

# **Customer Value Classification**

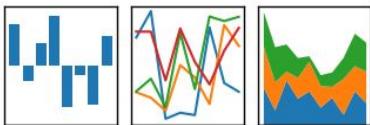
**Akuang Saechao, Victor  
Fateh, Nicholas Tang**

# Tools



pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



# Dataset

- UC Irvine
- Online Retail Transactions (xlsx)
- 541,909 from 01/12/2010 - 09/12/2011



InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	40513.3513888 8889	2.55	17850	United Kingdom

# Goal Process

Use assigned values to justify  
customer focus



Assign customer a value based on monetary worth



Classify Customers by some metric

# Determine Value

541885	581585	21684	SMALL MEDINA STAMPED METAL BOWL	12	40886.521528	0.85	15804.0	United Kingdom
541886	581585	22398	MAGNETS PACK OF 4 SWALLOWS	12	40886.521528	0.39	15804.0	United Kingdom
541887	581585	23328	SET 6 SCHOOL MILK BOTTLES IN CRATE	4	40886.521528	3.75	15804.0	United Kingdom
541888	581585	23145	ZINC T-LIGHT HOLDER STAR LARGE	12	40886.521528	0.95	15804.0	United Kingdom
541889	581585	22466	FAIRY TALE COTTAGE NIGHT LIGHT	12	40886.521528	1.95	15804.0	United Kingdom
541890	581586	22061	LARGE CAKE STAND HANGING STRAWBERY	8	40886.534028	2.95	13113.0	United Kingdom
541891	581586	23275	SET OF 3 HANGING OWLS OLLIE BEAK	24	40886.534028	1.25	13113.0	United Kingdom
541892	581586	21217	RED RETROSPOT ROUND CAKE TINS	24	40886.534028	8.95	13113.0	United Kingdom

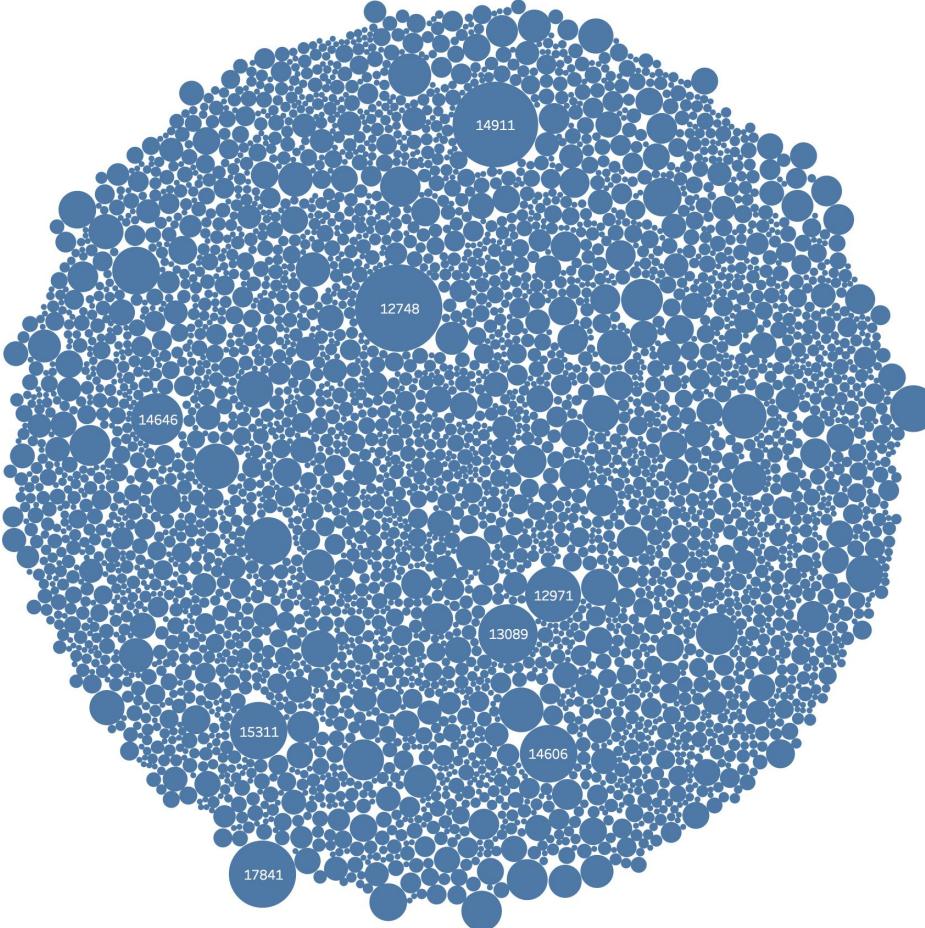
Set value to customer based on monetary worth

