

## Abstract

Recently virtual currency market is drawing more and more attention. Comparing to stock market, price change in the virtual currency market is more drastic. Meanwhile, it is difficult to predict price in short term in the virtual currency market. Generally, only basing on market indices, which is a common practice in stock and foreign exchange (FX) markets, in many cases short-term price prediction takes a combination of multiple indices into account. However, beside markets, there are other types of factors usually influence price changes. Social media is one of them.

According to previous researches, price change of Bitcoin (BTC), a representative of virtual currencies, is known to be highly related to text contents which are posted on social media like Twitter, Reddit, and so on. In a previous research, a price prediction methodology with a machine learning algorithm was proposed. The proposal took results of sentiment analysis score from relevant tweets and virtual currency market data as features for the machine learning algorithm. The experiment results indicated that comparing to using market information only, by taking Emotion Induction results into account, some of models yielded to higher accuracies. However, it only extracted features from data of that single day. It didn't consider market direction changes in time sequence. Thus, the prediction accuracy is not good enough. Moreover, all tweets are processed with the same importance, thus it didn't reflect real sentiment.

My proposed research takes tweets from Jan. 1, 2020 to Dec. 31, 2020 and market information as research targets for virtual currencies BTC, ETH, XRP and LTC. Emotion Induction scores of each tweet are calculated with VADER. This research proposes to introduce weights of each tweet for sentiment score, Moving Average (MA) curves and their slopes for the predictions. Weights of tweets are calculated from number of followers, number of replies, number of retweets, number of Likes, overall number of tweets on that single day, etc.

Number of followers, number of replies, number of retweets and number of Likes are used to improve accuracy of the sentiment score. This proposed research utilizes 7 types of weights: sentiment score, MA curves of market information and their slopes reflect features in time sequence. In this proposed research, Simple Moving Average (SMA) curves and Moving Average Convergence/Divergence (MACD) curves with 3-day, 5-day, 7-day time spans respectively, as well as slopes of individual MA curves, are used.

A combination (weights of sentiment score and the MA type) with the best results is extracted with machine learning models MLP, SVM and RF. An analysis is done on the prediction results with a model using this combination to show case effectiveness of this proposed research.

Finally, prediction results of this proposed research are compared to previous researches. Since number of data and its duration used in this research differ to those in previous researches, the comparison can not be a quantitative one, but is one to compare prediction accuracies before and after Sentiment Induction in each research. This proposed research shows effectiveness comparing to previous researches in more scenarios.

## 概要

注目度が高まっている仮想通貨市場は、株式市場より価格の変化が激しく、短期の価格予測は課題である。一般に、短期的な値動きの予測は、株式や FX と同様に、市場指標のみに基づいて、各指標の組み合わせで判断されることが多い。しかし、市場の外には、常に値動きを左右する様々な要因が存在する。ソーシャルメディアもその要因の一つである。

先行研究により、仮想通貨の代表のビットコイン (BTC) の値動きは、Twitter, Reddit などのソーシャルメディア上に投稿されたテキスト内容と関連度が高いことが分かっている。先行研究では関連 Tweet の感情推定結果や仮想通貨の市場データを特徴量として機械学習を用い価格を予測する手法が提案されている。その実験結果によると、感情推定とマーケット情報を使用する特徴量は、マーケット情報のみを使用する特徴量と比べて、一部のモデルではより良い精度を得られた。

しかし、当日のデータのみから特徴量を抽出しており、市場動向が時系列で変化することを考慮していないため、予測精度はまだ十分ではない。また、全ての Tweet を同じ影響度で扱っているため、真実の感情推定結果は反映されていない。

本研究では、仮想通貨 BTC, ETH, XRP, 及び LTC に対して、2020/1/1~2020/12/31 の Tweet とマーケット情報を取得して研究対象とする。VADER を用いて Tweet ごとの感情推定スコアを算出する。本研究では、感情推定スコアに対して、Follower 数, Reply 数, Retweet 数, Like 数, 及び当日の Tweet 取得総数などを考慮することで各 Tweet の重みを算出する方法や移動平均線及び傾きの導入を提案する。

Follower 数, Reply 数, Retweet 数, Like 数を考慮することで感情推定精度を向上させる。本研究の実験では重みを 7 種を使用した。感情推定結果とマーケット情報の移動平均線及び傾きで、時系列の特性を特徴量に反映させる。本研究の実験では感情推定スコア及びマーケット情報に対して、それぞれの 3 日, 5 日, 7 日の単純移動平均線, 指数平滑移動平均線及び移動平均線の傾きを使用した。この 2 点の組み合わせた特徴量を抽出し、機械学習モデル MLP, SVM, RF で実験し、一番良い組み合わせ (感情推定の重みと移動平均線の種類) を抽出した。そして、この組み合わせを使ったモデルの予測結果を分析し、本研究で提案した方法の有効性が明らかになった。

最後に、本研究での予測結果と先行研究の予測結果を比較した。本研究のデータ数及び期間は、先行研究とは違うため、定量的に比較できず、各感情推定導入前後の予測精度の差異などを比較した。本研究で提案した方法は、先行研究より多くの場合で有効であることが明らかになった。

# 目次

<b>第1章</b>	<b>序論</b>	<b>1</b>
1.1	研究背景 . . . . .	1
1.2	研究目的 . . . . .	2
1.3	論文構成 . . . . .	3
<b>第2章</b>	<b>既存研究</b>	<b>4</b>
2.1	先行研究 . . . . .	4
2.2	まとめ . . . . .	5
<b>第3章</b>	<b>関連知識</b>	<b>6</b>
3.1	本研究の目標 . . . . .	6
3.2	機械学習 . . . . .	6
3.2.1	多層パーセプトロン (MLP) . . . . .	6
3.2.2	サポートベクターマシン (SVM) . . . . .	8
3.2.3	ランダムフォレスト (RF) . . . . .	9
3.2.4	scikit-learn . . . . .	10
3.3	感情推定ツール VADER . . . . .	11
3.4	移動平均線 . . . . .	11
3.5	Tweet データ収集について . . . . .	12
<b>第4章</b>	<b>提案手法と実装</b>	<b>14</b>
4.1	データ収集及びメタデータの前処理 . . . . .	15
4.2	感情推定及びマーケットの特徴データ . . . . .	16
4.2.1	感情推定スコアを加工 . . . . .	16
4.2.2	マーケットデータ加工 . . . . .	22
4.3	特徴量の抽出 . . . . .	22
4.4	モデル実装と訓練 . . . . .	25
<b>第5章</b>	<b>実験評価</b>	<b>26</b>
5.1	実験準備 . . . . .	27
5.2	評価方法 . . . . .	28
5.3	評価結果 . . . . .	29
5.4	まとめ . . . . .	35
<b>第6章</b>	<b>おわりに</b>	<b>36</b>
6.1	本研究の成果 . . . . .	36
6.2	今後の課題 . . . . .	36

# 目 次

3.1	隠れ層 1 つの多層パーセプトロン . . . . .	7
3.2	sigmoid 関数 . . . . .	7
3.3	tanh 関数 . . . . .	8
3.4	relu 関数 . . . . .	8
3.5	サポートベクターマシン . . . . .	9
3.6	ランダムフォレスト . . . . .	10

# 表 目 次

4.1	tweet_xxx テーブルのカラム . . . . .	15
4.2	Follower 数, Reply 数, Retweet 数, Like 数の重み . . . . .	17
4.3	Tweet の Reply 数, Retweet 数, Like 数の統計 . . . . .	18
4.4	Tweet 感情推定のための日次の特徴データの組み合わせ . . . . .	19
4.5	Tweet 感情推定日次特徴組合せの展開 . . . . .	20
4.6	マーケット特徴組合せの展開 . . . . .	22
5.1	収集した Tweet の統計情報（仮想通貨別） . . . . .	27
5.2	仮想通貨の価格変動の統計情報（仮想通貨別） . . . . .	27
5.3	Follower 数, Reply 数, Retweet 数, Like 数の重み . . . . .	29
5.4	BTC の評価結果 . . . . .	30
5.5	ETH の評価結果 . . . . .	31
5.6	XRP の評価結果 . . . . .	32
5.7	LTC の評価結果 . . . . .	33
5.8	先行研究と比較 (BTC) . . . . .	34
6.1	VADER で感情推定の失敗例 . . . . .	36

# 第1章 序論

本章では、研究の背景、研究の目的、論文の構成について記す。

## 1.1 研究背景

注目度がますます高まっている仮想通貨 [1] 市場は、株式市場より価格の変化が激しく、短期、長期の価格予測は課題である。基本的に S2F[2], S2FX[2] などのモデルを用いることで、長期の値動きを精度良く予測できる。一方で短期の価格は影響要素が多く予測が難しい。一般に、短期的な値動きの予測は、株式や FX と同様に、市場指標のみに基づいて、各指標の組み合わせで判断されることが多い。しかし、市場の外には、常に値動きを左右する様々な要因が存在する。

また、ソーシャルメディアは仮想通貨市場に大きな影響を及ぼす。近年、ソーシャルメディアの普及により、誰もが公に自分の観点を表現し、感情を共有できるようになり、個人の影響力が拡大される。しかし、ソーシャルメディアが多くの方の生活に入り込むにつれ、私たちは個人の判断に影響を与えるほど、その影響を受けることも多くなっている。ソーシャルメディアが仮想通貨に与える影響は明らかである。例えば、世界で最も影響力のある人物の一人であるイーロン・マスクは、2021 年 2 月 7 日にテスラが 15 億ドル以上のビットコインを購入したと発表し、同時に自分の Twitter の顔写真をビットコインのマークに変更すると、その後の 1 日でビットコインの価格は米ドルに対して最大で 17 % 上昇した [3]。そして、仮想通貨に参入感が深いマスク氏は、仮想通貨の 1 種類 DOGE に関する発言をするたびに、市場にボラティリティを与えている。

