Introduction to Mushroom Learning

Machine Learing in mushroom context

by Victoria Szabo, Cornelia Gruber, Andreas Klaß and Felix Langer

LMU – Introduction to Machine Learning – students project – SoSe 2020

Structure

1.) Goals

2.) Data set

3.) Method Decisions

4.) Implementation with *mlr3*

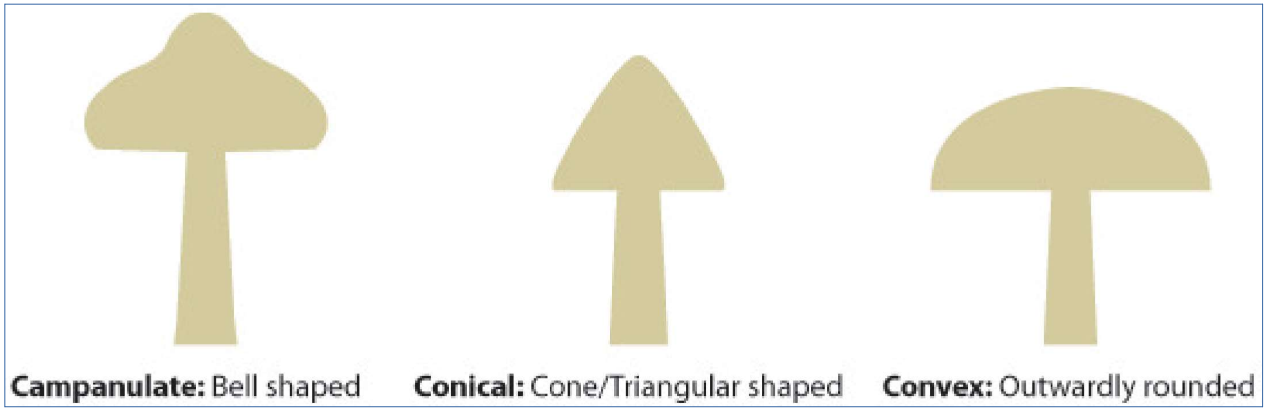
5.) Results

Goals

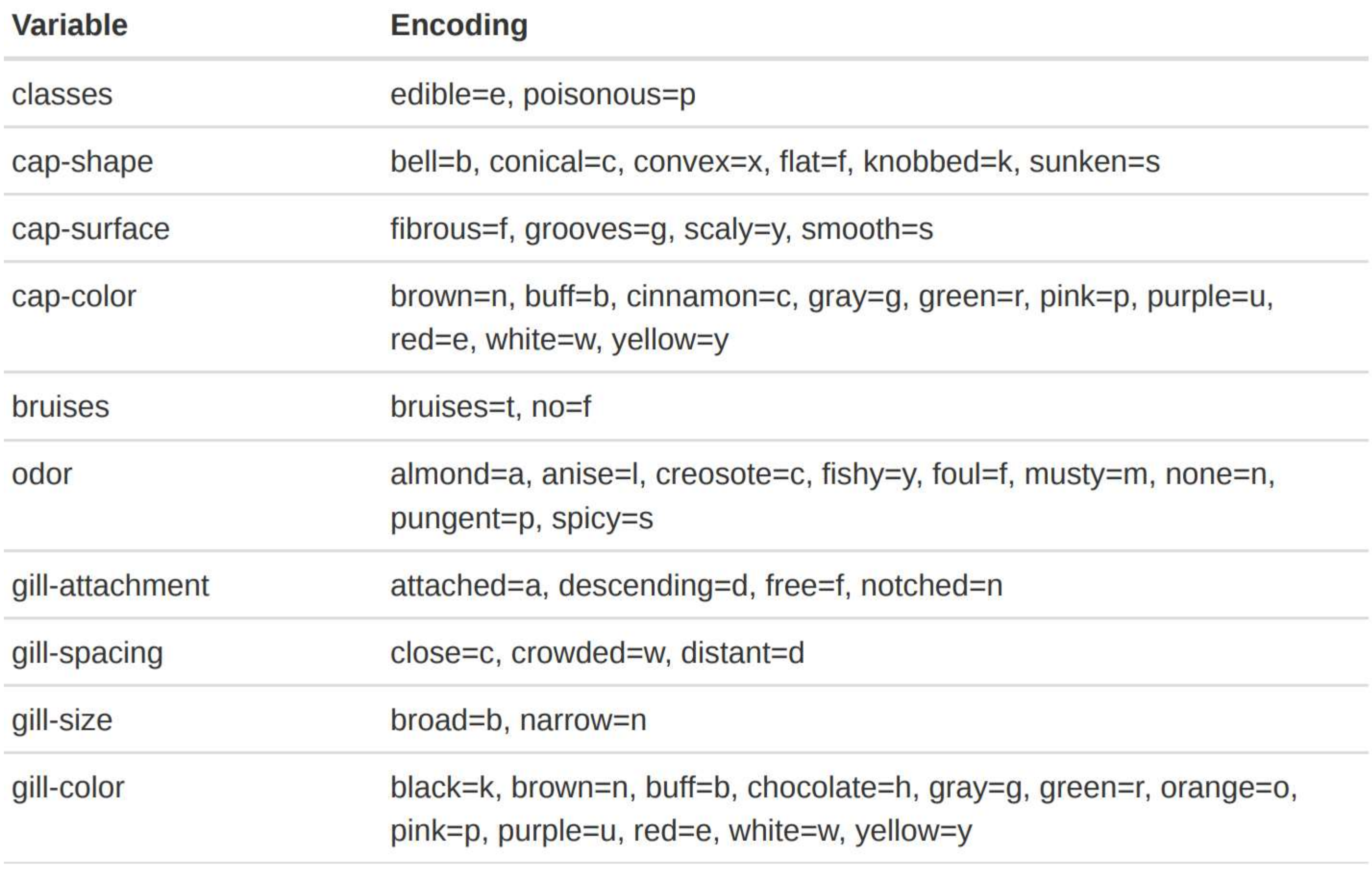
- Classification of mushrooms: edible or poisnous
- Using Machine Learing methods
- Using *mlr3*-Package