541893	581586	20685	DOORMAT RED RETROSPOT	10	40886.534028	7.08	13113.0	United Kingdom
541894	581587	22631	CIRCUS PARADE LUNCH BOX	12	40886.534722	1.95	12680.0	France
541895	581587	22556	PLASTERS IN TIN CIRCUS PARADE	12	40886.534722	1.65	12680.0	France

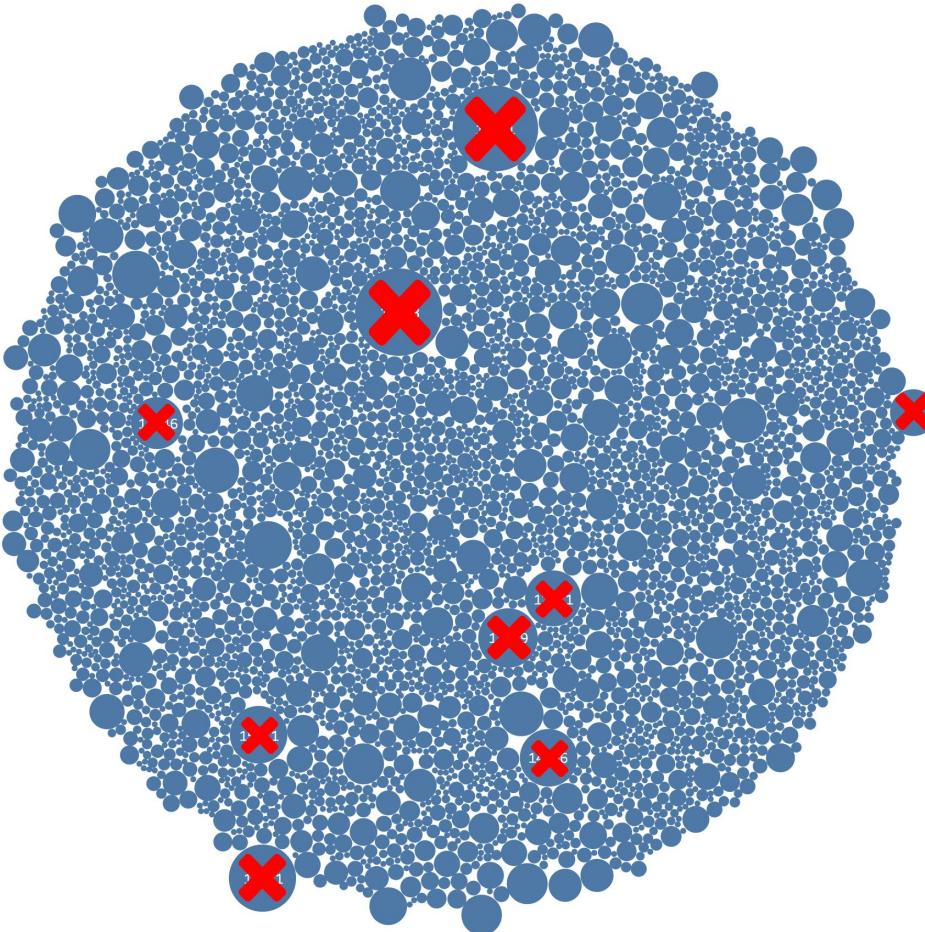
# Determine Value

541885	581585	21684	SMALL MEDINA STAMPED METAL BOWL	12	40886.521528	0.85	15804.0	United Kingdom
541886	581585	22398	MAGNETS PACK OF 4 SWALLOWS	12	40886.521528	0.39	15804.0	United Kingdom
541887	581585	23328	SET 6 SCHOOL MILK BOTTLES IN CRATE	4	40886.521528	3.75	15804.0	United Kingdom
541888	581585	23145	ZINC T-LIGHT HOLDER STAR LARGE	12	40886.521528	0.95	15804.0	United Kingdom
541889	581585	22466	FAIRY TALE COTTAGE NIGHT LIGHT	12	40886.521528	1.95	15804.0	United Kingdom
541890	581586	22061	LARGE CAKE STAND HANGING STRAWBERY	8	40886.534028	2.95	13113.0	United Kingdom
541891	581586	23275	SET OF 3 HANGING OWLS OLLIE BEAK	24	40886.534028	1.25	13113.0	United Kingdom
541892	581586	21217	RED RETROSPOT ROUND CAKE TINS	24	40886.534028	8.95	13113.0	United Kingdom
541893	581586	20685	DOORMAT RED RETROSPOT	10	40886.534028	7.08	13113.0	United Kingdom
541894	581587	22631	CIRCUS PARADE LUNCH BOX	12	40886.534722	1.95	12680.0	France
541895	581587	22556	PLASTERS IN TIN CIRCUS PARADE	12	40886.534722	1.65	12680.0	France

