|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Advantages** | **Disadvantages** | **Comments** |
| **Bagging** | * Robust against outliers and noise * Decrease the variance | * Slow when complex * Lack of transparency in underlying model * Sensitive to bias. | * Ensemble method Family * Predict bankruptcy |
| **Gradient boosting Classifier** | * Robustness to outliers. * Trees are build sequentially, which can improve over the previous trees. * Prone to overfitting unless tree depth and learning rate are controlled correctly * Fast training without sacrificing accuracy, * can handle different types of predictor variables (numerical, categorical) (heterogeneous features) * Accommodate missing data * Predictive power * Feature-importance vector. | * scalability * since is sequential it can hardly be parallelized * slow in some cases * Cannot compute Conditional class probabilities * Long sequential computation times. | * Ensemble method Family * Costume churn, predict costumer loss. (Salford systems use Gradient boosting classifier) * This mode tends to arrive at somewhat better results that other ensemble methods. * Compare to bagging will be interesting. * Higgs Boson Discovery. The large hadron collider dataset * Ranking websites * Ecology |
| **Random Forest**  **Classifier** | * Efficiently in large data sets * Applicable to both regression and classification problems * Is not parametric, therefore no formal distribution assumption. * Can handle highly non-linear interactions and classification boundaries. * highly accurate classifier. * Stability. if you change the data a little, the individual trees may change, but the forest is relatively stable, because is the combination of many trees. * Maintains accuracy when a large proportion of data is missing. * Gives estimates of what variables are important in the classification * Less likely to overfit than a decision tree * Generates an internal unbiased estimate of generalization error. * Provides an experimental way to detect variable interactions | * Difficult to interpret * Slow to evaluation * If the data includes groups of correlated features of equal relevance for the output variable, then small groups are favoured over large groups | * Ensemble method Family * Video classification for YouTube (decide which video is appropriate or not) * Improves DTs model by reducing overfitting without losing the correctness of the outputs. * Random Forest is a collection of DTs with a small max-depth, to avoid overfitting. * Because to the number of DTs in the forest the error is not going to increase. * Random Forest is a way to reduce bias-variance trade-off in DTs. * Xbox Kinect is used for real time human pose recognition |
| **AdaBoost**  **Classifier** | * Can be used by any type of data, textual, numeric, discrete. * Can be combined with any other learning algorithm * Less prone to overfitting * Simple to implement * Fast, versatile * Agnostic to the classifier | * Sensitive to noisy data and outliers. * The performance depends on data and weak learner. * Weak classifiers too complex leads to overfitting or low margins. | * Face detection, text classification * Binary classification where the model needs to classify if is a face or is a background image * Pre-processing is important. * Ensemble method Family boosting type * During the training, it continuously gives more weight to misclassified labels to allow the classifier to focus on the harder cases which increases the overall model performance |
| **Logistic Regression** | * Low Variance * Probability for outcomes * Robust to noise * Can also be used in big data scenarios. | * Can hardly handle categorical features. * High bias * You must assume the features are roughly linear and the problem is linearly separable. * Limited to capture complex features in the data, when are not linear. | * Medical outcomes (survival studies) * Social science (treatment effects) |
| **Support Vector Machines (SVM)** | * Works well in complex domains where there is not a clear margin of separation. * Perform well with non-linear boundary (depends of the kernel used) * Handle high dimensional data well * Best suited for problems with complex domains where there are clear margins of data separation. * Separation planes through custom kernels. | * Don’t perform well in larger data sets, because the training time happens to be cubic to the size of the data set. * Needs fine tune the parameters * Don’t work well with noise data. So, where the classes are very overlapping, you must count independent evidence. * Susceptible to overfitting when the data has noisy or overlaps. * Long train in large data sets | * Performs like Logistic Regression when no linear separation. The main reason to use SVM instead LR is because your problem might not be linearly separable. Use a SVM with a nonlinear Kernel example (RBF). * Text classification, Image recognition * Handwritten digit identification * Protein identification * Image recognition * Writing / digit recognition |
| **Stochastic Gradient Descent Classifier (SGDC)** |  | * Sensitive to feature scaling * Require several hyperparameters such as the regularization parameter and the number of iterations. | * Text classification and natural language processing. |
| **K-Nearest Neighbours (KNeighbors)** | * The computation cost is very high since the algorithm hast to compute distance to all training samples which leads to the curse of dimensionality * KNN is not parametric * Is not influenced by noise in the data * Every decision is based on locality * Easy to implement | * If you increase the number of features the computational cost increase exponentially * It’s hard to find what K distance function should use without experimentation. Therefore, are hard to interpret it. * The query time of KNNs is higher than the training time, since is a lazy learner. | * Used when you need to find similar items by calculating the distance function. * Used in recommender system * It’s a lazy algorithm |
| **Decision Trees** | * Can analyse both numerical and categorical data * Non-parametric * Work fast if is a simple estructure. | * Tends to overfit with many features but we can pick the optimal max-depth to avoid the problem | * Used in astronomy * Complexity of O(mnlg) where m is the number of features, and n is the number of rows. |
| **Gaussian Naïve Bayes (GaussianNB)** | * Computationally fast * Simple to implement * Works well with high dimensions * You need less training data, because converge quickly * Good for few categories variables | * Relies on independence assumption and will perform badly if this assumption is not met * If the model encounters unseen feature-label combination (not trained before). It will incorrectly estimate likelihood as 0 which can cause it to incorrectly classify the label. | * Text classification |