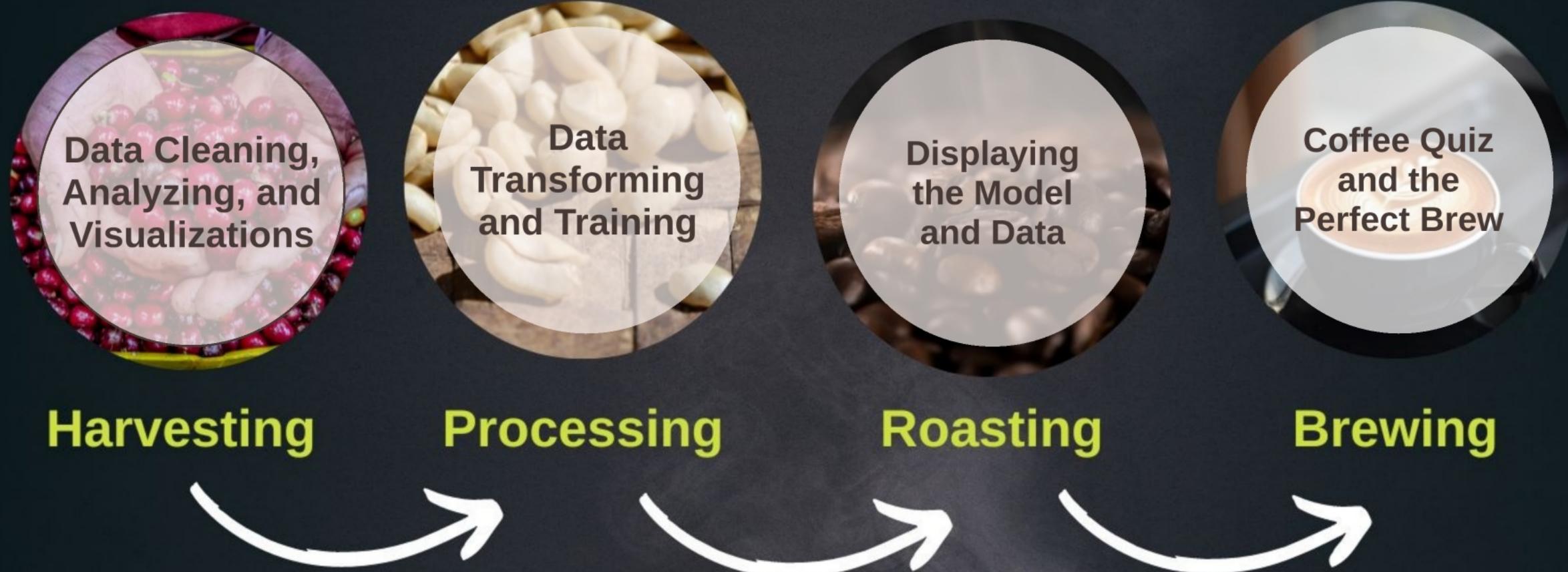


# From the tree to the coffee cup

## Project 4 Machine Learning



Presented by Ankita Sarkar, Iqra Imam, Kim Sernett, Marissa Gallegos, Raelle Nalos, and Angel Toscano



## About Our Data

Dataset was scraped from coffeereview.com and contained over 7000 ratings and reviews of different roasted coffee. Dataset is from Kaggle

Cleaning and Transforming

Data Visuals

	<b>title</b>	<b>rating</b>	<b>acidity_structure</b>	<b>aftertaste</b>	<b>aroma</b>	<b>body</b>	<b>flavor</b>	<b>with_milk</b>	<b>agtron</b>	<b>blind_assessment</b>	<b>bottom_line</b>	<b>coffee_origin</b>	<b>est_price</b>	<b>notes</b>	<b>review_date</b>	<b>roast_level</b>	<b>roaster</b>	<b>roaster_location</b>
0	Bolivia Manantial Gesha	93	9.0	8.0	9	8	9	NaN	60/78	Richly aromatic, floral-toned. Magnolia, cocoa...	This washed Boliva Gesha has all the aromatics...	Caranavi, Bolivia	\$30.00/12 ounces	Produced by Angel Mamani Chambi, entirely of t...	Jan-23	Medium-Light	Red Rooster Coffee Roaster	Floyd, Virginia
1	Yellow Pacamara Carbonic Maceration Nanolot	92	8.0	8.0	9	8	9	NaN	60/78	Crisply sweet-savory. White peach, hop flowers...	A carbonic-macerated Yellow Pacamara grown in ...	Matagalpa growing region, Nicaragua	\$160.00/12 ounces	Produced by Benjamin Weiner at Finca Idealista...	Dec-22	Medium-Light	Eccentricity Coffee Co.	Cleveland, Ohio
2	Ethiopia Gera Genji Challa	94	9.0	8.0	9	9	9	NaN	59/77	Delicately aromatic, complex. Lilac, cocoa nib...	A nuanced, multilayered washed Ethiopia cup wi...	Agaro Gera, Jimma Zone, Oromia State, Ethiopia	\$28.00/12 ounces	Ethiopia coffees like this one are largely pro...	Dec-22	Medium-Light	Mostra Coffee	San Diego, California
3	Yirgacheffe Mengesha Natural	94	9.0	8.0	9	9	9	NaN	60/77	High-toned, fruit-driven. Boysenberry, pear, c...	A fruit medley in a cup — think boysenberry an...	Yirgacheffe growing region, southern Ethiopia	\$20.50/12 ounces	Produced at Mengesha Farm from selections of i...	Nov-22	Medium-Light	Regent Coffee	Glendale, California
4	Tropical Summer Colombia La Sierra	93	9.0	8.0	9	8	9	NaN	60/77	Fruit-driven, crisply chocolaty. Goji berry, d...	An experimentally processed Colombia, sweetly ...	La Sierra, Cauca Department, Colombia	\$18.99/8 ounces	Produced by smallholding farmers from trees of...	Nov-22	Medium-Light	Merge Coffee Company	Harrisonburg, Virginia

Standardizing  
the Price

```
# conversion rates to gram
gram = 0.035
kg = 35.274
mL = 0.034
lb = 453.592

# convert all unit_amt to ounces
for i in range(len(money_df)):
    if money_df.loc[i,'unit_measurement'] == 'gram':
        money_df.loc[i, 'unit_amt'] = money_df.loc[i, 'unit_amt']* gram
        money_df.loc[i,'unit_measurement'] = 'ounce'
    if money_df.loc[i,'unit_measurement'] == 'kilogram':
        money_df.loc[i, 'unit_amt'] = money_df.loc[i, 'unit_amt']* kg
        money_df.loc[i,'unit_measurement'] = 'ounce'
    if money_df.loc[i,'unit_measurement'] == 'milliliter':
        money_df.loc[i, 'unit_amt'] = money_df.loc[i, 'unit_amt']* mL
        money_df.loc[i,'unit_measurement'] = 'ounce'
    if money_df.loc[i,'unit_measurement'] == 'pound':
        money_df.loc[i, 'unit_amt'] = money_df.loc[i, 'unit_amt']* lb
        money_df.loc[i,'unit_measurement'] = 'ounce'

# update/drop columns to reflect changes
money_df = money_df.rename(columns={'unit_amt': "ounces"})
money_df = money_df.drop(columns = ['unit_measurement'])
```

```
# used easy-exchange-rates package for a more dynamic exchange rate and not rely on a specific date
from datetime import date

today = date.today().strftime('%Y-%m-%d')

from easy_exchange_rates import API
api = API()

def GetScalingData(c1):
    time_series = api.get_exchange_rates(
        base_currency='USD',
        start_date=today,
        end_date=today,
        targets=["USD",c1]
    )

    data_frame = api.to_dataframe(time_series)
    rate=data_frame[c1][0]
    return rate

# GetScalingData('CAD')
```

# Final Product

		title	rating	acidity_structure	aftertaste	aroma	body	flavor	blind_assessment	coffee_origin	notes	roast_level	city	state	usd_per_oz
0		Bolivia Manantial Gesha	93.0	9.0	8.0	9.0	8.0	9.0	Richly aromatic, floral-toned. Magnolia, cocoa...	Caranavi, Bolivia	Produced by Angel Mamani Chambi, entirely of t...	Medium-Light	Floyd	Virginia	2.5
1		Yellow Pacamara Carbonic Maceration Nanolot	92.0	8.0	8.0	9.0	8.0	9.0	Crisply sweet-savory. White peach, hop flowers...	Matagalpa growing region, Nicaragua	Produced by Benjamin Weiner at Finca Idealista...	Medium-Light	Cleveland	Ohio	13.33
2		Ethiopia Gera Genji Challa	94.0	9.0	8.0	9.0	9.0	9.0	Delicately aromatic, complex. Lilac, cocoa nib...	Agaro Gera, Jimma Zone, Oromia State, Ethiopia	Ethiopia coffees like this one are largely pro...	Medium-Light	San Diego	California	2.33
3		Yirgacheffe Mengesha Natural	94.0	9.0	8.0	9.0	9.0	9.0	High-toned, fruit-driven. Boysenberry, pear, c...	Yirgacheffe growing region, southern Ethiopia	Produced at Mengesha Farm from selections of i...	Medium-Light	Glendale	California	1.71
4		Tropical Summer Colombia La Sierra	93.0	9.0	8.0	9.0	8.0	9.0	Fruit-driven, crisply chocolaty. Goji berry, d...	La Sierra, Cauca Department, Colombia	Produced by smallholding farmers from trees of...	Medium-Light	Harrisonburg	Virginia	2.37
5		The Glistening Orchard Blend	91.0	7.0	8.0	9.0	8.0	9.0	Fruity and crisply chocolaty. Green banana, ca...	Colombia; Ethiopia	A blend of three coffees — a washed coffee fer...	Medium-Light	Taipei	Taiwan	1749.3
6		Tinamit Tolimán	93.0	8.0	8.0	9.0	9.0	9.0	Deeply sweet-tart, chocolate-toned. Dark choco...	San Lucas Tolimán, Lake Atitlán growing region...	Produced by smallholding farmers of Asociación...	Medium-Light	Antigua	Guatemala	1.33
7		Colombia La Esperanza 100% Geisha Hanashaku	94.0	9.0	8.0	9.0	9.0	9.0	Delicate, elegant, sweetly bright. Bergamot, c...	Valle de Cauca, Colombia	Produced at Finca Cerro Azul, entirely of the ...	Light	Taoyuan	Taiwan	3811.12
8		Colombia El Paraíso Floral Lychee	93.0	8.0	8.0	9.0	9.0	9.0	Floral-toned, richly sweet-savory. Black cherr...	Cauca Department, Colombia	Produced by Diego Bermudez entirely of the Cas...	Light	Taipei	Taiwan	2667.78
9		Ethiopia Bekele Heto Natural	93.0	8.0	8.0	9.0	9.0	9.0	Berry-driven, invitingly sweet-tart. Dried mul...	Worka-Sakaro, Gedeb District, Gedeo Zone, Ethi...	Produced by Bekele Heto from largely indigenou...	Light	Oakland	California	2.1

```
# conversion rates to gram
gram = 0.035
kg = 35.274
mL = 0.034
lb = 453.592

# convert all unit_amt to ounces
for i in range(len(money_df)):
    if money_df.loc[i,'unit_measurement'] == 'gram':
        money_df.loc[i, 'unit_amt'] = money_df.loc[i, 'unit_amt']* gram
        money_df.loc[i,'unit_measurement'] = 'ounce'
    if money_df.loc[i,'unit_measurement'] == 'kilogram':
        money_df.loc[i, 'unit_amt'] = money_df.loc[i, 'unit_amt']* kg
        money_df.loc[i,'unit_measurement'] = 'ounce'
    if money_df.loc[i,'unit_measurement'] == 'milliliter':
        money_df.loc[i, 'unit_amt'] = money_df.loc[i, 'unit_amt']* mL
        money_df.loc[i,'unit_measurement'] = 'ounce'
    if money_df.loc[i,'unit_measurement'] == 'pound':
        money_df.loc[i, 'unit_amt'] = money_df.loc[i, 'unit_amt']* lb
        money_df.loc[i,'unit_measurement'] = 'ounce'

# update/drop columns to reflect changes
money_df = money_df.rename(columns={'unit_amt': "ounces"})
money_df = money_df.drop(columns = ['unit_measurement'])
```

```
# used easy-exchange-rates package for a more dynamic exchange rate and not rely on a specific date
from datetime import date

today = date.today().strftime('%Y-%m-%d')

from easy_exchange_rates import API
api = API()

def GetScalingData(c1):
    time_series = api.get_exchange_rates(
        base_currency='USD',
        start_date=today,
        end_date=today,
        targets=["USD",c1]
    )

    data_frame = api.to_dataframe(time_series)
    rate=data_frame[c1][0]
    return rate

# GetScalingData('CAD')
```

