ML Certification Cheat sheet

Below are some of my prep material for this certification. Given the vast nature of the topics, i found the below useful. Hope this helps you as well in your preparation. **"Learn and be curious"**

Contact Meena for questions or additional guidance on this certification!

Basic Statistics

There will be atleast 2 questions on probability distributions. Mostly around identifying the type of distribution and focussed on the below 4.

Binomial Distribution

Binomial distribution is one in which the probability of repeated number of trials are studied

type: Discrete Pre-req parameters - 1 trials - fixed outcomes - 2 mean > variance each trial is independent

Poisson Distribution

Poisson Distribution gives the count of independent events occur randomly with a given period of time

type: Discrete Pre-req Parameters - 1 trials - infinite outcomes - infinite mean = varience events are independent rate at which event occurs is constant

- The number of emergency calls recorded at a hospital in a day.
- The number of thefts reported in an area on a day.
- The number of customers arriving at a salon in an hour.
- The number of suicides reported in a particular city.
- The number of printing errors at each page of the book.

The binomial distribution is one, whose possible number of outcomes are two, i.e. success or failure. On the other hand, there is no limit of possible outcomes in Poisson distribution

Normal / Gaussian Distribution

Normal distribution, chi-square distribution, and F-distribution are the types of continuous random variable

Normal Distribution = Gaussian Distribution

- The mean, median and mode of the distribution coincide.
- The curve of the distribution is bell-shaped and symmetrical about the line x=μ.
- The total area under the curve is 1.
- Exactly half of the values are to the left of the center and the other half to the right

Exponential Distribution

represents time between individual events

1. Length of time beteeen metro arrivals,

2. Length of time between arrivals at a gas station
3. The life of an Air Conditioner

Data Preparation & Feature Engineering

A good understanding on how to handle different scenarios in data preparation and Feature engineering is key. You can expect atleast 10 questions around this.

Best practices

- normalize / scale
- outliers:
- missing data handling: refer to imputation methods
- encoding: managing ordinal (categorical encoding) and nominal values (one hot)
- Binning: when to use quantile vs normal binning
- Handling class imbalance
- Shuffle: when to and when not to (time series dont shuffle)
- split: Types of splits

Class Imbalance

- SMOTE - if GAN is not specified. observations created are similar to the sample set.
- GANs - for more accurate models possible. resembles closely but not similar

Overfitting

Look for both machine learning and neural network techniques.

- Early stopping of epochs
- increase Regularization L1 / Ridge - reduce feature weights / importance L2 / Lasso - removes the feature / weights = 0
- increase dropout
- decrease features
- Reduce model complexity
- Randomize the sampling - after every epoch
- reducing complexity of the architecture
- Strategy for dropping feature - Low variance

Underfitting

Look for both machine learning and neural network techniques.

- add more variables
- more data
- train for longer period of time
- reduce regularization
- Use L2
- Early Convergence - Lower the learning rate

Imputation

- Categorical - Deep Learning
- Numerical - KNN
- if Outliers, use Median
- if no outliers, use mean

Sampling

- Stratified sampling, random
- k fold stratied sampling - imbalance class

Dimensionality Reduction

- PCA
- T-SNE
- K means

Time Series

- Cross Validation vs. K-Fold Cross validation
- Concepts: Observed Numbers Trend Seasonality Noise

Binning

- Quantile binning - unevenly distributed data and preserve the distribution
- Interval binning - loses the distribution visibility

General

large Batch Size - stuck in local minima small batch size - true minima large learning rate - overshoots global minima small learning rate - takes lot of time Reduced Learning rate: ensures the model is not affected by Outliers Large neural network - has vanishing gradient problem, Relu helps in avoiding this

Vanishing Gradient

- from multiplying together many small derivates of the sigmoid activation function in multiple layers
- use Relu