このようにソーシャルメディアの影響力は非常に大きいため、ソーシャルメディアが様々な側面に与える影響についての研究が進んでいる。これには、仮想通貨の価格への影響も含まれる。これまでの研究 [4] では、仮想通貨の代表のビットコインの値動きは、Twitter, Reddit などのソーシャルメディア上に投稿されたテキスト内容と関連度が高いことが分かっている。関連 Tweet の感情推定結果や仮想通貨の市場データを特徴量として機械学習を用い価格を予測する手法 [5] が提案されている。[5] の実験結果によると、機械学習と感情推定を利用したモデルにより、仮想通貨市場の LTC 価格の予測精度は、最高で Recall 0.66, Precision 0.80 であった。しかしまだ時系列の特性は考慮されておらず、Tweet の感情推定精度を向上することで、さらなる精度向上が見込める。

## 1.2 研究目的

本研究では、ソーシャルメディアの感情推定精度向上や、一定期間内の感情や市場動向の変化から適切な特徴量を抽出することを研究要素とし、過去の研究 [5] 成果に改良を加えて予測精度を向上させることに焦点を当てる。過去の研究 [5] では、Tweet のデータの感情分析結果とマーケットデータから特徴量を抽出し、多層パーセプトロン (MLP)、サポートベクターマシン (SVM)、ランダムフォレスト (RF) の精度を比較したが、ソーシャルメディアの感情変化やマーケットデータは当日のデータのみを使い、時系列であることを考慮していない。また、取得した Tweet の感情推定において、各 Tweet は同様に扱われて計算されている。しかし、Follower 数、Reply 数、Retweet 数、Like 数は Tweet ごとに大きく異なっている。これらの数値は Tweet の影響度を表しているため、Tweet ごとの影響度の重みを感情推定モデルに反映することにより、感情推定精度を向上させることが期待できる。本研究では、感情推定の際に Follower 数、Reply 数、Retweet 数、Like 数、及び当日の Tweet 取得総数などにより各 Tweet の重みを算出する方法を提案する。また、Tweet の感情、マーケットデータは時々刻々変化する時系列データであるため、過去の一定期間のデータは未来の変化に影響する。本研究では、感情推定結果、マーケットデータの移動平均と一定期間ごとのデータ変化の速度から特徴量を抽出する方法を提案する。本研究では Tweet ごとの重みの算出、ソーシャルメディア感情変化の時系列の特性を考慮する点が、従来研究と異なる新規性である。

## 1.3 論文構成

本報告書では、以下の構成にてソーシャルメディア感情推定と機械学習を用いた仮想通貨の価格変動予測の研究成果を記述する。

- 第 2 章 既存研究の感情推定方法，利用する技術，特徴量の抽出方法について述べる。
- 第 3 章 本研究の前提知識として機械学習の手法，感情推定の方法，移動平均線とその傾きの計算について述べる。
- 第 4 章 Follower 数，Reply 数，Retweet 数，Like 数を考慮して特徴量を抽出する方法について述べた後，データ収集から特徴量を抽出して実験までの実装について述べる。
- 第 5 章 抽出した特徴量を用いて各機械学習の方法で実験し，実験結果と考察について述べる。
- 第 6 章 本報告書のまとめと研究に対する課題について述べる。



## 第2章 既存研究

本章では、Twitter の感情推定と機械学習による仮想通貨の価格変動の予測に関する先行研究について述べる。

### 2.1 先行研究

Twitter の感情推定と機械学習による仮想通貨の価格変動の予測に関して、先行研究 [5] では、BTC, ETH, XRP, LTC 4 種類の主流仮想通貨を研究対象とし、Twitter から関連タグの 3 ヶ月分の Tweet データを収集し、語彙とルールに基づく感情分析ツール VADER[6] を用いて感情推定を行った。そして、4 種類仮想通貨のマーケットデータをその 3 ヶ月分を取得した。マーケットのみの特徴量、感情推定結果のみを用いた特徴量、マーケットと感情推定結果を組み合わせた特徴量を 3 種類を用意し、多層パーセプトロン (MLP)、サポートベクターマシン (SVM)、ランダムフォレスト (RF) を用いて実験し、その結果を分析し感情推定の特徴量の有用性について述べている。neu, neg, pos, compound は VADER で定義された感情を表す 4 つの指標である。先行研究ではこれを基いて日次の感情指標の 1 Tweet 当たりの平均値を算出する。マーケットの指標は日時の open (初値), close (終値), high (高値), low (安値), volumeto (ボリューム, 取引のドル計の金額) を使用した。

$$v_{compound} = \frac{\sum_{i=1}^n t_i^{compound}}{n} \quad (2.1)$$

$$v_{neu} = \frac{\sum_{i=1}^n t_i^{neu}}{n} \quad (2.2)$$

$$v_{neg} = \frac{\sum_{i=1}^n t_i^{neg}}{n} \quad (2.3)$$

$$v_{pos} = \frac{\sum_{i=1}^n t_i^{pos}}{n} \quad (2.4)$$

$$v_{pol} = \sqrt{v_{pos}v_{neg}} \quad (2.5)$$

そして、Tweet 感情推定指標のみを使用する場合は特徴量  $V_{1(t)}$ , マーケット指標のみを使う場合は特徴量  $V_{2(t)}$ , 両方を組み合わせて使う場合は特徴量  $V_{3(t)}$  を使用した。

$$V_{1(t)} = \begin{bmatrix} neu \\ neg \\ pos \\ compound \\ pol \end{bmatrix} \quad (2.6)$$

$$V_{2(t)} = \begin{bmatrix} open \\ close \\ high \\ low \\ volumeto \end{bmatrix} \quad (2.7)$$

$$V_{3(t)} = \begin{bmatrix} neu \\ neg \\ pos \\ compound \\ pol \\ open \\ close \\ high \\ low \\ volumeto \end{bmatrix} \quad (2.8)$$

この3つの特徴量を使って MLP, SVM, RF のモデルをそれぞれ訓練して実験を行った. 様々なハイパーパラメータの組み合わせてモデルを訓練し, 各モデルの一番優れた結果を評価する.

先行研究では以下の結論を述べた [5].

- 各仮想通貨の価格予測実験において, 少なくとも1つのモデルはランダムより Precision が高い.
- ランダムより Precision が高いモデルにおいて, Tweet 感情推定指標のみの特徴量  $V_{1(t)}$  を使うモデルは, XRP の SVM モデル, LTC の場合の MLP モデル, 2つがある.
- Tweet 感情推定指標のみの特徴量  $V_{1(t)}$  を使うモデルは, Majority モデルより正解率が高いモデルは, XRP, LTC の MLP と SVM モデル4つである. 4つとも正解率はマーケット指標のみの特徴量  $V_{2(t)}$  を使うモデルより低い.
- Tweet 感情推定指標とマーケット指標両方を組み合わせ特徴量  $V_{3(t)}$  を使うモデルは, マーケット指標のみの特徴量  $V_{2(t)}$  を使うモデルより Precision が高いのは, BTC と XRP の MLP モデル, LTC の SVM と RF, 4つがある.
- 組み合わせ特徴量  $V_{3(t)}$  に関して, 最も良いモデルは LTC の SVM である. Accuracy 0.66, Precision 0.80, Recall 0.67, F1-score 0.62 である.

## 2.2 まとめ

本章では, 感情推定指標を使った仮想通貨の価格変動予測について先行研究 [5] の手法と結果示した. 先行研究 [5] では, 感情推定指標だけをそのまま使うと, 良い結果を得られないことが示された, 感情推定指標とマーケット指標を組み合わせても, マーケット指標のみより良い結果を得られたモデルも少ないことが明らかになった.

## 第3章 関連知識

本章では、研究を進めるにあたって必要となる関連知識をまとめた。まず 3.1 項では本研究の目標について説明し、実験評価時に提案の有効性の判定方法も説明する。3.2 項では本研究に使う機械学習の手法の概要について説明し、本研究の実装時に使うライブラリについても説明する。3.3 項で本研究に使う感情推定ツール「VADER」の概要について説明する。3.4 項では株や FX のトレード用の重要指標である移動平均線の概要及び計算方法について説明する。3.5 項では本研究のデータ収集用のツールと API について説明する。

### 3.1 本研究の目標

実際のトレーディングでは、良い策は高い正確率（ここでは Precision）を達成することであり、予測の正確率を高めることによって、利益を上げるチャンスを得ることができる。先行研究と比べて、本研究では、Twitter の感情推定を処理することで、Precision を向上させる特徴量を見出すことを期待している。

この点を考慮し、先行研究 [5]2.1 では、感情推定を導入後の特徴量  $V_{3(t)}$  を用いて得た実験結果の Precision が、感情推定導入前の特徴量  $V_{2(t)}$  の Precision よりも大きい場合にのみ有効とする。先行研究でも Precision を最重要な指標であることを述べた（Precision は、ストラテジーがどのタイプの成行注文を作成するか決定する予測で何回正しかったかを示すため、最も重要な指標と考えられている [16]）。2.1 で述べた通り、先行研究では有効のパターンは BTC と XRP の MLP モデル、LTC の SVM と RF、4 つがある。本研究でも感情推定を導入後の特徴量は感情推定導入前の特徴量より高い Precision を得ることが有効とする。Precision が同じの場合 Recall が高い方が有効とする。

