



台風の弱体化を目的とした氷投下地点の推定及び考察

宮岡, 伶光

(Citation)

課題研究優秀論文集, 2025:223-262

(Issue Date)

2026-01

(Resource Type)

departmental bulletin paper

(Version)

Version of Record

(JaLCD0I)

<https://doi.org/10.24546/0100499445>

(URL)

<https://hdl.handle.net/20.500.14094/0100499445>



2025 年度 卒業研究最終論文

台風の弱体化を目的とした氷投下地点の推定及び考察

神戸大学附属中等教育学校 12 回生

6 年 3 組 35 番

宮岡 伶光

(指導教員 山本 拓弥、林 兵馬)

2025 年度 卒業研究最終論文

台風の弱体化を目的とした氷投下地点の推定及び考察

神戸大学附属中等教育学校 12 回生 6 年 3 組 35 番 宮岡 伶光

(指導教員 山本 拓弥、林 兵馬)

要旨

本研究は、氷の投下により台風の勢力を弱める「タイフーンショット構想」の実用化に向けて、勢力の強い台風が高確率で通過する地点を特定することを目的とし、データ解析およびシミュレーションを行った。対象データとして、気象庁が公表する過去 23 年間の台風データのうち、日本に上陸した台風（以下、上陸台風）に限定し、緯度・経度・中心気圧・最大風速の情報を用いた。まず、上陸台風の経路を Plotly により可視化した上で、緯度・経度を特徴量とした DBSCAN および K-means によるクラスタリングを実施した。さらに分析精度を高めるため、最大風速と中心気圧を加えた 4 変数を用いて、Gaussian Mixture Model (GMM) および Mean-Shift Clustering によるクラスタリングを行い、勢力の強い台風が通過しやすい地域を特定した。クラスタ数の決定には、K-means ではエルボー法およびシルエットスコア、GMM ではベイズ情報量規準 (BIC) を指標とした。また、ランダムフォレストにより特徴量の重要度を評価し、最も寄与の大きい特徴量に基づいて Gradient Boosting による予測モデルを構築し、将来的な通過地点をシミュレーションした。その結果を再度 Mean-Shift Clustering により分類することで、過去データと予測データに基づく通過範囲の比較を行った。これらの分析により、過去および将来において勢力の強い台風が通過しやすい地点は概ね一致することが確認された。最後に、令和元年の台風 19 号を事例とし、本研究で得られた通過範囲が氷の投下地点として適しているかを考察した。本研究はシミュレーションの精度など課題を残したものの、タイフーンショット計画実現に向けた可能性を示唆した。

台風の弱体化を目的とした氷投下地点の推定及び考察

Estimation and Evaluation of Ice Deployment Locations for Typhoon Weakening

宮岡 伶光
Remi Miyaoka

Abstract

This study explores the practical implementation of the “Typhoon Shot Project,” which aims to weaken typhoons by deploying ice. I analyzed 23 years of typhoon data from the Japan Meteorological Agency, focusing on typhoons that made landfall in Japan. Using latitude and longitude, we performed clustering with DBSCAN and K-means, then improved accuracy by incorporating central pressure and maximum wind speed. Clustering was further refined using Gaussian Mixture Models (GMM) and Mean-Shift Clustering. The optimal number of clusters was determined via the elbow method, silhouette score, and BIC. Feature importance was evaluated using Random Forest, and a predictive model was built with Gradient Boosting. I compared past and simulated future typhoon paths using clustering results. The analysis showed consistent high-risk areas for strong typhoons over time. Typhoon Hagibis (2019) was examined as a case study to assess ice deployment feasibility. While challenges remain, the study highlights the potential of this novel mitigation approach.

Keywords: typhoon, clustering analysis, regression modeling, statistics, python

目次

要旨	1
第 1 章 序論	1
第 1 節 はじめに	1
第 2 節 タイフーンショット構想とは.....	1
第 3 節 本研究の目的と意義.....	2
第 2 章 上陸台風のプロット	3
第 1 節 本調査の概要	3
第 2 節 調査目的	3
第 3 節 調査方法	3
第 4 節 調査結果	4
第 5 節 考察	4
第 3 章 2 種類の条件を特徴量としたクラスタリング分析	5
第 1 節 本調査の概要	5
第 2 節 調査目的	5
第 3 節 調査方法	5
第 4 節 調査結果	6
第 5 節 考察	8
第 4 章 4 種類の条件を特徴量としたクラスタリング分析	10
第 1 節 本調査の概要	10
第 2 節 調査目的	10
第 3 節 調査方法	10
第 4 節 調査結果	11
第 5 節 考察	14
第 5 章 機械学習を用いたシミュレーション分析	15
第 1 節 本調査の概要	15
第 2 節 調査目的	15
第 3 節 調査方法	15
第 4 節 調査結果	16
第 5 節 考察	18
第 6 章 結論	20
第 1 節 結論	20
第 2 節 今後の展望	20
謝辞	21
参考文献	22
付録	23

