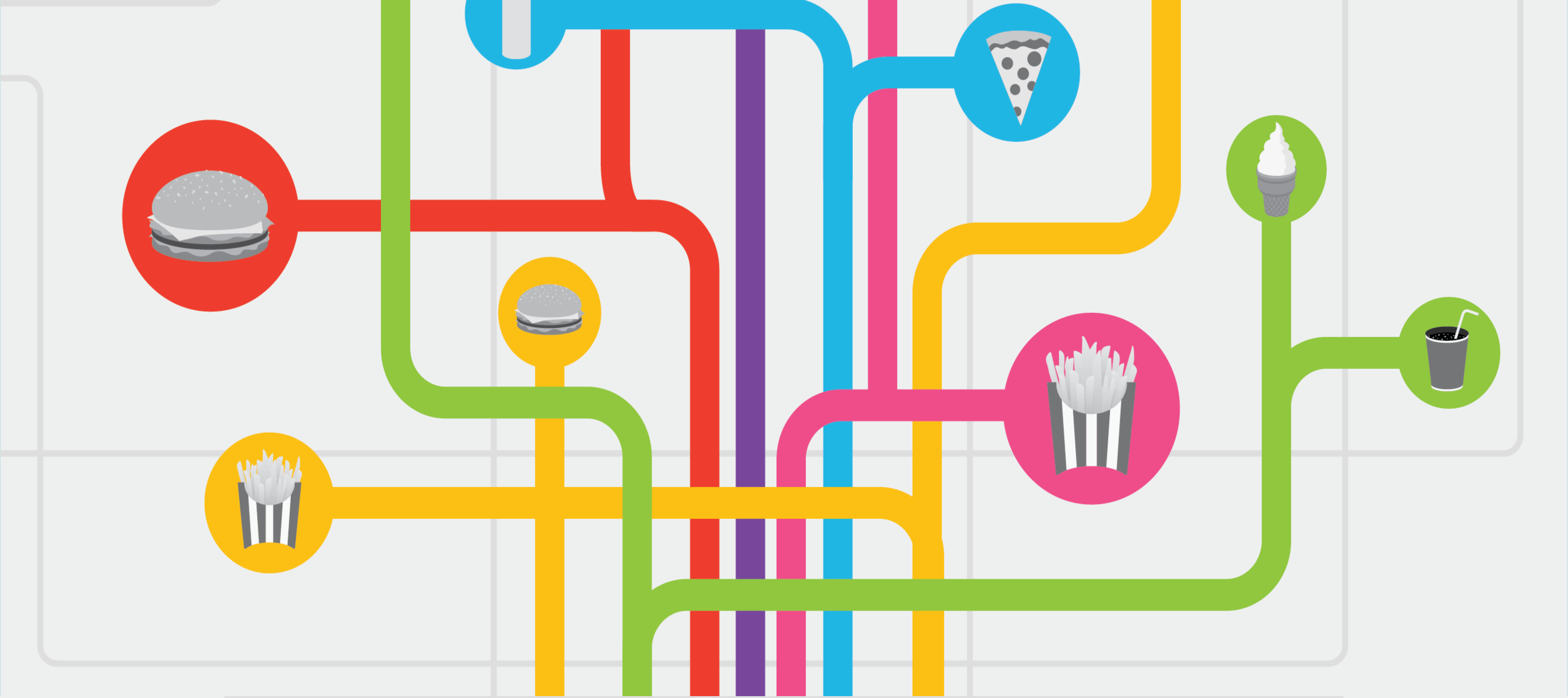
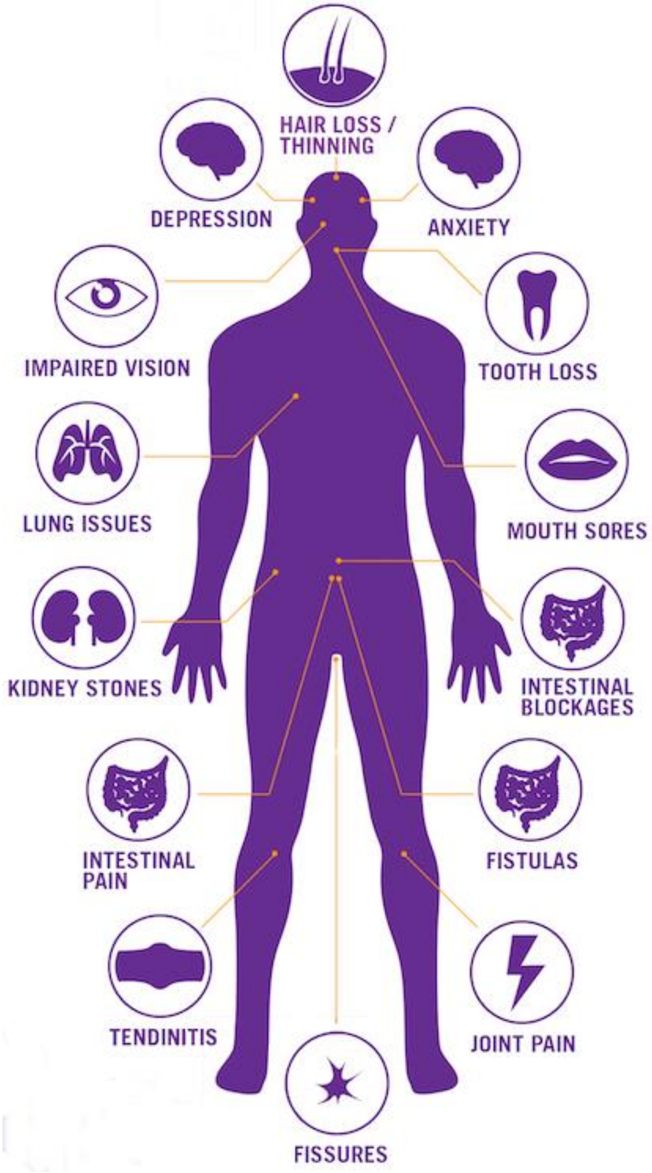