Model Training and Evaluation

Hyper Parameter Tuning

- Bayesian search / Optimization - for better results, jobs run one after another
- Random Search - for quicker results, parallel hyper parameter training jobs

Key Hyper parameters

- Learning Rate (eta) - Too large a learning rate might prevent the weights from approaching the optimal solution. Too small a value results in the algorithm requiring many passes to approach the optimal weights
- Model Size - reduce the model size by using L1 regularization or by specifically restricting the model size by specifying the maximum size

- Regularization (alpha)
- Number of passes - small datasets use higher number of passes, if the learning rate is not too high. large datasets single pass will do data Shuffling

Best practice / cost reduction of Hyper parameter tuning

- Small hyper parameter ranges
- less concurrent runs, Run one run at a time
- Log scales for parameter ranges
- Less number of hyper parameters
- Design distributed training jobs so that the objective metric reported is the one that you want

Confusion Matrix

- False Positive = Type 1 error
- False Negative = Type 2 error
- Recall (TP / (TP+FN)) is important when the cost of a false negative is higher than that of a false positive also called sensitivity, True positive rate medical cases where it doesn't matter whether we raise a false alarm but the actual positive cases should not go undetected!
- Precision is important in music or video recommendation systems, e-commerce websites, etc. Wrong results could lead to customer churn and be harmful to the business TP / (TP + FP)

Classification threshold

- Increases precision
- Reduces recall or stays the same
- A high number of TPs typically results in a high number of FPs and a low number of TNs

Binary Classification

- AUC measures the ability of the model to predict a higher score for positive examples as compared to negative examples get a sense of the prediction accuracy of your model from the AUC metric without picking a threshold 0 is true negative, 1 is true positive

Multi-class Clasification

- The macro average F1 score is the unweighted average of the F1-score over all the classes in the multiclass case

Sagemaker

- Cannot deploy sagemaker to EMR
- Sagemaker hosting services with 2 or more instances automatically does Multi-Az
- Create Model API is for inference Container
- Transfer learning - Keep the initial weights and remove the last layer
- Clustering vs. Classification - Check if the Classes are specified then classification

- Checking new model is stable before deployment - Conduct A/B testing
- Unless you add policy with S3FullAccess permission to the role, it is restricted to buckets with "sagemaker" in the bucket name. Strange but true
- Amazon SageMaker's pre-built containers do not include a container supporting code written in R
- EMR + Sagemaker -> Run your SageMaker Spark application on EMR by submitting your Spark application jar and any additional dependencies your Spark application uses Download the aws-sagemaker-spark-sdk component along with Spark on your EMR cluster Use Spot instances for task nodes only
- Sagemaker training job fails with no logs s3 bucket path issue wrong training image

ML instance recommendataions

- ml.m5.xlarge, ml.m5.4xlarge, and ml.m5.10xlarge
- ml.c5.xlarge, ml.c5.2xlarge, and ml.c5.8xlarge
- ml.p3.xlarge, ml.p3.8xlarge, and ml.p3.16xlarge

Training

- When you create a training job with the API, SageMaker replicates the entire dataset on ML compute instances by default. To make SageMaker replicate a subset of the data on each ML compute instance, you must set the S3DataDistributionType field to ShardedByS3Key
- Inbuilt Algorithm -> does Cloudwatch logs automatically
- Managed Spot training can be enabled to use spot instances
- Custom Algorithm -> needs explicit specification of metrics to Cloudwatch logs When using custom TensorFlow code, the Amazon SageMaker Python SDK supports script mode training scripts. Script mode has the following advantages: Script mode training scripts are more similar to training scripts you write for TensorFlow in general, so it is easier to modify your existing TensorFlow training scripts to work with Amazon SageMaker. Script mode supports both Python 2.7- and Python 3.6-compatible source files. Script mode supports Horovod for distributed training

Distributed training Types:

- Data Parallelism
- Model Parallelism (Horovod or Parameter server)
- MXnet - has distributed training
- tensorflow - use horovod