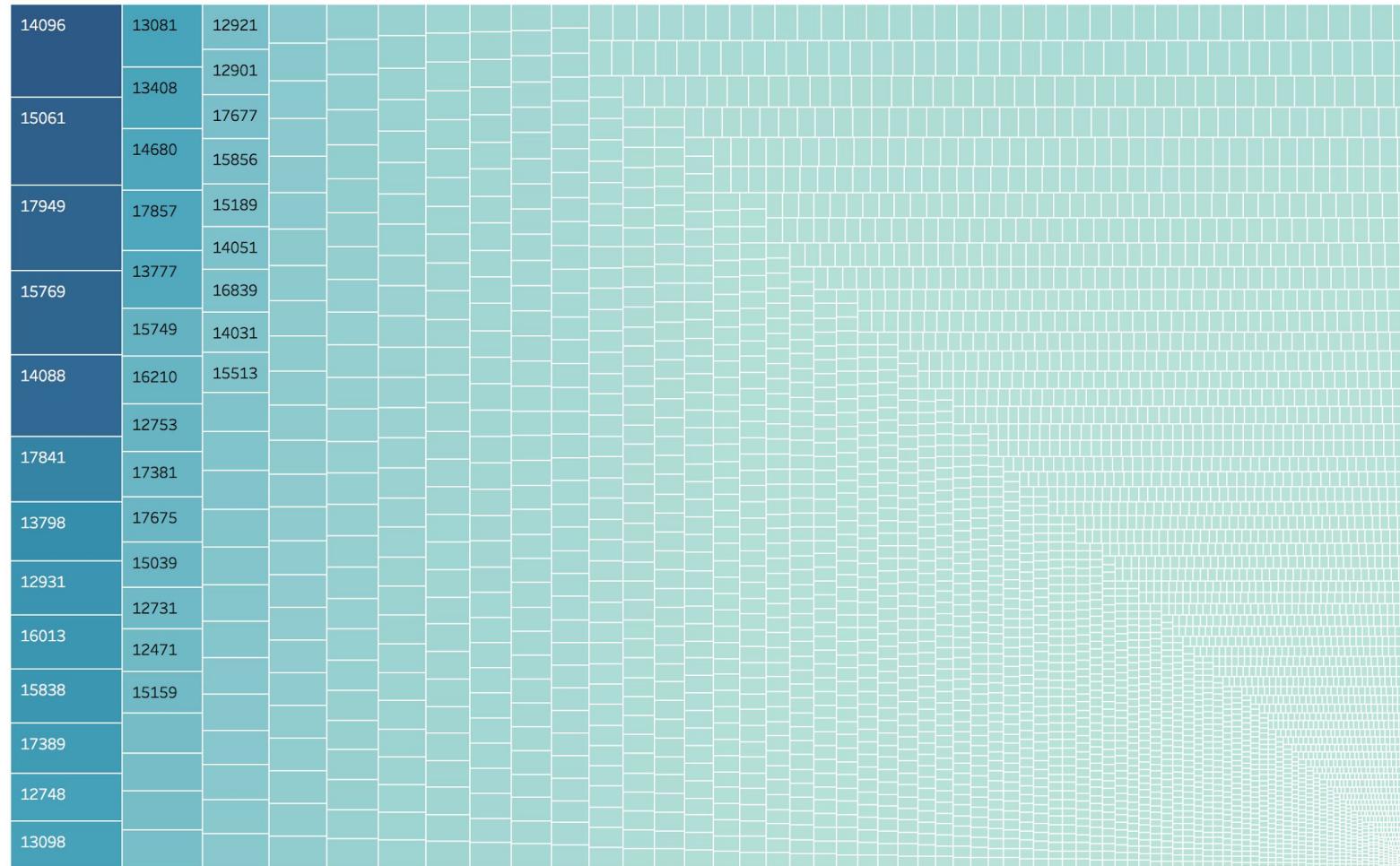
Assign a set of values to customers based on monetary worth



Customer ID. Size shows distinct count of Invoice No. The marks are labeled by Customer ID. The view is filtered on Customer ID, which excludes Null.



Customer ID. Size shows distinct count of Invoice No. The marks are labeled by Customer ID. The view is filtered on Customer ID, which excludes Null.

