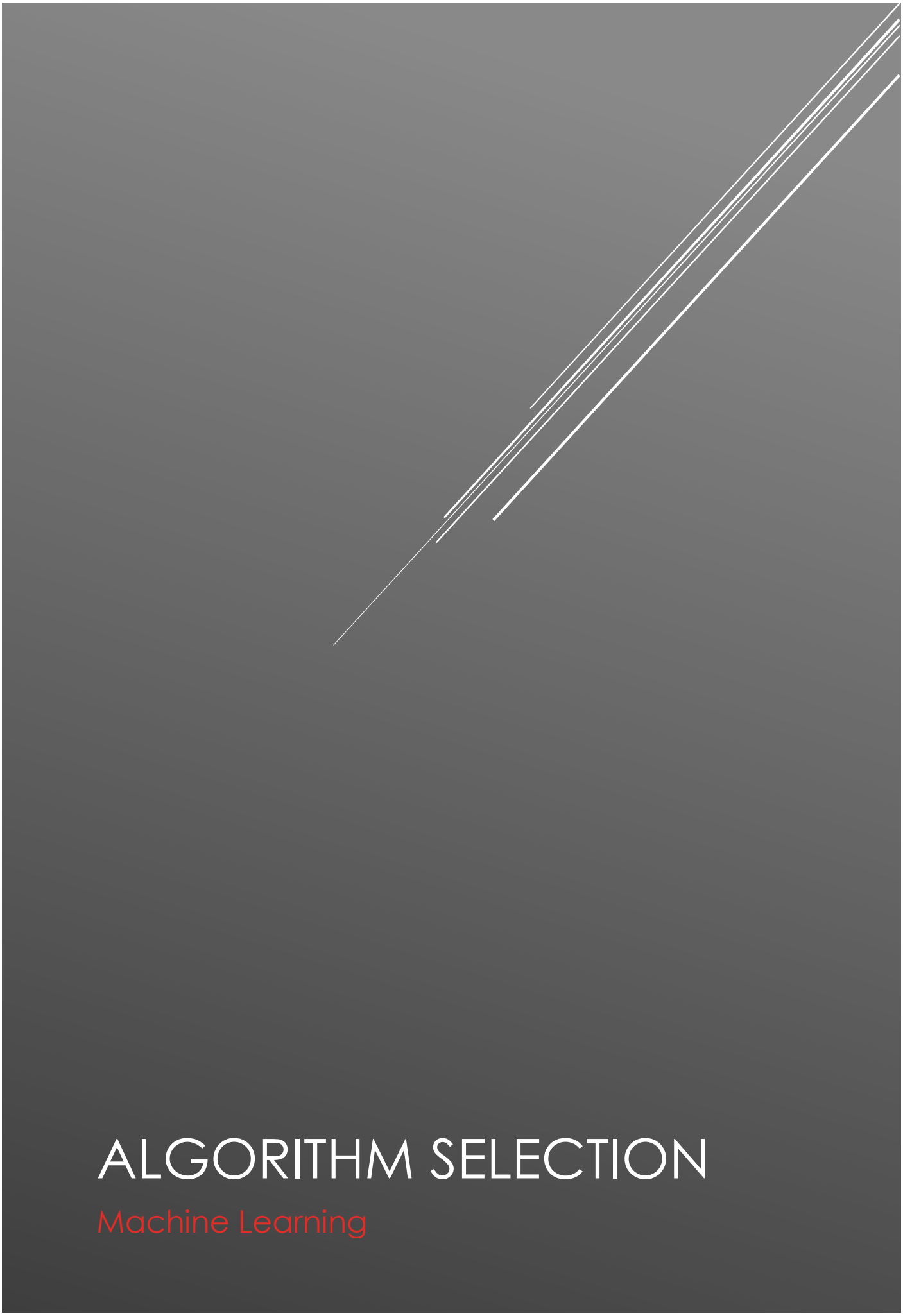

ALGORITHM SELECTION
Machine Learning

ALGORITHM SELECTION CRITERIA

CATEGORIZE PROBLEM

- Input category (supervised | unsupervised | reinforcement learning)
- Output category (regression | classification | clustering | anomaly detection)

UNDERSTAND CONSTRAINTS

- Data storage capacity
- Speed of prediction
- Speed of learning

UNDERSTAND DATA

- Sample set size
- Data types (categorical | continuous)
- Missing data sensitivity
- Outliers sensitivity
- Features multi-collinearity

