🏁 Wrap-up quiz 4

This quiz requires some programming to be answered.

Open the dataset `ames_housing_no_missing.csv` with the following command:

```
import pandas as pd

ames_housing = pd.read_csv("../datasets/ames_housing_no_missing.csv")
target_name = "SalePrice"
data = ames_housing.drop(columns=target_name)
target = ames_housing[target_name]
```

`ames_housing` is a pandas dataframe. The column “SalePrice” contains the target variable.

To simplify this exercise, we will only used the numerical features defined below:

```
numerical_features = [
    "LotFrontage", "LotArea", "MasVnrArea", "BsmtFinSF1", "BsmtFinSF2",
    "BsmtUnfSF", "TotalBsmtSF", "1stFlrSF", "2ndFlrSF", "LowQualFinSF",
    "GrLivArea", "BedroomAbvGr", "KitchenAbvGr", "TotRmsAbvGrd", "Fireplaces",
    "GarageCars", "GarageArea", "WoodDeckSF", "OpenPorchSF", "EnclosedPorch",
    "3SsnPorch", "ScreenPorch", "PoolArea", "MiscVal",
]

data_numerical = data[numerical_features]
```

Start by fitting a ridge regressor (`sklearn.linear_model.Ridge`) fixing the penalty `alpha` to 0 to not regularize the model. Use a 10-fold cross-validation and pass the argument `return_estimator=True` in `sklearn.model_selection.cross_validate` to access all fitted estimators fitted on each fold. As discussed in the previous notebooks, use an instance of `sklearn.preprocessing.StandardScaler` to scale the data before passing it to the regressor.

Question

How large is the largest absolute value of the weight (coefficient) in this trained model?

- a) Lower than 1.0 (1e0)
- b) Between 1.0 (1e0) and 100,000.0 (1e5)
- c) Larger than 100,000.0 (1e5)

Select a single answer

Hint: Note that the estimator fitted in each fold of the cross-validation procedure is a pipeline object. To access the coefficients of the `Ridge` model at the last position in a pipeline object, you can use the expression `pipeline[-1].coef_` for each pipeline object fitted in the cross-validation procedure. The `-1` notation is a negative index meaning “last position”.

Repeat the same experiment by fitting a ridge regressor (`sklearn.linear_model.Ridge`) with the default parameter (i.e. `alpha=1.0`).

Question

How large is the largest absolute value of the weight (coefficient) in this trained model?

- a) Lower than 1.0
- b) Between 1.0 and 100,000.0
- c) Larger than 100,000.0

Select a single answer

Question

What are the two most important features used by the ridge regressor? You can make a box-plot of the coefficients across all folds to get a good insight.

- a) `"MiscVal"` and `"BsmtFinSF1"`
- b) `"GarageCars"` and `"GrLivArea"`
- c) `"TotalBsmtSF"` and `"GarageCars"`

Select a single answer

Remove the feature `"GarageArea"` from the dataset and repeat the previous experiment.

Question

What is the impact on the weights of removing `"GarageArea"` from the dataset?

- a) None
- b) Completely changes the order of the most important features
- c) Decreases the standard deviation (across CV folds) of the `"GarageCars"` coefficient

Select all answers that apply

Question

What is the main reason for observing the previous impact on the most important weight(s)?

- a) Both garage features are correlated and are carrying similar information
- b) Removing the “GarageArea” feature reduces the noise in the dataset
- c) Just some random effects

Select a single answer

Now, we will search for the regularization strength that maximizes the generalization performance of our predictive model. Fit a **`sklearn.linear_model.RidgeCV`** instead of a `Ridge` regressor on the numerical data without the `"GarageArea"` column. Pass `alphas=np.logspace(-3, 3, num=101)` to explore the effect of changing the regularization strength.

Question

What is the effect of tuning `alpha` on the variability of the weights of the feature `"GarageCars"`? Remember that the variability can be assessed by computing the standard deviation.

- a) The variability does not change after tuning `alpha`
- b) The variability decreased after tuning alpha
- c) The variability increased after tuning alpha

Select a single answer

Check the parameter `alpha_` (the regularization strength) for the different ridge regressors obtained on each fold.

Question

In which range does `alpha_` fall into for most folds?

- a) between 0.1 and 1
- b) between 1 and 10
- c) between 10 and 100
- d) between 100 and 1000

Select a single answer

Now, we will tackle a classification problem instead of a regression problem. Load the Adult Census dataset with the following snippet of code and we will work only with **numerical features**.

```
adult_census = pd.read_csv("../datasets/adult-census.csv")
target = adult_census["class"]
data = adult_census.select_dtypes(["integer", "floating"])
data = data.drop(columns=["education-num"])
```

Question

How many numerical features are present in the dataset contained in the variable `data`?

- a) 3
- b) 4
- c) 5

Select a single answer

Question

Compare the generalization performance using the accuracy of the two following predictive models using a 10-fold cross-validation:

- a linear model composed of a `StandardScaler` and a `LogisticRegression`
- a `DummyClassifier` predicting the most frequent class

By comparing the cross-validation test scores of both models fold-to-fold, count the number of times the linear model has a better test score than the dummy classifier Select the range which this number belongs to:

- a) [0, 3]: the linear model is substantially worse than the dummy classifier
- b) [4, 6]: both models are almost equivalent
- c) [7, 10]: the linear model is substantially better than the dummy classifier

Select a single answer

Question

What is the most important feature seen by the logistic regression?

- a) `"age"`
- b) `"capital-gain"`
- c) `"capital-loss"`
- d) `"hours-per-week"`

Select a single answer

Now, we will work with **both numerical and categorical features**. You can load Adult Census with the following snippet:

```
adult_census = pd.read_csv("../datasets/adult-census.csv")
target = adult_census["class"]
data = adult_census.drop(columns=["class", "education-num"])
```

Create a predictive model where the categorical data must be one-hot encoded, the numerical data must be scaled, and the predictor is a logistic regression classifier.

Use the same 10-fold cross-validation strategy as above to evaluate this complex pipeline.

Question

Look at the cross-validation test scores for both models and count the number of times the model using both numerical and categorical features has a better test score than the model using only numerical features. Select the range which this number belongs to:

- a) [0, 3]: the model using both numerical and categorical features is substantially worse than the model using only numerical features
- b) [4, 6]: both models are almost equivalent
- c) [7, 10]: the model using both numerical and categorical features is substantially better than the model using only numerical features

Select a single answer

For the following questions, you can use the following snippet to get the feature names after the preprocessing performed.

```
preprocessor.fit(data)
feature_names = (preprocessor.named_transformers_["onehotencoder"]
                              .get_feature_names_out(categorical_columns)).tolist()
feature_names += numerical_columns
feature_names
```

There is as many feature names as coefficients in the last step of your predictive pipeline.

Question

Which of the following pair of features is most impacting the predictions of the logistic regression classifier based on the relative magnitude of its coefficients?

- a) `"hours-per-week"` and `"native-country_Columbia"`
- b) `"workclass_?"` and `"native_country_?"`
- c) `"capital-gain"` and `"education_Doctorate"`

Select a single answer

Question

What is the effect of decreasing the `C` parameter on the coefficients?

- a) shrinking the magnitude of the weights towards zeros
- b) increasing the magnitude of the weights
- c) reducing the weights' variance
- d) increasing the weights' variance
- e) it has no influence on the weights' variance

Select all answers that apply