# Final Product

	<b>title</b>	<b>rating</b>	<b>acidity_structure</b>	<b>aftertaste</b>	<b>aroma</b>	<b>body</b>	<b>flavor</b>	<b>with_milk</b>	<b>agtron</b>	<b>blind_assessment</b>	<b>bottom_line</b>	<b>coffee_origin</b>	<b>est_price</b>	<b>notes</b>	<b>review_date</b>	<b>roast_level</b>	<b>roaster</b>	<b>roaster_location</b>
0	Bolivia Manantial Gesha	93	9.0	8.0	9	8	9	NaN	60/78	Richly aromatic, floral-toned. Magnolia, cocoa...	This washed Boliva Gesha has all the aromatics...	Caranavi, Bolivia	\$30.00/12 ounces	Produced by Angel Mamani Chambi, entirely of t...	Jan-23	Medium-Light	Red Rooster Coffee Roaster	Floyd, Virginia
1	Yellow Pacamara Carbonic Maceration Nanolot	92	8.0	8.0	9	8	9	NaN	60/78	Crisply sweet-savory. White peach, hop flowers...	A carbonic-macerated Yellow Pacamara grown in ...	Matagalpa growing region, Nicaragua	\$160.00/12 ounces	Produced by Benjamin Weiner at Finca Idealista...	Dec-22	Medium-Light	Eccentricity Coffee Co.	Cleveland, Ohio
2	Ethiopia Gera Genji Challa	94	9.0	8.0	9	9	9	NaN	59/77	Delicately aromatic, complex. Lilac, cocoa nib...	A nuanced, multilayered washed Ethiopia cup wi...	Agaro Gera, Jimma Zone, Oromia State, Ethiopia	\$28.00/12 ounces	Ethiopia coffees like this one are largely pro...	Dec-22	Medium-Light	Mostra Coffee	San Diego, California
3	Yirgacheffe Mengesha Natural	94	9.0	8.0	9	9	9	NaN	60/77	High-toned, fruit-driven. Boysenberry, pear, c...	A fruit medley in a cup — think boysenberry an...	Yirgacheffe growing region, southern Ethiopia	\$20.50/12 ounces	Produced at Mengesha Farm from selections of i...	Nov-22	Medium-Light	Regent Coffee	Glendale, California
4	Tropical Summer Colombia La Sierra	93	9.0	8.0	9	8	9	NaN	60/77	Fruit-driven, crisply chocolaty. Goji berry, d...	An experimentally processed Colombia, sweetly ...	La Sierra, Cauca Department, Colombia	\$18.99/8 ounces	Produced by smallholding farmers from trees of...	Nov-22	Medium-Light	Merge Coffee Company	Harrisonburg, Virginia

Standardizing  
the Price



## About Our Data

Dataset was scraped from coffeereview.com and contained over 7000 ratings and reviews of different roasted coffee. Dataset is from Kaggle

