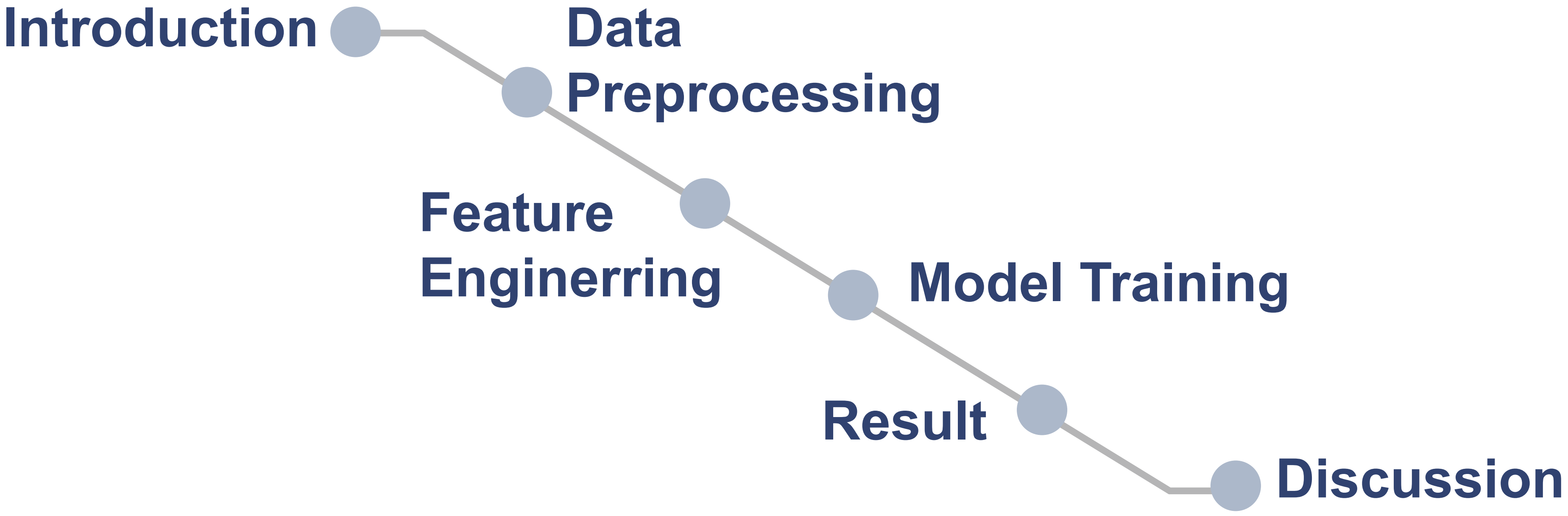
Tweets Influence on Bitcoin Price

Nan Jiang, Chang Ti, Yuxin Mao, Tianhao Wu


Introduction
Data
Preprocessing
Feature
Enginerring
Model Training
Result
Discussion

