Comparison of Data Science Algorithms

Classification: Predicting a categorical target variable

Algorithm	Description	Model	Input	Output	Pros	Cons	Use Cases
Decision trees	Partitions the data into smaller subsets where each subset contains (mostly) responses of one class (either "yes" or "no")	A set of rules to partition a data set based on the values of the different predictors	No restrictions on variable type for predictors	The label cannot be numeric. It must be categorical	Intuitive to explain to nontechnical business users. Normalizing predictors is not necessary	Tends to overfit the data. Small changes in input data can yield substantially different trees. Selecting the right parameters can be challenging	Marketing segmentation, fraud detection
Rule induction	Models the relationship between input and output by deducing simple "IF/THEN" rules from a data set	A set of organized rules that contain an antecedent (inputs) and consequent (output class)	No restrictions. Accepts categorical, numeric, and binary inputs	Prediction of target variable, which is categorical	Model can be easily explained to business users Easy to deploy in almost any tools and applications	Divides the data set in rectilinear fashion	Manufacturing, applications where description of model is necessary
k-Nearest neighbors	A lazy learner where no model is generalized. Any new unknown data point is compared against similar known data points in the training set	Entire training data set is the model	No restrictions. However, the distance calculations work better with numeric data. Data needs to be normalized	Prediction of target variable, which is categorical	Requires very little time to build the model. Handles missing attributes in the unknown record gracefully. Works with nonlinear relationships	The deployment runtime and storage requirements will be expensive Arbitrary selection of value of *k* No description of the model	Image processing, applications where slower response time is acceptable
Naïve Bayesian	Predicts the output class based on the Bayes' theorem by calculating class conditional probability and prior probability	A lookup table of probabilities and conditional probabilities for each attribute with an output class	No restrictions. However, the probability calculation works better with categorical attributes	Prediction of probability for all class values, along with the winning class	Time required to model and deploy is minimum Great algorithm for benchmarking. Strong statistical foundation	Training data set needs to be representative sample of population and needs to have complete combinations of input and output. Attributes need to be independent	Spam detections, text mining

Continued

Continued

Algorithm	Description	Model	Input	Output	Pros	Cons	Use Cases
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	A network topology of layers and weights to process input data	All attributes should be numeric	Prediction of target (label) variable, which is categorical	Good at modeling nonlinear relationships. Fast response time in deployment	No easy way to explain the inner working of the model Requires preprocessing data. Cannot handle missing attributes	Image recognition, fraud detection, quick response time applications
Support vector machines	Boundary detection algorithm that identifies/defines multi-dimensional boundaries separating data points belonging to different classes	The model is a vector equation that allows one to classify new data points into different regions (classes)	All attributes should be numeric	Prediction of target (label) variable, which can be categorical or numeric	Extremely robust against overfitting. Small changes to input data do not affect boundary and, thus, do not yield different results. Good at handling nonlinear relationships	Computational performance during training phase can be slow. This may be compounded by the effort needed to optimize parameter combinations	Optical character recognition, fraud detection, modeling "black-swan" events
Ensemble models	Leverages wisdom of the crowd. Employs a number of independent models to make a prediction and aggregates the final prediction	A meta-model with individual base models and an aggregator	Superset of restrictions from the base model used	Prediction for all class values with a winning class	Reduces the generalization error. Takes different search space into consideration	Achieving model independence is tricky Difficult to explain the inner working of the model	Most of the practical classifiers are ensemble

Regression: Predicting a numeric target variable

Algorithm	Description	Model	Input	Output	Pros	Cons	Use Case
Linear regression	The classical predictive model that expresses the relationship between inputs and an output parameter in the form of an equation	The model consists of coefficients for each input predictor and their statistical significance. A bias (intercept) may be optional	All attributes should be numeric	The label may be numeric or binominal	The workhorse of most predictive modeling techniques. Easy to use and explain to nontechnical business users	Cannot handle missing data. Categorical data are not directly usable, but require transformation into numeric	Pretty much any scenario that requires predicting a continuous numeric value
Logistic regression	Technically, this is a classification method. But structurally it is similar to linear regression	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	All attributes should be numeric	The label may only be binominal	One of the most common classification methods. Computationally efficient	Cannot handle missing data. Not intuitive when dealing with a large number of predictors	Marketing scenarios (e.g., will click or not click), any general two-class problem

Association analysis: Unsupervised process for finding relationships between items

Algorithm	Description	Model	Input	Output	Pros	Cons	Use Case
FP-Growth and Apriori	Measures the strength of co-occurrence between one item with another	Finds simple, easy to understand rules like {Milk, Diaper} → {Beer}	Transactions format with items in the columns and transactions in the rows	List of relevant rules developed from the data set	Unsupervised approach with minimal user inputs. Easy to understand rules	Requires preprocessing if input is of different format	Recommendation engines, cross-selling, and content suggestions

Clustering: An unsupervised process for finding meaningful groups in data

Algorithm	Description	Model	Input	Output	Pros	Cons	Use case
k-Means	Data set is divided into *k* clusters by finding *k* centroids	Algorithm finds *k* centroids and all the data points are assigned to the nearest centroid, which forms a cluster	No restrictions. However, the distance calculations work better with numeric data. Data should be normalized	Data set is appended by one of the *k* cluster labels	Simple to implement. Can be used for dimension reduction	Specification of *k* is arbitrary and may not find natural clusters. Sensitive to outliers	Customer segmentation, anomaly detection, applications where globular clustering is natural
DBSCAN	Identifies clusters as a high-density area surrounded by low-density areas	List of clusters and assigned data points. Default cluster 0 contains noise points	No restrictions. However, the distance calculations work better with numeric data. Data should be normalized	Cluster labels based on identified clusters	Finds the natural clusters of any shape. No need to mention number of clusters	Specification of density parameters. A bridge between two clusters can merge the cluster. Cannot cluster varying density data set	Applications where clusters are nonglobular shapes and when the prior number of natural groupings is not known
Self-organizing maps	A visual clustering technique with roots from neural networks and prototype clustering	A two-dimensional lattice where similar data points are arranged next to each other	No restrictions. However, the distance calculations work better with numeric data. Data should be normalized	No explicit clusters identified. Similar data points occupy either the same cell or are placed next to each other in the neighborhood	A visual way to explain the clusters. Reduces multi-dimensional data to two dimensions	Number of centroids (topology) is specified by the user. Does not find natural clusters in the data	Diverse applications including visual data exploration, content suggestions, and dimension reduction

