**ML Notes: Elements of Data Science**

**Why Machine Learning?**

**A. Difficulty in writing some programs**

Too Complex (facial recognition)

Too much data (stock market predictions)

Information available only dynamically (recommendation system)

**B. Use of data for improvement**

Humans are use to improving based on experience (data)

**C. A lot of data is available**

Product recommendations

Fraud detection

Facial recognition

Language understanding

**Types of Machine Learning**

Supervised Learning: Models learn from training data that has been labeled. Two subtypes. Regression (Numeric) and Classification (categorical) based on Target Type.

Unsupervised Learning: Models learn from test data that has not been labeled.

Reinforcement Learning: Models learn by taking actions that can earn rewards.

**ML Workflow**

Problem Formulation -> Data Collection -> EDA -> Data Preprocessing -> Feature Engg. -> Model Training -> Model Evaluation -> Model Tuning -> Model Debugging -> Productionization

**Key Issues in ML**

Data Quality/Model Quality

**Data Quality**

Consistency/ Accuracy / Noisy / Missing / Outliers / Bias / Variance

**Model Quality**

Underfitting vs Overfitting

Overfitting: Failure to generalize, too flexible, memorise the data. High variance (small change in training data leads to big change in results).

Underfitting: Failure to capture important patterns. Too simple. Not flexible. High bias ( lack of fit in certain regions).

**Supervised methods: Linear Regression**

Linear Regression for numeric target

Logistics regression for categorical target

Univariate Linear regression (one feature and one target), two parameters. W0 (intercept) and w1 (slope). Minimize SSE. Sum of squared errors. Assume error is gaussian distribution

Multi-variate Linear regression. Many features (>=2). One response. y=w0x0+ w1x1+ w2x2 etc.

**sklearn.linear\_model.LinearRegression**

**Logistics regression (Fraud transaction)**

1 or 0, yes or no. T or F.

Estimates the probability of the input belonging to one of the two classes: Positive and Negative.

**sklearn.linear\_model.LogisticRegression**

Sigmoid curve = 1/(1+e(-z)). Response between 0 and 1 always.

Vulnerability to outliers in the data.

Linearly Separable vs Non linearly separable.

Logistic regression is good for linearly separable.

**Problem Formulation**

Define appropriate business metrics. Simple Algo, Lot of data. Characterize Ml Problem. Establish ML Goal.

**Data Collection**

Sources: Logs/ Databases/ Websites (Crawling and Scraping) / Data providers.

Open Data on AWS

Sampling (subset of instances for training and testing)

Labelling (Gold standard answer for supervised learning)

Good samples should be unbiased. Random sampling.

Stratified Sampling (random sampling to each subgroup separately).

Issues with Sampling: Bias/ Seasonality (time of year) / Trends (patterns shift over time)

Leakage: (Train/test bleed: overlap of data)

**Labelling**

Labels are not always available directly. Sometimes labels can be inferred (say click-stream data). Human labels are preferable. Labelling Tools: Excel sheet. **Amazon Mechanical Turks**, Custom tools.

Sampling and Treatment Assignment.

Causation and Correlation.

**EDA. (Exploratory Data Analysis)**

Domain Knowledge: Amazon ML Solutions Lab. (Brainstorming, Custom Modeling, Training)

Data Schema: Combine data to S3 and use Sagemaker. No of Columns/ Type. Merge/Joins. Use pandas DataFrame merge/join . df.merge

**Data Statistics:** Overall stats. N. of rows/cols. **df.info, df.head, df.describe.**

**Attribute stats.** (univariate) Numerics attributes (mean/variance, **df.describe()**, **df.hist(),**

Stats for categorical attr. Histogram, most frequent, number of unique values. **df[attr1].value\_counts()** or seaborn’s distplot. Density Plot (**df[attr1].plot.kde)**, Histogram (**df[attr1].plot.hist),** Boxplot (**df.boxplot([attr1])**, scatterplot (df.plot.scatter(x,y)), Scatter Matrix. pd.scatter\_matrix

Target Attrs. np.bincount(y)

Multivariate stats: correlation, Contingency Tables/Cross Tabulation.

Correlations:Correlation Matrix HeatMap, Pearson Correlation, Scatterplot Matrix. Correlation from -1 to 1. Zero means no linear relationship. But it can be a non-linear relationship. **Seaborn.heatmap() or np.corrcoef().**

Covariance.

**EDA: Data Issues**

Messy (data on different scale, mixed data type), Noisy, Biased, Imbalanced (sample bias, Outliers), Correlated (Collinearity Problem)

Solution: Impute missing data

**Data Preprocessing: Encoding Categorical Variables**

Categorical ( also called discrete, finite set of values)

Binary (only 2 outcome)

Pandas dtype=”category”

Ordinal: Categories are ordered. E.g. Size.L>M>S

Nominal: Categories are unordered. E.g. Colour.

Algos require numeric value. Encoding to convert char to numeric.(Y,N to 0,1)

Encoding Ordinals. Use map function. **map.mapping()**.

For Target variable use **sklearn.preprocessing.LabelEncoder()**

Encoding Nominals: Encoding nominals with integers is wrong. Because ordering and size is meaningless. Use one hot encoding. **sklearn.preprocessing.OneHotEncoder().**

**pandas.get\_dummies(df)**

Define a hierarchy structure, group the levels by similarity to reduce the overall number of groups.