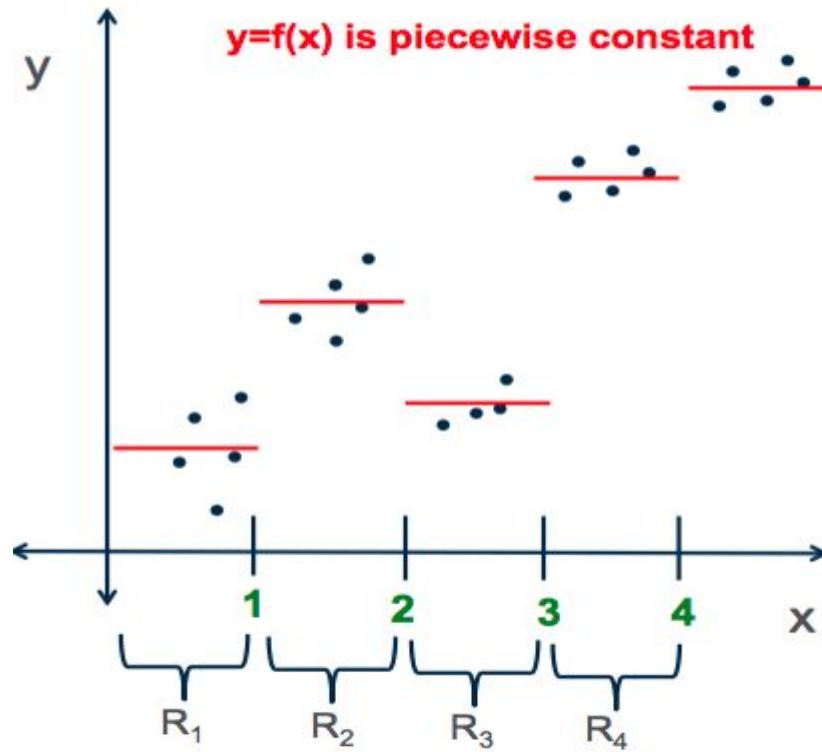


SUM([Unit Price]\*[Qua..  
0 57,121

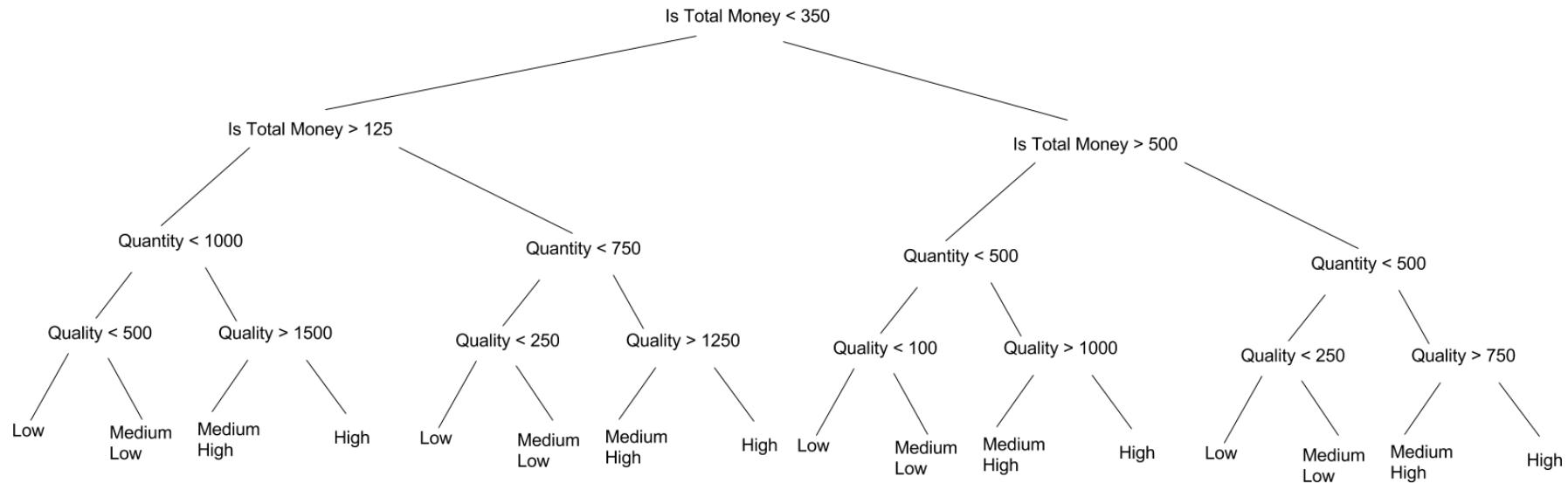
Customer ID. Color shows SUM([Unit Price]\*[Quantity]). Size shows SUM([Unit Price]\*[Quantity]). The marks are labeled by Customer ID. The data is filtered on average of Quantity and sum of Quantity. The average of Quantity filter ranges from 0 to 1,000. The sum of Quantity filter ranges from -303 to 29,069. The view is filtered on Customer ID and SUM([Unit Price]\*[Quantity]). The Customer ID filter excludes Null. The SUM([Unit Price]\*[Quantity]) filter ranges from 0 to 279,489.

