Designing, Testing, and Evaluating a Trading Algorithm

Melissa King, Maddie Rubin, Sam Seelig & Julia Leodori

Choosing Features

ECRPUS 1Y Index

- United States recession probability forecast
- Forecasts derived from monthly and quarterly Bloomberg surveys
- High probability of recession → stock prices falling

SPXSFRCS Index

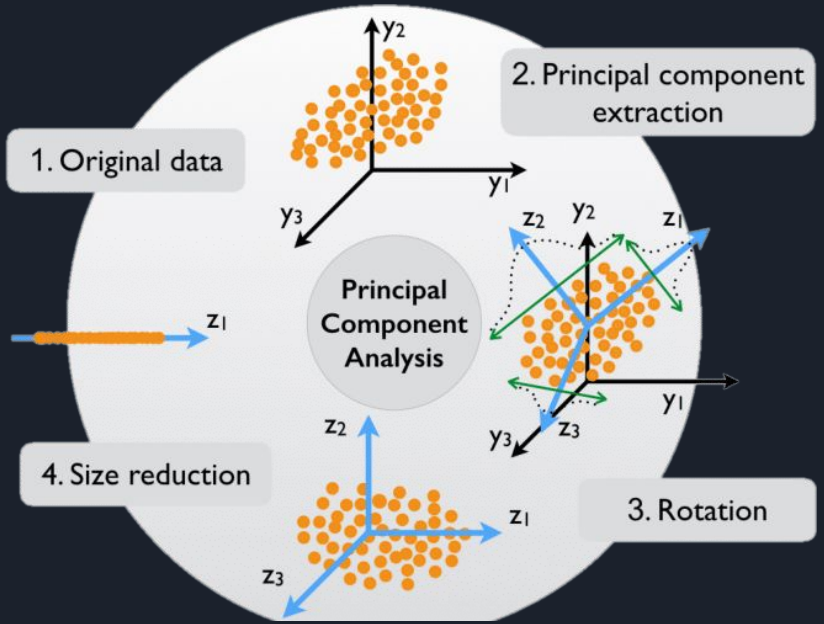
- S&P 500 Forecasts From Strategists Index
- Average year-end forecast for the S&P 500 Index
- Compiled from survey of Wall Street strategists by Bloomberg reporters

FDTRFTRL Index

- Lower bound of the Federal Funds short term target interest rate
- Reported intraday

