Textual Analysis

Hongfei Ge

Objective:

- Analyze 10ks of a large-cap company
- Build up a portfolio of ETFs to identify the trend of sector rotation through topic modeling

Data:

10 Ks

Process:

Topic Modeling

Classify Sectors

Result:

Portfolio

Menu

- Data Pre-process & LDA – Hongfei, Huan
- Classification – Junpeng, Shuwen
- Back Test – Xiang, Peijie

Data

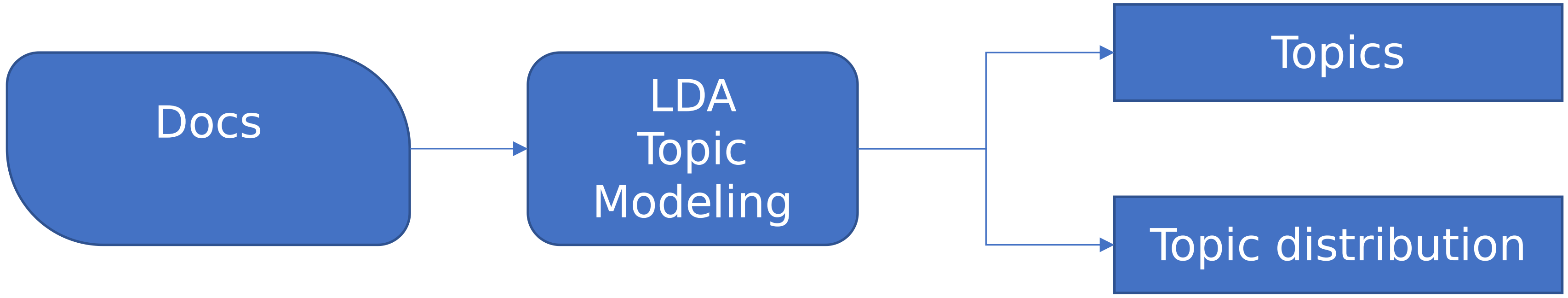
Textural Data Mining

- Download10 K & 10Q reports from 11 sectors, top 10 largest companies from each sector
- Extract text describing companies’ tactics for the market from 10K & 10Q reports
- Use pre-processed text data to generate feature matrix
- Feature matrix (bag of words & TFIDF)

Problems Encountered

- How to implement appropriate method to extract text data?
  - Xpath, Beautiful soup, Regex
  - x1 = re.search("(?m)^.{0,7}item.+[\n]{0,3}.*[\n]{0,3}business", text)