Cleaning and Transforming

Data Visuals

# Tableau Visualizations

Coffees  
Per Roast  
Level

Characteristics  
of Roast Level

Coffee  
Ratings

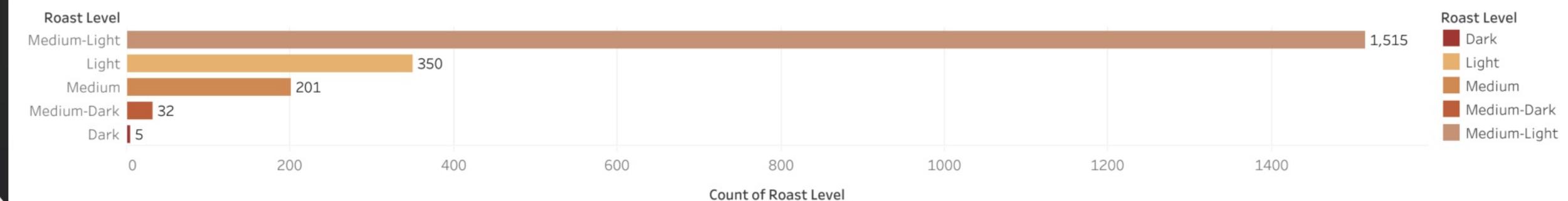
Coffee  
Origin

Most  
Expensive  
by Origin

Roast  
Level by  
Location

Most  
Expensive  
by Roast  
Level

## Number of Coffees per Roast Level



# Tableau Visualizations

Coffees  
Per Roast  
Level

Characteristics  
of Roast Level

Coffee  
Ratings

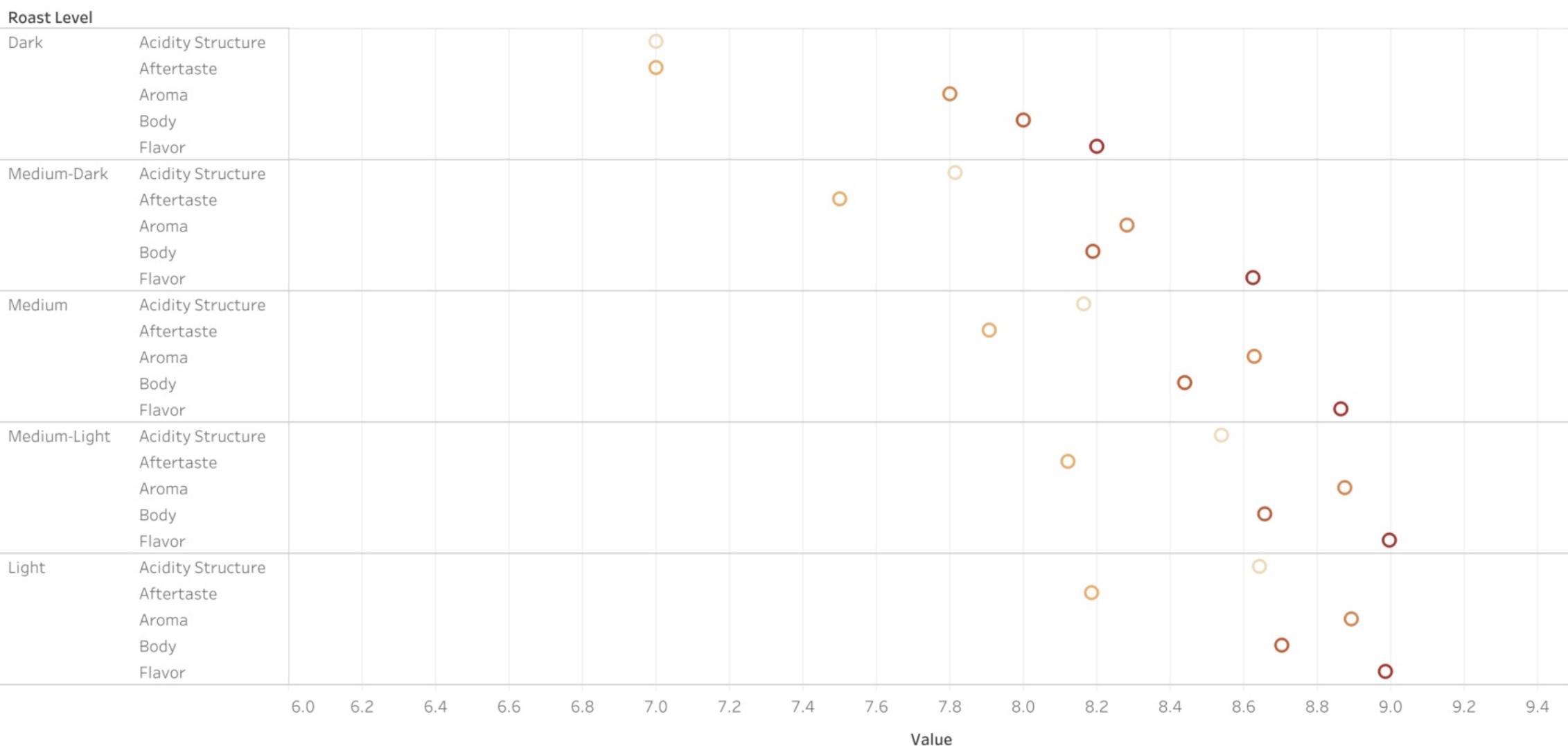
Coffee  
Origin

Most  
Expensive  
by Origin

Roast  
Level by  
Location

Most  
Expensive  
by Roast  
Level

# Characteristics of Roast



**Measure Names**

- Acidity Structure
- Aftertaste
- Aroma
- Body
- Flavor

# Tableau Visualizations

Coffees  
Per Roast  
Level

Characteristics  
of Roast Level

Coffee  
Ratings

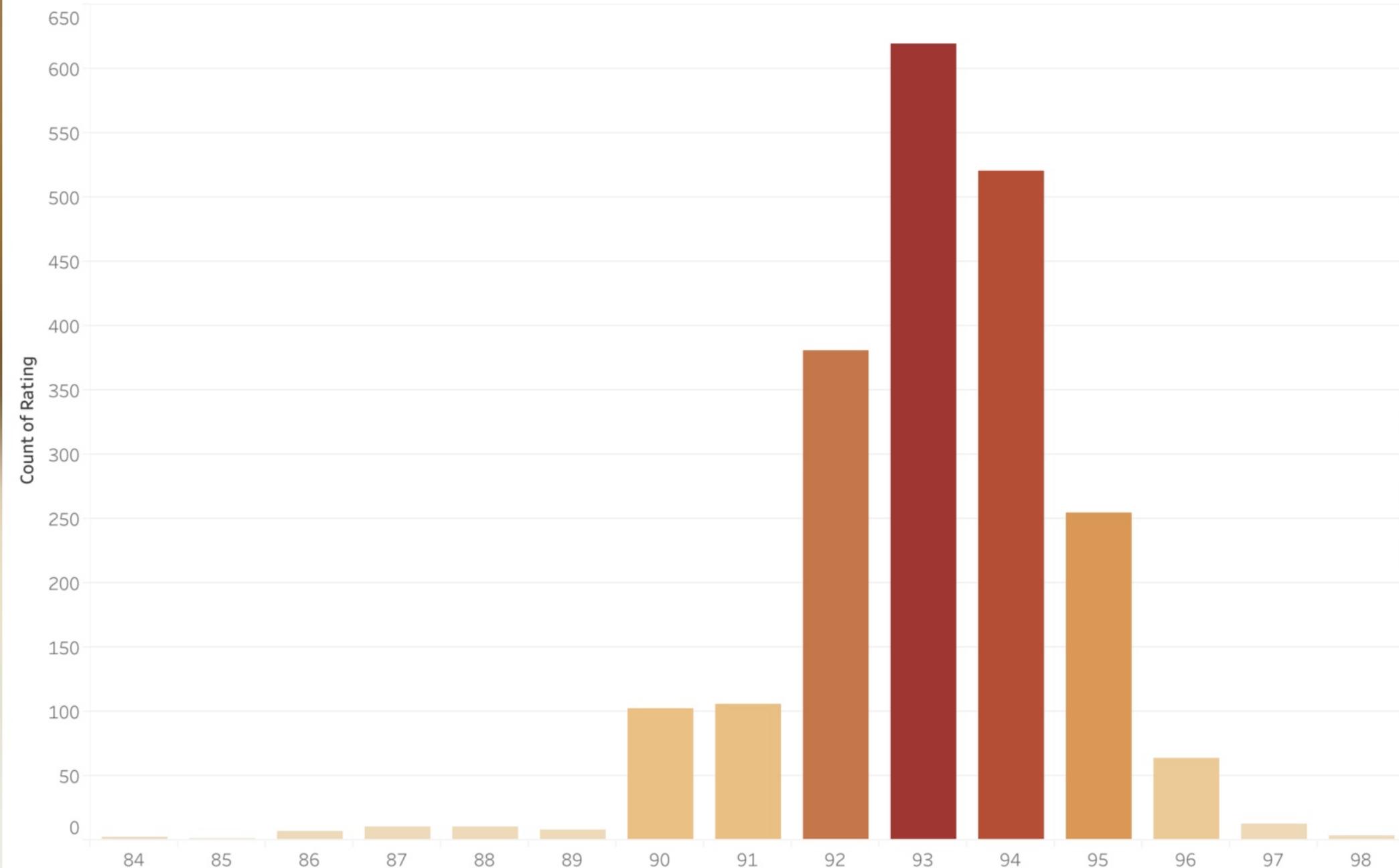
Coffee  
Origin

Most  
Expensive  
by Origin

Roast  
Level by  
Location

Most  
Expensive  
by Roast  
Level

# Coffee Ratings



# Tableau Visualizations

Coffees  
Per Roast  
Level

Characteristics  
of Roast Level

Coffee  
Ratings

Coffee  
Origin

Most  
Expensive  
by Origin

Roast  
Level by  
Location

Most  
Expensive  
by Roast  
Level

## Origin Countries



# Tableau Visualizations

Coffees  
Per Roast  
Level

Characteristics  
of Roast Level

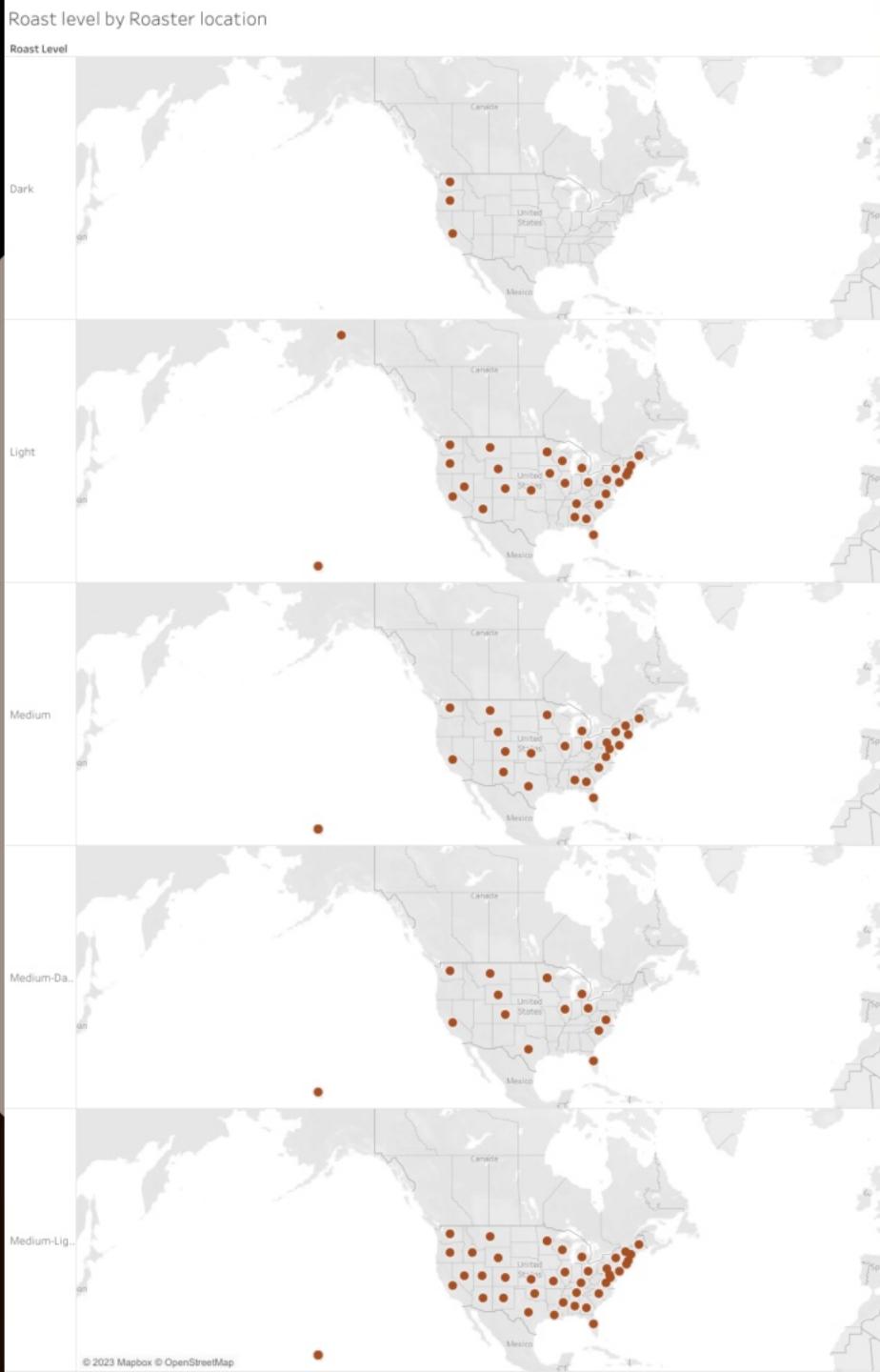
Coffee  
Ratings

Coffee  
Origin