01

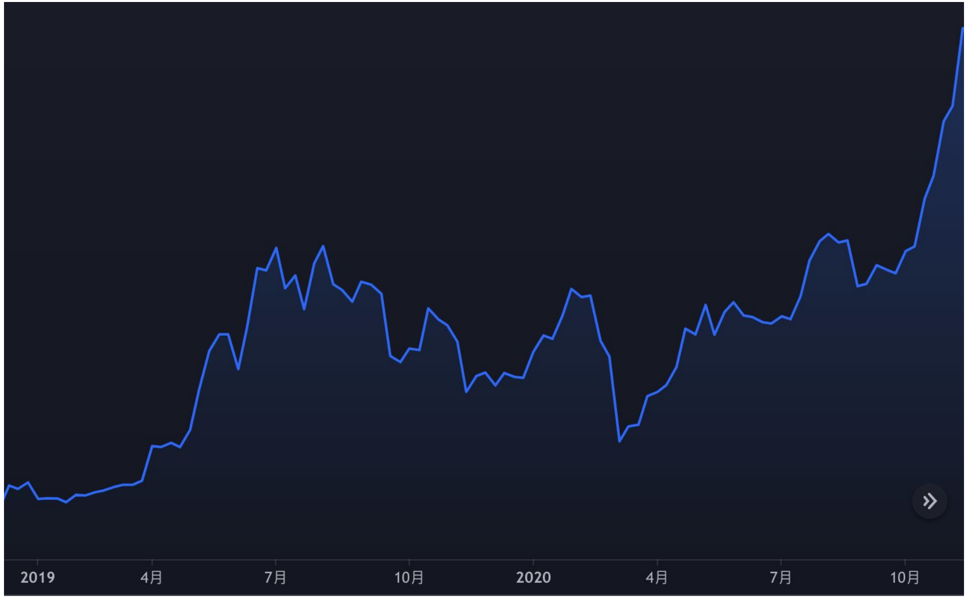
Introduction

01 Introduction – Bitcoin

- Bitcoin is a new cryptocurrency that was created in 2009.

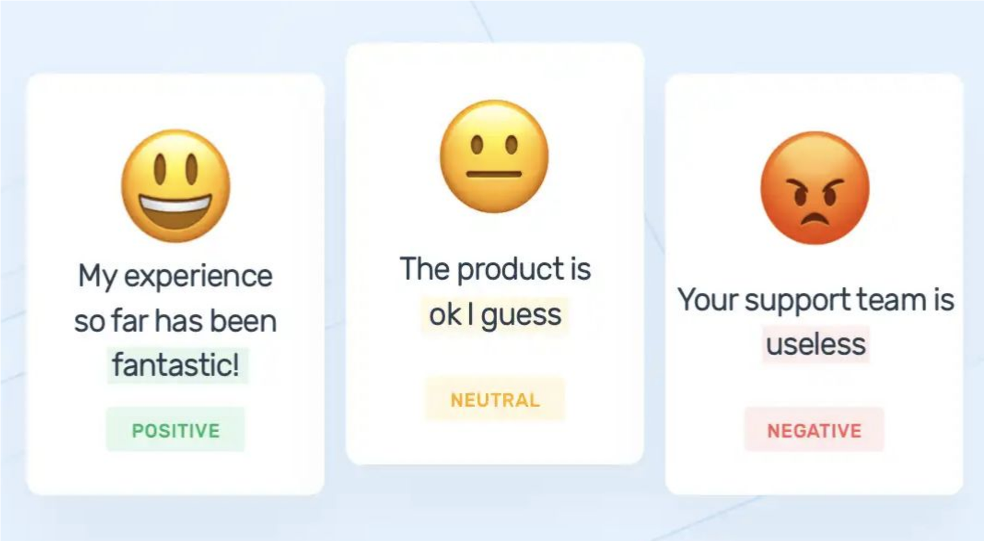
- Unlike investing in traditional currencies, bitcoin is not issued by a central bank or backed by a government.

  Much of hype is about getting rich by trading bitcoins, and that is why the price of bitcoin skyrocketed into thousands in 2017.



01 Introduction – Sentiment Analysis

- **Sentiment analysis** (or **opinion mining**) is a natural language processing technique used to determine whether data is positive, negative or neutral.
- Tweets can be categorized as positive or neutral or negative.