DETERMINING IBD TRIGGER FOODS USING MACHINE LEARNING AND PYTHON

WHAT'S IBD?

- Inflammatory bowel disease (IBD) describes a group of conditions, including Ulcerative Colitis (UC) and Crohn's disease (CD), impacting 1.6 million people in the US alone.
- Characterized by "gut" inflammation.
- Symptoms range from mild annoyances to life-threatening issues (blockages, cancer).
- Autoimmune, caused by a combination of genetic and environment factors.



WHAT'S FOOD GOT TO DO WITH IT?

- While foods' relationship with IBD remains understudied and controversial…
- …57% of IBD sufferers think diet can trigger symptom flare…
- …leading to food avoidance/malnourishment.
- Safe foods are thought to be person specific, in contrast to diseases like Celiac or lactose intolerance, where food issues are known.

WHY IT'S PERSONAL TO ME?

- In February 2016 I was diagnosed with Crohn's disease… and 10 ulcers.
- Medication has me <u>ulcer free, but not symptom free</u>.
- Certain foods can trigger flares lasting weeks.
- Trial and error to find safe foods is painful and takes a long time.

Real ulcers are gross, so here's some clipart:

You're welcome.

GOAL – WHAT CAN IBD SUFFERERS EAT?

1) Sub-clusters of diet?

