

# Introduction to Mushroom Learning

Machine Learning in mushroom context

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



# Structure

- 1.) Goals
- 2.) Data set
- 3.) Method Decisions
- 4.) Implementation with *mlr3*
- 5.) Results



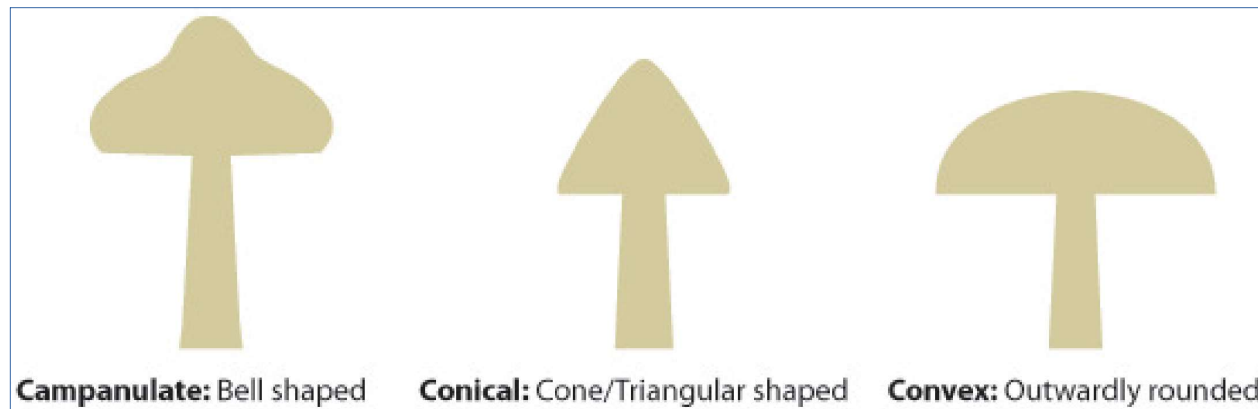
# Goals

- Classification of mushrooms: edible or poisonous
- Using Machine Learning methods
- Using *mlr3*-Package



# Data set

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



# 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

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

- Learner

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



# Implementation with *mlr3*

- Tuner

```
# 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

```
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:

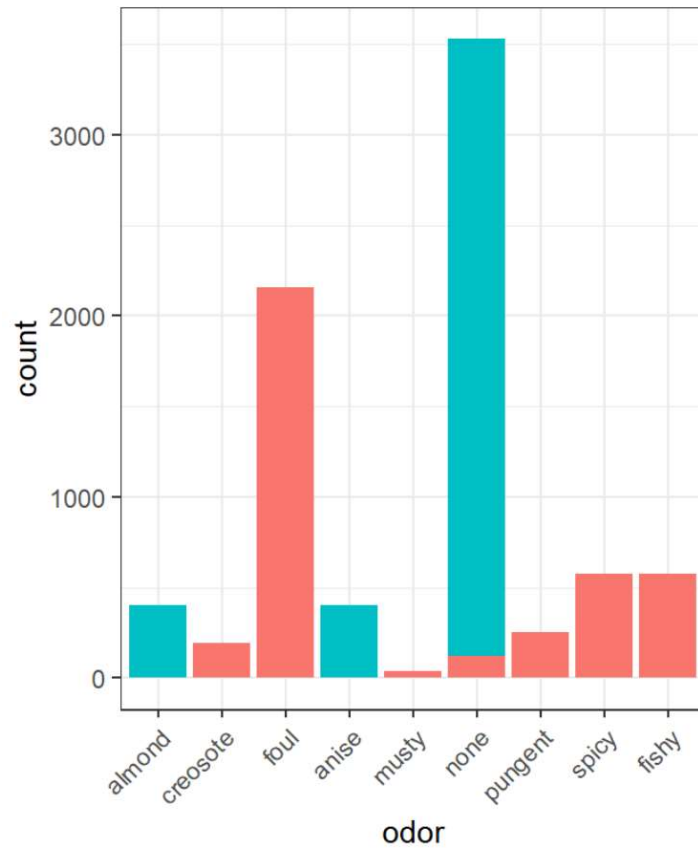
Method	AUC	FPR
Featureless	0.5000	1.000
Naive Bayes	0.9960	0.1156
Dicision Tree	0.9939	0.0122
Random Forest	1.0000	0.0000
KNN	1.0000	0.0003
Logistic Regression	1.0000	0.0000

- Warning messages with logistic regression



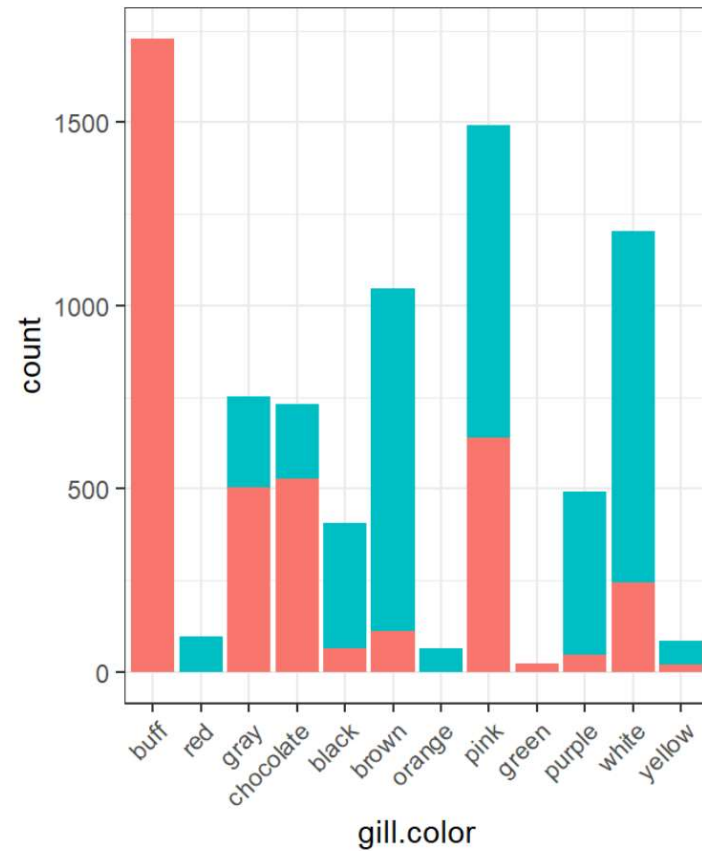
# Results

Distribution of class labels - odor



class edible poisonous

Distribution of class labels - gill.color



class edible poisonous



# Results

- Performance measures:

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

- Warning messages with logistic regression



Thank you!!!

