Data Mining

A Summary of Algorithms

Examples from "Data Science Concepts and Practice" 2018 MK : Vijay Kotu and Bala Deshpande

Terri Hoare – April 2021

Data Mining Taxonomy
Matching Problems to Data Mining Algorithms

- **Classification**
  Predicting a Categorical Target Variable (*supervised*)
- **Regression**
  Predicting a Numeric Target Variable (*supervised*)
- **Association**
  Unsupervised Process for Finding Relationships between Items
- **Clustering**
  Unsupervised Process for Finding Meaningful Groups in Data

https://www.youtube.com/watch?v=zDxh1dEt_Mo&index=2&list=PLssWC2d9JhOZZ6PCzJt2L2zUwA3RozrP

Feature Selection
Most Important Attributes

PCA (Filter-Based)	PCA is in reality a dimension reduction method. It combines the most important attributes into a fewer number of transformed attributes
Information Gain (Filter-Based)	Selecting attributes based on relevance to the target or label
Chi-Square (Filter-Based)	Selecting attributes based on relevance to the target or label
Forward Selection (Wrapper-Based)	Selecting attributes based on relevance to the target or label
Backward Elimination (Wrapper-Based)	Selecting attributes based on relevance to the target or label

Feature Selection
PCA (Filter-Based)

- **Model**
  N/A
- **Input**
  Numerical attributes
- **Output**
  Numerical attributes (reduced set). Does not really require a label
- **Pros**
  Efficient way to extract predictors that are uncorrelated to each other. Helps to apply Pareto principle in identifying attributes with highest variance
- **Cons**
  Very sensitive to scaling effects, i.e., requires normalization of attribute values before application. Focus on variance sometimes results in selecting noisy attributes
- **Use Cases**
  Most numeric- valued data sets that require dimension reduction

Feature Selection
Information Gain (Filter-Based)

- **Model** Similar to decision tree model
- **Input**
  No restrictions on variable type for predictors
- **Output**
  Data sets require a label. Can only be applied on data sets with nominal label
- **Pros**
  Same as decision trees
- **Cons**
  Same as decision trees
- **Use Cases**
  Applications for feature selection where target variable is categorical or numeric

Feature Selection
Chi-Square (Filter-Based)

- **Model**
  Uses the chi-square test of independence to relate predictors to label
- **Input**
  Categorical (polynominal) attributes
- **Output**
  Data sets require a label. Can only be applied on data sets with a nominal label
- **Pros**
  Very robust. A fast and efficient scheme to identify which categorical variables to select for a predictive model
- **Cons**
  Sometimes difficult to interpret
- **Use Cases**
  Applications for feature selection where all variables are categorical

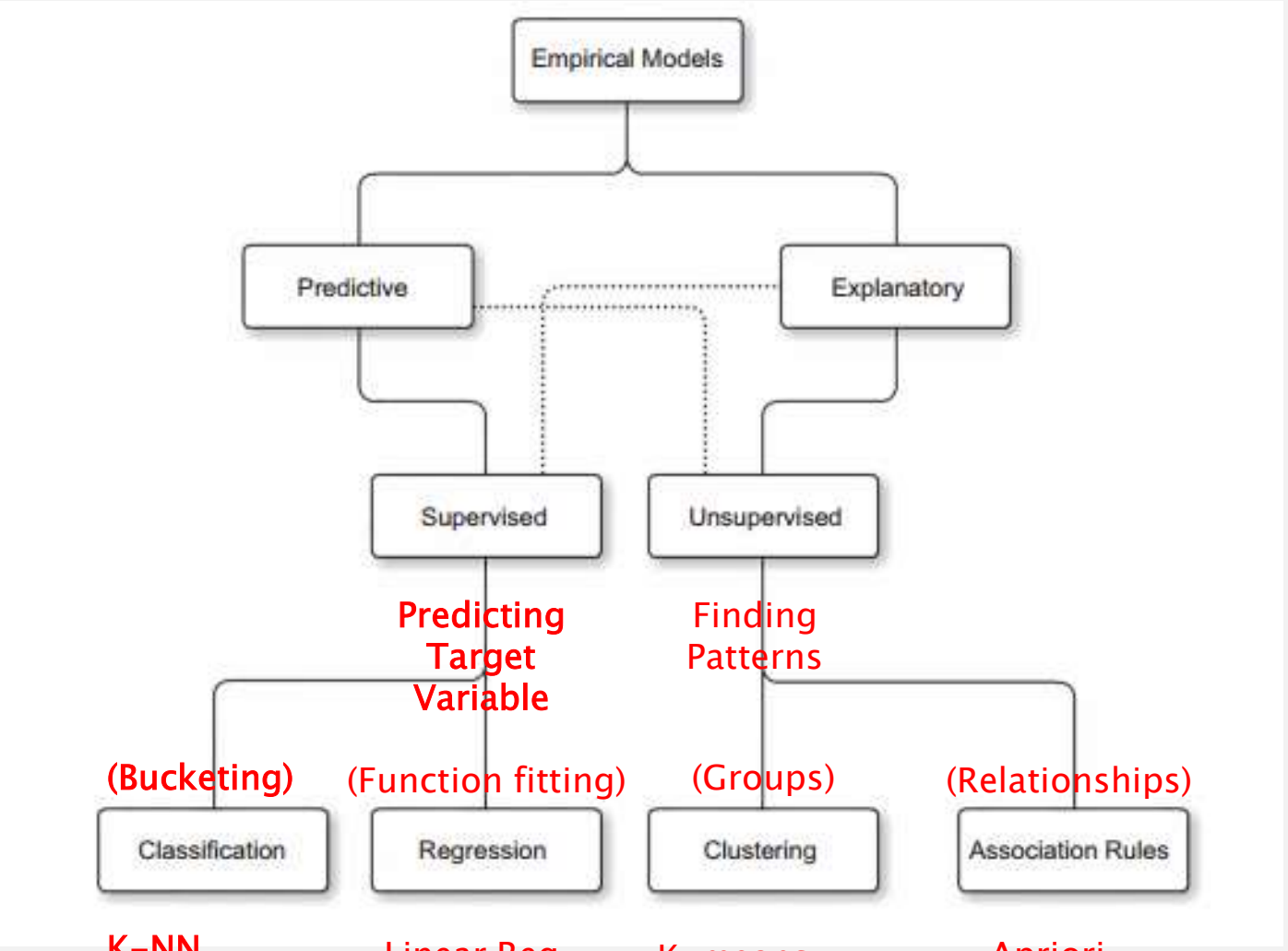
Feature Selection
Forward Selection (Wrapper-Based)

