Coffee Quality

Lead: Allyson McInnis
Kayli Aguilera, Annie Donnelly, Liberty Heise

This is our Best Bean TEAM

Allyson

Kayli

Annie

Liberty

Overview

01 Research

02 ETL

03 Machine Learning

04 Tableau

05 Analysis

01

Research

Research

Wine

Our first choice for our project was wine, but we could not find a dataset that had the information we wanted

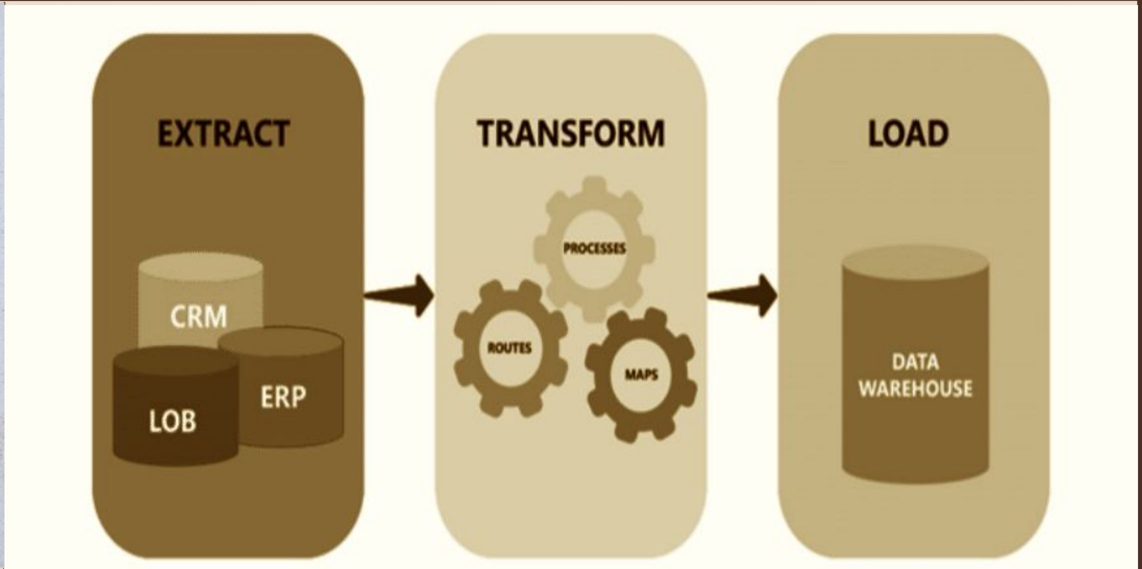
Chocolate

Our second choice was chocolate, but we ran into the same problem

Coffee

This dataset was just right!
It had the data that we needed for our project

Criteria



EXTRACT
CRM
ERP
LOB
TRANSFORM
PROCESSES
ROUTES
MAPS
LOAD
DATA
WAREHOUSE

02

ETL & SQL

Extract

Pulled data into editor

Transform

Cleaned out irrelevant information such as nulls and duplicates

Load

Loaded clean data into SQL

Extract

Extract

```python
# Import our dependencies
#from sklearn.model_selection import train_test_split
#from sklearn.preprocessing import StandardScaler
import pandas as pd
from datetime import datetime
import string
from sqlalchemy import create_engine

origin_df = pd.read_csv("../Resources/coffee_ratings.csv")
origin_df.head()
```

	total_cup_points	species	owner	country_of_origin	farm_name	lot_number	mill	ico_number	
0	90.58	Arabica	metad plc	Ethiopia	metad plc	NaN	metad plc	2014/2015	
1	89.92	Arabica	metad plc	Ethiopia	metad plc	NaN	metad plc	2014/2015	
2	89.75	Arabica	grounds for health admin	Guatemala	san marcos barrancas "san cristobal cuch	NaN	NaN	NaN	
3	89.00	Arabica	yidnekachew dabessa	Ethiopia	yidnekachew dabessa coffee plantation	NaN	wolensu	NaN	y
4	88.83	Arabica	metad plc	Ethiopia	metad plc	NaN	metad plc	2014/2015	