**Data Preprocessing: Handling Missing Values**

Process, drop or impute.

Use Pandas to check. df.is\_null.sum()

**Drop**: df.dropna(), drop row or column.

Risk of dropping rows: losing too much data, overfitting, underfitting(column drop), may bias sample. Before dropping, ask why data is missing? If it's missing randomly, impute it.

**Impute**: replace a missing value with estimated value. **sklearn.preprocessing.Imputer().**

Imputation Algorithm. Mean.

**sklearn.impute.MICEImputer**

Python **fancyimpute** package (KNNImpute, MICE, SoftImpute)

**Feature Engineering**

Build new features from existing features. Use intuition. Generate new features first, transformation of attributes. X square. x\* y,

**sklearn.feature\_extraction**

Filtering (applies to image and voice analysis a lot) filters frequencies (voice), channel (colour).

Scaling applies to numeric features. Rescale it to bring it on the same scales. Not all algos are sensitive to scale. Like Decision trees and random forests are not affected.

**Scaling methods.**

Mean/variance standardization. Remove mean (0). **sklearn.preprocessing.StandardScaler()**

MinMax Scaling. min=0, max=1.**sklearn.preprocessing.MinMaxScaler()**

MaxAbs Scaler. Removes abs of max. sklearn.preprocessing.MaxAbsScaler()

Robust Scaler.Q75-Q25. sklearn.preprocessing.RobustScaler()

Normalization is for rows. For multiple numeric features in a row. Used in text analysis.

sklearn.preprocessing.Normalizer()

**Transformation.**

Polynomial Transformation for numerical columns. X square, x cube etc.

Linear fit: y=a\*x+b

Quadratic Fit. y = a2 \* x2 + a1 \* x + c

sklearn.preprocessing.PolynomialFeatures. Use this to create higher order features.

It might lead to overfitting.

Other non linear transformations: **Log, Sigmoid**.

**Radial Basis Function** (RBF) is used in SVM (Support Vector Machine).

Gaussian RBF is the most common RBF used.

RBF is also used in Radial Basis Neural networks (RBNN).

**Feature Engineering: Text based Features.**

Bag of words model. Does not keep sequence. Vector of numbers.

Count Vectorizer. sklearn.feature\_extraction.text.CountVectorizer()

TfidVectorizer. sklearn.feature\_extraction.text.TfidVectorizer()

sklearn.feature\_extraction.text.HashingVectorizer()

**Supervised Learning**

Perceptron

Neural network Architecture.

CNN (Convolutional). Convolution layer and Pooling Layers.

RNN. (Recurrent) Result of Output goes to another input. Not one dir.

**Supervised Learning: K- nearest Neighbours (KNN)**.

Define a distance metric.

Euclidean distance.

Manhattan distance.

Any vector norm.

Assign the class label by majority vote. K = (square root of N)/2. large K more global behaviour.

Its non parametric. Requires keeping the original data set. All points of training data. Space complexity with growth of training data.

sklearn.neighbors.KNeighborsClassifier.

Suffers from the curse of Dimensionality.

**Supervised Learning: Linear and Non Linear Support Vector Machines**

Boundary Data point

Optimal Hyperplane to maximize margin. Popular in research. Support vector (boundary points)

**sklearn.svm.SVC**

Non Linear. Popular in research again. Needs to rem all boundary points. Computation sensitive. **sklearn.svm.SVC**

**Supervised Learning: Decision tree and Random Forests.**

Entropy and Ensemble

Entropy: Relative measure of disorder in the data source.

Nodes are split based on large information gain (IG).

One metric to quantify IG is to compare entropy before and after splitting.

Train by maximizing IG to choose the split

Less need for feature transformation

Susceptible to overfitting. Must prune the tree to avoid overfitting.

**sklearn.tree.DecisionTreeClassifier.**

Use Ensemble method to avoid overfitting. Learn multiple models (Tree) and combine the result. It's called Random Forest. Set of decision trees. Prediction: Average output probabilities. Reduce variance through averaging. No need to prune. More expensive to train and run.

**sklearn.ensemble.RandomForestClassifier**.

**Model Training: Validation Set**

Model Training and Tuning

Split the training data in two parts. Training set and validation set. Use a validation set as a test set during debugging and tuning.

Cross validation.

**Model Tuning: Bias Variance Trade off.**

Total error: Bias square + Variance + Irreducible error

Good model = Low bias and low variance.

Bias = Estimated - Actual

**Bias**: an error from flawed assumptions in the algorithm. High bias can cause an algorithm to miss important relationships between features and target outputs resulting in **underfitting**. Simple models lead to high bias.

Solution: Try new features. Decrease the degree of regularization.

**Variance**: an error from sensitivity to small variations in the training data. High variance can cause an algorithm to model random noise in the training set, resulting in **overfitting**. Complex models lead to high variance.

Solution: Increase training data. Decrease the number of features. Increase the regularization.

Use the learning curve to evaluate the model.

**Sklearn.learning\_curve.learning\_curve**

Training accuracy and validation accuracy plot.

**Model Debugging: Error Analysis**

Residual analysis: difference between actual and expected

For classification, it is easy to check errors against actual. .