- **Model**
  Works in conjunction with modelling methods such as regression
- **Input**
  All attributes should be numeric
- **Output**
  The label may be numeric or binominal
- **Pros**
  Multicollinearity problems can be avoided. Speeds up the training phase of the modelling process
- **Cons**
  Once a variable is added to the set, it is never removed in subsequent iterations even if its influence on the target diminishes
- **Use Cases**
  Data sets with a large number of input variables where feature selection is required

Feature Selection
Backward Elimination (Wrapper-Based)

- **Model**
  Works in conjunction with modelling methods such as regression
- **Input**
  All attributes should be numeric
- **Output**
  The label may be numeric or binominal
- **Pros**
  Multicollinearity problems can be avoided. Speeds up the training phase of the modelling process
- **Cons**
  Need to begin with a full model, which can sometimes be computationally intensive
- **Use Cases**
  Data sets with few input variables where feature selection is required

Classification



Classification
Predicting a Categorical Target Variable

K-Nearest Neighbours	A lazy learner where no model is generalized. Any new unknown data point is compared against similar known data point in the training set
Decision Trees	Partitions the data into smaller subsets where each subset contains (mostly) responses of one class (“yes” or “no”)
Rule Induction	Models the relationship between input and output by deducing simple IF/THEN rules from a data set
Naïve Bayes	Predicts the output class based on Bayes’ theorem by calculating class conditional probability & prior probability
Artificial Neural Networks	A computational and mathematical model inspired by the biological nervous system. The weights in the network learn to reduce the error between actual and prediction
Support Vector Machines	Essentially a boundary detection algorithm that identifies/ defines multidimensional boundaries separating data points belonging to different classes
Ensemble Models	Leverages wisdom of the crowd. Employs a number of independent models to make a prediction and aggregates the final prediction

Classification Models
k-Nearest Neighbours : Summary

- **Model**
  Entire training dataset is the model
- **Input**
  No restrictions. However, distance calculations work better with numeric data. Data needs to be normalised
- **Output**
  Prediction of target variable which is categorical
- **Pros**
  Requires little time to build model. Handles missing values in the unknown record gracefully. Works with non-linear relationships
- **Cons**
  Deployment runtime and storage requirements expensive. Arbitrary selection of the value of k. No description of model
- **Use Cases**
  Image processing applications where slower response time is acceptable
- **Operating Parameters**
  K; weighted vote; measure types