Inference

- Elastic Inference On Demand GPU Elastic inference can only be attached at instance launch [inference instance] by attaching elastic inference at notebook instance level, you can evaluate inference performance when building the model
- Request Serialized by you Deserialized by algorithm
- Response Serialized by algo Deserialized by you

Inference Pipelines

- Use Inference pipelines to chain multiple models in production deployment. SageMaker handles invocations as a sequence of HTTP requests
- can go up to 5 containers
- A pipeline model is immutable, but you can update an inference pipeline by deploying a new one using the UpdateEndpoint operation
- define the containers for a pipeline model using the CreateModel operation or from the console. Instead of setting one PrimaryContainer, you use the Containers parameter

Batch Transforms

- Use when no inference end point is required,
- getting inferences for large dataset,
- Associate input records with inference
- Supports json and csv
- split input files into mini-batches, when you create a batch transform job, set the SplitType parameter value to Line
- reduce time taken for processing by leveraging - MaxPayloadInMB, MaxConcurrentTransforms, or BatchStrategy Batch strategy - requires split type Max payload in MB - shd be > than size of a single observation
- SageMaker uses the Amazon S3 Multipart Upload API to upload results from a batch transform job to Amazon S3

Autoscaling

- can be done once model is deployed using the Endpoint runtime configurations
- Cooling period is between
- Scaling-in does not occur when there is no traffic: if a variant’s traffic becomes zero, SageMaker automatic scaling doesn't scale in. This is because SageMaker doesn't emit metrics with a value of zero.

control access to SageMaker notebooks to specific IAM Groups?

- put tags on SageMaker resources
- use ResourceTag conditions in IAM Policies to choose these tags of SageMaker instances

Custom Inference Container requirements?

- Your inference container responds to port 8080, and
- must respond to ping requests in under 2 seconds.
- Model artifacts need to be compressed in tar format, not zip

Deployment best practices

- Deploy in more than 1 instance. this will do multi-az automatically
- End point Configuration -> End Point -> Variants
- Ways of deploying multiple models - Use of single end point vs. multiple end points
- Other deployment strategies - Canary / AB testing ...

Security

Huge topic in the test. Lots of scenarios around the following

Infra security

- Access to Sagemaker notebooks
- Access to data in s3 from Sagemaker notebooks
- Training instances access to the data (inter container isolation...)
- control access to SageMaker notebooks to specific IAM groups attach tags to the group of sagemaker resources to be kept private to specific groups and use resourceTag in IAM policies Use the ResourceTag condition and add a Project tag to each notebook
- IP addresses can access an S3 bucket used to store sensitive model training data Resource-based policy
- Some intra-network data in-transit (inside the service platform) is unencrypted. This includes: Command and control communications between the service control plane and training job instances (not customer data). Communications between nodes in distributed processing jobs (intra-network). Communications between nodes in distributed training jobs (intra-network)

Data Security

- How to secure data using KMS
- Create a customer managed key in KMS and use it when creating your SageMaker Notebook instance
- Ensure the Notebook instance role is associated with the KMS key
- S3 bucket and data has a SSE-KMS key associated with it and specify the same SSE-KMS Key ID when you create the SageMaker notebook instance and training job.
- Create a customer managed key in KMS and use it when creating your SageMaker Notebook instance
- Encryption at Rest: SageMaker uses the AWS Key Management Service (AWS KMS) to encrypt the notebooks and data. SageMaker uses AWS managed customer master keys (CMKs) by default. For more control, you can specify your own customer managed CMKs
- Encryption at Transit: Requests to the SageMaker API and console are made over a secure (SSL) connection. You pass AWS Identity and Access Management roles to SageMaker to provide permissions to access resources on your behalf for training and deployment