Data set

- 8124 observations
- Binary target variable (edible or poisonous)
- 22 nominal features (characteristics of each mushroom)


Campanulate: Bell shaped
Conical: Cone/Triangular shaped
Convex: Outwardly rounded

Data set



Variable	Encoding
classes	edible=e, poisonous=p
cap-shape	bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
cap-surface	fibrous=f, grooves=g, scaly=y, smooth=s
cap-color	brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
bruises	bruises=t, no=f
odor	almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
gill-attachment	attached=a, descending=d, free=f, notched=n
gill-spacing	close=c, crowded=w, distant=d
gill-size	broad=b, narrow=n
gill-color	black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y

...

Structure

1.) Goals

2.) Data set

3.) Method Decisions

4.) Implementation with *mlr3*

5.) Results

Method decisions

- 6 classification methods:
  - Featureless
  - Naive Bayes
  - Decision Tree
  - Random Forest
  - KNN
  - Logistic Regression

Method decisions

- 6 classification methods:
  - Featureless
  - Naive Bayes
  - Decision Tree
  - Random Forest → + Tuning *mtry*
  - KNN → + Tuning *k*
  - Logistic Regression

Method decisions

- Generalisation Error & Hyperparameter tuning
  - Nested Resampling
    - Inner loop: 5-fold CV (Hyperparameter tuning)
    - Outer loop: 10-fold CV (final GE)
  - Optimization criteria: AUC
  - Further measures: False Positive Rate

Structure

1.) Goals

2.) Data set

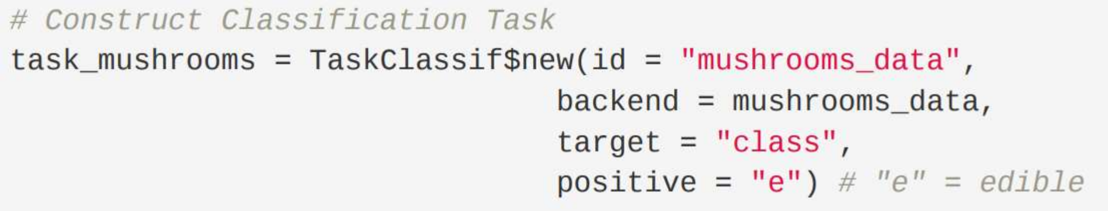
3.) Method Decisions

4.) Implementation with *mlr3*

5.) Results

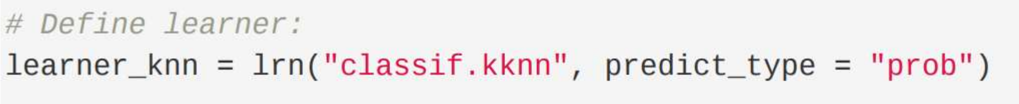
Implementation with *mlr3*

- Task

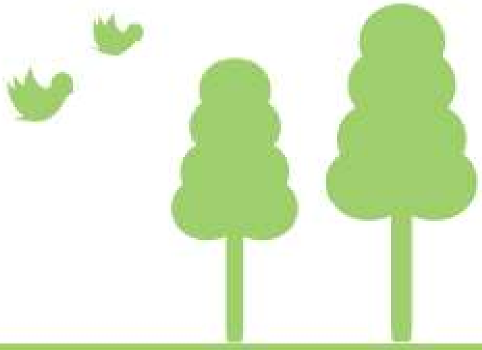


```r
# Construct Classification Task
task_mushrooms = TaskClassif$new(id = "mushrooms_data",
                                 backend = mushrooms_data,
                                 target = "class",
                                 positive = "e") # "e" = edible
```