上記の設定を踏まえ、先行研究では、MLP、SVM、RF モデルをそれぞれ用いて 4 つの仮想通貨について実験を行い、得られた 12 件の結果のうち、感情推定値の導入が有効だったのは 4 件のみである。具体的に有効だったのは、BTC では MLP、ETH では MLP、LTC では SVM と RF であった。本研究の目的は、感情推定値の処理方法や感情推定値に関連する新しい特徴を導入し、より多くのケースに対して有効な Precision を得ることである。さらには先行研究と同じケース（同じ仮想通貨と特徴量）でもより高い Precision を得ることである。

### 3.2 機械学習

#### 3.2.1 多層パーセプトロン (MLP)

多層パーセプトロン (MLP) とは、単純パーセプトロンに隠れ層（中間層）が加わった、複数のニューロン層（3 層以上）で構成されるニューラルネットワークである。各層は下の層と上の層に全結合である。図 3.1 に隠れ層 1 つの多層パーセプトロンを示す。

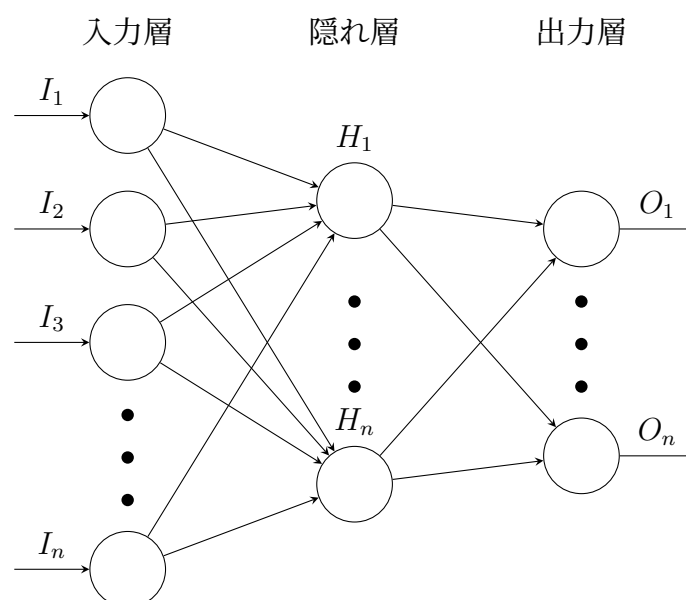


図 3.1: 隠れ層 1 つの多層パーセプトロン

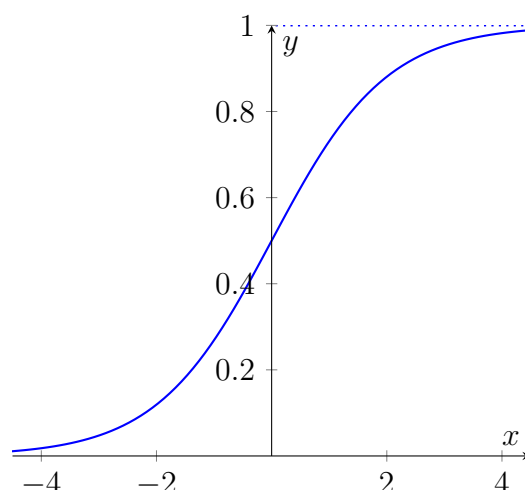


図 3.2: sigmoid 関数

出力層の出力は，最後の隠れ層のすべてのニューロンの出力にその重みを乗じ，出力層の出力の合計にバイアスを加えることによって得られる．最終的な出力は，出力層が受け取った入力を活性化関数を通してマッピングすることで得られる．

入力をどのように活性化させるかを決定することは活性化関数の役割である．多層パーセプトロンの活性化関数では，「sigmoid 関数，tanh 関数，relu 関数」が使われる．

sigmoid 関数は，図 3.2 のような，任意の入力値を 0.0 から 1.0 の範囲の数値に変換して出力する関数である．標準 sigmoid 関数の定義は下記の数式である．

$$f(x) = \frac{1}{1 + e^{-ax}} (a > 0) \quad (3.1)$$

また，logistic 関数は sigmoid 関数のより汎用的な関数である．

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}} \quad (3.2)$$

tanh 関数は，図 3.3 のような，任意の入力値を -1.0 から 1.0 の範囲の数値に変換して出力する関数である．tanh 関数の定義は下記の数式である．tanh 関数は sigmoid 関数と比べて，出

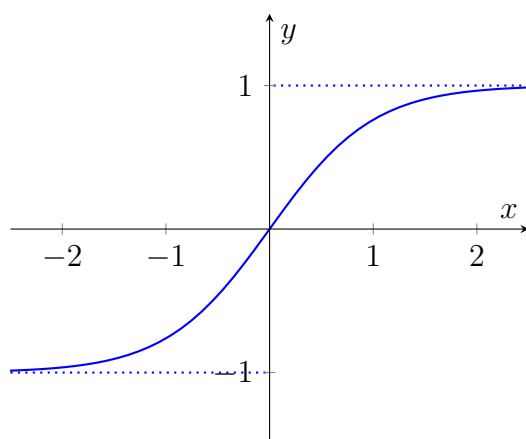


図 3.3: tanh 関数

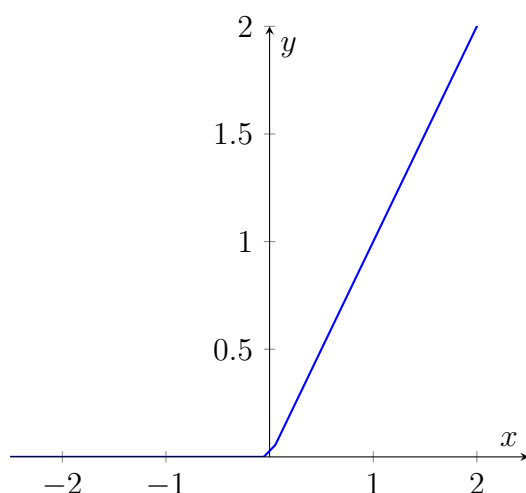


図 3.4: relu 関数

力値の幅は2倍である.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.3)$$

relu 関数とは, 図 3.4 のような, 入力値がゼロ以下の場合は常に出力値が0になり, 入力値が0以上の場合は出力値が入力値と同じになる関数である.

$$f(x) = \begin{cases} 0 & (x \leq 0) \\ x & (x > 0) \end{cases} \quad (3.4)$$

### 3.2.2 サポートベクターマシン (SVM)

サポートベクターマシンは, 統計的分類や回帰分析に広く用いられている教師あり学習法である. 図 3.5 に示すように, マージン (識別境界面と, 面に近いサンプルの間の距離) を最大化することで定義される線形分類器を基本モデルとする二値分類モデルであり, パーセプトロンと区別される.

サポートベクターマシンは, 線形および非線形のカテゴリ分類, 回帰, さらに異常検知に使用することができる. 複雑な中小規模のデータセットによる分類問題に多用されている. サポートベ

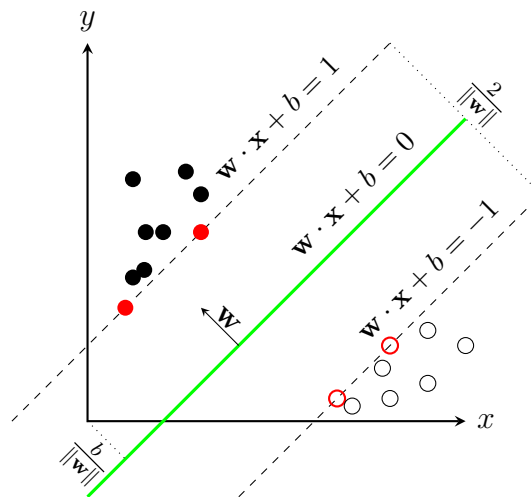


図 3.5: サポートベクターマシン

クターマシンは非確率的線形分類器であり，データセット内の異なるクラスを区別するために幾何学的手法を使用する．

サポートベクトルとは，超平面．このサポートベクトルを用いて，超平面の正と負のマージンを渡す．超平面は  $(n-1)$  次元の決定境界であり， $n$  はデータセットの列の数である．超平面は，異なるクラスの点/ベクトルを分離する（例えば，図 3.5 の緑色の線は超平面として機能する）．超平面は，マージンを最大化することで決定される．

カーネル関数とは，SVM において非線形データを高次元空間に射影し，超平面を用いてデータのクラス分けを行うために用いられる関数である．カーネル関数は色々考えられるが，最も使用されるカーネル関数は，「linear, poly, rbf, sigmoid」である．

カーネル関数の選定は，良い性能を持つ SVM モデルを構築するために重要である．サンプルの特徴量が多い場合，サンプルが線形分割可能であることが多く，線形カーネル関数が適切である．サンプル数が多くて，特徴量が少ない場合は，手動で特徴量を追加してサンプルを線形分割できるようにし，線形カーネル関数を使用することができる．サンプルの特徴量数とサンプル数両方とも少ない場合は，Gaussian カーネル関数（RBF カーネル関数の一種）が適している．

### 3.2.3 ランダムフォレスト (RF)

ランダムフォレストとは，アンサンブル学習モデルの 1 つである．アンサンブル学習では，複数の学習器を構築し，その結果を統合することで，単一の学習器よりも優れた汎化性能を得ることが期待できる．アンサンブル学習モデルには，学習器間の依存関係が強く連続的に学習するブースティング手法と，学習器間の依存関係がなく並列的に学習できるバギング手法がある．ランダムフォレストとはバギングアンサンブル学習手法の代表である．

図 3.6 に示すように，ランダムフォレストは多数の決定木で構成され，各決定木は互いに異なる．サンプルが入ったときにランダムフォレスト内の各決定木を個別に判定し，投票方式で最も投票数の多いカテゴリを最終的なモデル出力とする．これでランダムフォレストは決定木単体では過学習しやすいという欠点を克服する．

