Pyxis

Generate investment alpha with clustering

Alice, Beau, Christina, Tiffany

17 April 2019

What is the problem?

TIME

How Pyxis saves time

Generate
Ideas Faster

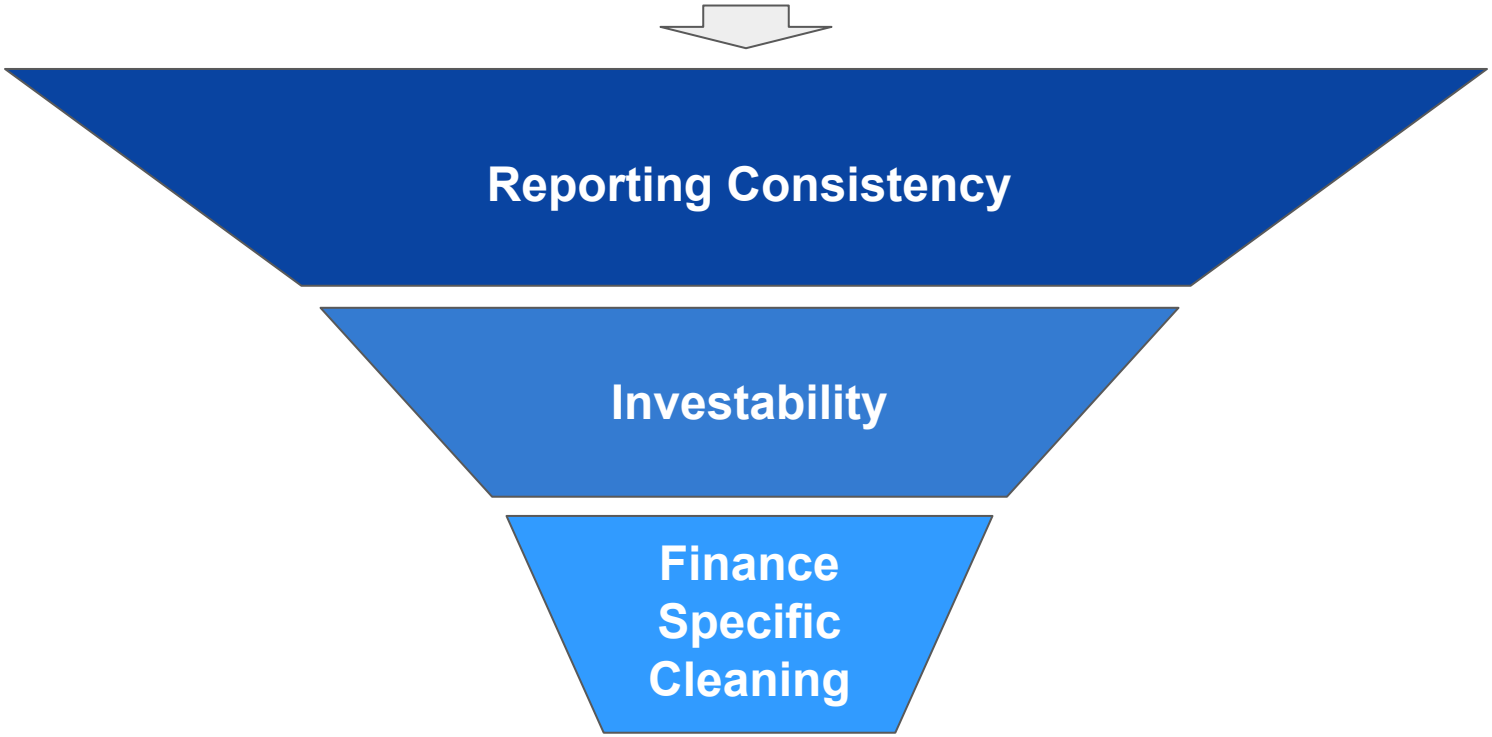
Detect
Fertile Sectors

Identify
Key Drivers

Data

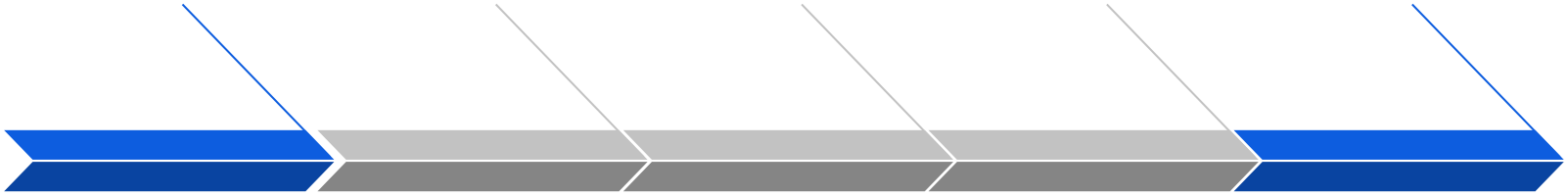
Collection, Filtering and Cleaning Funnel

