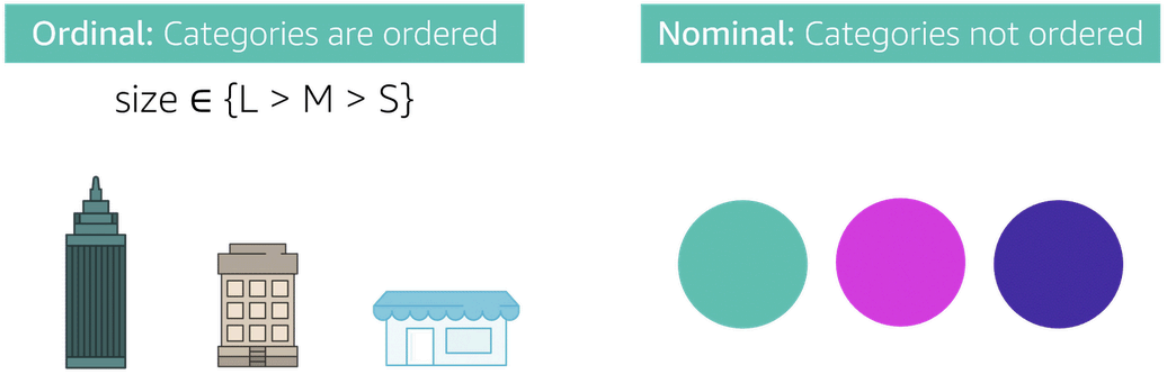
- **Data Types**

  Quantitative : 數值類 ---> 又分成連續和離散

  Qualitative : 分種類 Nominal Data (性別、血型)

  Ordinal : mix 上述兩種 例如：電影五星評分



- 資料結構種類:

  Structured data : 像RDS存的那種

  Unstructured data : 圖片

  Semi structured data: **json, xml, csv**

- **S3** 的 **Storage Tiers**（從經常存取到不常存取排列）

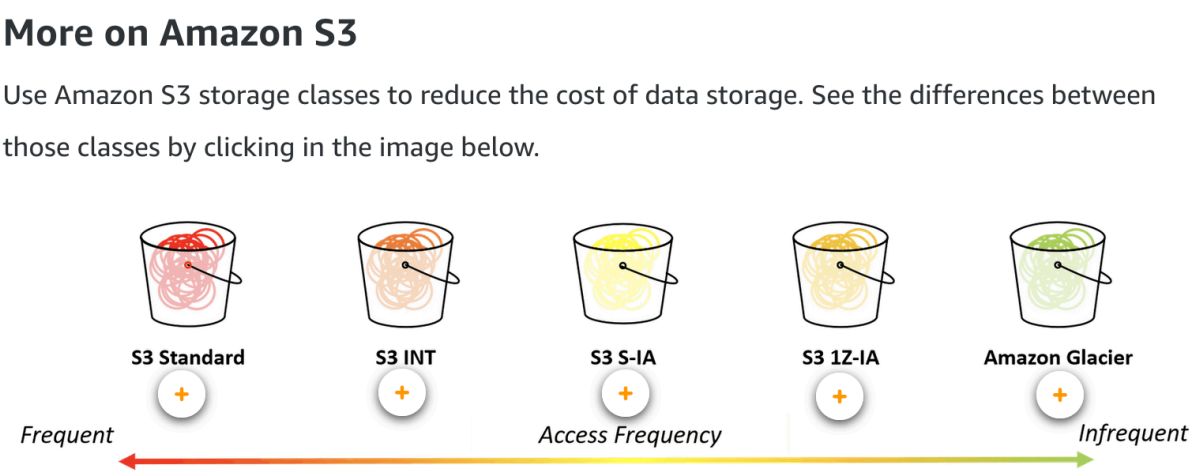
  Standard

  Intelligent-Tiering (varying access)

  Standard-Infrequent Access (long-lived less frequently)

  One Zone-Infrequent Access (long-lived less frequently)

S3 Glacier (archive data)



- **S3 life cycles policy**

  可以設定lifecycle讓obj隔一段時間從Standard移到Standard-IA

  https://docs.aws.amazon.com/zh_tw/AmazonS3/latest/user-guide/create-lifecycle.html
- **S3 Bucket** 加密方式

