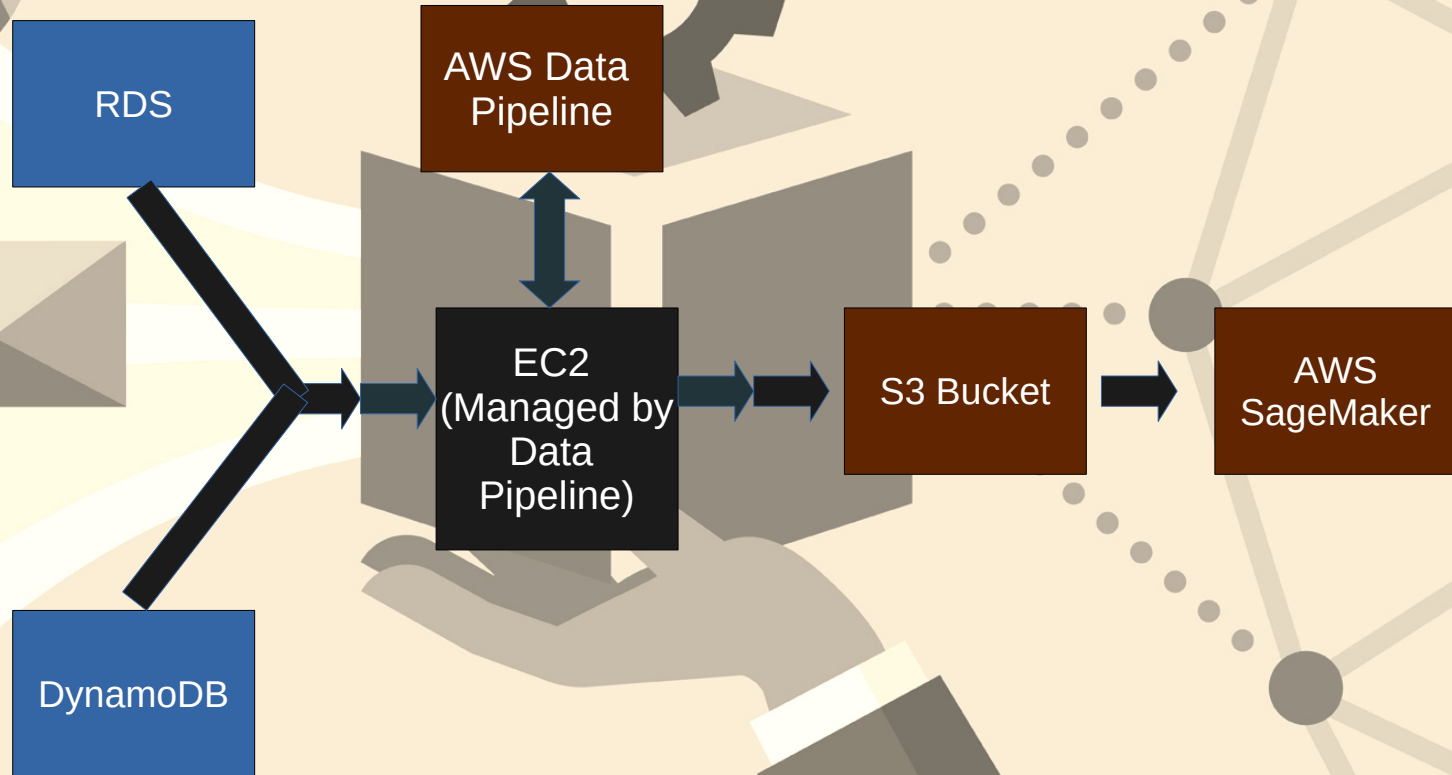
- ElastiCache

- Caching mechanism
- Not really used for Machine Learning

AWS Data Pipeline Features

- Destinations include S3, RDS, DynamoDB, Redshift, EMR
- Manages task dependencies
- Retries and notifies on failures
- Data source may be on-premise
- Highly available

AWS Data Pipeline Example





AWS Data Pipeline vs Glue

• Glue:

- Glue ETL: Runs Apache Spark Code
- Scala or Python based
- Focused on ETL
- Not focused on configuring or managing resources
- Catalogs data for Athena or Redshift Spectrum

• Data Pipeline:

- Orchestration service
- More control over:
 - The environment
 - Compute resources that run the code
 - Code itself
- Allows access to EC2 or EMR instances
- Creates resources in same account

AWS Batch

- Run batch jobs as Docker images
- Dynamic provisioning of the instances (EC2 and Spot Instances)
- Optimal quantity and type based on the volume and requirements
- No need to manage clusters, fully serverless
- Just pay for the underlying EC2 instances
- Schedule batch jobs using CloudWatch Events
- Orchestrate Batch Jobs using AWS Step Functions

AWS Batch vs Glue

• Glue:

- Glue ETL: Runs Apache Spark Code
- Scala or Python based
- Focused on ETL
- Not focused on configuring or managing resources
- Catalogs data for Athena or Redshift Spectrum

• Batch:

- For any computing job regardless of the job
- Must be provided with a Docker image
- The resources are created within the account and managed by Batch
- For a non-ETL requirement, Batch is a better option

AWS DMS – Database Migration Service

- Quickly and securely migrate databases to AWS
- Resilient and self-healing
- The source database remains available during the migration
- Supports:
 - Homogeneous migrations – Oracle to Oracle
 - Hetrogeneous migrations – SQL Server to Aurora
- Continuous Data Replication using CDC
- An EC2 instance is needed to perform the replication task

AWS DMS vs Glue

• Glue:

- Glue ETL: Runs Apache Spark Code
- Scala or Python based
- Focused on ETL
- Not focused on configuring or managing resources
- Catalogs data for Athena or Redshift Spectrum

• DMS:

- Continuous Data Replication
- Service for Database migration
- No data transformation
- Post migration, ETL can be performed

AWS Step Functions

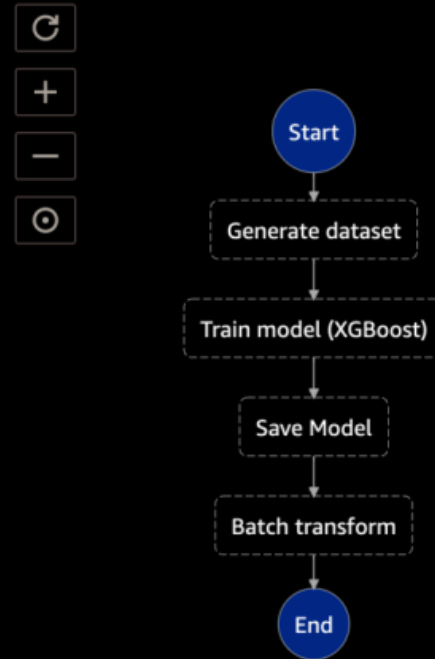
- Used to orchestrate and design workflows
- Easy visualizations
- Advanced Error Handling and Retry mechanism outside the code
- Audit of the history of workflows
- Ability to 'Wait' for an arbitrary amount of time
- Maximum execution time of the state machine is 1 year

