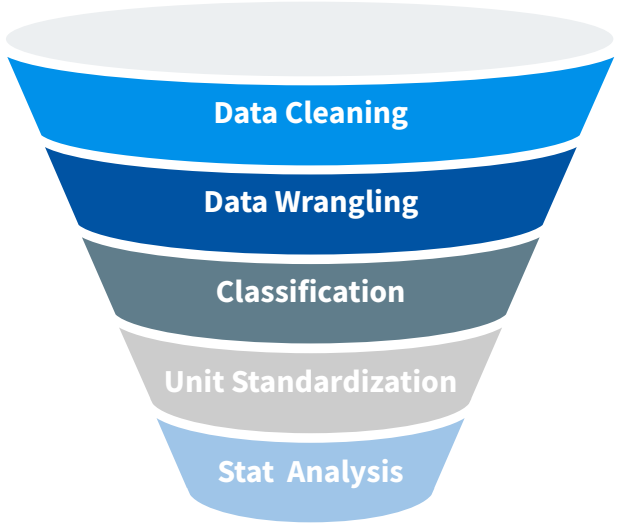
Automation of Purchasing Power Parity

Andy Chiv, Giovani Thai, Olivia Hartnett, Sarah Ellwein

Contents

- Introduction
- Research Questions
- Data Cleaning and Wrangling
- Methodology
- Results
- Problems and Improvement
- Conclusion



Introduction

Purchasing Power Parity (PPP):

Calculating the amount of goods and services that a single unit of currency in one country can purchase in another.

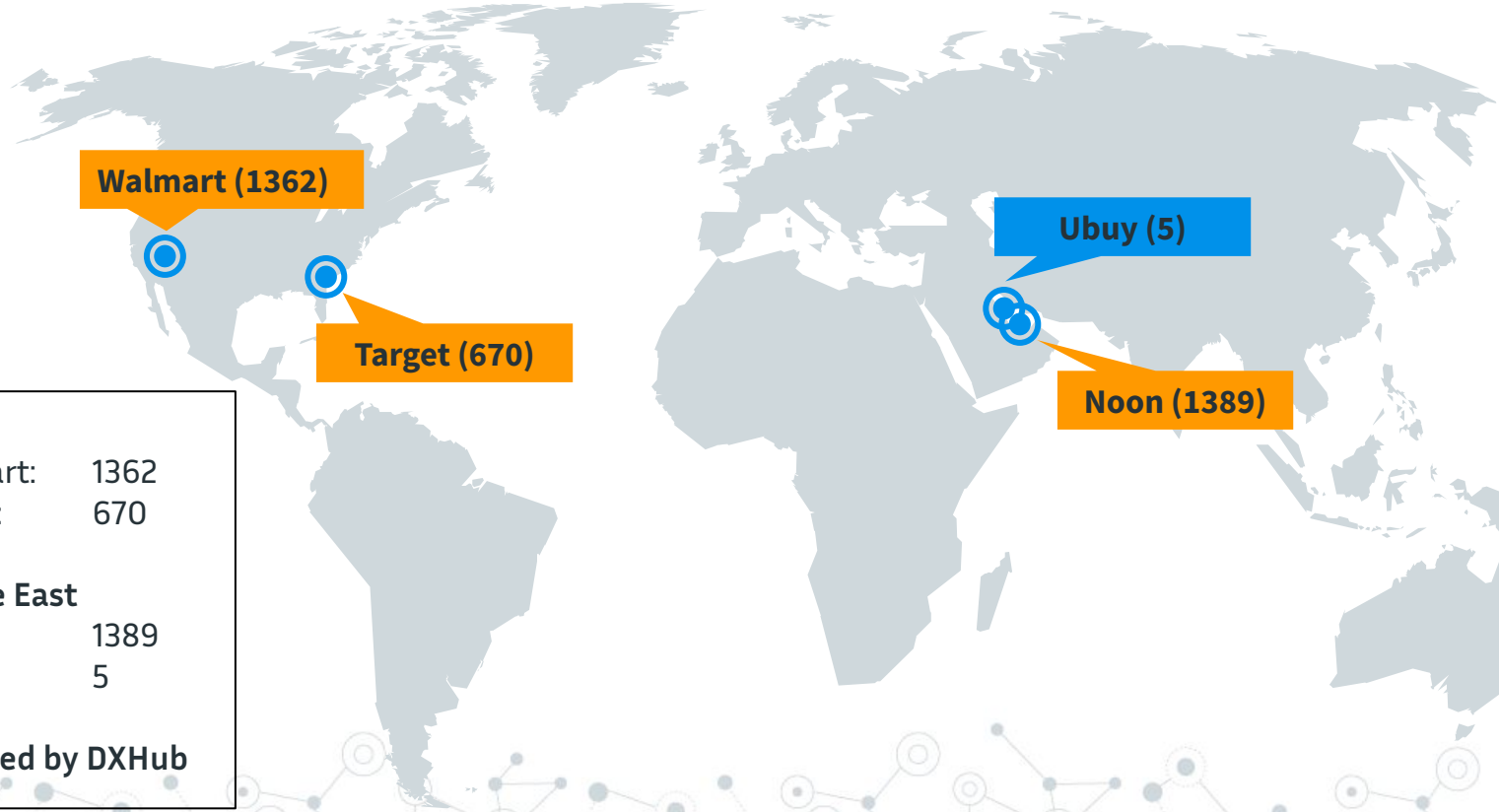
Why do we care?