# Classification And Regression Tree

- CART



# Decision Tree



# Algorithm

- Total Spent vs Total Quantity per Customer

CustomerID	UnitPrice	Quantity
17850	2.50	6
17850	2.75	8
13047	1.85	6
13047	4.25	24

$17850 = [ (2.50 + 2.75), (6 + 8) ] \rightarrow [(Total\ Spent), (Total\ Amount)]$

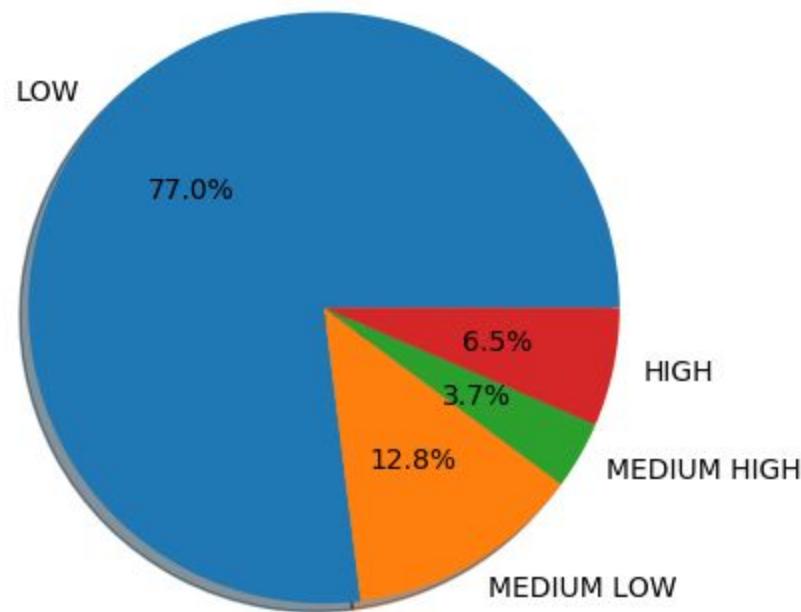
$13047 = [(1.85 + 4.25), (6 + 24) ] \rightarrow [(Total\ Spent), (Total\ Amount)]$

# Decision Tree Classifier

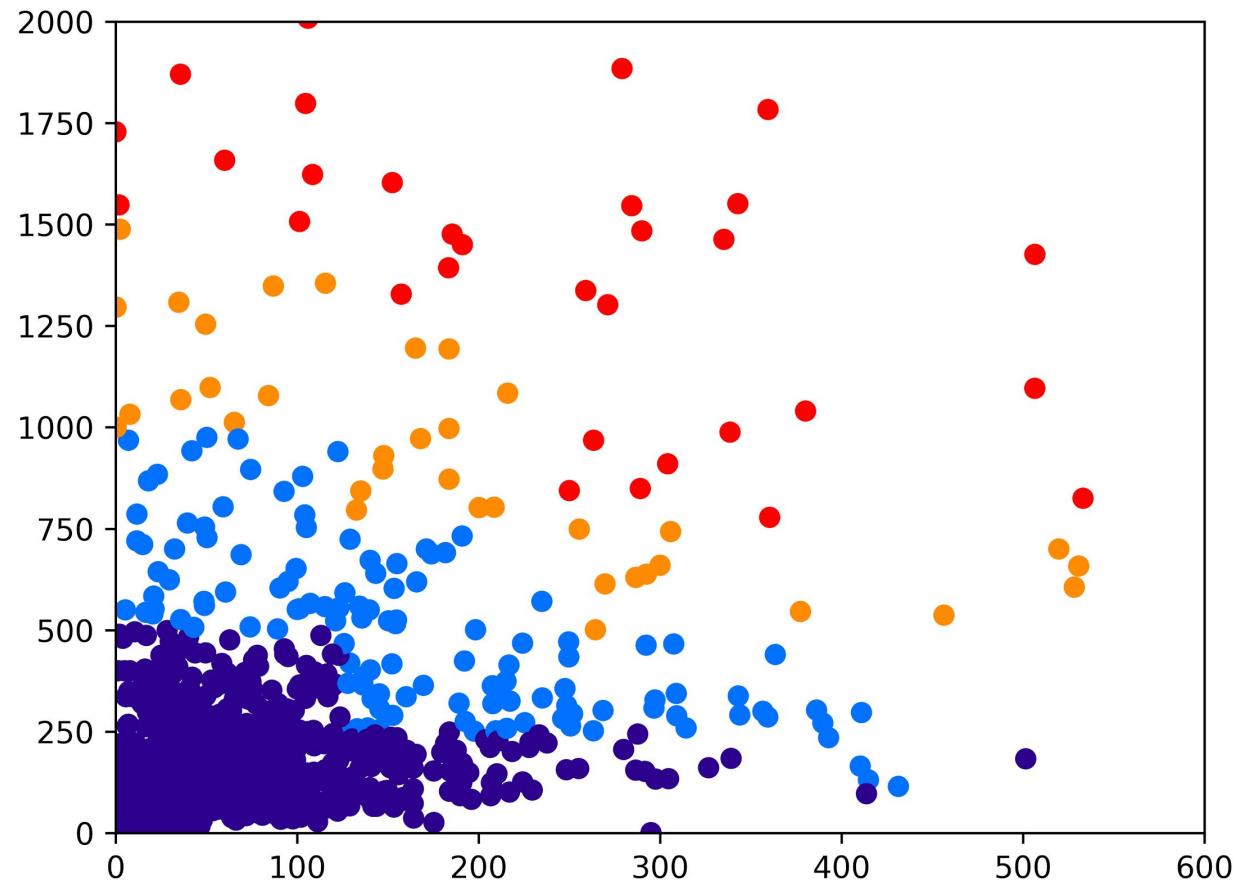
- Scikit-learn
- Classification And Regression Tree
- $X = [[5.25, 12], [12.50, 20], [7.35, 15]]$
- $Y = ["LOW", "MEDIUM LOW", "MEDIUM HIGH", "HIGH"]$

```
clf = tree.DecisionTreeClassifier()  
clf = clf.fit(X, Y)
```