Most  
Expensive  
by Origin

Roast  
Level by  
Location

Most  
Expensive  
by Roast  
Level



# Tableau Visualizations

Coffees  
Per Roast  
Level

Characteristics  
of Roast Level

Coffee  
Ratings

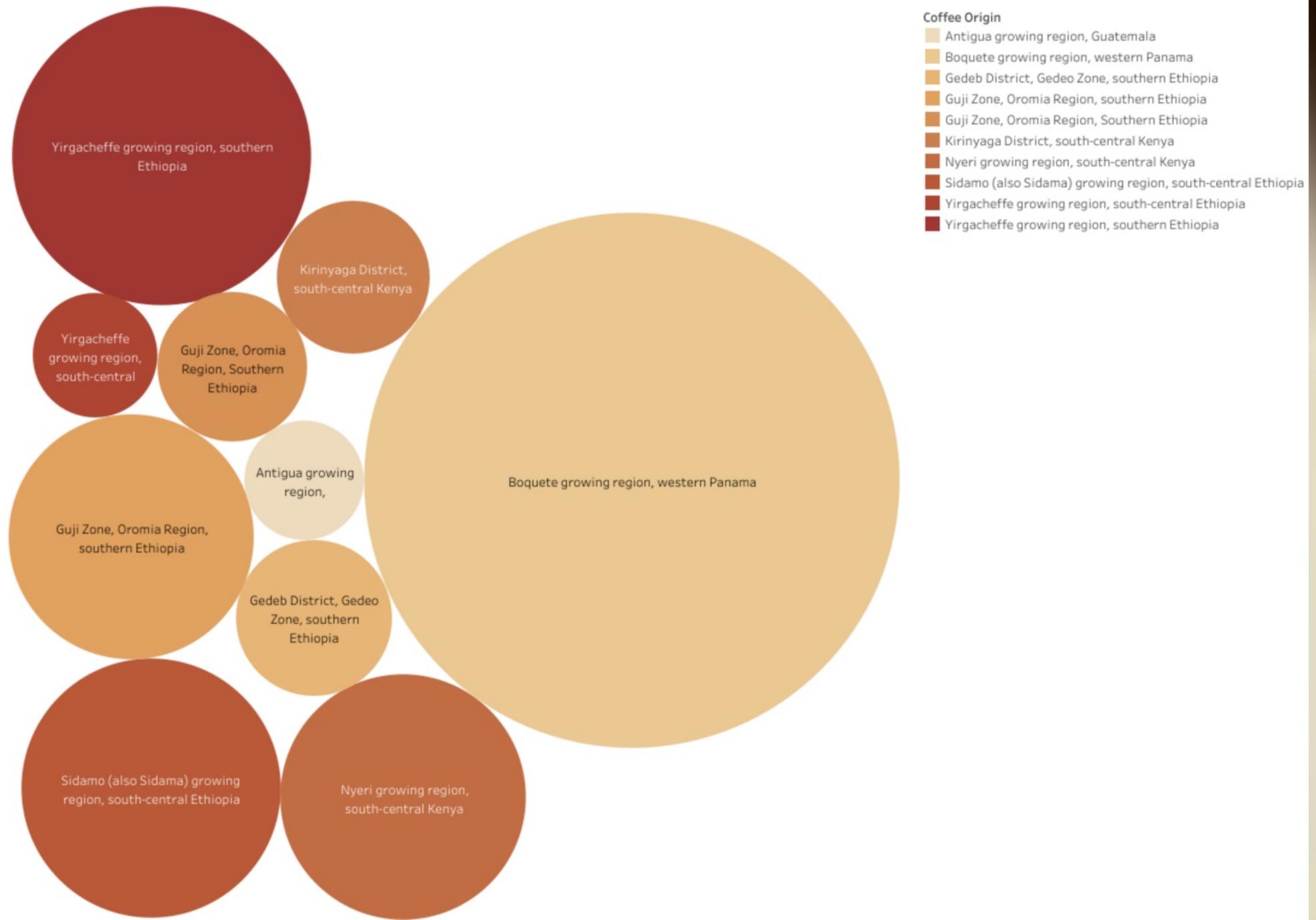
Coffee  
Origin

Most  
Expensive  
by Origin

Roast  
Level by  
Location

Most  
Expensive  
by Roast  
Level

## Most expensive by Origin



# Tableau Visualizations

Coffees  
Per Roast  
Level

Characteristics  
of Roast Level

Coffee  
Ratings

Coffee  
Origin

Most  
Expensive  
by Origin

Roast  
Level by  
Location

Most  
Expensive  
by Roast  
Level



# Tableau Visualizations

Coffees  
Per Roast  
Level

Characteristics  
of Roast Level

Coffee  
Ratings

Coffee  
Origin

Most  
Expensive  
by Origin

Roast  
Level by  
Location

Most  
Expensive  
by Roast  
Level



## About Our Data

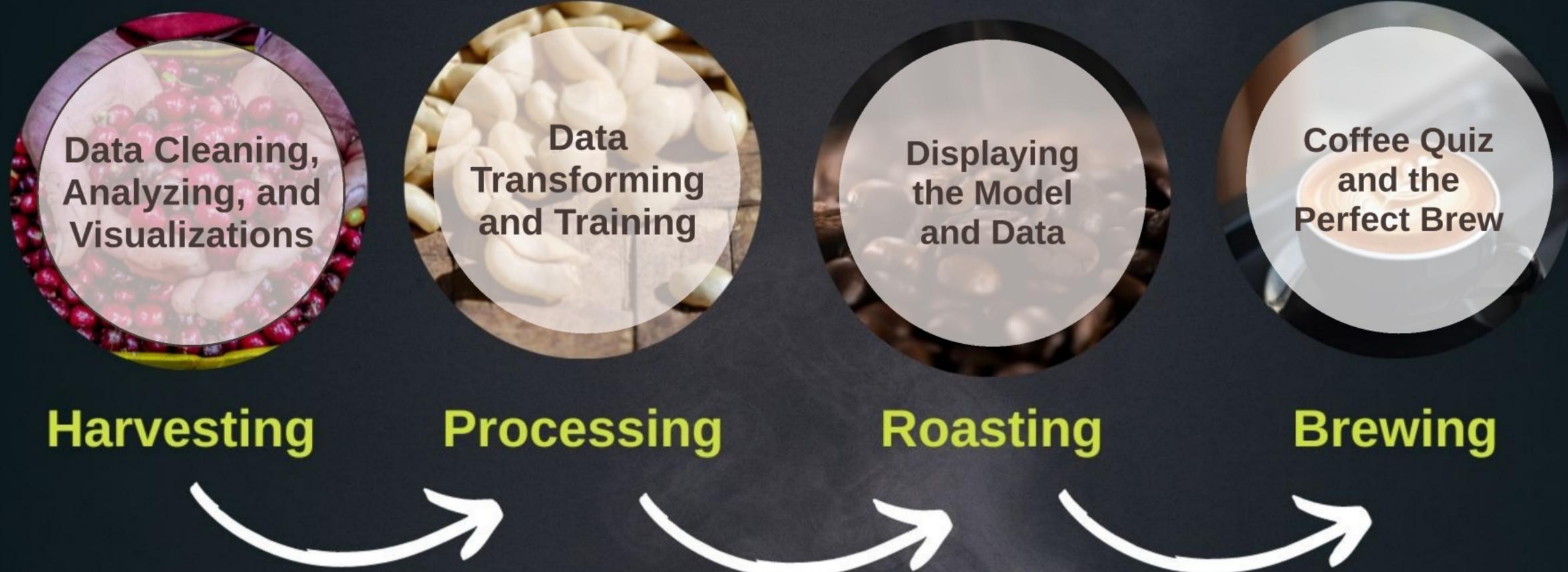
Dataset was scraped from coffeereview.com and contained over 7000 ratings and reviews of different roasted coffee. Dataset is from Kaggle

Cleaning and Transforming

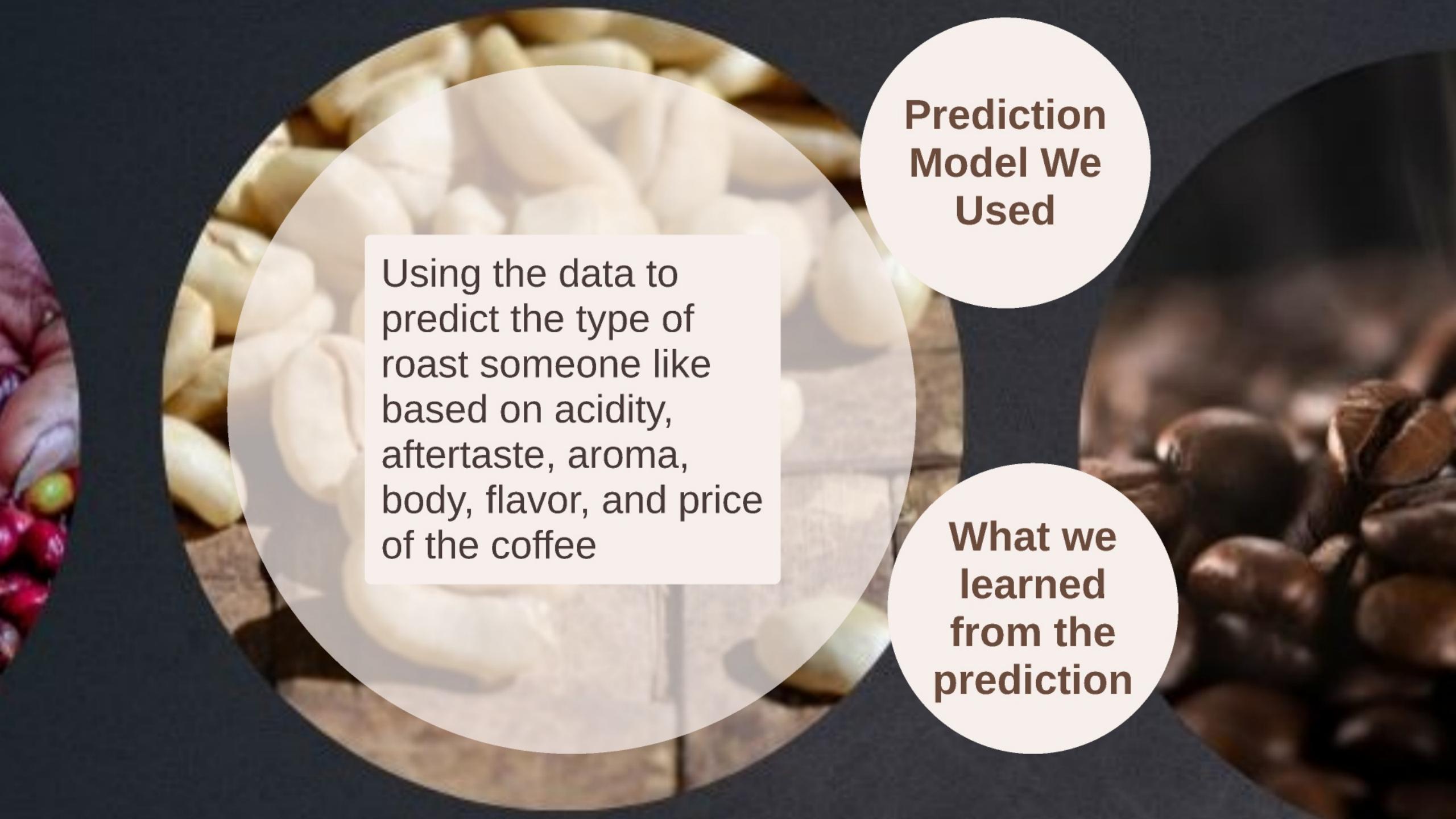
Data Visuals

# From the tree to the coffee cup

## Project 4 Machine Learning



Presented by Ankita Sarkar, Iqra Imam, Kim Sernett, Marissa Gallegos, Raelle Nalos, and Angel Toscano

A close-up photograph of coffee beans of various colors (green, yellow, brown) scattered on a dark surface.