BUSINESS CRITERIA

- Meeting business goals
- Pre-processing requirements
- Model accuracy
- Model scalability
- Model explain-ability
- Model complexity
    - # of features
    - Feature engineering needs
    - computational overhead
    - # of hyper-parameters

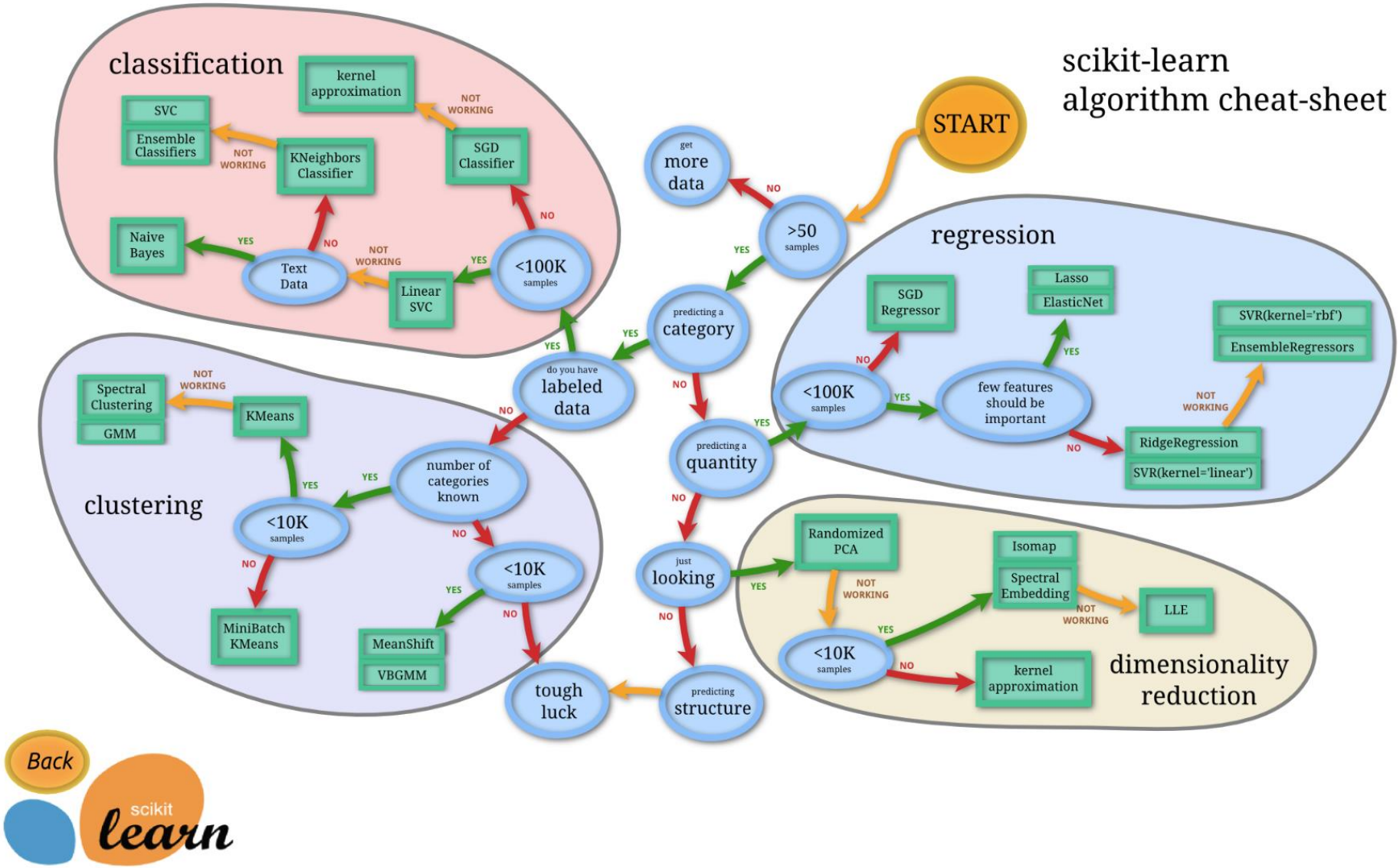
SELECTION QUESTIONS

- Number of training examples, (how large is the training dataset)
    - If small: high bias/low variance classifiers (e.g., Naïve Bayes), less likely to over-fit
    - If large: low bias/high variance classifiers (e.g., kNN or Logistic regression)
- Dimensionality of the feature space
- Is the problem linearly separable?
- Are features independent?
- Are features expected to be linearly dependent with the target variable?
- Is overfitting expected to be a problem/
- System requirements: speed, performance, memory usage
- Does it requires variables to be normally distributed?
- Does it suffers multi-collinearity issue?
- Does it do as well with categorical as continuous variables?
- Does it calculate CI without CV?
- Does it conduct variables selection without stepwise?
- Does it apply to sparse data?