Step Functions – Examples

Train a Machine Learning Model

```
{
  "StartAt": "Generate dataset",
  "States": {
    "Generate dataset": {
      "Resource": "<GENERATE_LAMBDA_FUNCTION_ARN>",
      "Type": "Task",
      "Next": "Train model (XGBoost)"
    },
    "Train model (XGBoost)": {
      "Resource": "arn:
<PARTITION>:states::sagemaker:createTrainingJob.sync",
      "Parameters": {
        "AlgorithmSpecification": {
          "TrainingImage": "<SAGEMAKER_TRAINING_IMAGE>",
          "TrainingInputMode": "File"
        },
        "OutputDataConfig": {
          "S3OutputPath": "s3://<S3_BUCKET>/models"
        },
        "StoppingCondition": {
          "MaxRuntimeInSeconds": 86400
        }
      },
      "ResourceConfig": {
        "InstanceCount": 1,
        "InstanceType": "ml.m4.xlarge",

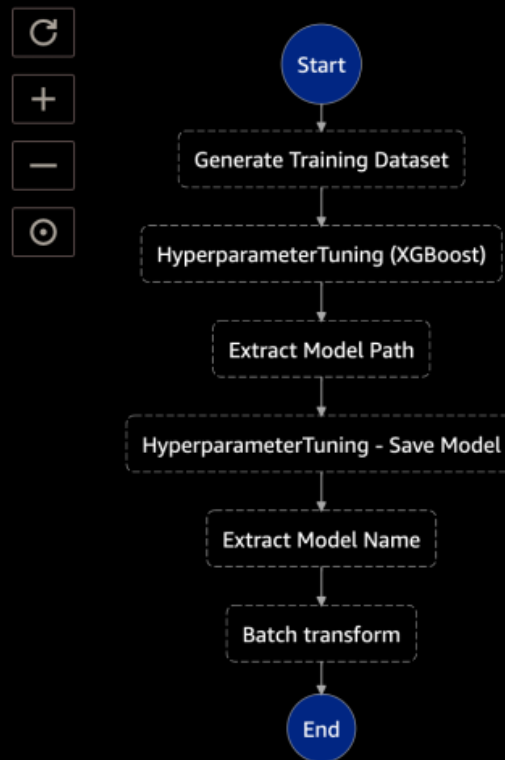
```



Step Functions – Examples

Tune a Machine Learning Model

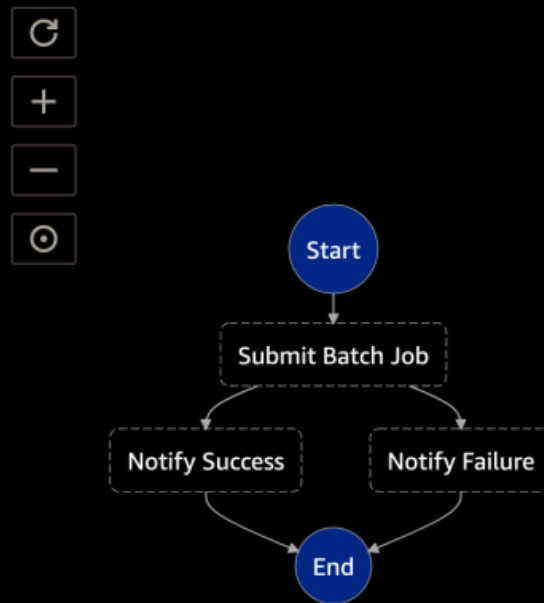
```
{
  "StartAt": "Generate Training Dataset",
  "States": {
    "Generate Training Dataset": {
      "Resource": "<GENERATE_LAMBDA_FUNCTION_ARN>",
      "Type": "Task",
      "Next": "HyperparameterTuning (XGBoost)"
    },
    "HyperparameterTuning (XGBoost)": {
      "Resource": "arn:
<PARTITION>:states::sagemaker:createHyperParameterTuningJob.sync",
      "Parameters": {
        "HyperParameterTuningJobName.$": "
<JOB_NAME_FROM_LAMBDA>",
        "HyperParameterTuningJobConfig": {
          "Strategy": "Bayesian",
          "HyperParameterTuningJobObjective": {
            "Type": "Minimize",
            "MetricName": "validation:rmse"
          },
          "ResourceLimits": {
            "MaxNumberOfTrainingJobs": 2,
            "MaxParallelTrainingJobs": 2
          },
          "ParameterRanges": {
```



Step Functions – Examples

Manage a Batch Job

```
{
  "Comment": "An example of the Amazon States Language for
notification on an AWS Batch job completion",
  "StartAt": "Submit Batch Job",
  "TimeoutSeconds": 3600,
  "States": {
    "Submit Batch Job": {
      "Type": "Task",
      "Resource": "arn:<PARTITION>:states:::batch:submitJob.sync",
      "Parameters": {
        "JobName": "BatchJobNotification",
        "JobQueue": "<BATCH_QUEUE_ARN>",
        "JobDefinition": "<BATCH_JOB_DEFINITION_ARN>"
      },
      "Next": "Notify Success",
      "Catch": [
        {
          "ErrorEquals": [ "States.ALL" ],
          "Next": "Notify Failure"
        }
      ]
    },
    "Notify Success": {
      "Type": "Task",
      "Resource": "arn:<PARTITION>:states:::sns:publish",
```



A conceptual illustration for Exploratory Data Analysis. It features a central hand holding a laptop, with gears above it, a network diagram to the right, and icons for a person, envelope, and documents on the left. The title 'Exploratory Data Analysis' is centered in a dark banner.

Exploratory Data Analysis

Exploratory Data Analysis – Kick Off

- Python has taken over as one of the most prominent language for Data Science and ML
- The exam, however, will never test on the knowledge of Python
- Knowing Python is a significant advantage
- Some of the key packages – Pandas, Numpy etc

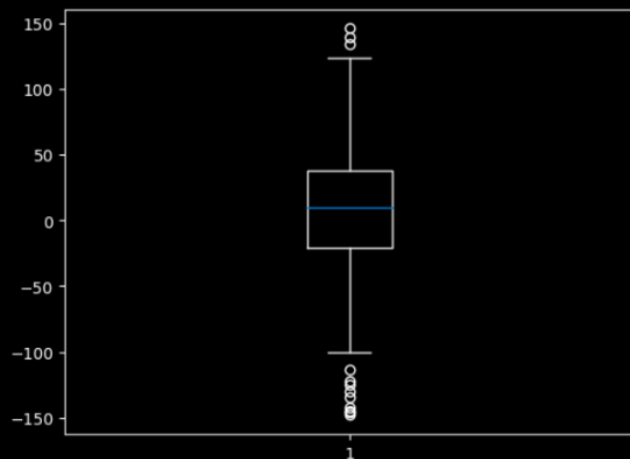
Pandas

- A python library for slicing and dicing the data
 - Data Frames
 - Series
- It is one of the most powerful data manipulation and data preprocessing libraries extensively used for handling Data Science related requirements with Python
- Pandas interoperates with numpy

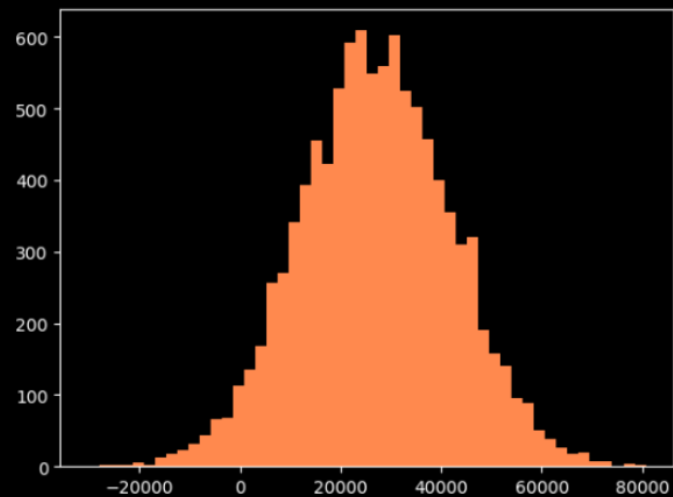
Matplotlib

- This library helps in visualizing the data

```
In [13]: uniformSkewed = np.random.rand(100) * 100 - 40  
high_outliers = np.random.rand(10) * 50 + 100  
low_outliers = np.random.rand(10) * -50 - 100  
data = np.concatenate((uniformSkewed, high_outliers, low_outliers))  
plt.boxplot(data)  
plt.show()
```