## Prediction Model We Used

Using the data to predict the type of roast someone like based on acidity, aftertaste, aroma, body, flavor, and price of the coffee

## What we learned from the prediction

# Over Sampled Trials

```
# Instantiate the random oversampler model
# # Assign a random_state parameter of 1 to the model
oversample = RandomOverSampler(random_state=1)
# Fit the original training data to the random_oversampler model
X_over, y_over = oversample.fit_resample(X, y)

# Count the distinct values of the resampled labels data ...

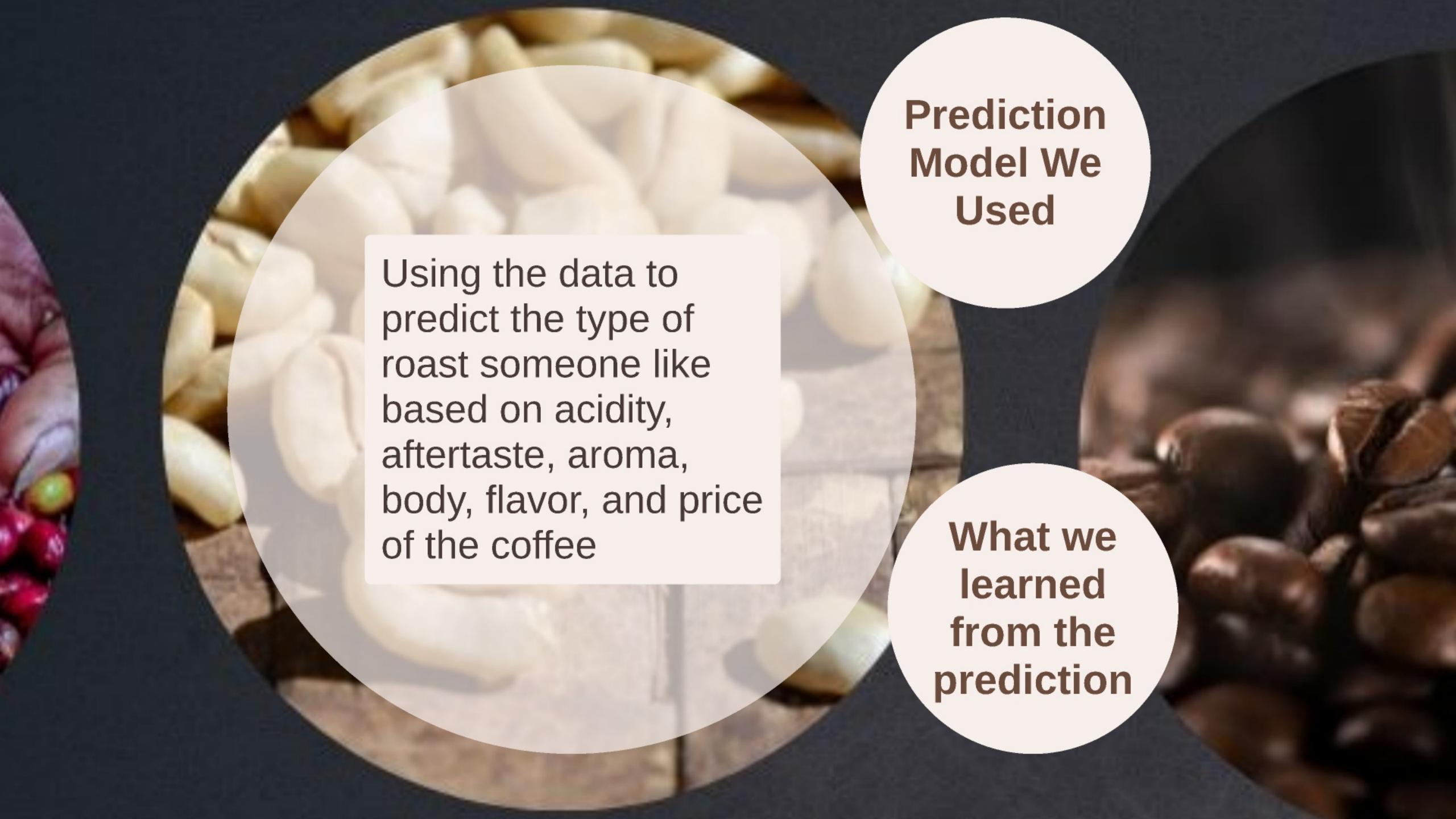
# Train_test_split the data
X_over_train, X_over_test, y_over_train, y_over_test = train_test_split(X_over,
                                                                    y_over,
                                                                    random_state=1,
                                                                    stratify=y_over)

# Create the MinMaxScaler instance
scaler = MinMaxScaler()

# Fit the MinMax Scaler with the training data
X_over_scaler = scaler.fit(X_over_train)

# Scale the training data
X_over_train_scaled = X_over_scaler.transform(X_over_train)
X_over_test_scaled = X_over_scaler.transform(X_over_test)

# Define the ML model
logreg_over= LogisticRegression(multi_class='multinomial', solver='lbfgs', random_state=1)
logreg_over
```

A close-up photograph of coffee beans of various colors (green, yellow, brown) scattered on a dark surface.

## Prediction Model We Used

Using the data to predict the type of roast someone like based on acidity, aftertaste, aroma, body, flavor, and price of the coffee

## What we learned from the prediction

```
# Score the model
print(f"Training Data Score: {logreg_over.score(X_over_train_scaled, y_over_train)}")
print(f"Testing Data Score: {logreg_over.score(X_over_test_scaled, y_over_test)}")
```

```
Training Data Score: 0.4567857771519099
```

```
Testing Data Score: 0.45617740232312565
```

```
# Predict outcomes for test data set
predictions = logreg_over.predict(X_over_test_scaled)
pd.DataFrame({"Prediction": predictions, "Actual": y_over_test})
```

	Prediction	Actual
1550	3	2
4664	1	4
75	1	2
5778	3	4
6482	5	5
...	...	...
5233	5	4
5671	1	4
1060	2	3
596	1	1
3238	1	1
1894 rows × 2 columns		

```
[33] # Display the accuracy score for the test dataset.
accuracy_score(y_over_test, predictions)
```

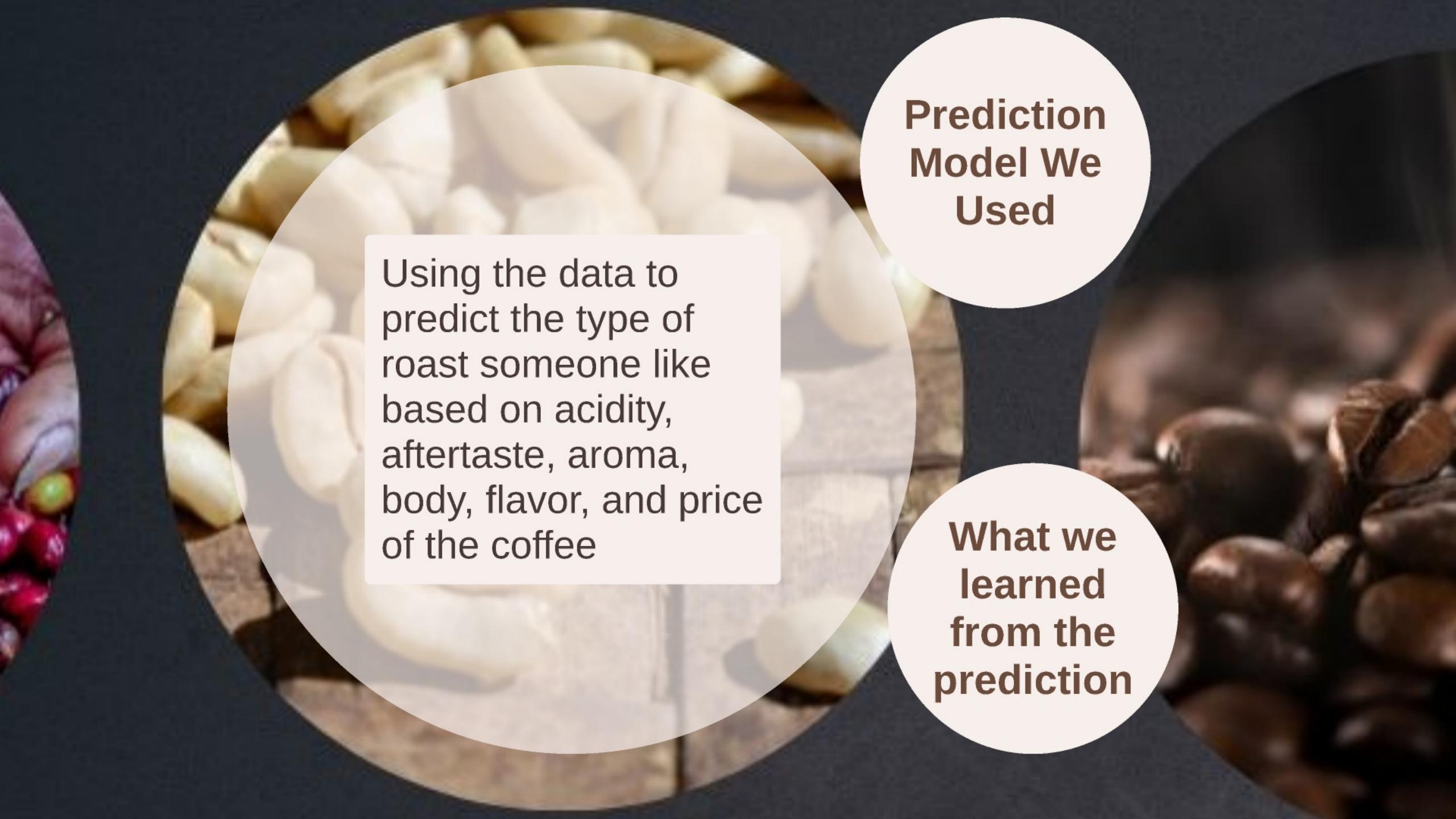
```
... 0.45617740232312565
```

```
[34] # Display the confusion matrix
confusion_matrix(y_over_test, predictions)
```

```
... array([[241,   83,   47,    3,    5],
       [202,   73,   81,   10,   13],
       [ 67,   75,  162,   28,   46],
       [ 55,   26,   93,   84,  121],
       [  0,   75,    0,    0, 304]], dtype=int64)
```

```
[35] # Print the classification report
target_names = ['Light', 'Medium-Light', 'Medium', 'Medium-Dark', 'Dark']
print(classification_report(y_over_test, predictions, target_names=target_names))
```

```
[35] ...          precision    recall  f1-score   support
                  ...
                  Light       0.43      0.64      0.51     379
        Medium-Light       0.22      0.19      0.21     379
                  Medium       0.42      0.43      0.43     378
        Medium-Dark       0.67      0.22      0.33     379
                  Dark       0.62      0.80      0.70     379
                  ...
            accuracy       0.46      0.46      0.44    1894
        macro avg       0.47      0.46      0.44    1894
    weighted avg       0.47      0.46      0.44    1894
```

A close-up photograph of coffee beans of various colors (green, yellow, brown) scattered on a dark surface.