Quarterly Financial Statements and Daily Share Prices
from Compustat & CRSP



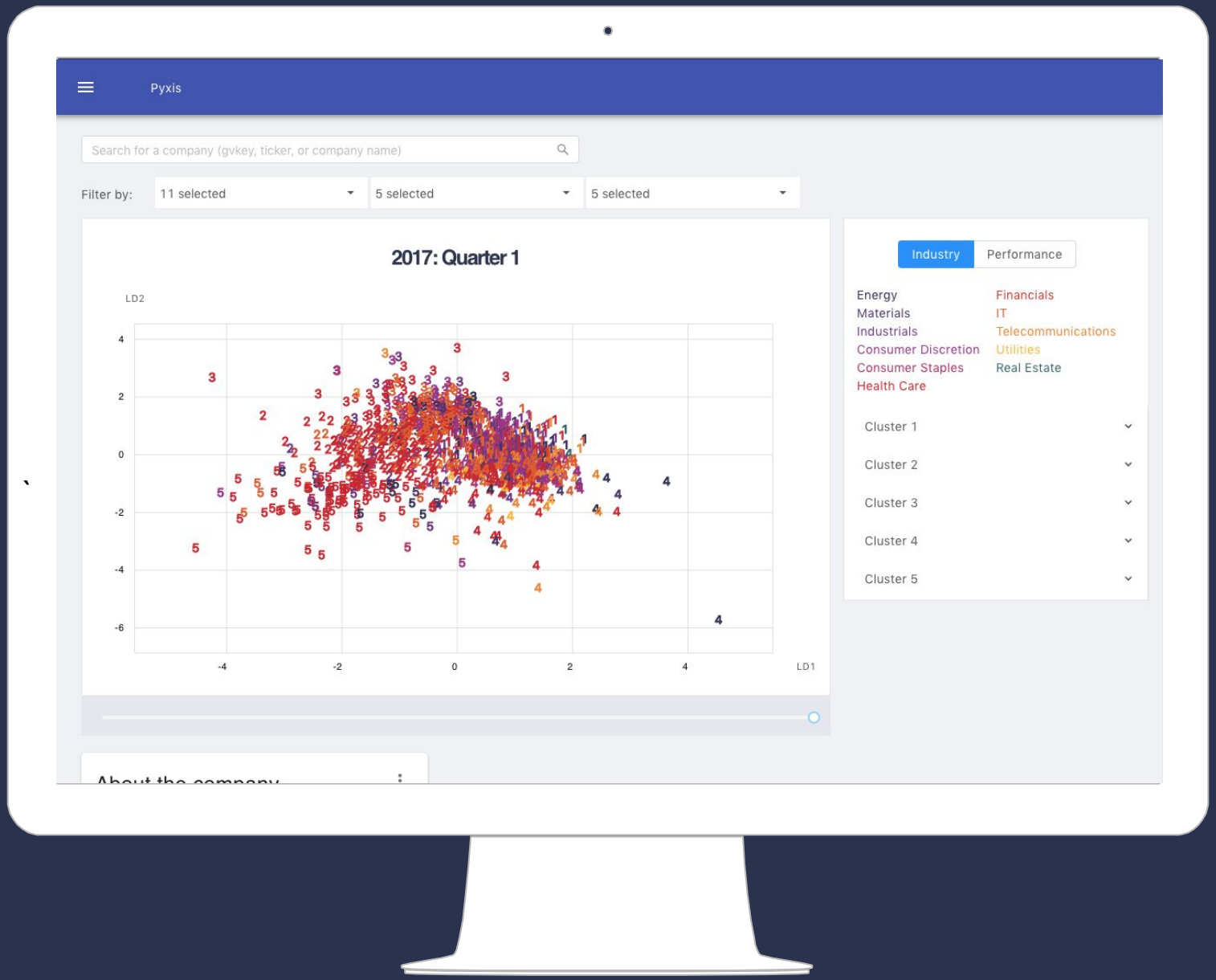
Clean data: ~ 200k rows, ~4000 companies

Proprietary Features



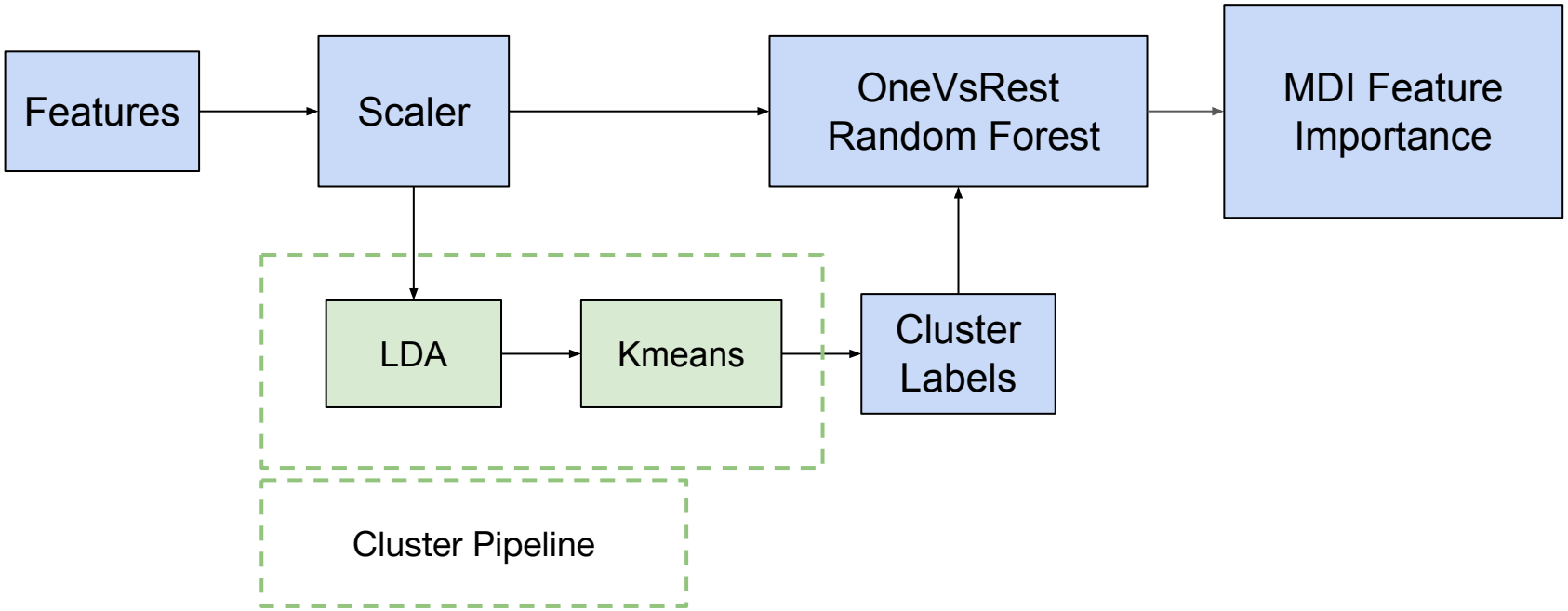
Clean Data	Smoothing	Feature Creation	Scaling	Clustering & Modeling
Quarterly information	12 Months Rolling Sum	Ratios - Profitability - Solvency - Asset Structure - Source of Capital - Growth rate	Square Root ->Log+1 MaxAbsScaler	

http://tiffapedia-pyxis.herokuapp.com/



How does Pyxis work under the hood?