- Reduce the amount of time and money surveying from store to store
- Useful for World Bank to sell products in different regions
- Allow governments to regulate product prices

Research Questions

1. How can products be categorized using classification modeling?
2. How can Purchasing Power Parity be automated to compare prices between stores, countries or regions?

Data Cleaning and Wrangling



USA
Walmart: 1362
Target: 670

Middle East
Noon : 1389
Ubuy: 5

Provided by DXHub

Important Variables

Store

Where products are sold

(Walmart, Target, Noon)

Category

Product Categories (33)

e.g. Vegetable, Butter, Bread, Beef

Chicken, Milk, Seafood

Description (Product Name)

Each product contains descriptive information, used to classify its category.

Price

Retail price (in dollars $)

Unit

Unit of the product

e.g. gram, ounce, liters ...

Unit Price

Standardized Unit Price

e.g. ($/kg) ...

Data Cleaning (Before)

	store	category	description	price	unit	Unit Price
Walmart	x	x	x	x	-	-
Target	x	x	x	x	-	-
Noon	x	x	x	x	-	-

Data Cleaning (After)

	store	category	description	price	unit	Unit Price
Walmart	x	x	x	x	X	X
Target	x	x	x	x	X	X
Noon	x	x	x	X	X	X

- Walmart: ***Unit and Unit Price*** by Regular Expression
- Target: ***Unit and Unit Price*** by Looping a list *of JSON Data*
- Noon: ***Unit and Unit Price*** by Regular Expression

Data Extraction

- Walmart: ***Unit and Unit Price*** by Regular Expression

```
# RegEx to extract unit prices
def extractPrice(r):
    if pd.isnull(r['displayedUnitPrice']):
        return np.nan
    price = float(re.search('[+-]?([0-9]*[.]?[0-9]+)', str(r['displayedUnitPrice'])).group(1))
    if '¢' in r['displayedUnitPrice']:
        return round(price/100, 2)
    return round(price, 3)
```

- Noon: ***Amount and Unit*** by Regular Expression
  - *Similar to Walmart*

- Target: ***Amount and Unit*** *through JSON Data*

  - Extracting the amount and unit in a list of JSON data

Methodology: Automation of PPP

Product Classification → **Unit Price Standardization** → **Statistical Analysis**

Product Classification is a process in grouping the product based on its description into its respective category:

Potential classification algorithm:

- Naive Bayes
- Decision Tree
- TF-IDF

Product price and amount are in different units.

Standardize Unit Price in metric form:

- oz -> kg
- lbs -> kg
- AED -> USD ($)

In each product category, Comparing unit price:

- Region vs Region
- Country vs Country
- Store vs Store

Deploying statistical approaches: Analysis of Variance (ANOVA), T-test

Product Classification

Goal: classify a product from a market website with consistent features

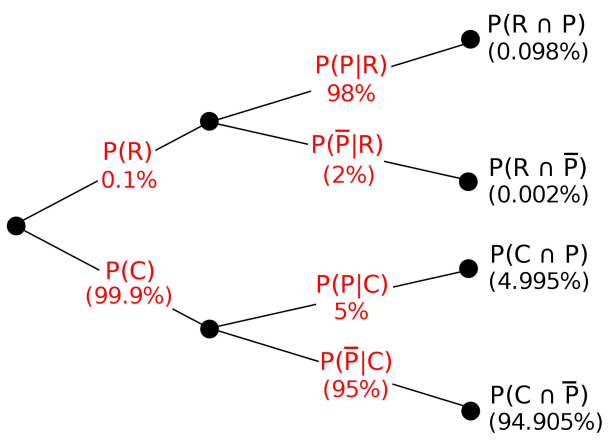
Product Classification: Choosing features

store	category	description	price	unit	Unit Price
	Label				

- Product name/description is the most ideal way to identify a product's category
  - This is how people identify categories
- Price, units, amount, etc. are not unique to a product

Product Classification: Choosing a model

- Classification problems require **supervised learning**
  - Cannot use unsupervised techniques like K-Means Clustering
- Our feature is text data
  - Not ideal using linear regression or topological data analysis
- **Naive Bayes Classifier**: Decisions on probability
  - Simple to understand
  - Easy implementation
  - Effective



Product Classification: Preprocessing features

1. Split product name into list of words
   (e.g. “Dynasty Jasmine Rice” -> “dynasty”, “jasmine”, “rice”)

2. Remove numbers and stop words
   (e.g. “the”, “is”, “are”)

3. Remove punctuation from words
   (e.g. “ben’s” -> “bens”)

4. Extract stem from words
   (e.g. “crunchy”, “crunchable” -> “crunch”)

5. Add to words if length is greater than 2

Product Classification: Model Strengths and Weaknesses

Strengths:

1. Products with descriptive name can be classified into their group (Naive Bayes with 73% accuracy rate)
2. Relatively fast compared to Decision Tree

Weaknesses:

1. Assumes variable independence (not always the case)
2. Dataset may be unreliable (issue with data collection/cleaning).

Product Classification: Limitations and Possible Improvements

Limitations:

- **Data cleaning**: majority of the project timeline, little time was given in implementing and evaluating other models
- **Data reliability**: multiple errors found in product categories
  (e.g. rice cookers are categorized as rice)

Future Improvements:

- **Experimenting different models**: decision trees, random forest
- **Improve features**: TF-IDF (emphasize weights of important features)
- **Further data cleaning**: create further subcategories of labels (vegetable can be split into different kinds), make corrections

Methodology: Unit Price Standardization

In order to accurately compare similar products and perform statistical analysis, we need the prices of our products to be standardized.

Unit prices

- $8/4 oz → $2/oz

Use universal system of measurement (metric units)

- oz, lb → kg

Methodology: Statistical Analysis

Product of Interest	Other Products	Product as a single Unit
Beef	*Seafood*	*Coffee -Maker*
Butter	*Frozen Fish*	*Microwave*
Potatoes	*Frozen Seafood*	*Rice-Cooker*

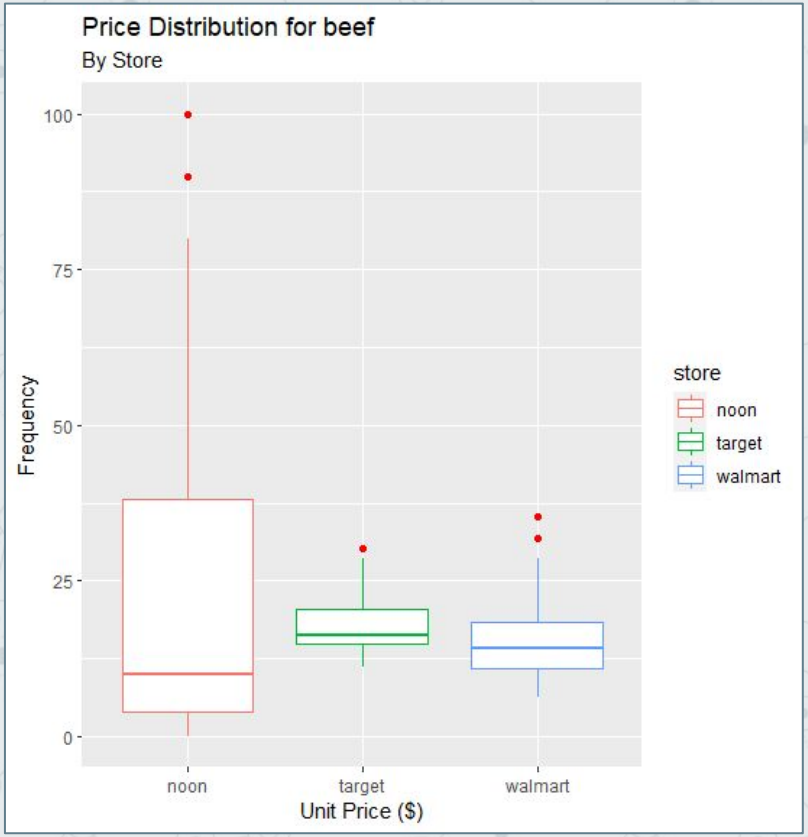
- ➔ Looking at categorical vs quantitative variables
- ➔ Some products remains incorrectly classified (Noon Data)
  - ◆ Any product as a single unit (ea) get dropped
  - ◆ Any unit price larger than 100$ is removed (outliers)