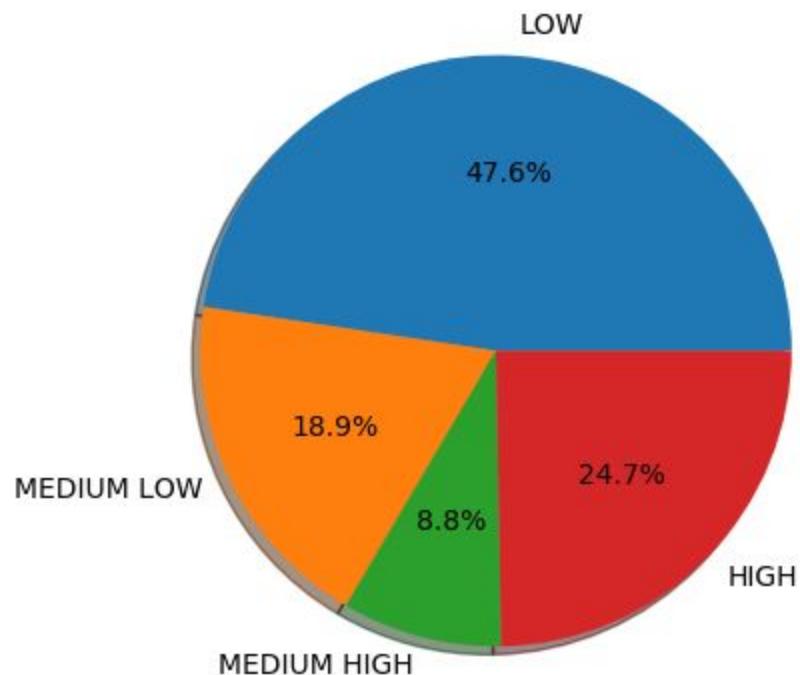
**10%**



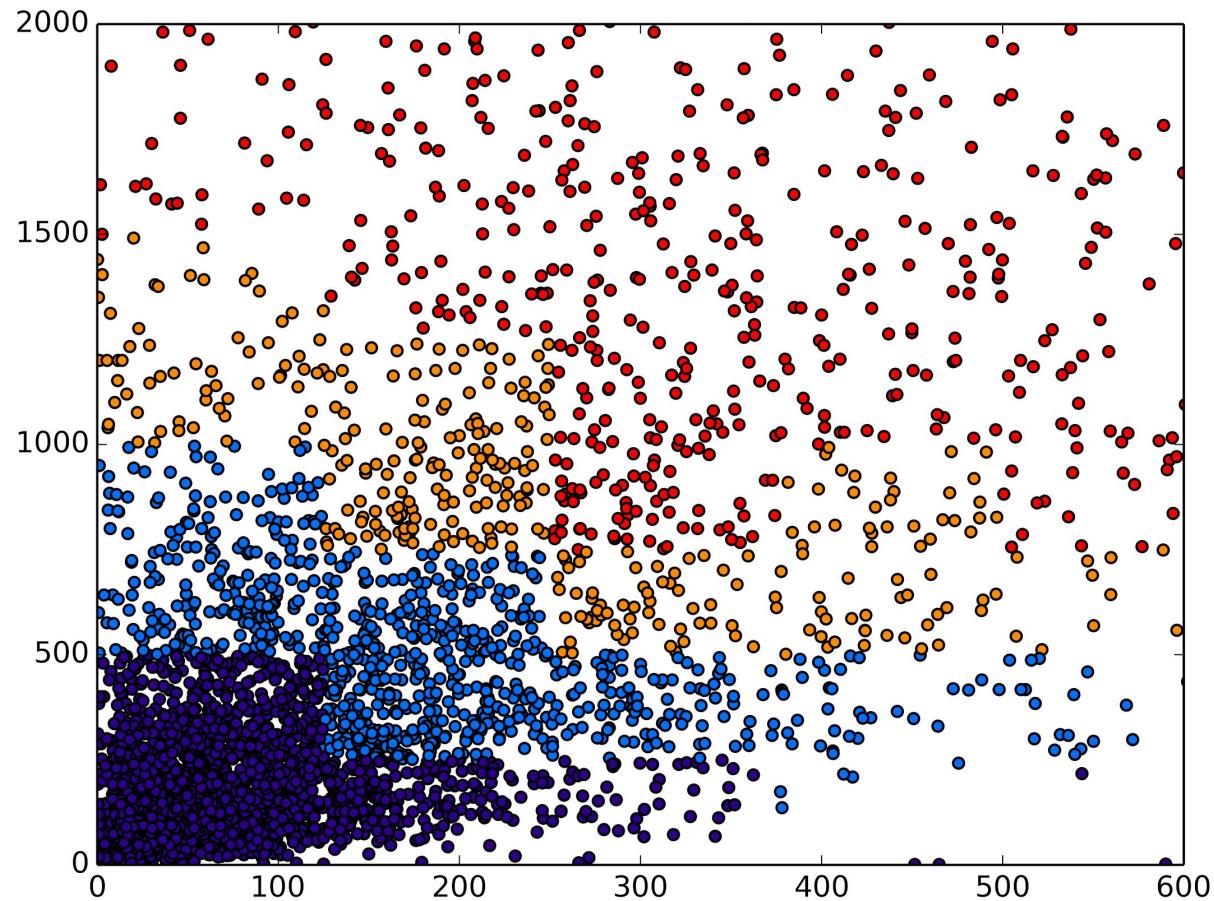
10%



**100%**



100%

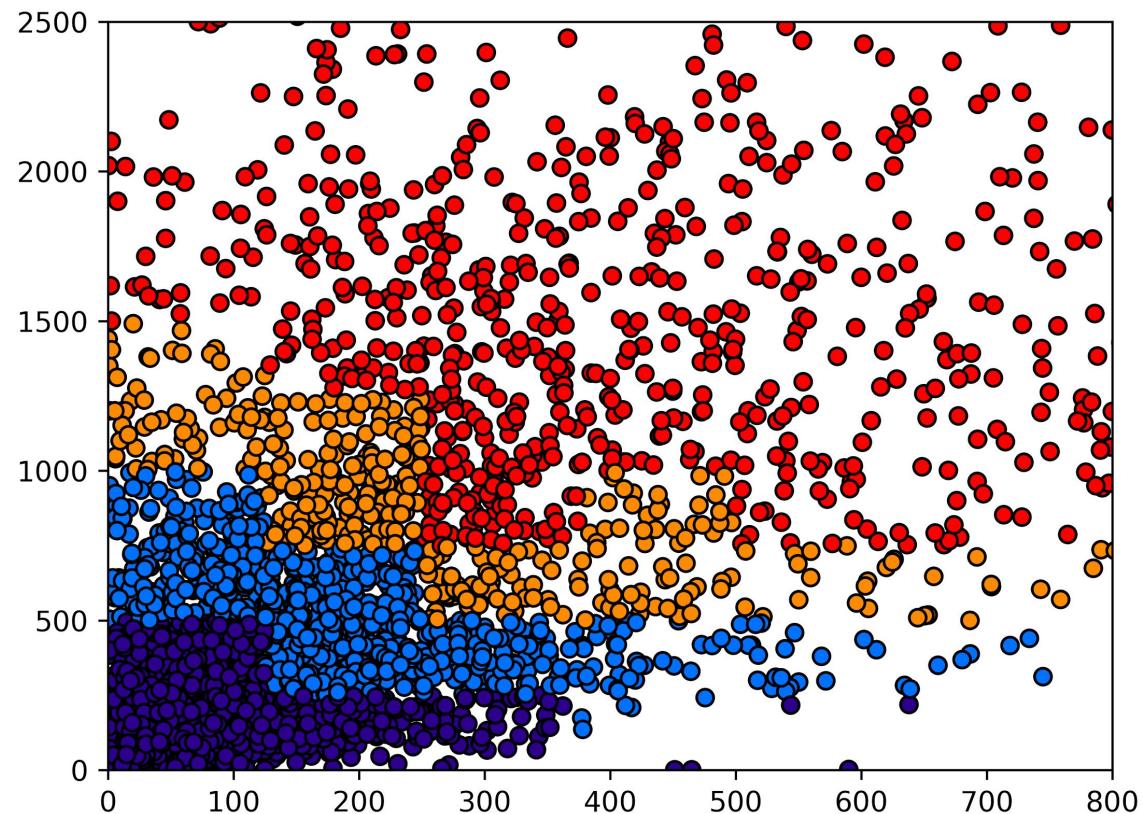