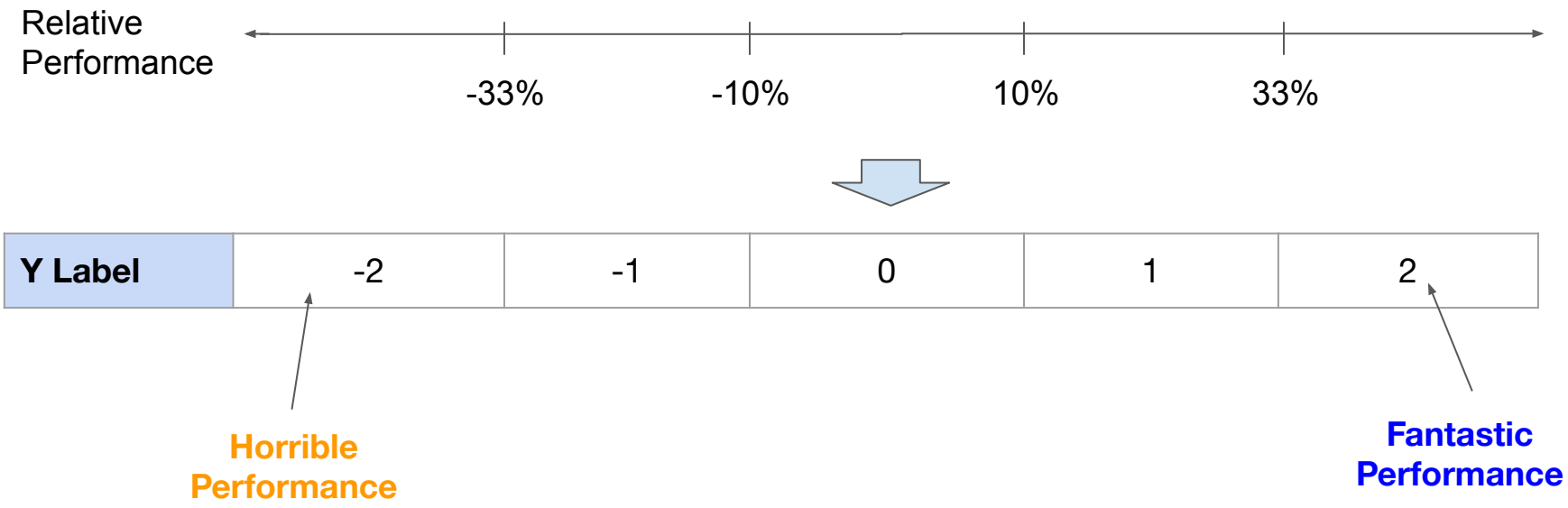
Cluster Interpretability



"Chapter 8: Feature Importance." *Advances in Financial Machine Learning*, by Prado Marcos López de, Wiley, 2018.

Predicting Forward Performance:
Y Labels

Labels - Relative Performance vs S&P 500



Predicting Forward Performance:
Train-Test Scheme

Train-Test Scheme

Q1 20X4	Q2 20X4	Q3 20X4	Q4 20X4	Q1 20X5	Q2 20X5	Q3 20X5	Q4 20X5	Q1 20X6
Train	Train	Train	Train	Test				
Train	Train	Train	Train	Train	Test			
Train	Train	Train	Train	Train	Train	Test		
Train	Train	Train	Train	Train	Train	Train	Test	
Train	Train	Train	Train	Train	Train	Train	Train	Test

The Use of Cluster Labels in Supervised Learning

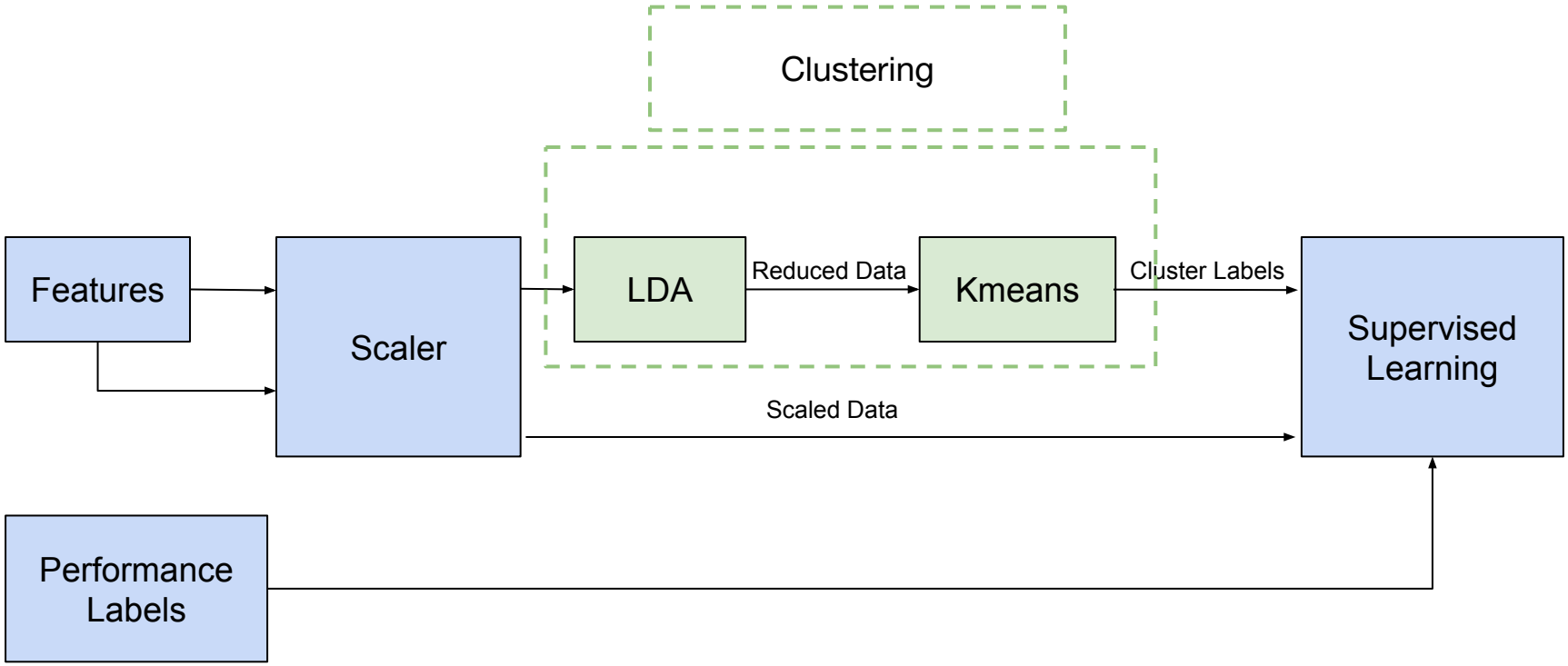
Base Model

Use **y label (t -1)**
To Predict **y label (t)**

Pyxis Model

Reduce-Cluster-Predict

The Use of Cluster Labels in Supervised Learning



Predicting Stock Performance is Challenging!