  共有SSE-S3, SSE-KMS, CSE-KMS, SSE-C, CSE-C
- **S3** 權限:

  預設僅有owner可以access bucket, 其他user可透過以下兩種來設定(都用json設定)

  1. Resource based : ACL (Access Control List)

  2. User based : IAM (Identity and Access Management)
- **AWS Glue**

  Extract, transform, load (ETL) data

  Could be fed into Amazon QuickSight

  Work with Redshift, S3, database in VPC on EC2

  有FindMatches功能可以去除重覆值

  使用 python/scala 作為語言
- **AWS Batch**

  Running batch computing job, 可作ETL

  Serverless , 可排程, 用來整理s3資料

  Using EC2 and spot Instance

  Could be schedule by **CloudWatch Events**

  Could be orchestrates by **AWS Step functions**

- **AWS Step Functions**

  Serverless flow by lambda

  State machine

  適合執行long-running job (但限 15 min 內)
- **AWS Kinesis** 四大服務

  1. Kinesis Video Streams : 影片串流分析

  2. Kinesis Data Streams : 資料流分析，每秒處理GB等級（real time）

  3. Kinesis Data Firehose: 將資料處理並傳入AWS其他服務 Ex: S3 Redshift (near real time)

  4. Kinesis Data Analytics : 可由SQL或Java處理，不需懂程式就可使用
- **Kinesis Data Streams**

  real-time service (70ms)

  可用來處發 lambda

  每個Shard 可吃1MB，預設500個，最高無上限

  有KPL、KCL、SDK API 三種套件可

  KPL和KCL比較高階，用法簡單，但若為了講求低延遲可選擇SDK API

  題目 : 若有五個source 每秒都會傳資料給Kinesis，每個資料小於100KB，那請問需要幾個shard?

  答案 : 一個，總流通量=5*100=500kb/s，一個shard有最大上限1MB/s 就已經足夠。
- **Kinesis Firehose**

  將資料轉到S3, Redshift, ElasticSearch, Splunk 的服務 (像Data stream就沒辦法直接做)

  Scaling 由 AWS 自己控管

  不算 real time (有60 秒延遲)

  可觸發lambda

  有convert json to parquet或orc功能，但無法csv 轉 json 這需要靠lambda

  https://docs.aws.amazon.com/zh_tw/firehose/latest/dev/record-format-conversion.html
- **Kinesis Data Analytics**

  Realtime

  SQL, Java Developer friendly

  Serverless
- 資料分佈

  Normal distribution 常態分佈

  Standard normal distribution，mean=0, std=1

Poisson distribution 跟時間有關

Binomial distribution

Bernoulli distribution : special case of Binomial, n=1

- **Time Series** 時序分析

  1. Level : average value in the series
  2. Trend : increasing or decreasing
  3. Seasonality : repeat short-term cycle
  4. Noise : random variation

- **Athena:**

  Query service, using SQL 也可以做一些ETL

  Serverless

  和 Glue Data Catalog 無縫接合

  Support Amazon ORC and Parquet

  若資料原為json或csv可考慮轉成Parquet 讓Athena更有效率分析

- **Quicksight**

  Serverless

  和 Redshift, S3, Athena, Aurora, RDS 無縫接合

  也有forecast功能，提供時序分析

- **EMR**

  Big data platform for ETL

  Leverage Spark, Hive, Hbase, Flink, Presto

- **EMR Data Storage**

  1.HDFS : distributed, scalable, for Hadoop

  2.EMRFS: EMRFS is an implementation of the Hadoop file system used for reading and writing regular files from Amazon EMR directly to Amazon S3.