- Learner

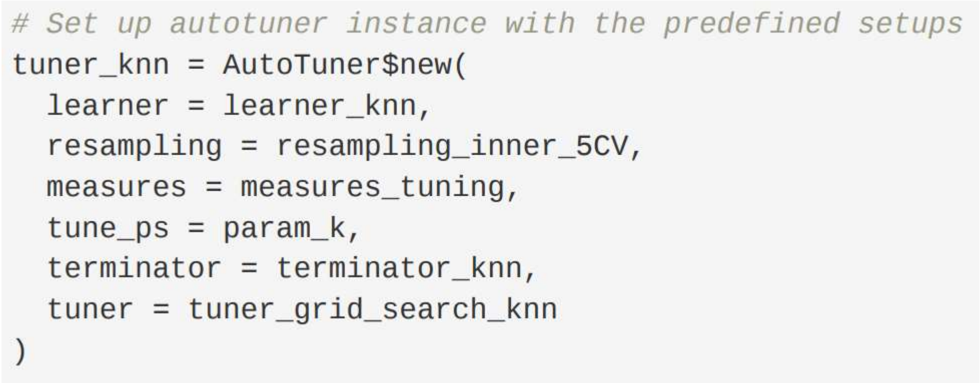


```r
# Define learner:
learner_knn = lrn("classif.kknn", predict_type = "prob")
```



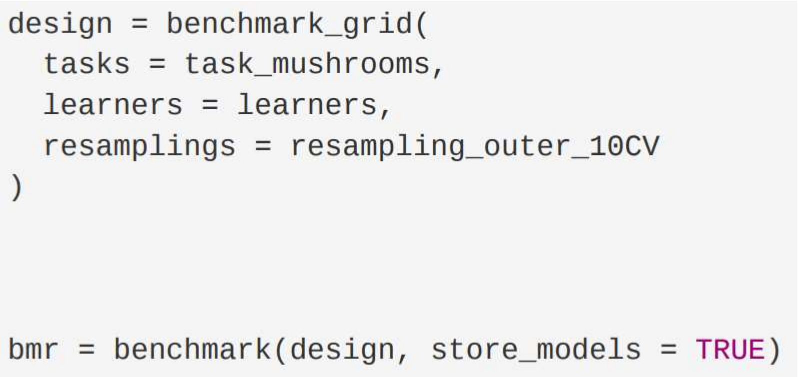
Implementation with *mlr3*

- Tuner



```r
# Set up autotuner instance with the predefined setups
tuner_knn = AutoTuner$new(
  learner = learner_knn,
  resampling = resampling_inner_5CV,
  measures = measures_tuning,
  tune_ps = param_k,
  terminator = terminator_knn,
  tuner = tuner_grid_search_knn
)
```

- Benchmark



```r
design = benchmark_grid(
  tasks = task_mushrooms,
  learners = learners,
  resamplings = resampling_outer_10CV
)



bmr = benchmark(design, store_models = TRUE)
```