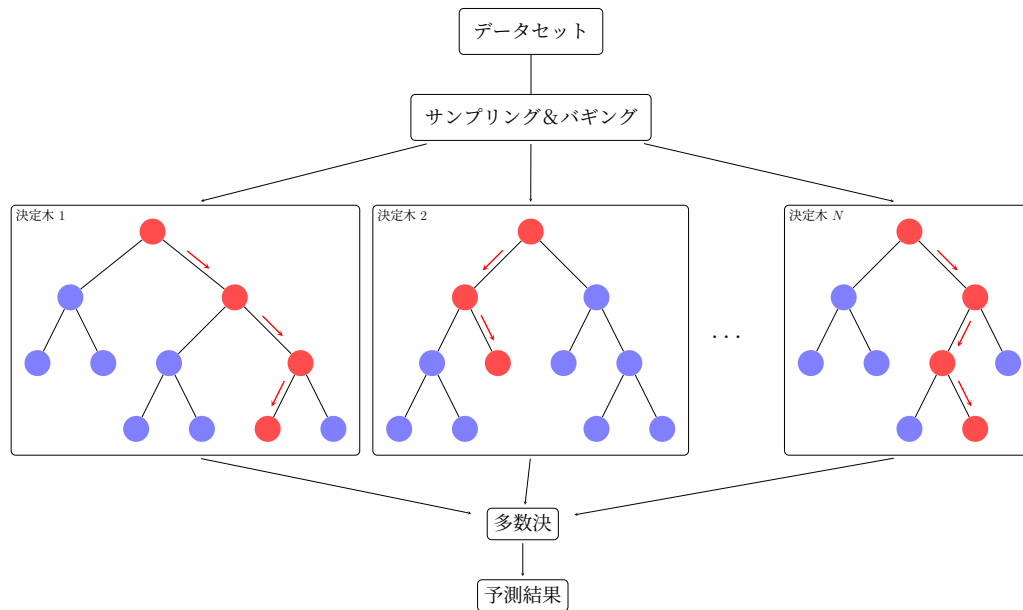


図 3.6: ランダムフォレスト

### 3.2.4 scikit-learn

scikit-learn[7] は python ベースの機械学習ライブラリであり、多くの機械学習アルゴリズムを提供する。scikit-learn を用いて、多くの機械学習モデルの訓練とテストを速く実装することができる。本研究では、MLPClassifier (多層パーセプトロン) [8], SVC (サポートベクターマシン) [9], RandomForestClassifier (ランダムフォレスト) [10] を使用する。以下は基本的な使用方法である。

```

1 # 1. MLP
2 from sklearn.neural_network import MLPClassifier
3
4 mlp = MLPClassifier(
5     solver='adam',
6     activation='tanh',
7     alpha=0.0002,
8     hidden_layer_sizes=(20,),
9     max_iter=2000,
10    early_stopping=True,
11    random_state=False)
12 mlp.fit(vector_x_train, y_train)
13 y_predict = mlp.predict(vector_x_test)

```

```

1 # 2. SVM
2 from sklearn.svm import SVC
3
4 svm = SVC(kernel='rbf')
5 svm.fit(vector_x_train, y_train)
6 y_predict = svm.predict(vector_x_test)

```

```

1 # 3. RF
2 from sklearn.ensemble import RandomForestClassifier
3
4 rf = RandomForestClassifier(n_estimators=100)
5 rf.fit(vector_x_train, y_train)
6 y_predict = svm.predict(vector_x_test)

```

### 3.3 感情推定ツール VADER

VADER (Valence Aware Dictionary and sEntiment Reasoner) は、語彙とルールに基づいた感情推定ツールである。特にソーシャルメディアで表現された感情に優れている。[11]

VADER は与えられた文章に対して、VADER は「compound,neg,neu,pos」4つのスコアを算出する。

compound は与えられた文章に対して、語彙内の各単語の感情価 (valence) スコアを合算し、ルールに従って調整された後、-1 (最もネガティブ) と +1 (最もポジティブ) の間になるように正規化される数値である。感情推定結果を一次元の数値として求める場合は、compound は一番有用なスコアである。感情をポジティブ、ニュートラル、ネガティブのいずれかに分類するための標準的な閾値として有用である。典型的な閾値は以下の通りである [12]。

- ポジティブ：  $\text{compound} \geq 0.05$
- ニュートラル：  $(\text{compound} > -0.05) \text{ and } (\text{compound} < 0.05)$
- ネガティブ：  $\text{compound} \leq -0.05$

pos, neu, neg の各スコアは、各カテゴリに分類される単語の割合である（したがって、これらはすべて加算されると1になる）。これらは、与えられた文章に対して感情がどのように伝えられ、レトリックに埋め込まれているかという文脈と表現を分析したい場合に、最も有用な測定基準である。しかしこれらの割合は、各語彙項目（単語、顔文字/絵文字、頭文字など）を肯定、否定、中立のクラスに「分類」したもので、「word-order sensitivity for sentiment-laden multi-word phrases, degree modifiers, word-shape amplifiers, punctuation amplifiers, negation polarity switches, or contrastive conjunction sensitivity」[12] の VADER ルールベースの拡張は考慮されていない。

また、VADER の辞書は英語のみを対応されるため、本研究では英語の Tweet のみを収集して使用する。

### 3.4 移動平均線

移動平均線 (MA) とは、統計的な手法を用いて一定期間の証券価格の平均値を算出し、その平均値を線で結び、証券の値動きの傾向を観察するための指標である。価格の推移や流れなど、相場の方向性を知る手がかりとなるテクニカルチャートの代表的なものである。現在の証券価格を表すローソク足と合わせて証券売買のタイミングを判断するためによく使用される。



加重期間の長さによって、長期移動平均、中期移動平均、短期移動平均の3種類がある。短期平均は通常3日、6日、10日、15日線、中期平均は通常25日、30日、75日線、長期平均は150日、200日、295日線がある。

平均値の計算方法によって単純移動平均線 (SMA)、加重移動平均線 (WMA)、指数平滑移動平均線 (EMA) など様々な種類がある。

式 3.5 は単純移動平均線 (SMA) の平均値の計算式である。n は期間値、 $p_i$  は i 日目の終値を表す。

$$SMA_n = \frac{1}{n} \sum_{i=1}^n p_i \quad (3.5)$$

式 3.6 は加重移動平均線 (WMA) の平均値の計算式である。n は期間値、 $p_i$  は i 日目の終値を表す。

$$WMA_n = \frac{\sum_{i=1}^n i \times p_i}{\sum_{i=1}^n i} \quad (3.6)$$

式 3.7 は指数平滑移動平均線 (EMA) の計算式である。n は期間値、t は本日、EMA は指数平滑移動平均、y は前日を表す。

$$\begin{cases} EMA_t = p_t \times k + EMA_y \times (1 - k) \\ k = \frac{2}{n + 1} \quad (n \geq 2) \end{cases} \quad (3.7)$$

移動平均線の傾き (Slope MA) は、移動平均が動く角度のことである。角度が急であればあるほど、支持や抵抗は大きくなる。

### 3.5 Tweet データ収集について

Tweet データを収集する最も便利な方法は、Twitter の公式 API を使用することである。しかし、アクセス数の制限 [13] がこの API を利用する際の大きな欠点である。この制限は、ユーザーのウェブページ上での検索操作をシミュレートして検索結果を得ることで、ある程度は取り除くことができる。Twint [14] は、この手法を実装したツールの一つである。Twint は Python で書かれた Twitter スクレイピングツールで、Twitter の API を使用せずに直接 Twitter からツイートをスクレイピングすることができる。Twitter の検索演算子を利用して、特定のユーザーのツイートをスクレイピングしたり、特定のトピック、ハッシュタグ、トレンドに関連するツイートをスクレイピングすることができる。また、Twitter ユーザーのフォロワー、ユーザーがいいねした Tweet、フォローしているユーザーをスクレイピングすることもできる。Twint を使用する利点は：

- ほぼ全てのツイートを取得可能。
- Python プログラムで組立てしやすい。
- 匿名で、Twitter のサインアップなしで利用できる。
- レート制限なし。

ただし、Twint を利用する場合でも、ユーザープロフィールを取得する際にはアクセス頻度が制限される。1 時間あたり最大 3200 人のプロフィールを取得することができる。本研究で収集したツイートは 100 万人近いユーザーからのものであり、これだけの数のユーザープロフィールを取得するには多くの時間が必要である。

## 第4章 提案手法と実装

1.2 項で述べた通り，先行研究では当日のマーケットデータと感情推定指標のみ使用しており，時系列であることを考慮していない．また，取得した Tweet の感情推定において，重みを付けておらず，各 Tweet は同様に扱われて計算されている．しかし，Follower 数，Reply 数，Retweet 数，Like 数は Tweet ごとに大きく異なっている．これらの数値は Tweet の影響度を表しているため，Tweet ごとの影響度の重みを感情推定モデルに反映することにより，感情推定精度を向上させることが期待できる．本研究では，感情推定の際に Follower 数，Reply 数，Retweet 数，Like 数，及び当日の Tweet 取得総数などにより各 Tweet の重みを算出する方法を提案する．

本章では生データ処理，特徴量の選定，モデル訓練手法及びそれぞれの実装方法を述べる．4.1 項では取得した Tweet の生データの処理法を述べる．4.2 項では感情推定指標及びマーケット指標を特徴量に様々な使える形式に加工する方法を述べる．4.3 項では特徴量の組合せ方法を述べる．4.4 項ではモデル訓練の方法について述べる．