第1章 序論

第1節 はじめに

台風は強風や豪雨、高潮を引き起こし、多大な人的・経済的被害をもたらす自然災害の一つである。特に日本は台風の進路上に位置する国でありその被害を受けやすい。令和元年に発生した台風19号では死者90名、行方不明者9名、住家の全半壊等4008棟、住家浸水70341棟のきわめて甚大な被害が報告された[1]。また、気候変動に関する政府間パネル(IPCC)第四次報告書によると、温室効果ガスによる熱帯域の海面水温上昇によって台風の勢力が今後増す可能性が指摘されており[2]、今後台風の被害も増大していく可能性がある。このようなことを踏まえると、今後台風による被害を考慮するための新たな対策を考えることは必要不可欠である。しかし、新たな具体的な台風被害防止策は現状行われていないのが実情であり、台風への備えといった防災意識の向上の呼びかけにとどまっている[3]。

第2節 タイフーンショット構想とは

2022年、横浜国立大学は新たな台風被害防止策案として「タイフーンショット構想」を発表した。本項ではその構想の内容を示す[4]。

タイフーンショット構想は2050年を目標に台風の勢力を制御し、さらにその一部をエネルギーとして取り出す、台風を資源活用した持続可能な社会実現を目指す構想だ。現在具体的に発表されている台風の制御方法は以下の図の通りである。

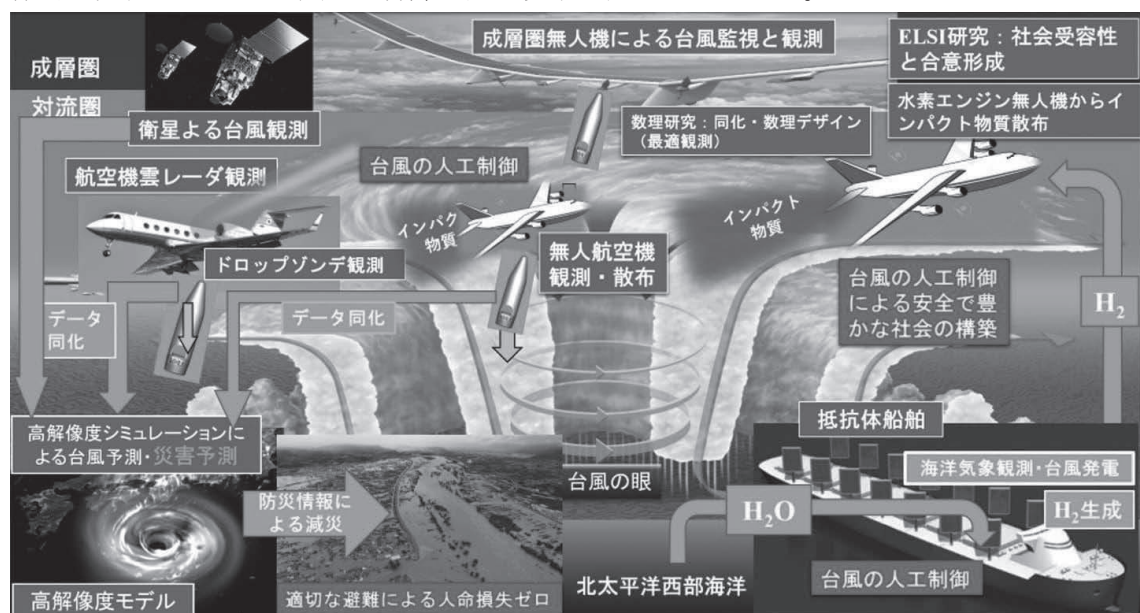


図1 タイフーンショット構想のイメージ図[4]

台風は太陽の熱によって暖められた海面から供給される水蒸気をエネルギー源としてお

り、この熱が台風の発達に大きく関係している[5]。本構想は、その台風のメカニズムを用いて、発生した台風の眼に無人航空機から氷またはドライアイスなどのインパクト物質を投下し台風の暖気核を冷やすことで、台風を弱体化させるという内容だ。これにより台風の制御が可能になり台風被害縮小が期待できる。

