(https://cognitiveclass.ai)

From Modeling to Evaluation

Introduction

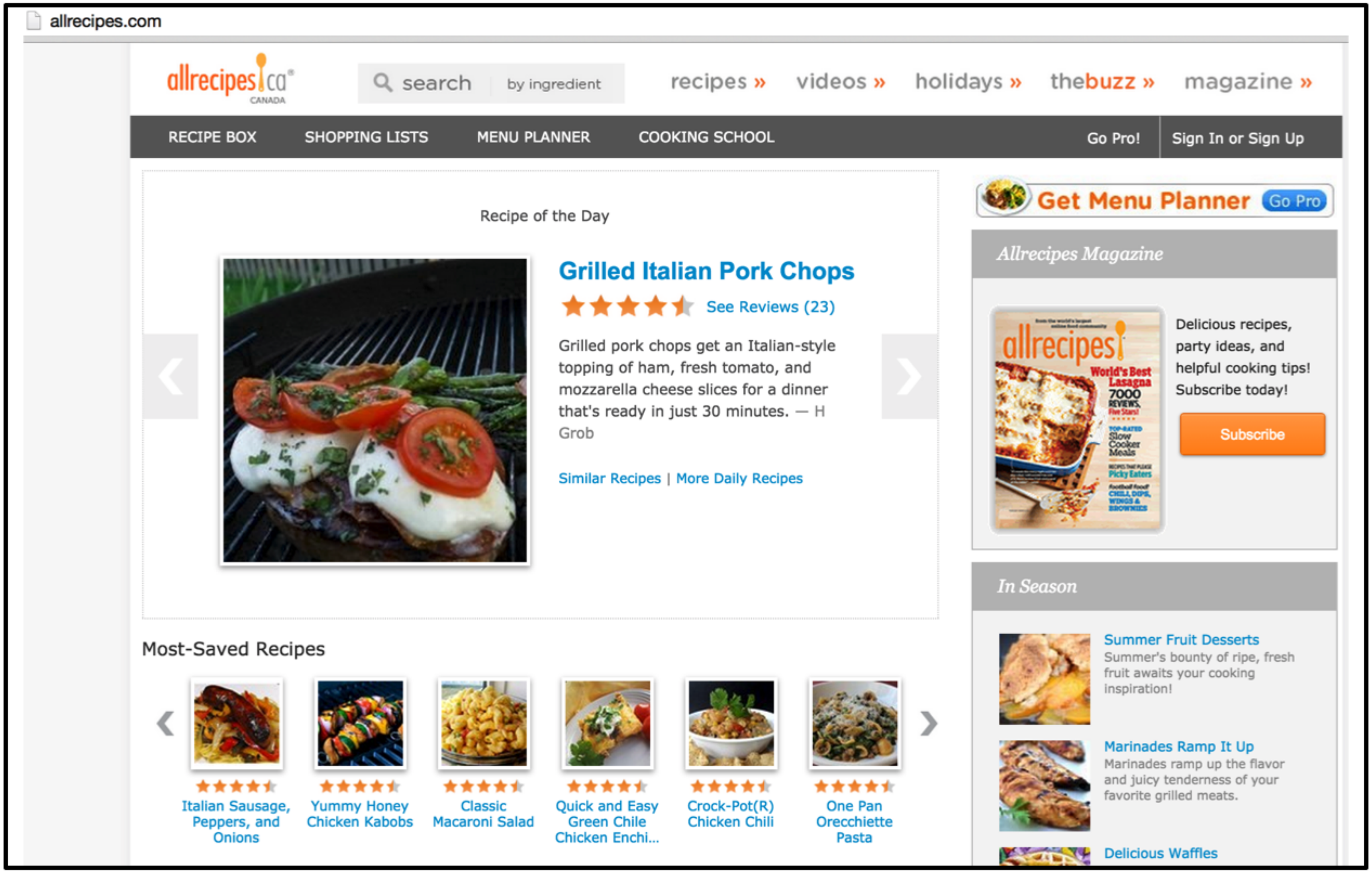
In this lab, we will continue learning about the data science methodology, and focus on the **Modeling** and **Evaluation** stages.