## 4.1 データ収集及びメタデータの前処理

データ収集は、ツイート収集、ツイート作成者プロフィール収集、マーケットデータ収集の3つに分かれる。まず、Twintを使って、「bitcoin」、「ethereum」、「ripple」、「litecoin」というタグが付いた Tweet を検索する。検索条件を設定するときは、Tagに加えて、言語、日付も設定する。本研究では英語のデータのみを収集する。また、本研究では、2020年1月1日から2020年12月31日までの通年のデータを収集するため、ループを使って Twint を呼び出し、ループ本体内で対象日を設定する必要がある。そして、収集した Tweet データの username を集めて、Twint を用いてユーザーのプロフィールを取得する。

収集したデータは、まず日付ごとに CSV ファイルとして保存される。データ照会などの後続処理を容易にするため、本研究では Postgresql データベースを用いて、収集した Tweet データを作成した tweet\_btc, tweet\_eth, tweet\_xrp, tweet\_ltc テーブルにインポートする。これらのテーブルは、取得した Tweet のメタデータと同じ構造を持っている。表 4.1 は取得した Tweet データ及び本研究で使用するテーブルの項目を示す。赤字のは本研究に使われる項目である。また、プロフィールデータから username, followers (Follower 数) を twitter\_user テーブルに保存する。

表 4.1: tweet\_xxx テーブルのカラム

id	conversation_id	created_at	date	time	timezone
user_id	username	name	place	tweet	language
mentions	urls	photos	replies_count	retweets_count	likes_count
hashtags	cashtags	link	retweet	quote_url	video
thumbnail	near	geo	source	user_rt_id	user_rt
retweet_id	reply_to	retweet_date	translate	trans_src	trans_dest

続いて、VADER を用いて収集した Tweet の感情スコアを算出する。

```
1 from vaderSentiment.vaderSentiment \
2     import SentimentIntensityAnalyzer
3
4 coins = ['btc', 'eth', 'xrp', 'ltc']
5 analyzer = SentimentIntensityAnalyzer()
6
7 for coin in coins:
8     with connector.cursor() as cur:
9         df = pd.read_sql(
10             sql=f'select \
11                 id, tweet \
12             from twet.tweet_{coin} \
13             order by date asc',
14             con=connector)
15         for index, row in df.iterrows():
16             senti_score = \
17                 analyzer.polarity_scores(row.tweet)
```

```

18         cur.execute(
19             f'insert into twet.vader_{coin} \
20             (id, neg, neu, pos, compound) \
21             values (%s, %s, %s, %s, %s)',
22             row.id,
23             senti_score['neg'],
24             senti_score['neu'],
25             senti_score['pos'],
26             senti_score['compound']
27         )

```

マーケットデータは、様々なサイトが API で公開されている。本研究は「cryptocompare.com」の API を用いて取得する。マーケットのメタデータは、始値 (open)、終値 (close)、高値 (high)、安値 (low)、ボリューム (volume) で構成される。いずれも本研究に使われる特徴データである。後ほど 4.3.2 のデータ加工のため、取得期限は 2019 年 12 月 1 日から 2020 年 12 月 31 日までとする。

## 4.2 感情推定及びマーケットの特徴データ

本研究は、4.1 で述べた Tweet ごとの感情推定スコア、日次のマーケットデータを複数方法で加工し、特徴データを作成する。作成した特徴データを定めたルールで特徴量を抽出して実験で使用する。

### 4.2.1 感情推定スコアを加工

本研究は、以下の方法で感情推定スコアを加工する。T は Tweet データを収集した期間中の日にちの集合である。  $n = |T|$ ,  $(t_i, u_i) \in N$  .  $t_i$  は  $i$  番目 Tweet の感情推定スコアである (compound, neg, neu, pos).  $u_i$  は  $i$  番目 Tweet の作者の Follower 数である。

- 先行研究で用いられた、単純な感情推定スコアの日ごとの平均値。

$$v_{compound} = \frac{\sum_{i=1}^n t_i^{compound}}{n} \quad (4.1)$$

$$v_{neu} = \frac{\sum_{i=1}^n t_i^{neu}}{n} \quad (4.2)$$

$$v_{neg} = \frac{\sum_{i=1}^n t_i^{neg}}{n} \quad (4.3)$$

$$v_{pos} = \frac{\sum_{i=1}^n t_i^{pos}}{n} \quad (4.4)$$

$$v_{pol} = \sqrt{v_{pos}v_{neg}} \quad (4.5)$$

$v_{pol}$  は全ての方法でも同じ定義であるため以下は省略する。この方法で計算した特徴データの名を SA1 とする。

- Follower 数を重要度として計算する.

$$v_{compound} = \frac{\sum_{i=1}^n u_i^{followers} t_i^{compound}}{n} \quad (4.6)$$

$$v_{neu} = \frac{\sum_{i=1}^n u_i^{followers} t_i^{neu}}{n} \quad (4.7)$$

$$v_{neg} = \frac{\sum_{i=1}^n u_i^{followers} t_i^{neg}}{n} \quad (4.8)$$

$$v_{pos} = \frac{\sum_{i=1}^n u_i^{followers} t_i^{pos}}{n} \quad (4.9)$$

この方法で計算した特徴データの名を SA2 とする.

- Follower 数, Reply 数, Retweet 数, Like 数を重み付けて重要度を計算する.  
w はこれら要素の重みである.

$$k_i = w^{followers} u_i^{followers} + w^{replies} t_i^{replies} + w^{retweets} t_i^{retweets} + w^{likes} t_i^{likes} \quad (4.10)$$

$$v_{compound} = \frac{\sum_{i=1}^n k_i t_i^{compound}}{n} \quad (4.11)$$

$$v_{neu} = \frac{\sum_{i=1}^n k_i t_i^{neu}}{n} \quad (4.12)$$

$$v_{neg} = \frac{\sum_{i=1}^n k_i t_i^{neg}}{n} \quad (4.13)$$

$$v_{pos} = \frac{\sum_{i=1}^n k_i t_i^{pos}}{n} \quad (4.14)$$

表 4.2 では本研究に使用する重みを示す.

表 4.2: Follower 数, Reply 数, Retweet 数, Like 数の重み				
	Follower 数の重み $w^{followers}$	Reply 数の重み $w^{replies}$	Retweet 数の重み $w^{retweets}$	Like 数の重み $w^{likes}$
SA3	1	1	1	1
SA4	0	1	1	1
SA5	0	1	0.5	0.1
SA6	0.01	1	0.5	0.1
SA7	0.001	1	0.5	0.1

この方法で計算した特徴データの名を SA3～SA7 とする.

1 行目は, 各要素が Tweet の重要度に与える影響が均等であることを表している. 2 行目と 3 行目は, Follower 数が Tweet の重要性に与える影響が無視できることを表している. 2 行目

は、Follower 数以外の要素で、Tweet の重要度に同等の影響を与えるものを表す。3 行目以降は、Reply 数、Retweet 数、Like 数が、Tweet の重要度に異なる影響を与えられられる。「Reply 数」が最も影響力が大きく、次いで「Retweet 数」、「Like 数」は最も影響力が小さい。

また、表 4.3 に示すように、Reply、Retweet、Like 数は全て 0 の Tweet が全体の半分以上を占めている。これらの Tweet を完全に無視することはできないため、平均値を計算する際に小さな重み (0.001) を与える。同様に、対象となった Twitter ユーザー総数 966747 人のうち 59903 人 (6.2%) は Follower 数が 0 であり、これらも 0.001 の重みを与える。

表 4.3: Tweet の Reply 数, Retweet 数, Like 数の統計

仮想通貨	Tweet 数	Reply 数 (P)	Retweet 数 (R)	Like 数 (L)	(PRL) 全て 0 の Tweet 数
BTC	8,392,367	5,525,025	6,601,215	27,808,385	4,732,408
ETH	3,820,090	2,687,632	4,087,468	13,371,784	2,150,136
XRP	3,323,695	2,390,975	2,485,078	15,718,324	1,613,597
LTC	1,216,653	658,387	1,275,760	4,390,027	635,707

よって、 $k_i$  の計算式は以下になる。

$$\begin{cases} j \in \{followers, replies, retweets, likes\} \\ k_i^j = \begin{cases} w^j u_i^j & (u_i^j > 0) \\ 0.001 & (u_i^j = 0) \end{cases} \\ k_i = \sum k_i^j \end{cases} \quad (4.15)$$

以下は感情推定スコアの平均値計算の実装である。

```

1 # 感情推定スコアの平均値を求める
2 # w は重み
3 # w.follower フォロワー数の重み
4 # w.reply リプライ数の重み
5 # w.retweet リツイート数の重み
6 # w.like 「いいね！」数の重み
7 # w.nodata フォロワー数、リプライ数、リツイート数、「いいね！」数
8 # は 0 の場合の重み
9 def calc_sentiment_mean(df, w):
10     df_r = df.replace(np.nan, 0)
11     df2 = pd.DataFrame([], \
12         columns=['neg', 'neu', 'pos', 'compound'])
13     follower = (df_r['followers'] * w.follower) \
14         .replace(0, w.nodata)
15     reply = (df_r['replies_count'] * w.reply) \
16         .replace(0, w.nodata)
17     retweet = (df_r['retweets_count'] * w.retweet) \
18         .replace(0, w.nodata)

```

```

19 like = (df_r['likes_count'] * w.like)\
20         .replace(0,w.nodata)
21 totalWeight = follower + reply + retweet + like
22
23 df2['neg'] = df_r['neg'] * totalWeight
24 df2['neu'] = df_r['neu'] * totalWeight
25 df2['pos'] = df_r['pos'] * totalWeight
26 df2['compound'] = df_r['compound'] * totalWeight
27 return df2.mean(numeric_only=True)

