

- ▶ Welcome
- ▶ Introduction: Machine Learning concepts
- ▼ **Module 1. The Predictive Modeling Pipeline**

Module overview

Tabular data exploration

Quiz M1 


Fitting a scikit-learn model on numerical data

Quiz M1 

Handling categorical data

Quiz M1 

Wrap-up quiz

Wrap-up quiz 

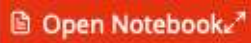
Main take-away

- ▶ Module 2. Selecting the best model
- ▶ Module 3. Hyperparameter tuning

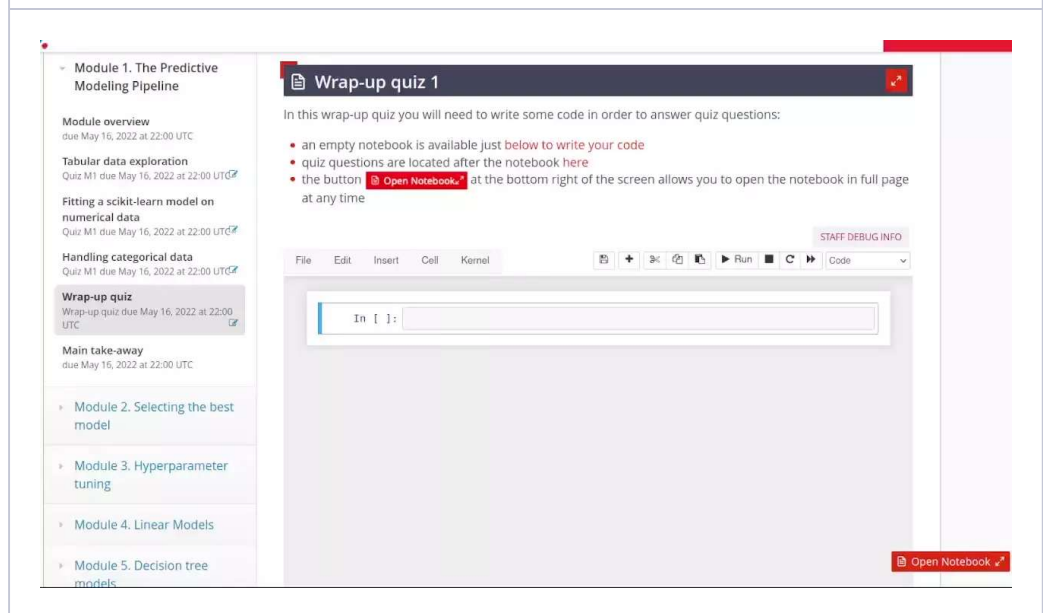
Wrap-up quiz 1



In this wrap-up quiz you will need to write some code in order to answer the quiz questions:

- an empty notebook is available just below to write your code
- quiz questions are located just after the notebook here
- the button  at the bottom right of the screen allows you to open the notebook in full page at any time

+ Demo video of the notebook user interface



 Open Notebook 

Module 1 - wrap up

- ▶ Module 5.
Decision tree
models
- ▶ Module 6.
Ensemble of
models
- ▶ Module 7.
Evaluating
model
performance
- ▶ Conclusion
- ▶ Appendix

Importing Pandas

```
In [1]: import pandas as pd
```

Loading Dataset

```
In [16]: ames_housing = pd.read_csv("../datasets/ames_housi
target_name = "SalePrice"
data, target = ames_housing.drop(columns=target_na
```

We did not encounter any regression problem yet. Therefore, we convert the regression target into a classification target to predict whether or not an house is expensive. "Expensive" is defined as a sale price greater than \$200,000.

```
In [17]: target = (target > 200_000).astype(int)
```

EDA

```
In [4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 79 columns):
#   Column                Non-Null Count  Dtype
---  -
0   MSSubClass             1460 non-null   int64
1   MSZoning               1460 non-null   object
2   LotFrontage            1460 non-null   float64
3   LotArea                1460 non-null   int64
4   Street                 1460 non-null   object
5   Alley                  1460 non-null   object
```

Powered by



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



```
target_name = "SalePrice"
data, target = ames_housing.drop(columns=target_name),
ames_housing[target_name]
target = (target > 200_000).astype(int)
```

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

We did not encounter any regression problem yet. Therefore, we convert the regression target into a classification target to predict whether or not an house is expensive. "Expensive" is defined as a sale price greater than \$200,000.

Question 1 (1/1 point)

Use the `data.info()` and `data.head()` commands to examine the columns of the dataframe. The dataset contains:

- ☐ a) only numerical features
- ☐ b) only categorical features
- ☒ c) both numerical and categorical features

You have used 1 of 1 submissions

Question 2 (1/1 point)

How many features are available to predict whether or not a house is expensive ?

- ☒ a) 79

- ☐ b) 80



You have used 1 of 1 submissions

Question 3 (1/1 point)

How many features are represented with numbers?

