Cvičení 27.4. – metoda RandomForest

1. nainstalovat knihovnu randomForest

```
> data(iris)
> summary(iris)
> library(randomForest)
> set.seed(71)
```

randomForest(x, y=NULL,
xtest=NULL, ytest=NULL,
ntree=500,
mtry=if (!is.null(y) && !is.factor(y))max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x))),
replace=TRUE / jestli může být vzorek vybrán vícekrát
classwt=NULL /váha jednotlivých kategorií závisle proměnné, defaulně mají všechny stejnou váhu
strata, /parametr pro stratifikovaný výběr
sampsize = if (replace) nrow(x) else ceiling(.632*nrow(x)), /parametr pro stratifikovaný výběr
nodesize = if (!is.null(y) && !is.factor(y)) 5 else 1, /minimání počet vzorků v terminálním uzlu, defaulně 1 pro klasifikaci a 5 pro regresi
maxnodes = NULL, /maximální počet terminálních uzlů stromu
importance=FALSE,
localImp=FALSE, /významnost každého vzorku
nPerm=1, / zatím pouze pro regresi, počet iterací kdy jsou OOB vzorky permutovány pro výpočet importance proměnných
proximity, / výpočet matice těsnosti
oob.prox=proximity, / matice těsnosti pouze pro OOB vzorky
norm.votes=TRUE, / vyjádřeno jako podíl, jinak přímo počet
do.trace=FALSE, / ukáže výstupy procesu hledání
keep.forest=!is.null(y) && is.null(xtest), corr.bias=FALSE, keep.inbag=FALSE, ...) / FALSE – výsledek lesa nebude uložen ve finálním výstupu.

1. vytvoříme les

```
> les1 <- randomForest(Species ~ ., data=iris, importance=TRUE, proximity=TRUE) /Species je závisle proměnná, použijeme všechny prediktory,výsledek uložíme do souboru les1
>print(les1) / výsledky z náhodného lesa
```

2. závislost počtu stromů v lese na celkové chybě lesa

```
> plot(randomForest(Species ~ ., data=iris, keep.forest=FALSE, ntree=100))
> hist(treesize(les1)) /histogram počtu terminálních uzlů
```

3. Nastavení počtu proměnných v modelu (parametr *mtry*)

```
>tuneRF(x, y, mtryStart, ntreeTry=50, stepFactor=2, improve=0.05, trace=TRUE, plot=TRUE, doBest=FALSE)
>mtryles1 <- tuneRF(iris[,1:4], iris[,5], stepFactor=1.5)
```

mtry – defoultní hodnoty jsou (sqrt(p)) pro klasifikaci, kde p je počet prediktorů a regresi (p/3)
improve – relativní zlepšení OOB chyby musí být větší než stanovená hodnota, aby probíhalo další vyhledávání optimálního počtu prediktorů (mtry)
trace – ukáže výstupy procesu hledání

3. významnost proměnných
>importance(les1, type=2)
Type- 1=mean decrease in accuracy, 2=mean decrease in node impurity
>round(importance(les1), 2) / významnost proměnných zaokrouhlená na 2 desetinná místa

4. graf významnosti proměnných

>varImpPlot(x, sort=TRUE, n.var=min(30, nrow(x$importance)))
>varImpPlot(les1)

5. proměnné použité v RF

varUsed(x, by.tree=FALSE, count=TRUE)
count- celková frekvence použití proměnné v lese
by.tree – rozepsáno pro jednotlivé stromy
>varUsed(randomForest(Species~., iris, ntree=100))
>varUsed(randomForest(Species~., iris, ntree=50),by.tree=TRUE, count=TRUE)

6. doplnění chybějících hodnot pomocí proximity

rfImpute(x, y, iter=5, ntree=300, ...)
>iris.chybej <- iris
>set.seed(111)
>for (i in 1:4) iris.chybej[sample(150, sample(20)), i] <- NA /přidáme prázdné hodnoty
>iris.chybej
>set.seed(222)
>iris.dopln <- rfImpute(Species ~ ., iris.chybej)
>set.seed(333)
>irisLes2 <- randomForest(Species ~ ., iris.dopln)
>print(irisLes2)