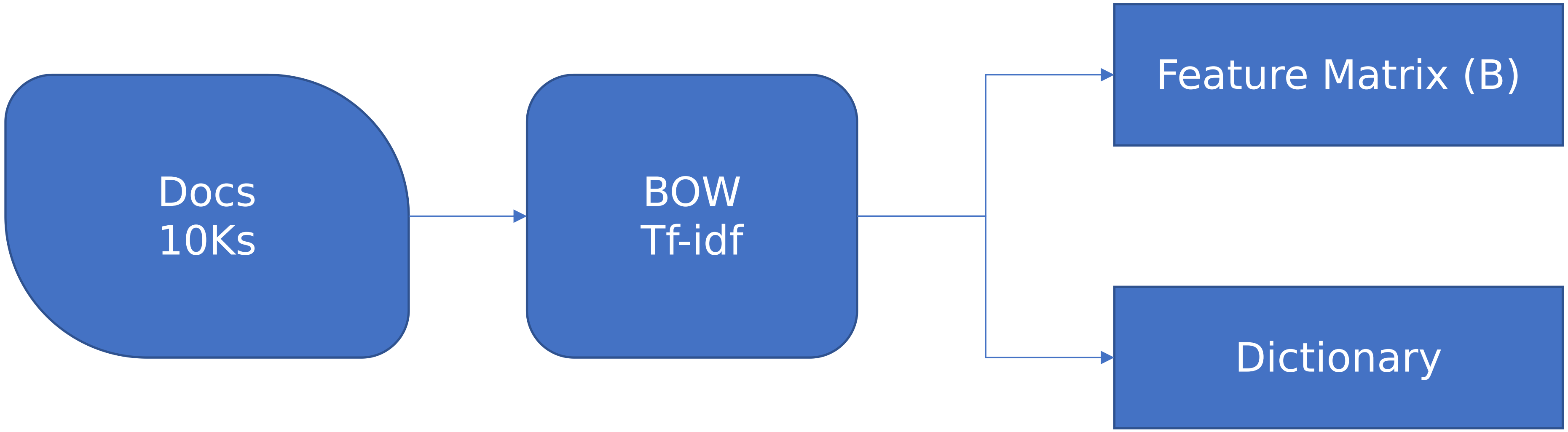
Methodology



- Words Distribution
  - Quantify text
  - 10Ks of 11 industries
  - A long vector of N elements with sum of 1
  - BOW dictionary

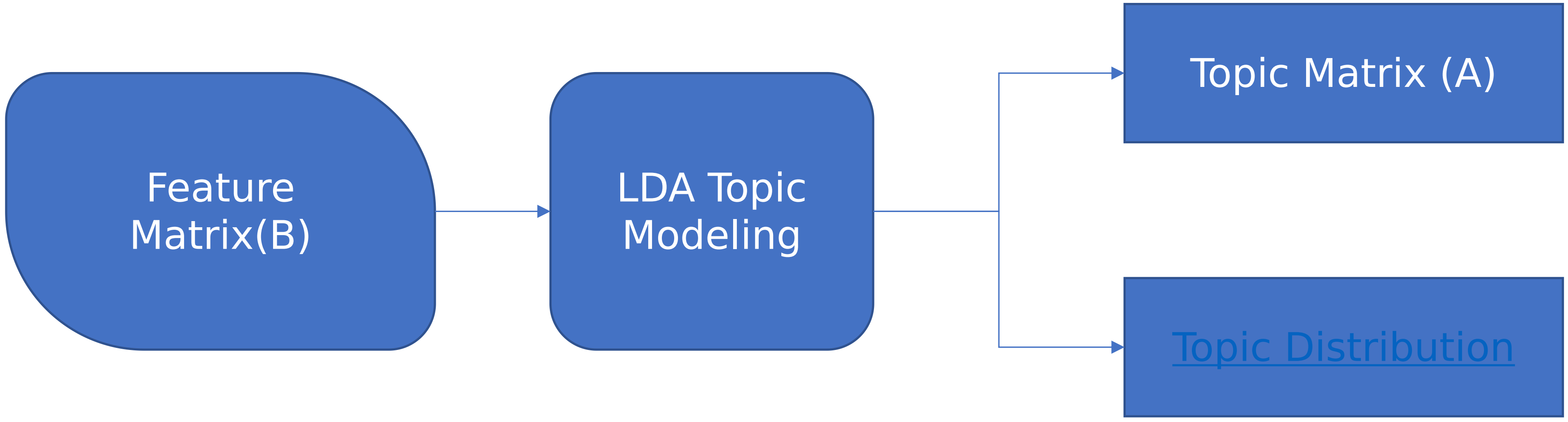
- Feature Matrix
  - tf-idf
  - LDA
  - Classify topics into 11 industries

  - $Amazon2018_{10k} = \mathrm{a}_1\mathrm{T1} + \mathrm{a}_2\mathrm{T2} + \ \ldots\ + \mathrm{a}_{60}\mathrm{T60} \quad (*)$
  - $(*) = b_1 Ind_1 + b_2 Ind_2 + \ldots + b_{11} Ind_{11}$



- Bag of Words (BOW)
- TF-IDF

- Feature Matrix
  - Frequency of the word occurred in this doc and not occurred in other doc
- Dictionary
  - Count all words occurred in all docs



- N dimensions -> 60 dimensions
- Word distribution

How to classify topics? / How to interpret topics?