02

Related Work

02 Related Work

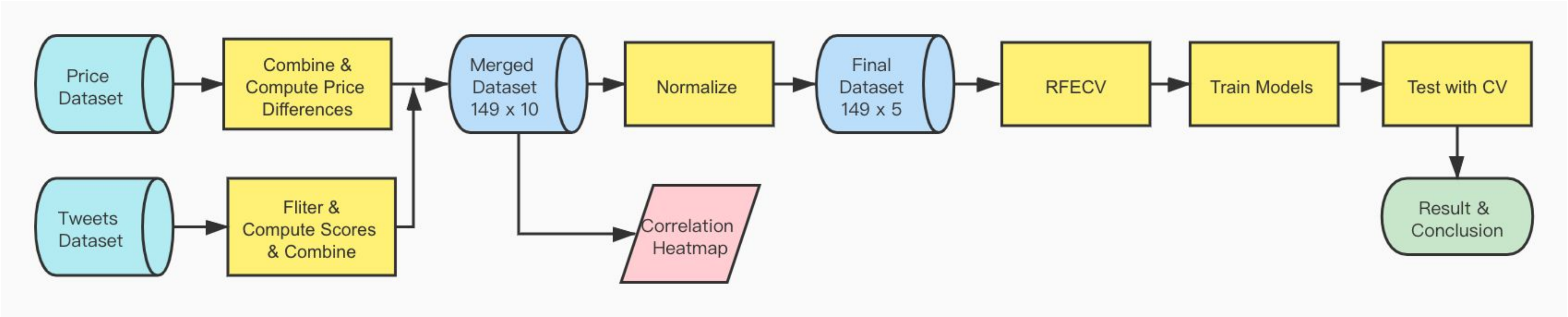
- **PERSPECTIVE: Bitcoin price is predictable by using sentiment analysis of bitcoin-related tweets.**
- METHOD:

1) Recurrent Neural Networks (RNNs) with Long Short Term Memory LSTM)
2) Standard method ARIMA

- **PERSPECTIVE: Elon Musk has an extremely powerful influence on the cryptocurrency market. However, some argues that the silent majority is the ones who dominate the bitcoin price.**
- METHOD:

1) VECM (Vector Error Correction Model) to study the relationship between social media and the monetary value of bitcoin.

Overview


Price Dataset
Combine & Compute Price Differences
Tweets Dataset
Fliter & Compute Scores & Combine
Merged Dataset 149 x 10
Correlation Heatmap
Normalize
Final Dataset 149 x 5
RFECV
Train Models
Test with CV
Result & Conclusion

03

Data Preprocessing

Approach 1

03 Data Preprocessing

01 Missing value

a. Tweets with missing date

b. Tweets with missing number of replies & likes

02 Meaningless text

a. hashtags & URLs

b. non-English words

c. Repeated words e.g. BTC, btc, bitcoin

03 Irrelevant Features

a. User names and ID etc.

