Introduction

Machine learning competitions are a great way to improve your data science skills and measure your progress.

In this exercise, you will create and submit predictions for a Kaggle competition. You can then improve your model (e.g. by adding features) to improve and see how you stack up to others taking this course.

The steps in this notebook are:

1. Build a Random Forest model with all of your data (**X** and **y**)
2. Read in the "test" data, which doesn't include values for the target. Predict home values in the test data with your Random Forest model.
3. Submit those predictions to the competition and see your score.
4. Optionally, come back to see if you can improve your model by adding features or changing your model. Then you can resubmit to see how that stacks up on the competition leaderboard.

Recap

Here's the code you've written so far. Start by running it again.

In [1]:
```
# Code you have previously used to load data
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
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. We changed the directory structure to simplify sub
mitting to a competition
iowa_file_path = '../input/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', 'Bedro
omAbvGr', '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_ma
e))

# Define the model. Set random_state to 1
rf_model = RandomForestRegressor(random_state=1)
rf_model.fit(train_X, train_y)
rf_val_predictions = rf_model.predict(val_X)
rf_val_mae = mean_absolute_error(rf_val_predictions, val_y)

print("Validation MAE for Random Forest Model: {:,.0f}".format(rf_val_mae))
```

```
Validation MAE when not specifying max_leaf_nodes: 29,653
Validation MAE for best value of max_leaf_nodes: 27,283
Validation MAE for Random Forest Model: 22,762
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/ensemble/forest.py:248: Future
Warning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

Creating a Model For the Competition

Build a Random Forest model and train it on all of **X** and **y**.

In [2]:
```
# To improve accuracy, create a new Random Forest model which you will train o
n all training data
rf_model_on_full_data = RandomForestRegressor()


# fit rf_model_on_full_data on all data from the
rf_model_on_full_data.fit(X, y)
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/ensemble/forest.py:248: Future
Warning: The default value of n_estimators will change from 10 in version 0.2
0 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

Out[2]:
```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
           max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=None,
           oob_score=False, random_state=None, verbose=0, warm_start=False)
```

Make Predictions

Read the file of "test" data. And apply your model to make predictions

In [ ]:
```
# path to file you will use for predictions
test_data_path = '../input/test.csv'

# read test data file using pandas
test_data = pd.read_csv(test_data_path)

# create test_X which comes from test_data but includes only the columns you u
sed for prediction.
# The list of columns is stored in a variable called features
test_X = test_data[features]

# make predictions which we will submit.
test_preds = rf_model_on_full_data.predict(test_X)

# The lines below shows you how to save your data in the format needed to scor
e it in the competition
output = pd.DataFrame({'Id': test_data.Id,
                       'SalePrice': test_preds})

output.to_csv('submission.csv', index=False)
```

Test Your Work

After filling in the code above:

1. Click the **Commit and Run** button.
2. After your code has finished running, click the small double brackets **<<** in the upper left of your screen. This brings you into view mode of the same page. You will need to scroll down to get back to these instructions.
3. Go to the output tab at top of your screen. Select the button to submit your file to the competition.
4. If you want to keep working to improve your model, select the edit button. Then you can change your model and repeat the process.

Congratulations, you've started competing in Machine Learning competitions.

Continuing Your Progress

There are many ways to improve your model, and **experimenting is a great way to learn at this point.**

The best way to improve your model is to add features. Look at the list of columns and think about what might affect home prices. Some features will cause errors because of issues like missing values or non-numeric data types.

Level 2 of this course will teach you how to handle these types of features. You will also learn to use **xgboost**, a technique giving even better accuracy than Random Forest.

Other Courses

The **Pandas course** will give you the data manipulation skills to quickly go from conceptual idea to implementation in your data science projects.

You are also ready for the **Deep Learning** course, where you will build models with better-than-human level performance at computer vision tasks.

Course Home Page

Learn Discussion Forum.