Traditional way

Select top 20 most frequent words for each topic

Subjective & Tedious

“ human intelligence”

Machine learning classifier

Unsupervised learning

No direct sample training data …

Transform on B

- Transform words-count into frequency/distribution
- Similarities between feature matrix(A) and topic matrix (B)
  - Same number of columns(N)
  - Based on the same dictionary

Classifying Industries

- **Data split**
  - **Training set**
  - **Testing set**
- **Classifier algorithm selection**
  - **pros**
  - **cons**
- **Evaluation**
  - **Accuracy score**
  - **F1-score**

Classifiers

- **Naive Bayes**
- **Support Vector Machines (SVMs)**
- **Gradient Boosted Decision Trees**
- **K nearest neighbors (KNN)**

Pros and Cons of 4 Classifiers

KNN (K nearest neighbor)

- <u>Pros</u>
- No training and testing involved
- handles multiclass classification

- <u>Cons</u>
- Performs poorly on high dimensional datasets
- slow to predict new instances

GBDT

- Pros
- Robust to missing data
- Can learn non-linear hypothesis function

- Cons
- May not be fast due to many hyperparameters need to be adjusted

SVM

- Pros
- Robust against overfitting
- The optimization problem is convex and have unique solution

- Cons
- Feature scaling is required
- Many hyperparameters and not intuitive

Naive Bayes

- Pros
- it is easy to implement and much
- Require less training data
- No distribution requirements
- good for few categories variable

- Cons
- Assumes that the features are independent, which is rarely true

Example of score

Accuracy score

In [47]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: ", "%.4f" %accuracy)
```

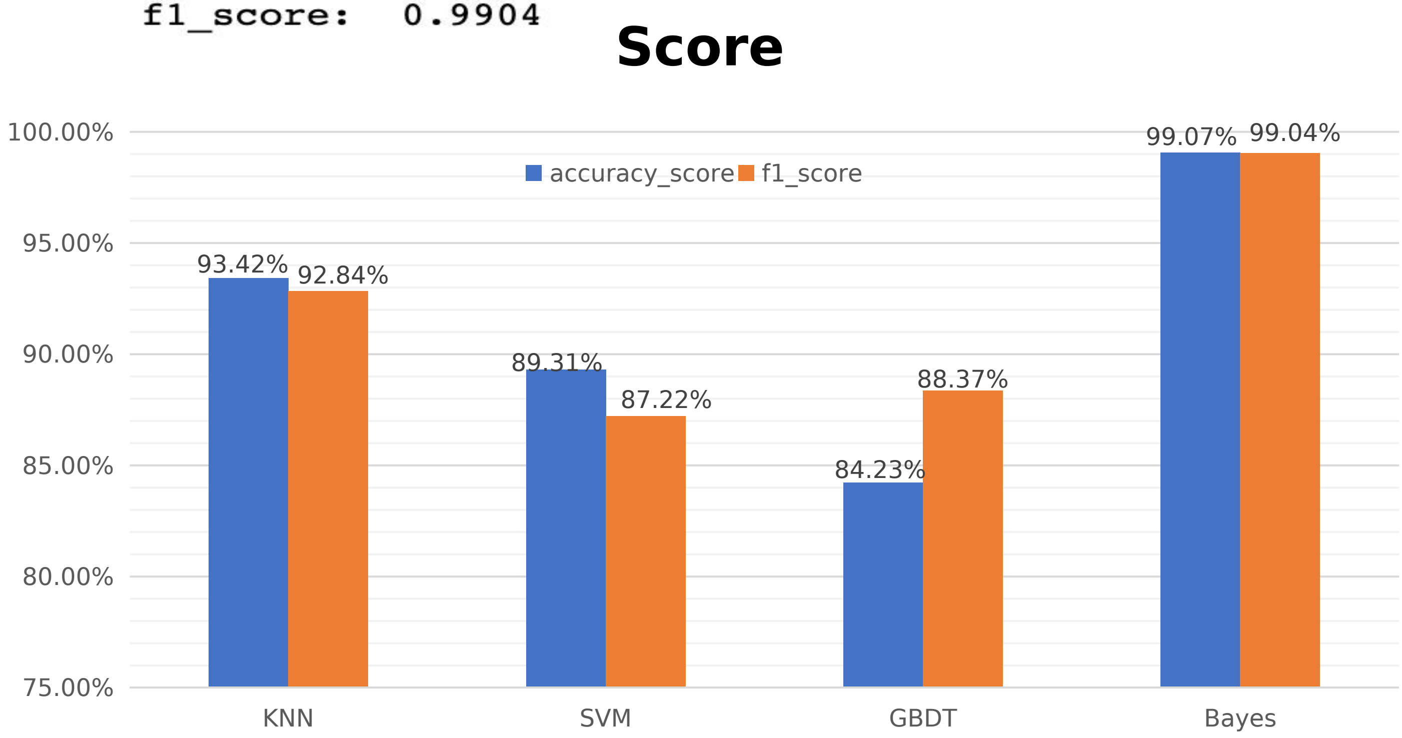
```
Accuracy:  0.9907
```

f1_score

In [48]:
```
print ("f1_score: ", "%.4f" %f1_score(y_test, y_pred,average='weighted'))
```