Evaluation

Evaluation Steps

- Compute the weighted F1 Score for each quarter
- Compute the median across all quarters

	Random Guess	Base	Pyxis
Weighted F1 Score	8.6%	24.0%	37.1%
Accuracy	23.0%	24.4%	39.0%

**Do sell-side analysts exhibit differential target price forecasting ability?*, by Bradshaw, Mark T. and Brown, Lawrence D. and Huang, Kelly, 2013.

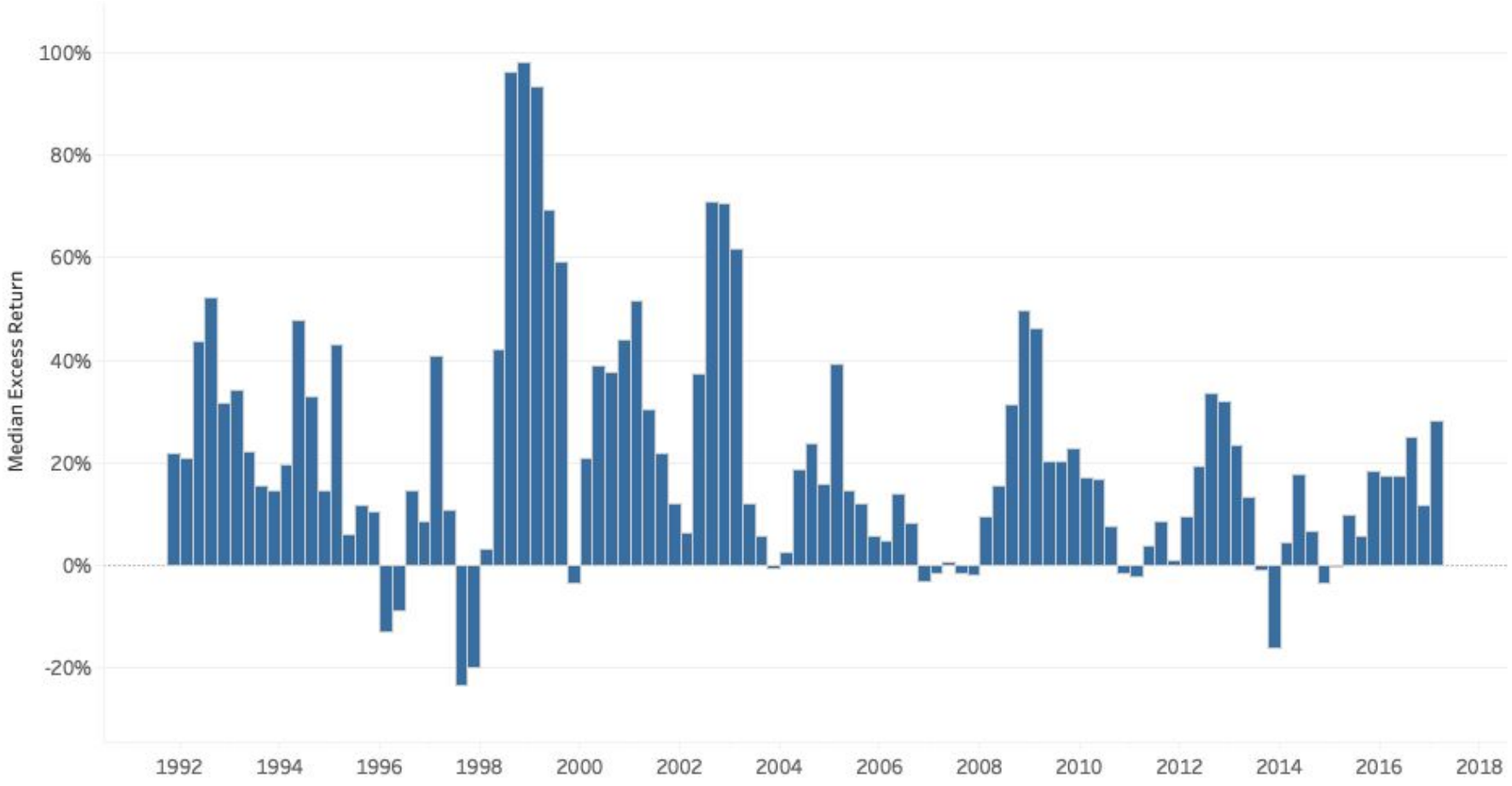
Pyxis is better at predicting Extreme Cases!