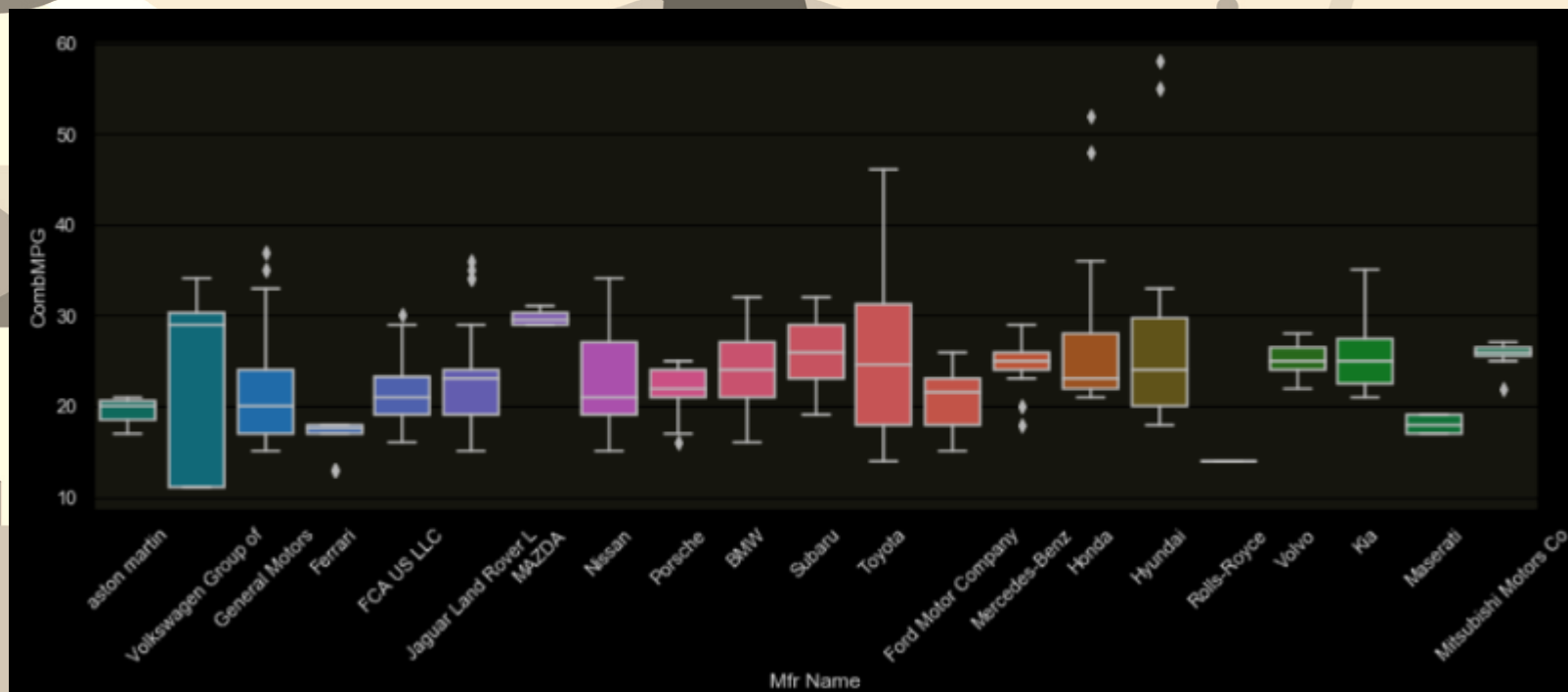


```
In [12]: incomes = np.random.normal(27000, 15000, 10000)  
plt.hist(incomes, 50)  
plt.show()
```