7. Predikce nových vzorků pomocí RF

predict(object, newdata, type="response", norm.votes=TRUE, predict.all=FALSE, proximity=FALSE...)
type=response, prob a vote – zvolíme si ty výstupu, a) predikované hodnoty, b) matice pravděpodobností ke kategoriím, C) matice hlasování stromů (u klasifikace stejné s prob)
norm.votes=TRUE – vyjádřeno jako podíl, jinak přímo počet
>set.seed(111)
>ind <- sample(2, nrow(iris), replace = TRUE, prob=c(0.8, 0.2)) / vytvoříme proměnnou s hodnotami 1 a 2, prob udává poměr indexů ve sloupci)
>irisLes3<- randomForest(Species ~ ., data=iris[ind == 1,]) / první soubor použijeme pro vytvoření lesa
>iris.pred <- predict(irisLes3, iris[ind == 2,]) / druhý soubor pro predikci nových hodnot
>iris.pred / výsledek predikce
>table(observed = iris[ind==2, "Species"], predicted = iris.pred) /výsledek predikce v tabulce
>predict(irisLes3, iris[ind == 2,], predict.all=TRUE) / predikce jednotlivými stromy
>predict(irisLes3, iris[ind == 2,], proximity=TRUE) / matice měření těsnosti (proximity)
>iris.pred <- predict(irisLes3, iris[ind == 2,], type="prob")
>iris.pred
>iris.pred <- predict(irisLes3, iris[ind == 2,], norm.votes = FALSE)
>iris.pred

8. Efekt proměnných na predikci

partialPlot(x, pred.data, x.var, which.class,w, plot = TRUE, add = FALSE,n.pt = min(length(unique(pred.data[, xname])), 51), rug = TRUE, xlab=deparse(substitute(x.var)), ylab="",main=paste("Partial Dependence on", deparse(substitute(x.var))),...)

```
>data(iris)
>set.seed(543)
>iris.rf <- randomForest(Species~., iris)
>partialPlot(iris.rf, iris, Petal.Width, "versicolor")
```

9. Měření odlehlosti

```
>set.seed(1)
>iris.rf <- randomForest(iris[,-5], iris[,5], proximity=TRUE) /musíme mít spočítanou matici proximity
>plot(outlier(iris.rf), type="h",col=c("red", "green", "blue")[as.numeric(iris$Species)]) / as.numeric -seřazeno podle kategorií
```

10. prototypy kategorií

classCenter(x, label, prox, nNbr = min(table(label))-1)
nNbr / počet nejbližších sousedů

```
>iris.rf <- randomForest(iris[,-5], iris[,5], prox=TRUE)
>iris.p <- classCenter(iris[,-5], iris[,5], iris.rf$prox)
>plot(iris[,3], iris[,4], pch=21, xlab=names(iris)[3], ylab=names(iris)[4], bg=c("red", "blue", "green")[as.numeric(factor(iris$Species))], main="Iris Data with Prototypes") /pch – tvar a výplň znaků
>points(iris.p[,3], iris.p[,4], pch=21, cex=2, bg=c("red", "blue", "green")) /cex – velikost textu oproti defaulnímu nastavení
```

(Pch pch=19: solid circle, pch=20: bullet (smaller circle), pch=21: filled circle, pch=22: filled square, pch=23: filled diamond, pch=24: filled triangle point-up, pch=25: filled triangle

11. výběr jednotlivého stromu

```
getTree(randomForest(iris[,-5], iris[,5], ntree=10), 3, labelVar=TRUE) / výběr 3. stromu
```

10. graf scaling coordinates matice těsnosti MDS

MDSplot(rf, fac, k=2, palette=NULL, pch=20, ...)

```
>set.seed(1)
>data(iris)
>iris.rf <- randomForest(Species ~ ., iris, proximity=TRUE, keep.forest=FALSE)
>MDSplot(iris.rf, iris$Species)
>MDSplot(iris.rf, iris$Species, palette=rep(1, 3), pch=as.numeric(iris$Species)) /použití symbolů místo barev
```

Datový soubor v package MASS

```
> data(shuttle,package="MASS")
> summary(shuttle)
```

Jde o reálná data, představující shromážděná doporučení expertní komise při přípravě letů raketoplánu. Tato doporučení popisují okolnosti, za kterých by měla posádka nechat přistávací manévr na počítači (faktor use má pak hodnotu ***auto***) a za kterých má být přistání provedeno ručně (use s hodnotou ***noauto***). Rozhodování je ovlivněno stabilitou raketoplánu na předem naplánované dráze sestupu (stability), velikostí (error) a směrem (sign) odchylky od dráhy, směrem větru v místě přistání (wind – čelní resp. v zádech), sílou větru (magn) a také viditelností v oblasti přistání (vis). Ačkoliv poskytnutá data popisují v podstatě všechny možné kombinace podmínek (28 = 256 kombinací), pro jejich efektivní použití, a také pro ověření jejich konzistentnosti, byl tento návod převeden do soustavy pravidel