Transform

1

Transform

```
#choose the most important flavor criteria and make a dataframe
flavor_profile_df = origin_df[["total_cup_points", "aroma","flavor", "aftertaste", "acidity", "body", "balance", "sweetness", "moisture"]
flavor_profile_df
```

	total_cup_points	aroma	flavor	aftertaste	acidity	body	balance	sweetness	moisture
0	90.58	8.67	8.83	8.67	8.75	8.50	8.42	10.00	0.12
1	89.92	8.75	8.67	8.50	8.58	8.42	8.42	10.00	0.12
2	89.75	8.42	8.50	8.42	8.42	8.33	8.42	10.00	0.00
3	89.00	8.17	8.58	8.42	8.42	8.50	8.25	10.00	0.11
4	88.83	8.25	8.50	8.25	8.50	8.42	8.33	10.00	0.12

```
#choose the most important demographic & processing criteria and make a dataframe
demographic_df = origin_df[["country_of_origin", "owner", "harvest_year", "grading_date", "altitude","processing_method"]]
demographic_df
```

	country_of_origin	owner	harvest_year	grading_date	altitude	processing_method
0	Ethiopia	metad plc	2014	April 4th, 2015	1950-2200	Washed / Wet
1	Ethiopia	metad plc	2014	April 4th, 2015	1950-2200	Washed / Wet
2	Guatemala	grounds for health admin	NaN	May 31st, 2010	1600 - 1800 m	NaN
3	Ethiopia	yidnekachew dabessa	2014	March 26th, 2015	1800-2200	Natural / Dry
4	Ethiopia	metad plc	2014	April 4th, 2015	1950-2200	Washed / Wet

Transform

2

```
# convert the 'Date' column to datetime format
demographic_df['grading_date']= pd.to_datetime(demographic_df['grading_date'])
# Check the format of 'Date' column
demographic_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1002 entries, 0 to 1336
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   country_of_origin  1002 non-null   object
 1   owner              1002 non-null   object
 2   harvest_year       1002 non-null   object
 3   grading_date       1002 non-null   datetime64[ns]
 4   altitude           1002 non-null   object
 5   processing_method  1002 non-null   object
dtypes: datetime64[ns](1), object(5)
memory usage: 54.8+ KB
```

```
demographic_df['grading_year'] = pd.DatetimeIndex(demographic_df['grading_date']).year
demographic_df
```

```
C:\Users\a_don\AppData\Local\Temp\ipykernel_22196\102522987.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexin
  demographic_df['grading_year'] = pd.DatetimeIndex(demographic_df['grading_date']).year
```

	country_of_origin	owner	harvest_year	grading_date	altitude	processing_method	grading_year
0	Ethiopia	metad plc	2014	2015-04-04	1950-2200	Washed / Wet	2015
1	Ethiopia	metad plc	2014	2015-04-04	1950-2200	Washed / Wet	2015
3	Ethiopia	yidnekachew dabessa	2014	2015-03-26	1800-2200	Natural / Dry	2015
4	Ethiopia	metad plc	2014	2015-04-04	1950-2200	Washed / Wet	2015
9	Ethiopia	diamond enterprise plc	2014	2015-03-30	1795-1850	Natural / Dry	2015

Transform

3

- What's happening here?

```
#  Import and read the charity_data.csv.
cleaned_df = pd.read_csv("../Resources/New_ETL.csv")
cleaned_df.head()
```

	Unnamed: 0	country_of_origin	owner	harvest_year	grading_date	altitude	processing_method	grading_year
0	0	Ethiopia	metad plc	2014	2015-04-04	1985	Washed / Wet	2015
1	1	Ethiopia	metad plc	2014	2015-04-04	1985	Washed / Wet	2015
2	3	Ethiopia	yidnekachew dabessa	2014	2015-03-26	1985	Natural / Dry	2015
3	4	Ethiopia	metad plc	2014	2015-04-04	1985	Washed / Wet	2015
4	9	Ethiopia	diamond enterprise plc	2014	2015-03-30	1800	Natural / Dry	2015

harvest_year
2014
2014
NA
2014
2014
2013
2012
March 2010
March 2010
2014
2014
2014
2014
Sept 2009 - April 2010
March 2010
2014
May-August
2009/2010

altitude
1950-2200
1950-2200
1600 - 1800 m
1800-2200
1950-2200
NA
NA
1570-1700
1570-1700
1795-1850
1855-1955
meters above sea level: 1.872
meters above sea level: 1.943
2000 ft
1570-1700
meters above sea level: 2.080
1200-1800m
NA
1450
1700-2000m
meters above sea level: 2.019
1300 msnm
1320
meters above sea level: 2.112