Anomaly detection: Supervised and unsupervised techniques to find outliers in data

Algorithm	Description	Model	Input	Output	Pros	Cons	Use Case
Distance–based	Outlier identified based on distance to *k*th nearest neighbor	All data points are assigned a distance score based on nearest neighbor	Accepts both numeric and categorical attributes. Normalization is required since distance is calculated	Every data point has a distance score. The higher the distance, the more likely the data point is an outlier	Easy to implement. Works well with numeric attributes	Specification of *k* is arbitrary	Fraud detection, preprocessing technique
Density-based	Outlier is identified based on data points in low-density regions	All data points are assigned a density score based on the neighborhood	Accepts both numeric and categorical attributes. Normalization is required since density is calculated	Every data point has a density score. The lower the density, the more likely the data point is an outlier	Easy to implement. Works well with numeric attributes	Specification of distance parameter by the user. Inability to identify varying density regions	Fraud detection, preprocessing technique
Local outlier factor	Outlier is identified based on calculation of relative density in the neighborhood	All data points as assigned a relative density score based on the neighborhood	Accepts both numeric and categorical attributes. Normalization is required since density is calculated	Every data point has a density score. The lower the relative density, the more likely the data point is an outlier	Can handle the varying density scenario	Specification of distance parameter by the user	Fraud detection, preprocessing technique

Deep learning: Training using multiple layers of representation of data

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.

Recommenders: Finding the user's preference of 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

Time series forecasting: Predicting future value of a variable

Algorithm	Description	Model	Input	Output	Pros	Cons	Use Case
Decomposition	Decompose the time series into trend, seasonality, and noise. Forecast the components	Models for the individual components	Historical value	Forecasted value	Increased understanding of the time series by visualizing the components	Accuracy depends on the models used for components	Application where the explanation of components is important

Continued

Continued

Algorithm	Description	Model	Input	Output	Pros	Cons	Use Case
Exponential smoothing	The future value is the function of past observations	Learn the parameters of the smoothing equation from the historical data	Historical value	Forecasted value	Applies to wide range of time series with or without trend or seasonality	Multiple seasonality in the data make the models cumbersome	Cases where trend or seasonality is not evident
ARIMA (autoregressive integrated moving average)	The future value is the function of auto correlated past data points and the moving average of the predictions	Parameter values for (p,d, q), AR, and MA coefficients	Historical value	Forecasted value	Forms a statistical baseline for model accuracy	The optimal p,d,q value is unknown to begin with	Applies on almost all types of time series data
Windowing-based machine learning	Create cross-sectional data set with time lagged inputs	Machine learning models like regression, neural networks, etc.	Historical value	Forecasted value	Uses any machine learning approaches on the cross-sectional data	The windowing size, horizon, and skip values are arbitrary	Applies to user cases where the time series has trend and/or seasonality

Feature selection: Selection of most important attributes

Algorithm	Description	Model	Input	Output	Pros	Cons	Use Case
PCA (principal component analysis) filter-based	combines the most important attributes into a fewer number of transformed attributes	Each principal component is a function of attributes in the dataset	Numerical attributes	Numerical attributes (reduced set). Does not really require a label	Efficient way to extract predictors that are uncorrelated to each other. Helps to apply Pareto principle in identifying attributes with highest variance	Sensitive to scaling effects, i.e., requires normalization of attribute values before application. Focus on variance sometimes results in selecting noisy attributes	Most numeric-valued data sets require dimension reduction
Info gain (filter-based)	Selecting attributes based on relevance to the target or label	Similar to decision tree model	No restrictions on variable type for predictors	Data sets require a label. Can only be applied on data sets with nominal label	Same as decision trees	Same as decision trees	Applications for feature selection where target variable is categorical or numeric
Chi-square (filter-based)	Selecting attributes based on relevance to the target or label	Uses the chi-square test of independence to relate predictors to label	Categorical (polynominal) attributes	Data sets require a label. Can only be applied on data sets with a nominal label	Extremely robust. A fast and efficient scheme to identify which categorical variables to select for a predictive model	Sometimes difficult to interpret	Applications for feature selection where all variables are categorical

Continued

Continued

Algorithm	Description	Model	Input	Output	Pros	Cons	Use Case
Forward Selection (wrapper-based)	Selecting attributes based on relevance to the target or label	Works in conjunction with modeling methods such as regression	All attributes should be numeric	The label may be numeric or binominal	Multicollinearity problems can be avoided. Speeds up the training phase of the modeling process	Once a variable is added to the set, it is never removed in subsequent iterations even if its influence on the target diminishes	Data sets with a large number of input variables where feature selection is required
Backward elimination (wrapper-based)	Selecting attributes based on relevance to the target or label	Works in conjunction with modeling methods such as regression	All attributes should be numeric	The label may be numeric or binominal	Multicollinearity problems can be avoided. Speeds up the training phase of the modeling process	Need to begin with a full model, which can sometimes be computationally intensive	Data sets with few input variables where feature selection is required