Bloomberg

Dimensionality Reduction - PCA

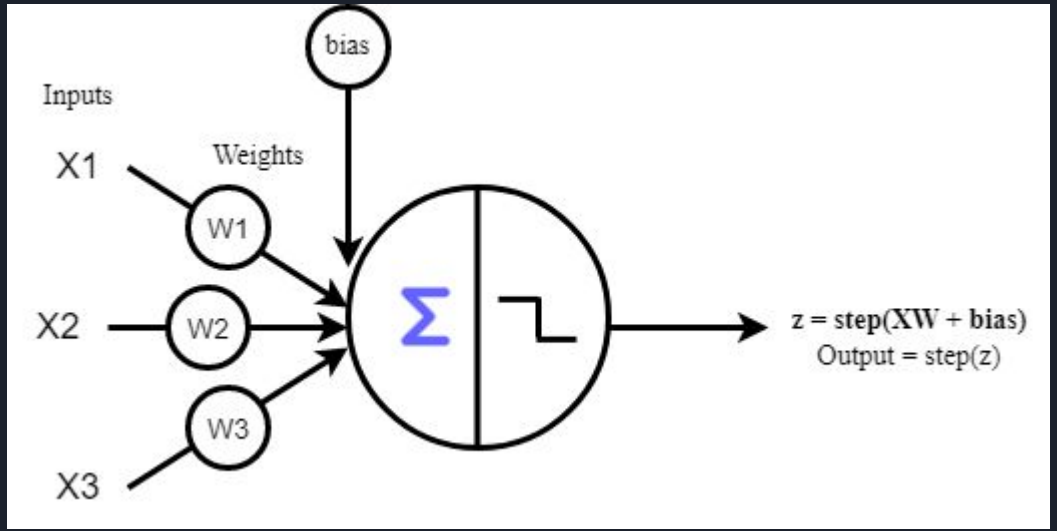


source: research gate

Dimensionality Reduction - PCA

- We decided to use PCA to reduce our data, because we are well acquainted with this method and it would allow us to easily reduce our data into a singular dimension for classification.
- We set n_components = 1, because we needed 1D data for the perceptron, and we are already working in very few dimensions (3) with our initial features. We think this was a safe bet, that would allow our data to be manageable without losing much of its character.
- In the future, we might also consider adding in more principal components, or even experimenting with the combination of features (i.e. two instead of three).

Choosing a Classifier - Perceptron

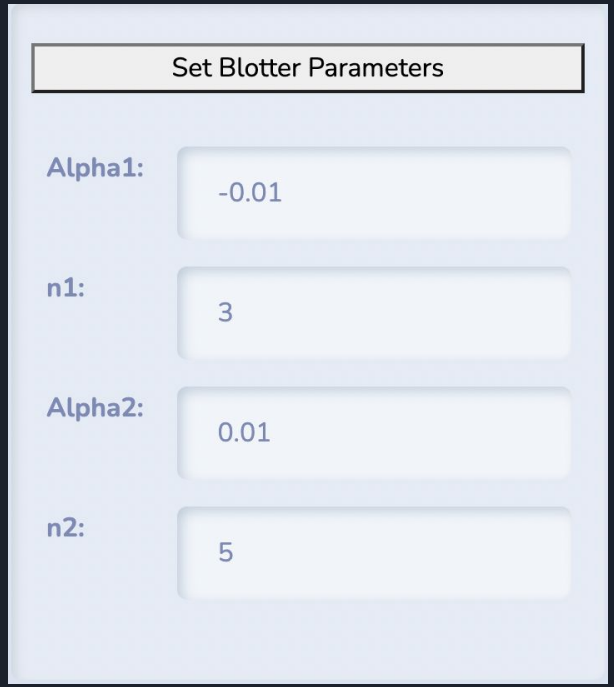


Source : pycodemates.com

Choosing a Classifier - Perceptron

- We chose to classify using the python Perceptron package
  - We experimented with both Linear Regression and the Perceptron, and we ultimately found that we had a better understanding of the Perceptron, and decided to move forward with it.
  - Additionally, we are not interested in the probability of each classification. We just need a definitive classification
- Preset Lookback Window = 26
  - Considering that two of our three features updated on a monthly basis, we wanted to ensure that our lookback window was larger than a month in business days.
  - Yet, we also wanted to make sure that we weren't overfitting our data or taking away too general of trends using a larger lookback window.
  - After some trial and error, we landed on 26
- Eta = 0.1
  - We think it would be better to underfit than overfit to our data, given that we would prefer it to drop good dates rather than keep bad dates. As such, we kept the learning rate low, at 0.1

Strategy



Keeping with our previous assignments, we decided to continue with the following trading strategy:

1. Submit a **buy limit order** at (1+Alpha1) * close_price
2. If the **limit order** does not fill within n1 = 3 days→ cancel order
3. If the **buy limit order** fills → issue a **sell limit order** at (1+Alpha2) * entry_price
4. If the **exit order** does not fill within n2 = 5 days → cancel order and issue a **market order** to sell (assuming the day's closing price)

Blotter

5	2020-01-10	IVV	ENTER	BUY	LMT	324.77	SUBMITTED
5	2020-01-14	IVV	ENTER	BUY	LMT	324.77	CANCELLED
6	2020-01-13	IVV	ENTER	BUY	LMT	323.87	SUBMITTED
6	2020-01-15	IVV	ENTER	BUY	LMT	323.87	CANCELLED
7	2020-01-14	IVV	ENTER	BUY	LMT	326.14	SUBMITTED
7	2020-01-16	IVV	ENTER	BUY	LMT	326.14	CANCELLED
8	2020-01-15	IVV	ENTER	BUY	LMT	325.63	SUBMITTED
8	2020-01-17	IVV	ENTER	BUY	LMT	325.63	CANCELLED
9	2020-01-16	IVV	ENTER	BUY	LMT	326.36	SUBMITTED
9	2020-01-21	IVV	ENTER	BUY	LMT	326.36	CANCELLED