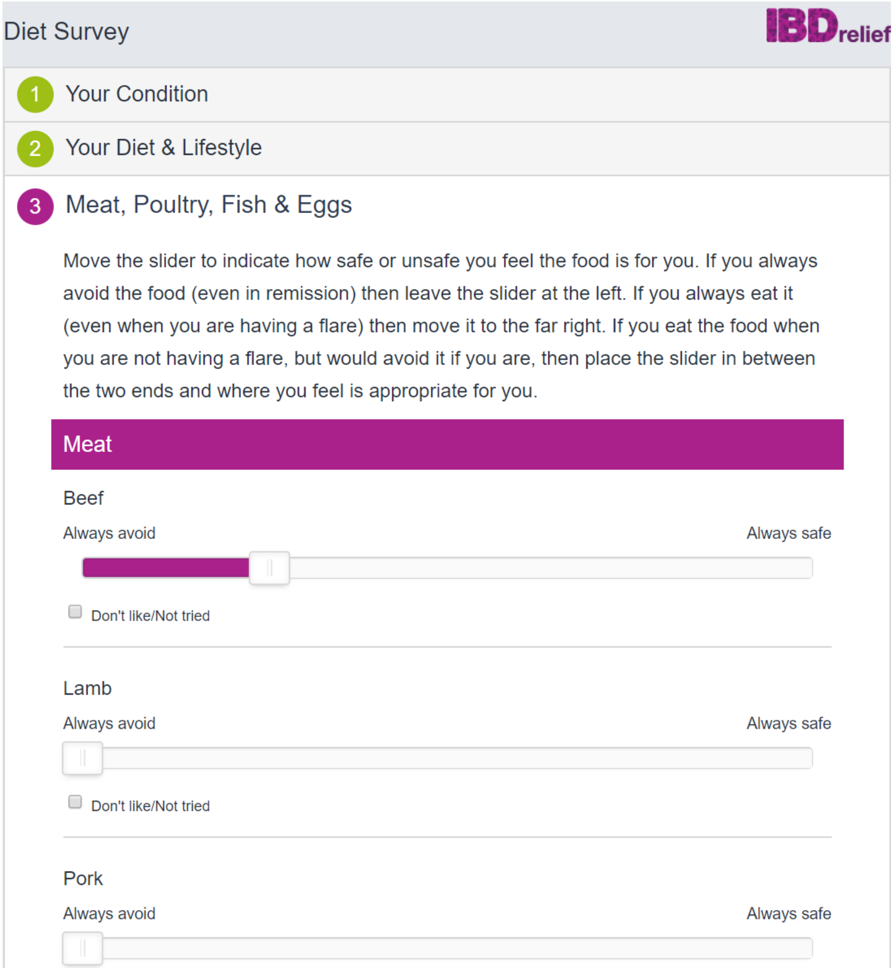
2) Relationships between individual foods or groups of foods?

3) Nutrients that impact food tolerance?

4) Can food tolerance/intolerance be predicted with a reasonable degree of accuracy for an IBD sufferer with only a few "known" safe/unsafe foods?

MATERIALS

- **Small data set: 670, 250-food survey responses** from IBD sufferers about food tolerances. 570 usable.
- **Nutrient information for each surveyed food** from the USDA's nutrient database API.
- **Python** 3.6.1 and **Jupyter Notebook**
  - Analysis: apyori, numpy, pandas, PyFIM, scikit-learn, scipy, sqlite
  - Visualization: graphviz, matplotlib, seaborn



The [online] survey utilizes a sliding scale to accept answer inputs, which are stored as integer values in a range from 0 through 10. A checkbox for each question gives the option to not answer questions individually.

Check out Introduction to Data Mining by Tan, Steinbach, and Kumar, Chapter 6 for an introduction to the basic concepts (free online).

ANALYSIS – ASSOCIATION RULE LEARNING

- A rule-based machine learning method for discovering interesting patterns between variables in large databases, in a human-understandable way. Two steps:
  - **Frequent Itemset Mining (FIM).** Find all "frequent" subsets, generally as measured by a Support threshold.
  - **Rule Generation.** Generate "interesting" rules, commonly as measured by Confidence and Lift.
- **Uses:** market basket analysis, web mining, document analysis, telecommunication alarm diagnosis, network intrusion detection, bioinformatics

FP-GROWTH FOR THE EFFICIENCY WIN

- Brute forcing FIM is exponential - O(2^n)
- FP-Growth is quadratic - O(n^2)
  1. [Iteratively] build compact data structure
  2. [Recursively] extract frequent itemsets
- Downside: Complicated
  - Many wrong implementations in Python
  - Used PyFIM – some limitations, but accurate