Classification Models
Decision Trees : Summary

- **Model**
  Set of rules to partition data set based on values of different predictors
- **Input**
  No restriction on variable type for predictors
- **Output**
  Label cannot be numeric, must be categorical
- **Pros**
  Intuitive to explain to non-technical business users. Normalising predictors not necessary
- **Cons**
  Tends to over-fit the data. Small changes in input data can yield substantially different trees. Selecting right parameters can be challenging. Divides dataset in rectilinear fashion
- **Use Cases**
  Marketing segmentation, fraud detection
- **Operating Parameters**
  Split Criterion; Maximal Depth; Apply Pruning / Pre-pruning; Minimal Gain; Minimal Leaf Size; Minimal Size for Split; Number of Pre-pruning Alternatives

Classification Models
Rule Induction : Summary

- **Model**
  Set of rules that contain an antecedent (inputs) and consequent (output class)
- **Input**
  No restrictions. Accepts categorical, numeric and binary inputs
- **Output**
  Prediction of target variable which is categorical
- **Pros**
  Model can be easily explained to business users. Easy to deploy in almost any tools and applications
- **Cons**
  Divides dataset in rectilinear fashion
- **Use Cases**
  Manufacturing, applications where description of model is necessary
- **Operating Parameters**
  Criterion; Sample Ratio; Pureness; Minimal Prune Benefit

Classification Models
Naïve Bayesian : Summary

- **Model**
  A lookup table of probabilities and conditional probabilities for each attribute with an output class
- **Input**
  No restrictions. However, probability calculation works better with categorical attributes
- **Output**
  Prediction of probability for all class values, along with the winning class
- **Pros**
  Time required to model and deploy is minimum. Great algorithm for benchmarking. Strong statistical foundation
- **Cons**
  Training dataset needs to be representative sample of population and needs to have complete combinations of input and output. Attributes need to be independent
- **Use Cases**
  Spam detection, text mining
- **Operating Parameters**
  Laplace Correction

Classification Models
Artificial Neural Networks : Summary

- **Model**
  A network topology of layers and weights to process input data
- **Input**
  All attributes should be numeric
- **Output**
  Prediction of target (label) variable, which is categorical
- **Pros**
  Good at modelling nonlinear relationships. Fast response time in deployment
- **Cons**
  No easy way to explain the inner working of the model. Requires pre-processing data. Cannot handle missing attributes
- **Use Cases**
  Image recognition, fraud detection, quick response time applications
- **Operating Parameters**
  Hidden Layers; Training Cycles; Learning Rate; Momentum; Decay; Shuffle; Normalize; Error Epsilon

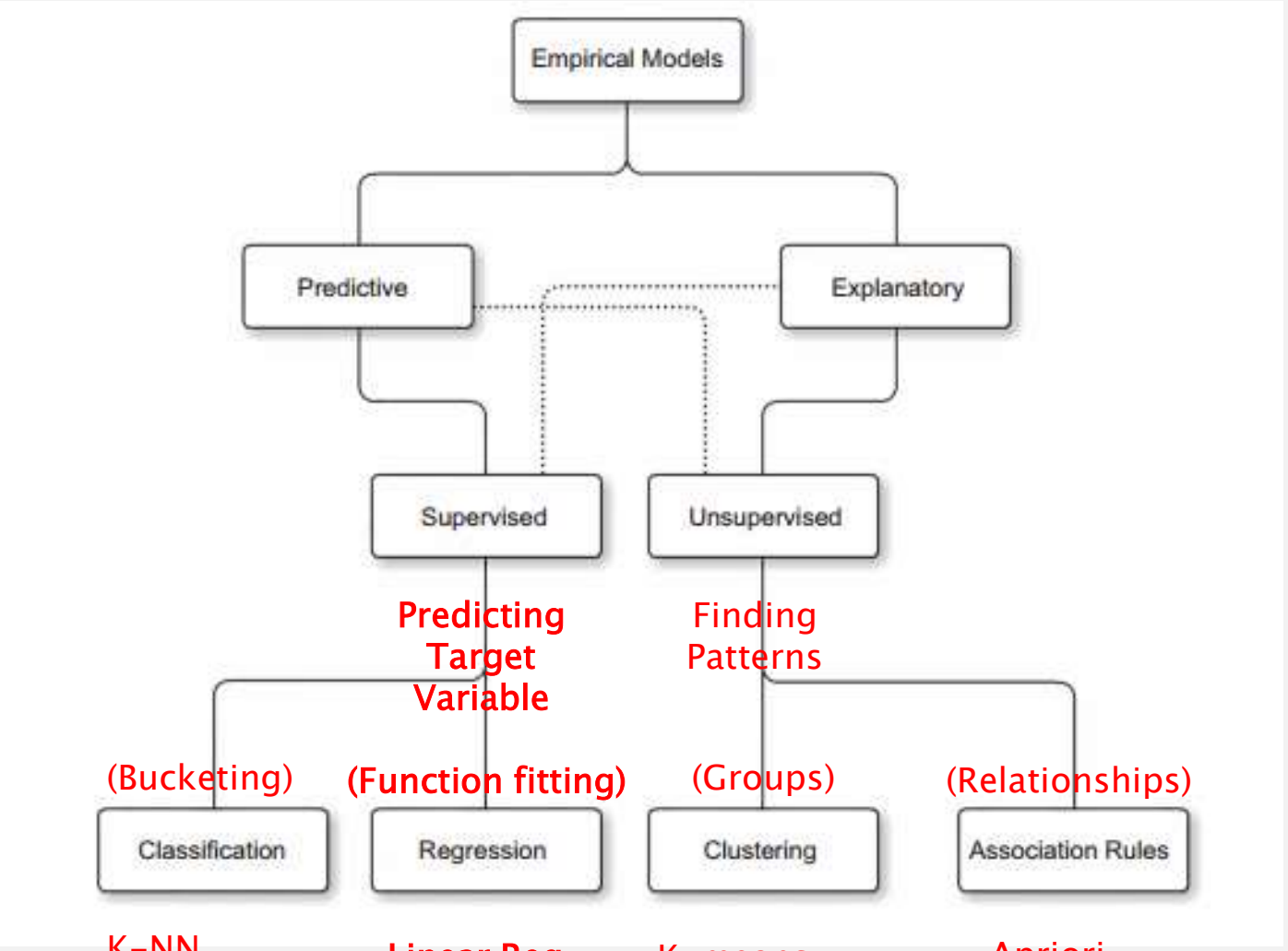
Classification Models
Support Vector Machines: Summary

