AWS ML Speciality Cheat Sheet

Last Updated: July 2024

[**Exploratory Data Analysis 2**](#_iv8hqqqisx0m)

[**Data Engineering 2**](#_wfrnugi1gdtg)

[S3 2](#_mgoe7ltwmmwg)

[Kinesis Data Streams (KDS) 2](#_c4leryngl7zr)

[Amazon Data Firehose 2](#_tzl37hyxavye)

[Kinesis Video Streams 3](#_rhznhe3trfjk)

[Kinesis Data Analytics 3](#_ff2wk25qqq3e)

[Glue 3](#_osz457hq5vdc)

[Preprocessing 3](#_fe46bbcvak5)

[Example ETL Pipelines 3](#_f0erdhblcecr)

[**Sagemaker 4**](#_bndlofe9au5u)

[Hosting 4](#_hvsxs9hzg9l9)

[Training 4](#_3wtazx4fm5d1)

[**Hyperparameter Tuning 5**](#_2korkiq6676f)

[Search Algorithms 5](#_wc6qqnod8dvp)

[Code 5](#_hbtg7jugr2me)

[XGBoost 5](#_8tbsmoax4im1)

[**IAM 5**](#_qmq22u64v11d)

[**AWS ML Services 5**](#_facxmy5lst00)

[Rekognition 5](#_ql19iu9me8zv)

[Object Detection 6](#_cni10jlkpj42)

[Image Classification 6](#_nn6jypcjojcs)

[Amazon Translate 6](#_jqphprhw4wal)

[Amazon Comprehend 6](#_7m9zi985446)

[Amazon Lex API 6](#_ghbm0zexdrd8)

[Amazon Textract 6](#_ws09blttj2eg)

[Linear Learner 6](#_dghqy9bbjlz1)

[Random Cut Forest 6](#_nytp0xglhaxq)

[AWS Ground Truth 6](#_k0md33v601vj)

[Privatelink 7](#_k6jnehbpvbiw)

[**ML Python Libraries 7**](#_xevdo8nhx79g)

[Scikit Learn 7](#_g4f7xyv9ijau)

[PySpark 7](#_hyg3voovbrqt)

[XGBoost 7](#_mlb3jgnqo1ze)

[**Algorithms 7**](#_kbcchcsgsxip)

[Bag of Words 7](#_35dwl2asvsil)

[N-GRAM 7](#_vw5391k2egmd)

[TFIDF 8](#_rxm3arf8wyj)

[OSB 8](#_mo14sjdmxyba)

[Quantile Binning Transformation 8](#_yn3bg7ed4ewx)

[**Validation 8**](#_xzagt3u6fevw)

[**Overfitting 8**](#_t9t3z1nrlbz5)

[**Evaluation Metrics 8**](#_ea55q8l3tgl6)

[Precision 9](#_4awvf8w63gug)

[Recall 9](#_y6z6yyt5d8o8)

# Exploratory Data Analysis

* For [Sagemaker built-in XGBoost](https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost.html), Feature importance is defined only by the gbtree booster. Comparison on the gbtree and gblinear algorithms is covered [here](https://www.statworx.com/en/content-hub/blog/xgboost-tree-vs-linear/).
* Note t-SNE for visualisation and dimensionality reduction, alongside PCA. PCA is good for consolidating independent relationships (reducing collinear dependencies).

# Data Engineering

Data Stores, ETL. Note that there are other DB’s like Redshift and Aurora that come up. Briefly read [this](https://docs.aws.amazon.com/whitepapers/latest/aws-overview/database.html).

## S3

* Files can be from 0 bytes to 5TB.
* Unlimited storage.
* Universal namespace (names must be unique globally).

## Kinesis Data Streams (KDS)

Get Streaming data from data producers (IoT, click stream) and leverage shards. Once data is in shard, data consumers can process this data such as EC2, Lambda, Kinesis Data Analytics or EMR. Multiple consumers can consume from a shard.

## Amazon Data Firehose

Streaming, though fully managed (no longer have to deal with shards). Data can be pre-processed using Lambda. Can be used as a way to store directly in S3 or Redshift -> DynamoDB via S3 Events.