Common patterns: Data error/ Labeling errors / under over represented sub classes data / discriminating information is not captured in features.

**Model Tuning: Regularization**

Overfitting errors can be reduced using regularization - a technique that helps evenly distribute weights among features.

Adding penalty score for complexity to cost function

Regularization in Linear models: by penalizing large weights. Two types. L1 (Lasso, abs weight) and L2 (Ridge, weight square) . L2 popular. Scaling is important. In L1, weights can be reduced to zero (feature drop). Larger the alpha, larger the regularization.

sklearn. linear\_model.Ridge

sklearn. linear\_model.Lasso

sklearn. linear\_model.ElasicNet (Both L1 and L2)

Strength of regularization c = 1/alpha.

Logistics regularization.

**Model Tuning: Hyperparameter Tuning**

Model Tuning choices

Neural network: learning rate/nodes/layer.

SVM: Optimal C parameter.

Decision Tree: Min number of samples at the leaf node.

Logistics. Regression: Optimal regularization parameter.

We use one of the configurations. It’s not derived from training data.

Technique: grid Search (for two parameters), Random search.

**sklearn.grid\_search.GridSearchCV**. GridSearch Cross Validation. Compute intensive.

**RandomizedSearchCV**

**Model Tuning**

Training data tuning: small. Collect or add sample data. Create synthetic data.

Feature set tuning: Add features, different transformations of same feature. Apply dimensionality reduction (to avoid overfitting) of weak features.

**Model Tuning: Feature Extraction**

reduce s computation.

Maps data into smaller feature space.

Techniques:

Principal component analysis (PCA): unsupervised approach. **sklearn.decomposition.PCA**

Linear Discriminant Analysis (LDA): supervised linear approach for feature extraction. **sklearn.discrimant\_analysis.LinearDiscriminantAnalysis.** Can reduce to atmost #classes-1

Kernel versions for non linear data. Kernel PCA

**Model Tuning: Bagging/Boosting**

Feature selection and extraction is a manual process. Bagging and Boosting are automatic/semi-automatic ways.

Bagging (Bootstrap aggregating): generate a group of weak learners. Used for high variance and low bias. Reduces variance and keeps the same bias. Uses a subset of columns and rows. Combine the result

**sklearn.ensemble.BaggingClassifier**

**sklearn.ensemble.BaggingRegressor**

Boosting. Assign strengths to weak learners. Iteratively train learners. For high bias models.

Training a sequence of samples.

**sklearn.ensemble.AdaBoostClassifier**

**sklearn.ensemble.AdaBoostRegressor**

**sklearn.ensemble.GradientBoostingClassifier.**

**XGBoostLibrary**. Very common python library.

**Sample Data**

Sklearn.datasets.load\_breast\_cancer

**Types of Production environments**

Batch Predictions

Online Predictions

Online Training

**Model Evaluation Metrics**

Confusion Matrix: sklearn.metrics.confusion\_matrix

Accuracy Score:

Precision: Proportion of positive predictions that are actually correct

Recall: Proportion of positive sets that are positive

F1-Score: Combination (harmonic mean) of Precision and Recall

**Cross Validation:** Train and evaluate on distinct data sets

**Sklearn.model\_selection.train\_test\_split**

**K-Fold Cross Validation:** for small sets

Partition data into K folds

K between 5 and 10 to start with

Leave one out cross validation: for very small data sets.

Stratified K-fold cross validation: for advance data or seasonal data.

**Metrics for Linear Regression**

Mean squared error (MSE): **sklearn.metrics.mean\_squared\_error**: Average squared over entire dataset

R square error: between 0 and 1. Fraction of variance accounted for by the model. Higher the square. Better the model .Adjusted R square. **sklearn.metrics.r2\_score**

Confidence interval: Quantifies margin of error between sample metric and true metric due to sampling randomness. Typical CI used are 90,95,99. Z-score. Quantifies how much the value is above or below the mean in terms of std deviation.

Confusion matrix does not make sense for regression.

**Using ML models in production: Storage**

Model and pipeline persistence

Vendor independent xml based language for storing ML models(PMML). Supported by KNIME, sklearn and spark mllib.

A/B testing or shadow testing.

Book: Rules of Machine Learning: Best practices for ML Engineering. Martin Zinkevich

**Using ML models in production: Monitoring and maintenance**

Availability of data may change over time

Software version may change. Python 2 to python 3.

Spike due to black friday

Change in business goal.

**Using ML models in production: Using AWS**

**Sagemaker**: Build, train, deploy ML models at scale

Build: Pre-built notebooks and built in high performance algorithms.

Train: One click training, hyperparameter optimization

Deploy: one - click deployment fully managed with auto-scaling

**Rekognition** Image/Video: Deep learning models.

**Lex**: Build Chatbots to engage customers.Deep learning like NLU (Natural Language understanding) or ASR (Automatic speech recognition)

**Polly**: Text to Speech. Two dozen languages. Wide variety of natural sounding voices.

**Comprehend**: NLP (natural Language Processing) service, Large text understanding. Language, key words. positive/negative

**Translate**: Machine translation service: Localize content for international users.

**Transcribe**: Voice to Text.

**DeepLens**: hardware device. HD video camera. Integrates with sagemaker and lambda.

Glue: ETL Jobs.