F1 score by Label across Time

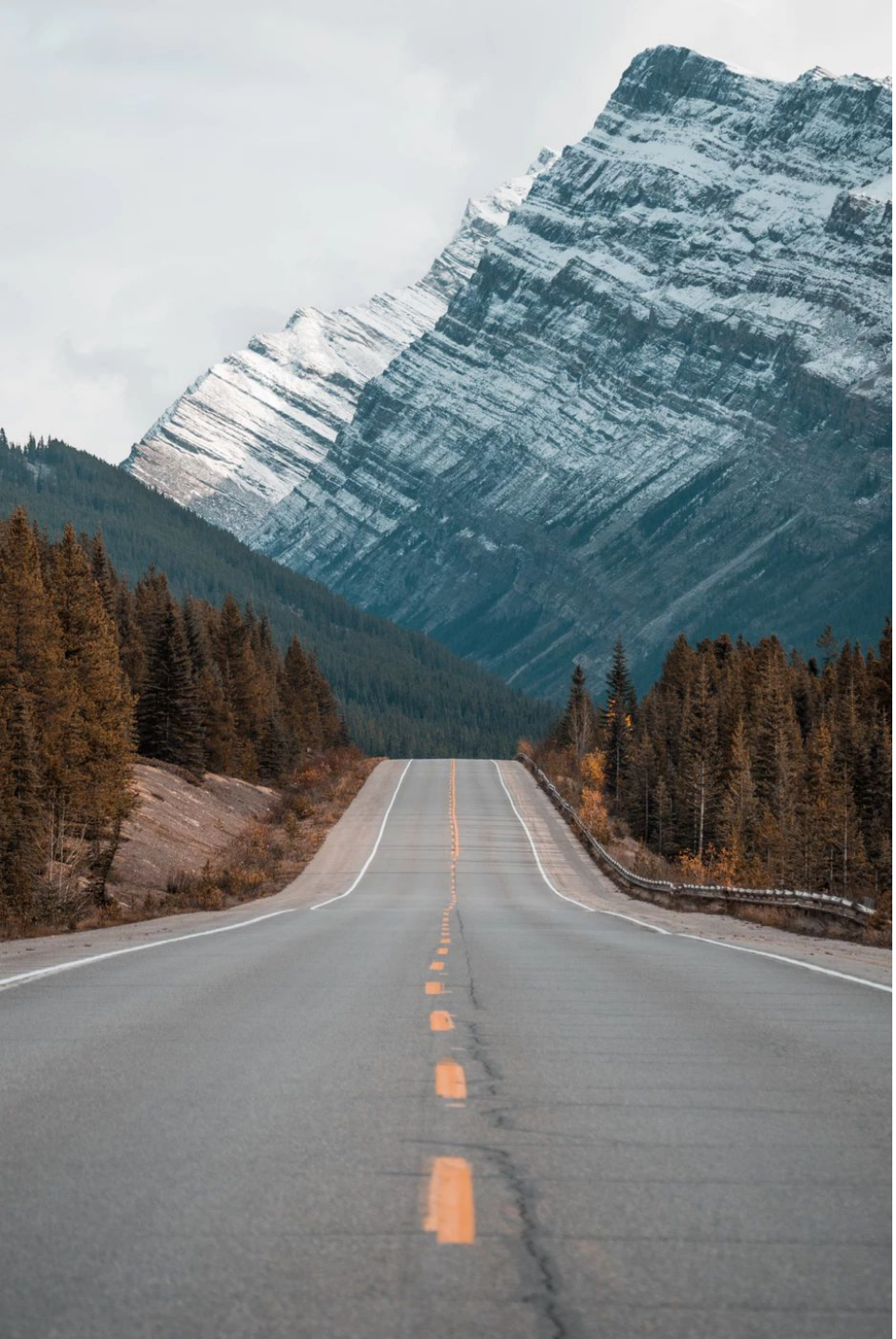
Label	Excess Return (%)	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	Avg.
-2	< -33																												49%
-1	-33 to -10																												31%
0	-10 to 10																												26%
1	10 to 33																												19%
2	> 33																												45%

Buying the predicted big winners!

Median Excess Return (alpha*) for Pred Label = 2



*Alpha = stock return - S&P 500 index return



Concluding Thoughts

User feedback

- Useful insights into the equity markets
- Strong results in stock performance prediction

Key challenges

- Understanding and processing the Compustat data
- Achieving good separation of clusters

Future improvements

- Macroeconomic regime indicators
- Time based models
- 3D visualization

See note below

Problem you are solving and for whom specifically?

- Why is this an important problem to solve, what would be the impact and potential level of impact? What's the mission?

Show your MVP solution. How and why it provides the initial solution to the problem you are solving?

- Show and explain how your MVP works/functionalities (website demo)
- What are the inputs/data (feature eng slide), data science/technical approach (SL slides), and outputs?
- How did you evaluate your solution? (evaluation slide)
- Highlight what's working well, key challenges / learning you had, and hypothetical next steps/future improvements if you were to continue with the project. (Concluding Thoughts Slide)

- If you had more time, what would be on your roadmap?

1. state the problem (Tiffany)
2. state what the outputs are (Tiffany)
3. show how we engineered the data (Alice)
4. demo (Alice)
5. what the models were, how we trained them, what architecture we used (Beau)
6. evaluation (Christina)
7. concluding thoughts (Christina)

In addition to the final presentation, each team should create a web-based deliverable that allows viewers to understand your project in some detail.Also please upload a project summary to the Berkeley website as described in https://docs.google.com/document/d/18CANgH-TcYohdR3tsU6oxDlR6K5-vLiIZyp1pfxx4zU/edit?usp=sharing

(Powerpoint is optional. In the past, teams provided Powerpoint presentations for supplementary purpose only.)

- It is your decision to include or exclude GitHub or other code repository as a link to the website (depending on if you want to make it public or not). If you do not include on the website, please send to us in case we need to reference back to something.

- It is your decision to embed the demo in the web deliverable or link to a separate web application (often driven by use case, as observed by examples below). If appropriate, interactive demos, that allow users to explore your data and results, or submit their own data for analysis, are encouraged.

- It is your decision to include a more in-depth write-up ("technical white paper) or not. For some groups, we have suggested a white paper because bulk of the project work is on algorithm development so it seems worthwhile documenting that major work.

- Any material (algorithms, code, results, visuals, etc.) you have used from outside sources should be appropriately attributed.
You can see some of the web-based deliverables from prior semesters:

About Pyxis

Pyxis is a capstone project for UC Berkeley's W210 course. We began our journey by wondering if we could classify stocks differently from the traditional industrial classification codes or valuation metrics. Our solution uses clustering to identify similar businesses based on operating metrics. Speaking with portfolio managers in the field we understood that interpretability of the clusters was paramount. We built this logic into the product so users can understand what factors drive stocks to be in the same cluster. We took our project a step further by trying to prove these clusters contained information more than just being nice to look at. To do this we utilized them in the prediction of relative forward returns--a notoriously difficult task. We hope you enjoy using Pyxis as much as we did building it. Please do not hesitate to reach out to us if you want to provide feedback or explore ways to further improve Pyxis.