- **Model**
  A vector equation that allows us to classify new data points into different regions (classes)
- **Input**
  All attributes should be numeric
- **Output**
  Prediction of target (label) variable, which can be categorical or numeric
- **Pros**
  Very robust against over-fitting. Small changes to input data do not affect boundary and thus do not yield different results. Good at handling nonlinear relationships
- **Cons**
  Computational performance during training phase can be slow. This may be compounded by the need to optimise parameter combinations
- **Use Cases**
  Optical character recognition, fraud detection, modelling “black swan” events
- **Operating Parameters**
  Kernel Type; Kernel Cache; C; Convergence Epsilon; Max Iterations; Scale; L- pos; L-neg; Epsilon; Balance Cost; Quadratic Loss pos-neg

Classification Models
Ensemble Models : Summary

- **Model**
  A meta-model with individual base models and an aggregator
- **Input**
  Superset of restrictions from the base model used
- **Output**
  Prediction for all class values with a winning class
- **Pros**
  Reduces the generalisation error. Takes different search space into consideration
- **Cons**
  Achieving model independence is tricky. Difficult to explain the inner working of the model
- **Use Cases**
  Most of the practical classifiers are ensemble
- **Approaches**
  Different models / parameters within models / training record sets / attribute sets e.g. Vote; Bagging; Boosting; AdaBoost; Random Forest

Regression


Empirical Models
Predictive
Explanatory
Supervised
Unsupervised
Predicting Target Variable
Finding Patterns
(Bucketing)
(Function fitting)
(Groups)
(Relationships)
Classification
Regression
Clustering
Association Rules
K-NN
Decision Trees
Rule Induction
Naïve Bayes
ANN
SVM
Ensemble
Linear Reg.
Logistic Reg.
Deep Learning
K-means
DBSCAN
SOMs
Anomaly-Distance
Anomaly-Density
Anomaly-Local
Outlier Factor
Apriori
FP Growth
Graph-based
Recommender - Collaborative filtering
Recommender - Content-based filtering

Regression
Predicting a Numeric Target Variable

Linear Regression	The classical predictive model that expresses the relationship between inputs and an output parameter in the form of an equation
Logistic Regression	Technically, this is a classification method. But structurally it is similar to linear regression. Logit (logarithm of the odds)

Regression Models
Linear Regression : Summary

- **Model**
  The model consists of coefficients for each input predictor and their statistical significance. A bias (intercept may be optional)
- **Input**
  All attributes should be numeric
- **Output**
  The label may be numeric or binomial
- **Pros**
  The workhorse of most predictive modelling techniques. Easy to use and explain to non technical business users
- **Cons**
  Cannot handle missing data. Categorical data are not directly usable, but require transformation into numeric
- **Use Cases**
  Any scenario that requires predicting a continuous numeric variable
- **Operating Parameters**
  Feature Selection (e.g. none, greedy); Eliminate Collinear Features; Use Bias
  Note:- will default MLR; use Select Attributes otherwise