Table of Contents

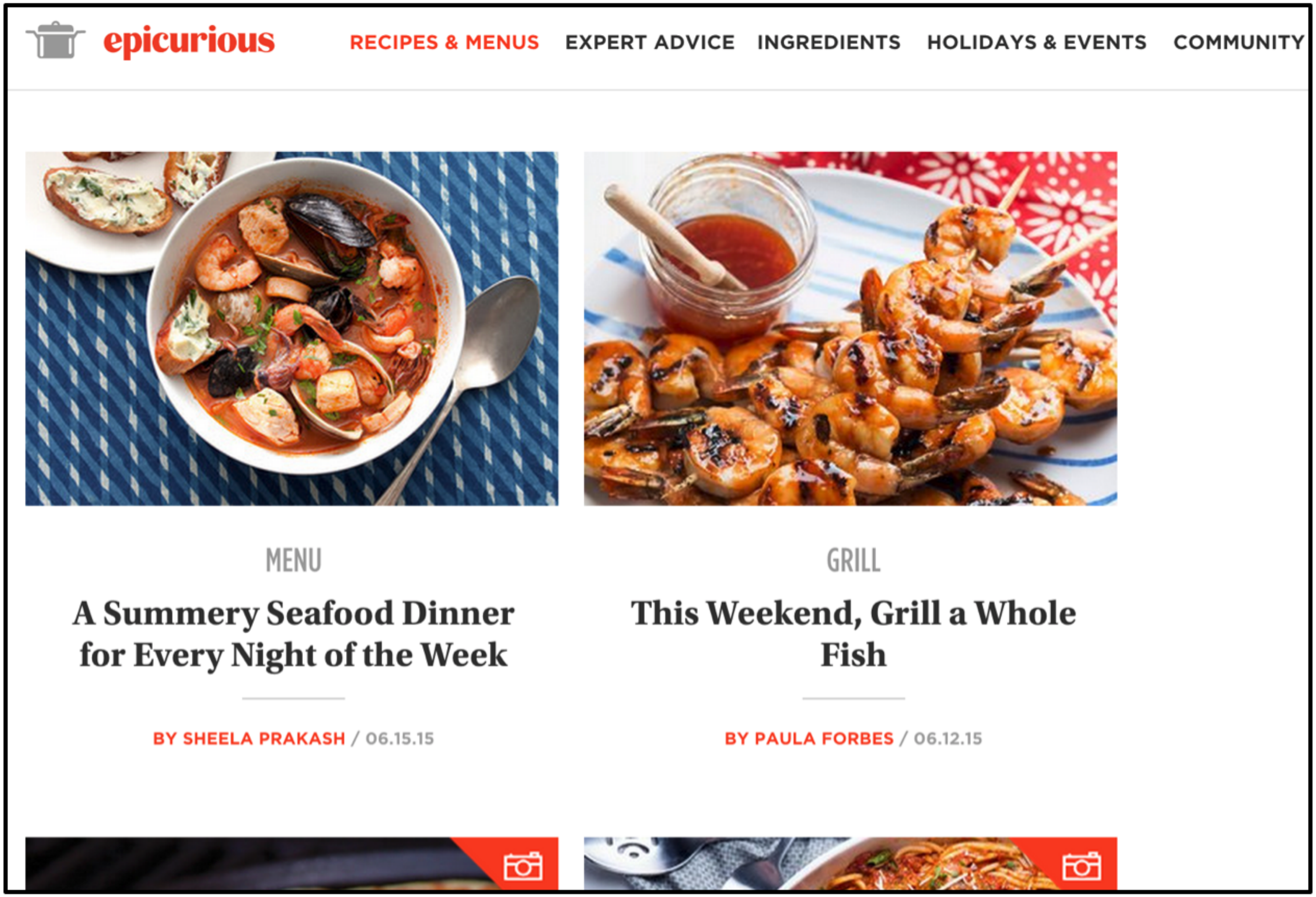
Recap

In Lab **From Understanding to Preparation**, we explored the data and prepared it for modeling.

