🏁 Wrap-up quiz 3

This quiz requires some programming to be answered.

Load the dataset file named `penguins.csv` with the following command:

```
import pandas as pd


penguins = pd.read_csv("../datasets/penguins.csv")

columns = ["Body Mass (g)", "Flipper Length (mm)", "Culmen Length (mm)"]
target_name = "Species"

# Remove lines with missing values for the columns of interest
penguins_non_missing = penguins[columns + [target_name]].dropna()

data = penguins_non_missing[columns]
target = penguins_non_missing[target_name]
```

`penguins` is a pandas dataframe. The column “Species” contains the target variable. We extract the numerical columns that quantify some attributes of such animals and our goal is to predict their species based on those attributes stored in the dataframe named `data`.

Inspect the loaded data to select the correct assertions:

Question

Inspect the target variable and select the correct assertions from the following proposals.

- a) The problem to be solved is a regression problem
- b) The problem to be solved is a binary classification problem (exactly 2 possible classes)
- c) The problem to be solved is a multiclass classification problem (more than 2 possible classes)

Select a single answer

Hint: `target.nunique()`is a helpful method to answer to this question.

Question

Inspect the statistics of the target and individual features to select the correct statements.

- a) The proportions of the class counts are balanced: there are approximately the same number of rows for each class
- b) The proportions of the class counts are imbalanced: some classes have more than twice as many rows than others
- c) The input features have similar scales (ranges of values)

Select all answers that apply

Hint: `data.describe()`, and `target.value_counts()` are methods that are helpful to answer to this question.

Let’s now consider the following pipeline:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline


model = Pipeline(steps=[
    ("preprocessor", StandardScaler()),
    ("classifier", KNeighborsClassifier(n_neighbors=5)),
])
```

Question

Evaluate the pipeline using stratified 10-fold cross-validation with the `balanced-accuracy` scoring metric to choose the correct statement in the list below.

You can use:

- `sklearn.model_selection.cross_validate` to perform the cross-validation routine;
- provide an integer `10` to the parameter `cv` of `cross_validate` to use the cross-validation with 10 folds;
- provide the string `"balanced_accuracy"` to the parameter `scoring` of `cross_validate`.
- a) The average cross-validated test balanced accuracy of the above pipeline is between 0.9 and 1.0
- b) The average cross-validated test balanced accuracy of the above pipeline is between 0.8 and 0.9
- c) The average cross-validated test balanced accuracy of the above pipeline is between 0.5 and 0.8

Select a single answer

Question

Repeat the evaluation by setting the parameters in order to select the correct statements in the list below. We recall that you can use `model.get_params()` to list the parameters of the pipeline and use `model.set_params(param_name=param_value)` to update them. Remember that one way to compare two models is comparing the cross-validation test scores of both models fold-to-fold, i.e. counting the number of folds where one model has a better test score than the other

- a) Looking at the individual cross-validation scores, using a model with `n_neighbors=5` is substantially better (at least 7 of the cross-validations scores are better) than a model with `n_neighbors=51`
- b) Looking at the individual cross-validation scores, using a model with `n_neighbors=5` is substantially better (at least 7 of the cross-validations scores are better) than a model with `n_neighbors=101`
- c) Looking at the individual cross-validation scores, a 5 nearest neighbors using a `StandardScaler` is substantially better (at least 7 of the cross-validations scores are better) than a 5 nearest neighbors using the raw features (without scaling).

Select all answers that apply

We will now study the impact of different preprocessors defined in the list below:

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import QuantileTransformer
from sklearn.preprocessing import PowerTransformer


all_preprocessors = [
    None,
    StandardScaler(),
    MinMaxScaler(),
    QuantileTransformer(n_quantiles=100),
    PowerTransformer(method="box-cox"),
]
```

The Box-Cox method is common preprocessing strategy for positive values. The other preprocessors work both for any kind of numerical features. If you are curious to read the details about those method, please feel free to read them up in the preprocessing chapter of the scikit-learn user guide but this is not required to answer the quiz questions.

Question

Use `sklearn.model_selection.GridSearchCV` to study the impact of the choice of the preprocessor and the number of neighbors on the stratified 10-fold cross-validated `balanced_accuracy` metric. We want to study the `n_neighbors` in the range `[5, 51, 101]` and `preprocessor` in the range `all_preprocessors`.

Which of the following statements hold:

- a) Looking at the individual cross-validation scores, the best ranked model using a `StandardScaler` is substantially better (at least 7 of the cross-validations scores are better) than using any other preprocessor
- b) Using any of the preprocessors has always a better ranking than using no preprocessor, irrespective of the value `of n_neighbors`
- c) Looking at the individual cross-validation scores, the model with `n_neighbors=5` and `StandardScaler` is substantially better (at least 7 of the cross-validations scores are better) than the model with `n_neighbors=51` and `StandardScaler`
- d) Looking at the individual cross-validation scores, the model with `n_neighbors=51` and `StandardScaler` is substantially better (at least 7 of the cross-validations scores are better) than the model with `n_neighbors=101` and `StandardScaler`

Select all answers that apply

Hint: pass `{"preprocessor": all_preprocessors, "classifier__n_neighbors": [5, 51, 101]}` for the `param_grid` argument to the `GridSearchCV` class.

Question

Evaluate the generalization performance of the best models found in each fold using nested cross-validation. Set `return_estimator=True` and `cv=10` for the outer loop. The scoring metric must be the `balanced-accuracy`. The mean generalization performance is

- a) better than 0.97
- b) between 0.92 and 0.97
- c) below 0.92

Select a single answer

Question

Explore the set of best parameters that the different grid search models found in each fold of the outer cross-validation. Remember that you can access them with the `best_params_` attribute of the estimator. Select all the statements that are true.

- a) The tuned number of nearest neighbors is stable across folds
- b) The tuned number of nearest neighbors changes often across folds
- c) The optimal scaler is stable across folds
- d) The optimal scaler changes often across folds

Select all answers that apply

Hint: it is important to pass `return_estimator=True` to the `cross_validate` function to be able to introspect trained model saved in the `"estimator"` field of the CV results. If you forgot to do for the previous question, please re-run the cross-validation with that option enabled.