Data

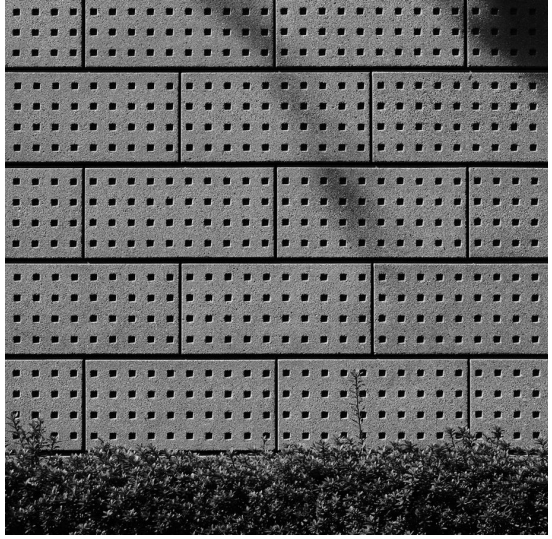
1	Universe	• Compustat Unrestated Quarterly Financial Statements • CRSP Monthly Pricing Data • 1990 - 2018
2	Tradability Filters	• $100M Market Cap (Inflation Adjusted) • $1M Dollar Volume (Inflation Adjusted) • No Financial Firms
3	Data Cleaning	• NAs to Zeros • Gap Filling • Delisting and Relisting

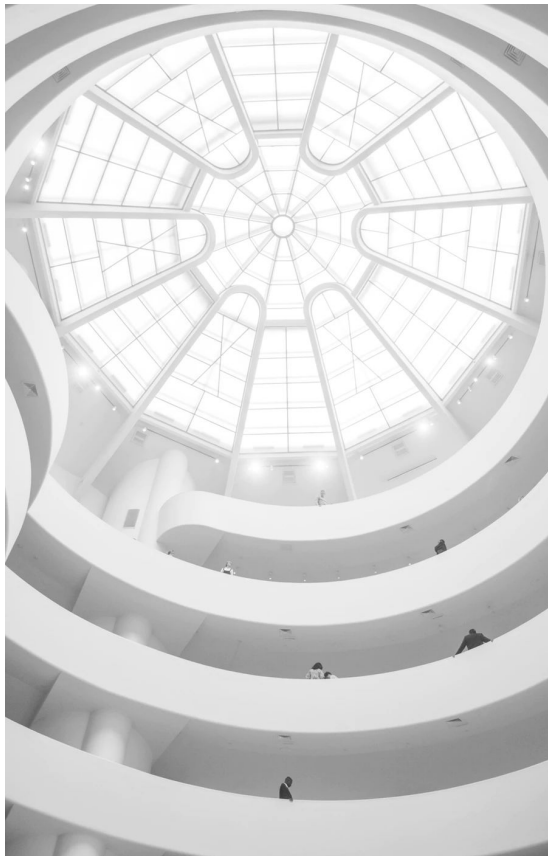
Proprietary Features

1	Quarterly to TTM	• Trailing Twelve Months (TTM) for Income Statement and Cash Flow Items

2	Feature Creation	• 9 Feature Categories:
		  ○ Profitability
		  ○ Asset Structure
		  ○ Solvency
		  ○ Utilization
		  ○ Liquidity
		  ○ Deployment
		  ○ Sourcing of Capital
		  ○ Growth
		  ○ Acceleration