過去にこの構想と似た内容の実験は 1960 年代にアメリカで「ストームフューリー計画」と称して行われた[6]。航空機からハリケーンにヨウ化銀をまくこの実験は一時的に 10~30%の威力の減少が確認できたものの、ハリケーンの動きを予測しきることができず、大きな成果は得られなかった。この結果を踏まえてこの計画の実行は断念することになりその後研究も行われていない。タイフーンショット構想ではこのような過去を踏まえ、台風の正確な予測を行うためにコンピューターを用いたシミュレーション技術を駆使することを発表している。台風科学技術研究センター長である筆保教授はインタビューで「半世紀前にできなかったことは、コンピューターの中での影響評価です。いまでは数値シミュレーションというコンピューター上で台風を再現することができます。コンピューターの世界で発生させたバーチャル台風人間が介入した場合と、しなかった場合の影響を比較できるようになりました。」と述べており[7]、この構想実現にコンピューター技術の利用が不可欠であることがわかる。本研究では、データサイエンスを用いてタイフーンショット構想をテーマにした研究を行った。

第 3 節 本研究の目的と意義

本研究はタイフーンショット構想の台風制御における第一歩となる氷投下地点に焦点をあて、氷投下地点の条件となる勢力の強い台風が高確率で通過する地点を決定することを目的とした。これにより氷を投下する範囲がある程度絞られるため、タイフーンショット構想実現に大きくつながることが期待できる。

第2章 上陸台風のプロット

第1節 本調査の概要

気象庁から発表されている 2001 年から 2023 年の台風の観測データから上陸台風のデータを抽出し、そのデータの緯度及び経度から Python の Plotly を用いて地図上にプロットした。場所のみのプロットに加えて最大風速の大小で階級分けをしたプロットも行い、氷投下予想地点の決定への手がかりが見つからないか分析した。

第2節 調査目的

過去のデータから上陸台風がどのような経路を通りどの地点で最大風速が大きくなるのか傾向をつかむことを目的として、過去の上陸台風の場所を地図上にプロットし分析を行った。

第3節 調査方法

第1項 データの取得及び前処理

本研究では、気象庁の公式ウェブサイトから公開されている 2001 年から 2023 年に観測された台風のデータを利用した。このデータは台風が発生した日時から一日 4 回、その台風の緯度や経度、最大風速や中心気圧など様々な観点で観測しデータは Excel 形式で提供されている[8]。このデータから上陸台風のみを抽出し、そのファイルを Python のデータ分析ライブラリである Pandas を用いて読み込んだ。まず、`pandas.read_excel` 関数を用いてデータをデータフレーム形式に変換した。その後、以下のような前処理を行った。

まず、最大風速の値に基づいて、独自のカテゴリ「低」「中」「高」の風速階級を作成した。気象庁が発表している低圧部及び熱帯低気圧に該当する最大風速である台風（33KT 以下）は「低」、非常に強い台風、猛烈な台風と分類される台風（最大風速 85KT 以上）は「高」とし、それ以外の台風は「中」と階級づけた[9]。この階級分けにより、台風と熱帯低気圧、威力が普通の台風と非常に強い台風とを区別することができる。階級分けした風速階級は `apply` メソッドを用いて新しい列として追加した。

第2項 Plotly を用いた可視化

まず、緯度と経度のみを用いたシンプルなプロットを行った。このプロットでは台風の上陸地点を地図上に表示し、位置情報を直感的に把握できるようにした。次に、場所だけではない詳細な情報を得るために、前項で分類した風速階級に基づいて階級ごとにそれぞれ異なる色でプロットを行った。具体的には `color_discrete_map` を用いて風速階級の「低」「中」「高」の順番に「青」「黄」「赤」の色で表示した。これにより、最大風速の大小と地理的分布の関係性を把握できるようになった。

