

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

Introduction

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 apply what you've learned and move up the leaderboard.

Begin by running the code cell below to set up code checking and the filepaths for the dataset.

In []:

```
# Set up code checking
from learntools.core import binder
binder.bind(globals())
from learntools.machine_learning.ex7 import *

# Set up filepaths
import os
if not os.path.exists("../input/train.csv"):
    os.symlink("../input/home-data-for-ml-course/train.csv", "../input/train.csv")
    os.symlink("../input/home-data-for-ml-course/test.csv", "../input/test.csv")
```

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

In []:

```
# Import helpful libraries
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split

# Load the data, and separate the target
iowa_file_path = '../input/train.csv'
home_data = pd.read_csv(iowa_file_path)
y = home_data.SalePrice

# Create X (After completing the exercise, you can return to modify this line!)
features = ['LotArea', 'YearBuilt', '1stFlrSF', '2ndFlrSF', 'FullBath', 'BedroomAbvGr',
            'TotRmsAbvGrd']

# Select columns corresponding to features, and preview the data
X = home_data[features]
X.head()

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

# Define a random forest model
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))
```

Train a model for the competition

The code cell above trains a Random Forest model on `train_X` and `train_y`.

Use the code cell below to build a Random Forest model and train it on all of `x` and `y`.

In []:

```
# To improve accuracy, create a new Random Forest model which you will train on all training data
rf_model_on_full_data = RandomForestRegressor(random_state=1)

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

Now, 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 used for prediction.
# The list of columns is stored in a variable called features
features = ['LotArea', 'YearBuilt', '1stFlrSF', '2ndFlrSF', 'FullBath', 'BedroomAbvGr', 'TotRmsAbvGrd']
test_X = test_data[features]

# make predictions which we will submit.
test_preds = rf_model.predict(test_X)
print(test_preds)
```

Before submitting, run a check to make sure your `test_preds` have the right format.

In []:

```
# Check your answer (To get credit for completing the exercise, you must get a "Correct" result!)
step_1.check()
# step_1.solution()
```

Generate a submission

Run the code cell below to generate a CSV file with your predictions that you can use to submit to the competition.

In []:

```
# Run the code to save predictions in the format used for competition scoring

output = pd.DataFrame({'Id': test_data.Id,
                        'SalePrice': test_preds})
output.to_csv('submission.csv', index=False)
```

Submit to the competition

To test your results, you'll need to join the competition (if you haven't already). So open a new window by clicking on [this link](#). Then click on the **Join Competition** button.

Next, follow the instructions below:

1. Begin by clicking on the **Save Version** button in the top right corner of the window. This will generate a pop-up window.
2. Ensure that the **Save and Run All** option is selected, and then click on the **Save** button.
3. This generates a window in the bottom left corner of the notebook. After it has finished running, click on the number to the right of the **Save Version** button. This pulls up a list of versions on the right of the screen. Click on the ellipsis (...) to the right of the most recent version, and select **Open in Viewer**. This brings you into view mode of the same page. You will need to scroll down to get back to these instructions.
4. Click on the **Output** tab on the right of the screen. Then, click on the file you would like to submit, and click on the blue **Submit** button to submit your results to the leaderboard.

You have now successfully submitted to the competition!

If you want to keep working to improve your performance, select the **Edit** button in the top right of the screen. Then you can change your code and repeat the process. There's a lot of room to improve, and you will climb up the leaderboard as you work.

Continue 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. To add more features to the data, revisit the first code cell, and change this line of code to include more column names:

```
features = ['LotArea', 'YearBuilt', '1stFlrSF', '2ndFlrSF', 'FullBath', 'BedroomAbvGr', 'TotRmsAbvGrd']
```

Some features will cause errors because of issues like missing values or non-numeric data types. Here is a complete list of potential columns that you might like to use, and that won't throw errors:

- 'MSSubClass'
- 'LotArea'
- 'OverallQual'
- 'OverallCond'
- 'YearBuilt'
- 'YearRemodAdd'
- 'BsmtFinSF1'
- 'BsmtFinSF2'
- 'BsmtUnfSF'
- 'TotalBsmtSF'
- '1stFlrSF'
- '2ndFlrSF'
- 'LowQualFinSF'
- 'GrLivArea'
- 'BsmtFullBath'
- 'BsmtHalfBath'
- 'FullBath'
- 'HalfBath'
- 'BedroomAbvGr'
- 'KitchenAbvGr'
- 'TotRmsAbvGrd'
- 'Fireplaces'
- 'GarageCars'
- 'GarageArea'
- 'WoodDeckSF'
- 'OpenPorchSF'

- 'EnclosedPorch'
- '3SsnPorch'
- 'ScreenPorch'
- 'PoolArea'
- 'MiscVal'
- 'MoSold'
- 'YrSold'

Look at the list of columns and think about what might affect home prices. To learn more about each of these features, take a look at the data description on the [competition page](#).

After updating the code cell above that defines the features, re-run all of the code cells to evaluate the model and generate a new submission file.

What's next?

As mentioned above, some of the features will throw an error if you try to use them to train your model. The [Intermediate Machine Learning](#) 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.

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.

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