☐ a) 0

☒ b) 36

☐ c) 42

☐ d) 79

Hint: You can use the method `df.select_dtypes` or the function `sklearn.compose.make_column_selector` as shown in the notebook.

You have used 1 of 1 submissions

Question 4 (1/1 point)

Refer to the dataset description regarding the meaning of the dataset.

Among the following columns, which columns express a quantitative numerical value (excluding ordinal categories)?

☒ a) "LotFrontage" ✓

☒ b) "LotArea" ✓

☐ c) "OverallQual"



☒ e) "YearBuilt" ✓



Select all answers that apply

EXPLANATION

Solution: a) b) e)

"OverallQual" and "OverallCond" are ordinal categorical variables. While technically "YearBuilt" is more a date than a quantity, it is fine for machine learning models to treat it as a quantity because the year of construction is directly related to the age of the house.

You have used 1 of 2 submissions

We consider the following numerical columns:

```
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",
]
```

Question 5 (1/1 point)

Now create a predictive model that uses these numerical columns as input data. Your predictive model should be a pipeline composed of a `sklearn.preprocessing.StandardScaler` to scale these numerical data and a `sklearn.linear_model.LogisticRegression`.

What is the accuracy score obtained by 10-fold cross-validation (you can set the parameter `cv=10` when calling `cross_validate`) of this pipeline?



☐ b) ~0.7

☒ c) ~0.9

You have used 1 of 1 submissions

Question 6 (1/1 point)

Instead of solely using the numerical columns, let us build a pipeline that can process both the numerical and categorical features together as follows:

- the `numerical_features` (as defined above) should be processed as previously done with a `StandardScaler` ;
- the left-out columns should be treated as categorical variables using a `sklearn.preprocessing.OneHotEncoder` . To avoid any issue with rare categories that could only be present during the prediction, you can pass the parameter `handle_unknown="ignore"` to the `OneHotEncoder` .

What is the accuracy score obtained by 10-fold cross-validation of the pipeline using both the numerical and categorical features?

☐ a) ~0.7

☒ b) ~0.9

☐ c) ~1.0

You have used 1 of 1 submissions

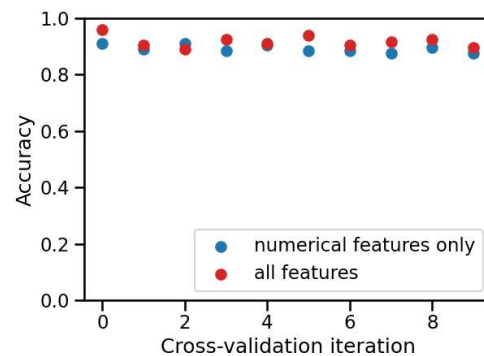
Question 7 (1/1 point)



merely by chance (e.g. when using random sampling during cross validation), and not because one model predicts systematically better than the other.

Another way is to compare 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. This provides some extra information: are some partitions of the data making the classification task particularly easy or hard for both models?

Let's visualize the second approach.



Select the true statement.

The number of folds where the model using all features perform better than the model using only numerical features lies in the range:

☐ a) [0, 3]: the model using all features is consistently worse

☐ b) [4, 6]: both models are almost equivalent

☒ c) [7, 10]: the model using all features is consistently better



EXPLANATION

solution: c)

To answer the question, we can now compare the score of each fold to investigate if this improvement is generally happening on all folds of the cross-validation:



```
indices = np.arange(len(test_score_num))
plt.scatter(
    indices, test_score_num, color="tab:blue", label="numerical
features only"
)
plt.scatter(
    indices,
    test_score_all,
    color="tab:red",
    label="all features",
)
plt.ylim((0, 1))
plt.xlabel("Cross-validation iteration")
plt.ylabel("Accuracy")
_ = plt.legend(loc="lower right")

print(
    "A model using all features is better than a"
    " model using only numerical features for"
    f" {sum(test_score_all > test_score_num)} CV iterations out of 10."
)
```

We observe that 9 times out of 10, the model based on both numerical and categorical features is better than the model that only uses numerical features.

You have used 1 of 1 submissions

YOUR EXPERIENCE

According to you, the 'Wrap-up Quiz' of this module was:

- ☐ **Too easy, I got bored**
- ☐ **Adapted to my skills**
- ☐ **Difficult but I was able to follow**
- ☐ **Too difficult**

Submit



☐ less than 30 minutes

☐ 30 min to 1 hour

☐ 1 to 2 hours

☐ 2 to 4 hours

☐ more than 4 hours

☐ I don't know


Submit

FORUM (EXTERNAL RESOURCE)



 New topic 

[Home](#) > [M1. The Predictive Modeling Pipeline](#) > [M1. Wrap-up quiz 1](#)

| Topic | Replies | Last reply |
|--|---------|------------------------------------|
|  About the M1. Wrap-up quiz 1 category brospars 27 Oct | 0 | 27 Oct brospars |

There are no more M1. Wrap-up quiz 1 topics. Ready to [start a new conversation?](#)

About...

Help and Contact

Terms of use



Terms and conditions