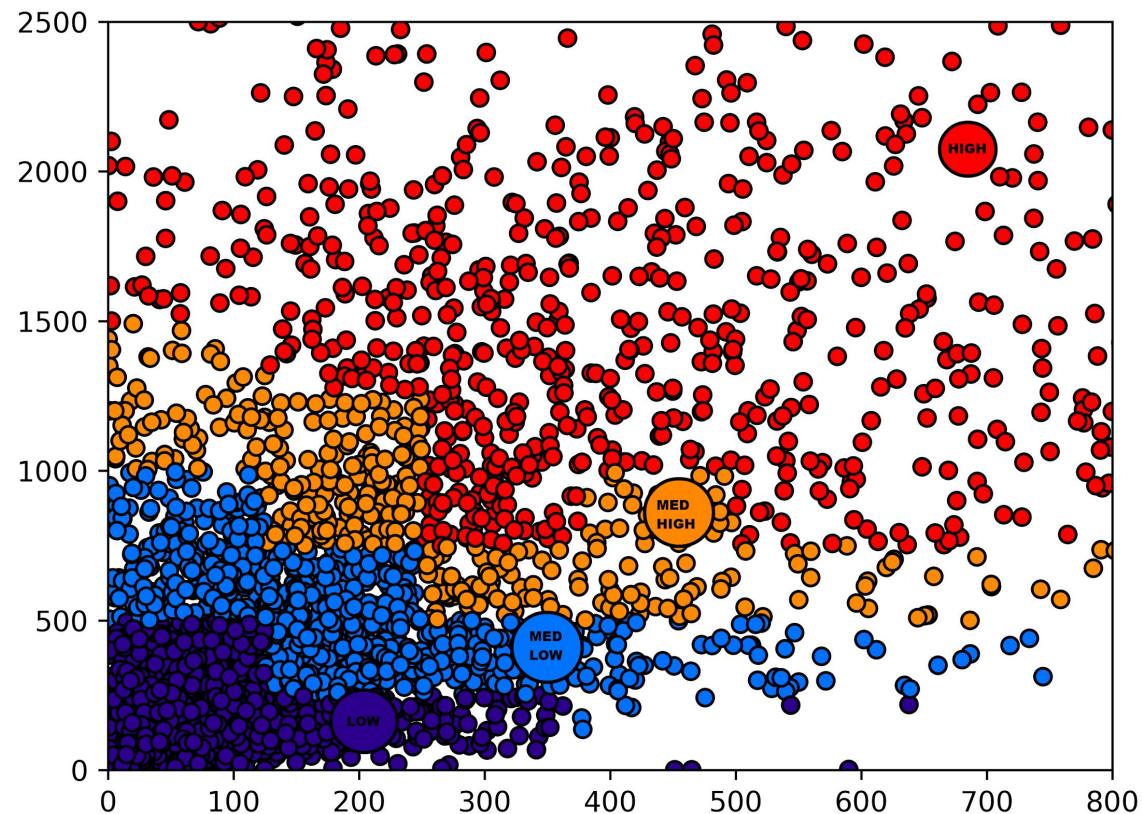
# Application

- Prediction for customers to determine their value to the business

```
clf.predict([[50, 1800], [50, 1300], [50, 800], [50, 300]])
```

```
[[50, 1800], [50, 1300], [50, 800], [50, 300]]  
['HIGH' 'MEDIUM HIGH' 'MEDIUM LOW' 'LOW']
```





# DEMO

Q & A