

Fiinal_Project_Team5(Safe to eat or deadly poison?)

1. Pick a dataset - mushrooms(predict if it is poisonous or not)

2. Store it in buckets within AWS

3. Clean data in jupyter notebook and store it in postgres

4. ML - we could train the model (log. regression) in jupyter notebook and then store the results in table or back in postgres

5. Use html, javascript to do an page that can be filtered to determine safe or unsafe to eat(example -UFO), D3

User choses filters js reads what they choose and save it in a temp location and then have python run based of event listener(click) and then spit out on website

Listener in js would run the python file and then return the answer

6. use flask and html to display the results in webpage

7. Do we need any graphs and what of?

Viz of the categories that present most in poisonous

8.

9.

10.

Room 1 and Room 5: you both asked about how to get a .py file to run in Javascript in order for a user filter to run an ML model on demand and spit out the results to be displayed on the front end

it looks like the best way to do this is something we didn't learn in this class:

<https://stackoverflow.com/questions/13175510/call-python-function-from-javascript-code>

From the `document.getElementsByTagName` I guess you are running the javascript in a browser.

26 The traditional way to expose functionality to javascript running in the browser is calling a remote URL using AJAX. The X in AJAX is for XML, but nowadays everybody uses JSON instead of XML.

For example, using jQuery you can do something like:

```
$.getJSON('http://example.com/your/webservice?param1=x&param2=y',
function(data, textStatus, jqXHR) {
    alert(data);
})
```

You will need to implement a python webservice on the server side. For simple webservices I like to use [Flask](#).

A typical implementation looks like:

```
@app.route("/your/webservice")
def my_webservice():
    return jsonify(result=some_function(**request.args))
```

You can run IronPython (kind of Python.Net) in the browser with [silverlight](#), but I don't know if NLTK is available for IronPython.

Page 1

Headers - clickable - eg

Home

MarketplaceBeanManufacturingDataMap

Title

About page/use

Filter to determine poison - eg

Marketplace

Search the companies

Filter Search								
<div>Enter a Year (2006-2016)</div> <div>2014</div> <div>Enter a County of the company location</div> <div></div> <div>Enter a Company Name</div> <div></div> <div>Enter a Rating</div> <div></div> <div>Filter Table</div>								
Company	Bean Origin City	Review Year	Cocoa %	Company Location	Rating	Bean Type	Bean Origin Country	
A. Morin	Carenero	2014	70%	France	2.75	Criollo	Venezuela	
A. Morin	Sur del Lago	2014	70%	France	3.5	Criollo	Venezuela	
A. Morin	Puerto Cabello	2014	70%	France	3.75	Criollo	Venezuela	
A. Morin	Madagascar	2013	70%	France	3	Criollo	Madagascar	
A. Morin	Chuao	2013	70%	France	4	Trinitario	Venezuela	
Acalli	Tumbes, Norandino	2015	70%	U.S.A.	3.75	Criollo	Peru	
Adi	Vanua Levu	2011	60%	Fiji	2.75	Trinitario	Fiji	
Adi	Vanua Levu, Toto-A	2011	80%	Fiji	3.25	Trinitario	Fiji	
Adi	Vanua Levu	2011	88%	Fiji	3.5	Trinitario	Fiji	
Adi	Vanua Levu, Ami-Ami-CA	2011	72%	Fiji	3.5	Trinitario	Fiji	
Aequare (Gianduja)	Los Rios, Quevedo, Arriba	2009	55%	Ecuador	2.75	Forastero (Arriba)	Ecuador	
Aequare (Gianduja)	Los Rios, Quevedo, Arriba	2009	70%	Ecuador	3	Forastero (Arriba)	Ecuador	
Ah Cacao	Tabasco	2009	70%	Mexico	3	Criollo	Mexico	

Page 2

See list of poisonous with the filter D3 eg

Viz of common things in poisonous

Manufacturing Dash

Use the interactive charts below to explore the data

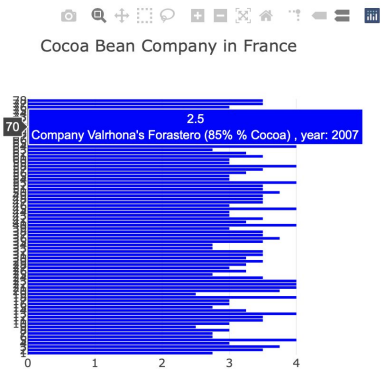
Company Country Location

France

Cocoa Bean Info

Comany Location : France

Average Rating : 3.33



Grading rubric

Requirement	points	notes
Web programming technologies (HTML/CSS/Flask)	25	
Machine Learning technologies (sklearn/pyspark)	25	
At least 2 of the explicitly listed technologies	10	Pandas, Matplotlib, Bootstrap, Plotly, D3.js, Leaflet, SQL Database, MongoDB Database, Google Cloud SQL, Amazon AWS, Tableau
Publicly availability via webhosting (Heroku)	5	
Presentation	20	
Slide Deck	15	