第4節 調査結果

Plotly でプロットした結果を以下の図 2-1 及び 2-2 に示した。

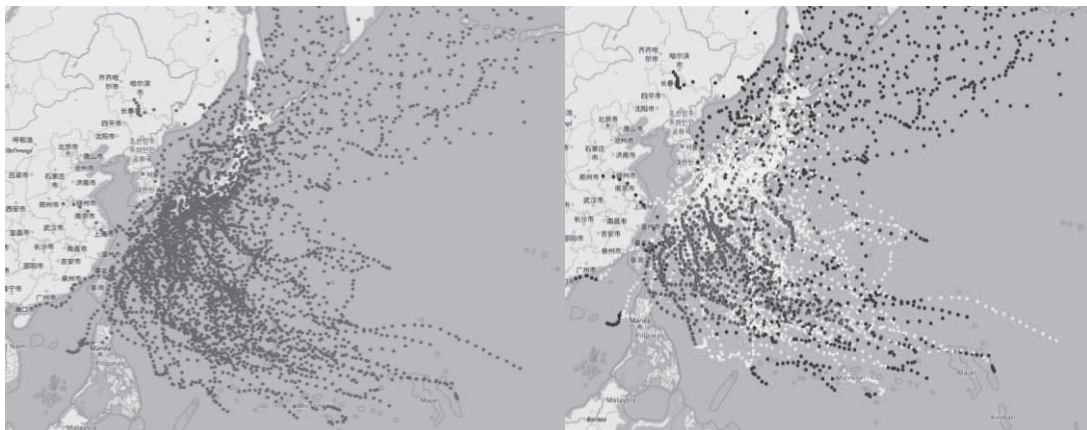


図 2-1 上陸台風プロット図

図 2-2 上陸台風プロット図（階級分けあり）

図 2-1 は緯度経度のみの上陸台風プロット図、図 2-2 は風速階級による階級分けを行った上陸台風プロット図である。多少のばらつきはあるものの図 2-1 から台風の進路は基本的に南西方向から日本の沖縄あたりを經由して北西方向に移動する曲線状であることがわかった。また、図 2-2 からは風速階級は互いに似たような緯度に存在することが多く、図 2-1 からわかった台風の経路と照らし合わせると、発生した台風は北緯 20～30 度あたりで勢力を最大にしてそこから最大風速を落とし勢力が弱まることが読み取れた。

第5節 考察

図 2-2 で階級ごとに同じような緯度帯に存在しているのは、台風が全体を通して同じような経路をたどっているからではないかと考えた。経路と同様に図 2-2 の分布から台風発生直後の熱帯低気圧から太平洋を横断している間に海からの熱をエネルギー源として勢力を増し、日本上陸後はその海の熱を受け取ることができないため勢力を落とすという台風の勢力の一連の流れがあるのではないかと考えた。この結果から、氷投下地点は最大風速が経路上で最大となる北緯 20～30 度あたりが最適であると考えた。しかし、この結果の図だけでは具体的な氷投下地点の決定にはつながらず、さらなる分析が必要であると考えられる。

第 3 章 2 種類の条件を特徴量としたクラスタリング分析

第 1 節 本調査の概要

本調査では、上陸台風のデータから緯度と経度の二つの条件を特徴量としたクラスタリング分析を行い、緯度と経度の条件から具体的な氷投下予想地点を決定することが可能かどうか考察した。

第 2 節 調査目的

緯度と経度の二つの条件でクラスタリング分析を行うことで上陸台風の地理的パターンをより詳細に把握し、第 2 章の結果とも照らし合わせて具体的な氷投下予想地点決定への手がかりを見つけることを目的とした。

第 3 節 調査方法

第 1 項 DBSCAN を用いたクラスタリング手法

DBSCAN とは密度ベースの空間クラスタリングアルゴリズムである。この解析手法は、高密度の領域をクラスターとして識別し低密度の領域をノイズとして扱うため、氷投下予想地点の決定につながると考え、この手法を用いた。DBSCAN の主なパラメータは最小サンプル数と呼ばれるクラスターを形成するために必要な最小のデータ数と eps と呼ばれる高密度であると定義するためのデータ間の距離の閾値である。最適なパラメータを用いて DBSCAN が実行できるように、それぞれのパラメータで最適化を行った。

まず、最小サンプル数の最適化するために、シルエットスコアを用いた分析を行った。シルエットスコアとはクラスタリングの信憑性を評価する指標の一つであり、1 であるほど信憑性が高いことを表す。この分析では eps の値を仮で 4 と設定し、最小サンプル数を 2 から十分大きな数である 300 まで変化させて、それぞれの値でシルエットスコアを計算し、最も高いスコアを示す最小サンプル数を最適な数とした。

次に、eps パラメータの最適値を決定するために、K-distance plot を使用した。この方法で、各データポイントの k 番目 (k=最小サンプル数) に近い隣接点までの距離を計算し、x 軸をデータポイント、y 軸を 4 番目に近い隣接点までの距離としてグラフで表した。このグラフを解釈する際、グラフのエルボーポイントと呼ばれる急激な変化点に着目した。エルボーポイントは、データ間の距離が急激に増加し始める点であり、この点以降のデータポイントはより低密度な領域に属していると考えられるため、このエルボーポイントを eps の最適値の候補とした。またこの eps の最適数が仮で設定した 4 という値からも大幅にずれていないか確認した。

最後に、最適化されたパラメータを用いて DBSCAN クラスタリングを実施し、分析を行った。

第2項 K-means を用いたクラスタリング手法

K-means とはデータポイントを k 個のクラスターに分割してクラスタリングする学習アルゴリズムである。この解析手法は、指定されたクラスター数から各クラスターの最適なセントロイド（クラスターの中心）を計算することができ、そのセントロイドの場所が具体的な氷投下地点特定につながるのではないかと考え、この手法を用いた。