3	Feature Scaling	• Sqrt and Log+1 Transformers • MaxAbsScaler





1 **Database**
PostgresDB

2 **Backend**
Django

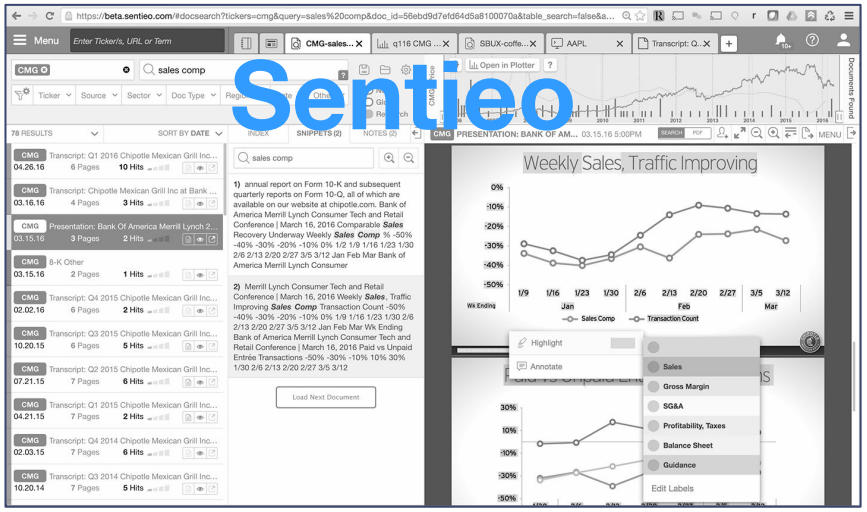
3 **Frontend**
React

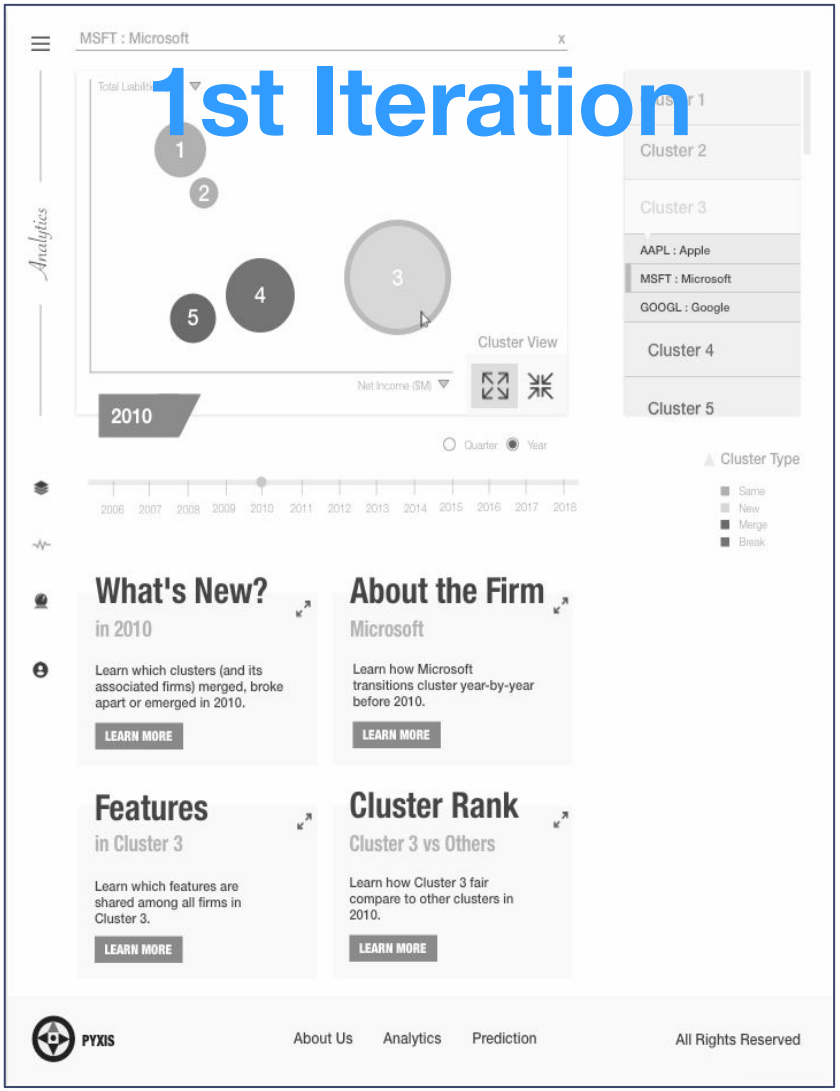
What Differentiates Pyxis

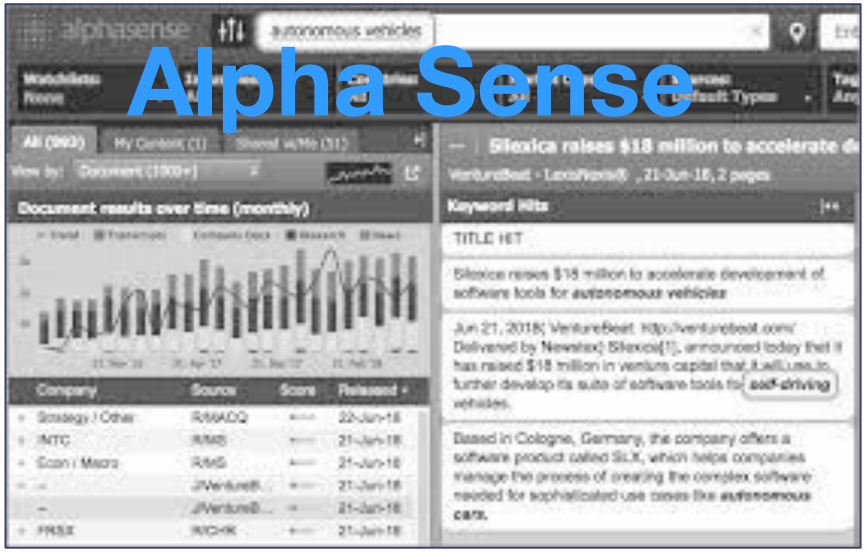
9 min

3 sec

What Differentiates Pyxis







Evaluation

Evaluation Steps

- Compute the weighted F1 Score for each quarter
- Compute the median across all quarters

	Base	Cluster-less	Cluster-ful
Weighted F1 Score	24.0%	34.2%	37.1%
Accuracy	24.4%	36.1%	39.0%

F1 score by Label across Time

Label	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	Grand..
-2	44%	47%	52%	51%	52%	61%	64%	60%	53%	53%	50%	34%	37%	43%	43%	44%	48%	36%	39%	49%	53%	45%	48%	54%	58%	55%	56%	**49%**
-1	34%	31%	33%	38%	33%	31%	29%	29%	28%	17%	18%	24%	23%	26%	32%	32%	36%	31%	28%	32%	38%	33%	35%	32%	32%	32%	39%	**31%**
0	36%	29%	30%	27%	22%	21%	17%	13%	16%	15%	15%	19%	24%	25%	27%	28%	30%	27%	25%	32%	34%	32%	33%	32%	36%	33%	32%	**26%**
1	25%	23%	26%	17%	18%	12%	11%	8%	10%	15%	19%	18%	18%	20%	18%	16%	23%	23%	17%	20%	24%	20%	20%	21%	25%	25%	20%	**19%**
2	52%	58%	52%	53%	49%	39%	37%	43%	55%	50%	53%	55%	52%	45%	47%	38%	30%	44%	54%	40%	36%	45%	39%	34%	36%	43%	48%	**45%**

Pyxis:

The Story Behind Our Product

“She knew there were **opportunities**

...if only she has **scalable, differentiated insights**.”

