

ML model	Assumptions	Advantages	Disadvantages	Feature Scaling	Missing Data	Outliers	Suitable for	Learning	Example Use
Naïve Bayes Classifier	Features are independent	<ul style="list-style-type: none"> Performs well with categorical variables Converges faster: less training time Good with moderate to large training data sets Good when dataset contains several features 	<ul style="list-style-type: none"> Correlated features affect performance 	No	Can handle missing data (it ignores missing data)	Robust to outliers	<ul style="list-style-type: none"> Classification Multiclass classification 	Supervised	<ul style="list-style-type: none"> Sentiment Analysis Document categorisation Email Spam Filtering
Support Vector Machine (SVM)	None	<ul style="list-style-type: none"> Good for datasets with more variables than observations Good performance Good of-the-shelf model in general for several scenarios Can approximate complex non-linear functions 	<ul style="list-style-type: none"> Long training time required Tuning is required to determine which kernel is optimal for non-linear SVMs 	Yes	Sensitive	Robust to outliers	<ul style="list-style-type: none"> Classification Regression 	Supervised	<ul style="list-style-type: none"> Stock market forecasting Value at risk determination
Linear Regression	<ul style="list-style-type: none"> Linear relation between features and target Residuals are normally distributed Homoscedasticity 	<ul style="list-style-type: none"> Interpretability Little tuning 	<ul style="list-style-type: none"> Correlated features may affect performance Extensive feature engineering required 	Yes	Sensitive	Sensitive	Regression	Supervised	<ul style="list-style-type: none"> Sales forecasting House pricing
Logistic Regression	<ul style="list-style-type: none"> Linear relation between features and the log odds Residuals are normally distributed Homoscedasticity 	<ul style="list-style-type: none"> Interpretability Little tuning 	<ul style="list-style-type: none"> Correlated features may affect performance Extensive feature engineering required 	Yes	Sensitive	Potentially sensitive	Classification	Supervised	<ul style="list-style-type: none"> Risk Assessment Fraud Prevention
Classification and Regression Trees	None	<ul style="list-style-type: none"> Interpretability Render feature importance Less data pre-processing required 	<ul style="list-style-type: none"> Do not predict a continuous output (for regression) It does not predict beyond the range of the response values in the training data. Overfits 	No	Some implementations do not need missing data imputation. The one in Scikit-learn does	Robust to outliers	<ul style="list-style-type: none"> Classification Regression 	Supervised	<ul style="list-style-type: none"> Risk Assessment Fraud Prevention
Random Forests	None	<ul style="list-style-type: none"> Interpretability Render feature importance Less data pre-processing required Do not overfit (in theory) Good performance /accuracy Robust to noise Little if any parameter tuning required Apt for almost any machine learning problem 	<ul style="list-style-type: none"> Do not predict a continuous output (for regression) It does not predict beyond the range of the response values in the training data Biased towards categorical variables with several categories Biased in multiclass problems toward more frequent classes 	No	Some implementations do not need missing data imputation. The one in Scikit-learn does.	Robust to outliers	<ul style="list-style-type: none"> Classification Regression 	Supervised	<ul style="list-style-type: none"> Credit Risk Assessment Predict breakdown of mechanical parts (automobile industry). Assess probability of developing a chronic disease (healthcare) Predicting the average number of social media shares
Gradient Boosted Trees	None	<ul style="list-style-type: none"> Great performance Apt for almost any machine learning problem It can approximate most non-linear functions 	<ul style="list-style-type: none"> Prone to overfit Needs some parameter tuning 	No	Some implementations do not need missing data imputation (e.g. xgboost). The one in Scikit-learn does.	Robust to outliers	<ul style="list-style-type: none"> Classification Regression 	Supervised	<ul style="list-style-type: none"> Same as Random Forests
K-nearest neighbours	None	<ul style="list-style-type: none"> Good performance 	<ul style="list-style-type: none"> Slow when predicting Susceptible to high dimension (lots of features) 	Yes	Sensitive	Robust to outliers	<ul style="list-style-type: none"> Classification Regression 	Supervised	<ul style="list-style-type: none"> Gene expression Protein-protein interaction Content retrieval (of webpages for example)
AdaBoost	None	<ul style="list-style-type: none"> It doesn't overfit easily Few parameters to tune 	<ul style="list-style-type: none"> Can be sensitive to noise and outliers 	No	Can handle	Sensitive	<ul style="list-style-type: none"> Classification Regression 	Supervised	<ul style="list-style-type: none"> Same as Random Forests, less used however, as xgboost and lightGBMs are more popular implementations of gradient boosted machines
Neural Networks	None	<ul style="list-style-type: none"> Can approximate any function Great Performance 	<ul style="list-style-type: none"> Long training time Several parameters to tune, including neuronal architecture Prone to overfit Little interpretability 	Yes	Sensitive	Can handle outliers, and it affects performance if they are too many	<ul style="list-style-type: none"> Classification Regression 	Supervised	<ul style="list-style-type: none"> Image analysis Forecasting Text analysis
K-Means Clustering	<ul style="list-style-type: none"> clusters are spherical clusters are of similar size 	<ul style="list-style-type: none"> Fast training 	<ul style="list-style-type: none"> Need to determine k, the number of clusters Sensitive to initial points and local optima 	Yes	In the Scikit-learn implementation, missing data needs to be imputed	Sensitive	<ul style="list-style-type: none"> Segmentation 	Unsupervised	<ul style="list-style-type: none"> Customer segmentation Outlier detection
Hierarchical clustering		<ul style="list-style-type: none"> No a priori information about the number of clusters required 	<ul style="list-style-type: none"> Final number of clusters to be decided by the scientist Slow training 	Yes	Sensitive	Sensitive	<ul style="list-style-type: none"> Segmentation 	Unsupervised	<ul style="list-style-type: none"> Customer segmentation Gene analyses
PCA	<ul style="list-style-type: none"> Correlation among features 	<ul style="list-style-type: none"> Captures most of the variance in a smaller number of features 	<ul style="list-style-type: none"> Number of principal components that explain most of the variance to be determined by the user 	Yes	Sensitive	Sensitive	Reducing feature space to train machine learning models	Unsupervised	<ul style="list-style-type: none"> Creating few, informative, variables from tons of data