DSSTNE (Deep Scalable Sparse Tensor Network Engine):Neural Network Engine

**Common mistakes**

----------------------------

**Machine Learning Terminology and Processes**

Deep Learning AMI comes with these frameworks.

Apache MXNet for Product Recommendations

Caffe and Caffe2 for computer vision projects.

TensorFLow.

Managed Learning platform. EMR. SparkML.

API based tools. Rekognition, Polly, Lex (Conversation)

Storage: S3.

Data Analytics. Athena, Redshift. EMR.

**Cross Validation Techniques**

Split training data in validation set too.

Leave one out. LOOC. Leave one data out.

KFold. Randomly split data in k folds.

**Evaluation Metrics.**

**Only for regression target values.**

MAPE.Mean Absolute Percent Error.

RMSE. Root Mean Square error.

R square = 1-(Model Mean Squared Error/Variance)

**Evaluation Metrics.**

**Only for Classification**

Confusion Matrix

ROC Curve- for Binary classification.

Precision-Recall

Non Linear feature transformation. Binning. Quadratic, Log, Polynomial.

**Domain specific transformations.**

Text Features.

Stop words removal

Lower casing, punctuation removal.

Cutting off very high/low percentiles.

TF-IFD normalization.

Web-page features. Multiple fields of text: URL. anchor in out. Title.

**Parameters Tuning**

Loss Function:

Square: regression, classification.

Hinge: classification only. More robust to outliers.

Logistics: classification only. Better for skewed class distributions.

Learning parameters. Decay rate.

Regularization: prevent overfitting by constraining weights to be small.

-----

**Types of Machine Learning Solutions.**

Computer Vision

NLP Natural Language Processing

Chat bots.

**Seeing Clearly.**

Computer Vision Theory: Acquire Images, Process, Analyze, Understand.

Amazon Rekognition. Image and Video.

Object and Activity detection, Pathing (Where is this object, direction of object), Celebrity recognition (Amazon database of celebrities, known faces), Face recognition (for unknown faces), Unsafe video detection (terms and conditions if unknowns users upload video on your app, violent, bloody images. System can discard or flag off. ). Facial analysis (happy or sad ).

Image Classification.

Localization Technique. App: You only look once. (YOLO).

Semantic Segmentation. Composition, Dimension.

CV Theory Solutions:

AWS DeepLens.

Satellite Image Classification with SageMaker.

Automated Video Editing.

**Speaking of:**

Machine Translation and NLP.

Real Time translations into many languages.

Cross -lingual communications

Chat

Ticketing

Helpdesk

Email.

NLP as a service. 100+ languages. Neural network foundation.

**NLP Solutions**

Comprehend, Transcribe, Translate, Understanding Neural networks, Neural Machine Translation with Sockeye.

**Your call is important to us:**

Communicating with chatbots

Polly, Lex,

Building a dynamic **conversational** bot:

Intents and AWS Lambda

Build and test with Lex

Taking the bot to production

Gather insights from customer interactions

Automate deployments.

------------------------------

**CRISP DM on AWS**

Cross Industry Standard Process - Data Mining

Phases of CRISP DM. 6 phases.

1. Business Understanding: Business Objective: Suitable/Not suitable for ML
2. Data Understanding: Data Collection, Data Properties, Exploration, Quality, AWS Services: Glue ETL, Athena, Quicksight, ggplot, matplotlib.
3. Data Preparation: Select dataset, feature engg.
4. Modeling: Select and Create Model, Model Test plan , Tune Model

AWS: EMR+ Spark, EC2+ DAMI, RStudio for preparation and modeling.

1. Evaluation: Model evaluation, Business objective evaluation, Final decision.
2. Deployment: Maintenance and monitoring. EC2, ECS, Batch, Lambda,

Deployment tools, AWS CodeDeploy, AWS OpsWorks, Elastic BeanStalk

Infrastructure: CloudFormation.

Preparing the data

1. Cleaning: missing data handling, corrupt data, noise.
2. Transforming. Derive additional attributes. Attribute transformation.
3. Merging: merge data using join or concatenation.
4. Formatting: randoming shuffle data.

Neural network framework using mxnet, tensorflow.

**MXNet deployment**

**Training**

mx.sym.FullyConnected(data, num\_hidden)

#softmaxloss

Lenet = mx.sym.SoftmaxOutput(data, name=”softmax”)

#create a trainable module on GPU 0

lenet\_model=mx.mod.Module(symbol=lenet, context=mx.gpu())

Lenet\_model.fit

**Evaluation**

lenet.model.score()

**Store Deployment model**

Lenet.save - saves network, also called symbol.

Lenet\_model.save\_params - save model parameters.

------------------------------

# Developing Machine Learning Applications

Sagemaker: Build, train and deploy ML.

Provides.

1. Hosted notebook instances. To build, explore ML data/model.
2. Jobs: compute for training jobs to train models.
3. Endpoints: Deploy model on http endpoints.

Sagemaker Neo

Neo Compilation Decouples framework from Platform.

Provides model compatibility. Framework Agnostic ML model

Neo Compiler Container

Shared Object Library.

Key benefits of Neo:

Popular deep learning and decision tree models.

MXNet, TensorFlow, PyTorch, XGBoost.

