Reminder Precision und Recall:

Precision: TP/(TP+FP)

Recall: TP/(TP+FN)

GINI-Koefficient:

* Methode um Ungleichheit von Werten einer Häufigkeitsverteilung zu bestimmen. Wird verwendet um die Qualität eines binären Klassifikators vorherzusagen
* 0 bedeutet perfekte Gleichheit (oder einen nutzlosen Klassifikator) während ein Gini-Index von 1 maximale Ungleichheit (also einen perfekten Klassifikator) wiederspiegelt.
* Basiert theoretisch auf der Lorenz-Kurve (TP-Rate (y-Achse) als Funktion der Perzentile einer Population (x-Achse))
* GINI-Index ist unabhängig vom Modell und hängt nur von der Lorenz-Kurve ab

Absolute MCC (Matthews Correlation Coefficient):

* der MCC gibt ein Verfahren, das misst, wie gut ein binärer Klassifikator wirklich positive, falsch positive, sowie wirklich und falsch negative vorhersagt.
* ist ein Korrelationskoeffizient, da es vorhersagt wie korreliert die wirklichen und die Vorhergesagten Werte sind
* Wert von 1 = perfekter Klassifikator, -1 = sagt immer falschen Wert voraus, 0 sagt, dass der Klassifikator nicht besser als zufälliges Raten ist

F1-Score:

* wie gut werden positive Fälle vorhergesagt? (harmonisches Mittel von Precision und Recall)
* Score von 1: Perfekte Precision und Recall – kein positiver Fall als negativ vorhergesagt
* ist eine der beiden Metriken sehr klein, geht der Wert gegen null

F0.5-Score:

* gewichtetes harmonisches Mittel von Precision und Recall
* gewichtet Precision höher als Recall
* angewandt, wenn: FPs sind schlimmer als FNs (z.B. mehr kosten, wenn falsch vorhergesagt, dass positiv)

F2-Score:

* wie F0.5-Score, aber mehr Gewicht zu recall (mehr Penalty für False Negatives)

Accuracy:

* Rate korrekter Vorhersagen als Prozent aller vorhersagen
* Nicht geeignet für unbalancierte Klassen, da immer majority class vorhergesagt wird

Logloss:

* Wie weit ist der vorhergesagte Wert (unkalibrierte vorhergesagte Werte) vom wirklichen Wert entfernt?
  + Beispiel: Sagt ein Modell hohe Wahrscheinlichkeiten für die richtige Klasse voraus, wie z.B. .80 oder ist die Range sehr hoch und es geht Richtung raten (z.B. nur .50)?
* Je niedriger der Logloss, desto besser (0 würde bedeuten, dass das Modell korrekt eine Wahrscheinlichkeit von 0% oder 100% vorhersagt)

AUC:

* Evaluiert, wie gut ein binäres Klassifikationsmodell TPs von FPs unterscheiden kann
  + AUC von 1 ist ein perfekter Klassifikator, während AUC von .5 eher Richtung „Random Guessing“ geht.
* Im Regelfall schlechter für unbalancierte Klassen geeignet

AUCPR (Area under the Precision-Recall Curve):

* Wie gut kann ein Modell zwischen Precision-Recall-Paaren unterscheiden  
  -> True Negatives sind nicht so schlimm, wird empfohlen für sehr unbalancierte Daten

Metriken, die gut für unbalancierte Klassen sind:

* MCC (Matthews Correlation Coefficient)
* AUCPR (Area under the Precision-Recall Curve)

**STOP-Metriken von h2o:**

* Missklassifikationsrate:
  + Modell muss Missklassifikationsrate um einen gewissen Wert erhöhen, um weiter zu iterieren
  + Anzahl inkorrekt klassifiziert durch Gesamtanzahl
* Lift Top Group
  + 1% mit höchsten vorhergesagten Werten muss verbessert werden
  + Lift = Rate korrekt klassifizierter positiver Observationen zur Anzahl der absoluten Anzahl positiver Observationen in einer Gruppe
* Deviance (Abweichung)
  + wenn die Abweichung nicht mehr besser wird, soll es aufhören zu verbessern
* Mean-Per-Class-Error:
  + Stoppt, wenn sich diese Rate nicht mehr verbessert
* Andere Möglichkeiten:
  + Logloss, Mean-Square-Error (MSE), Root-MSE (RMSE), MAE, RMSLE und AUC

**Model Performance Graphen:**

* Confusion Matrix
  + gibt TP, FN, TN, FP aus
  + Spalten sind vorhersagen, Reihen sind wirkliche Werte
* Variable Importance (VarImp)
  + statistische Signifikanz jeder Variable in den Daten im Zusammenhang mit ihrem Effekt auf das Modell
  + Prozentsatz gibt dabei an, wie wichtig die Variable, skaliert auf 100% ist.
* ROC-Curve
  + Ratio von TP zu FP
* Hit Ratio: (h2o.hit\_ratio\_table)
  + Anzahl der richtigen vorhersagen aus kompletten Anzahl der vorhersagen
* Standardized Coefficient Magnitudes (h2o.std\_coef\_plot)
  + Beziehung eines spezifischen Features zur abhängigen Variable
  + ähnlich wie VarImp-Plot
  + positiver Zusammenhang = orange, negativ = blau
  + nur bei GLMs anwendbar
* Partial Dependence Plots (h2o.partialPlot)
  + Graphische Repräsentation des marginalen Effekts einer Variable auf die Klassenwahrscheinlichkeit (bei Klassifikation) oder Vorhersage (bei einer Regression).
  + Nur möglich bei Modellen, die nur numerische Werte enthalten
* Leaf Node Assignment (h2o.predict\_leaf\_node\_assignment)
  + gibt an, in welchem Modell, welche Reihe wo landet Anhand der Entscheidungskriterien
* Predict Contributions (h2o.predict\_contributions)
  + Reine Vorhersage von Baum-basierten Modellen ist eine Summe der Prädiktoren der individuellen Bäume, bevor die wirklichen vorhersagen erhalten werden
* Predict Stage Probablities (h2o.staged\_predict\_proba)
  + kann nur bei GBM verwendet werden
  + gibt aus, wie die vorhergesagten Wahrscheinlichkeiten zu jedem Stadium des Modells war
* Predict Feature Frequency (h2o.feature\_frequencies)
  + nur unterstützt für GBM, DRF, und IF
  + Anzahl, wie oft ein Feature bei Vorhersagen in einem Baummodell verwendet wurde

**Checkpoints**

* In DRF, GBM und Deep Learning – baut bei referenzierten Modellen weiter und kein neues Modell! -> evolutionärer Algorithmus
* ntrees und epochs sollen angegeben werden, wie wenn Modell von neu gebaut werden würde -> müssen immer größer als die orginalen Werte sein
* Dateb müssen immer gleich aufgebaut sein
* mehr Info: <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/checkpointing-models.html>

**Interaktives Interface im Webbrowser für h2o: FLOW**

Siehe <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/flow.html> für Infos.