Start with simple like logistic regression to set a baseline and only make it more complicated if you need to. At that point, tree ensembles, and in particular Random Forests since they are easy to tune, might be the right way to go. If you feel there is still room for improvement, try GBDT or get even fancier and go for Deep learning

REFERENCES

- Choose the right machine learning algorithm
- Choosing right estimator
- ML algorithms pros and cons
- Modern machine learning algorithms: strengths and weaknesses
- ML algorithms cheat sheets


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
NO
Text Data
NOT WORKING
YES
Linear SVC
<100K samples
NO
YES
get more data
NO
>50 samples
YES
regression
SGD Regressor
Lasso
ElasticNet
predicting a category
YES
NO
<100K samples
NO
YES
few features should be important
YES
SVR(kernel='rbf')
EnsembleRegressors
NOT WORKING
RidgeRegression
NO
SVR(kernel='linear')
do you have labeled data
YES
NO
predicting a quantity
YES
NO
clustering
Spectral Clustering
GMM
NOT WORKING
KMeans
YES
number of categories known
YES
NO
<10K samples
NO
MiniBatch KMeans
<10K samples
YES
NO
MeanShift
VBGMM
just looking
YES
NO
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
tough luck
predicting structure
Back
scikit
learn

Algorithm	Applications	Advantages	Disadvantages
Principal Component Analysis (PCA	• Dimensionality reduction • Data pre-processing method for other ML algorithms sensitive to feature multi-collinearity	• Only the data pre-processing option of highly correlated features to prevent overfitting for other ML algorithms sensitive to multi-collinearity • Visual representation of highly dimensional features	• Sometimes is difficult to interpret low dimensional representation of variables (principal components) as linear combination of features
K-Means Method	Finding natural grouping of observations: • Customer segmentation • Grouping similar items in e-commerce • Social network analysis	• Flexible enough to get reasonable results if you pre-process your data and engineer useful features • Fast and simple algorithm • Can process large datasets • Often terminates at a local optimum • Applicable only when mean is defined	• Not applicable for categorical data • Unable to handle noisy data • Need to define number of clusters in advance • Not suitable to discover clusters with non-convex shapes • Poor performance for not globular clusters
K-nearest Neighbour (k-NN) Classifier		• Nonparametric • Zero cost in the learning process • Classifying any data whenever finding similarity measures of any given instance intuitive approach • Robust to outliers on the predictors	• Hard to interpret the result • Performs poorly for high-dimensional data, use regression or ensembles • Lack of explicit model training • Susceptible to correlated inputs and irrelevant features • Very difficult in handling data of mixed types • Memory intensive as they need to save each training observation
Affinity Propagation		• Used for smaller and uneven classes patterns • No need to specify number of clusters	• Slow and memory intensive, so it does not scale well with datasets • Poor performance for non-globular clusters
Hierarchical or Agglomerative Clustering		• Clusters are not assume to be globular • Scales well with dataset	• Need to choose number of the clusters
DBSCAN Clustering		• Density based clustering • Not assume globular clusters • Scalable with dataset	• Difficult to tune sensitive hype-parameters
Linear Regression (Regularized)	• Time to go one location to another • Predicting sales of particular product next month • Predict real estate prices	• Easy to understand – clearly seen what the biggest drivers of the model are • Can model only linear relationships between variables	• Not flexible enough to capture more complex patterns and adding the right interaction terms or polynomials can be tricky and time consuming