```

上記全ての方法で計算された感情の日ごとの推定結果に対して，3.4で述べた単純移動平均線（SMA），単純移動平均線の傾き（Slope SMA），指数平滑移動平均線（EMA），指数平滑移動平均線の傾き（Slope EMA）を計算する．それぞれ期間が3日，5日の値を計算する．この方法で計算した特徴データの名をS-SMAM-N，S-EMAM-Nとする．Mは上記平均値スコア名に付いた番号を表す．Nは期間の数値を表す．特徴量を抽出時に，平均値のみ，あるいは移動平均の組み合わせを使用する．組合せの一覧を表4.4で示す．また，表4.5では表4.4の展開であり，各パターンの内訳項目を示す．

表 4.4: Tweet 感情推定のための日次の特徴データの組み合わせ

-	No-MA	SMA3	EMA3	SMA5	EMA5
SA1	SA1	SA1+S-SMA1-3	SA1+S-EMA1-3	SA1+S-SMA1-5	SA1+S-EMA1-5
SA2	SA2	SA2+S-SMA2-3	SA2+S-EMA2-3	SA2+S-SMA2-5	SA2+S-EMA2-5
SA3	SA3	SA3+S-SMA3-3	SA3+S-EMA3-3	SA3+S-SMA3-5	SA3+S-EMA3-5
SA4	SA4	SA4+S-SMA4-3	SA4+S-EMA4-3	SA4+S-SMA4-5	SA4+S-EMA4-5
SA5	SA5	SA5+S-SMA5-3	SA5+S-EMA5-3	SA5+S-SMA5-5	SA5+S-EMA5-5
SA6	SA6	SA6+S-SMA6-3	SA6+S-EMA6-3	SA6+S-SMA6-5	SA6+S-EMA6-5
SA7	SA7	SA7+S-SMA7-3	SA7+S-EMA7-3	SA7+S-SMA7-5	SA7+S-EMA7-5



表 4.5: Tweet 感情推定日次特徴組合せの展開

連番	SA	S-MA	内訳項目
1	SA1	-	compound,neg,neu,pos,pol
2	-	S-SMA1-3	compound_sma3,compound_sma_slope3,neg_sma3, neg_sma_slope3,neu_sma3,neu_sma_slope3,pos_sma3,pos_sma_slope3
3	SA1	S-SMA1-3	compound,neg,neu,pos,pol,compound_sma3,compound_sma_slope3, neg_sma3,neg_sma_slope3,neu_sma3,neu_sma_slope3, pos_sma3,pos_sma_slope3
4	-	S-EMA1-3	compound_ema3,compound_ema_slope3,neg_ema3, neg_ema_slope3,neu_ema3,neu_ema_slope3,pos_ema3,pos_ema_slope3
5	SA1	S-EMA1-3	compound,neg,neu,pos,pol,compound_ema3,compound_ema_slope3, neg_ema3,neg_ema_slope3,neu_ema3,neu_ema_slope3, pos_ema3,pos_ema_slope3
6	-	S-SMA1-5	compound_sma5,compound_sma_slope5,neg_sma5, neg_sma_slope5,neu_sma5,neu_sma_slope5,pos_sma5,pos_sma_slope5
7	SA1	S-SMA1-5	compound,neg,neu,pos,pol,compound_sma5,compound_sma_slope5, neg_sma5,neg_sma_slope5,neu_sma5,neu_sma_slope5, pos_sma5,pos_sma_slope5
8	-	S-EMA1-5	compound_ema5,compound_ema_slope5,neg_ema5, neg_ema_slope5,neu_ema5,neu_ema_slope5,pos_ema5,pos_ema_slope5
9	SA1	S-EMA1-5	compound,neg,neu,pos,pol,compound_ema5,compound_ema_slope5, neg_ema5,neg_ema_slope5,neu_ema5,neu_ema_slope5, pos_ema5,pos_ema_slope5
...	...	.....	.....
...	...	.....	.....
...	...	.....	.....
62	-	S-EMA7-5	compound_ema5,compound_ema_slope5,neg_ema5, neg_ema_slope5,neu_ema5,neu_ema_slope5,pos_ema5,pos_ema_slope5
63	SA7	S-EMA7-5	compound,neg,neu,pos,pol,compound_ema5,compound_ema_slope5, neg_ema5,neg_ema_slope5,neu_ema5,neu_ema_slope5, pos_ema5,pos_ema_slope5

単純移動平均線、指数平滑移動平均線、移動平均線の傾きの計算は pandas を利用する.

```
1 import pandas as pd
2
3 # 単純移動平均線を作成
4 def create_sma(close, span):
5     return close.rolling(window = span).mean()
6
7 # 指数平滑移動平均線を作成
8 def create_ema(close, span):
9     sma = create_sma(close, span)[:span]
10    return pd.concat([sma, close[span:]])\
11        .ewm(span = span, adjust = False).mean()
12
13 # 移動平均線の傾きを作成
14 def create_ma_slope(ma, span):
15     ma_slope = []
16     for i in range(len(ma)):
17         if i <= span + 1:
18             ma_slope.append(np.nan)
19         else:
20             ma_slope.append((ma[i] - ma[i - span]) /\
21                             (i - (i - span)))
22     return ma_slope
```

### 4.2.2 マーケットデータ加工

まずは、4.1 で述べたマーケット日次データに対して、直近 15 日の高値、安値を取得する。高値更新や安値更新はトレンド時の重要なシグナルであるため、特徴データとして予測精度向上させることを期待できる。

そして、3.4 で述べた単純移動平均線 (SMA)、単純移動平均線の傾き (Slope SMA)、指数平滑移動平均線 (EMA)、指数平滑移動平均線の傾き (Slope EMA) を計算する。それぞれ期間が 3 日、5 日の値を計算する。この方法で計算した特徴データの名を M-SMAN、M-EMAN とする。N は期間の数値を表す。特徴量を抽出時に、価格のみ、あるいは移動平均の組み合わせを使用する。

表 4.6: マーケット特徴組合せの展開

連番	M-MA	内訳項目
1	-	open,close,high,low,volumeto
2	M-SMA3	open,close,high,low,volumeto,high15,low15,sma3,sma_slope3
3	M-EMA3	open,close,high,low,volumeto,high15,low15,ema3,ema_slope3
4	M-SMA5	open,close,high,low,volumeto,high15,low15,sma5,sma_slope5
5	M-EMA5	open,close,high,low,volumeto,high15,low15,ema5,ema_slope5

## 4.3 特徴量の抽出

本研究では、4.2 で示した特徴データの最適な組み合わせを求めることを試みる。表 4.5 に示すデータと表 4.6 に示すデータから、本研究の実験に用いる特徴量として、0-1 行の各組み合わせを取り上げる。これらの固有値の組み合わせは 383 通りがある。

以下は特徴量のパターンである。4.2 感情推定の計算方法及び移動平均線の種類により更に展開できる。

マーケット情報のみの場合:

$$V_{(t)} = \begin{bmatrix} open \\ close \\ high \\ low \\ volumeto \end{bmatrix} \quad (4.16)$$

マーケット情報及び移動平均線の場合:

$$V_{(t)} = \begin{bmatrix} open \\ close \\ high \\ low \\ volumeto \\ high15 \\ low15 \\ ma \\ ma\_slope \end{bmatrix} \quad (4.17)$$

感情推定平均値のみの場合:

$$V_{(t)} = \begin{bmatrix} compound \\ neg \\ neu \\ pos \\ pol \end{bmatrix} \quad (4.18)$$

感情推定平均値の移動平均線のみの場合:

$$V_{(t)} = \begin{bmatrix} compound\_ma \\ compound\_ma\_slope \\ neg\_ma \\ neg\_ma\_slope \\ neu\_ma \\ neu\_ma\_slope \\ pos\_ma \\ pos\_ma\_slope \end{bmatrix} \quad (4.19)$$

感情推定平均値及び移動平均線の場合:

$$V_{(t)} = \begin{bmatrix} compound \\ neg \\ neu \\ pos \\ pol \\ compound\_ma \\ compound\_ma\_slope \\ neg\_ma \\ neg\_ma\_slope \\ neu\_ma \\ neu\_ma\_slope \\ pos\_ma \\ pos\_ma\_slope \end{bmatrix} \quad (4.20)$$

マーケット情報と感情推定平均値の場合:

$$V_{(t)} = \begin{bmatrix} open \\ close \\ high \\ low \\ volumeto \\ compound \\ neg \\ neu \\ pos \\ pol \end{bmatrix} \quad (4.21)$$

マーケット情報と感情推定平均値の移動平均線の場合：

$$V_{(t)} = \begin{bmatrix} open \\ close \\ high \\ low \\ volumeto \\ compound\_ma \\ compound\_ma\_slope \\ neg\_ma \\ neg\_ma\_slope \\ neu\_ma \\ neu\_ma\_slope \\ pos\_ma \\ pos\_ma\_slope \end{bmatrix} \quad (4.22)$$

マーケット情報と感情推定平均値及び移動平均線の場合：

$$V_{(t)} = \begin{bmatrix} open \\ close \\ high \\ low \\ volumeto \\ compound \\ neg \\ neu \\ pos \\ pol \\ compound\_ma \\ compound\_ma\_slope \\ neg\_ma \\ neg\_ma\_slope \\ neu\_ma \\ neu\_ma\_slope \\ pos\_ma \\ pos\_ma\_slope \end{bmatrix} \quad (4.23)$$

マーケット情報，感情推定の平均値，及び両方の移動平均線の場合:

$$V_{(t)} = \begin{bmatrix} \textit{open} \\ \textit{close} \\ \textit{high} \\ \textit{low} \\ \textit{volumeto} \\ \textit{high15} \\ \textit{low15} \\ \textit{ma} \\ \textit{ma\_slope} \\ \textit{compound} \\ \textit{neg} \\ \textit{neu} \\ \textit{pos} \\ \textit{pol} \\ \textit{compound\_ma} \\ \textit{compound\_ma\_slope} \\ \textit{neg\_ma} \\ \textit{neg\_ma\_slope} \\ \textit{neu\_ma} \\ \textit{neu\_ma\_slope} \\ \textit{pos\_ma} \\ \textit{pos\_ma\_slope} \end{bmatrix} \quad (4.24)$$