指定するクラスター数を最適化するためにエルボー法とシルエットスコアを併用して用いた。エルボー法とはクラスター数とクラスター内の分散（以降 WCSS と呼ぶ）の関係から、最適なクラスター数を決定する手法である。クラスター数を 2 から 10 まで変化させ、各クラスターに対する WCSS を計算し、その変化をグラフで表した。クラスター数の増加に伴って WCSS は減少していくことが予想されるが、その WCSS の減少率が急激に低下した点をエルボーポイントとし、最適なクラスター数とした。シルエットスコアとは第1項で述べたように、クラスタリングの信憑性を評価する指標の一つである。各クラスター数に対してシルエットスコアを計算し、1 に最も近くなったクラスター数を最適とした。この 2 つの観点から最終的な最適なクラスター数を決定し、それを用いて K-means クラスタリングを実施し、分析を行った。用いたコードは以下のとおりである。

第4節 調査結果

第1項 DBSCAN 解析の結果

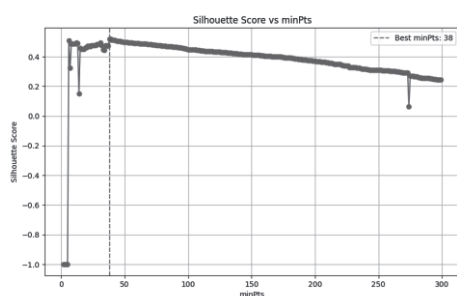


図3 シルエットスコア値の変化の推移

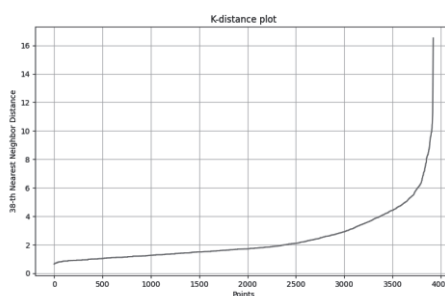


図4 K-distance plot を行った結果

図3は最小サンプル数の最適値を求めるために行ったシルエットスコアの推移である。図3から $\text{eps}=4$ として設定したときのシルエットスコアが最大になるのは最小サンプル数が 38 の時であることがわかる。また、最小サンプル数は 50 から 300 の間で減少傾向にあるため、300 より大きい値でシルエットスコアが最大値になることはないことが推測できるため、このデータにおける最適な最小サンプル数は 38 だと分かった。この結果を用いて K-distance plot を行った結果が図4である。図4からエルボーポイントの eps の値はおよそ 4.5 である。これは最初に設定した eps の値の 4 と大きな差はなく、最適な最小サン

プル数を変える必要はないと考えた。つまり、この 2 つの結果から最小サンプル数は 38、 $\text{eps}=4.5$ と設定するのが最適であると考えた。最適なパラメータを用いて行った DBSCAN の解析結果が以下の図 5 である。

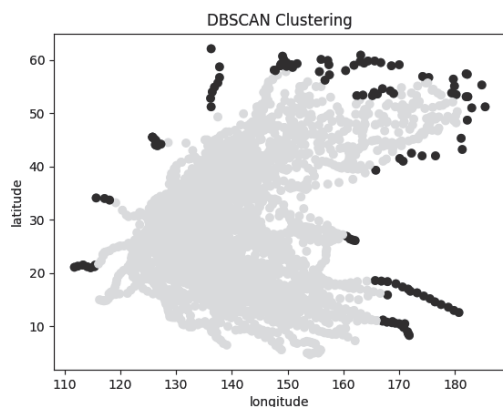


図 5 DBSCAN の解析結果

図 5 より大半の台風の経路地点が高密度な領域として分布されており、ノイズとされる低密度な領域を通る台風はほとんどないことがわかった。これは台風の経路において例外とされる異様なルートを通る台風がめったに存在していないことを示していると考えられる。つまり、氷投下予想地点もこの高密度の領域に投下すべきであることがいえる。しかし、図 5 で示されている高密度領域は範囲が広く、具体的な氷投下予想地点を定めることは難しいと考えた。