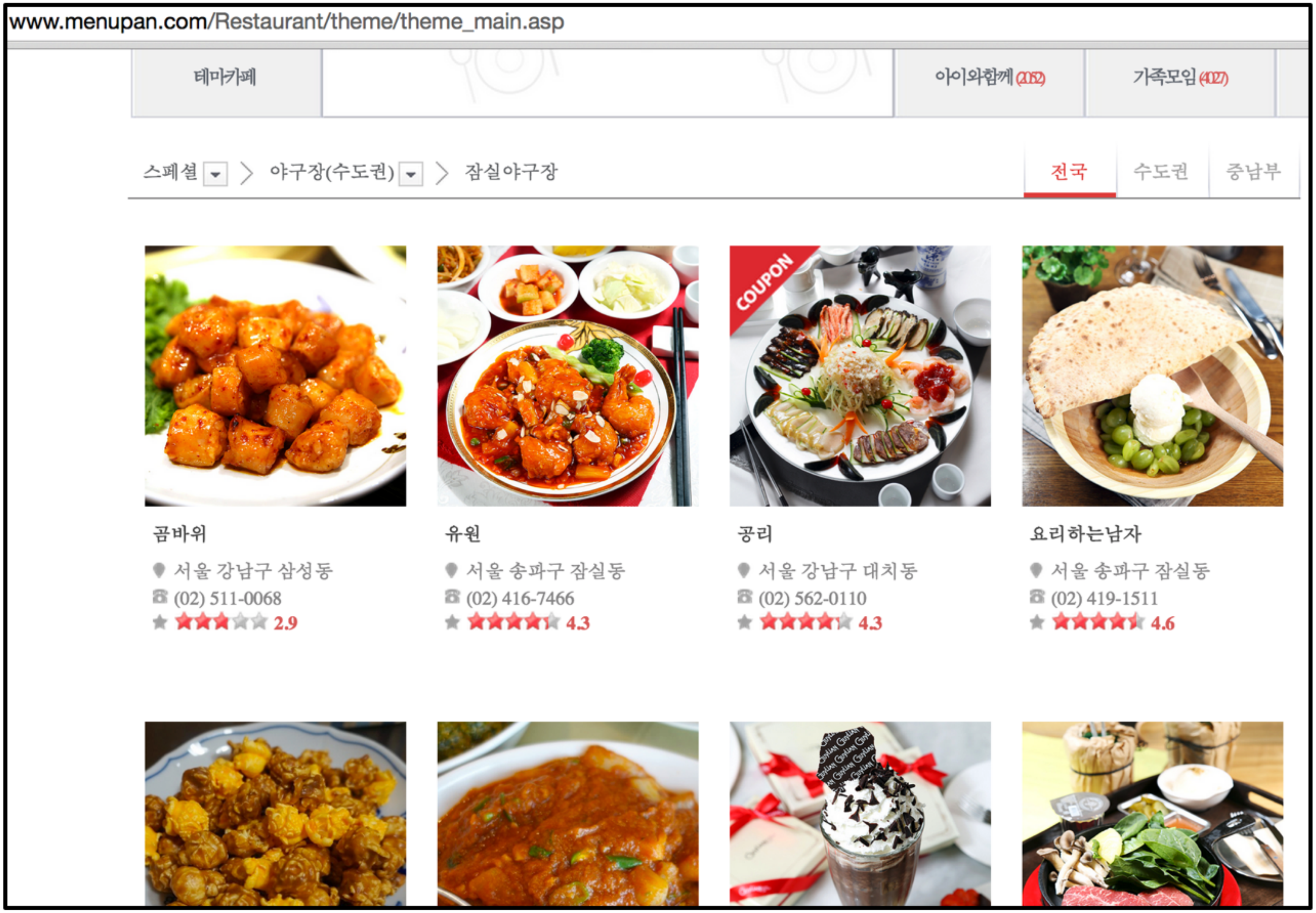
The data was compiled by a researcher named Yong-Yeol Ahn, who scraped tens of thousands of food recipes (cuisines and ingredients) from three different websites, namely:



www.allrecipes.com



www.epicurious.com



www.menupan.com

For more information on Yong-Yeol Ahn and his research, you can read his paper on [Flavor Network and the Principles of Food Pairing (http://yongyeol.com/papers/ahn-flavornet-2011.pdf)](http://yongyeol.com/papers/ahn-flavornet-2011.pdf).

Important note: Please note that you are not expected to know how to program in python. This lab is meant to illustrate the stages of modeling and evaluation of the data science methodology, so it is totally fine if you do not understand the individual lines of code. We have a full course on programming in python, [Python for Data Science (http://cocl.us/PY0101EN_DS0103EN_LAB4_PYTHON)](http://cocl.us/PY0101EN_DS0103EN_LAB4_PYTHON), so please feel free to complete the course if you are interested in learning how to program in python.

Using this notebook:

To run any of the following cells of code, you can type **Shift + Enter** to excute the code in a cell.

Download the library and dependencies that we will need to run this lab.

In [1]:

We already placed the data on an IBM server for your convenience, so let's download it from server and read it into a dataframe called **recipes**.