```
f1_score:  0.9904
```

Comparison Chart

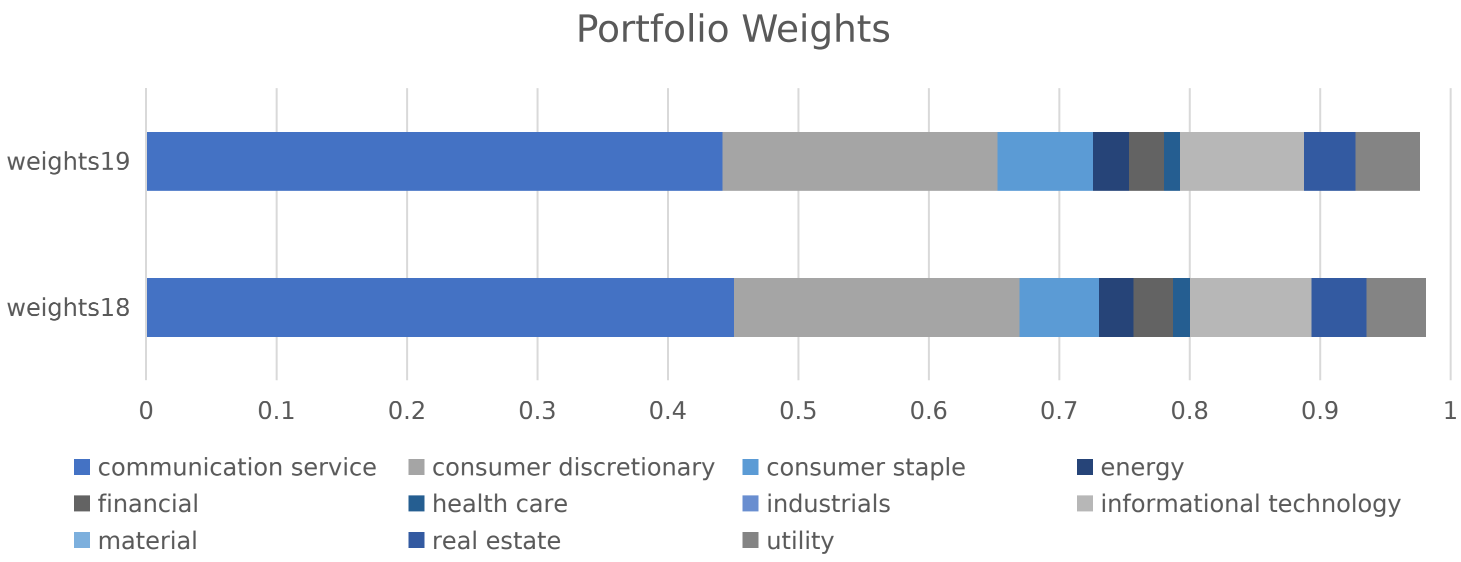


Results

Construction of the Imitated ETF Portfolio

- Weights:

	weights18	weights19
communication service	0.450787	0.442016
consumer discretionary	0.218519	0.210893
consumer staple	0.061279	0.072958
energy	0.026286	0.027818
financial	0.030383	0.026842
health care	0.012835	0.011886
industrials	0	0
informational technology	0.093448	0.095367
material	0	0
real estate	0.04205	0.039378
utility	0.045478	0.049154



ETF Products Selection

- Object: choose the ETF products in each industries that best follows the benchmark index