第 2 項 K-means 解析の結果

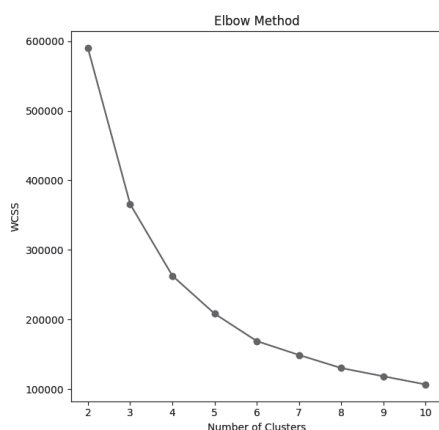


図 6-1 エルボー法の結果

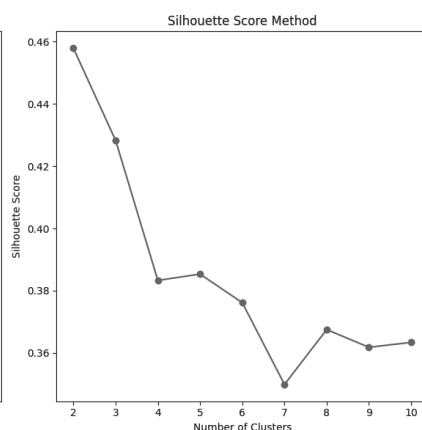


図 6-2 シルエットスコアの計算結果

図 6-1 からクラスター数を 3 のとき WCSS の減少率が一番変化していることがわかる。つまりクラスター数を 3 より大きくしても WCSS の減少は緩やかになり、クラスター数をそれ以上大きくする必要がないため、エルボー法の結果からはクラスター数 3 が最適であ

るといえる。しかし、図 6-2 ではクラスター数 2 のときにシークレットスコアが最大で 1 に一番近い値であり、エルボー法の結果と一致しない。図 6-2 を見るとクラスター数 2、3 の時のシークレットスコアの値は順に 0.46、0.43 であり、シークレットスコアの値に大きな差はないと考えたため、今回はクラスター数 3 が最適であると考えた。この結果を用いて K-means 解析を行った結果が以下の図 7 である。

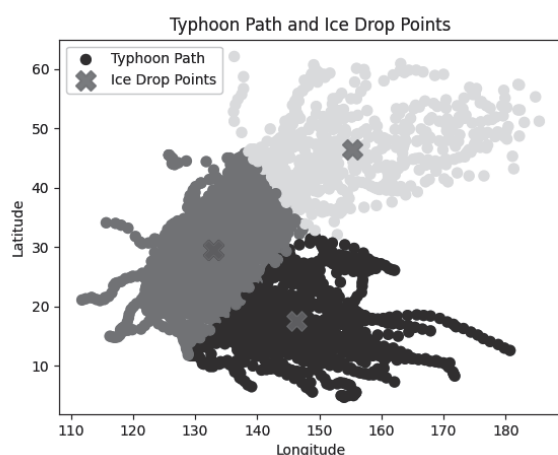


図 7 K-means の解析結果

上陸台風データは上の図 7 のように 3 つに区分けされることができ、それぞれにセントロイドが重複することなく存在している。この図 7 と図 2-2 の風速階級別のプロット図を比較すると中緯度に位置しているクラスターに風速階級が高位な台風が位置していることが多い傾向にあり、このクラスターのいずれかの地点で落とすのが良いのではないかと考えた。しかし、図 7 は緯度と経度のみを特徴量としているため、場所の条件のみでのクラスタリング解析であるためこの解析のみでは氷投下予想地点の決定は難しいと考えた。

第 5 節 考察

本節では、2 つの解析手法を用いた解析結果をふまえ、この解析方法の問題点を考察する。

まず DBSCAN の解析結果では、大半が高密度の領域に含まれており、氷地点を特定することが困難であった。これは 2001 年～2023 年の 23 年間の上陸台風のビッグデータを扱ったことにより、地図上のほとんどの地点に点が密集している状態であったため、DBSCAN が高密度だと認識してしまったことが要因だと考えられる。また、eps と最小サンプル数の最適化を行う際、最初に eps=4 と設定してそのうえで最適化を行ったため、この eps=4 という設定が正しいのかどうかの吟味も必要である。

次に K-means 法ではクラスタリングはできたものの、場所以外の条件との関係性がないため、具体的な氷投下地点の決定は困難であった。また、図 6-2 から採用したクラスター数 3 のシルエットスコアの値は 0.4 と低くそもそもクラスタリング結果自体に信憑性があるのか疑問が残った。図 6-1 のエルボー法でもクラスター数 3 で WCSS の減少率に変化が

生じたものの顕著な差ではなかったため、このクラスター数が最適といえるのかどうか不明である。

2つの解析方法に総合していえることは緯度と経度のみの特徴量では台風の発生に直接関係している条件が考慮されておらず、具体的な氷投下予想地点を決定するのが難しいということだ。クラスタリング結果は相関関係を示すものであり因果関係を示すものではないので、特定の地域でクラスターが形成されたからといって、最大風速や中心気圧などの台風の勢力に関係する条件が考慮されていない限り、その地域が必ずしも最適な氷投下予想地点であるといえない。そのため今後の研究では特徴量の種類を増やした多次元的な分析が求められることがいえる。