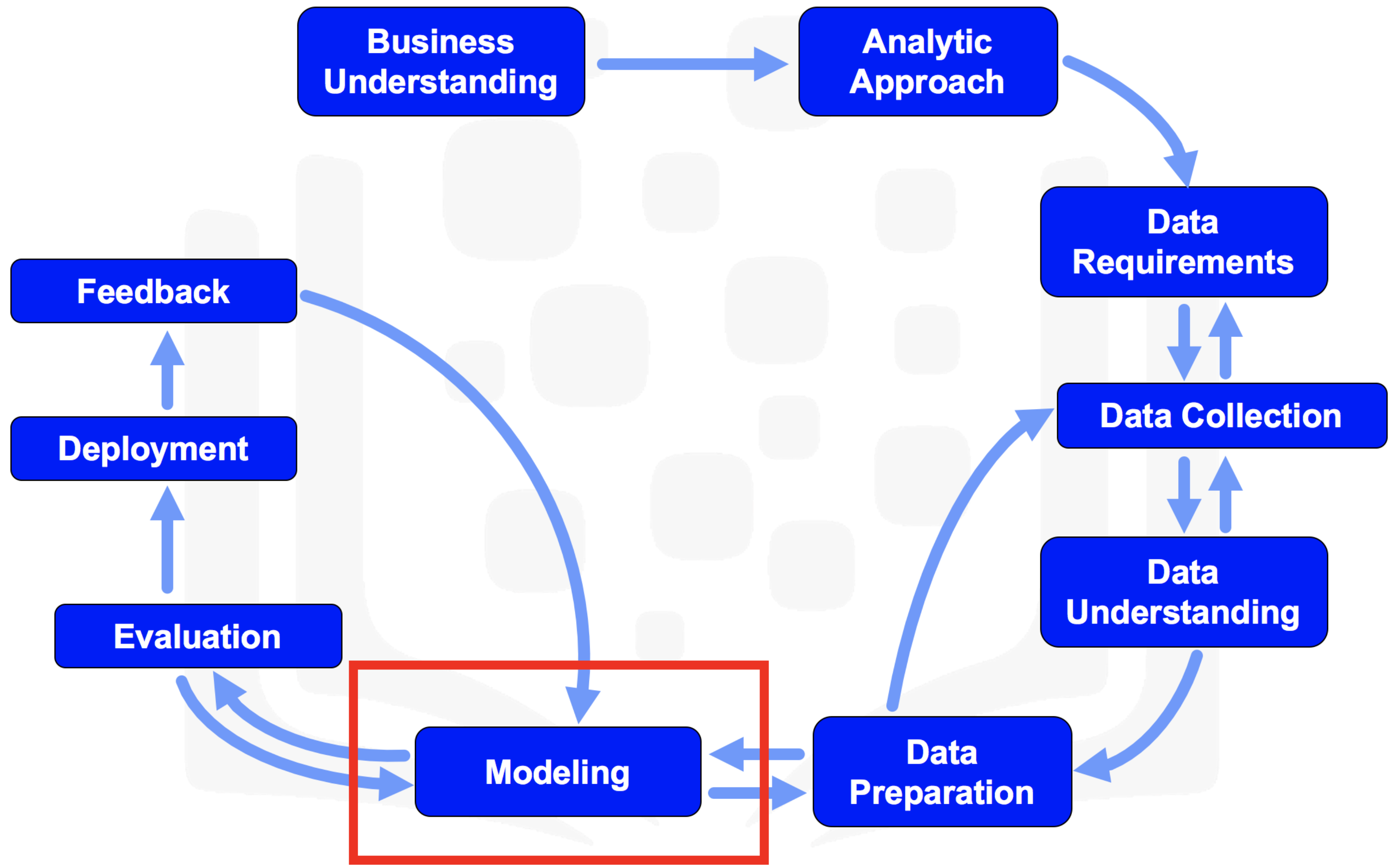
In [2]:

```
Data read into dataframe!
```

We will repeat the preprocessing steps that we implemented in Lab **From Understanding to Preparation** in order to prepare the data for modeling. For more details on preparing the data, please refer to Lab **From Understanding to Preparation**.

In [4]:

Data Modeling


Business Understanding
Analytic Approach
Data Requirements
Data Collection
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Feedback

Download and install more libraries and dependies to build decision trees.

In [5]:

```
Collecting package metadata (current_repoda
ta.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/jupyterlab/co
nda/envs/python

  added / updated specs:
    - python-graphviz


The following packages will be downloaded:

    package                    |
build
    ---------------------------|-----------
------
    ca-certificates-2020.1.1   |
0         125 KB
```

```
    certifi-2020.4.5.1         |
py36_0         155 KB
    openssl-1.1.1g             |       h7b6
447c_0         2.5 MB
    python-graphviz-0.13.2     |
py_0          24 KB
    ---------------------------------------
---------------------

Total:         2.8 MB

The following NEW packages will be INSTALLE
D:

  python-graphviz    pkgs/main/noarch::pyth
on-graphviz-0.13.2-py_0

The following packages will be UPDATED:

  openssl            conda-forge::openssl-
1.1.1f-h516909a_0 --> pkgs/main::openssl-1.
```

```
1.1g-h7b6447c_0

The following packages will be SUPERSEDED b
y a higher-priority channel:

  ca-certificates    conda-forge::ca-certif
icates-2020.4.5~ --> pkgs/main::ca-certific
ates-2020.1.1-0
  certifi            conda-forge::certifi-2
020.4.5.1-py36h~ --> pkgs/main::certifi-202
0.4.5.1-py36_0



Downloading and Extracting Packages
ca-certificates-2020 | 125 KB    | ########
############################# | 100%
openssl-1.1.1g       | 2.5 MB    | ########
############################# | 100%
python-graphviz-0.13 | 24 KB     | ########
############################# | 100%
```

```
certifi-2020.4.5.1   | 155 KB    | ########
################################# | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Check the data again!

In [6]:

Out[6]:

	cuisine	almond	angelica	anise	anise_seed	a
0	vietnamese	0	0	0	0	
1	vietnamese	0	0	0	0	
2	vietnamese	0	0	0	0	
3	vietnamese	0	0	0	0	
4	vietnamese	0	0	0	0	

[bamboo_tree] Only Asian and Indian Cuisines

Here, we are creating a decision tree for the recipes for just some of the Asian (Korean, Japanese, Chinese, Thai) and Indian cuisines. The reason for this is because the decision tree does not run well when the data is biased towards one cuisine, in this case American cuisines. One option is to exclude the American cuisines from our analysis or just build decision trees for different subsets of the data. Let's go with the latter solution.

Let's build our decision tree using the data pertaining to the Asian and Indian cuisines and name our decision tree *bamboo_tree*.

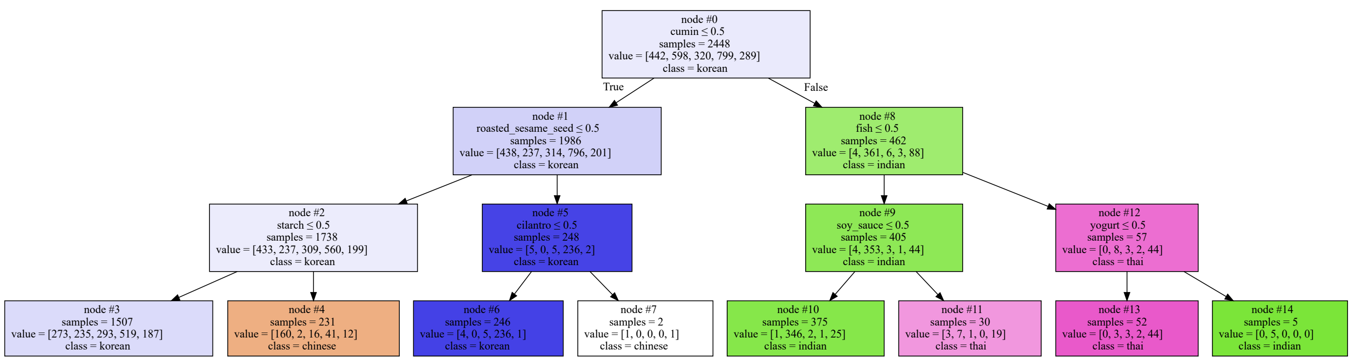
In [7]:

```
Decision tree model saved to bamboo_tree!
```

Let's plot the decision tree and examine how it looks like.

In [8]:

Out[8]:



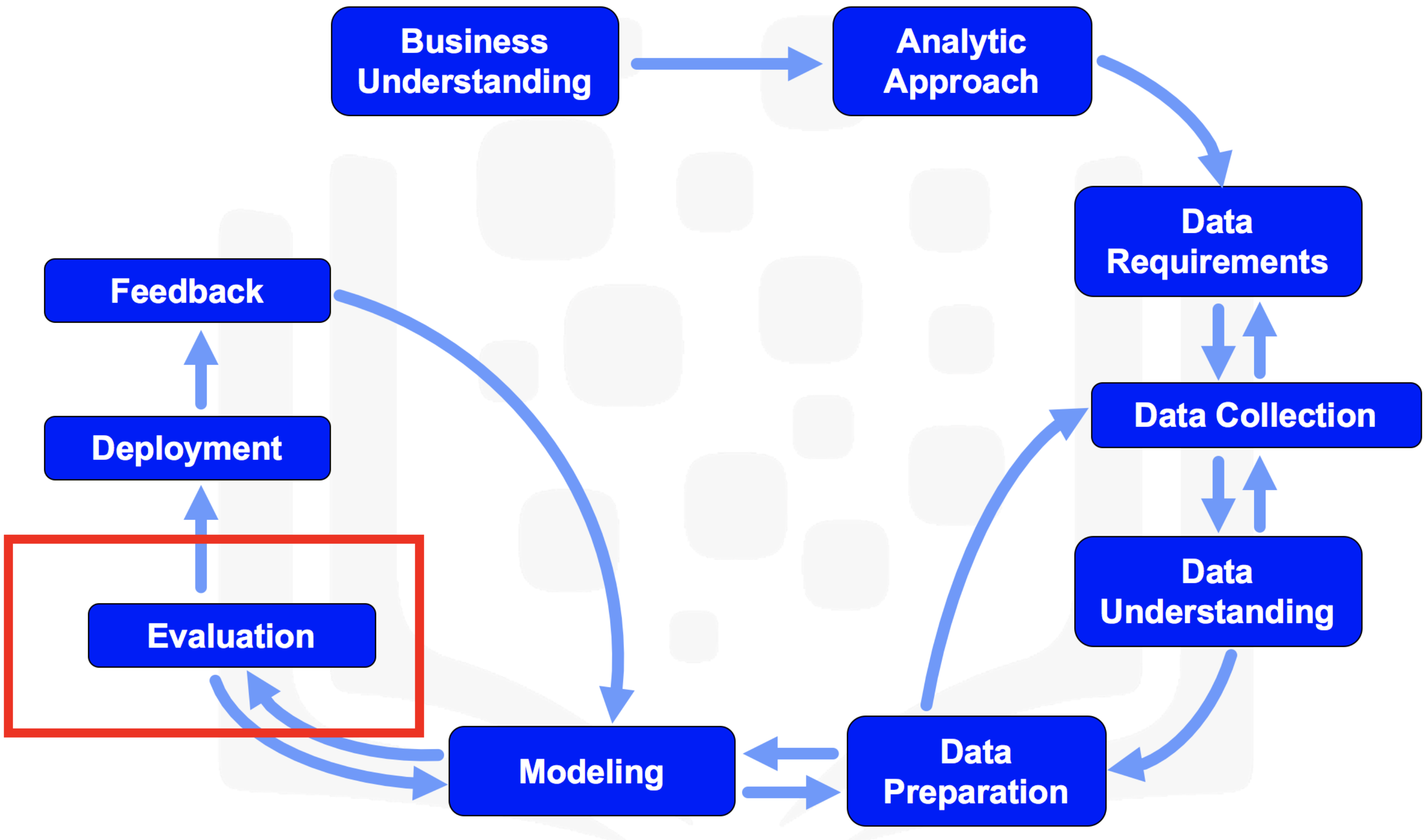
The decision tree learned:

- If a recipe contains *cumin* and *fish* and **no** *yoghurt*, then it is most likely a **Thai** recipe.
- If a recipe contains *cumin* but **no** *fish* and **no** *soy_sauce*, then it is most likely an **Indian** recipe.

You can analyze the remaining branches of the tree to come up with similar rules for determining the cuisine of different recipes.

Feel free to select another subset of cuisines and build a decision tree of their recipes. You can select some European cuisines and build a decision tree to explore the ingredients that differentiate them.

Model Evaluation



To evaluate our model of Asian and Indian cuisines, we will split our dataset into a training set and a test set. We will build the decision tree using the training set. Then, we will test the model on the test set and compare the cuisines that the model predicts to the actual cuisines.

Let's first create a new dataframe using only the data pertaining to the Asian and the Indian cuisines, and let's call the new dataframe **bamboo**.

In [9]:

Let's see how many recipes exist for each cuisine.

In [10]:

Out[10]:

```
korean      799
indian      598
chinese     442
japanese    320
thai        289
Name: cuisine, dtype: int64
```

Let's remove 30 recipes from each cuisine to use as the test set, and let's name this test set **bamboo_test**.

In [11]:

Create a dataframe containing 30 recipes from each cuisine, selected randomly.

In [12]:

Check that there are 30 recipes for each cuisine.

In [13]:

Out[13]:

```
thai        30
chinese     30
indian      30
japanese    30
korean      30
Name: cuisine, dtype: int64
```

Next, let's create the training set by removing the test set from the **bamboo** dataset, and let's call the training set **bamboo_train**.

In [14]:

Check that there are 30 *fewer* recipes now for each cuisine.

In [15]:

Out[15]:

```
korean      769
indian      568
chinese     412
japanese    290
thai        259
Name: cuisine, dtype: int64
```

Let's build the decision tree using the training set, **bamboo_train**, and name the generated tree **bamboo_train_tree** for prediction.

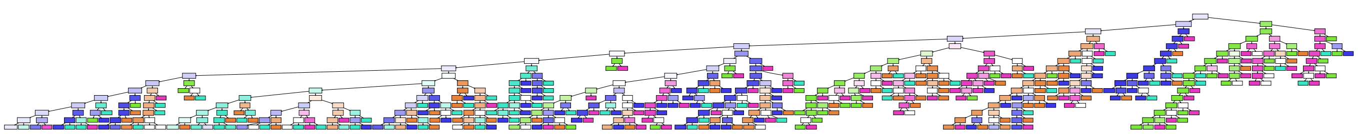
In [16]:

```
Decision tree model saved to bamboo_train_t
ree!
```

Let's plot the decision tree and explore it.

In [17]:

Out[17]:



Now that we defined our tree to be deeper, more decision nodes are generated.

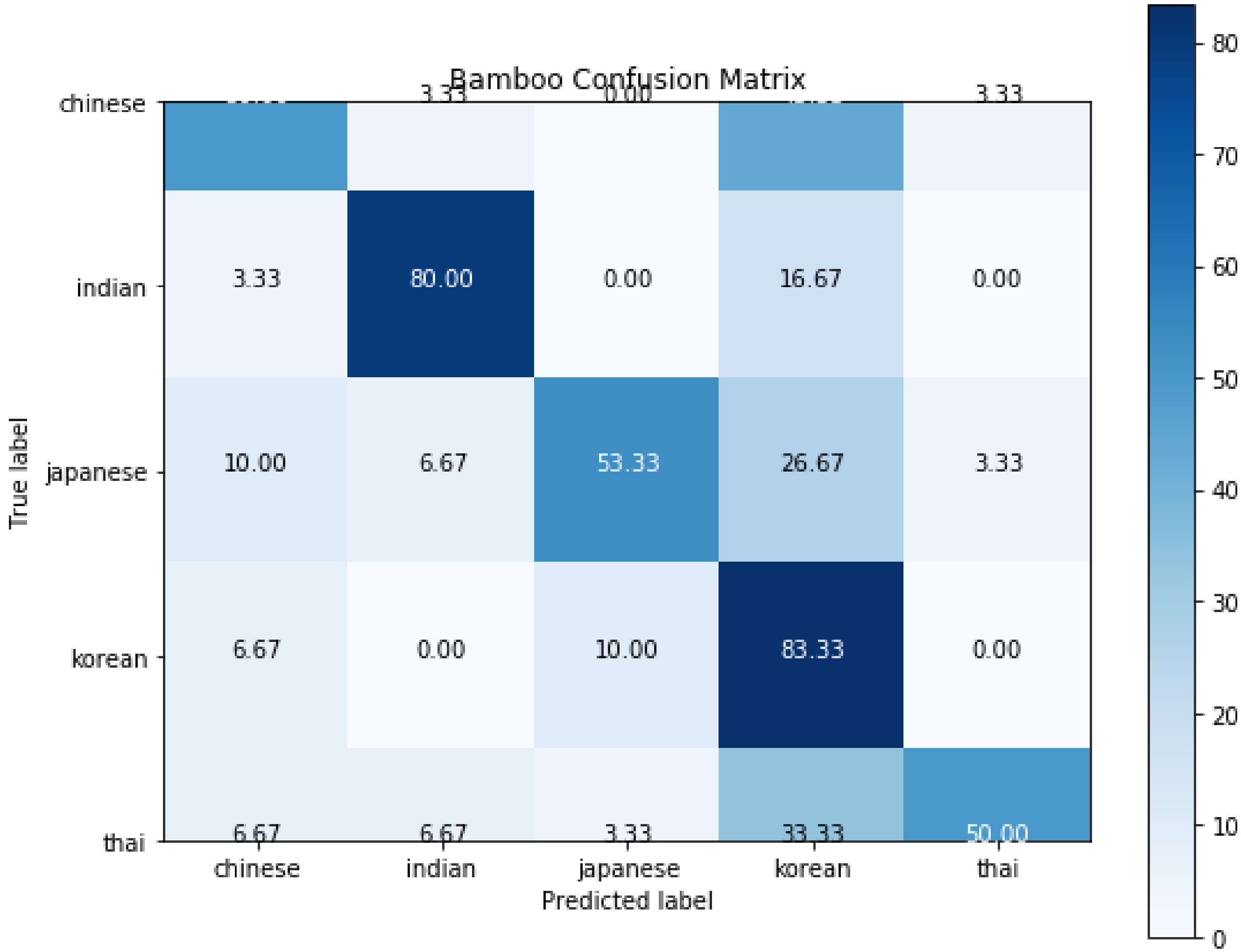
Now let's test our model on the test data.

In [18]:

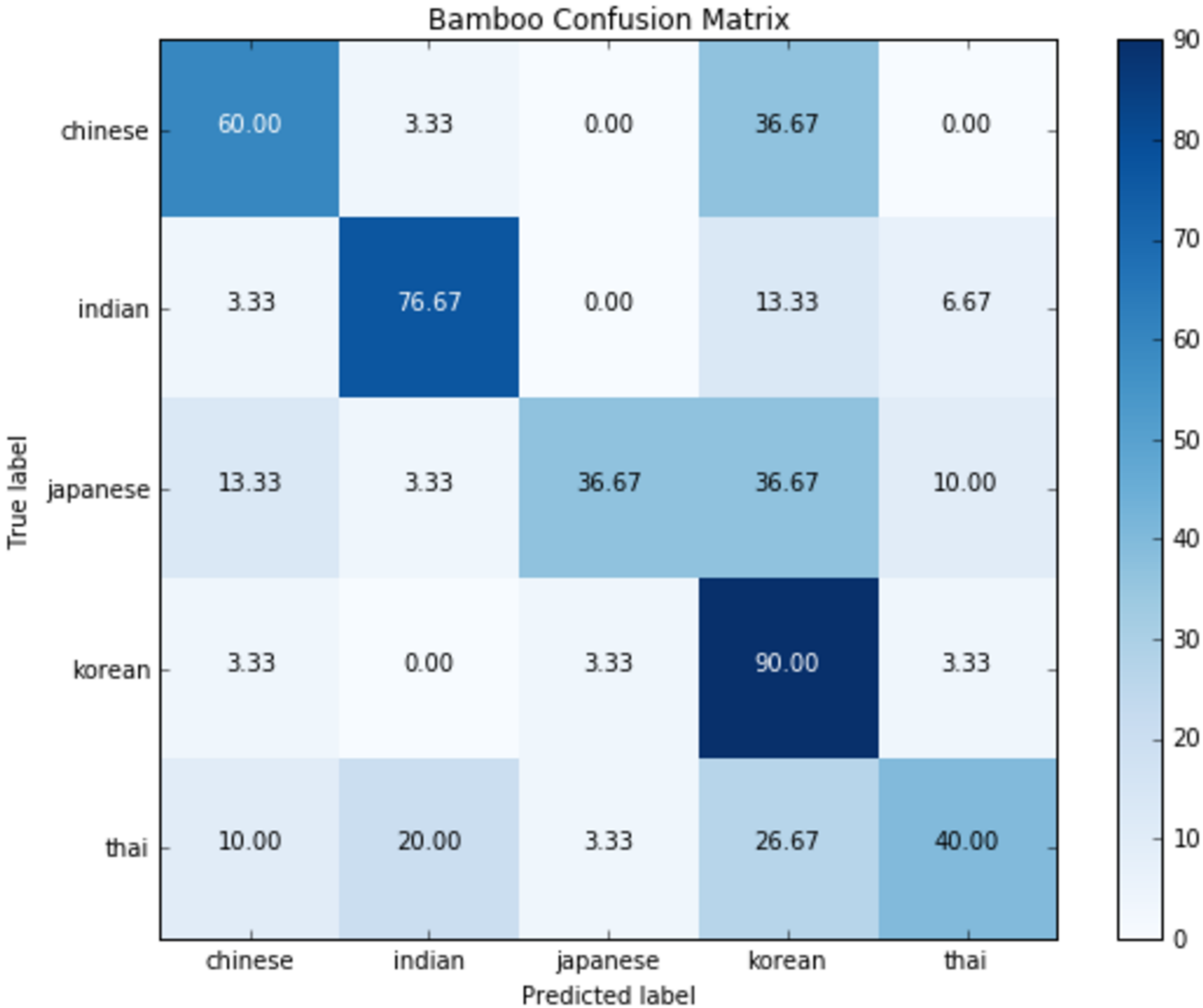
To quantify how well the decision tree is able to determine the cuisine of each recipe correctly, we will create a confusion matrix which presents a nice summary on how many recipes from each cuisine are correctly classified. It also sheds some light on what cuisines are being confused with what other cuisines.

So let's go ahead and create the confusion matrix for how well the decision tree is able to correctly classify the recipes in **bamboo_test**.

In [19]:


Bamboo Confusion Matrix
True label
Predicted label
chinese
indian
japanese
korean
thai
3.33
0.00
3.33
3.33
80.00
0.00
16.67
0.00
10.00
6.67
53.33
26.67
3.33
6.67
0.00
10.00
83.33
0.00
6.67
6.67
3.33
33.33
50.00
chinese
indian
japanese
korean
thai
80
70
60
50
40
30
20
10
0

After running the above code, you should get a confusion matrix similar to the following:


Bamboo Confusion Matrix
True label
chinese
indian
japanese
korean
thai
Predicted label
chinese
indian
japanese
korean
thai
60.00 3.33 0.00 36.67 0.00
3.33 76.67 0.00 13.33 6.67
13.33 3.33 36.67 36.67 10.00
3.33 0.00 3.33 90.00 3.33
10.00 20.00 3.33 26.67 40.00
0
10
20
30
40
50
60
70
80
90

The rows represent the actual cuisines from the dataset and the columns represent the predicted ones. Each row should sum to 100%. According to this confusion matrix, we make the following observations:

- Using the first row in the confusion matrix, 60% of the **Chinese** recipes in **bamboo_test** were correctly classified by our decision tree whereas 37% of the **Chinese** recipes were misclassified as **Korean** and 3% were misclassified as **Indian**.