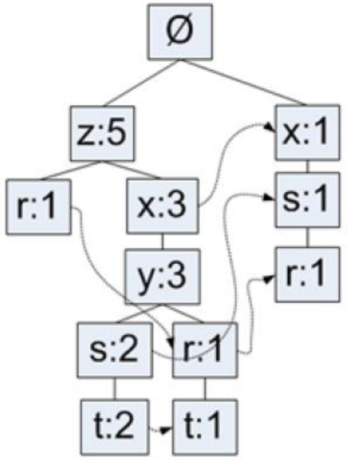
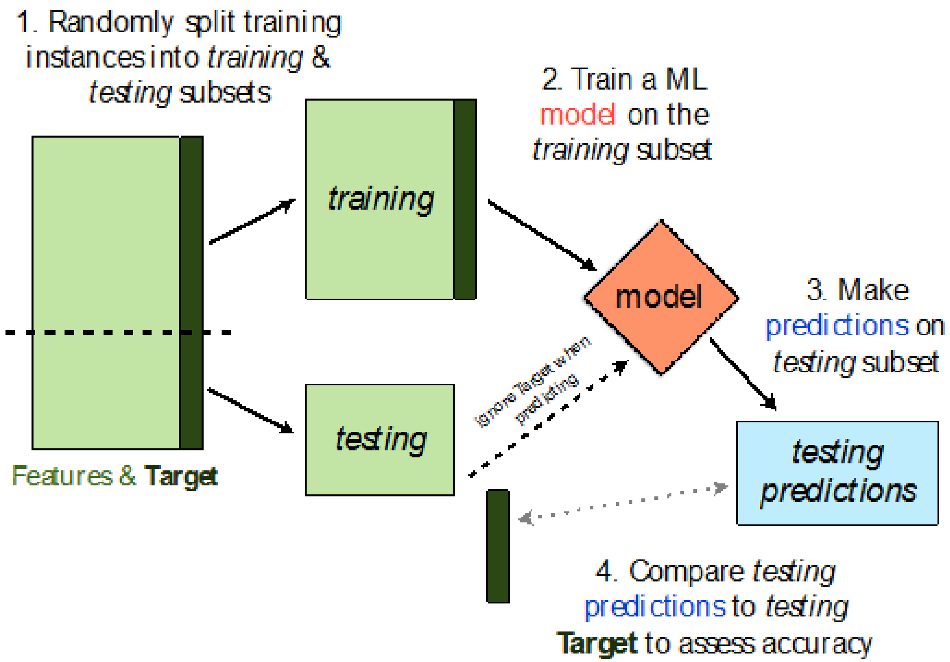


Figure 12.1 **An example FP-tree. The FP-tree looks like a generic tree with links connecting similar items.**

TID	Items in transaction
001	r, z, h, j, p
002	z, y, x, w, v, u, t, s
003	z
004	r, x, n, o, s
005	y, r, x, z, q, t, p
006	y, z, x, e, q, s, t, m

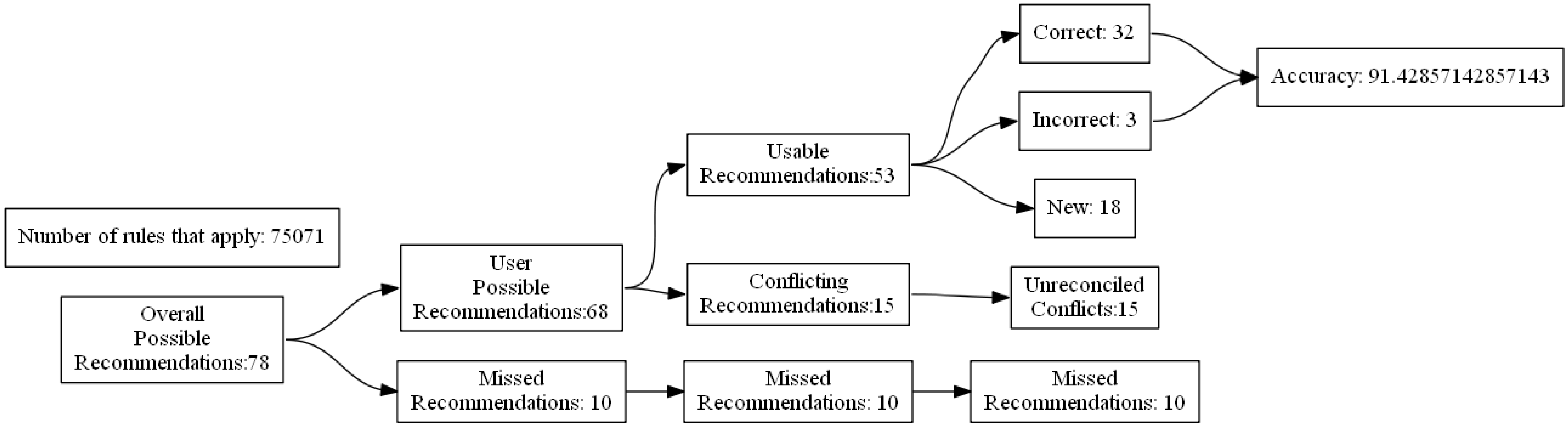
Check out Machine Learning in Action by Peter Harrington, Chapters 11+12 for step-by-step fp-growth code in Python.