- Selection criteria: lowest tracking error
- Selection base: any ETF products that is c with equities traded in
- Selection results:

$$TE = \sqrt{\frac{\sum_{i=1}^{n}(R_P - R_B)^2}{N-1}}$$

Where:

TE = Tracking Error

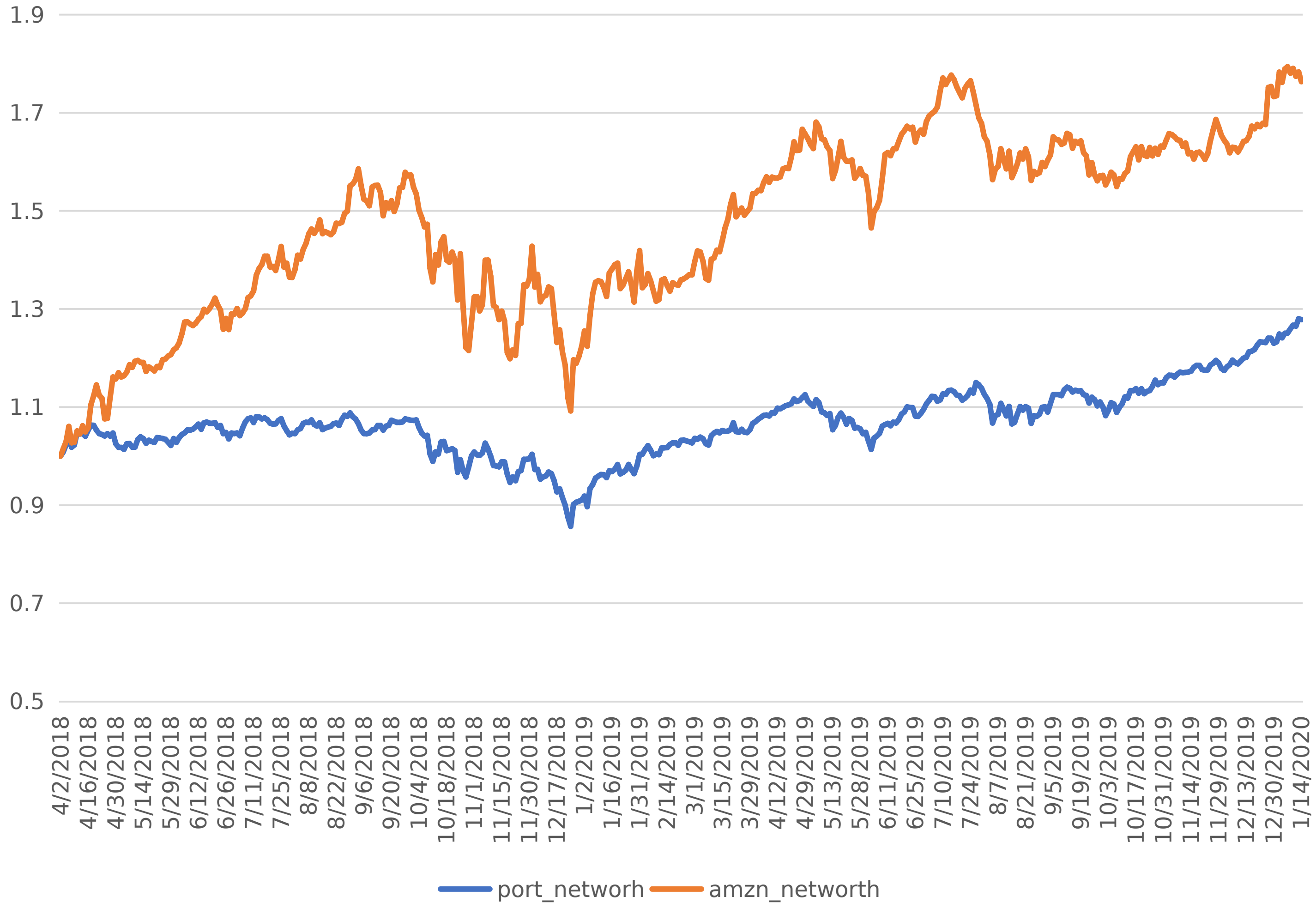
R_P = Return of Manager or Fund

R_B = Return of Benchmark

N = Number of Return Periods

Communication Service	Consumer Discretionary	Consumer Staple	Energy	Financial	Health Care	Industries	Technologies	Material	Real Estate	Utility
VOX	CHIQ	UGE	QCLN	PSP	IXJ	IFLY	IXN	WOOD	REML	TBLU

Back Test

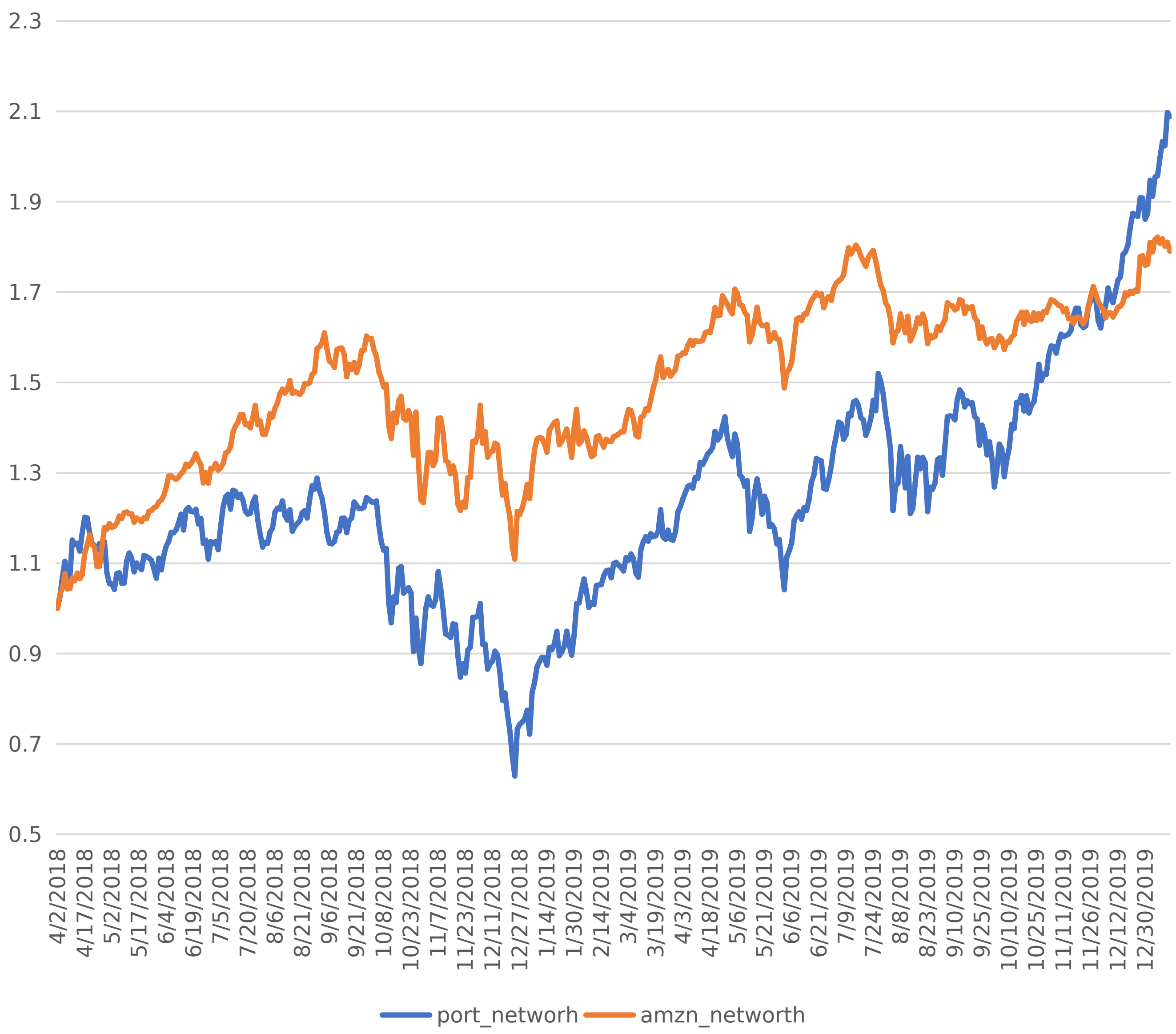


- Object: compare ETF portfolio and AMZN return result
- Time period: April 2nd, 2018 till January 20th, 2020
- Log Return= $\log(\frac{P_{today}}{P_{yesterday}})$
- We can imitate the overall upward and downward trends of AMZN daily

Back Test

- 3-Times leveraged
- With enough leverage, we could imitate AMZN’s returns or even outperform it. However, it could leverage our ETF portfolio’s volatility too.
- $\sigma_{AMZN} = 0.18627$

$\sigma_{Port-ETF} = 0.25067$



Future research

- Determine ETF products selection under different criterion’ s impact on final ETF portfolio return
- Incorporate more textual resource input to check the changes in classification’s result and accuracy
- ……