Sagemaker Algos

XGBoost

- csv or Libsvm NO proto-buf
- CPU powered, memory optimized, GPU's can be used
- Use built in or open source
- Use Weight index for labels to provide importance for certain rows
- Metrics (validation and test) Regression: mae, mse, rmse, map Classification: auc, f1, mlogloss, logloss
- Hyper Parameters learning objective eval_metric (see above) eta(earning rate), alpha(L1), lambda (L2), gamma (split Loss), num_runs, batch_size

- Booster models gbtree - feature importance enabled for gbtree gblinear dart
- Prevent Overfitting subsample, Max depth and eta

Linear Learner

- text/csv & record-io protobuf (float32 tensors)
- GPU and CPU
- predictor type: regression, binary, multi class
- Loss type: Logistic, softmax - gets probability hinge loss doe not provide probability
- Metrics (validataion and Test)
- Hyper parameters learning rate, batch size, l1...
- Remove outliers
- Mandates Shuffling Normalizing

PCA

- Unsupervised
- 2 modes regular - for sparse and moderate number of observations randomized - for dense data

K-Means

- expects data to be provided in the train channel (recommended S3DataDistributionType=ShardedByS3Key), with an optional test channel (recommended S3DataDistributionType=FullyReplicated)
- Metrics: msd and ssd (distances - mean square and sum of squared)
- Hyper parameter: mini batch size, epochs, extra center factor..

Factorization Machines

- Discrete recommendations
- High dimensional sparse data
- use of CPU recommended for sparse data, use GPU for dense data
- record IO protobuf with float 32 tensors
- Metrics regression (RMSE) classification mode(accuracy, Cross entropy, f_beta)
- Does not support CSV

Random Cut forest

- Unsupervised. but has label_size = 1
- text/csv and record io protobuf
- Anamoly detection
- requires optional test channel
- CPU recommended. no additional benefit with GPU
- train channel only supports S3DataDistributionType=ShardedByS3Key
- Metric: f1 score
- Hyperparameter: num_of_trees, samples per tre

K-Nearest Neighours

- 3 steps: Sampling, Dimensionality reduction and Creating index
- Index creation is key to find the Neighbour: Flat, inverted, product quantization
- Classification: nearest neighbor, Regression: Average
- for batch transform - json lines as inputs
- Metrics Regression: RMSE Classifcation: Accuracy
- Hyper parameters : k and sample size
- The Elbow Method Calculate the Within-Cluster-Sum of Squared Errors (WSS) for different values of k

Object2Vec

- Improve the data embeddings of the high-dimensional objects: identify duplicate support tickets or find the correct routing based on similarity of text in the tickets
- identifying relationship between tweets

Seq to Seq

- machine transaltion, speech to text, text summarization
- Convert text from one language to other: Spanish to English
- Summarize a long text corpus: an abstract for a research paper
- Convert audio files to text: transcribe call center conversations for further analysis
- data type: recordio proto buf with integers
- 3 channels: train, validation and vocab
- Metrics: accuracy, perplexity and bleu

IP insights

- Unsupervised
- more of anamoly detection
- training in GPU, inference in CPU
- Metric: validation:discriminator_auc
- No support for record-io

Image classification

- uses resnet
- supports jpg, png and mxnet record io
- file and pipe mode
- S3DataDistributionType of the S3DataSource to FullyReplicated (full data set copied)

Object Detection

- uses resnet and vgg
- supports jpg, png and mxnet record io
- file and pipe mode
- augmented manifest image format

Semantic segmentation

- built using the MXNet Gluon framework and the Gluon CV toolkit, and provides you with a choice of three build-in algorithms to train a deep neural network. You can use the Fully-Convolutional Network (FCN) algorithm , Pyramid Scene Parsing (PSP) algorithm, or DeepLabV3
- Encoder / decoder
- train, train_annotation, validation, and validation_annotation

Blazing text

- Assign pre-defined categories to documents in a corpus: categorize books in a library into academic disciplines
- Not parallelizable
- Mode: Word2Vec: order does not matter, skip gram and CBOW