2xperformance speedup and 100x memory footprint reduction.

Various EC2 instances and edge devices.

**ML Algos explained. Sagemaker**

Supervised, UnSupervised,Reinforcement, Deep Learning

Liner Supervised algo: Linear Learner (Linear + Logistics Regression), Support Vector Machine (SVM), Perceptron

Non Linear Supervised Algo: Decision Tree (Random Forest, XGBoost (Gradient Boosted Trees), Factorization Machines. Polynomial, Neural Networks.

Unsupervised: Clustering (different algo parameters), Anomaly detection (Random Cut Forest), Topic Modelling (Discover topics, used in Comprehend). .

KMeans Clustering, PCA: Principal Component Analysis. LDA Latent Dirichlet Allocation - also in Comprehend, Anomaly Detection, Hot Spot Detection (in Kinesis Data analytics)

Time Series Forecasting: Supervised Learning.

Deep Learning: Neurons. NLP, Image /speech recognition. Runs on multiple gpus during training. CNN for image recognition/classification.

RNNs. Recurrent. Takes input from previous output. LSTM (Long Short term memory) Have memory, Speech recognition and translation. Sockeye.

Sagemaker:

Image CLassification: ResNet (CNN)

Sequence to Sequence: seq2seq (RNN for text summarizing, translation, TTS).

Neural Topic Modelling (NTM)

DeepAR Forecasting: Time Series Prediction.

**Sagemaker Automatic Model Tuning.**

Algorithms (Built in), Frameworks (mxnet etc), Docker (your own algo in docker container), Tune(wraps up training algo).

Build: Notebook.

Train, Distributed computing. Jobs.

Deploy: endpoint. Invoke for prediction.

**Hyperparameters.**

Neural networks. Learning rate, layers, regularization, drop-out.

Trees: Number, Depth, Boosting step size.

Clustering: Number, Initialization, Pre-processing.

Tuning,

Manual, guess, experience.

Brute Force: Grid, Random, Sobol.

Meta Model. Another model on top of model.

Sagemaker uses Meta model. Gaussian process regression model

Bayesian optimization decides where to search next. Gradient free.

Three types, Continuous, Integer, Multi value.

HyperparameterTuner() inbuilt python sdk for automated model tuning.

Hyperparameter relationship

**Advanced analytics with AWS**

Sagemaker and Spark.

Spark runs locally on sagemaker notebooks

Sagemaker algos are compatible with spark mllib.

Connect notebook to emr spark endpoints using apache livy.

Out of the box classes.

Model definition: SageMakerEstimator, parameters of model,Algo, Input data, instance type.

Training and Inference: SageMakerModel, evaluation.

DataFrame -> Estimator -> Model

(InputData) -> (Algorithm) -> (model Created)

estimator.fit() -> Model created using sagemaker training job.

model.transform(testdata) -> gives prediction.

SageMakerResourceCleanup

ML Pipeline.

Spark algo -> Sagemaker Algo

Example: ML Pipeline with PCA on Spark (It runs on notebook spark cluster) and K-Means on SageMaker (job runs on sagemaker infra).

Pipeline(stages[])

Iterate through models in pipeline to cleanup resources.

**Anomaly detection with AWS**

Random cut forest (Amazon’s own algorithm, improvement of isolation forest) with sagemaker and kinesis data analytics.

Random cut tree. Each tree works on sub-sample. Each element is leaf in tree.

Each tree built on a random sample.

Anomaly Score: Displacement. A point is anomaly if its insertion greatly increases the tree size.

Model endpoints can be called by lambda on s3 event notification on fresh raw data. Lambda can derive anomaly score and if it is beyond threshold can generate cloudwatch alarm.

Other way is EC2 streaming data to Kinesis data stream and data analytics running on top of it.

**Recommendation system using mxnet and gluon.**

**Collaborative filtering** using matrix Factorization.

Collaborative filtering: User based, item based.

User factors, item factors. Multiply together to fill the matrix. User \* item.

Embedding Layer: extracts importance of feature from data.

Step by step coding in Gluon.

Linear.

1. Load data in array iterator: gluon.data.DataLoader(gluon.data.DataSet, ) Data Iterator
2. Define the network: gluon.Block, gluon.nn.Embedding(), implement forward; net.collect\_params()
3. Initialize the network: mx.init.Xavier(magnitude;2.24); net.collect\_params().initialize.
4. Choosing the loss function/ objective function. Loss\_function = gluon.loss.L2Loss();
5. Choosing optimizer: sgd, adam. trainer=gluon.Trainer(,’sgd’,learningrate, wd, momentum);Training data iterator= glouon.data.DataLoader; SparseMatrixDataSet.
6. Training the model. Loop through epoch. mx.autograd.record(), autograd is library for gradient computation. Compute loss using function. Loss = loss\_function(predictions, label)
7. Measuring the performance.
8. Adding non linearity.
9. Measuring the performance.

Matrix factorization works on small data/catalogues

**Content based** recommendation.

**Hybrid model**, combination of collaborative filtering and content based.

**DSSM: Deep Structured Semantic Models.**

Matrix Factorization

Neural networks: Good at picking up semantic intent, great at image captioning.

Out of network is a tensor.

Uses out of several networks as Embedding Layer for an enriched recommendation system.

