Recap

Here's the code you've written so far.

In [1]:

```
# Code you have previously used to load data
import pandas as pd
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor


# 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)
# Create target object and call it y
y = home_data.SalePrice
# Create X
features = ['LotArea', 'YearBuilt', '1stFlrSF', '2ndFlrSF', 'FullBath', 'BedroomAbvGr',
'TotRmsAbvGrd']
X = home_data[features]

# Split into validation and training data
train_X, val_X, train_y, val_y = train_test_split(X, y, random_state=1)

# Specify Model
iowa_model = DecisionTreeRegressor(random_state=1)
# Fit Model
iowa_model.fit(train_X, train_y)

# Make validation predictions and calculate mean absolute error
val_predictions = iowa_model.predict(val_X)
val_mae = mean_absolute_error(val_predictions, val_y)
print("Validation MAE when not specifying max_leaf_nodes: {:,.0f}".format(val_mae))

# Using best value for max_leaf_nodes
iowa_model = DecisionTreeRegressor(max_leaf_nodes=100, random_state=1)
iowa_model.fit(train_X, train_y)
val_predictions = iowa_model.predict(val_X)
val_mae = mean_absolute_error(val_predictions, val_y)
print("Validation MAE for best value of max_leaf_nodes: {:,.0f}".format(val_mae))


# Set up code checking
from learntools.core import binder
binder.bind(globals())
from learntools.machine_learning.ex6 import *
print("\nSetup complete")
```

```
Validation MAE when not specifying max_leaf_nodes: 29,653
Validation MAE for best value of max_leaf_nodes: 27,283

Setup complete
```

Exercises

Data science isn't always this easy. But replacing the decision tree with a Random Forest is going to be an easy win.

Step 1: Use a Random Forest

In [2]:

```
from sklearn.ensemble import RandomForestRegressor

# Define the model. Set random_state to 1
rf_model = RandomForestRegressor(random_state=1)

# fit your model
rf_model.fit(train_X, train_y)

# Calculate the mean absolute error of your Random Forest model on the validation data
rf_preds = rf_model.predict(val_X)
rf_val_mae = mean_absolute_error(val_y, rf_preds)

print("Validation MAE for Random Forest Model: {}".format(rf_val_mae))

step_1.check()
```

```
Validation MAE for Random Forest Model: 22762.42931506849
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/ensemble/forest.py:248: Fut
ureWarning: The default value of n_estimators will change from 10 in versi
on 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

Correct

In [3]:

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

Hint: Review the code above with a DecisionTreeRegressor. Use the RandomForestRegressor instead

Solution:

```
rf_model = RandomForestRegressor()

# fit your model
rf_model.fit(train_X, train_y)

# Calculate the mean absolute error of your Random Forest model on the val
idation data
rf_val_predictions = rf_model.predict(val_X)
rf_val_mae = mean_absolute_error(rf_val_predictions, val_y)
```

Think about Your Results

Under what circumstances might you prefer the Decision Tree to the Random Forest, even though the Random Forest generally gives more accurate predictions? Weigh in or follow the discussion in this discussion thread.

Keep Going

So far, you have followed specific instructions at each step of your project. This helped learn key ideas and build your first model, but now you know enough to try things on your own.

Machine Learning competitions are a great way to try your own ideas and learn more as you independently navigate a machine learning project. Learn **how to submit your work to a Kaggle competition**.

Course Home Page