JSON Lines - Blazing text and Deep AR

Transcribe - audio to text LEX - identified entities etc..if only chat bots with no voice, use LEX Polly - Conversation, text to voice Rekognition - won't know about your company logo, nor will Object Detection until you have trained it first Fraud detector - Online fraud insights

Forecasting Models

CNN QR - only forecast model that accepts related time series data without future values

DeepAR+ - best with large datasets containing hundreds of feature time series. The algorithm accepts forward-looking related time series and item metadata

Prophet - works best with time series with strong seasonal effects and several seasons of historical data

NPTS - is especially useful when working with sparse or intermittent time series. Forecast provides four algorithm variants: Standard NPTS, Seasonal NPTS, Climatological Forecaster, and Seasonal Climatological Forecast

Arima - especially useful for simple datasets with under 100 time series

Ground Truth

Annotation Consolidation - automated way to manage mislabels by workers image labelling - Bounding box and semantic segmentation Text labelling - Named entity recognition

tf-idf

- correlate data from one document to another

Visualization

Correlation - Heatmap Density of values - Heatmap Distribution 1 variable - Histogram, Box plot 2 variables - scatter Relationship 2 variable - Scatter 3 variables - Bubble chart Comparision Single value & Comparision - Bar Chart Data changes over time (3 or more variables) - Line Plot Composition - Pie, Stacked bar, Stacked Chart

Related AWS Services

Kinesis data streams

Need for encrypting IoT data

- Stream data in thro Kinesis Data Streams.
- KInesis data streams encrypt the data
- send the data to Kinesis Firehouse
- Firehose stores the encrypted data in S2
- used by kinesis analytics

Shard Record

- partition key [id]
- sequence [seq number]
- data

Shard

- 1 MB write, 2 MB read per second
- 1000 records per second
- totally of 500 shards (soft limit)
- 24hrs storage, extend to 7 days

Data Ingestion

- Kinesis API [put, get] is synchronous.
  Less set up time [any sdk], no retry, no delays in processing
- KPL: Producer Library is asynchronous, requires java app set up in EC2. takes time aggregate the data streams retry mechanism Multiple input sources - Use KPL running on EC2
- Kinesis agent: Single input

Consumption options

- Kinesis firehouse
- Kinesis data analytics
- Kinesis client library
- Lambda by polling mechanism
- Cannot write directly in to S3.. needs Firehose

Kinesis Firehose / Delivery streams

Writes in to S3, Redshift, DB, Lambda etc... Delivery stream stream name record [with the data] No sharding concept here basic transformation - like compress, filter, convert changing the recordformat

Kinesis Data Analytics

- have some ML models like Random Cut Forest and HOTSPOT
- transformation natively using SQL & Java (flink)

IOT related

- IOT Analytics does not do encryption.

Glue

Glue ETL — FindMatchesML

- Use when the data is being transformed
- model for deduplication
- UTF-8 without BOM
- Number of columns < 100

Custom Library (python): Upload the custom library as a .zip archive onto S3. Before your customers create an ETL job, include the S3 link as a script library and job parameter

Glue Crawler

invokes custom classifiers first, Depending on the results that are returned from custom classifiers, AWS Glue might also invoke built-in classifiers. If a classifier returns certainty=1.0 during processing, it indicates that it's 100 percent certain that it can create the correct schema. AWS Glue then uses the output of that classifier.

If no classifier returns certainty=1.0, AWS Glue uses the output of the classifier that has the highest certainty. If no classifier returns a certainty greater than 0.0, AWS Glue returns the default classification string of UNKNOWN.

Lambda

Defulat lambda timeout is 3 seconds. this needs to be increased default memory setting is 3 MB

S3-> Glue Crawler -> Athena -> Quick sight Redshitt cluster -> Create tables -> Use copy command (for all data files) from S3 -> trouble shoot