Transform

4

```
cleaned_df["year_diff"] = cleaned_df["grading_year"]- cleaned_df["harvest_year"]
cleaned_df
```

	Unnamed: 0	country_of_origin	owner	harvest_year	grading_date	altitude	processing_method	grading_year	year_diff
0	0	Ethiopia	metad plc	2014	2015-04-04	1985	Washed / Wet	2015	1
1	1	Ethiopia	metad plc	2014	2015-04-04	1985	Washed / Wet	2015	1
2	3	Ethiopia	yidnekachew dabessa	2014	2015-03-26	1985	Natural / Dry	2015	1
3	4	Ethiopia	metad plc	2014	2015-04-04	1985	Washed / Wet	2015	1
4	9	Ethiopia	diamond enterprise plc	2014	2015-03-30	1800	Natural / Dry	2015	1

df > csv

Load

- Concatinate the two dataframes
- Push to PostgreSQL
- Have usable tables to make data into Tableau tables

	Unnamed: 0.1 bigint	Unnamed: 0 bigint	country_of_origin text	owner text	harvest_year bigint	grading_date text	altitude bigint	processing_method text	grading_year bigint	year_diff bigint
1	0	0	Ethiopia	metad plc	2014	2015-04-04	1985	Washed / Wet	2015	1
2	1	1	Ethiopia	metad plc	2014	2015-04-04	1985	Washed / Wet	2015	1
3	2	3	Ethiopia	yidnekac...	2014	2015-03-26	1985	Natural / Dry	2015	1
4	3	4	Ethiopia	metad plc	2014	2015-04-04	1985	Washed / Wet	2015	1
5	4	9	Ethiopia	diamond ...	2014	2015-03-30	1800	Natural / Dry	2015	1
6	5	10	Ethiopia	mohamm...	2014	2015-03-27	1900	Natural / Dry	2015	1
7	6	11	United States	cqi q coff...	2014	2015-03-13	2	Washed / Wet	2015	1
8	7	12	United States	cqi q coff...	2014	2015-03-13	2	Washed / Wet	2015	1
9	8	15	United States	cqi q coff...	2014	2015-03-13	2	Washed / Wet	2015	1
10	9	18	China	yunnan c...	2015	2016-04-07	1450	Washed / Wet	2016	1
11	10	19	Ethiopia	essencec...	2014	2015-03-25	1850	Natural / Dry	2015	1
12	11	20	United States	cqi q coff...	2014	2015-03-13	2	Washed / Wet	2015	1
13	12	21	Costa Rica	the coffe...	2014	2014-04-02	1300	Washed / Wet	2014	0

	Unnamed: 0 bigint	total_cup_points double precision	aroma double precision	flavor double precision	aftertaste double precision	acidity double precision	body double precision	balance double precision	sweetness double precision	moisture double precision
1	0	90.58	8.67	8.83	8.67	8.75	8.5	8.42	10	0.12
2	1	89.92	8.75	8.67	8.5	8.58	8.42	8.42	10	0.12
3	2	89.75	8.42	8.5	8.42	8.42	8.33	8.42	10	0
4	3	89	8.17	8.58	8.42	8.42	8.5	8.25	10	0.11
5	4	88.83	8.25	8.5	8.25	8.5	8.42	8.33	10	0.12
6	5	88.83	8.58	8.42	8.42	8.5	8.25	8.33	10	0.11
7	6	88.75	8.42	8.5	8.33	8.5	8.25	8.25	10	0.11
8	7	88.67	8.25	8.33	8.5	8.42	8.33	8.5	9.33	0.03
9	8	88.42	8.67	8.67	8.58	8.42	8.33	8.42	9.33	0.03
10	9	88.25	8.08	8.58	8.5	8.5	7.67	8.42	10	0.1
11	10	88.08	8.17	8.67	8.25	8.5	7.75	8.17	10	0.1
12	11	87.92	8.25	8.42	8.17	8.33	8.08	8.17	10	0
13	12	87.92	8.08	8.67	8.33	8.42	8	8.08	10	0

03

Machine Learning

Machine Learning

Elbow Graph

We used an elbow graph to visualize our clusters.