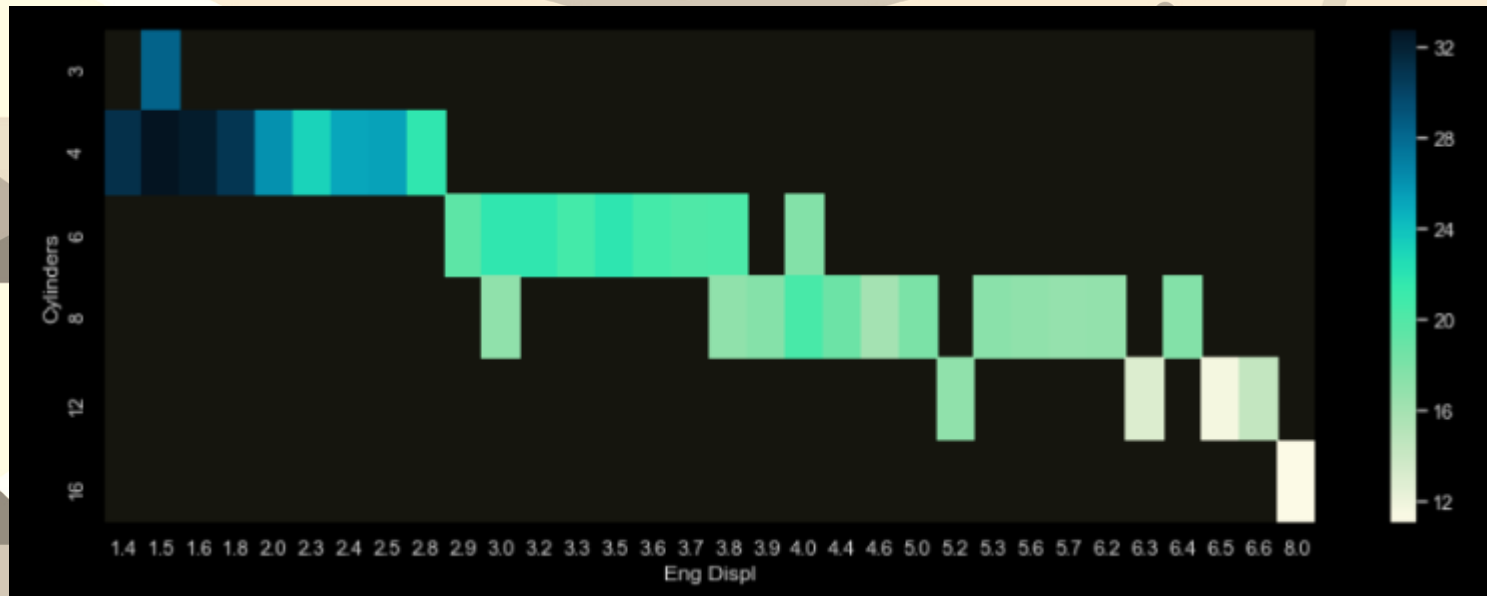
Seaborn

- Example: Box-and-whiskers plot



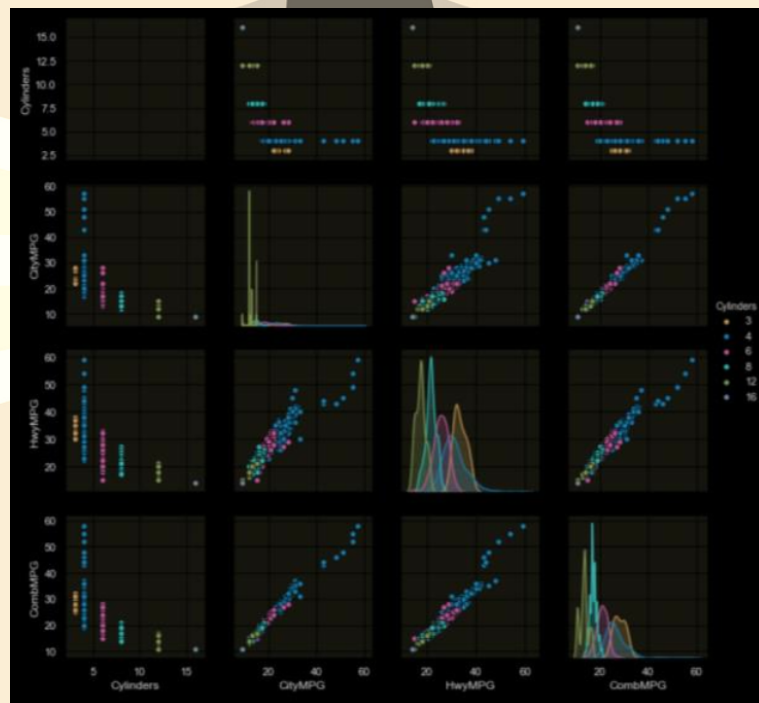
Seaborn

- Example: Heat-Map



Seaborn

- Example: Pair plot



scikit_learn

- Python library for machine learning models
- Pre-requisite – some familiarity with Machine Learning
- Designed in such a fashion that application of modules is similar across models and algorithms
- Also has some data pre-processing capabilities that are useful – ‘preprocessing’ module

Data Science – Types of Data



- Numerical
- Categorical
- Ordinal

Numerical Data

- Represents some sort of quantitative measurement
 - Heights of people, page load times, stock prices etc.
- Discrete Data
 - Integer based
 - Often counts of some event
 - How many purchases a customer made in an year?
 - How many times the coin flipped head?
- Continuous Data
 - Has an infinite number of possible values
 - How much time did it take for a user to check out?
 - How much rain fell on a give day?

Categorical Data

- Qualitative data that has no inherent mathematical meaning
 - Gender, Yes/no(binary data), Race, State of Residence, product Category, Political Party etc
- Numbers can be assigned to the categories in order to represent them in a more compact way, but the numbers dont have any mathematical meaning

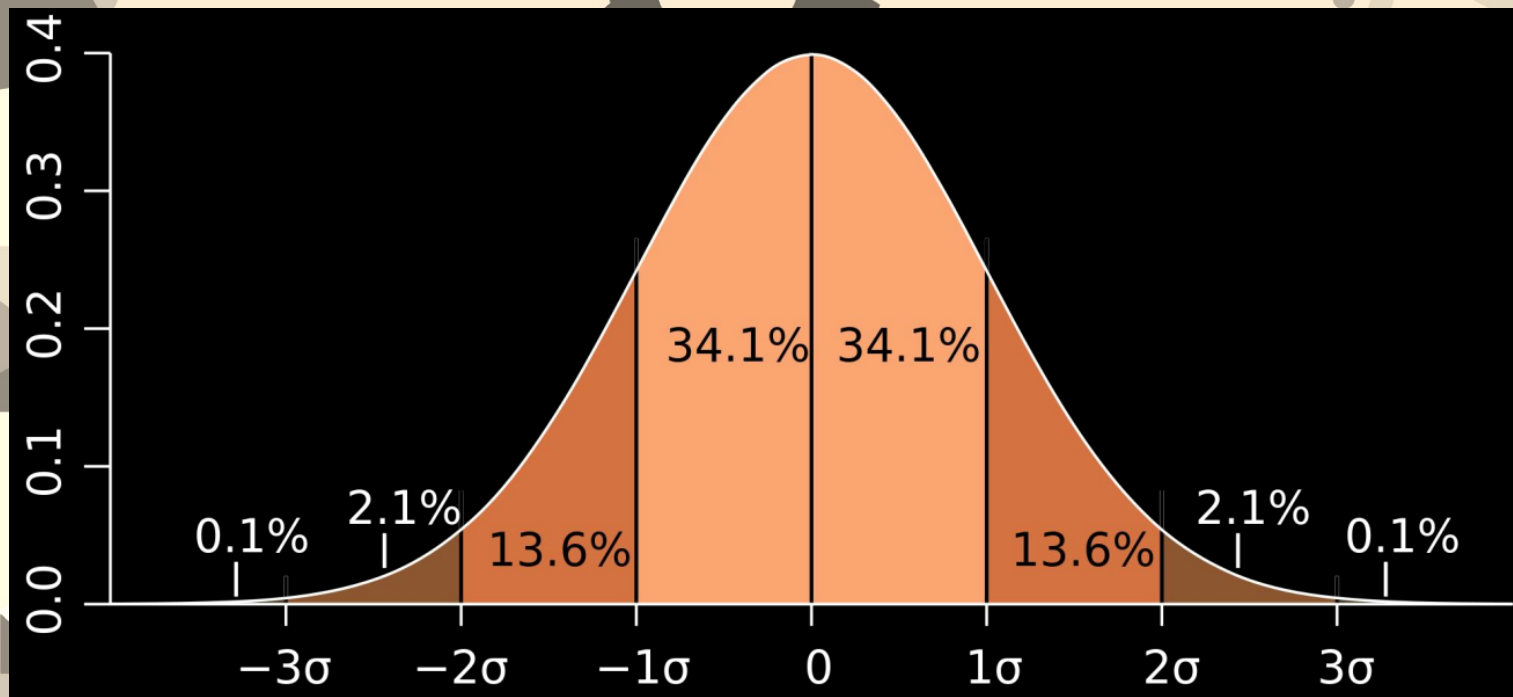
Ordinal Data

- A mix of numerical and categorical data
- Categorical data that has a mathematical meaning
- Example:
 - Movie ratings on the scale of 5 (1-5)
 - Ratings must be one of 1, 2, 3, 4, 5
 - These values have a mathematical meaning
 - 1 is the worst rating, 5 being the best

