



# Aiolux Inflation-related Economic Indicators for Sectors Model Project

William Dai, Jolina Shen, Jaki Tang, Pathik Rupwate  
Advisor: Roger Moore



# Team Information



William Dai




Jolina Shen



Jaki Tang



Pathik Rupwate

- 
1. Client Introduction
  2. Business Problem
  3. Goal of Analysis
  4. Literature Review
  5. Data Sources
  6. Data and Sampling
  7. Analysis Plan
  8. Methodology for Model Evaluation  
and Validation
  9. Expected Findings

# Agenda



# Client Introduction

<https://aiolux.com>

## ***Bloomberg meets Amazon Web Services***



### Problem

Complex insights essential for investment optimization out of reach for most (except large firms & sophisticated hedge funds)

Most financial advisories have limited tech/analyst resources. Ability to do ad hoc deep dives is even more scarce. Retail investors are even more disadvantaged

Processing large amounts of information challenging with spreadsheets



### Quick Examples

- Anomalous performance/technicals detected by Artificial Intelligence (e.g. unusually poor streak) collates historical performance in similar situations

- Instant deep analysis of how stocks or sectors performed leading up to & after key events (e.g. 9/11, 2020 Presidential Election etc.)

- Portfolio Correlation Matrix intelligently suggests negatively correlated / uncorrelated assets to help diversification

# Business Problem

- Fill in the gap for tech/analyst resources
- Save time for ad hoc deep dives research to keep retail investors from being even more disadvantaged
- Retail investors would need to interpret how economic event will affect their portfolio and create investment opportunities for them.
  - CPI released every month and are broadly used by the business community to understand inflation
- **Our model is your own I-Banking Analyst, but faster**



CPI (Consumer Price Index)

# Goal of Analysis

When new CPI data is released every month, Aiolut will automatically surface insights predicting 50,100, 200-day impact of the Consumer Discretionary sector relative to the broader market (S&P 500).

## Prediction Model:

- Use CPI to build classification models by sector to predict **relative performance** to S&P 500 index
- CPI looks at cost changes from the perspective of Consumers for their day-to-day lives. It is important as a driver from the demand side, and may affect which sector investors choose to invest
- Relative Performance is important here because that strips out impact of broad market issues that may be affecting all sectors



# Literature Review - Machine learning techniques for stock market prediction

- Use ANN, SVM, DT to predict stock price with various sources of data
- SVM has the worst performance since it faces the curse of dimensionality
- ANN and Boosting Decision Trees have a better result especially after PCA
- Boosting Decision Tree has a slightly better result

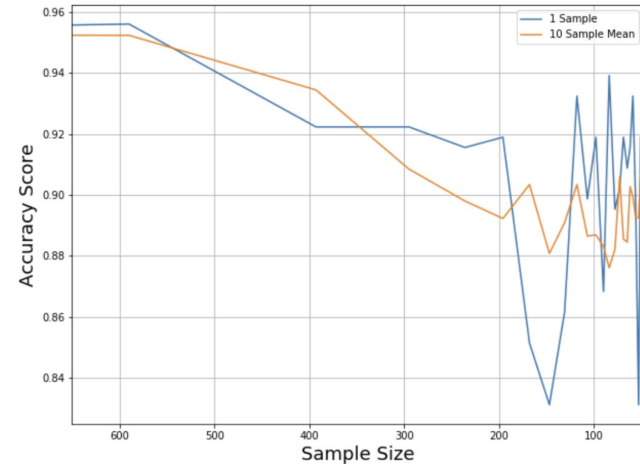
Paper	Sources for Knowledge Base			AI Approach
	Traditional	Crowd-sourcing	News	
Kimoto et al. (1990)	✓			ANN
Lee and Jo (1999)	✓			Time Series
K.-j. Kim and Han (2000)	✓			ANN, GA
K.-j. Kim (2003)	✓			SVM
Qian and Rasheed (2007)	✓			ANN, DT
S.-T. Li and Kuo (2008)	✓			ANN
Schumaker and Chen (2009)			✓	SVM
Vu et al. (2012)		✓		DT
M.-Y. Chen, Chen, Fan, and Huang (2013)	✓			ANN
Adebiyi, Adewumi, and Ayo (2014)	✓			ANN, ARIMA
Nguyen, Shirai, and Velcin (2015)		✓		SVM
Shynkevich, McGinnity, Coleman, and Belatreche (2015)			✓	ANN, SVM
Chourmouziadis and Chatzoglou (2016)	✓			Fuzzy System
<b>Our Financial Expert System</b>	✓	✓	✓	ANN, SVM, DT