Structure

1.) Goals

2.) Data set

3.) Method Decisions

4.) Implementation with *mlr3*

5.) Results

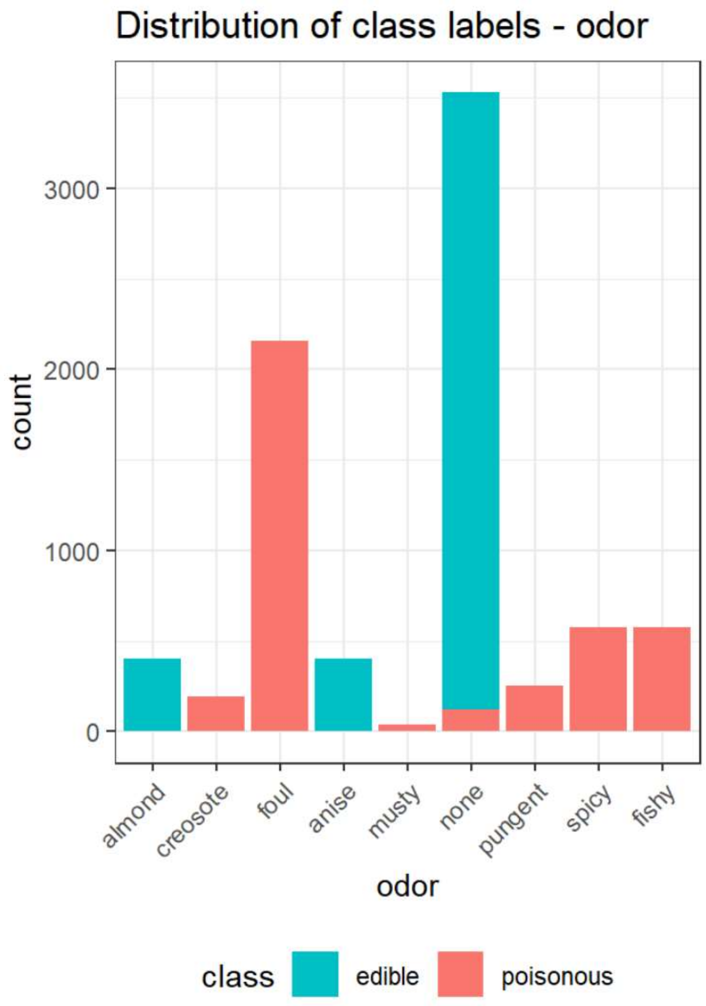
Results

- Performance measures:

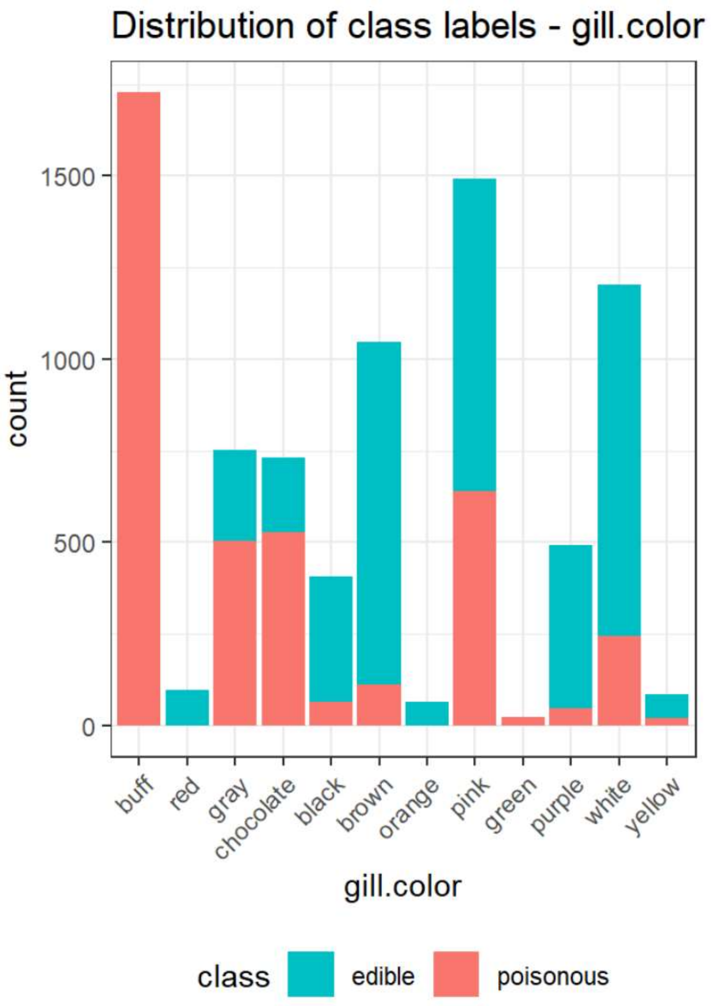
Method	AUC	FPR
Featureless	0.5000	1.000
Naive Bayes	0.9960	0.1156
Dicision Tree	0.9939	0.0122
Random Forest	1.0000	0.0000
KNN	1.0000	0.0003
Logistic Regression	1.0000	0.0000

- Warning messages with logistic regression

Results



Distribution of class labels - odor



Distribution of class labels - gill.color

Results

- Warning messages with logistic regression
- Performance measures:

Method	**AUC**	**FPR**
Featureless	0.5000	1.000
Naive Bayes	0.9960	0.1156
Dicision Tree	0.9939	0.0122
Random Forest	**1.0000**	**0.0000**
KNN	1.0000	0.0003
Logistic Regression	1.0000	0.0000

Thank you!!!