上記パターンによる，抽出した特徴量の特徴数は，5, 8, 9, 10, 13, 14, 17, 18, 22 となる．全て種類の特徴量を抽出し，Dataset としてファイルを保存する．

## 4.4 モデル実装と訓練

scikit-learn の MLPClassifier, SVC, RandomForestClassifier を用いてモデルを実装する．モデル訓練とテスト時に，データセットの 80 - 20 で分割して，80 %は訓練用，20 %はテスト用とする．データセットは 2020/1/6～2020/12/30 の日次データとなる．

## 第5章 実験評価

本章では，実験の方法と結果及び評価について示す．

5.2 項では実験の準備及び手順について記載する．

5.3 項では評価指標について記載する．

5.4 項では，実験の結果の中に，一番良い結果の感情推定平均値及び指数平均線アルゴリズムを選定し，各特徴量のパターンの評価指標を比較した．

5.5 項では，実験結果をまとめた．

## 5.1 実験準備

実験のために、3.5 で示す方法で、2020/1/1～2020/12/31 の Twitter データ（英語のみ）の収集を行う。収集した Tweet の統計情報を表 5.1 で示す。

表 5.1: 収集した Tweet の統計情報（仮想通貨別）

仮想通貨	Tweet 数	トータルの割合
BTC	8,392,367	50.10%
ETH	3,820,090	22.80%
XRP	3,323,695	19.84%
LTC	1,216,653	7.26%

同じ期間のマーケット情報を「cryptocompare.com」の API を用いて収集する [15]。表 5.2 で収集したマーケット情報の価格動向を示す。

表 5.2: 仮想通貨の価格変動の統計情報（仮想通貨別）

仮想通貨	値上がり日数	値下がり日数
Dataset		
BTC	208	152
ETH	200	160
XRP	191	169
LTC	189	171
テスト部分		
BTC	119	81
ETH	114	86
XRP	107	93
LTC	108	92

移動平均線は元の時系列データと関係があるため、実験時にデータセットをシャッフルできない。データセットには 360 のデータ件がある。本研究では、最初のサブデータセットとして 1 番目から 200 件を選択し、そのうち最初の 80 %（160 項目）をトレーニングデータセットとして、最後の 20 %（40 項目）をテストデータセットとして使用する。そして、40 件を後ろにずらして、つまり 41 番目のデータから始まる 200 項目が 2 番目のサブデータセットとする。こうして、合計 5 つのサブデータセットが生成される。各モデルのフルデータセットに対して、5 回の訓練およびテストが行われる。（MLP では異なるサイズの隠れ層を用いて訓練を行うため、回数は多くなる）

モデルを訓練する前に、scikit-learn の StandardScaler でデータセットを標準化する。

MLP のインプット層のサイズは特徴数と一致する。出力層のサイズは 1 で固定する。隠れ層のサイズは、特徴数による変動する。一定サイズ範囲の隠れ層により繰り返し学習し、最も良い結果だけを本レポートで掲載する。MLP の最適化アルゴリズムは adam、活性化関数は双曲線正接関数 Tanh を使用する。

SVM では RBF カーネルを使用し、それ以外のパラメータはデフォルトのままを使用する。



RF では決定木の数量は 1000 を使用し、それ以外のパラメータはデフォルトのままを使用する。

## 5.2 評価方法

Accuracy, Precision, Recall, F1-score を使用して結果を評価する。Precision を優先で評価される。

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \quad (5.1)$$

$$Precision = \frac{t_p}{t_p + f_p} \quad (5.2)$$

$$Recall = \frac{t_p}{t_p + f_n} \quad (5.3)$$

$$f1 = 2 \frac{v_{precision} v_{recall}}{v_{precision} + v_{recall}} \quad (5.4)$$

$t_p$  は実測値と予測値両方が正である数量。

$t_n$  は実測値が負、予測値が正である数量。

$f_p$  は実測値と予測値両方が負である数量。

$f_n$  は実測値が正、予測値が負である数量。

Accuracy は全てのテストサンプルが正しく分類された比率である。Precision は正に正しく分類されたサンプルと、正に分類された全てのサンプルの比率である。Recall は正に正しく分類されたサンプルと、実際の全ての正のサンプルの比率である。F1 スコアは Precision と Recall の調和平均である。仮想通貨の価格動向の予測は、誤った予測結果が多いと取引が損となるため Precision は最も重要なスコアとみなされる。

## 5.3 評価結果

4.3で述べた特徴量を抽出し MLP, SVM, RF モデルで実験を行って, 結果を総合的に評価し, 表 5.3 で示す SA4 (Follower 数, Reply 数, Retweet 数, Like 数を同じ重み) かつ指数移動平均線 (マーケットでは 3 日, Twitter では 5 日) は一番良い結果を得た. この特徴量は BTC の SVM では Precision 0.73, Recall 0.67 の予測結果を取れた.

表 5.3: Follower 数, Reply 数, Retweet 数, Like 数の重み

	Follower 数の重み $w^{followers}$	Reply 数の重み $w^{replies}$	Retweet 数の重み $w^{retweets}$	Like 数の重み $w^{likes}$
SA1	-	-	-	-
SA2	1	0	0	0
SA3	1	1	1	1
SA4	0	1	1	1
SA5	0	1	0.5	0.1
SA6	0.01	1	0.5	0.1
SA7	0.001	1	0.5	0.1

表 5.4 に BTC の評価結果を示す. BTC の場合は, MLP, SVM, RF いずれも良い性能を示す. マーケット, Twitter と Twitter の移動平均の組み合わせの特徴量では Precision 0.73, Recall 0.63, Accuracy 0.62 である. 他も 2 つの特徴量の組み合わせも同じ 0.73 の precision を達成したが, 他の評価指標はこれほど及ばない. また, Twitter 移動平均が含まれる特徴量は全て Precision 0.7 以上である. SVM では, 特徴量は「マーケットと Twitter と両方の移動平均」の場合, Precision も 0.73 であるが, 他の評価指標は MLP の最良結果に及ばない. RF でも特徴量は「マーケットと Twitter と両方の移動平均」の場合, 最も良い Precision 0.76 を達成した. どんなモデルに対しても, 移動平均特徴を追加後に, 性能が良くなる.

表 5.4: BTC の評価結果

モデル	特徴量種類	Precision
MLP	マーケットのみ	0.69
	マーケットとマーケット移動平均	0.73
	Twitter のみ	0.50
	Twitter と Twitter 移動平均	0.72
	マーケットと Twitter	0.71
	マーケットと Twitter と Twitter 移動平均	0.73
	マーケットと Twitter と両方の移動平均	0.73
SVM	マーケットのみ	0.47
	マーケットとマーケット移動平均	0.60
	Twitter のみ	0.61
	Twitter と Twitter 移動平均	0.64
	マーケットと Twitter	0.72
	マーケットと Twitter と Twitter 移動平均	0.68
	マーケットと Twitter と両方の移動平均	0.73
RF	マーケットのみ	0.55
	マーケットとマーケット移動平均	0.67
	Twitter のみ	0.63
	Twitter と Twitter 移動平均	0.64
	マーケットと Twitter	0.49
	マーケットと Twitter と Twitter 移動平均	0.51
	マーケットと Twitter と両方の移動平均	0.76

表 5.5 で ETH の評価結果を示す．ETH の場合は，一番良い性能のモデルは SVM である．MLP の場合，Twitter のみの場合は Precision 0.71 であり，マーケットのみの場合の 0.66 より高い．マーケットと Twitter と両方の移動平均の場合 Precision が 0.73 である．SVM の場合，特徴量「マーケットと Twitter と Twitter 移動平均」では一番良い性能の Precision 0.74，Recall 0.53 である．移動平均を加えることで Precision が大幅に改善された．RF の場合，マーケットのみが一番良い性能の Precision 0.63 となる．

表 5.5: ETH の評価結果

モデル	特徴量種類	Precision
MLP	マーケットのみ	0.66
	マーケットとマーケット移動平均	0.46
	Twitter のみ	0.71
	Twitter と Twitter 移動平均	0.57
	マーケットと Twitter	0.58
	マーケットと Twitter と Twitter 移動平均	0.60
	マーケットと Twitter と両方の移動平均	0.73
SVM	マーケットのみ	0.47
	マーケットとマーケット移動平均	0.59
	Twitter のみ	0.56
	Twitter と Twitter 移動平均	0.57
	マーケットと Twitter	0.63
	マーケットと Twitter と Twitter 移動平均	0.74
	マーケットと Twitter と両方の移動平均	0.68
RF	マーケットのみ	0.63
	マーケットとマーケット移動平均	0.61
	Twitter のみ	0.53
	Twitter と Twitter 移動平均	0.59
	マーケットと Twitter	0.59
	マーケットと Twitter と Twitter 移動平均	0.56
	マーケットと Twitter と両方の移動平均	0.55

表 5.6 で XRP の評価結果を示す。RF の場合は、一番良い性能のモデルは MLP である。MLP の場合は、特徴量「マーケットと Twitter と両方の移動平均」では一番の性能の Precision0.64 を達成した。SVM では特徴量「マーケットのみ」の性能が一番良い。Twitter 及び移動平均線を加えても効果がない。RF では特徴量「マーケットとマーケット移動平均」の性能が一番良い。マーケット移動平均は有意義のを示す。ただし、Twitter 関連の特徴を加えてもより良い結果を得られなかった。