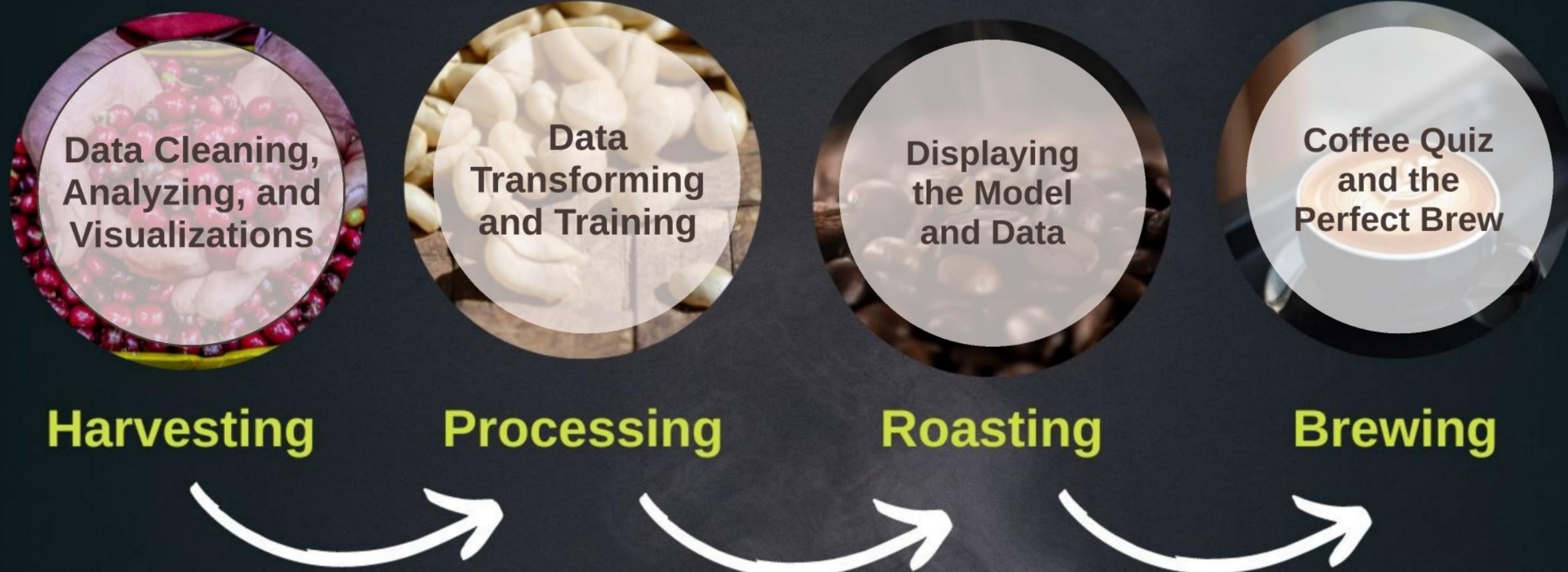
## Prediction Model We Used

Using the data to predict the type of roast someone like based on acidity, aftertaste, aroma, body, flavor, and price of the coffee

## What we learned from the prediction

# From the tree to the coffee cup

## Project 4 Machine Learning



Presented by Ankita Sarkar, Iqra Imam, Kim Sernett, Marissa Gallegos, Raelle Nalos, and Angel Toscano

A close-up photograph of dark coffee beans in the foreground, with a cup of coffee featuring a latte art design visible in the background.

Flask App

Javascript

HTML and  
CSS

Now That We Have  
A Machine That  
Makes Predictions  
We Need to Figure  
Out How to Display  
It To The User



```
Group Project 4 > coffee_venture > updated_ML_app > app.py > ...
  1  from flask import Flask, render_template, request, jsonify
  2  import pandas as pd
  3  import pickle
  4  import json
  5
  6  app = Flask(__name__)
  7
  8  # Load the ml model
  9  with open('logistic_regression_oversampled.pkl', 'rb') as f:
 10  |   log_reg_model = pickle.load(f)
 11
 12 # Load the coffee data into a Pandas dataframe
 13 coffee_data = pd.read_csv('coffee_clean_final.csv')
 14
 15 @app.route('/')
 16 def index():
 17 |   return render_template('coffee.html')
 18
 19 @app.route('/predict', methods=['POST'])
 20 def predict():
 21 |   # Get the values selected by the user in the HTML form
 22 |   print(f"the request form is {request.data}")
 23 |   form = request.data.decode('utf-8')
 24 |   form = json.loads(form)
 25 |   print(f'the dictionary form is {form}')
 26 |   acidity = form['acidity']
 27 |   aftertaste = form['aftertaste']
 28 |   aroma = form['aroma']
 29 |   body = form['body']
 30 |   flavor = form['flavor']
 31 |   price = form['price']
 32
 33
 34 input_data = [[acidity, aftertaste, aroma, body, flavor, price]]
 35 roast_prediction = int(log_reg_model.predict(input_data)[0])
 36
 37
 38
 39 if roast_prediction == 1:
 40 |   roast_prediction = "Light"
 41 elif roast_prediction == 2:
 42 |   roast_prediction = "Medium-Light"
 43 elif roast_prediction == 3:
 44 |   roast_prediction = "Medium"
 45 elif roast_prediction == 4:
 46 |   roast_prediction = "Medium-Dark"
 47 elif roast_prediction == 5:
 48 |   roast_prediction = "Dark"
 49 else:
```

A close-up photograph of dark coffee beans in the foreground, with two cups of coffee in the background. One cup is filled with a dark coffee, and the other is filled with a lighter-colored coffee, possibly latte or cappuccino.

**Flask App**

**Javascript**

**HTML and  
CSS**

**Now That We Have  
A Machine That  
Makes Predictions  
We Need to Figure  
Out How to Display  
It To The User**

```
Group Project // coffee-venture / updated_ml_app / static / -- concepts / coffee-venture-ml-app.js callback.js
```

```
1 // Function to update the slider value display
2 function updateSliderValue(slider, output) {
3   const sliderValue = document.getElementById(slider).value;
4   document.getElementById(output).innerHTML = sliderValue;
5 }
6
7 // Set initial slider values
8 updateSliderValue('acidity', 'acidity-value');
9 updateSliderValue('aftertaste', 'aftertaste-value');
10 updateSliderValue('aroma', 'aroma-value');
11 updateSliderValue('body', 'body-value');
12 updateSliderValue('flavor', 'flavor-value');
13 updateSliderValue('price', 'price-value');
14
15 // Update slider values when sliders are moved
16 document.querySelectorAll('input[type="range"]').forEach(slider => {
17   slider.addEventListener('input', () => {
18     const sliderId = slider.getAttribute('id');
19     const outputId = `${sliderId}-value`;
20     updateSliderValue(sliderId, outputId);
21   });
22 });
23
24 // Process form submission
25 const form = document.getElementById('coffee-form');
26 form.addEventListener('submit', event => {
27   console.log('hello')
28   event.preventDefault();
29
30
31   const preferences = {
32     acidity: parseInt(document.getElementById('acidity').value),
33     aftertaste: parseInt(document.getElementById('aftertaste').value),
34     aroma: parseInt(document.getElementById('aroma').value),
35     body: parseInt(document.getElementById('body').value),
36     flavor: parseInt(document.getElementById('flavor').value),
37     price: parseFloat(document.getElementById('price').value)
38   };
39
40   // Send coffee preferences to Flask app for prediction
41   fetch('/predict', {
42     method: 'POST',
43     body: JSON.stringify(preferences),
44     headers: {
45       'Content-Type': 'application/json'
46     }
47   })
48     .then(response => response.json())
49     .then(data => {
```

A close-up photograph of dark coffee beans in the foreground, with a cup of coffee featuring a latte art design visible in the background.

**Flask App**

**Javascript**

**HTML and  
CSS**

**Now That We Have  
A Machine That  
Makes Predictions  
We Need to Figure  
Out How to Display  
It To The User**

```
6      <title>Coffee Preference Quiz</title>
7      <link rel="stylesheet" href="../static/styles.css">
8      <style>
9          .container {
10              display: flex;
11              flex-direction: column;
12              align-items: center;
13              justify-content: center;
14              min-height: 100vh;
15          }
16
17      form {
18          display: flex;
19          flex-direction: column;
20          align-items: center;
21          justify-content: center;
22          margin: 0 auto;
23      }
24  </style>
25 </head>
26
27 <body>
28     <div class="container">
29         <header>
30             <h1>Coffee Preference Quiz</h1>
31
32         </header>
33         <main>
34             <div style="text-align: center;">
35                 <p>Select your coffee preferences:</p>
36             </div>
37             <form id="coffee-form">
38                 <label for="acidity" style="font-size: 25px"> Acidity</label>
39                 <label for="acidity"></label> (How tart or sour the coffee tastes) </label>
40                 <input type="range" class="slider" id="acidity" name="acidity" min="6" max="10">
41                 <span id="acidity-value">0</span><br><br>
42
43                 <label for="aftertaste" style="font-size: 25px"> Aftertaste</label>
44                 <label for="aftertaste"></label> (The lingering flavor in your mouth after you drink the
45                     coffee):</label>
46                 <input type="range" class="slider" id="aftertaste" name="aftertaste" min="6" max="10">
47                 <span id="aftertaste-value">0</span><br><br>
48
49                 <label for="aroma" style="font-size: 25px"> Aroma</label>
50                 <label for="aroma"></label> (The smell of the coffee)</label>
51                 <input type="range" class="slider" id="aroma" name="aroma" min="6" max="10">
52                 <span id="aroma-value">0</span><br><br>
53
54                 <label for="body" style="font-size: 25px"> Body</label>
```

A close-up photograph of dark coffee beans in the foreground, with a cup of coffee featuring a latte art design in the background.

