# Exercise: Machine Learning Competitions

# 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 micro-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.

# 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

from learntools.core import \*

# Path of the file to read. We changed the directory structure to simplify submitting to a competition

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))

# 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))

# Creating a Model For the Competition

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

# 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\_on\_full\_data.fit(X,y)

# Make Predictions

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

# 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

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 how to save predictions in format used for competition scoring

# Just uncomment them.

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\*\* button.

2. After your code has finished running, click the "Open Version" button. This brings you into the "viewer mode" for your notebook. You will need to scroll down to get back to these instructions.

3. Click \*\*Output\*\* button on the left of your screen.

This will bring you to a part of the screen that looks like this:

![](https://imgur.com/a/QRHL7Uv)

Select the button to submit and you will see your score. You have now successfully submitted 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 to submit again. There's a lot of room to improve your model, and you will climb up the leaderboard as you work.

# 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.

The [Intermediate Machine Learning](https://www.kaggle.com/learn/intermediate-machine-learning) micro-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 Micro-Courses

The \*\*[Pandas Micro-Course](https://kaggle.com/Learn/Pandas)\*\* 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](https://kaggle.com/Learn/Deep-Learning)\*\* micro-course, where you will build models with better-than-human level performance at computer vision tasks.