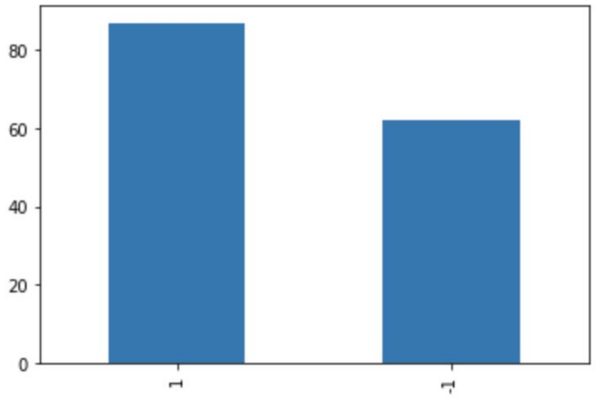
04 Sentiment Analysis

a. vaderSentiment

b. TextBlob

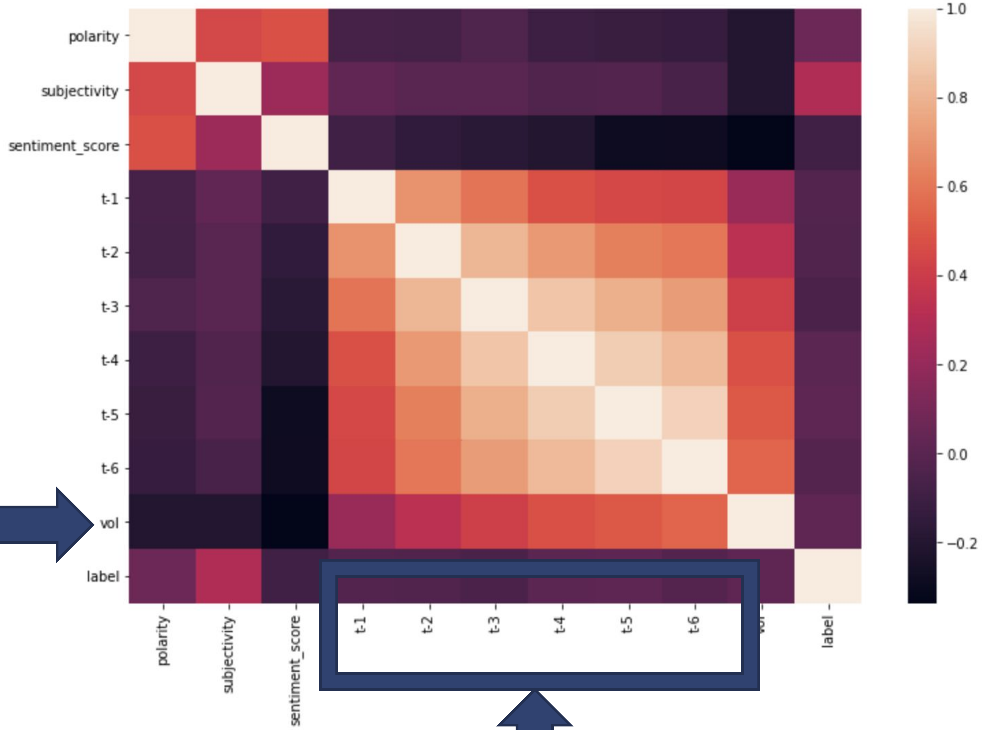
03 Data Preprocessing

	Tweets_raw	Price_raw	Merged
rows	1,992,712	224,640	149
columns	9	8	10



Columns	Names	Descriptions
1	polarity	A float in the range of [-1,1] where 1 means positive and -1 means negative.
2	subjectivity	A float in the range of [0,1] where 1 means subjective and 0 means objective.
3	t-1	price of today - price of one day ago
4	t-2	price of today - price of two days ago
5	t-3	price of today - price of three days ago
6	t-4	price of today - price of four days ago
7	t-5	price of today - price of five days ago
8	t-6	price of today - price of six days ago
9	vol	number of tweets
10	label	1 as rise in price and -1 as fall in price

Observation



Number of Tweets

Price Difference t-1 to t-6

03

Data Preprocessing

Approach 2

03 Data Preprocessing

	Tweets_raw	Price_raw	Merged
rows	1,992,712	224,640	149
columns	9	8	10

Columns	Names	Descriptions
1	sent_score	A float in the range of [-1, 1] where 1 means positive, -1 means negative and 0 means neutral.
2	Vol	Number of tweets per day
3	t-1	price of today minus price of one day before
4	t-2	price of today minus price of two days before
5	t-3	price of today minus price of three days before
6	t-4	price of today minus price of four days before
7	t-5	price of today minus price of five days before
8	t-6	price of today minus price of six days before
9	label	1 as rise in price, and -1 as fall in price from the second day open price minus today's

04

Feature Engineering

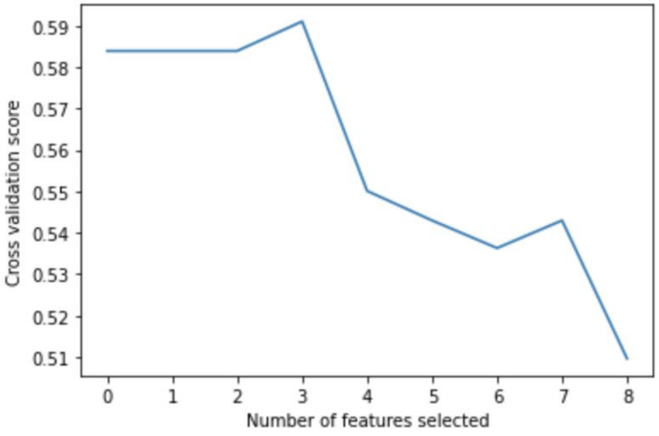
Approach 1

04 Feature Engineering

01

RFECV

- Recursive Feature Elimination with Cross Validation (RFECV)
- Eliminate 5 features that have less influence on bitcoin price
- Observation: t-1, t-2, t-4 and t-6 are eliminated while t-3 and t-5 are maintained