Flask App

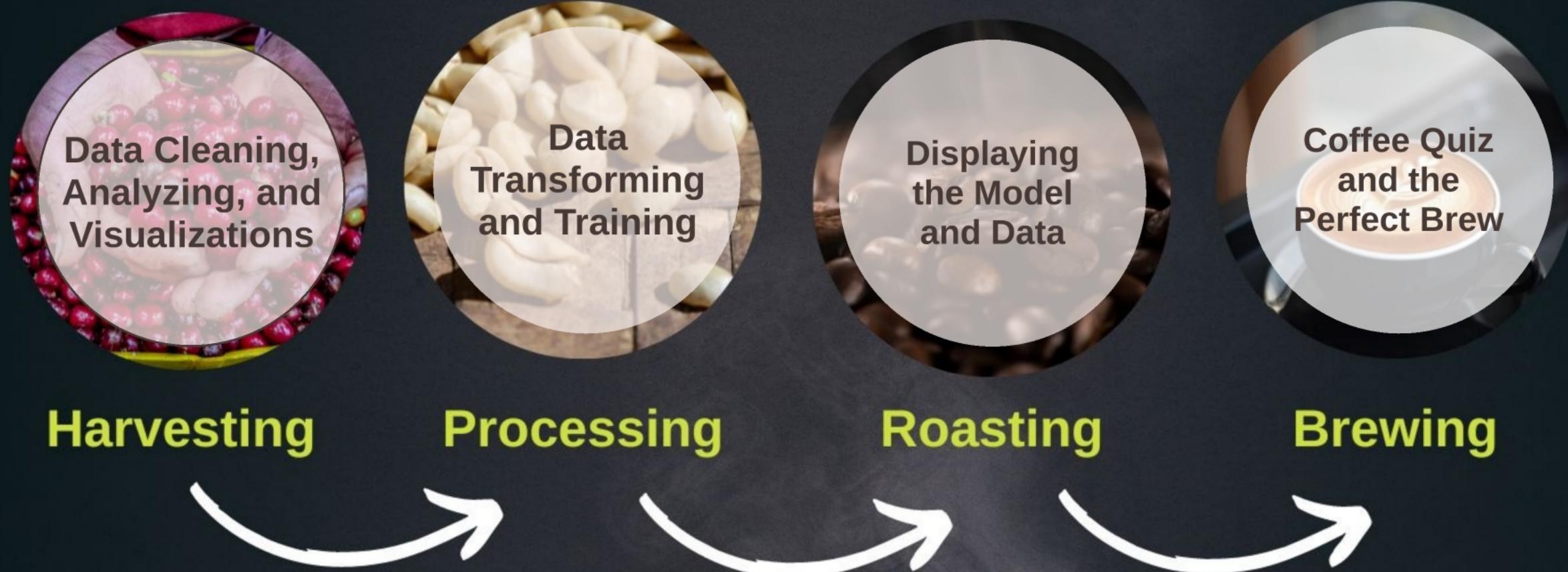
Javascript

Now That We Have  
A Machine That  
Makes Predictions  
We Need to Figure  
Out How to Display  
It To The User

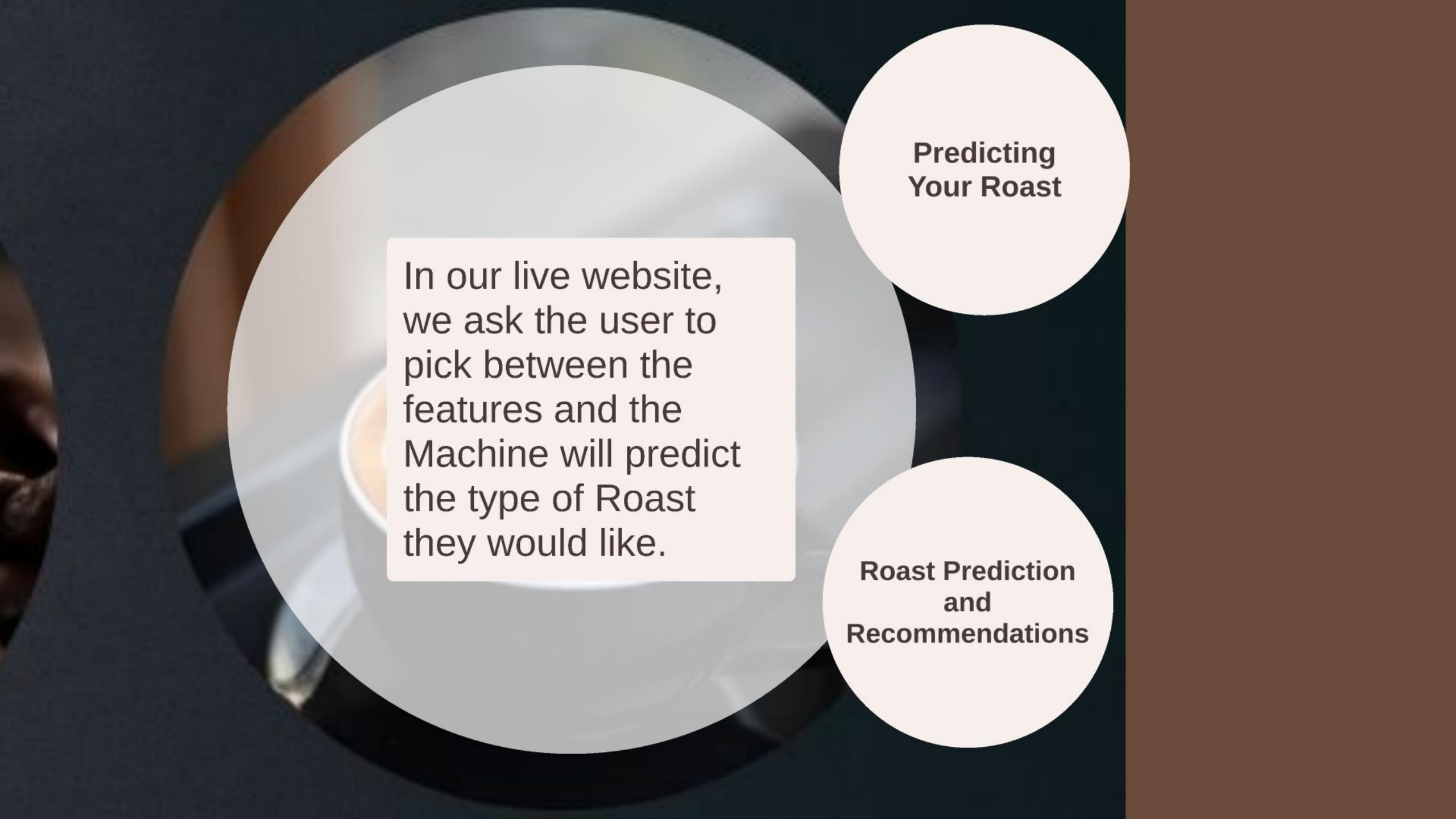
HTML and  
CSS

# From the tree to the coffee cup

## Project 4 Machine Learning



Presented by Ankita Sarkar, Iqra Imam, Kim Sernett, Marissa Gallegos, Raelle Nalos, and Angel Toscano



## Predicting Your Roast

In our live website, we ask the user to pick between the features and the Machine will predict the type of Roast they would like.

## Roast Prediction and Recommendations

# Coffee Preference Quiz

Select your coffee preferences:

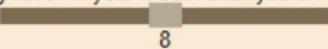
## Acidity

(How tart or sour the coffee tastes)



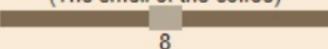
## Aftertaste

(The lingering flavor in your mouth after you drink the coffee):



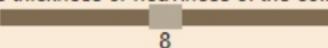
## Aroma

(The smell of the coffee)



## Body

(The thickness or heaviness of the coffee):



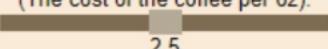
## Flavor

(The overall taste of the coffee with is the combined outcome of acidity, aftertaste, aroma and body):



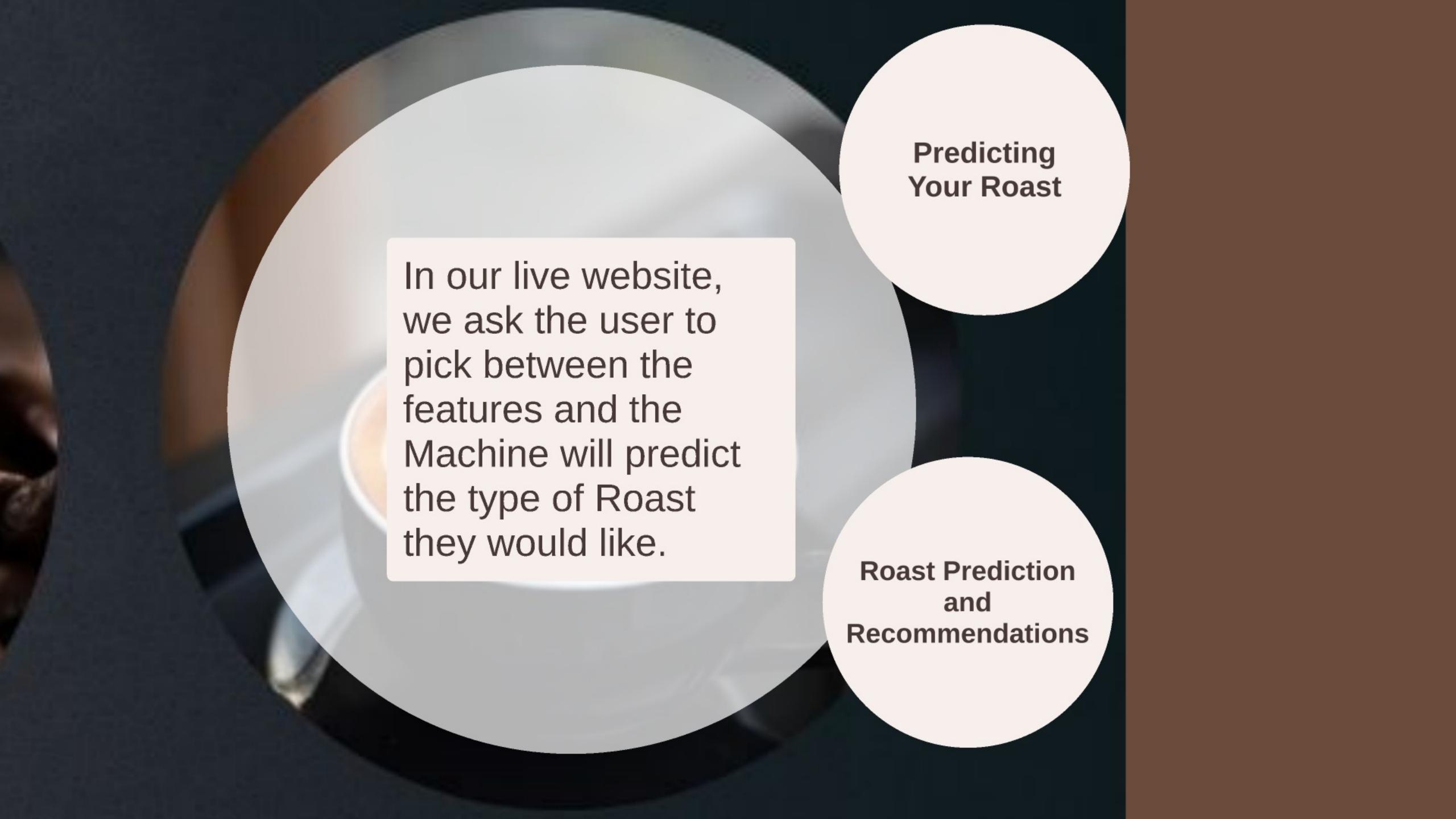
## Price

(The cost of the coffee per oz):



Brew

Enjoy a cup of coffee!