Scenario #	Description
1	Market data
2	Market data, technical indicators
3	Market data, technical indicators, Wikipedia Traffic
4	Market data, technical indicators, Google news counts
5	Market data, technical indicators, Wikipedia Traffic, generated features
6	Market data, technical indicators, Google news counts, generated features
7	Market data, technical indicators, Wikipedia traffic, Google news counts, and generated features

Source: Application of machine learning techniques for stock market prediction

# Literature Review Summary - Log-F(m,m) Logit

- Log-F(m,m) logistic regression, a descendant of Firth's logit, specifically designed to handle small datasets.
- The method can deal with MLE's small sample size bias
- Log-F(m,m) logistic regression on the basis that penalized maximum likelihood estimation is mathematically identical to Bayesian analysis



Source: The Best Classifier for Small Datasets: Log-F(m,m) Logit



# Stock Price Performance Data

## Target Variable

	Date	S&P 500	S&P 500 Performance	S&P 500 Real Estate	S&P 500 Real Estate Performance	S&P 500 Real Estate Outperform/Underperform
0	22-Apr-22	4,271.78	-6.78%	308.86	0.10%	OUTPERFORM
1	1-Apr-22	4,582.64	4.36%	308.54	9.70%	OUTPERFORM
2	1-Mar-22	4,391.27	-1.01%	281.27	-4.60%	UNDERPERFORM
3	1-Feb-22	4,435.98	-3.01%	294.83	-8.33%	UNDERPERFORM
4	1-Jan-22	4,573.82	-2.16%	321.63	10.16%	OUTPERFORM

## Target Variable Data Preparation

1. Get [S&P 500 Overall](#) and S&P 500 Sectors (For example, [Real Estate](#)) Data from Aiolux API.
2. Calculate Monthly Performance for S&P 500 Overall and each S&P 500 Sectors. During implementation, we will use each Sectors' 50, 100, & 200-Day Performance comparing to S&P 500 Overall instead of Monthly Performance.
3. Compare S&P 500 Sector Performance with S&P 500 Overall Performance to decide whether it is **OUTPERFORM/UNDERPERFORM**.

# CPI Data

The screenshot shows a REST client interface with a POST request to the URL `https://api.bls.gov/publicAPI/v2/timeseries/data/?startyear=2000&endyear=2022`. The request body is a JSON object with the key `seriesid` and the value `CUSR0000SA0,CUSR0000SAC,CUSR0000SAH,CUSR0000SAH1,CUSR0000SAS4`. The response is a JSON object with the following structure:

```
{
  "status": "REQUEST_SUCCEEDED",
  "responseTime": 478,
  "message": [
    "Year range has been reduced to the system-allowed limit of 10 years."
  ],
  "Results": {
    "series": [
      {
        "seriesID": "CUSR0000SA0",
        "data": [
          {
            "year": "2009",
            "period": "M12",
            "periodName": "December",
            "value": "217.347",
            "footnotes": [
              {}
            ]
          }
        ]
      },
      {
        "year": "2009",
        "period": "M11",
        "periodName": "November",
        "value": "217.234",
        "footnotes": [
          {}
        ]
      },
      {
        "year": "2009",
        "period": "M10",
        "periodName": "October",
        "value": "216.509",
        "footnotes": [
          {}
        ]
      }
    ]
  }
}
```

**Data source:** US Bureau of Labor Statistics [API](#)

## API Request:

- **Input:** Series IDs (e.g. CUSR0000SAH), Start Year, End Year.
- **Output:** Monthly CPI values by Series ID.

## CPI Data Preparation

1. Run Python Script to Request JSON data from API.
2. Parse JSON Data to Pandas DataFrame.
3. Each S&P 500 Sector would have several related CPIs. Need to join them together.

# CPI Data

	year	period	overall	commodities	housing	shelters	transportations
0	2022	M03	287.708	219.353	293.668	346.516	341.104
1	2022	M02	284.182	214.826	291.549	344.758	334.305
2	2022	M01	281.933	212.082	290.151	342.974	329.726
3	2021	M12	280.126	210.452	288.259	341.963	326.397
4	2021	M11	278.524	208.467	286.849	340.475	326.256
5	2021	M10	276.590	206.035	285.453	338.865	323.995
6	2021	M09	274.214	203.215	283.532	337.298	323.329
7	2021	M08	273.092	201.975	281.979	335.888	326.470
8	2021	M07	272.184	200.804	280.900	335.262	330.441
9	2021	M06	270.955	199.370	279.750	333.807	333.374

Each S&P 500 Sector would be related to several kinds of CPI data.