Regression Models
Logistic Regression : Summary

- **Model**
  The model consists of coefficients for each input predictor that relate to the "logit". Transforming the logit into probabilities of occurrence (of each class) completes the model
- **Input**
  All attributes should be numeric (Note Weka also allows categorical)
- **Output**
  The label may only be binomial
- **Pros**
  One of the most common classification methods. Computationally efficient
- **Cons**
  Cannot handle missing data. Not very intuitive when dealing with a large number of predictors
- **Use Cases**
  Marketing scenarios (e.g. will click or not click), any general two class problem
- **Operating Parameters**
  RapidMiner implementation as for SVM (kernel etc)

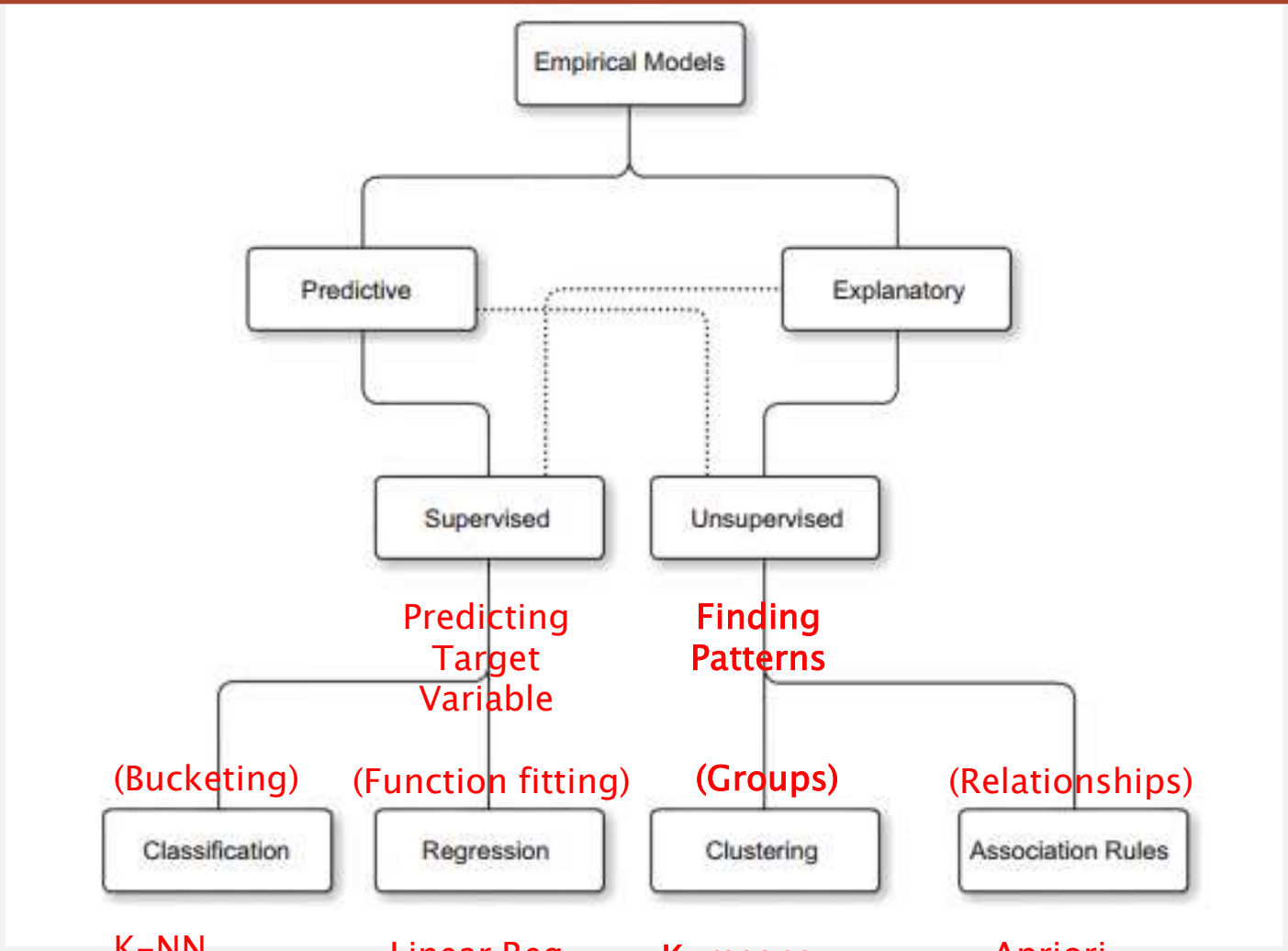
Classification Models
Deep Learning

Deep Learning (AI) – function fitting –
Training using multiple layers of representation of data –
Building on Linear and Logistic Regression Concepts