Columns	Names	Descriptions
1	polarity	A float in the range of [-1,1] where 1 means positive and -1 means negative.
2	subjectivity	A float in the range of [0,1] where 1 means subjective and 0 means objective.
3	t-3	price of today - price of three days ago
4	t-5	price of today - price of five days ago
5	label	1 as rise in price and -1 as fall in price

04 Feature Engineering

02 Normalize

- Significantly different range
- Not normal ditributed
- MinMaxScaler in Scikit-learn

	Feature	Range
0	polarity	0.274779
1	subjectivity	0.320560
2	sentiment_score	0.559362
3	t-1	1449.380000
4	t-2	1918.120000
5	t-3	2380.610000
6	t-4	2409.450000
7	t-5	2447.720000
8	t-6	2571.060000
9	vol	100194.000000

	Feature	Range
0	polarity	1.0
1	subjectivity	1.0
2	sentiment_score	1.0
3	t-1	1.0
4	t-2	1.0
5	t-3	1.0
6	t-4	1.0
7	t-5	1.0
8	t-6	1.0
9	vol	1.0

04

Feature Engineering

Approach 2

04 Feature Engineering

01 Simple Return

- Calculate Simple Regression for t-x(x=1,2,3,4,5,6)

$$Simple\ Return = \left\{\frac{Present\ NAV - Starting\ NAV}{Starting\ NAV}\right)$$

DateIndex	date	sent_score	Vol	t-1	t-2	t-3	t-4	t-5	t-6	label
2019-01-01	2019-01-01	0.309938	21	-0.0351741	-0.0099797	-0.0506145	0.0288909	-0.0305867	-0.022663	1
2019-01-02	2019-01-02	0.0172364	11	0.0355546	-0.000870142	0.0252201	-0.0168595	0.0654727	0.00388044	1
2019-01-03	2019-01-03	0.35013	23	0.0174755	0.0536515	0.0165902	0.0431363	0.000321403	0.0840924	-1
2019-01-04	2019-01-04	0.172615	20	-0.0266139	-0.00960344	0.0256097	-0.0104652	0.0153744	-0.026301	1
2019-01-05	2019-01-05	0.170825	28	0.00912617	-0.0177306	-0.000564908	0.0349696	-0.00143456	0.0246409	-1

04 Feature Engineering

01

Z-score

- Get Z score from the prior fifteen days
- Round up to four digits
- Delete first fifteen days

$$Z = \frac{x - \mu}{\sigma}$$

DateIndex	sent_score	Vol	t-1	t-2	t-3	t-4	t-5	t-6	label
2019-01-16	0.628	-1.1589	1.2482	0.4245	0.336	0.3082	-1.1823	-1.1725	1
2019-01-17	-0.9264	-2.4674	3.7187	3.477	3.2255	2.9888	-3.6626	-3.656	1
2019-01-18	1.0302	-2.1363	3.5366	3.7086	3.7132	3.7085	-3.5503	-3.5495	-1
2019-01-19	-0.6096	-1.7263	2.3933	2.6315	2.7395	2.8368	-2.4286	-2.4318	1
2019-01-20	0.5785	-1.4375	1.2733	1.4822	1.6013	1.7101	-1.3046	-1.3097	-1