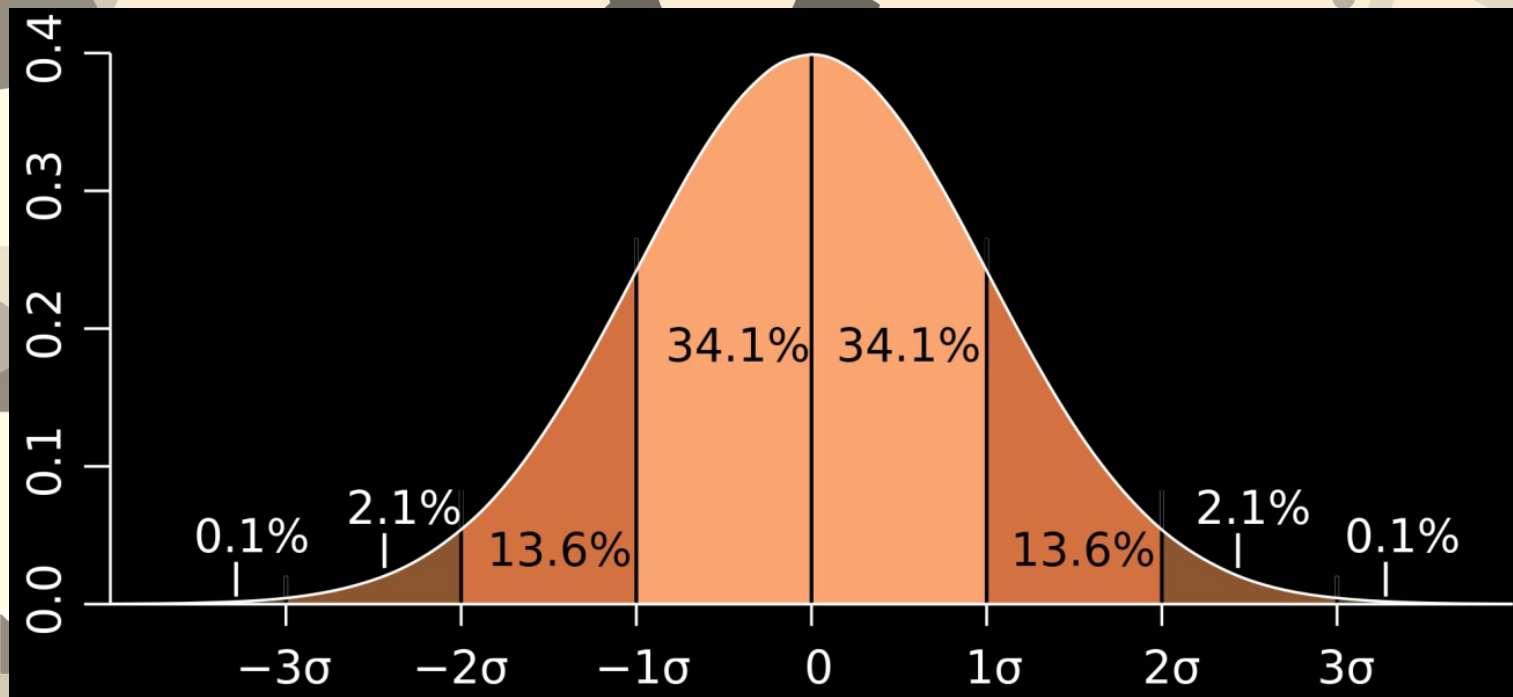
Data Distributions

- Normal Distribution
- Probability Density Function – Continuous Data
- Probability Mass Function – Discrete Data
- Poisson Distribution
- Binomial Distribution
- Bernoulli Distribution – special case of Binomial ($n=1$)