Use Cases – AI; Sound; Time Series; Text; Image; Video

Layer Type	Description	Input	Output	Pros	Cons	Use Cases
Convolutional	Based on the concept of applying filters to incoming two-dimensional representation of data, such as images. Machine learning is used to automatically determine the correct weights for the filters.	A tensor of typically three or more dimensions. Two of the dimensions correspond to the image while a third is sometimes used for color/channel encoding.	Typically the output of convolutional layer is flattened and passed through a dense or fully connected layer which usually terminates in a softmax output layer.	Very powerful and general purpose network. The number of weights to be learned in the conv layer is not very high.	For most practical classification problems, conv layers have to be coupled with dense layers which result in a large number of weights to be trained and thus lose any speed advantages of a pure conv layer.	Classify almost any data where spatial information is highly correlated such as images. Even audio data can be converted into images (using fourier transforms) and classified via conv nets.
Recurrent	Just as conv nets are specialized for analyzing spatially correlated data, recurrent nets are specialized for temporally correlated data: sequences. The data can be sequences of numbers, audio signals, or even images	A sequence of any type (time series, text, speech, etc).	RNNs can process sequences and output other sequences (many to many), or output a fixed tensor (many to one).	Unlike other types of neural networks, RNNs have no restriction that the input shape of the data be of fixed dimension.	RNNs suffer from vanishing (or exploding) gradients when the sequences are very long. RNNs are also not amenable to many stacked layers due to the same reasons.	Forecasting time series, natural language processing situations such as machine translation, image captioning.

Clustering


Empirical Models
Predictive
Explanatory
Supervised
Unsupervised
Predicting Target Variable
Finding Patterns
(Bucketing)
(Function fitting)
(Groups)
(Relationships)
Classification
Regression
Clustering
Association Rules
K-NN
Decision Trees
Rule Induction
Naïve Bayes
ANN
SVM
Ensemble
Linear Reg.
Logistic Reg.
Deep Learning
K-means
DBSCAN
SOMs
Anomaly-Distance
Anomaly-Density
Anomaly-Local
Outlier Factor
Apriori
FP Growth
Graph-based
Recommender - Collaborative filtering
Recommender - Content-based filtering

Clustering
Unsupervised Proc. – Finding Meaningful Groups in Data

K-Means	Data set is divided into k clusters by finding k centroids
DBSCAN	Identifies clusters as a high-density area surrounded by low-density areas
Self Organising Maps	A visual clustering technique with roots from neural networks and prototype clustering

Clustering Models
K-means Summary

- **Model**
  Algorithm finds k centroids and all the data points are assigned to the nearest centroids, which form a cluster
- **Input**
  No restrictions. However, the distance calculations work better with numeric data. Data should be normalized
- **Output**
  Data set is appended by one of k cluster labels
- **Pros**
  Simple to implement. Can be used for dimension reduction
- **Cons**
  Specification of k is arbitrary and may not find natural clusters. Sensitive to outliers
- **Use Cases**
  Customer segmentation, anomaly detection, applications where globular clustering is natural
- **Operating Parameters**
  K (no. of clusters); Add cluster as an attribute; Max. runs (with diff. initial centroids); Measure Type (Euclidean, Manhatten, Jaccard, Cosine Similarity); Max. Optimization Steps (iterations assign. data points re-calc. centroids)

Clustering Models
DBSCAN Summary

- **Model**
  List of clusters and assigned data points. Default Cluster 0 contains noise points.
- **Input**
  No restrictions. However, the distance calculations work better with numeric data. Data should be normalized
- **Output**
  Cluster labels based on identified clusters
- **Pros**
  Finds the natural clusters of any shape. No need to mention number of clusters
- **Cons**
  Specification of density parameters. A bridge between two clusters can merge the cluster. Can not cluster varying density data set
- **Use Cases**
  Applications where clusters are nonglobular shapes and when the prior number of natural groupings is not known
- **Operating Parameters**
  Epsilon and MinPoints (k-Dist. Graphs)

Clustering Models
SOM Summary

- **Model**
  A two- dimensional lattice where similar data points are arranged next to each other
- **Input**
  No restrictions. However, the distance calculations work better with numeric data. Data should be normalized
- **Output**
  No explicit clusters identified. Similar data points occupy either the same cell or are placed next to each other in the neighbourhood
- **Pros**
  A visual way to explain the clusters. Reduces multidimensional data to two dimensions
- **Cons**
  Number of centroids (topology) is specified by the user. Does not find natural clusters in the data
- **Use Cases**
  Diverse applications including visual data exploration, content suggestions, and dimension reduction
- **Operating Parameters**
  Grid dimensions

