Machine Learning Models in R

	Algorithm	R-package	Function	Prediction
Unsupervised Analysis	**Unsupervised Learning**			
	k-means	base	kmean(data)	model$cluster
	Hierarchical Clustering	base	hclust(d=dist(as.matrix(data)))	cutree(model, k)
	Gaussian mixture models	mclust	Mclust(data, k)	model$classification
	DBSCAN	dbscan	dbscan(data,eps=0.5, MinPts = 10)	model$cluster
	Principal Component Analysis (PCA)	base	prcomp(as.matrix(data))	model$x
	Calculating the optimal number of clusters			
	NbClust	NbClust	NbClust(iris[,1:4], distance = "euclidean", min.nc=2, max.nc=8, method = "complete")	Voting
	External Evaluation of the clusters			
	Jaccard	clusteval	cluster_similarity(class, model$cluster, similarity="jaccard", method="independence")	None
	Rand	clusteval	cluster_similarity(class, model$cluster, similarity="rand", method="independence")	None
	Internal evaluation of clusters			
	Dunn	clValid	dunn(distance = dist(as.matrix(data, method = "euclidean"), clusters = model$cluster)	None
	Silhuete	clues	get_Silhouette(as.matrix(data, model$cluster, disMethod = "Euclidean")	None

	Algorithm	R-package	Function	Prediction
Supervised Analysis	**Regression & Classification Learners**			
	Linear regression	base	lm(y ~ x1+x2+...+xn, data)	predict(model, newdata)
	Logistic regression	base	glm(y ~ x1+x2+...+xn, data,family="binomial")	predict(model, newdata, type="response")
	Ridge (L2 Regularization)	glmnet	glmnet(x, y_train, nlambda = 25, **alpha = 0**, family = 'gaussian', lambda = lambdas)	predict(model, s = 25, newx = newdata)
	LASSO (L1 Regularization)	glmnet	glmnet(x, y_train, nlambda = 25, **alpha = 1**, family = 'gaussian', lambda = lambdas)	predict(model, s = 25, newx = newdata)
	ElasticNet (L1/L2 Regularization)	glmnet	glmnet(x, y_train, nlambda = 25, **alpha = 0.5**, family = 'gaussian', lambda = lambdas)	predict(model, s = 25, newx = newdata)
	Poison regression	base	glm(y ~ x1+x2+...+xn, data,family="poisson")	predict(model, newdata, type="response")
	Naive Bayes	e1071	naiveBayes(y ~ x1+x2+...+xn, data)	predict(model, newdata, type="raw")
	k-Nearst Neighborhood (kNN)	class	knn(train, test, cl, k = 3, prob=TRUE)	knn(train, test, cl, k = 3, prob=TRUE)
	Support Vector Machine (SVM)	liquidSVM	svm(y ~ x1+x2+...+xn, data)	predict(model, newdata)
	Classification Trees			
	CART	tree	tree(y ~ x1+x2+...+xn, data)	predict(model, newdata, type="class")
		rpart	rpart(y ~ x1+x2+...+xn, data)	predict(model, newdata, type="class")
	C4.5	RWeka	J48(y ~ x1+x2+...+xn, data)	predict(model, newdata, type="class")
	C5.0	C50	C5.0(y ~ x1+x2+...+xn, data)	predict(model, newdata, type="class")
	Conditional Trees	party; partykit	ctree(y ~ x1+x2+...+xn, data)	predict(model, newdata)
	Ensemble Learners - Bagging			
	Random Forest	ranger	ranger(y ~ x1+x2+...+xn, data)	predict(model, newdata)
	Ensemble Learners - Boosting			
	Ada Boost	fastAdaboost	adaboost(y ~ x1+x2+...+xn, data, iter_num)	predict(model, newdata)
	eXtreme Gradient Boosting	xgboost	xgboost(data = data, label = y, max_depth = 2, eta=1,nrounds=2, objective="binary:logistic")	predict(model, newdata)