表 5.6: XRP の評価結果

モデル	特徴量種類	Precision
MLP	マーケットのみ	0.61
	マーケットとマーケット移動平均	0.62
	Twitter のみ	0.53
	Twitter と Twitter 移動平均	0.62
	マーケットと Twitter	0.48
	マーケットと Twitter と Twitter 移動平均	0.63
	マーケットと Twitter と両方の移動平均	0.64
SVM	マーケットのみ	0.58
	マーケットとマーケット移動平均	0.43
	Twitter のみ	0.53
	Twitter と Twitter 移動平均	0.50
	マーケットと Twitter	0.58
	マーケットと Twitter と Twitter 移動平均	0.55
	マーケットと Twitter と両方の移動平均	0.57
RF	マーケットのみ	0.55
	マーケットとマーケット移動平均	0.60
	Twitter のみ	0.54
	Twitter と Twitter 移動平均	0.50
	マーケットと Twitter	0.58
	マーケットと Twitter と Twitter 移動平均	0.53
	マーケットと Twitter と両方の移動平均	0.52

表 5.7 で LTC の評価結果を示す。 LTC の場合は、一番良い性能のモデルも MLP である。 MLP の場合は、特徴量「マーケットと Twitter と両方の移動平均」では一番の性能の Precision0.70 を達成した。 SVM では特徴量「Twitter と Twitter 移動平均」の性能が一番良い。 RF では特徴量「マーケットとマーケット移動平均」の性能が一番良い。 マーケット移動平均だけを追加されて、Precesion, Reall, 及び Accuracy 全て向上できた。 ただし、更に Twitter 関連の特徴を加えてもより良い結果を得られなかった。

表 5.7: LTC の評価結果

モデル	特徴量種類	Precision
MLP	マーケットのみ	0.63
	マーケットとマーケット移動平均	0.62
	Twitter のみ	0.58
	Twitter と Twitter 移動平均	0.60
	マーケットと Twitter	0.55
	マーケットと Twitter と Twitter 移動平均	0.63
	マーケットと Twitter と両方の移動平均	0.70
SVM	マーケットのみ	0.47
	マーケットとマーケット移動平均	0.44
	Twitter のみ	0.39
	Twitter と Twitter 移動平均	0.52
	マーケットと Twitter	0.31
	マーケットと Twitter と Twitter 移動平均	0.41
	マーケットと Twitter と両方の移動平均	0.44
RF	マーケットのみ	0.65
	マーケットとマーケット移動平均	0.68
	Twitter のみ	0.57
	Twitter と Twitter 移動平均	0.61
	マーケットと Twitter	0.62
	マーケットと Twitter と Twitter 移動平均	0.48
	マーケットと Twitter と両方の移動平均	0.46

先行研究 2.1 と同じ特徴量のケースでも，高い Precision を得ることが示された．表 5.8 を示すように，BTC の場合，先行研究と同じ処理方式の感情推定特徴量と本研究提出の重みつきで計算した感情推定特徴量の実験結果である．本研究で提案した手法は，Twitter の感情推定のみの特徴を用いても，マーケットと Twitter の感情推定の特徴量を用いても，ほぼすべての場合では Precision が向上し，Precision だけでなく Accuracy や Recall もほとんどのケースで向上することがわかった．

表 5.8: 先行研究と比較 (BTC)

モデル	特徴量種類	Accuracy	Precision
MLP	マーケットのみ	0.51	0.69
	Twitter のみ (先行)	0.42	0.53
	Twitter のみ (本研究)	0.50	0.50
	マーケットと Twitter (先行)	0.60	0.60
	マーケットと Twitter (本研究)	0.60	0.71
SVM	マーケットのみ	0.52	0.47
	Twitter のみ (先行)	0.60	0.60
	Twitter のみ (本研究)	0.58	0.61
	Twitter と Twitter 移動平均 (先行)	0.54	0.56
	Twitter と Twitter 移動平均 (本研究)	0.62	0.64
RF	マーケットのみ	0.45	0.55
	Twitter のみ (先行)	0.47	0.53
	Twitter のみ (本研究)	0.54	0.63
	マーケットと Twitter (先行)	0.48	0.48
	マーケットと Twitter (本研究)	0.51	0.49

## 5.4 まとめ

すべての仮想通貨において、他の機械学習モデルに比べて MLP は比較的良い予測精度を達成した。MLP モデルでは、感情推定とその移動平均が最終結果にプラスの影響を与えた。SVM モデルは、BTC と ETH については MLP と同様に、移動平均の特徴を追加することで Precision が大幅に向上したが、XRP と LTC についてはそうではなかった。RF の場合、Market の移動平均はすべての仮想通貨に有効だが、感情推定は予測結果にプラスの影響を与えなかった。

本研究で提案した感情推定の平均値の算出方法（Follower, Reply, Retweet, Like の数に重み付けをして感情推定値を処理する）では、本研究で利用した Tweet の期間において、先行研究の方法より高い Precision を得られた。さらに移動平均線（及び傾き）を加えることで、多くの場合ではより高い精度も得られる結果となった。また、マーケットデータのみを使用した場合よりも、より多くのケースで良好な結果が得られた。本研究のデータ数及び期間は、先行研究とは違うため、結果を直接定量的に比較ができないが、各感情推定導入前後の予測精度の差異などを比較した。3.1 で示すように、先行研究では感情推定導入後の 24 パータンのうち、有効なパターンは 4 つがあり、有効率は 16.7% である。本研究では、感情推定導入後の 60 パータンのうち、有効なパターンは 19 個があり、有効率は 31.7% である。以上のことから、本研究において提案した方法は、先行研究より多くの場合で有効であることが明らかになった。



## 第6章 おわりに

### 6.1 本研究の成果

本研究では、Tweet の感情推定結果を加工し、平均値の算出に Follower, Reply, Retweet, Like の重み付けを導入し、移動平均とその傾きによる特性量に時系列の考慮も導入することで、先行研究よりも優れた結果を得られた。

よって、ツイートの感情推定の平均値を算出する際には、「Follower 数」、「Reply 数」、「Reweeet 数」、「Like!数」の影響を無視できないことが明らかになった。そして、Follower の数よりも Reply, Retweet, Like の数の方が重要である。つまり、「Reply 数」、「Reweeet 数」、「Like!数」は、その Tweet の【質】をよりよく表す指標となる。

また、移動平均線とその傾きは、人が取引をする際の指標としてだけでなく、機械学習においても非常に有用な特徴量であることが、今回の研究で明らかになった。移動平均とその傾きの特徴量に加えることで、多くの場合に予測結果が改善されることがわかった。

### 6.2 今後の課題

本研究の結果は、Tweet の感情推定結果をより適切に扱うことで予測 Precision が向上することを示し、Tweet の感情推定精度をさらに向上させることができれば、予測結果はさらに向上すると考えられる。VADER では辞書とルールに基づいたツールであって、表 6.1 ニュートラル的単語を使った文に対して推定は失敗しやすい。1 番目はポジティブな感情を示したが VADER での推定結果はニュートラルとなる。2 番目はネガティブな感情を示したが VADER での推定結果はニュートラルとなる。したがって、今後の研究において、感情推定を改善する手段（例えば、Bert の使用）を検討することが重要となる。つまり、VADER より良い感情推定結果を得ることが今後の課題の 1 つである。

表 6.1: VADER で感情推定の失敗例

-	Tweet	compound	neg	neu	pos
1	Don't be left behind, cryptocurrency is the future, be a partner not a trader. Buy and HODL.	0.00	0.00	1.00	0.00
2	We should all sold for \$BTC at this breakout candle.	0.00	0.00	1.00	0.00

また、本研究では、Tweet の感情推定結果を日ごとで算出しているが、Tweet の感情を市場

へ反映することは、間隔が短いほど市場の動きを正確に予測することができると思われる。1時間単位での学習や予測においても本提案手法は有効な可能性があり、Tweetの感情がマーケットに与える影響を様々な時間スケールで検証することも今後の課題の一つである。

本研究では、MLP、SVM、RFの3つの機械学習モデルを使用したか、これらも特徴量や仮想通貨が異なる条件下で異なるパフォーマンスを示した。これら3つのモデルを組み合わせ、VotingClassifierを用いてより良い予測結果を得ることも今後の課題の1つである。

# 謝辞

本研究を進めるにあたって、岡田研究室のいろいろな方にお世話になりました。また岡田研の東京サテライトメンバーの皆様には同じ社会人学生として励みになり、頑張れた事に感謝します。最後に長期履修で4年間、進みが遅いながらも終始暖かくご指導いただきました岡田将吾教授に深く感謝致します。

# 参考文献

- [1] <https://ja.wikipedia.org/wiki/仮想通貨>
- [2] <https://planbtc.com/>
- [3] <https://www.bbc.com/news/business-55939972>
- [4] David Garcia and Frank Schweitzer. Social signals and algorithmic trading of Bitcoin. R. Soc. Open Sci.2015,2, 150288.
- [5] Franco Valencia;Alfonso Gomez-Espinosa and Benjamin Valdes-Aguirre. Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning, Entropy 2019, 21(6), 589.
- [6] <https://github.com/cjhutto/vaderSentiment>
- [7] <https://scikit-learn.org/>
- [8] [https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html)
- [9] <https://scikit-learn.org/stable/modules/svm.html#svc>
- [10] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- [11] <https://github.com/cjhutto/vaderSentiment#readme>
- [12] <https://github.com/cjhutto/vaderSentiment#about-the-scoring>
- [13] <https://developer.twitter.com/ja/docs/twitter-api/rate-limits>
- [14] <https://github.com/twintproject/twint>
- [15] <https://min-api.cryptocompare.com/documentation?key=Historical&cat=dataHistoday>
- [16] Franco Valencia;Alfonso Gomez-Espinosa and Benjamin Valdes-Aguirre. Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning, Entropy 2019, 21(6), 589,p7.