Clustering
Anomaly Detection : Finding Outliers

Distance Based	Outlier identified based on distance if kth nearest neighbour
Density Based	Outlier is identified based on data points in low-density regions
Local outlier factor	Outlier is identified based on calculation of relative density in the neighbourhood

Anomaly Detection
Distance Based Summary

- **Model**
  All data points are assigned a distance score based on nearest neighbour
- **Input**
  Accepts both numeric and categorical attributes. Normalization is required since distance is calculated
- **Output**
  Every data point has a distance score. The higher the distance, the more likely the data point is an outlier
- **Pros**
  Easy to implement. Works well with numeric attributes
- **Cons**
  Specification of k is arbitrary
- **Use Cases**
  Fraud detection, pre- processing technique
- **Operating Parameters**
  K; distance calculations

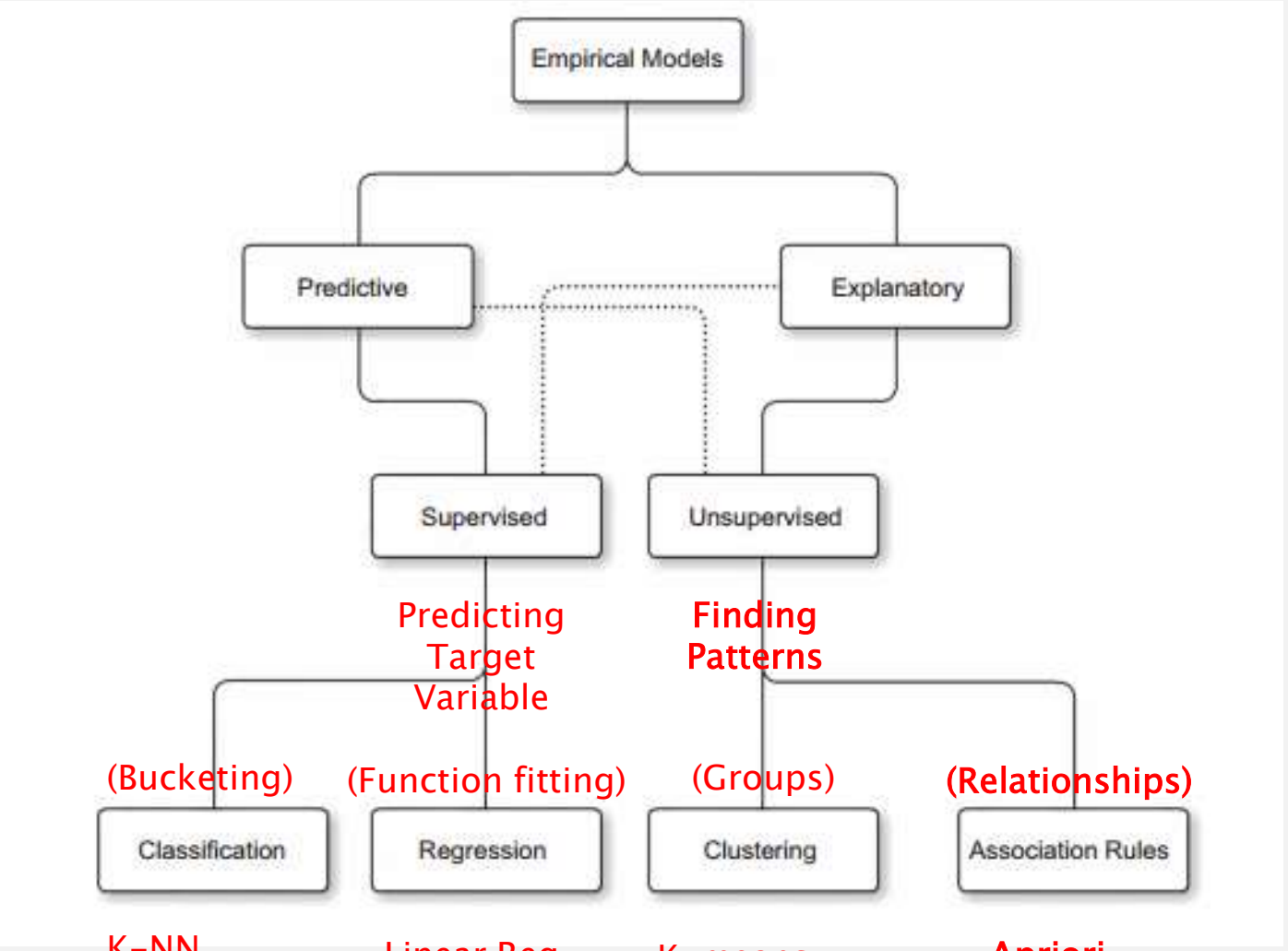
Anomaly Detection
Density Based Summary

- **Model**
  All data points as assigned a density score based on the neighbourhood
- **Input**
  Accepts both numeric and categorical attributes. Normalization is required since density is calculated
- **Output**
  Every data point has a density score. The lower the density, the more likely the data point is an outlier
- **Pros**
  Easy to implement. Works well with numeric attributes
- **Cons**
  Specification of distance parameter by the user. Inability to identify varying density regions
- **Use Cases**
  Fraud detection, pre-processing technique
- **Operating Parameters**
  Epsilon; MinPoints; distance calculations