Machine Learning Models in Python

Algorithm	R-package	Function	Prediction
Unsupervised Analysis			
Unsupervised Learning			
k-means	sklearn.cluster	mod=KMeans(n_cluster=k, random_state=0).fit(X)	mod.labels_mod.predict(X)
Hierarchical Clustering	sklearn.cluster	mod = AgglomerativeClustering(n_clusters=k, affinity='euclidian', linkage='ward').fit_predict(X)	mod.labels_mod.predict(X)
Gaussian mixture models	sklearn.mixture	mod = GMM(n_components=k).fit(X)	mod.predict(X)
DBSCAN	sklearn.cluster	mod = DBSCAN(eps=3, min_samples = 10).fit(X)	mod.labels_mod.predict(X)
Principal Component Analysis (PCA)	sklearn.decomposition	mod = PCA(n_components=k).fit(X)	mod.explained_variance_ratio_ mod.singular_values_
External Evaluation of the clusters			
Jaccard	sklearn.metrics	metrics.jaccard_score(y,y_hat)	None
Rand	sklearn.metrics	metrics.adjusted_rand_score(y,y_hat)	None
Internal evaluation of clusters			
Dunn	jqmcvi.base	base.dunn(clusters)	None
Silhuete	sklearn.metrics	metrics.silhouette_score(X,clusters)	None

Supervised Analysis

Algorithm	Libraries	Function	Prediction
Regression & Classification Learners			
Linear regression	sklearn.linear_model	mod = LinearRegression ().fit(X,y)	mod.predict(X)
Logistic regression	sklearn.linear_model	mod = LogisticRegression ().fit(X,y)	mod.predict(X)
Ridge (L2 Regularization)	sklearn.linear_model	mod = Ridge(alpha=alpha,normalize=True).fit(X,y)	mod.predict(X)
LASSO (L1 Regularization)	sklearn.linear_model	mod = Lasso(alpha=alpha,normalize=True, max_iter=1e5).fit(X,y)	mod.predict(X)
ElasticNet (L1/L2 Regularization)	sklearn.linear_model	mod = ElasticNet(random_state=0).fit(X,y)	mod.predict(X)
Poison regression	---	---	---
Naive Bayes	sklearn.naive_bayes	mod = GaussianNB().fit(X,y) mod = MultinomialNB().fit(X,y)	mod.predict(X)
Support Vector Machine (SVM)	sklearn.svm	mod = svm.SVC(C=1, gamma=0.1).fit(X,y) mod = svm.SVR(C=1, gamma=0.1).fit(X,y)	mod.predict(X)
k-Nearest Neighbors (kNN)	sklearn.neighbors	mod = KNeighborClassifier(n_neighbors).fit(X,y) mod = KNeighborRegressor(n_neighbors).fit(X,y)	mod.predict(X)
Classification Trees			
CART	sklearn.tree	mod = DecisionTreeClassifier(random_state=2) mod = DecisionTreeRegressor(random_state=2) mod.fit(X,y)	mod.predict(X)
Ensemble Learners - Bagging			
Random Forest	sklearn.ensemble	mod = RandomForestClassifier(n_estimator=10, max_depth=2, random_state=2).fit(X,y) mod = RandomForestRegressor(n_estimator=10, max_depth=2, random_state=2).fit(X,y)	mod.predict(X)
Ensemble Learners - Boosting			
Ada Boost	sklearn.ensemble	mod = AdaBoostClassifier(random_state=1).fit(X,y) mod = AdaBoostRegressor(random_state=1).fit(X,y)	mod.predict(X)
eXtreme Gradient Boosting	xgboost	dtrain = xgboost.DMatrix(np.array(X), label=y) param = { 'max_depth':2, 'eta':1, 'objective':"binary:logistic"} mod = xgboost(param, dtrain, num_rounds)	dtest = xgboost.DMatrix(np.array(X)) mod.predict(dtest)