05

Training & Result

Approach 1

05 Training & Result

		Selected 4 Features				All Features			
	model	test accuracy	test precision	test recall	test f1	test accuracy	test precision	test recall	test f1
0	DummyClassifier	0.543448	0.624079	0.575163	0.594868	0.475862	0.543887	0.562745	0.550760
1	LogisticRegression	0.590805	0.592816	0.964706	0.733344	0.442759	0.550357	0.378431	0.436103
2	SVC	0.583908	0.583908	1.000000	0.737188	0.577011	0.580952	0.988235	0.731584
3	DecisionTreeClassifier	0.536322	0.611313	0.622222	0.610414	0.583678	0.646056	0.646405	0.635386
4	RandomForestClassifier	0.596552	0.638411	0.783007	0.696419	0.509195	0.573261	0.649673	0.584453
5	KNeighborsClassifier	0.536782	0.594828	0.669935	0.623849	0.523678	0.589014	0.620915	0.602515
6	GaussianNB	0.623678	0.641644	0.863399	0.729405	0.529425	0.558670	0.765359	0.591289
7	VotingClassifier	0.583448	0.603439	0.874510	0.711122	0.550575	0.589494	0.783660	0.665021

05 Training & Result

	model	test accuracy	test precision	test recall	test f1
0	DummyClassifier	0.543448	0.624079	0.575163	0.594868
1	LogisticRegression	0.590805	0.592816	0.964706	0.733344
2	SVC	0.583908	0.583908	1.000000	0.737188
3	DecisionTreeClassifier	0.536322	0.611313	0.622222	0.610414
4	RandomForestClassifier	0.596552	0.638411	0.783007	0.696419
5	KNeighborsClassifier	0.536782	0.594828	0.669935	0.623849
6	GaussianNB	0.623678	0.641644	0.863399	0.729405
7	VotingClassifier	0.583448	0.603439	0.874510	0.711122

05

Training & Result

Approach 2

05 Training & Result

Index	model	test accuracy	test precision	test recall	test f1
0	DummyClassifier	0.597151	0.597151	1	0.74773
1	SVC	0.551852	0.57769	0.9125	0.706161
2	GaussianNB	0.47094	0.544588	0.65	0.590975
3	RandomForestClassifier	0.469801	0.552164	0.6	0.573776
4	DecisionTreeClassifier	0.476638	0.572821	0.5125	0.533482
5	LogisticRegression	0.560114	0.58786	0.8625	0.697868
6	KNeighborsClassifier	0.439886	0.524286	0.6625	0.585285
7	Ensemble	0.485185	0.545706	0.7875	0.644099

05

Training & Result

Index	model	test accuracy	test precision	test recall	test f1
0	SVC	0.551852	0.57769	0.9125	0.706161
1	GaussianNB	0.47094	0.544588	0.65	0.590975
2	RandomForestClassifier	0.469801	0.552164	0.6	0.573776
3	LogisticRegression	0.560114	0.58786	0.8625	0.697868
4	KNeighborsClassifier	0.439886	0.524286	0.6625	0.585285
5	Ensemble	0.469801	0.538784	0.7375	0.621506

05 Training & Result

Index	model	test accuracy	test precision	test recall	test f1
0	SVC	0.551852	0.57769	0.9125	0.706161
1	GaussianNB	0.47094	0.544588	0.65	0.590975
2	LogisticRegression	0.560114	0.58786	0.8625	0.697868
3	Ensemble	0.545014	0.581409	0.8375	0.68474

06

Discussion

06 Discussion

Limitations

- **Time**

a) Didn't collect our own dataset ⇨ The data we used was collected in 2019
b) Didn't translate tweets written by non-English languge

- **Hardware**

a) Limited capability for processing large datasets ⇨ Extract 149 days from the raw dataset