Anomaly Detection
Local Outlier Factor Summary

- **Model**
  All data points as assigned a relative density score based on the neighbourhood
- **Input**
  Accepts both numeric and categorical attributes. Normalization is required since density is calculated
- **Output**
  Every data point has a density score. The lower the relative density, the more likely the data point is an outlier
- **Pros**
  Can handle the varying density scenario
- **Cons**
  Specification of distance parameter by the user
- **Use Cases**
  Fraud detection, pre-processing technique
- **Operating Parameters**
  Distance calculations

Association Rules



K-NN
Decision Trees
Rule Induction
Naïve Bayes
ANN
SVM
Ensemble

Linear Reg.
Logistic Reg.
Deep Learning

K-means
DBSCAN
SOMs
Anomaly-Distance
Anomaly-Density
Anomaly-Local
Outlier Factor

Apriori
FP Growth
Graph-based
Recommender - Collaborative filtering
Recommender - Content-based filtering

Association Analysis
Unsupervised Process – Finding Relations

FP-Growth and Apriori	Measures the strength of co-occurrence between one item with another

Association Rule Algorithms
FP Growth and Apriori

- **Model**
  Finds simple easy to understand rules like {Milk, Diaper} → {Beer}
- **Input**
  Transactions format with items in the columns and transactions in the rows
- **Output**
  List of relevant rules developed from the data set
- **Pros**
  Unsupervised approach with minimal user inputs. Easy to understand rules
- **Cons**
  Requires pre-processing if input is of a different format
- **Use Cases**
  Recommendation engines, cross-selling, and content suggestions
- **Concepts**
  Frequent item set; Support of an item; Support of a Rule; Confidence of a Rule; Lift of a Rule; Conviction of a Rule

Association Analysis
Recommenders – finding user preference for an item

Algorithm	Description	Assumption	Input	Output	Pros	Cons	Use Case
Collaborative Filtering - neighborhood based	Find a cohort of users who provided similar ratings. Derive the outcome rating from the cohort users	Similar users or items have similar likes	Ratings matrix with user-item preferences.	Completed ratings matrix	The only input needed is the ratings matrix Domain agnostic	Cold start problem for new users and items Computation grows linearly with the number of items and users	eCommerce, music, new connection recommendations
Collaborative Filtering - Latent matrix factorization	Decompose the user-item matrix into two matrices (P and Q) with latent factors. Fill the blank values in the ratings matrix by dot product of P and Q	User's preference of an item can be better explained by their preference of an item's character (inferred)	Ratings matrix with user-item preferences.	Completed ratings matrix	Works in sparse matrix More accurate than neighborhood based collaborative filtering	Cannot explain why the prediction is made	Content recommendations
Content-based filtering	Abstract the features of the item and build item profile. Use the item profile to evaluate the user preference for the attributes in the item profile	Recommend items similar to those the user liked in the past	User-item rating matrix and Item profile	Completed ratings matrix	Addresses cold start problem for new items Can provide explanations on why the recommendation is made	Requires item profile data set Recommenders are domain specific	Music recommendation from Pandora and CiteSeer's citation indexing
Content-based - Supervised learning models	A personalized classification or regression model for every single user in the system. Learn a classifier based on user likes or dislikes of an item and its relationship with item attributes	Every time a user prefers an item, it is a vote of preference for item attributes	User-item rating matrix and Item profile	Completed ratings matrix	Every user has a separate model and could be independently customized. Hyper personalization	Storage and computational time	eCommerce, content, and connection recommendations

Association Analysis
Unsupervised Process – Graph Algorithms

GRAPH Centralities	These algorithms determine the importance of distinct nodes in a network e.g. Page Rank; Betweenness Centrality; Closeness Centrality; Eigenvector Centrality
GRAPH Community Detection	These algorithms evaluate how a group is clustered or partitioned, as well as its tendency to strengthen or break apart e.g. Triangle Count; Clustering Coefficient
GRAPH Path Finding	These algorithms help find the shortest path or evaluate the availability and quality of routes e.g. Minimum Weight Spanning Tree; All pairs and Single Source Shortest Path