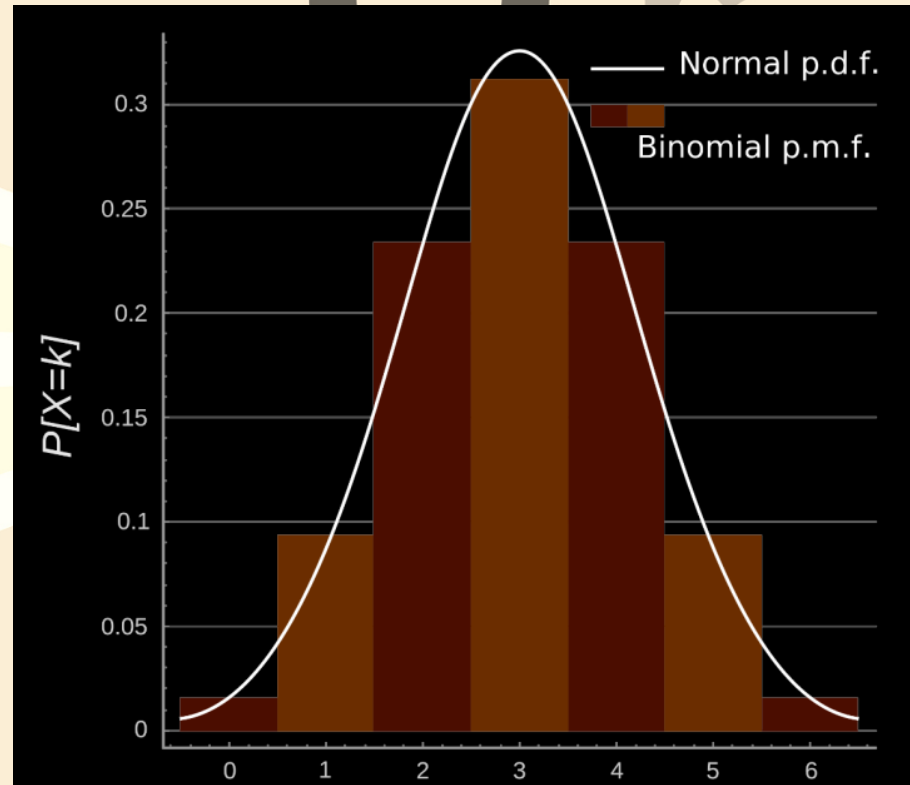
Normal Distribution



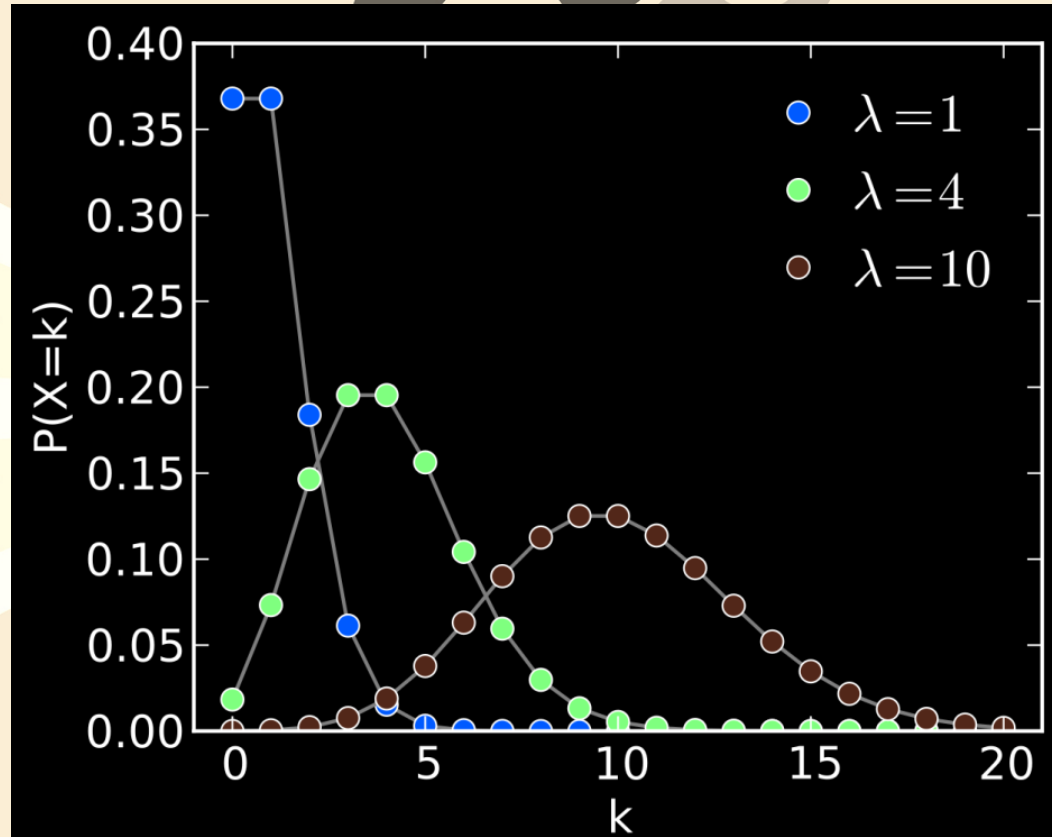
Probability Density Function



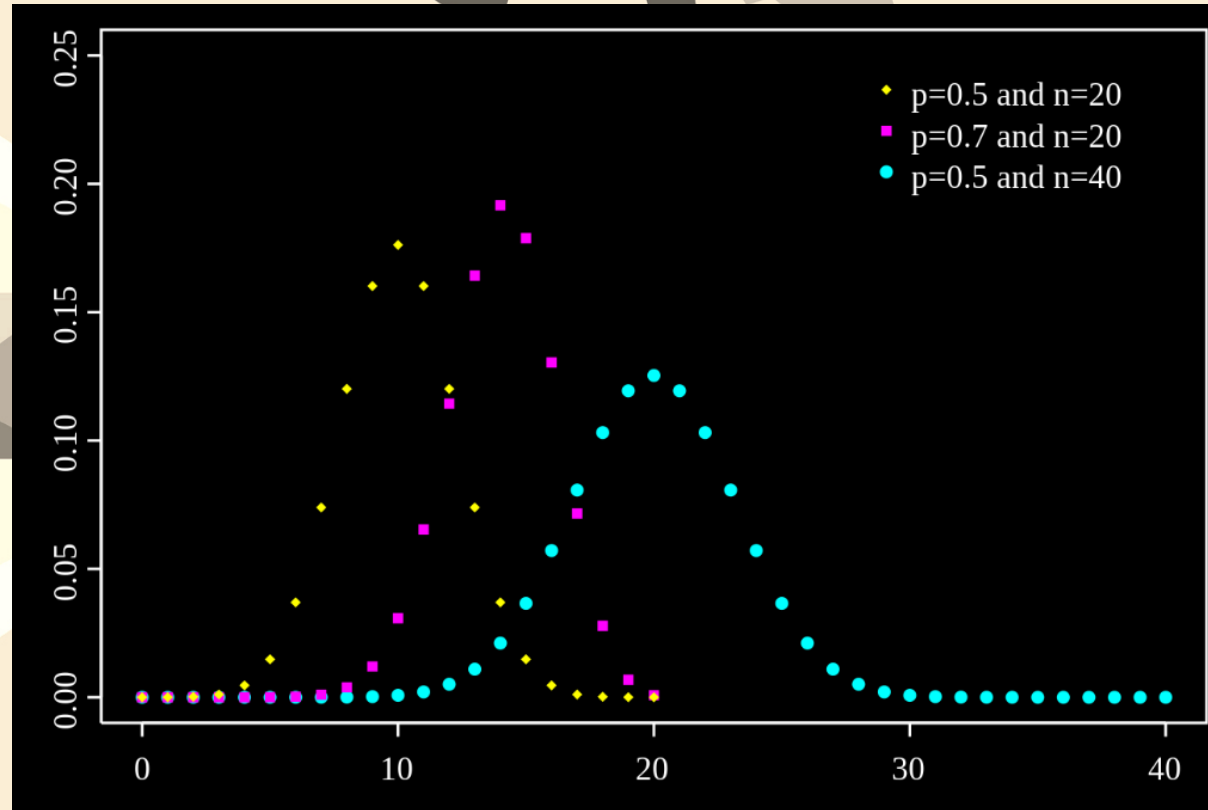
Probability Mass Function



Poisson Distribution Function



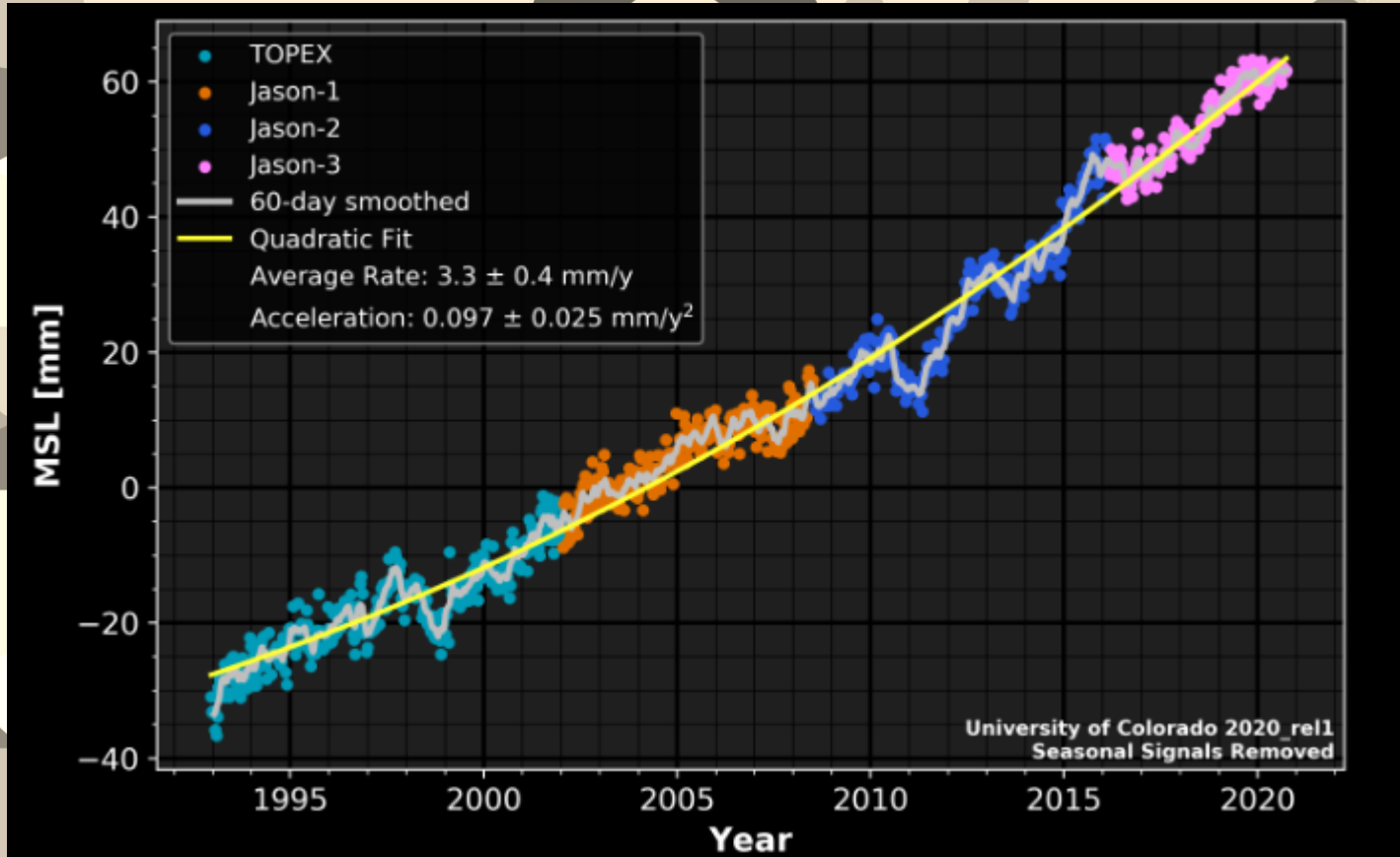
Binomial Distribution



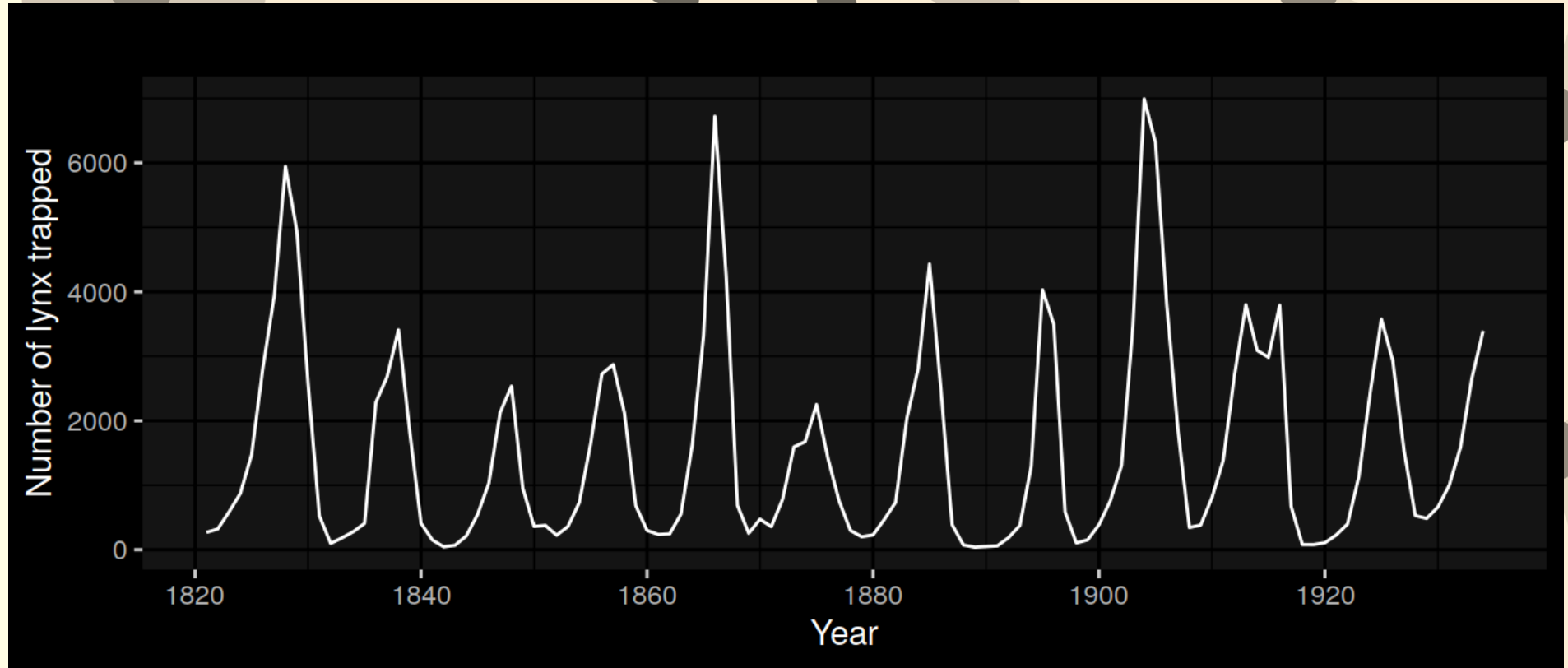
Time Series Analysis

- Series of data points over time