第4章 4種類の条件を特徴量としたクラスタリング分析

第1節 本調査の概要

本調査では上陸台風のデータから緯度や経度の条件に加えて最大風速と中心気圧も含んだ4種類の条件を特徴量としてクラスタリング分析を行い、クラスター同士で条件を比較することで、氷を投下するのに最適なクラスターを導き、具体的な氷投下予想地点を決定した。

第2節 調査目的

最大風速や中心気圧を含んだ4つの条件でクラスタリング分析を行うことで上陸台風の地理的分布と台風の勢力との関係性を把握し、具体的な氷投下予想地点を決定することを目的とした。

第3節 調査方法

第1項 GMMを用いたクラスタリング手法

GMM法とは複数の正規分布の混合モデルを用いてクラスタリングする確率的モデリング手法である。そのため、データの前処理として、scikit-learnのStandardScalerを用いて特徴量を標準化した。標準化によって各特徴量の平均が0、分散が1に統一されるため、異なるスケールの特徴量間の比較が可能になる。GMM法もK-meansと同様、クラスター数を指定する統計手法であるので、最適なクラスター数を適用するためにベイズ情報量基準(BIC)を用いた。BICとはモデルの複雑さとデータへの適合度のバランスを評価する指標であり、BIC値が最小になった時のクラスター数を最適なクラスター数として決定できる。1から9までクラスター数を変化させたときのそれぞれのBICを計算し、その変化をグラフに表した。最適なクラスター数を用いて、GMM法でのクラスタリング分析を行った。

クラスタリングを行った後、クラスターごとの最大風速や中心気圧の大きさを比較するために箱ひげ図を作った。これによってどのクラスターに位置する台風が一番勢力が強い台風なのかわかるようにした。

第2項 Mean-Shift Clusteringを用いたクラスタリング手法

Mean-Shift Clustering法とはデータ密度の高い領域を探す非パラメトリックな統計手法の一つである。GMM解析同様、データの前処理として、4つの特徴量をscikit-learnのStandardScalerを用いて標準化した。非パラメトリック手法であるため、クラスター数を事前に指定する必要はなく、データ密度の高い地点を探すことができるため今回この統計手法を用いた。また、第1項と同様に、クラスターごとの最大風速と中心気圧を箱ひげ図で比較した。なおMean-Shift Clustering結果で示す図に記載されている「Cluster Center」

とは各クラスターの中で最も高密度な場所を示している。用いたコードは以下のとおりである。

第4節 調査結果

第1項 GMM 解析の結果

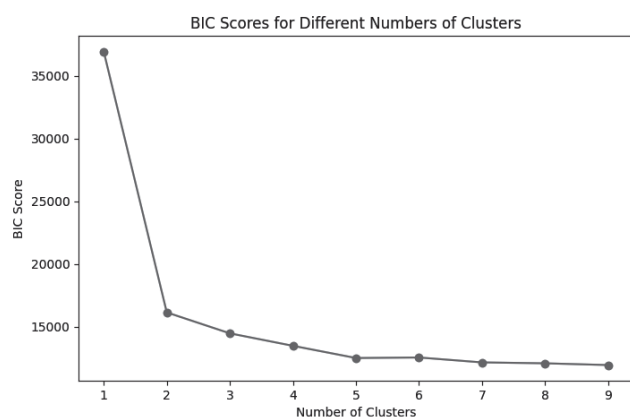


図8 クラスター数ごとの BIC 値の推移

図8より BIC 値はクラスター数5までは減少し、6で少し増加し、7でまた少し減少するといった変化が見られる。クラスター数5,7,8,9に関してはBIC値の値がほとんど変わっていないことから、その中で一番クラスターの数が少ない5を最適なクラスター数として決定した。これは、クラスター数の数が増えれば増えるほど1つのクラスターに属するデータの数が減り、クラスター同士の比較やクラスターの傾向をつかむことが困難になると考えたため、今回は5,7,8,9のなかで一番少ない5を最適なクラスター数とした。この結果を用いてクラスター数5でGMM解析を行った結果が以下の図9である。

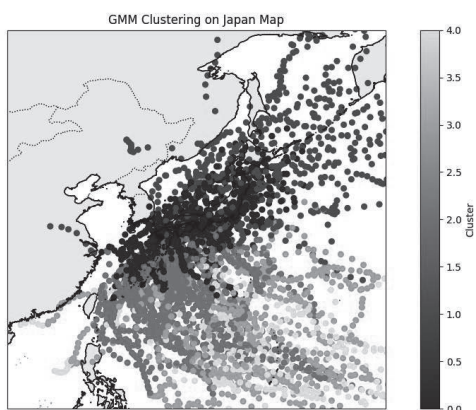


図9 GMM の解析結果

図9から、同緯度帯に位置している点同士が同じクラスターとして分類されているこ

とが比較的多い。このことから、同緯度帯であれば気温や海面水温に差はなく個体同士での台風の勢力にばらつきが少ないのではないかと仮説を立て、その仮説の元クラスターごとに最大風速と中心気圧の値をそれぞれ箱ひげ図で表し比較した。以下がそのグラフである。