Gluon is pytorch like imperative API for MXNet.

Sagemaker built in algo: Factorization Machine Algo implemented using MXNet.

Import sagemaker.

Fm = sagemaker.estimator.Estimator(containers, )

Fm.set\_hyperparameters

fm.fit(train, test)

Fm\_predictor = fm.deploy(instancetype, instance\_count) -creates ECS env and endpoint.

fm\_predictor.predict(data)

**Development and Deployment (without sagemaker built in algo)**

Loss Function

Scalability, offline batch computation and saving results, caching, can help.

**Logging and Measurement**

Display Bias.

May not work on historical data but work on future data.

Online comparisons are critical.

Metrics:

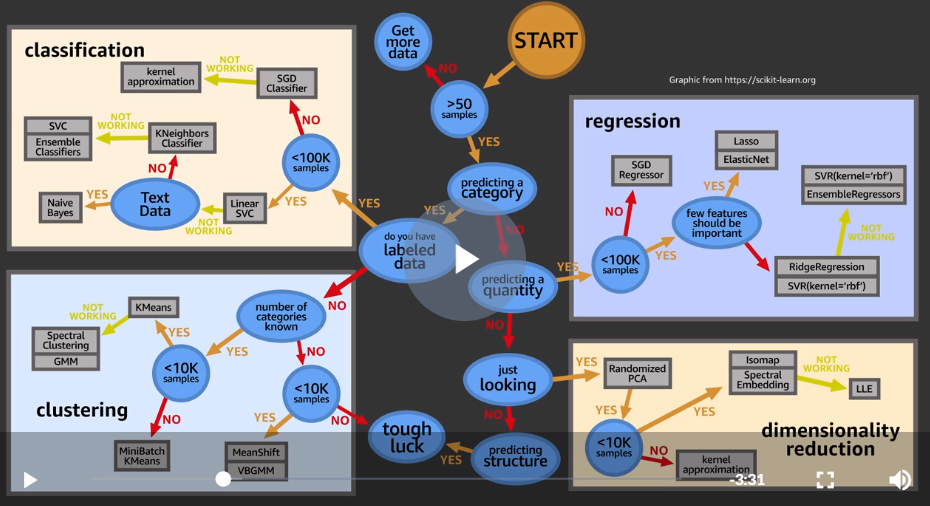
Recall, Precision.

Forecast customer score.

---------------

**ML Exam Basics**

**ML for Business Challenges.**

****

**Image Classification:**

Algorithms for labelling

Naive Bayes

Decision Tree

Logistics Regression

Support Vector Machine

Deep Neural Network.

**Reinforcement Learning: Robot Programming**

Learns from feedback.

State, current env.

Action

Future data

Difference with supervise learning:

No presentation of input/ output pairs.

Rewards based.

Agent needs useful experiences.

Evaluation of the system is often concurrent with learning. (Online learning)

**Deep Learning (subset of ML which itself is subset of AI)**

Backpropagation Algo in 1986.

LSTM and CNN in 1998

GPU training in 2007 started GPU era.

DL: Learn tasks using ANN. Uses many layers of non linear processing units. Algo can be supervised on unsupervised. High level features derived from low level features. Works on raw features. Output of one layer is input to next layer.

Each node has weighted input, sum (bias) and activation function.

Types of NN. Feedforward (text, speech) and Recurrent NN (LSTM, speech recognition)

Usecases: Text , Speech, Image, Predictive and Time series analysis.

Challenge for Neural networks: SCALE

AlexNet: CNN (Year 2012), Number of layers 8,

ResNet-152: CNN using residual learning, 2015, Layers 152, Millions of neurons and parameters.

DL-based AWS services: Lex (Conversational), Polly (Speech), Rekognition (Image).

DAMI:MXNet, TensorFLow, Microsoft Cognitive Toolkit (CNTK), Caffe, Caffe2 Theano, Torch and Keras. For linux and ubuntu.

**Amazon Polly**

---------------------

Turns text into life like speech.

Based on SSML (XML based mark up language) <speak></speak>

Supports Lexicon. Build dictionary of your own.

**Amazon Lex** ( Used in A**Lex**a)

--------------

Conversational interface for voice/text chat bots.

ASR Speech recognition for speeech to text and NLU

Action/Intents and sample utterance and slots.

Call center bots, Amazon connect -> Lex -> Lambda.

Informational bots. Lex -> Lambda -> DB.

Enterprise Productivity bots: Lex-> API Gateway -> Lambda -> Business application.

Initialization and Validation Lambda

Intent Fulfilment Lambda.

Bot Deployment Considerations

. Lex APi, Runtime API and model building API. two actions (user input to Lex). PostContent, PostText

. Versioning: Supported for Intents, Slots and Bots.

. Deployment: Channel. Facebook, Amazon Connect. Lex -> MobileHub

. Security: Cognito to secure Lex Runtime API.

. Monitoring.

CloudWatch Metrics:

Runtime: RuntimeLambdaErrors, MissedUtteranceCount,

Dimensions grouping: BotName, BotAlias, Operation, InputMode.

Dimensions grouping: BotName, BotVersion, Operation, InputMode.

