Machine Learning Course Home Page

This exercise will test your ability to read a data file and understand statistics about the data.

In later exercises, you will apply techniques to filter the data, build a machine learning model, and iteratively improve your model.

The course examples use data from Melbourne. To ensure you can apply these techniques on your own, you will have to apply them to a new dataset (with house prices from Iowa).

The exercises use a "notebook" coding environment. In case you are unfamiliar with notebooks, we have a 90-second intro video.

Exercises

Run the following cell to set up code-checking, which will verify your work as you go.

In [1]:

```
# Set up code checking
from learntools.core import binder
binder.bind(globals())
from learntools.machine_learning.ex2 import *
print("Setup Complete")
```

```
Setup Complete
```

Step 1: Loading Data

Read the Iowa data file into a Pandas DataFrame called `home_data` .

In [2]:

```
import pandas as pd

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

# Fill in the line below to read the file into a variable home_data
home_data = pd.read_csv(iowa_file_path)

# Call line below with no argument to check that you've loaded the data correctly
step_1.check()
```

Correct

In [3]:

```
# Lines below will give you a hint or solution code
#step_1.hint()
#step_1.solution()
```

Step 2: Review The Data

Use the command you learned to view summary statistics of the data. Then fill in variables to answer the following questions

In [4]:

```python
# Print summary statistics in next line
home_data.describe()
```

Out[4]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	Ye
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.267808	
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	30.202904	
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	

8 rows × 38 columns

In [5]:

```python
# What is the average lot size (rounded to nearest integer)?
avg_lot_size = home_data['LotArea'].mean().round(0)

# As of today, how old is the newest home (current year - the date in which it was buil
newest_home_age = 2022 - max(home_data['YearBuilt'])

# Checks your answers
step_2.check()
```

Correct

In [6]:

```python
#step_2.hint()
#step_2.solution()
```

Think About Your Data

The newest house in your data isn't that new. A few potential explanations for this:

1. They haven't built new houses where this data was collected.
2. The data was collected a long time ago. Houses built after the data publication wouldn't show up.

If the reason is explanation #1 above, does that affect your trust in the model you build with this data? What about if it is reason #2?

How could you dig into the data to see which explanation is more plausible?

Check out this **discussion thread** to see what others think or to add your ideas.

Keep Going

You are ready for **Your First Machine Learning Model.**

Machine Learning Course Home Page