[SEMI-]NOVEL APPROACHES

1. **Logically ternary data** instead of binary
   - Adds information, but creates conflicts
   - New method of conflict resolution needed
2. **Monte Carlo cross-validation**
   - Association Rule Learning is inherently self validating, but need model comparability
   - Evaluation method (accuracy) determined by applicable subsets of rules, per tested transactions

VALIDATION



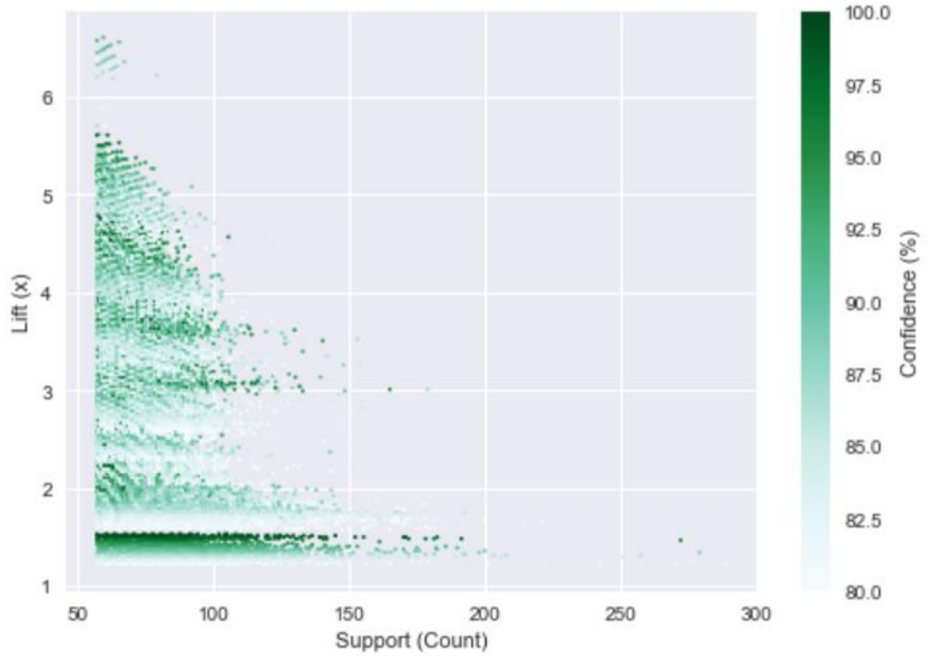
Number of rules that apply: 75071
Overall Possible Recommendations:78
User Possible Recommendations:68
Missed Recommendations: 10
Usable Recommendations:53
Conflicting Recommendations:15
Missed Recommendations: 10
Correct: 32
Incorrect: 3
New: 18
Unreconciled Conflicts:15
Missed Recommendations: 10
Accuracy: 91.42857142857143

