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$cluster)	None
	Silhouette	clues	get_Silhouette(as.matrix(data, model$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")
	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")
	Support Vector Machine (SVM)	e1071	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	randomForest	randomForest(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	Library	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_
	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_
	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	jaccard_similarity_score(y_true, y_pred)	None
	Rand	sklearn.metrics	adjusted_rand_score(y_true, y_pred)	None
	Internal evaluation of clusters			
	Davies-Bouldin	sklearn.metrics	davies_bouldin_score(X, labels)	None
	Silhouette	sklearn.metrics	silhouette_score(X, labels, sample_size, random_state=0, metric="euclidean")	None

	Algorithm	Libraries	Function	Prediction
Supervised Analysis	**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)
	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)
			mod = svm.SVR(C=1, gamma=0.1)
			mod.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)
			mod = RandomForestRegressor(n_estimator=10, max_depth=2, random_state=2)
			mod.fit(X,y)
				mod.predict(X)

	Ensemble Learners - Boosting			

	Ada Boost	sklearn.ensemble	mod = AdaBoostClassifier(random_state=1)
			mod = AdaBoostRegressor(random_state=1)
			mod.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)