- Discrete data samples taken over discrete points of time over a period of time
- Can depict:
 - Trends
 - Seasonality
 - Or both

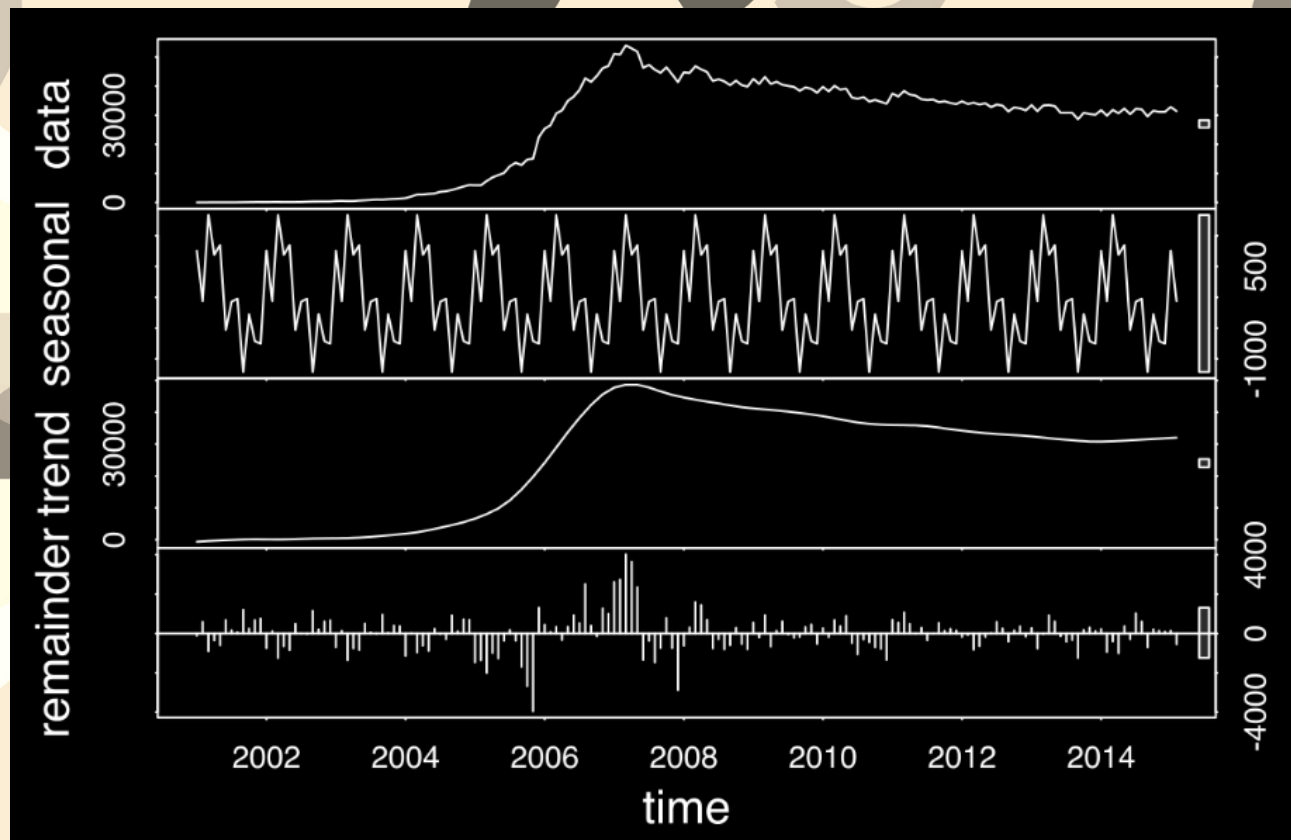
Trends



Seasonality



Both



Noise

- Some variations are just random in nature
- Seasonality + Trends + Noise = Time Series
- Models
 - Additive: If the seasonal variation is constant
 - Multiplicative: If the seasonal variation increases as the trend increases