Algorithm	Applications	Advantages	Disadvantages
Linear Regression (Regularized)	• Predict stock price movements • Predict student scores • Impact of blood alcohol content on coordination • Predict monthly gift card sales and improve yearly revenue projections	• Model can be easily updated with new data using stochastic gradient descent	• Perform poorly if there are non-linear relationships • Tendency for model to over-it, risk can be reduced by regularization (Lasso, Ridge, Elastic-Net) • Sensitive to outliers • Unstable in case features are redundant, multi-collinear
Naïve Bayes	• Sentiment analysis and test classification • Recommendation systems like Netflix, Amazon • Spam detectors • Face recognition	• Easy to train as it converge quicker than discriminative models (ex. Logistic regression) under conditional independence assumption, so you need less training data • Scale well with growing dataset • Along with the simplicity outperform highly sophisticated classification methods (in case of not proper tuning) • Good choice if CPU and memory resources are a limiting factor • Can easily obtain the probability for a prediction • Relatively simple and straightforward to use • Can deal with some noise and missing data • Nonparametric – non distribution requirements • Good for few categorical variables	• Can handle multiple classes • Prone to bias when increasing the number of training sets • Assumes all features are independent and equally important, which is unlikely in real-world cases • Sensitive to how the input data is prepared • Can't learn interactions between features • Suffers multi-collinearity, but even if conditional independence assumption is not met it perform pretty well in practice • Often beaten by more sophisticated models properly tuned

Algorithm	Applications	Advantages	Disadvantages
Support Vector Machines (SVM)	• Used for pattern recognition and classification with two classes • Popular in text classification problems • Detecting persons with common diseases such as diabetes • Hand-written character recognition • Text categorization – news articles by topics • Stock market price prediction	• Can utilize predictive power of linear combination of inputs • Good prediction in a variety of situations • Low generalization error • Easy to interpret results • Use non-linear kernels to model complex decision boundaries • Robust to overfitting especially in high dimensional space • Linear kernel is similar to Logistic regression which can also be used with different kernels • Good in a high-dimensional space (e.g text classification) • High accuracy • Good theoretical guarantees regarding overfitting • No distribution requirement • Compute hinge loss • Flexible selection of kernel for non-linear correlation • Not suffer multi-collinearity	• Very black box • Sensitive to tuning parameters and kernel choice • Testing data should be near to training data • Hard to interpret • Can be inefficient to train as it is memory-intensive for large datasets • Not for most „industry scale“ applications (anything beyond the toy or lab problem) • Weak in natural handling of mixed data types and scalability to large datasets • Currently in industry Random Forest are preferred over SVM's
Stochastic Gradient Descent	Large scale and sparse machine learning problems • Text classification • Natural Language Processing	• Simple yet very efficient approach to discriminative learning • Used for large scale learning in order of magnitude 10^5 for training examples and features • Easy implementation (lots of opportunities for code tuning)	• Only linear classifier comparable with SVC using linear kernel or Logistic Regression • Sensitive to feature scaling

Algorithm	Applications	Advantages	Disadvantages
Decision Trees	• Used very rarely, but basis of very efficient ensemble methods, like random forest or gradient tree boosting Excellent tools for helping to choose between several courses of action • Investment decisions • Customer churn • Bank loan defaulters • Build vs. buy decisions • Sales lead qualifications	• Easily handle feature interactions and complex non-linear relationships due to branching algorithm based on maximizing information gain • Non-parametric – no distribution requirements • Less sensitive for outliers • Easily interpretable • Can handle missing data • Able to handle both numerical and categorical data • Performs well with large datasets • Heuristic • No suffer multi-collinearity • Handle very well large dimensional spaces as well as large number of training examples	• Can't work on (linear) combinations of features • Relatively less predictive in many situations • Practical decision-tree learning algorithms cannot guarantee to return the globally optimal decision tree • Can lead to overfitting, but ensemble methods prevent overfitting by feature or training sample randomization • Not often used its own for prediction because it's also often too simple and not powerful enough for complex data • Good for only few categorical variables • Can easily over-fit, this can be alleviated by using ensembles like RF or GBM
Bagging		• Helps reduce variance and avoid overfitting	• Loss of interpretability of the model • Problem with high bias if not modelled properly • Computationally expensive
Random Forest	• Genetics • Predict patients for high risks • Predict part failures in manufacturing • Predict loan defaulters	• Can solve regression and classification problems with large datasets • Highly scalable to any number of dimensions with quite acceptable performance • A sort of „wisdom of the crowd", tends to result in very high quality models out of the box, without extra need of testing and tunning • Train each tree independently, using a random sample for the data, so the trained model is more robust than single decision tree, and less likely to over-fit	• Can be slow to output predictions relative to other algorithms • Not easy to understand predictions • Learning can be slow • Not possible to iteratively improve