## Example of S&P 500 Real Estate:

- Overall CPI
- Commodities CPI
- Housing CPI
- Shelter CPI
- Transportation CPI

# Gaps & Concerns in Data

S&P 500 data is from Dec 1946 but S&P 500 Sectors is only from Jan 1999.

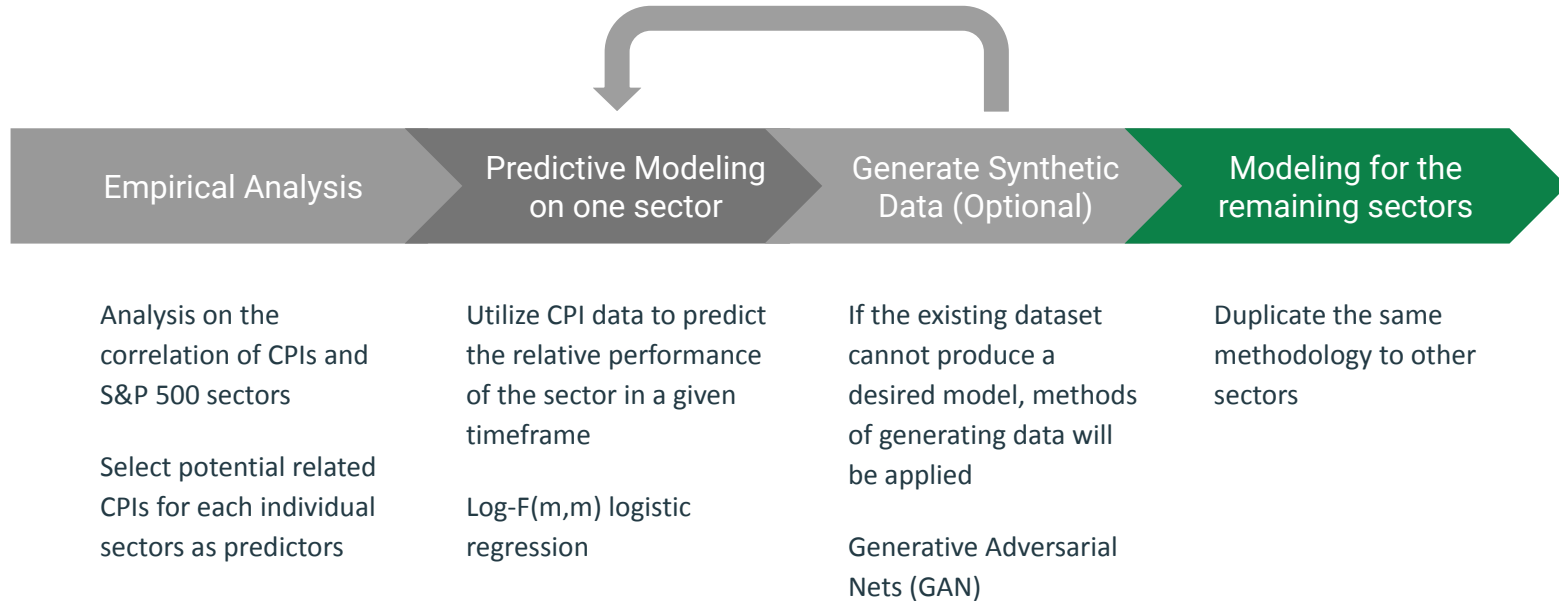
CPI data only available per month.

Therefore, the number of our data would be around 267 per sector.