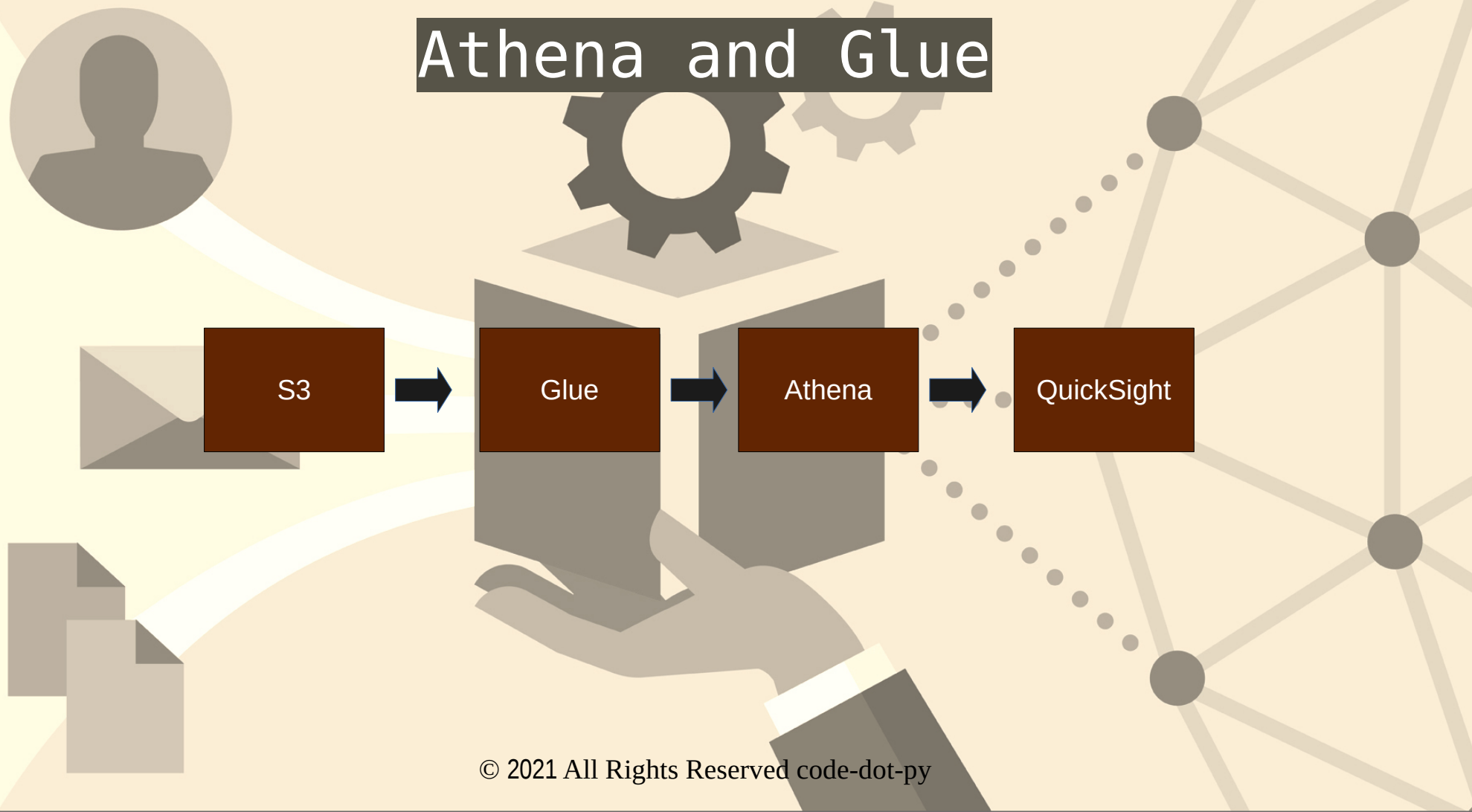
Amazon Athena

- Serverless interactive query service for S3 (SQL)
 - No need to load the data, it stays in S3
- Presto under the hood
- Serverless
- Supports many data formats like CSV, JSON, ORC, Parquet, Avro
- Supports unstructured, semi-structured, structured data

Amazon Athena - Examples

- Ad-hoc queries of web logs
- Querying the staging data before loading it to Redshift
- Analyze CloudTrail/CloudFront/VPC/ELB etc logs in S3
- Integration with Jupyter, Zeppelin, RStudio notebooks
- Integration with QuickSight
- Integration via ODBC/JDBC with other visualization tools

Athena and Glue



Athena Cost Model

- Pay-as-you-go
 - \$5 per TB scanned
 - Successful or cancelled queries count, failed queries don't
 - No charges for DDL – CREATE, ALTER, DROP etc
- Saves a lot of money by using columnar formats
 - ORC, Parquet
 - Save 30-90% and get better performance
- Glue and S3 have their own charges

Athena Security

- Access control:
 - IAM, ACLs, S3 bucket policies
 - AmazonAthenaFullAccess / AWSQuickSightAthenaAccess
- Encrypt results at rest in S3 staging directory
 - Server-side encryption with S3-managed key (SSE-S3)
 - Server-side encryption with KMS key (SSE-KMS)
 - Client-side encryption with KMS key (CSE-KMS)
- Cross-account access in S3 bucket policy possible
- Transport Layer Security (TLS) encrypts in-transit (between Athena and S3)

Athena Anit-patterns

The background of the slide is a light beige color with several abstract illustrations. In the top left, there is a grey silhouette of a person's head and shoulders. In the top center, there are two interlocking grey gears. In the center, a grey hand is holding a 3D cube. To the right of the hand, there is a network diagram consisting of several grey circles connected by thin grey lines. In the bottom left, there are some grey geometric shapes that look like folded paper or documents.

- Highly formatted reports / visualization
 - Use QuickSight for this
- ETL
 - Use Glue for this

Amazon QuickSight

- Business analytics and visualizations in the cloud
- Fast, easy, cloud-powered business analytics service
- Allows all employees in an organization to:
 - Build visualizations
 - Perform ad-hoc analysis
 - Quickly get business insights from data
 - Medium agnostic (any browser, any device)
- Serverless



QuickSight Data Sources

- Redshift
- Aurora / RDS
- Athena
- EC2-hosted databases
- Files (S3 or on-premises):
 - XLS, XLSX
 - CSV, TSV
 - Common or extended log format
- Data preparation also allows for limited ETL

SPICE (QuickSight Processing Engine)

- QuickSight datasets are imported into SPICE:
 - Super-fast, Parallel, In-memory, Calculation Engine
 - Uses columnar storage, in-memory, machine code generation
 - Accelerates interactive queries on large datasets
- Each user gets 10GB of SPICE
- Highly available / durable
- Scales to hundreds of thousands of users



QuickSight Use Cases

- Interactive ad-hoc exploration / visualization of data
- Dashboards and KPIs
- Stories
 - Guided tours through specific views of an analysis
 - Convey key points, thought process, evolution of an analysis
- Analyze / visualize data from:
 - Logs in S3
 - On-premise databases
 - AWS (RDS, Redshift, Athena, S3)
 - SaaS applications like Salesforce
 - Any JDBC/ODBC data source

QuickSight's Machine Learning Insights

- **Anomaly Detection**
 - Random-cut forest algorithm to process data and identify outliers
- **Forecasting**
 - Enables non-technical users to forecast key business matrices
 - Uses random-cut forest to detect seasonalities and trends
- **Auto-narratives**
 - Build simple dashboards automatically that convey the information from the data in easily understandable way



QuickSight Anti-patterns

- Highly formatted canned reports
 - QuickSight is for ad-hoc queries, analysis and visualization
- ETL:
 - Use Glue instead
 - Though, QuickSight can perform some basic transformations

QuickSight Security

- Multi-factor authentication on your account
- VPC connectivity
 - Add QuickSight's IP address range to database security groups
- Row-level security
- Private VPC access
 - Elastic Network Interface, AWS Direct Connect



QuickSight User Management

- Users defined via IAM or email signup
- Active Directory integration is available with QuickSight Enterprise Edition

QuickSight Pricing

- Annual Subscription:
 - Standard: \$9/user/month
 - Enterprise: \$18/user/month
- Extra SPICE capacity(beyond 10GB)
 - \$0.25(standard) \$0.38(enterprise)/GB/month
- Month-to-month:
 - Standard: \$12/GB/month
 - Enterprise: \$24/GB/month
- Enterprise edition
 - Encryption at rest
 - Microsoft Active Directory Integration