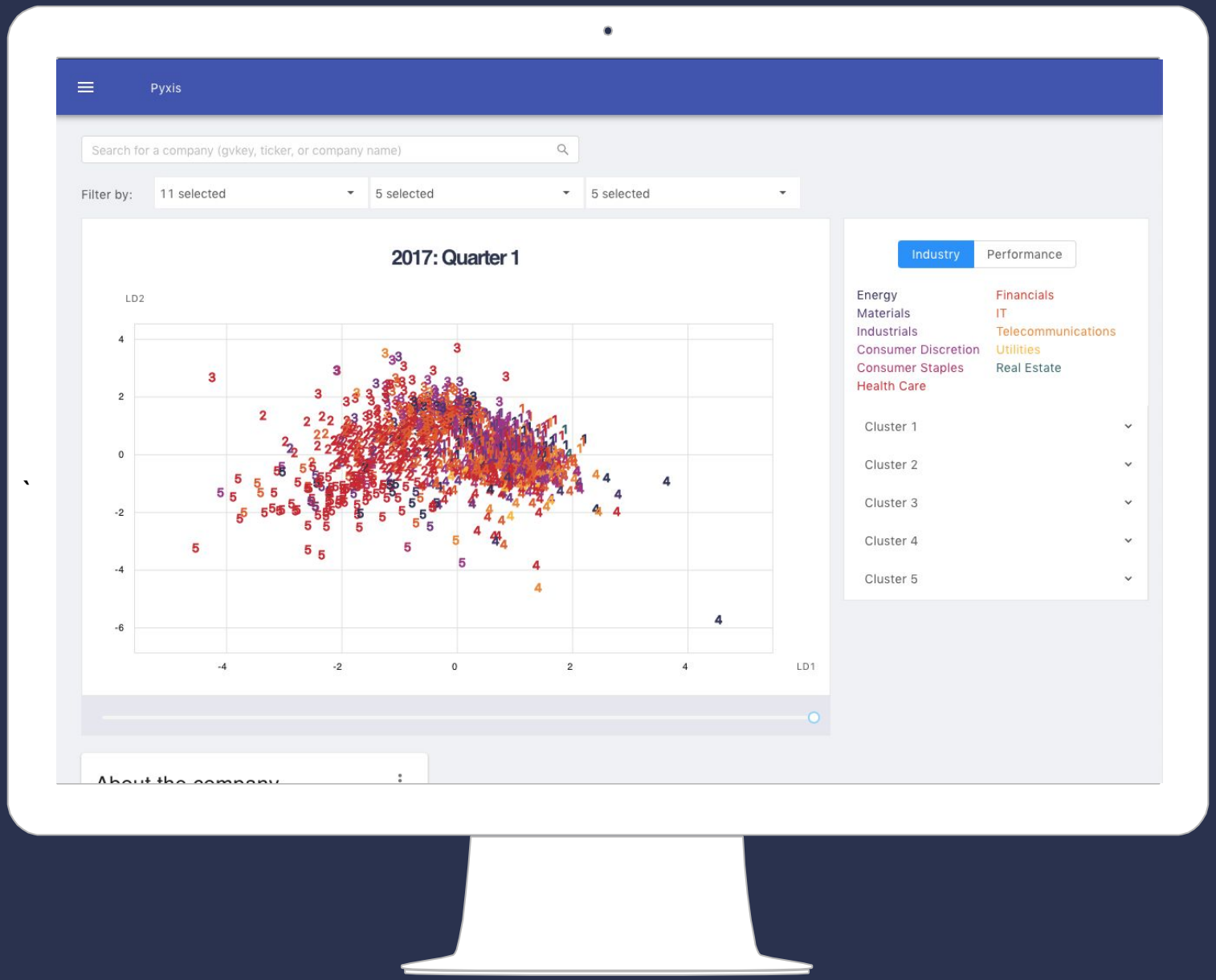
What is the problem?

“If only I had a data-driven product to point me to the right **direction**… and save me **time**…”

OUR STORY

How can a manager have a **differentiated view** of markets **that is accurate**?

http://tiffapedia-pyxis.herokuapp.com/analytics/



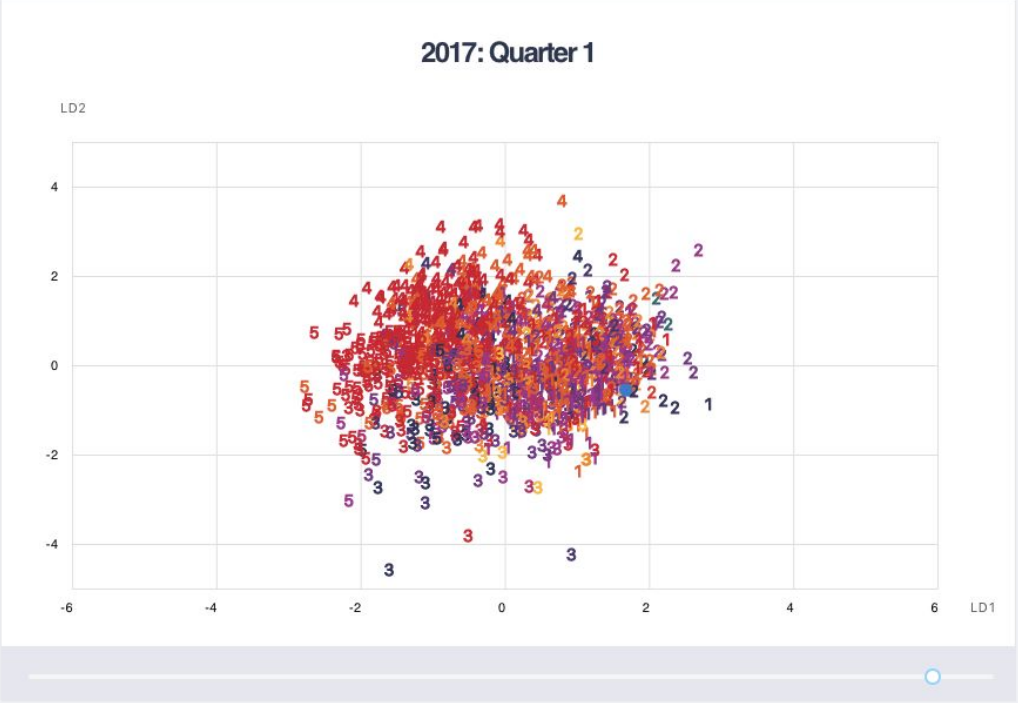
Pyxis

Pyxis

apple inc

(AAPL | 1690): APPLE INC

Filter by: 11 selected | 5 selected | 5 selected



Industry | Performance

Energy
Materials
Industrials
Consumer Discretion
Consumer Staples
Health Care
Financials
IT
Telecommunications
Utilities
Real Estate

Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5

About the company

APPLE INC

Learn more how APPLE INC performs over time.

PYXIS

Providing equity portfolio managers differentiated insights of the market via clustering.

WHO WE ARE
About us
Contact us

SUPPORT
FAQ
Report a bug

Predicting Forward Performance:
Y Labels

Labels - Relative Performance vs S&P 500

Relative Performance	< -33%	-33% to -10%	-10% to 10%	10% to 33%	> 33%
Y Label	-2	-1	0	1	2

Horrible

Fantastic

Iteration Changes

- Website
  - PostgresDB + Django Backend + React Frontend (best frameworks standard)
- Data Pipeline
  - ~ 2000 points per quarter, over 500k points total (350 MB data)
    -> 1.5 hours to clean and put in DB
  - SVG loads on computer's time, not on cloud's time
  - Extract data at the beginning (7 min load)
    -> whittle down to the bare necessities and call as needed (3 sec load)
- Visual
  - Remove anything unnecessary that can cause distractions
  - Progressive disclosure of information: filter, slider, tooltip, menu
  - Designated location: filter functionalities vs data vis vs legend vs detailed information
  - Left to Right
  - Allow multiple filters
  - Consistency