- **Apache Spark**

  Batch processing, in-memory caching 一種處理的工具，不會儲存

  Support Java Scala Python

  針對Sagemaker有KMeansSageMakerEstimator, PCASageMakerEstimator, and XGBoostSageMakerEstimator 三種工具可以training

- **Scaling** 正規化

  Normalization :用最大最小到0~1

Standardization :減平均除標準差

當有outlier 建議使用Standardization 來做

- **Imputation**，遇到**NaN**的補救方法

  1.可以用mean 或 median來補，若有outlier建議用median會比mean來得好

  2.直接移除

  3.用監督式學習來補，此種方法通常較為精確，卻也複雜

- **AWS DeepAR**

  Time-series algorithm by sagemaker

  使用RNN

  就算feature有NaN也可以解決

- **Binning** 分桶

  quantile binning(根據數量來切，每個桶子內樣本數差不多)

  interval binning(根據數值來切，每個桶子的範圍相同)

- **Log transform**

  Suit for skewed dataset，數據必須都是正數

  可降低outlier的影響

- **Unbalanced Data** 處理（假設現在正樣本多 負樣本少）

  1. Undersampling 從正樣本中抽取部分來降低正樣本數量

  2. Oversampling 從負樣本中複製樣本來增加數量

  3. Generate synthetic data 用合成方式產生一些負樣本(SMOTE)

  SMOTE 利用knn產生樣本 (Synthetic Minority Oversampling TEchnique)

- **Ridge Regression = L2 Regression**

  error += alpha*slope^2

  Dense Output, 計算有效率

- **Lasso Regression = L1 Regression**

  error += alpha* abs(slope)

  Userful for feature selection

  適合有多個獨立變數是無用時

  Sparse Output, 計算較無效率

- Confusion matrix

  1. Accuracy

  2. F1 Score

  3. Precision (TP / TP+FP)

4. Recall, Sensitivity (TP / TP+FN)

5. Specificity (TN / FP+TN)

- **SageMaker GroundTruth**

  Ground Truth 標注工具

  有三種人員可使用這些工作流程提供標籤：

  1. Amazon Mechanical Turk 工作人員
  2. 自己員工
  3. 第三方廠商

  Ground Truth 也可從這些標籤中學習，並且自動對物件進行標籤操作。

- **Sagemaker Mode**

  Training的input dataset有pipe mode 和 file mode

  Pipe mode 較有效率, file mode 要先下載csv比較沒效率

  不是每個內建演算法都支援pipe mode

- **Sagemaker BYOC**

  可以放置客製化程式碼

  https://docs.aws.amazon.com/sagemaker/latest/dg/build-container-to-train-script-get-started.html

- 訓練有**Amazon SageMaker Python SDK (高階)**或**AWS SDK for Python**（低階）

  SageMaker Python SDK 用 model.fit 訓練

  AWS SDK用CreateTrainingJob 開始訓練

- **Sagemaker inference PipelineModel**

  可在notebook將多個Model合成變成連續推論，此時使用HTTP作為protocol

- 部署方式有兩種：

  SageMaker Hosting Services: endpoint

  Python SDK 用 .deploy()

  AWS SDK用.create_model()

  Sagemaker batch transform：starts an endpoint, generates inferences on the stored dataset, outputs the inference predictions, and then shuts down the endpoint.

  Python SDK 用 .transformer()

  AWS SDK用 .create_transform_job()

- 官方**Sagemaker Pre-build container:**

  Apache MXNet, TensorFlow, PyTorch, and Chainer, Scikit-learn and Spark ML

- 推論

  SageMaker Hosting Services: 用 .predict()

  Sagemaker batch transform：用 .invoke_endpoint()
- **Sagemaker Checkpoint**

  在CreateTrainingJob. 可以設定要存checkpoint，這樣中途訓練炸掉還可以從checkpoint繼續
- **Monitor SageMaker**

  CloudWatch Logs：記錄訓練過程的訊息

  CloudTrail ：紀錄API call 、IP address

  CloudWatch Events：紀錄SageMaker training, hyperparameter tuning, or batch transform job的狀態改變
- **Sagemaker Notebook:**

  notebook其實有大小限制，如果不夠可以透過更新升級機器來增加，但是要減少notebook容量只能重新創

  The default is 5 GB. ML storage volumes are encrypted, so SageMaker can't determine the amount of available free space on the volume. Because of this, you can increase the volume size when you update a notebook instance, but you can't decrease the volume size. If you want to decrease the size of the ML storage volume in use, create a new notebook instance with the desired size.