RESULTS

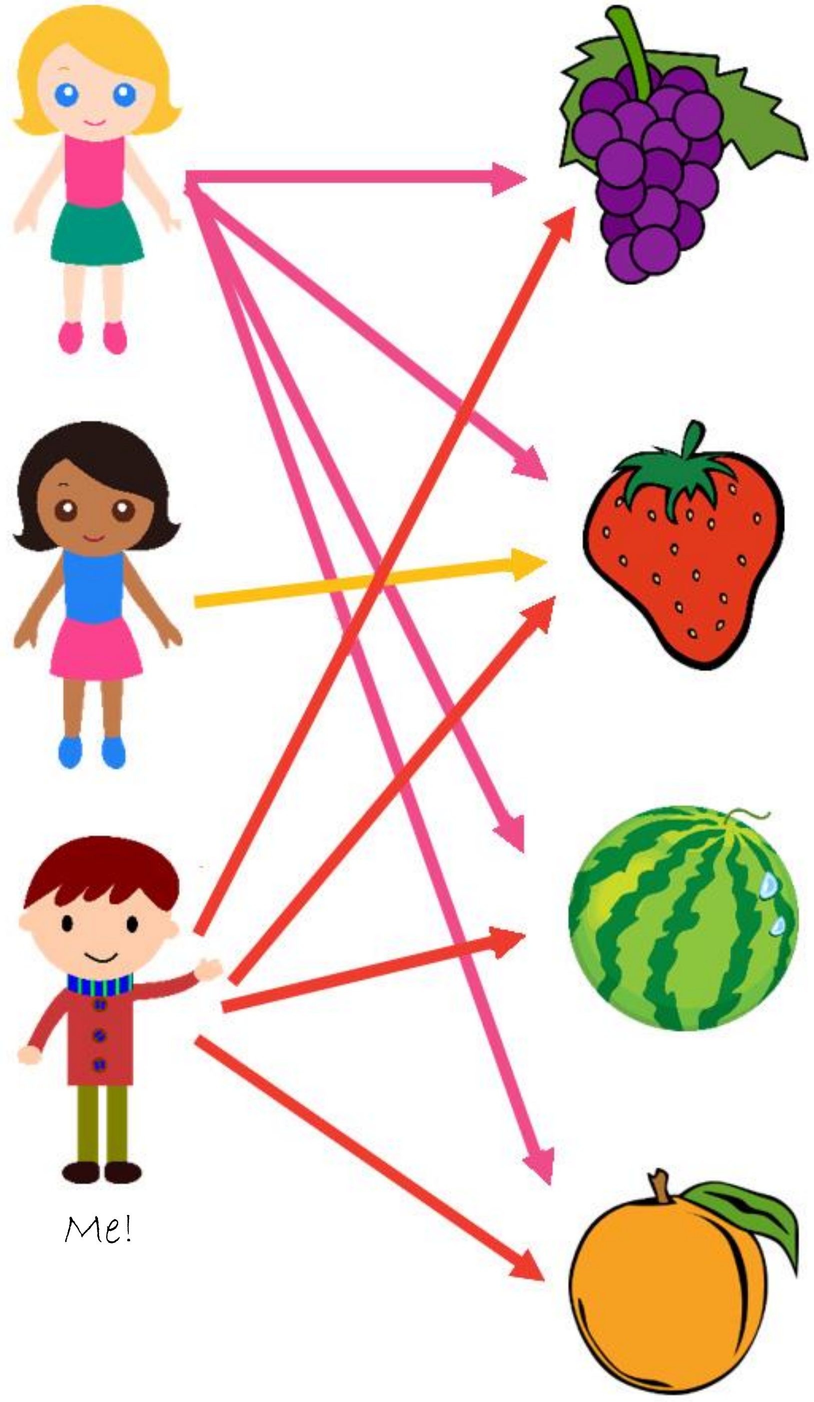
- Recommendations at least 80%+ accurate, usually 90%+
- Average 18-19 new recommendations pp.
- Commonly recommended foods: leeks, lettuce, garlic, honeydew melon, cod, cantaloupe, chicken eggs, basil, cucumber, white potatoes.
- Commonly conflicting foods: fruit, dairy, cruciferous vegetables

THE FULL MODEL

- ✓ 888,926 rules generated
- ✓ Rules for 74% of possible recommendations, with >80% confidence
  - ✓ **Can eat rules:** animals, 'staple' veges (carrots, cucumber, lettuce, tomato, potato), white rice
  - ✓ **Can't eat rules:** apple juice, coffee, cola, raisins
  - ✓ **Cut rules:** not alcohol of various types



Total Foods: 164 → Rules for: 121 → Conflicts for: 43 → Up to 78 usable



IBDALIZER

- Recommendation tool using input survey data
- Background output:

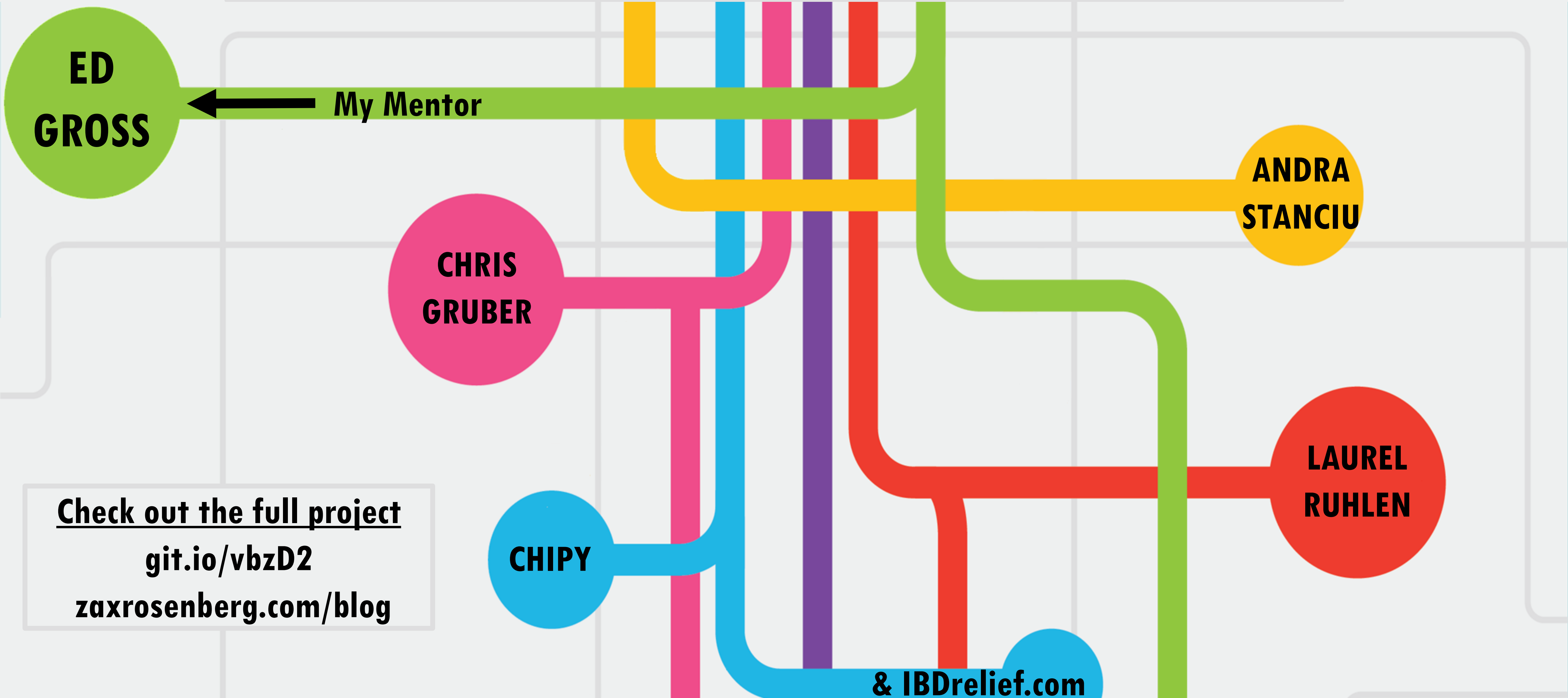
	Consequent	Antecedents	Support (Count)			Confidence (%)			Lift (x)		
		<lambda>	min	mean	max	min	mean	max	min	mean	max
0	Almond	[(Cashews, Bananas, Chicken) ...	46	49	62	80.00%	82.47%	91.30%	4.74x	4.88x	5.41x
1	Apple	[(Strawberries, Rice, White ...	46	49	98	80.00%	83.88%	97.83%	2.68x	2.81x	3.28x
2	Apple Juice	[(Cherries, Lemon, Raspbe ...	46	46	46	80.43%	80.43%	80.43%	3.60x	3.60x	3.60x
3	Bananas	[(Butter, Sweet Potato, C ...	46	50	87	80.00%	82.37%	93.88%	1.88x	1.94x	2.21x
...	...	...	...	...	...	...	...	...	...	...	...
15	not Cola	[(not Lemonade, Chicken), (not ...	57	64	120	80.00%	83.70%	90.28%	2.62x	2.74x	2.96x

```
df = df.groupby(['Consequent']).agg({'Antecedents': lambda ante: list(ante),
                                    'Support (Count)': ['min', 'mean', 'max'],
                                    'Confidence (%)': ['min', 'mean', 'max'],
                                    'Lift (x)': ['min', 'mean', 'max']
                                   }).reset_index()
```

FUTURE WORK

- **Update survey for recommendations**
- **Integrate live recommendation system into the survey** (with feedback and "learning")
- **Apply more advanced association techniques**, including hierarchical and clustering
- **Use my USDA nutrient database tool to identify relevant nutrients**

THANK YOU!



<u>Check out the full project</u>

git.io/vbzD2

zaxrosenberg.com/blog