- Using the Indian row, 77% of the **Indian** recipes in **bamboo_test** were correctly classified by our decision tree and 3% of the **Indian** recipes were misclassified as **Chinese** and 13% were misclassified as **Korean** and 7% were misclassified as **Thai**.

Please note that because decision trees are created using random sampling of the datapoints in the training set, then you may not get the same results every time you create the decision tree even using the same training set. The performance should still be comparable though! So don't worry if you get slightly different numbers in your confusion matrix than the ones shown above.

Using the reference confusion matrix, how many **Japanese** recipes were correctly classified by our decision tree?

Your Answer: 36.67%

Double-click **here** for the solution.

Also using the reference confusion matrix, how many **Korean** recipes were misclassified as **Japanese**?

Your Answer: 3.33%

Double-click **here** for the solution.

What cuisine has the least number of recipes correctly classified by the decision tree using the reference confusion matrix?

Your Answer: Japanese 36.67%

Double-click **here** for the solution.

Thank you for completing this lab!

This notebook was created by [Alex Aklson (https://www.linkedin.com/in/aklson/)](https://www.linkedin.com/in/aklson/). We hope you found this lab session interesting. Feel free to contact us if you have any questions!

This notebook is part of the free course on **Cognitive Class** called *Data Science Methodology*. If you accessed this notebook outside the course, you can take this free self-paced course, online by clicking [here (https://cocl.us/DS0103EN_LAB4_PYTHON)](https://cocl.us/DS0103EN_LAB4_PYTHON).