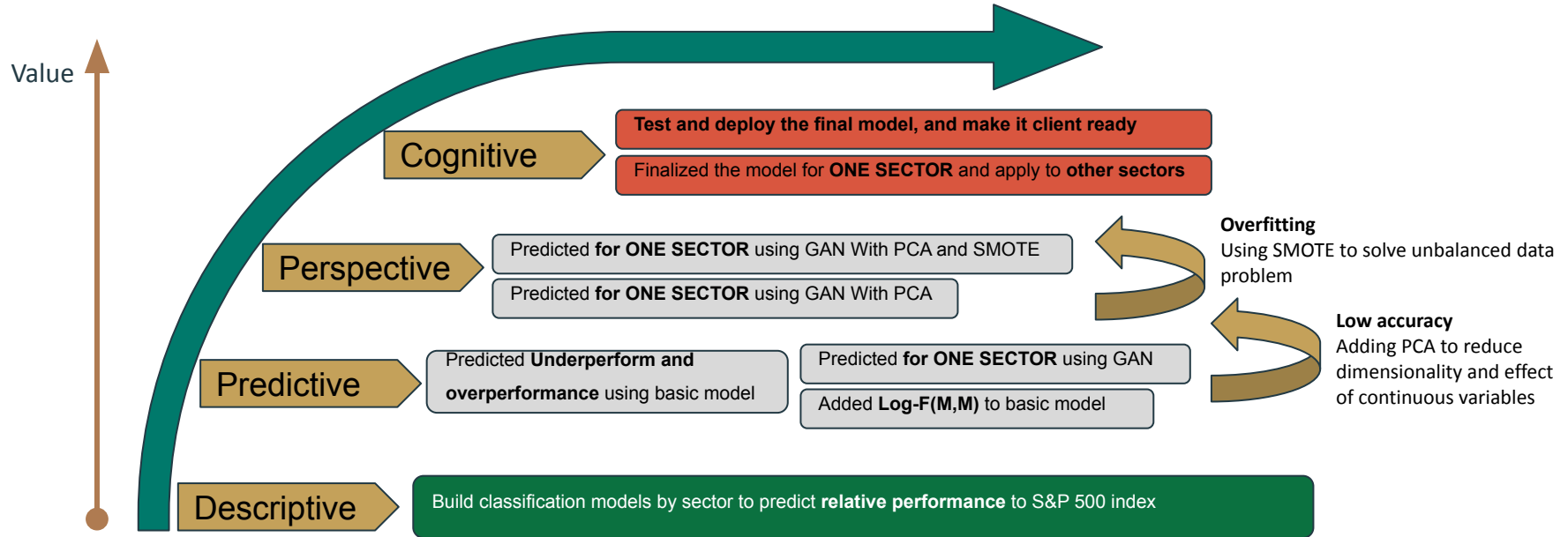
The Problems with Small Datasets:

1. They contain less information, so they produce less accurate models.
2. Small samples are much less likely to accurately reflect the population's distribution.
3. Maximum likelihood estimation procedure that powers many classification algorithms is only unbiased with large datasets.

# Analysis Roadmap

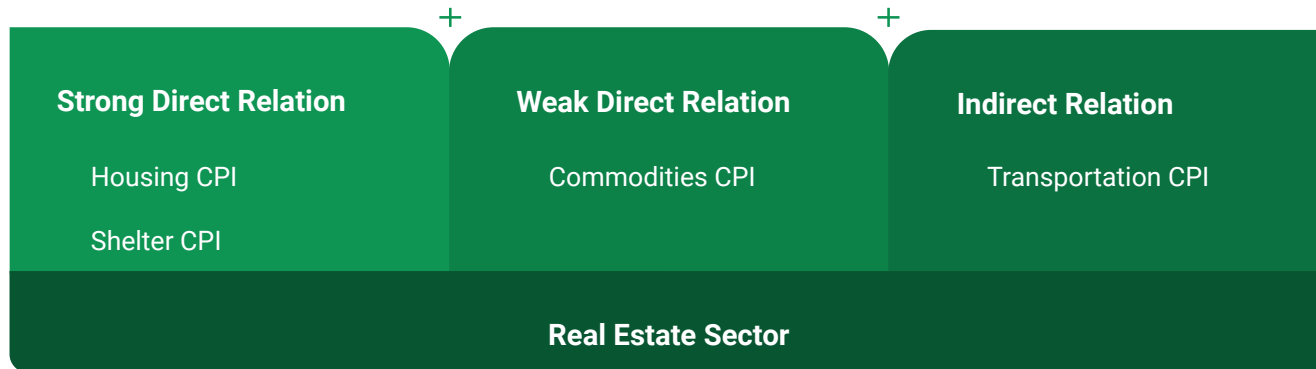


# Value Creation



# Empirical Analysis on CPIs

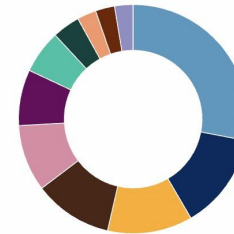
- Through official definition and calculation of different CPIs to define related ones to each S&P 500 sector:
  - Strong direct relation
  - Weak direct relation
  - Indirect relation
- Based on the level of relationship, different weights would be assigned for each corresponding CPI in modeling phase