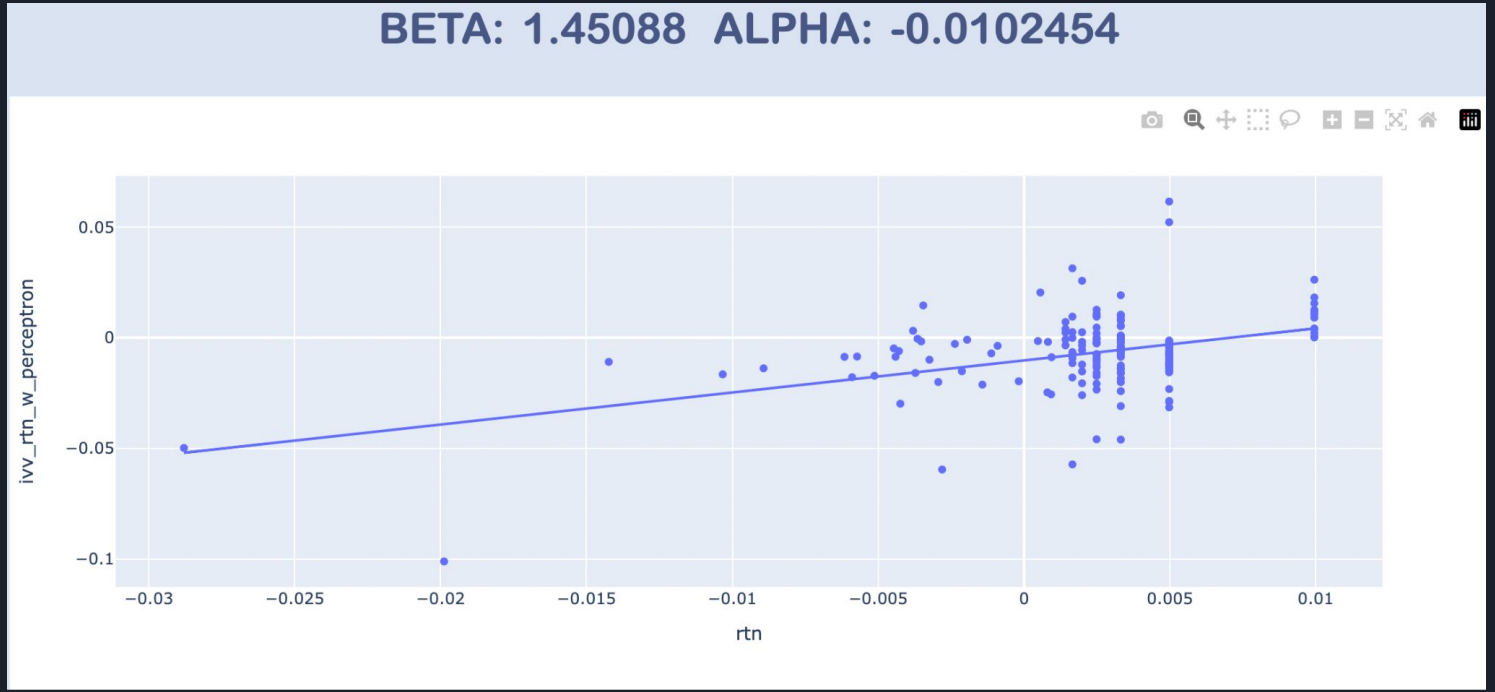
Ledger (Dumb)

trade_id	asset	dt_enter	dt_exit	success	n	rtn
1	IVV	2020-01-06		0	3	
2	IVV	2020-01-07		0	3	
3	IVV	2020-01-08		0	3	
4	IVV	2020-01-09		0	3	
5	IVV	2020-01-10		0	3	
6	IVV	2020-01-13		0	3	
7	IVV	2020-01-14		0	3	
8	IVV	2020-01-15		0	3	
9	IVV	2020-01-16		0	3	

