Machine Learning Algorithms Summary Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithm | Binary Classification | Multiclass Classification | Regression | Advantages | Disadvantages |
| Logistic regression  glm(family = "binomial") | Checkmark with solid fill |  |  | Very common /understood  Can calculate feature importance (which predictors are the most predictive) | Parameter estimates can be unstable when a lot of separation between classes  Lower performance compared to other models when predictors are normally distributed in each of the classes  Limited to binary classification (can use multinomial logistic regression for multiclass classification) |
| Naive Bayes  naiveBayes() | Checkmark with solid fill | Checkmark with solid fill |  | Not as strong of assumptions as QDA and LDA  Reduction in variance | Can be biased |
| kNN  knn3() | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Nonparametric (can lead to better performance) | The sample size needs to be much larger than the number of predictors  Does not indicate which predictors are important |
| QDA  qda() | Checkmark with solid fill | Checkmark with solid fill |  | Better performance over LDA when covariance structures differ across classes | Need a lot of data and computing resources due to high number of parameters  Performance will suffer if covariance structure is similar across classes (this method assumes they are not)  Strong assumptions about mean and variance distributions |
| LDA  lda() | Checkmark with solid fill | Checkmark with solid fill |  | Computationally efficient  Outperforms QDA when covariance structures are similar across classes  Needs less data than QDA | Will not outperform QDA if covariance structures differ across classes  Strong assumptions about mean and variance distributions |
| Decision trees  tree() or rpart() | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Interpretability  Outperforms linear models when relationship between outcome and predictors is complex and non-linear | High variance |
| Random Forest  randomForest() | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | High performance  Can calculate feature importance (which predictors are the most predictive)  Outperforms linear models when relationship between outcome and predictors is complex and non-linear | Less interpretability |

Machine Learning Steps (Classification; binary or categorical outcome)

1. Split data into training and test sets

- We have been doing a 50/50 split in class but in practice the most common split is 60-80% training set

2. Fit model using only the training set

3. Use the predict function to predict probabilities for the test set only

4. Convert the predicted probabilities into a class prediction (0 or 1 for binary classification; 0, 1, 2, ... , K for K classes)

5. Calculate performance metrics using the confusionMatrix function (accuracy, sensitivity, specificity for binary classification; overall accuracy and class accuracy for each class for multiclass classification)

Machine Learning Steps (Regression; continuous outcome)

1. Split data into training and test sets

- We have been doing a 50/50 split in class but in practice the most common split is 60-80% training set

2. Fit model using only the training set

3. Use the predict function to predict values for the outcome using the test set only (will be continuous numbers; not probabilities)

4. Calculate performance metrics (mean square error (MSE), mean absolute error (MAE), etc.)