* [Encryption (SSE-KMS)](https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingKMSEncryption.html), [Firehose Encryption options](https://docs.aws.amazon.com/firehose/latest/dev/create-configure.html).
* Transforms in [Data Firehose](https://docs.aws.amazon.com/firehose/latest/dev/data-transformation.html) output RecordId, result and data as fields, otherwise something went wrong during transform.

## Kinesis Video Streams

Stream video feeds from devices. Streams data to S3 or other ML services such as Rekognition.

## Kinesis Data Analytics

NOT streams, feeds off Kinesis Data Streams or Firehose.

Can apply functions to streaming data like Random Cut Forest, which other streaming tools cannot do.

## Glue

Fully managed ETL, making it easy to clean, categorize, enrich and move data between data stores.

* Uses Glue crawlers to infer schemas and build the catalog.
* [Adding Classifiers](https://docs.aws.amazon.com/glue/latest/dg/add-classifier.html).
* [Cataloging Tables](http://docs.aws.amazon.com/glue/latest/dg/add-crawler.html).
* Matching similar records in Glue is about finding similar structured data and matching data within that structure. [Lake Formation matching](https://docs.aws.amazon.com/glue/latest/dg/machine-learning.html) on the other hand is for matching data with different schemas.
* The Glue FindMatches accuracy-cost parameter governs the algorithm power vs speed of execution. ‘accuracy’ will enforce the set precision-recall parameter, while ‘lower cost’ improves speed of execution, while sacrificing performance.

## Preprocessing

Use Sagemaker Batch Transform when you need to do the following:

* Preprocess datasets to remove noise or bias that interferes with training or inference from your dataset.
* Get inferences from large datasets.
* Run inference when you don't need a persistent endpoint.
* Associate input records with inferences to help with the interpretation of results.

AWS Glue has a [DymanicFrame](https://docs.aws.amazon.com/glue/latest/dg/aws-glue-api-crawler-pyspark-extensions-dynamic-frame.html) class to handle data where the schema isn’t known in advance.

## Example ETL Pipelines

**IoT Model**

IoT Core -> IoT Analytics. Sensor data ingest to analytics for enrichment and analysis.  
Sagemaker -> IoT Greengrass. Model training and deployment.

**IoT Data Encryption**

Kinesis Data Streams (encryption with KMS) -> Store Data at Rest -> Kinesis Data Streams -> Amazon Data Firehose -> S3.  
Kinesis Data Stream stores data at rest, so you need to encrypt it at this point to keep the data secure.

**On Prem to Cloud Data Engineering + IoT**

| Mainframe | AWS Storage Gateway | Amazon Data Firehose |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | Lambda Integration | CSV | S3 |
|  | IoT Devices | Amazon Data Firehose |  |  |  |

*\*Pipeline flows from left to right*

# Sagemaker

## Hosting

[Model Hosting Options](https://docs.aws.amazon.com/sagemaker/latest/dg/realtime-endpoints-options.html) is all you need.

## Training

* TrainingJobAnalytics used to generate visualisations for training.
* Regex matching logs is essential for metrics to arrive at Cloudwatch metrics dashboard.
* You can shard input S3 data by key, placing a subet of data into each training instance. E.g.

sagemaker.s3\_input(

s3\_data='s3://...',

content\_type='csv',

distribution='ShardedByS3Key'

)



**Metrics**  
MSD - good for visualising clustering during training.

Cross Entropy - loss function commonly used when optimising classification models.

# Hyperparameter Tuning

## Search Algorithms

* Random Search - More training jobs for most models to reach optimum. Can run in parallel.
* Bayesian Search - Less training jobs for most models to reach optimum. Sequential (can’t run in parallel).

## Code

HyperParameterTuningJobConfig -> CreateHyperParameterTuningJob.  
Need to set metric objective name and ranges.

## XGBoost

Hyperparameters and intuitions from the [AWS docs](https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost_hyperparameters.html).

# IAM

Understand [AWS IAM basics](https://aws.amazon.com/iam/). For example, Sagemaker notebook needs access to a certain service and only a data scientist profile is allowed to access the notebook.