```
In [17]: (lda_model_tfidf.get_document_topics(doc_term_matrix[3]))

Out[17]: [(0, 0.93815696),
          (1, 0.015476214),
          (2, 0.015419823),
          (3, 0.015527253),
          (4, 0.015419807)]
```

labama" + 0.003*"traditional" + 0.002*"kemper" + 0.002*"igcc" + 0.001*"ppas"
Topic: 25 Word: 0.000*"bros" + 0.000*"warner" + 0.000*"eog" + 0.000*"turner" + 0.000*"abbvies" + 0.000*"televisio
n" + 0.000*"film" + 0.000*"programming" + 0.000*"abbvie" + 0.000*"chevron"
Topic: 26 Word: 0.000*"duke" + 0.000*"streaming" + 0.000*"abbvies" + 0.000*"gm" + 0.000*"dvd" + 0.000*"abbvie" +
0.000*"membership" + 0.000*"vehicle" + 0.000*"content" + 0.000*"video"
Topic: 27 Word: 0.000*"sempra" + 0.000*"verizon" + 0.000*"goldman" + 0.000*"sachs" + 0.000*"wireless" + 0.000*"wp
z" + 0.000*"pipeline" + 0.000*"[illegible]" + 0.000*"sdge" + 0.000*"tape"
Topic: 28 Word: 0.00[illegible]*"sce" + 0.003*"sces" + 0.002*"eme" + 0.002*"edison" + 0.001*"emes" + 0.001*"homer" + 0.000
"cpuc" + 0.000"sempra" + 0.000*"pge" + 0.000*"utility"
Topic: 29 Word: 0.004*"ge" + 0.00[illegible]*"lasalle" + 0.002*"jll" + 0.002*"gecc" + 0.001*"client" + 0.001*"lang" + 0.001
"estate" + 0.001"sustainability" + 0.000*"jones" + 0.000*"real"
Topic: 30 Word: 0.0[illegible]1*"sempra" + 0.006*"eog" + 0.004*"socalgas" + 0.004*"sdge" + 0.004*"rmr" + 0.004*"eogs" + 0.00
2*"reit" + 0.002*"sdges" + 0.001*"song" + 0.001*"energia"
Topic: 31 Word: 0.002*"pp" + 0.002*"ethylene" + 0.002*"amg" + 0.001*"po" + 0.001*"coproducts" + 0.001*"tjx" + 0.00
1*"biosimilars" + 0.001*"opcal" + 0.001*"merchandise" + 0.001*"propylene"
Topic: 32 Word: 0.005*"verizon" + 0.005*"fcc" + 0.005*"wireless" + 0.003*"broadband" + 0.003*"kwe" + 0.003*"video"
+ 0.003*"voice" + 0.003*"cable" + 0.002*"idenix" + 0.002*"spectrum"
Topic: 33 Word: 0.00[illegible]*"fpl" + 0.00[illegible]*"fpls" + 0.00[illegible]*"nee" + 0.00[illegible]*"nees" + 0.00[illegible]*"neer" + 0.001*"neers" + 0.001*"nu
clear" + 0.000*"wind" + 0.000*"mw" + 0.000*"vice"
Topic: 34 Word: 0.004*"halliburton" + 0.003*"fracturing" + 0.003*"anadarko" + 0.003*"hydraulic" + 0.003*"naturalga
s" + 0.002*"anadarkos" + 0.002*"ilim" + 0.002*"vice" + 0.002*"paper" + 0.002*"president"
Topic: 35 Word: 0.004*"railroad" + 0.002*"rail" + 0.001*"pacific" + 0.001*"locomotive" + 0.001*"stb" + 0.001*"ptc"
+ 0.001*"intermodal" + 0.001*"fra" + 0.001*"grain" + 0.001*"coast"
Topic: 36 Word: 0.005*"nike" + 0.002*"footwear" + 0.002*"apparel" + 0.001*"athletic" + 0.001*"converse" + 0.001*"s
port" + 0.000*"sempra" + 0.000*"dominion" + 0.000*"sdge" + 0.000*"lng"
Topic: 37 Word: 0.003*"dcp" + 0.002*"phillips" + 0.002*"duke" + 0.002*"sweeny" + 0.001*"l" + 0.001*"cpchem" + 0.00
1*"refinery" + 0.001*"tx" + 0.001*"borger" + 0.001*"ponca"
Topic: 38 Word: 0.007*"schlumberger" + 0.002*"drilling" + 0.001*"reservoir" + 0.001*"characterization" + 0.001*"we
sterngeco" + 0.001*"schlumbergers" + 0.001*"cameron" + 0.001*"geomarket" + 0.001*"downhole" + 0.001*"oilfield"
Topic: 39 Word: 0.004*"ibm" + 0.003*"ford" + 0.003*"ibms" + 0.003*"vehicle" + 0.002*"hereby" + 0.001*"client" + 0.
001*"pmt" + 0.001*"automotive" + 0.001*"cloud" + 0.001*"cognitive"
Topic: 40 Word: 0.003*"copper" + 0.002*"ptfi" + 0.001*"molybdenum" + 0.001*"leach" + 0.001*"grasberg" + 0.001*"min
ing" + 0.001*"cow" + 0.001*"ore" + 0.001*"mill" + 0.001*"cerro"

Communication
service