# Predictive Modeling: Classification on sector performance

- Leverage CPI information to predict the influence of different CPIs on relative performance of each S&P 500 sector
- Utilize the result from empirical analysis to filter predictors for each sector
- In total 33 models: 11 sectors, 3 predictive performance
  - Predictive performance: A sector's relative 50, 100, & 200-day performance comparing to S&P

## Sector Breakdown

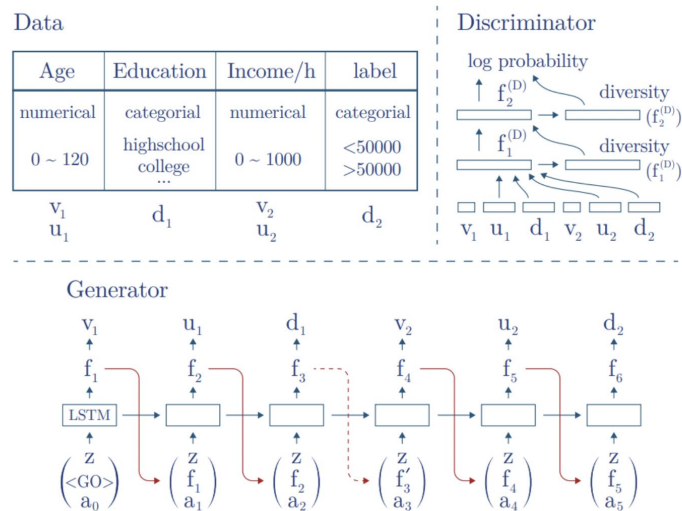


SECTOR	INDEX WEIGHT
Information Technology	28.0%
Health Care	13.6%
Consumer Discretionary	12.0%
Financials	11.1%
Communication Services	9.4%
Industrials	7.9%
Consumer Staples	6.1%
Energy	3.9%
Utilities	2.7%
Real Estate	2.7%
Materials	2.6%



# Quantitative Methodology: Generative Adversarial Nets (GANs)

- Composes of two deep networks: the generator and the discriminator and both of them simultaneously trained
- The task for the generator is to generate samples, which won't be distinguished from real samples by the discriminator
- Tabular GAN:
  - Generator: Generate a numerical variable in 2 steps.
    - generate the value scalar  $V$
    - generate the cluster vector  $U$  eventually applying tanh
  - Discriminator: Multi-Layer Perceptron (MLP) with LeakyReLU and BatchNorm
    - The first layer used concatenated vectors  $(V, U, D)$  among with mini-batch diversity with feature vector from LSTM
    - The loss function is the KL divergence term of input variables with the sum ordinal log loss function

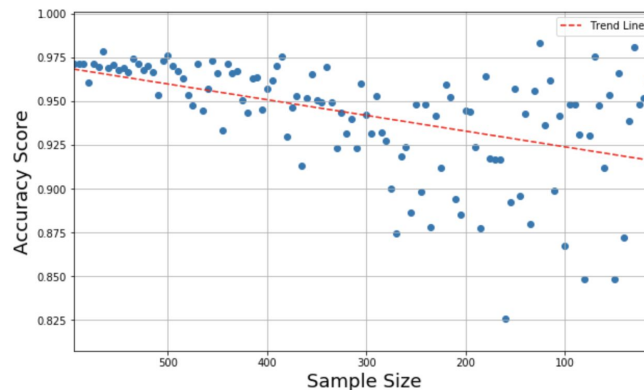


Example of using TGAN to generate a simple census table

# Quantitative Methodology: Log-F(m,m)

$$\log-F(m,m)(\beta) = J(\beta) + \sum \frac{m}{2} m\beta - m(\ln(1 + e^{\beta}))$$

- A variation of logistic regression that specifically designed to handle small datasets
- Modify the procedure to prevent bias by introducing a penalty that uses the log-F distribution as the basis
- The method can be realized using an augmented version of the data as input for normal logistic regression



	const	X1	X2	X3	y	sample_weights
0	1	0.884370	0.583869	0.481830	0	1.0
1	1	0.752774	0.569937	0.865408	1	1.0
2	1	0.677080	0.101350	0.362602	1	1.0
3	0	1.000000	0.000000	0.000000	0	0.5
4	0	1.000000	0.000000	0.000000	1	0.5
5	0	0.000000	1.000000	0.000000	0	0.5
6	0	0.000000	1.000000	0.000000	1	0.5
7	0	0.000000	0.000000	1.000000	0	0.5
8	0	0.000000	0.000000	1.000000	1	0.5