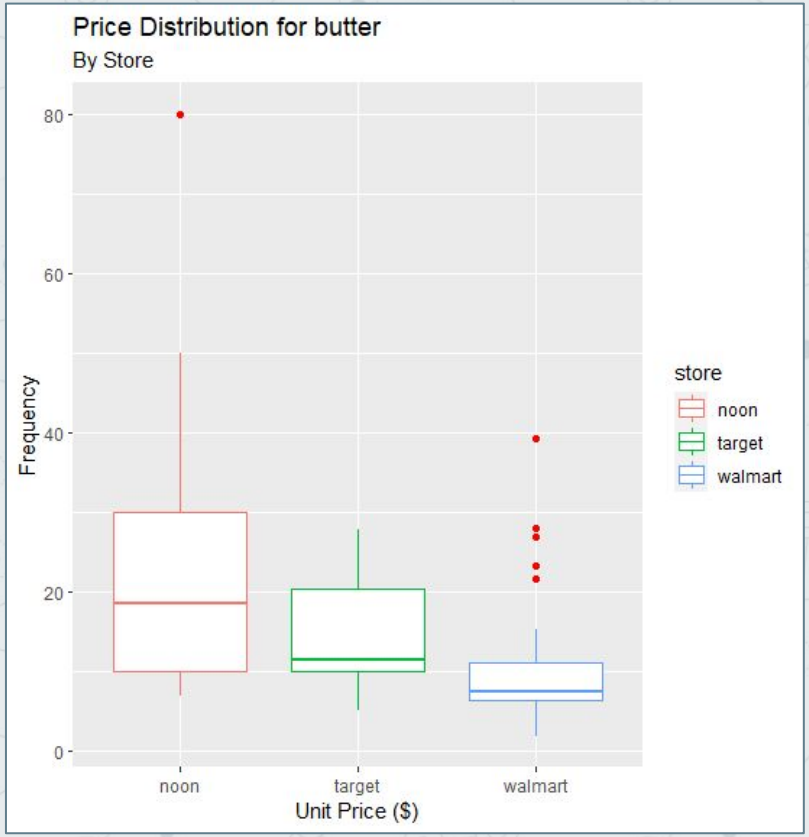
ANOVA Summary

Store	Beef ($)	Butter ($)	Potatoes ($)
Noon	22.94	21.39	28.08
Target	18.12	14.12	8.75
Walmart	15.61	11.61	4.46
p-value	0.00612*	0.00146*	1.8e-11*

- ➔ This analysis provides a structure to compare prices between stores, countries, or regions
- ➔ The statistical results:
  - ◆ Ensure that the ANOVA assumptions are met, otherwise, analysis is not reliable
  - ◆ Not yet reliable due to limitation of the data, p-value is not trustworthy

Distribution of Products





Conclusion

Strengths:

1. Products with descriptive information can be classified into their group using classification algorithm (Naive Bayes with 73% accuracy rate)
2. Create a pipeline that automates the process of PPP comparisons
    a. Classifying the products
    b. Converting unit and price into metric form
    c. Run statistical analysis: T-test, F-test (ANOVA)

Weaknesses:

1. The automation has not been validated due to to data limitation
2. Each product category could have sub-categories to increase the accuracy rate

Conclusion

- Automating PPP is crucial for global markets
- Non-uniformly formatted product data
- Suggestions can help future Business Analyst, Mathematica Data Scientists