

This notebook is an exercise in the [Introduction to Machine Learning](#) course. You can reference the tutorial at [this link](#).

Recap

So far, you have loaded your data and reviewed it with the following code. Run this cell to set up your coding environment where the previous step left off.

```
In [ ]: # Code you have previously used to load data
import pandas as pd

# Path of the file to read
iowa_file_path = '../input/home-data-for-ml-course/train.csv'

home_data = pd.read_csv(iowa_file_path)

# Set up code checking
from learntools.core import binder
binder.bind(globals())
from learntools.machine_learning.ex3 import *

print("Setup Complete")
```

Exercises

Step 1: Specify Prediction Target

Select the target variable, which corresponds to the sales price. Save this to a new variable called `y`. You'll need to print a list of the columns to find the name of the column you need.

```
In [ ]: # print the list of columns in the dataset to find the name of the prediction target
```

```
In [ ]: y = home_data.SalePrice  
  
step_1.check()
```

```
In [ ]: # The lines below will show you a hint or the solution.  
# step_1.hint()  
# step_1.solution()
```

Step 2: Create X

Now you will create a DataFrame called `X` holding the predictive features.

Since you want only some columns from the original data, you'll first create a list with the names of the columns you want in `X`.

You'll use just the following columns in the list (you can copy and paste the whole list to save some typing, though you'll still need to add quotes):

- * LotArea
- * YearBuilt
- * 1stFlrSF
- * 2ndFlrSF
- * FullBath
- * BedroomAbvGr
- * TotRmsAbvGrd

After you've created that list of features, use it to create the DataFrame that you'll use to fit the model.

```
In [ ]: # Create the list of features below
feature_names = ['LotArea', 'YearBuilt', '1stFlrSF', '2ndFlrSF', 'FullBath',
                 'BedroomAbvGr', 'TotRmsAbvGrd']

# select data corresponding to features in feature_names
X = home_data[feature_names]

step_2.check()
```

```
In [ ]: # step_2.hint()
# step_2.solution()
```

Review Data

Before building a model, take a quick look at **X** to verify it looks sensible

```
In [ ]: # Review data
# print description or statistics from X
print(X.describe())

# print the top few lines
print(X.head())
```

Step 3: Specify and Fit Model

Create a `DecisionTreeRegressor` and save it `iowa_model`. Ensure you've done the relevant import from `sklearn` to run this command.

Then fit the model you just created using the data in `X` and `y` that you saved above.

```
In [ ]: from sklearn.tree import DecisionTreeRegressor
#specify the model.
#For model reproducibility, set a numeric value for random_state when specifying the model
iowa_model = DecisionTreeRegressor(random_state=1)
```

```
# Fit the model
iowa_model.fit(X,y)

step_3.check()
```

```
In [ ]: # step_3.hint()
        # step_3.solution()
```

Step 4: Make Predictions

Make predictions with the model's `predict` command using `X` as the data. Save the results to a variable called `predictions`.

```
In [ ]: predictions = iowa_model.predict(X)
        print(predictions)
        step_4.check()
```

```
In [ ]: # step_4.hint()
        # step_4.solution()
```

Think About Your Results

Use the `head` method to compare the top few predictions to the actual home values (in `y`) for those same homes. Anything surprising?

```
In [ ]: # You can write code in this cell
```

It's natural to ask how accurate the model's predictions will be and how you can improve that. That will be your next step.

Keep Going

You are ready for [Model Validation](#).

Have questions or comments? Visit the [Learn Discussion forum](#) to chat with other Learners.