# Result Presenting - Log-F(m,m)

```
Call:
lm(formula = S.P.500.Real.Estate..Sector..Performance.... ~ housing +
  shelters + transportations, data = train)
```

Residuals:

Min	1Q	Median	3Q	Max
-18.8121	-1.9261	0.0676	1.9340	21.1126

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-15.40697	49.10776	-0.314	0.754
housing	0.29266	1.07348	0.273	0.786
shelters	-0.23798	0.77409	-0.307	0.759
transportations	0.04310	0.08335	0.517	0.606

Residual standard error: 5.413 on 95 degrees of freedom  
(1 observation deleted due to missingness)  
Multiple R-squared: 0.003065, Adjusted R-squared: -0.02842  
F-statistic: 0.09736 on 3 and 95 DF, **p-value: 0.9613**

Log-f(m,m)

Regular Linear Regression

```
logistf(formula = S.P.500.Real.Estate..Sector..Performance.... ~
  housing + shelters + transportations, data = train)
```

Model fitted by Penalized ML  
Coefficients:

	coef	se(coef)	lower 0.95	upper 0.95	chisq	p	method
(Intercept)	-71.6383925	22.32874066	-119.49181843	-29.8484777	12.074446	0.0005111766	2
housing	1.3749921	0.47644031	0.48181322	2.3926749	9.533493	0.0020175580	2
shelters	-1.0708124	0.34974920	-1.82142039	-0.4162954	10.903922	0.0009596086	2
transportations	0.1442282	0.04625694	0.06035121	0.2507732	13.399036	0.0002517536	2

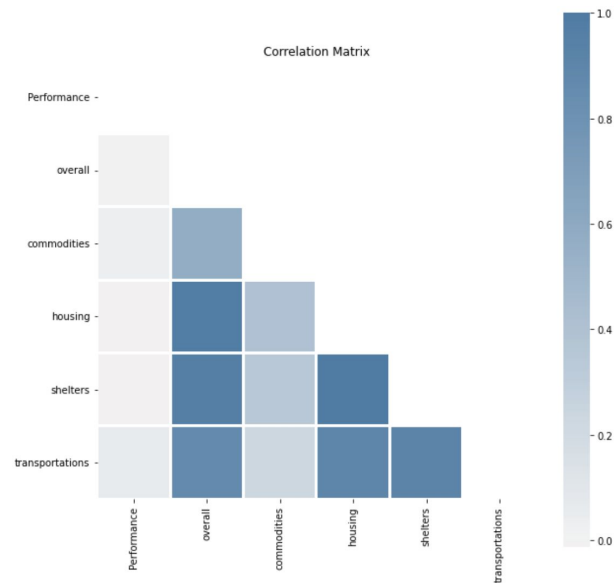
Method: 1-wald, 2-Profile penalized log-likelihood, 3-None

Likelihood ratio test=17.08671 on 3 df, **p=0.0006783051**, n=99  
wald test = 14.17173 on 3 df, **p = 0.002680483**

# Result Presenting - Base Model

Info	
Rows	111
Columns	7
Size in Memory	6.9 KB

	Data Type	Nulls	Zeros	Min	Median	Max	Mean	Standard Deviation	Unique	Top Frequency
Performance	int64	0	0	0	0	1	0.48	0.50	2	58
month	object	0	0						12	10
overall	float64	0	0	231.015	244.16	280.13	247.15	12.17	110	2
commodities	float64	0	0	177.51	185.19	210.45	185.81	5.99	111	1
housing	float64	0	0	223.46	250.41	288.26	251.55	17.73	111	1
shelters	float64	0	0	258.23	296.46	341.96	297.23	24.37	111	1
transportations	float64	0	0	275.043	306.70	333.37	304.67	17.17	111	1