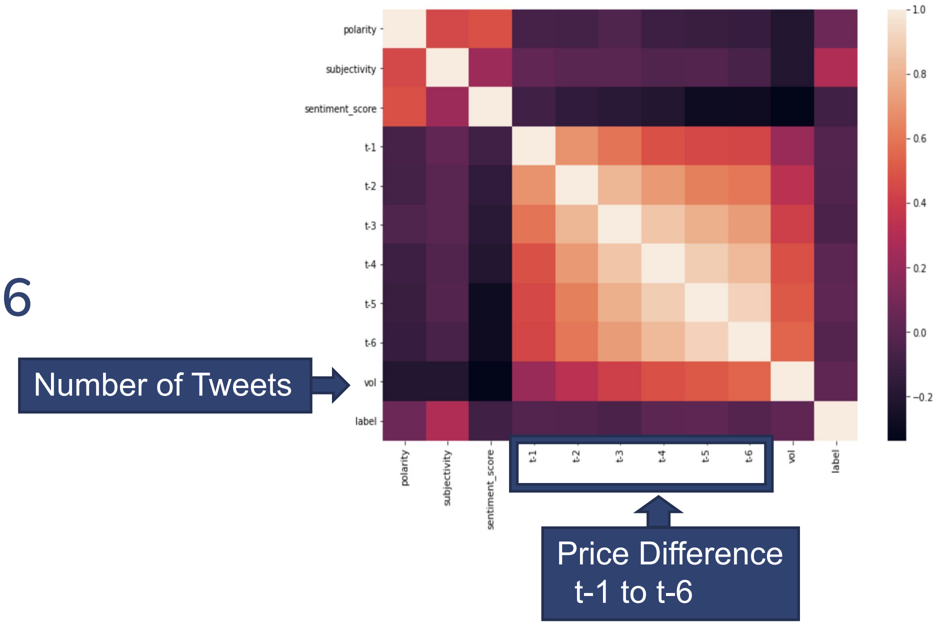
- **Future Improvement**

a) A database containing latest data will be used
b) More tweets will be taken into account

06 Discussion

- Lighter color = higher correlation

  which increase from t-1 to t- 6
- t-6 and vol are highly correlated
- The number of tweets is easily affected by price fluctuations six days ago



06 Discussion

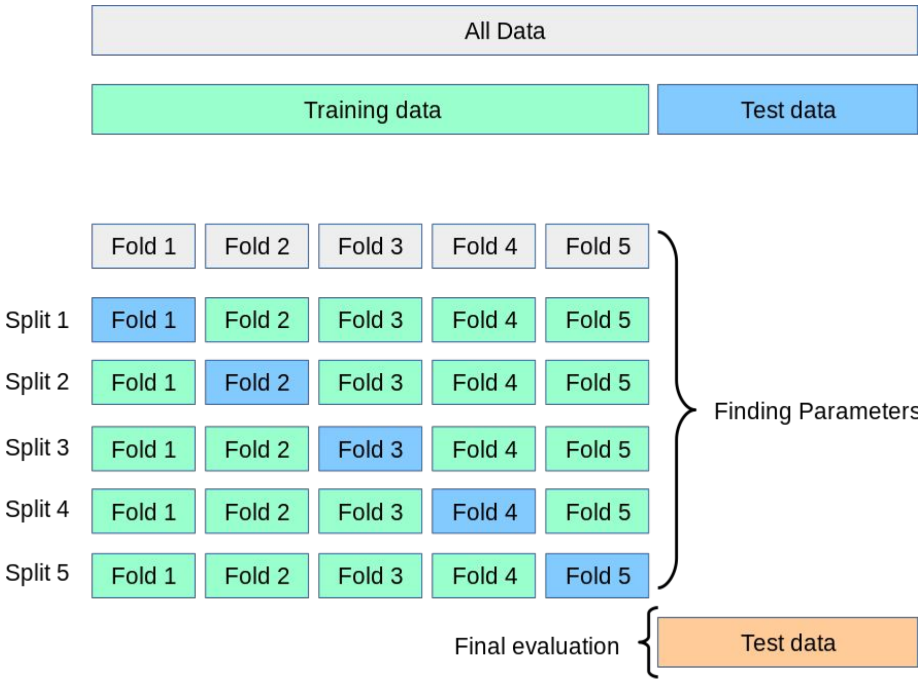
Cross Validation (CV)

1st time: the first data is used as the test set, the other four are used as the training set

2nd time: the second data is used as the test set, and the other four are used as the training set;

… (and so on)

→ Finally get the average evaluation



Why is this necessary?

When random state=42 for data splitting, the accuracy of all models are particularly high

Since the dataset contains only 149 days, **bias** could exist if hold-out is used

07

Thank you for watching