Missed Utterances

**Seeing Clearly**

**----------------**

**Rekognition**

Searchable image library, S3 -> Lambda -> Detect Object using Rekognition -> Store label in ES -> Property Search

Image Moderation: inappropriate content detection

Sentiment analysis: Analysis faces. Marketing reports.

**DeepLens**

Wireless enabled camera and development platform. Offers latest deep learning technology to develop computer vision applications.Intel atom processor.

DeepLens Ecosystem: Sagemaker, Lambda, GreenGrass.

Lambda sends videostream it to Sagemaker model to process and receives processed project stream.

Supports MXNet, Caffe, TensorlLow.

**Semantic Segmentation**

DeepLearning Problem

CNN is for global image classification.

Localization Techniques tells info about an image. What? Where? How many?

Object detection.

Semantic Segmentation is Fine Localization Technique. (Per pixel classification)

Task, Data, Architecture and potential loss.

Semantic Segmentation vs Instance segmentation

Input Data is image, 3d array, x,y being position and z being pixel intensity

Output data is Label Vector or Mask Array

Architecture: AutoEncoder. U-Net, A fully Convolutional Autoencoder with Skipped connections.

Loss: Average Categorical Cross entropy, Dice coefficient

**Issues in practice:** Highly computational on high resolution images. Down sampling. Resizing input. Cropping, Class Imbalance. Solution: Crop from positive windows.

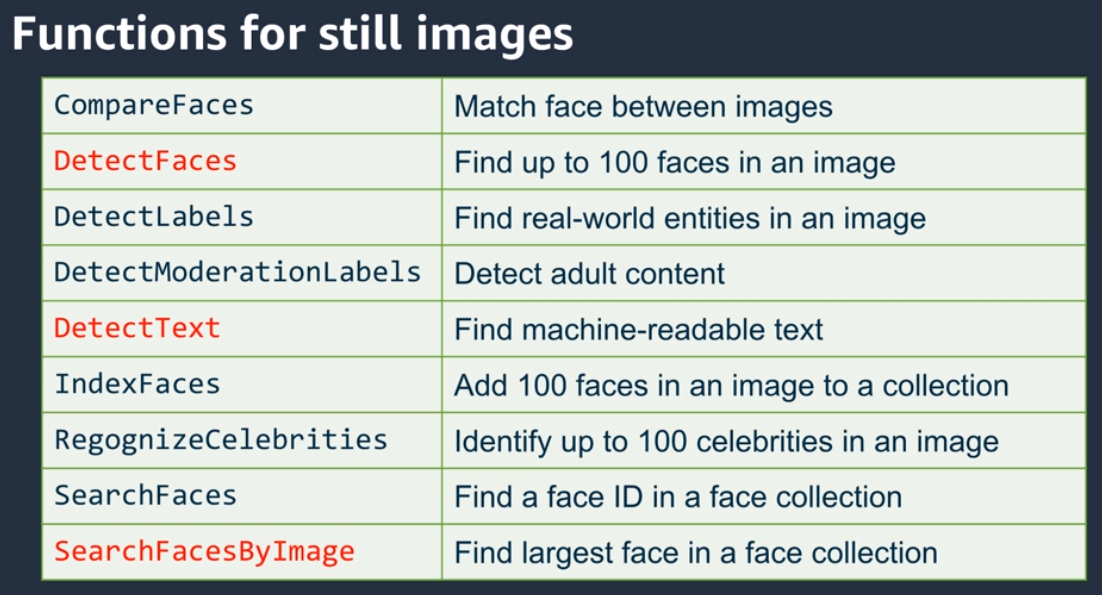
**Rekognition: Automated video editing**

Images, Video, Streaming Video

capture.read(), capture.resize(), rekognition.detect\_faces

Rekognition.search\_faces\_by\_image

detect\_text()



**Video, processed video of given person.**

Create-collection, index-faces (image from s3), start-face-search (returns job id), get-face-search (by job id) , it returns timestamp and faces.

Clip Stitching with Elastic TransEncoder. Takes start time and duration.



**Video Stream**

Send stream to kinesis video stream which sends it to Rekognition.

Live camera -> Kinesis video stream -> Rekognition video. -> Kinesis Stream -> Lambda -> IOT Things

**Rekognition: Building Computer vision based smart applications**

Computer Vision

------------------------------------

Udemy ML A-Z

Regression

Machine Learning Regression models:

1. Simple Linear Regression
2. Multiple Linear Regression
3. Polynomial Regression
4. Support Vector for Regression (SVR)
5. Decision Tree Classification
6. Random Forest Classification

<https://sds-platform-private.s3-us-east-2.amazonaws.com/uploads/P14-Regression-Pros-Cons.pdf>

<https://sds-platform-private.s3-us-east-2.amazonaws.com/uploads/P14-Regularization.pdf>

Machine Learning Classification models:

Logistic Regression

K-Nearest Neighbors (K-NN)

Support Vector Machine (SVM)