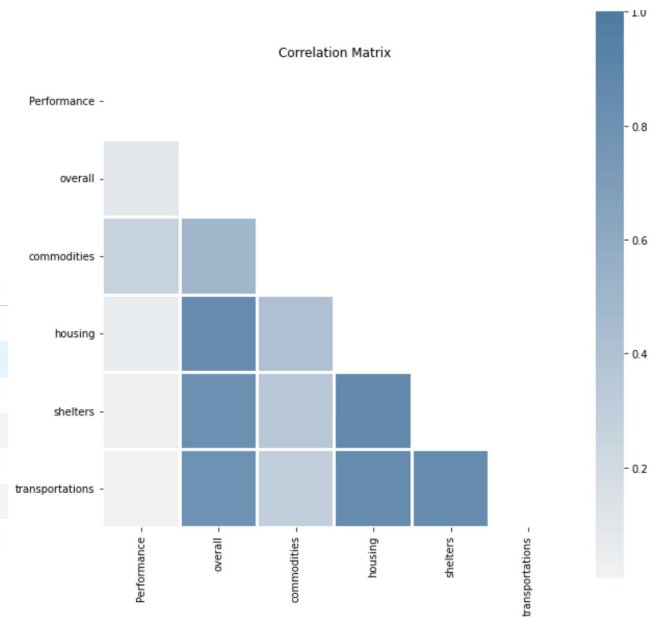
# Result Presenting - Base model with PCA

	Name	CV Mean Accuracy	CV Mean Precision	CV Mean Recall	CV Mean F-1 Score	Accuracy	Precision	Recall	F-1 Score
0	LogisticRegression	0.566667	0.6125	0.52	0.495177	0.5	0.5	0.357143	0.416667
1	KNeighborsClassifier	0.524242	0.477143	0.526667	0.482814	0.535714	0.533333	0.571429	0.551724
2	DecisionTreeClassifier	0.523485	0.462976	0.59	0.496947	0.357143	0.375	0.428571	0.4
3	RandomForestClassifier	0.496212	0.405079	0.44	0.391648	0.535714	0.529412	0.642857	0.580645
4	AdaBoostClassifier	0.450758	0.330556	0.333333	0.306197	0.535714	0.529412	0.642857	0.580645
5	GradientBoostingClassifier	0.532576	0.505278	0.52	0.484567	0.607143	0.6	0.642857	0.62069
6	ExtraTreesClassifier	0.478788	0.447222	0.473333	0.418846	0.535714	0.533333	0.571429	0.551724
7	BaggingClassifier	0.496212	0.372341	0.396667	0.344249	0.535714	0.533333	0.571429	0.551724
8	SVC	0.568182	0.552698	0.53	0.492489	0.5	0.5	0.357143	0.416667
9	GaussianNB	0.568182	0.577698	0.66	0.582796	0.428571	0.4375	0.5	0.466667

# Result Presenting - GAN

Info	
Rows	10000
Columns	7
Size in Memory	547.0 KB

	Data Type	Nulls	Zeros	Min	Median	Max	Mean	Standard Deviation	Unique	Top Frequency
Performance	int64	0	0	0	1	1	0.51	0.50	2	5054
month	object	0	0						12	4615
overall	float64	0	0	223.93	247.32	291.58	247.15	13.49	10000	1
commodities	float64	0	0	177.51	186.73	222.65	189.029	8.30	9998	2
housing	float64	0	0	209.74	253.14	299.37	252.49	20.41	10000	1
shelters	float64	0	0	239.63	294.94	352.74	296.44	25.12	10000	1
transportations	float64	0	0	259.77	308.54	337.27	301.12	18.96	10000	1



# Result Presenting - GAN with PCA

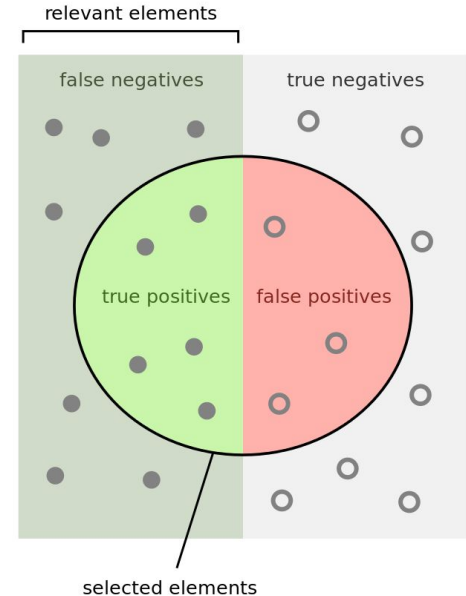
Variance: Projected dimension

62.0%: 0.05 \* month + 0.48 \* overall + 0.24 \* commodities + 0.49 \* housing + 0.49 \* shelters + 0.47 \* transportations  
 17.4%: 0.91 \* month + -0.07 \* overall + 0.36 \* commodities + -0.04 \* housing + 0.01 \* shelters + -0.18 \* transportations  
 13.3%: -0.39 \* month + -0.05 \* overall + 0.89 \* commodities + -0.08 \* housing + -0.14 \* shelters + -0.14 \* transportations  
 3.0%: -0.06 \* month + 0.60 \* overall + -0.09 \* commodities + 0.19 \* housing + 0.00 \* shelters + -0.77 \* transportations  
 2.5%: -0.11 \* month + -0.55 \* overall + -0.01 \* commodities + 0.21 \* housing + 0.71 \* shelters + -0.37 \* transportations  
 1.8%: -0.02 \* month + 0.30 \* overall + 0.02 \* commodities + -0.82 \* housing + 0.49 \* shelters + 0.03 \* transportations

