Cheatography

Machine Learning in R and Python Cheat Sheet

by Jingyi Feng (jenniferfjy) via cheatography.com/169260/cs/35433/

Introduction

This cheat sheet provides a comparison between basic data processing technique as well as machine learning models in both R and Python.

Documentations

https://scikit-learn.org/stable/auto_examples/index.html

https://seaborn.pydata.org/

https://cran.r-project.org/web/packages/rpart/index.html

https://cran.r-project.org/web/packages/caret/index.html

https://cran.r-project.org/web/packages/randomForest/index.html

https://www.rdocumentation.org/packages/stats/versions/3.6.2

Load dataset in R

library(datasets)	Import packages
data(iris)	Load dataset
head(iris)	Look up the first 6 rows of the dataset
summary(iris)	Get summary statistics of each columns
names(iris)	Get the column names

Data preprocessing in R

scaling = preProcess(data, method = c('center', 'scale'))	Create scaling based on data
data_scaled = predict(scaling, data)	Apply scaling to data
train_partition = createDataPartition(y, p = 0.8, list = FALSE)	Balanced splitting based on the outcome (80/20 split)
data_train = data[train_partition,]	Split data into train and test sets
data_test = data[-train_partition,]	Split data into train and test sets

Supervised learning models in R

model = lm(data, y ~ x)	Simple linear regression
model = lm(data, y ~ x1 + x2 + x3)	Multiple linear regression
summary(model)	Print summary statistics from linear model
predictions = predict(object, newdata)	Make prediction based on the model object
model = glm(data, y ~ x1 + x2 + x3, family = 'binomial')	Logistic regression

Supervised learning models in R (cont)

model = svm(data, y ~ x1 + x2 + x3, params)	Support vector machines (SVM)
model = rpart(data, y ~ x1 + x2 + x3, params)	Decision trees
model = randomForest(data, y ~ x1 + x2 + x3, params)	Random forest
data_xgb = xgb.DMatrix(data, label)	Transform the data into DMatrix format
model = xgb.train(data_xgb, label, params)	Gradient boosting models
predictions = knn(train, test, cl, params)	k-NN with labels cl and parameters (e.g., number of neighbors)

Unsupervised learning models

model = kmeans(x, params)	K-Means clustering
model = prcomp(x, params)	Principal components analysis (PCA)

Model performance in R

RMSE(pred, actual)	Root mean square error
R2(pred, actual, form = 'traditional')	Proportion of the variance explained by the model
mean(actual == pred)	Accuracy (how accurate positive predictions are)
confusionMatrix(actual, pred)	Confusion matrix
auc(actual, pred)	Area under the ROC curve
f1Score(actual, pred)	Harmonic mean of precision and recall

Data visualization in R

geom_point(x, y, color, size, fill, alpha)	Scatter plot
geom_line(x, y, color, size, fill, alpha, linetype)	Line plot
geom_bar(x, y, color, size, fill, alpha)	Bar chart
geom_boxplot(x, y, color)	Box plot
geom_tile(x, y, color, fill)	Heatmap

Import file in Python

import pandas as pd	Import package
df = pd.read_csv()	Read csv files
df.head(n)	Look up the first n rows of the dataset
df.describe()	Get summary statistics of each columns
df.columns	Get column names

Data Processing in Python

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)	Split the dataset into training (80%) and test (20%) sets
scaler = StandardScaler()	Standardize features by removing the mean and scaling to unit variance
X_train = scaler.fit_transform(X_train)	Fit and transform scalar on X_train
X_test = scaler.transform(X_test)	Transform X_test

Supervised learning models in Python

model = LinearRegression()	Linear regression
model.fit(X_train, y_train)	Fit linear model
model.predict(X_test)	Predict using the linear model
LogisticRegression().fit(X_train, y_train)	Logistic regression
LinearSVC.fit(X_train, y_train)	Train primal SVM
SVC().fit(X_train, y_train)	Train dual SVM
DecisionTreeClassifier().fit(X_train, y_train)	Decision tree classifier
RandomForestClassifier().fit(X_train, y_train)	Random forest classifier
GradientBoostingClassifier().fit(X_train, y_train)	Gradient boosting for classification
XGBClassifier().fit(X_train, y_train)	XGboost classifier
KNeighborsClassifier().fit(X_train, y_train)	k-NN

Unsupervised learning models

KMeans().fit(X)	K-Means clustering
PCA().fit(X)	Principal component analysis (PCA)

Model performance in Python

metrics.mean_squared_error(y_true, y_pred, squared=False)	Root mean squared error
metrics.r2_score(y_true, y_pred)	Proportion of the variance explained by the model
metrics.confusion_matrix(y_true, y_pred)	Confusion matrix
metrics.accuracy_score(y_true, y_pred)	Accuracy classification score
metrics.roc_auc_score()	Compute ROC-AUC from prediction scores
f1_score(y_true, y_pred, average='macro')	Harmonic mean of the precision and recall

Data visualization in Python

sns.scatterplot(x, y, hue, size)	Scatter plot
sns.lineplot(x, y, hue, size)	Line plot
sns.barplot(x, y, hue)	Bar chart
sns.boxplot(x, y, hue)	Box plot
sns.heatmap(data, linecolor, linewidth)	Heatmap