Algorithm	Applications	Advantages	Disadvantages
Random Forest		• 2 parameters: number of trees and number of features to be selected at each node • Good for parallel or distributed computing • Lower classification error and better scores than decision trees • Perform as well as or better than SVMs, but far easier for humans to understand • Good with uneven data sets with missing variables • It helps identify significant features (feature importance) • Train faster than SVMs	
AdaBoost	Basic and more complex recognition problems • Biology • Computer vision • Speech processing	• Easily interpretable • Can be manually tuned • Can achieve similar classification results with much less tweaking of parameters or settings • Less sensitive to overfitting	• Can be sensitive to noisy data and outliers
Gradient Boosting	•	• High performing • Build trees one at a time, each new tree corrects some errors made by the previous trees, the model becomes even more expressive • Usually performs better than Random forests, but it is harder to tune	• A small change in the feature set or training set can create radical changes in the model • Not easy to understand predictions • 3 parameters – number of trees, depth, of trees, and learning rate: trees are generally shallow • The hyper parameters are hardest to tune and more prone to overfitting. RFs can almost work „out of box“ • Training takes slower since trees are built sequentially
Linear Discriminant Analysis	•	• Compute the addition of multivariate distribution • Compute confidence intervals	• Require normal distribution • Not good for few categorical variables • Suffers multi-collinearity

Algorithm	Applications	Advantages	Disadvantages
Logistic Regression	• Predicting the customer churn • Credit scoring • Fraud detection • Measuring effectiveness of marketing campaigns	• Probabilistic interpretation • Easy understand feature importance • Provides confidence intervals • Quickly update classification model to incorporate new data using stochastic gradient descent • Output can be interpreted as probability, which can be used for ranking instead of classification • Can easily „feature engineering“ most non-linear features into linear ones • Robust to noise • Can use L1 and L2 to avoid overfitting (and for feature selection) • Efficient, can be distributed (ADMM) • No need to worry about features being correlated	• Does not handle the missing value of continuous variables • Suffers multi-collinearity • Sensitive to extreme values of continuous variables • Sometimes too simple to capture complex relationships between variables • Tendency for model to over-fit • No distribution requirements • Variable selection • Features are expected to be roughly linear • Cannot handle categorical (binary) variables well • Underperform when there are multiple or non-linear decision boundaries Don‘t use when: • If the variables are normally distributed and the categorical variables all have 5+ categories: use Linear discriminant analysis • If the correlations are mostly nonlinear: use VSM • If sparsity and multi-collinearity are a concern: Adaptive Lasso with Ridge (weights) + Lasso
		Lasso (L1) • No distribution requirement	Lasso (L1) • Variable selection • Suffers multi-collinearity
		Ridge (L2) • No distribution requirement	Ridge (L2) • No variable selection • No suffer multi-collinearity

Algorithm	Applications	Advantages	Disadvantages
Neural Networks and Deep Learning	• Object recognition • Feature extractions • Computer vision • Speech recognition	• Used for classification and regression problems • Good prediction generally • Some toelrance to correlated inputs • Incorporating the predictive power of different combinations of inputs • Can handle extremely complex tasks – no other algorithms comes close to image recognition • Good to model the non-linear data with large number of input features • Widely used in industry • Many open source implementations • No need for complex feature engineering usually performed by hidden layers • Can learn extremely complex patterns due to hidden layers modelling intermediary representation of features • Good in image classification, video, audio, text • Performs better than any other algorithms • Can be easily updated with new data using batch propagation • Flexible as their architecture can be adapted for many types of problems	• Not a general purpose technique for classification as it needs very large amount of data to train • Not robust to outliers • Susceptible to irrelevant features • Difficult in dealing with big data with complex model • Only for numerical inputs, vectors with constant number of values, and datasets with non-missing data • „black box, the classification boundaries are hard to understand intuitively • Computational expensive • The trained model depends crucially on initial parameters • Difficult to troubleshoot when they don't work as expected • Not sure if they will generalize well to data not in training set • Multi-layer neural networks are usually hard to train, and require tuning lots of parameters what needs more expertise in architecture and hyper-parameters • Not probabilistic, unlike their more statistical or Bayesian counterparts. The continues number output (eg. A score) can be difficult to translate that into a probability • Computationally and memory expensive • Usually outperform by ensembles for classical machine learning problems