	Name	CV Mean Accuracy	CV Mean Precision	CV Mean Recall	CV Mean F-1 Score	Accuracy	Precision	Recall	F-1 Score
0	LogisticRegression	0.747	0.691146	0.755804	0.721969	0.7312	0.660136	0.739504	0.69757
1	KNeighborsClassifier	0.729	0.690553	0.68239	0.686309	0.7156	0.658216	0.668893	0.663512
2	DecisionTreeClassifier	0.6721	0.624882	0.614723	0.619424	0.654	0.583562	0.609733	0.59636
3	RandomForestClassifier	0.7379	0.695753	0.705403	0.700378	0.7288	0.667572	0.703244	0.684944
4	AdaBoostClassifier	0.7424	0.686814	0.748666	0.716293	0.732	0.659091	0.747137	0.700358
5	GradientBoostingClassifier	0.7523	0.698707	0.756962	0.726553	0.74	0.668074	0.754771	0.708781
6	ExtraTreesClassifier	0.7362	0.695533	0.698726	0.69703	0.7288	0.67098	0.692748	0.68169
7	BaggingClassifier	0.7211	0.696134	0.635437	0.664259	0.7164	0.67349	0.627863	0.649877
8	SVC	0.7478	0.687468	0.769613	0.726187	0.732	0.656977	0.754771	0.702487
9	GaussianNB	0.7469	0.690315	0.757644	0.722362	0.7312	0.659593	0.741412	0.698113

# Model Evaluation and Validation

- Use k-fold cross validation
- Classification accuracy
- Precision and recall
  - Recall is especially useful as we need to measure how good our model is at correctly predicting the classes.
- F1 score
- ROC curve and AUC
  - We aim to increase the true positive rate (TPR) while keeping false positive rate (FPR) low. How many false positives we can tolerate?
  - The closer the AUC is to 1, the better the classifier is.



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$



# Expected Findings

## *Does the sector outperform the broader market?*

- Predicting performance of a sector over the next 50, 100 & 200 trading days relative to the the broader market (S&P 500).
- Does the sector outperform or underperform compared to the overall S&P 500?

This should help potential investors decide which sectors to invest in and for what period of time.

Sector	50 Days	100 Days	200 Days
Real Estate	Underperform	Underperform	Overperform
Technology	Overperform	Overperform	Overperform

# Data bias and Ethics Discussion

## Possible biases in our models:

- Data Preparation: We use GAN to generate synthetic data. Any biases in the generating distribution enlargen in the augmented dataset.
- Model development: Which predictors so we use in our dataset can contribute to aggregation bias.
- Model evaluation: Which metrics do we use to select the best models? We must get the most accurate predictions for our clients.
- We use CPI and S&P 500 data to model our predictions, therefore we can assume that the training data does match our intended population. However, using GAN can introduce biases which we reduce to significant levels.
- We believe that our models are ethical. We use existing data to predict which sectors will perform well compared to the S&P500 to give our clients the information on where they should invest their portfolios.

Thank you!

Q&A

# Reference

<https://www.iosrjournals.org/iosr-jce/papers/Vol18-issue1/Version-3/D018132226.pdf>

<https://www.mdpi.com/2227-7072/7/2/26>

<https://arxiv.org/abs/1801.02143>

<https://medium.com/@remycanario17/log-f-m-m-logit-the-best-classification-algorithm-for-small-datasets-fc92fd95bc58>

<http://etd.auburn.edu/bitstream/handle/10415/5652/Application%20of%20machine%20learning%20techniques%20for%20stock%20market%20prediction.pdf?sequence=2&isAllowed=y>

<https://towardsdatascience.com/review-of-gans-for-tabular-data-a30a2199342>

AIOLUX API:

<https://us-central1-alpine-agent-164921.cloudfunctions.net/daily-prices?symbol=XLK&from=2005-12-01&to=2006-01-01>

<https://aiolux.com/detail/time-machine-historical-summary?date=2022-03-26>