  只有在 /home/ec2-user/SageMaker 底下的資料可以在不同notebook使用
- **Sagemaker Elastic Inference**

  加速推論的加速器，和GPU相比節省很多

  如果使用Multi-Model Endpoints 則無法使用此功能
- **Multi-Model Endpoint Deployments**

  無法用EI，無法用GPU

  BYOC時，設置docker環境變數SAGEMAKER_MULTI_MODEL=true
- **BYOC**規定：

  1. responds to port 8080

  2. socket connection requests within 250 ms.

  3 Model artifacts need to be compressed in tar.gz format.

  https://docs.aws.amazon.com/sagemaker/latest/dg/your-algorithms-inference-code.html
- **SageMaker Debugger**

  https://docs.aws.amazon.com/sagemaker/latest/dg/train-debugger.html

- **Apache Spark with SageMake**
  https://docs.aws.amazon.com/sagemaker/latest/dg/apache-spark.html
- **HyperParameterTuningJobConfig**
  優化超參數的設定
  要指定name、最大值、最小值和scale方式
  Scale方式有 Auto, Linear, Logarithmic (0.0001~1.0), ReverseLogarithmic (0<=x<1.0)
  設置完HyperParameterTuningJobConfig後，用create_hyper_parameter_tuning_job 執行超參數優化
  Best Practices:
  1. 選擇哪些優化參數
  2. 設定範圍，範圍不要太大
  3. 用Logarithmic 來scale
  4. 一次跑一個 tuning job
  5. 能的話開多個 instance
- **Monitor SageMaker with Amazon CloudWatch**
  Cloudwatch 可以知道每分鐘多少 InvokeEndpoint被呼叫等功能
  https://docs.aws.amazon.com/sagemaker/latest/dg/monitoring-cloudwatch.html
  Stdout和stderr也會被記錄在此
- **Log SageMaker API Calls with AWS CloudTrail**
  CloudTrail captures all API calls for SageMaker, 除了 InvokeEndpoint 外
  當在做model tuning jobs 時，如果你stop也會被cloudtrail紀錄
- **Security**
  To encrypt the machine learning (ML) storage volume that is attached to notebooks, processing jobs, training jobs, hyperparameter tuning jobs, batch transform jobs, and endpoints, you can pass a AWS Key Management Service (AWS KMS) key to Amazon SageMaker.
  Enabling inter-container traffic encryption can increase training time
  若要保護 in Transit 的資料
  Console操作的話只需要勾選打開即可
  API的話，設置EnableInterContainerTrafficEncryption=true
- **Sagmemaker 的container 版號**
  Algorithms that are parallelizable can be deployed on multiple compute instances for distributed training. For the Training Image and Inference Image Registry Path column,

use the :1 version tag to ensure that you are using a stable version of the algorithm. You can reliably host a model trained using an image with the :1 tag on an inference image that has the :1 tag. Using the :latest tag in the registry path provides you with the most up-to-date version of the algorithm, but might cause problems with backward compatibility. Avoid using the :latest tag for production purposes.

- **Early Stop怎麼停？**
  https://docs.aws.amazon.com/sagemaker/latest/dg/automatic-model-tuning-early-stopping.html
  If the value of the objective metric for the current training job is worse (higher when minimizing or lower when maximizing the objective metric) than the median value of running averages of the objective metric for previous training jobs up to the same epoch, SageMaker stops the current training job.
- **Network isolation**
  不支援：Chainer, PyTorch, Scikit-learn, SageMaker Reinforcement Learning
  https://docs.aws.amazon.com/sagemaker/latest/dg/mkt-algo-model-internet-free.html
- **Amazon SageMaker Neo**
  將其他框架(TF, MXNet等)部署到ARM, Nvidia 等環境的工具
  Neo currently supports image classification models exported as frozen graphs from TensorFlow, MXNet, or PyTorch, and XGBoost models.:
- **Amazon SageMaker Random Cut Forest**
  非監督式學習，用於異常檢測
  會給每筆資料異常的機率，這些資料通常在3個std外
- **PCA**
  非監督式學習
  有兩個mode:
  1. Regular: 適合 sparse data
  2. Random: 適合多個feature