# AWS ML Services

AWS services that I’ve seen in an exam.

## Rekognition

TBC

* Detecting and analyzing faces:

<https://docs.aws.amazon.com/rekognition/latest/dg/faces.html>

## Object Detection

Identifies all instances of an object in an image.

## Image Classification

Multi-class classification of objects, usually one per image.

## Amazon Translate

Language translation.

## Amazon Comprehend

Note that there are various lower level API that provide insight into different elements of text analysis as detailed in the [docs](https://docs.aws.amazon.com/comprehend/latest/dg/API_Operations_Amazon_Comprehend.html), e.g. DetectSentiment API.

## Amazon Lex API

Conversation API for voice and text.

## Amazon Textract

Extract text data from scanned documents.

## Linear Learner

Supervised learning algorithms for classification and regression.

## Random Cut Forest

Anomaly detection. Can be used with Kinesis Data Analytics.

## AWS Ground Truth

Can be used with Kinesis Data Analytics.  
[Automatic data labeling](https://docs.aws.amazon.com/sagemaker/latest/dg/sms-automated-labeling.html) is a paid way to keep improving a model that iteratively feeds in new labeled data and retrains.

[Annotation consolidation](https://docs.aws.amazon.com/sagemaker/latest/dg/sms-annotation-consolidation.html) is used to automatically get more accurate labels using image analysis.

## Privatelink

Not explicitly an ML Service, though it [facilitates VPC connectivity to endpoints outside of the VPC](https://aws.amazon.com/privatelink/). You’ll need a VPC endpoint and VPN if you want to connect on-prem to the cloud, then Privatelink facilitates connecting both VPC’s together.

# ML Python Libraries

ML libraries that are fundamental to Data Science with big data.

## Scikit Learn

Understand the feature engineering tools, as well as basic data exploration algorithms.

* OneHotEncoder.
* SimpleImputer.
* OriginalEncoder.
* LabelBinarizer.
* MinMaxScaler.
* StandardScaler.

## PySpark

Understand the various modules in [PySpark](https://spark.apache.org/docs/latest/api/python/index.html#:~:text=PySpark%20is%20the%20Python%20API,for%20interactively%20analyzing%20your%20data.), especially transformation and training.

## XGBoost

[Introductory material](https://www.kaggle.com/alexisbcook/xgboost)

# Algorithms

Algorithms that come up during scenario-based questions. Understanding when to use each is key.

## Bag of Words

Histogram (word count) of words in a document.

## N-GRAM

Good for matching phrases, like in phishing detection.

## TFIDF

Term Frequency - Inverse Document Frequency (TF-IDF) measures how important a term is within a document relative to a collection of documents (i.e., relative to a corpus).

## OSB

Best understood through the use of an [example](https://docs.aws.amazon.com/machine-learning/latest/dg/data-transformations-reference.html#orthogonal-sparse-bigram-osb-transformation).

## Quantile Binning Transformation

The quantile binning processor takes two inputs, a numerical variable and a parameter called bin number, and outputs a categorical variable. The purpose is to discover non-linearity in the variable's distribution by grouping observed values together.

# Validation

**Techniques**

[K-fold cross validation vs Leave-one-out cross validation](https://www.baeldung.com/cs/cross-validation-k-fold-loo)  
Time series cross validation - used for time series analysis

**Scoring**

*F1* - Top performing binary classifier.

*Adjusted Mutual Info Score* - Clustering evaluation. Adjustment to mutual info score to account for mutual information due to large number of groupings, irrespective of whether or not actual information is shared.

*Rand Score* - Clustering evaluation. Similarity between two clusterings.

*Completeness Score* - Clustering evaluation. Calculate completeness of a cluster labeling given a ground truth.

# Overfitting

Lasso Regularisation (L1) - Handles outliers well.

Ridge Regularisation (L2) - Does not handle outliers that well.

Dropout - Dropout with outliers takes more effort than L1.

Early Stopping - Early stopping does not help with outliers.

# Evaluation Metrics

Common classification evaluation metrics that come up in the exam. This comes up a lot. Have a good understanding of which business scenario precision or recall is better for.

## Precision

## Recall