Regression Models

When the regressors showed that the coffee dataset could possibly be predictive it was applied to a deep neural network.

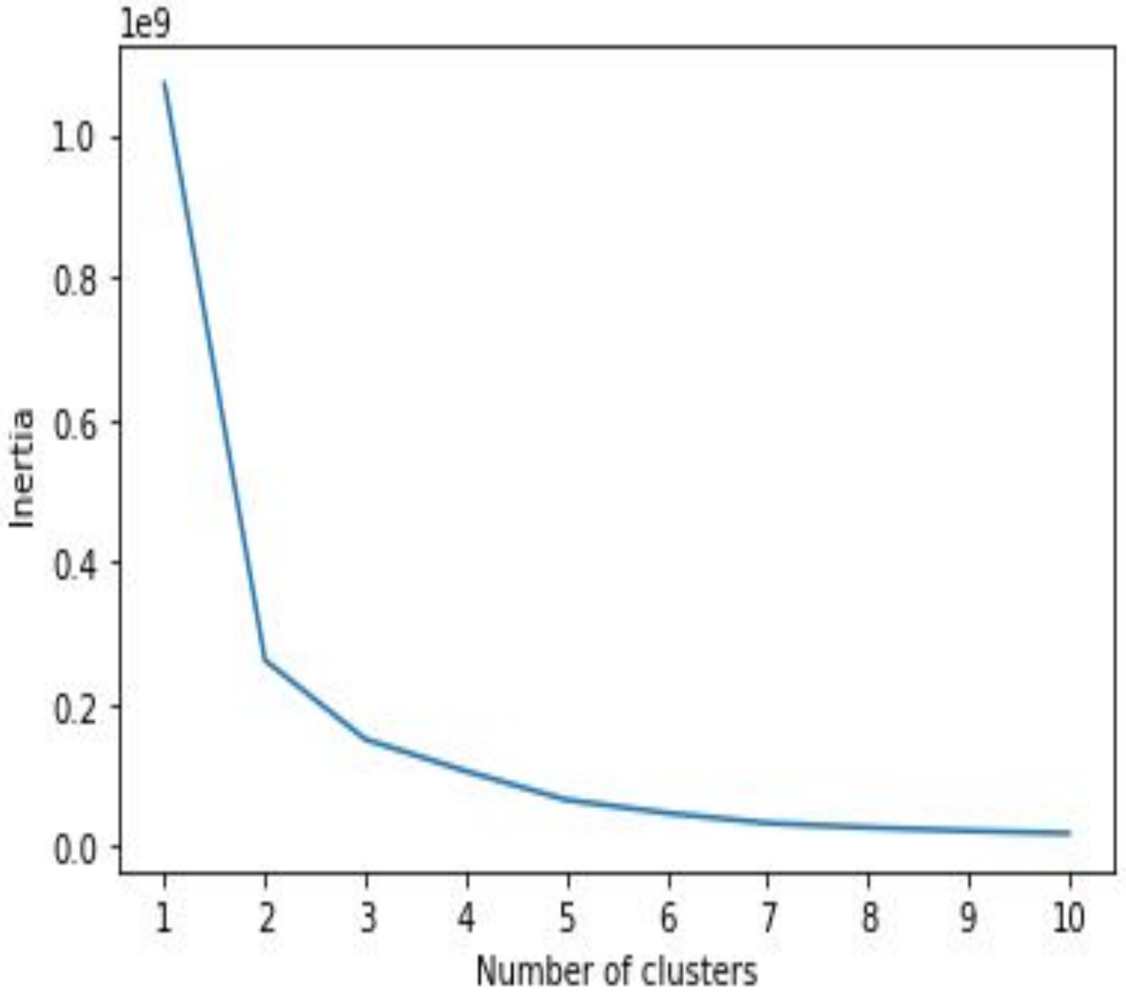
Neural Network

The deep neural network was used to see what predictions, or if we could make predictions based upon the data that we had

Trial & Error

We ran into many problems with the data and models.