Kernel SVM

Naive Bayes

Decision Tree Classification

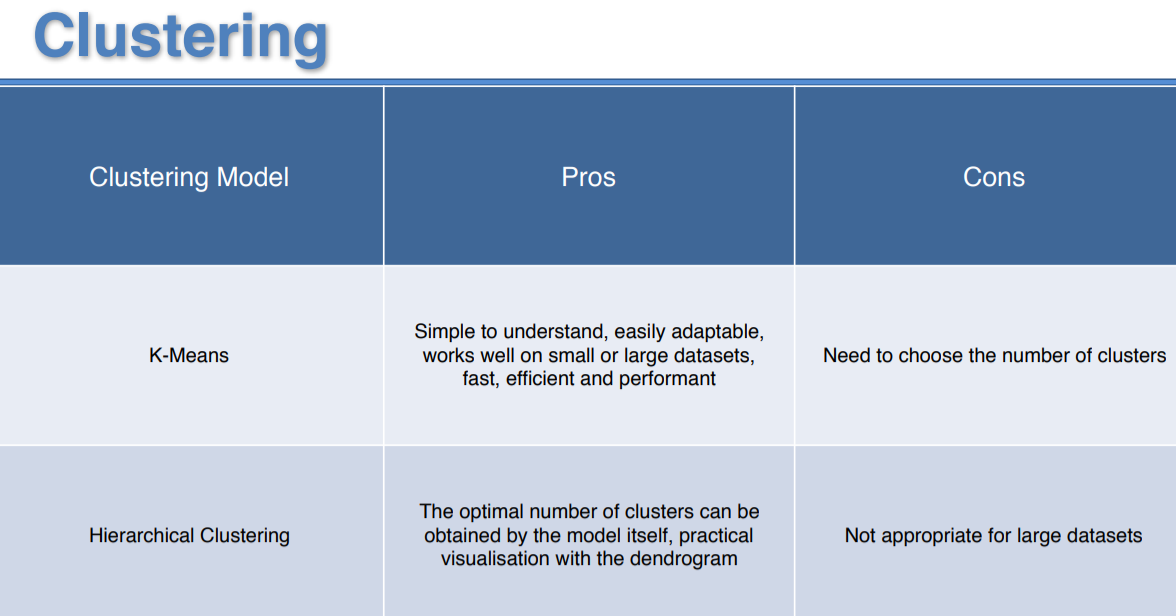
Random Forest Classification

Classification models include linear models like Logistic Regression, SVM, and nonlinear ones like K-NN, Kernel SVM and Random Forests.

<https://sds-platform-private.s3-us-east-2.amazonaws.com/uploads/P14-Classification-Pros-Cons.pdf>

Machine Learning Clustering models:

1. K-Means Clustering
2. Hierarchical Clustering



Association Rule Learning models:

1. Apriori
2. Eclat

Reinforcement Learning models: (e.g for walking)

1. Upper Confidence Bound (UCB)
2. Thompson Sampling

Also, a great complimentary resource for this chapter is the following book (you will recognize the author):

<https://www.amazon.com/gp/mpc/A18C1YWPOVXRKS>

Natural Language Processing

Speaking of classification algorithms, most of NLP algorithms are classification models, and they include Logistic Regression, Naive Bayes, CART which is a model based on decision trees, Maximum Entropy again related to Decision Trees, Hidden Markov Models which are models based on Markov processes.

A very well-known model in NLP is the Bag of Words model. It is a model used to preprocess the texts to classify before fitting the classification algorithms on the observations containing the texts.

Deep Learning

Deep Learning models can be used for a variety of complex tasks:

* Artificial Neural Networks for Regression and Classification
* Convolutional Neural Networks for Computer Vision
* Recurrent Neural Networks for Time Series Analysis
* Self Organizing Maps for Feature Extraction
* Deep Boltzmann Machines for Recommendation Systems
* Auto Encoders for Recommendation Systems

Dimensionality Reduction

There are two types of Dimensionality Reduction techniques:

1. Feature Selection
2. Feature Extraction

Feature Selection techniques are

Backward Elimination,

Forward Selection,

Bidirectional Elimination,

Score Comparison and more.

We covered these techniques in Part 2 - Regression.

Feature Extraction techniques:

1. Principal Component Analysis (PCA) - unsupervised.
2. Linear Discriminant Analysis (LDA) - supervised
3. Kernel PCA
4. Quadratic Discriminant Analysis (QDA)

Sagemaker built in algo.

<https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-algo-docker-registry-paths.html>

The Amazon SageMaker linear learner algorithm provides a solution for both classification and regression problems.

Amazon SageMaker Random Cut Forest (RCF) is an unsupervised algorithm for detecting anomalous data points within a data set. These are observations which diverge from otherwise well-structured or patterned data. Anomalies can manifest as unexpected spikes in time series data, breaks in periodicity, or unclassifiable data points.With each data point, RCF associates an anomaly score. Low score values indicate that the data point is considered "normal." High values indicate the presence of an anomaly in the data. The definitions of "low" and "high" depend on the application but common practice suggests that scores beyond three standard deviations from the mean score are considered anomalous.

The [XGBoost](https://github.com/dmlc/xgboost) (eXtreme Gradient Boosting) is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm that attempts to accurately predict a target variable by combining an ensemble of estimates from a set of simpler, weaker models.

Factorization machines are a good choice for tasks dealing with high dimensional sparse datasets, such as click prediction and item recommendation.

**Precision and Recall**

[**https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall**](https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall)

**Precision** refers to the percentage of your results which are relevant.

What proportion of positive identifications was actually correct?

**Recall** refers to the percentage of total relevant results correctly classified by your algorithm.

What proportion of actual positives was identified correctly? 