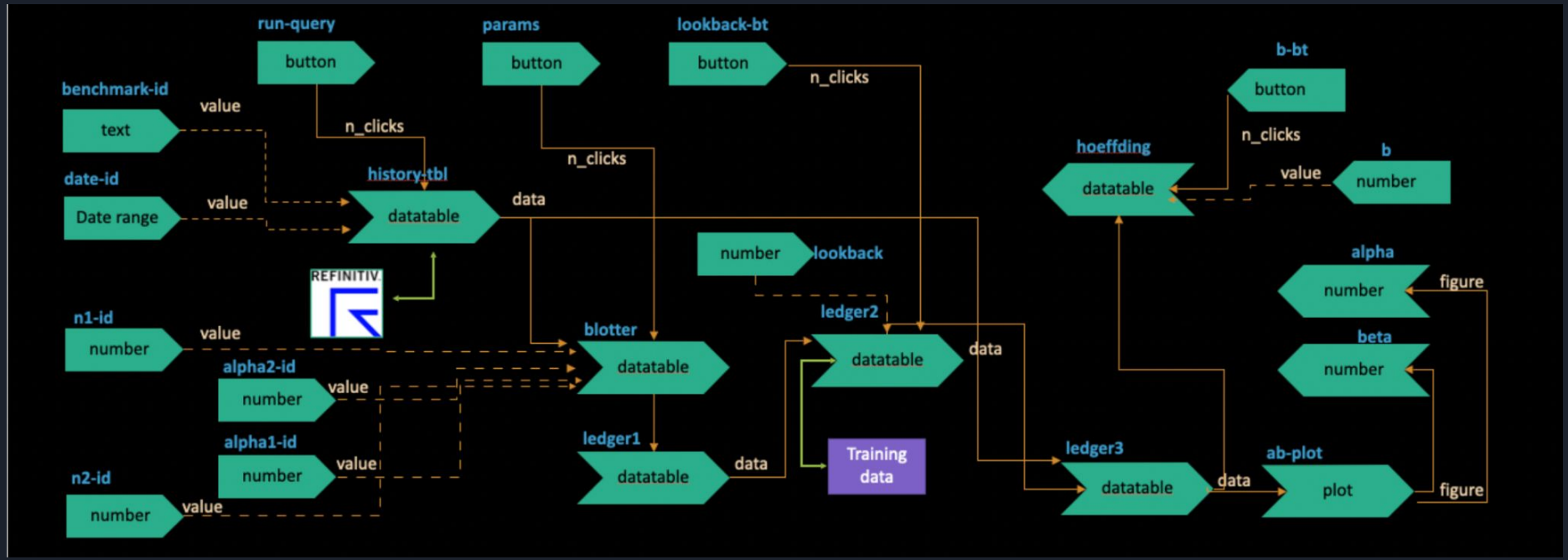
Ledger (Smart)

trade_id	asset	dt_enter	rtn	success	prediction
26	IVV	2020-02-11		0	0
27	IVV	2020-02-12		0	0
28	IVV	2020-02-13		0	0
29	IVV	2020-02-14		0	0
30	IVV	2020-02-18	-0.014585640267478724	-1	-1
31	IVV	2020-02-19	-0.01546461308268613	-1	-1
32	IVV	2020-02-20	-0.01984425148104786	-1	-1
33	IVV	2020-02-21	-0.020733913683343774	-1	-1
34	IVV	2020-02-24	-0.011041426579481532	-1	-1

Benchmark Chart for Alpha & Beta



Reactive Graph



Analysis using Hoeffding's Inequality

- Using a = 1% and b = -30%, we found Hoeffding's inequality to be 20.8%
- This means that there is at most a 30% chance that our strategy returns deviated from IVV by pure chance.
- As such, under these assumptions of alpha and beta, we are willing to keep our strategy in place.
- Of course, -30% as a lower bound is a very liberal constraint. We may want to tighten this to be more confident in our algorithm.

Lower Bound:	-0.40	-0.30	-0.20	-0.15
Hoeffding:	11.9%	20.8%	45.35%	78.15%