## Predicting Your Roast

In our live website, we ask the user to pick between the features and the Machine will predict the type of Roast they would like.

## Roast Prediction and Recommendations

You should try a **Light** coffee.

We recommend serving with breakfast or as a mild coffee for any time of day.

**These are the top three coffees we recommend you try:**

1. **Bucafe Rwanda** : This is a Deeply sweet, richly savory. Dried apricot, roasted cacao nib, freesia, almond, tamarind in aroma and cup. Balanced, sweet-savory structure; syrupy-smooth mouthfeel. The finish leads with notes of dried stone fruit and spicy florals in the short, rounding to almond and cocoa in the long, with a hint of tamarind-like sweet-tartness.

The acidity is 9. The aftertaste is 9. The aroma is 9. The body is 9. The flavor is 7.

The overall rating of this coffee is 93.

2. **Colombia Finca El Paraiso Geisha Letty** : This is a Ultra-high-toned, fruit-driven, richly floral. Wisteria, dried nectarine, pistachio brittle, calamansi lime, tarragon in aroma and cup. Deeply sweet structure with resoundingly bright, lively acidity; delicately silky, viscous mouthfeel. Long, floral-toned finish supported by bittersweet citrus.

The acidity is 9. The aftertaste is 9. The aroma is 10. The body is 9. The flavor is 10.

The overall rating of this coffee is 97.

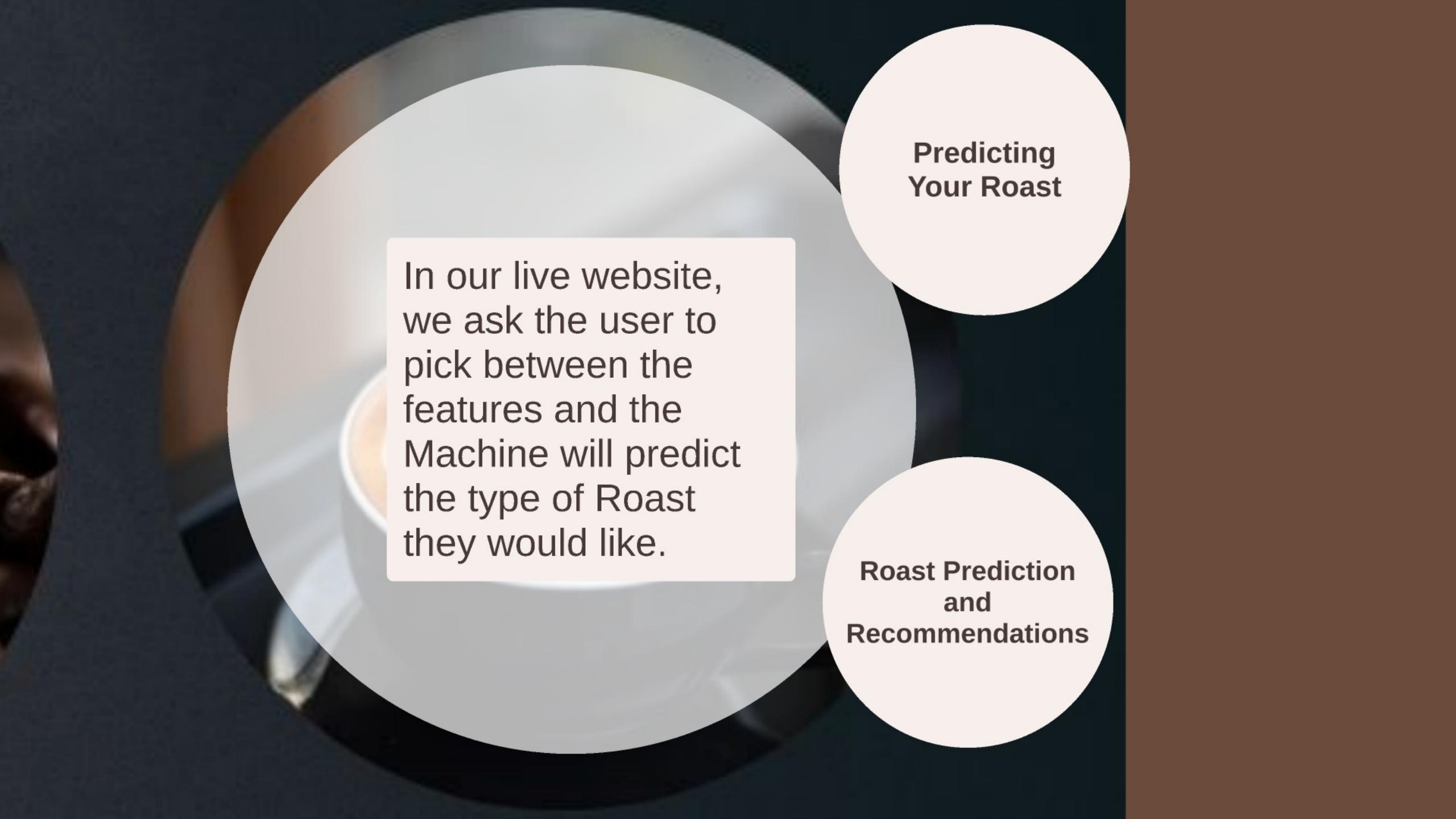
3. **Testi Ayla Double Ethiopia** : This is a High-toned, floral-driven, lyrically sweet. Wisteria, Keemun tea, white nectarine, fresh-cut basil, pink peppercorn in aroma and cup. Impeccably crystalline, balanced structure with buoyant acidity; delicately satiny mouthfeel. Very long, integrated, satisfying finish, with fruit and floral notes singing into the long.

The acidity is 10. The aftertaste is 9. The aroma is 9. The body is 9. The flavor is 10.

The overall rating of this coffee is 97.

**Enjoy a cup of coffee!**





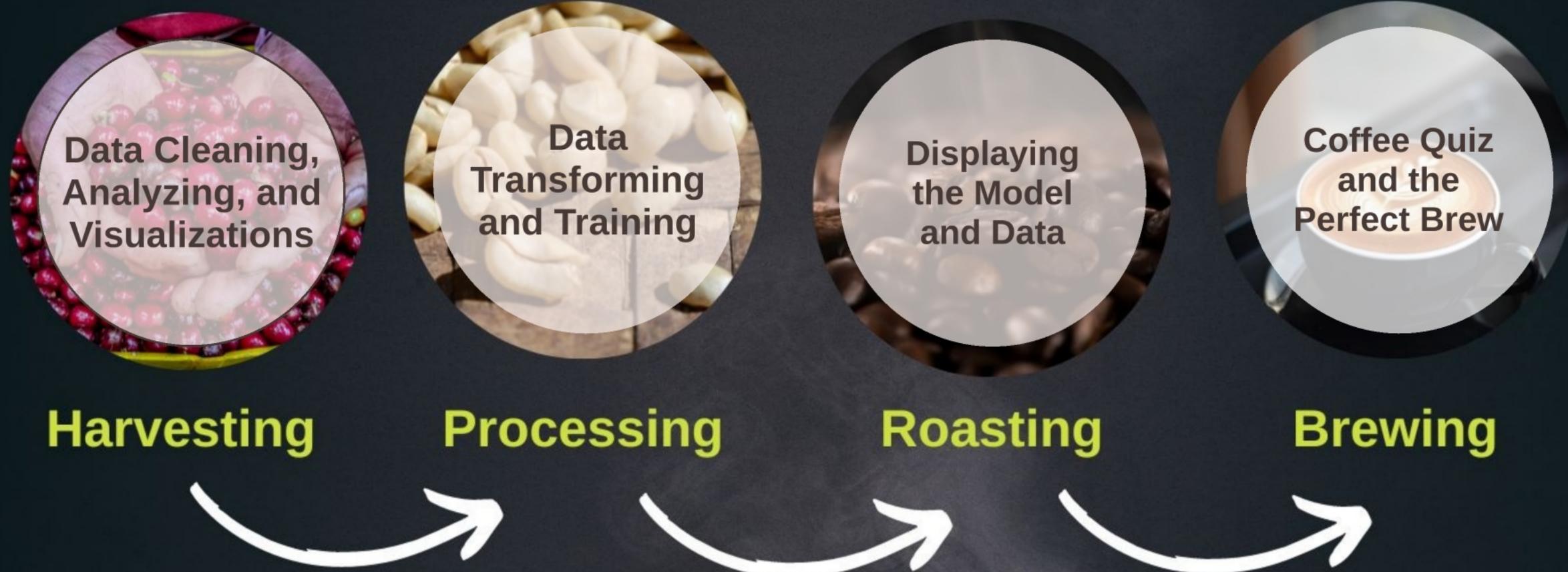
## Predicting Your Roast

In our live website, we ask the user to pick between the features and the Machine will predict the type of Roast they would like.

## Roast Prediction and Recommendations

# From the tree to the coffee cup

## Project 4 Machine Learning



Presented by Ankita Sarkar, Iqra Imam, Kim Sernett, Marissa Gallegos, Raelle Nalos, and Angel Toscano