Elbow Curve



Regressions

Model:
LinearRegression

Train score:
0.9420567543856417
Test Score:
0.9583597279014427

Model:
RandomForestRegressor

Train score:
0.9976956754114562
Test Score:
0.982460835724622

Model:
AdaBoostRegressor

Train score:
0.9786443326765295
Test Score:
0.970629358520981

Model:
KNeighborsRegressor

Train score:
0.9538354395245838
Test Score:
0.9555396940320356

Model:
ExtraTreesRegressor

Train score:
0.9999999983617538
Test Score:
0.9826364880349643

Model:
SVR

Train score:
0.968402170682052
Test Score:
0.9516711703813926

Loss & Accuracy

Coffee	Loss: -1237.0526123046875	Accuracy: 0.0
Flavor	Loss: -1237.0526123046875	Accuracy: 0.0
Demographic	Loss: -845.5838012695312	Accuracy: 0.0

The Issue

	harvest_year	altitude	grading_year	country_id	owner_id	method_id	total_cup_points	aroma	flavor	aftertaste	acidity	body	balance	sweetness	moisture
0	2014	1950-2200	2015	1	1	1	90.58	8.67	8.83	8.67	8.75	8.50	8.42	10.0	0.12
1	2014	1950-2200	2015	1	1	1	89.92	8.75	8.67	8.50	8.58	8.42	8.42	10.0	0.12
2	2014	1800-2200	2015	1	2	2	89.75	8.42	8.50	8.42	8.42	8.33	8.42	10.0	0.00
3	2014	1950-2200	2015	1	1	1	89.00	8.17	8.58	8.42	8.42	8.50	8.25	10.0	0.11
4	2014	1795-1850	2015	1	3	2	88.83	8.25	8.50	8.25	8.50	8.42	8.33	10.0	0.12
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
997	2015	1000	2016	33	267	2	81.08	7.42	7.42	7.25	7.58	6.92	7.17	10.0	0.06
998	2013	750m	2013	33	268	2	81.00	7.25	7.25	7.17	7.50	7.33	7.17	10.0	0.11
999	2013	750m	2013	33	268	2	81.00	7.42	7.08	7.08	7.33	7.25	7.58	10.0	0.00
1000	2012	3000'	2012	2	268	2	81.00	7.33	7.17	7.17	7.67	7.33	7.17	10.0	0.11
1001	2014	795 meters	2014	2	269	2	81.00	7.42	7.25	7.17	7.50	7.25	7.17	10.0	0.09

04

Tableau

Coffee Characteristics

Aroma	The intensity of smell once the coffee has been freshly brewed
Body	The intensity of how the coffee feels in the mouth in terms of weight
Flavor	The taste of the coffee when it enters the mouth
Acidity	Term used in cupping that describes the flavors, tartness, and vigorous taste
Sweetness	When cupping coffee, the intensity of sugariness that is present when swooshing in the mouth
Aftertaste	The intensity of the flavor and the smell of the coffee once it has been tasted and spit out

TOTAL COFFEE CUPPING QUALITY SCORE		
90 - 100	OUTSTANDING	SPECIALTY COFFEE
85 - 89.99	EXCELLENT	
80 - 84.99	VERY GOOD	
< 80.0	BELOW SPECIALTY COFFEE QUALITY	NOT SPECIALTY COFFEE

Searching for the best cup of coffee, in Tableau

05

Analysis

Final Analysis + Take Away

- Our data was skewed towards collecting data on high ranked coffee from a variety of countries.
- Our data was linear which means our data could have simply shown it's cards during multiple linear regressions.
- Our data was not complex enough to benefit from neural networks and machine learning.
- Linear Regression Analysis should always come first in order to determine if the data needs further Machine Learning analysis. Our scores for LR were very high, versus our scores in the ML, which were extremely low.
- A deeper look into the data is always important to gather general information and determine how the data interacts.
- Using different modes of analysis help inform and improve intuition and understanding of data.
- Next time, we would test for linear correlations first to ensure balanced data and choose a different data set to demonstrate machine learning at it's best.

Coffee is Good, Coffee is Great,
Enjoy your Coffee, Life can Wait