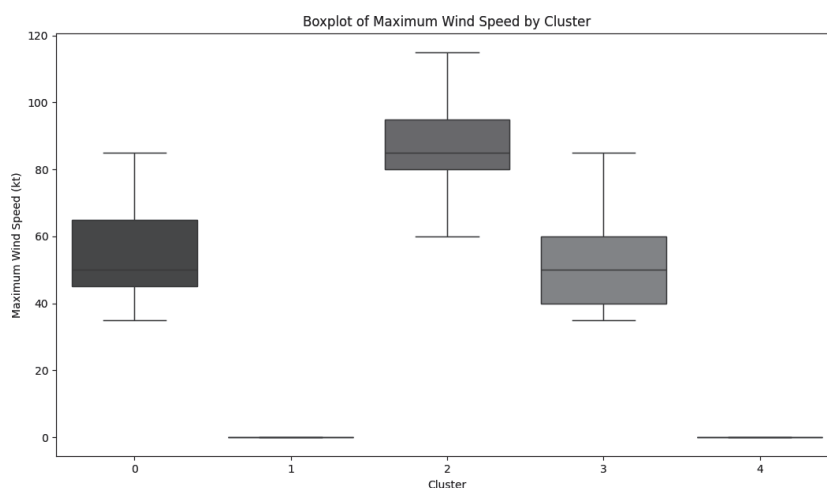


図 10-1 GMM クラスターごとの最大風速の箱ひげ図

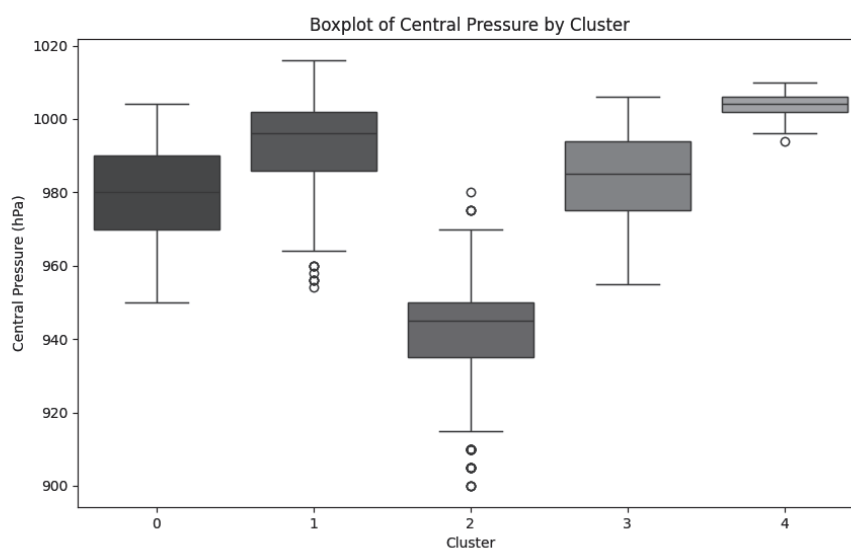


図 10-2 GMM クラスターごとの中心気圧の箱ひげ図

図 10-1 及び図 10-2 の縦軸から箱ひげ図の最大値と最小値の差は大きくなく（外れ値は除く）、仮説は立証されおり同クラスター内の個体同士は台風の勢力にばらつきが少ないことがいえる。図 10-1 から、最大風速が一番大きいクラスターはクラスター2であり、クラスター1 と 4 は風速がほとんどなく、クラスター0 と 3 は同じような風速であることが読み取れる。また、図 10-2 からは中心気圧が一番低いクラスターもクラスター2であり、クラスター0 と 3 が中位で同じような値を示し、クラスター1 と 4 が他と比べて気圧が高

いクラスターであることがわかった。このことから、クラスター2 に位置している台風は最大風速が大きく中心気圧が低い、勢力の一番強い台風であることがいえる。また同様に、クラスター1 と 4 が勢力の弱い台風（または熱帯低気圧）であり、クラスター0 と 3 は勢力が中位の台風のクラスターであるといえる。これは第 1 章第 2 節でも述べたように台風が海からの水蒸気に含まれる熱をエネルギー源としていて、上陸後はその勢力を落とすということを証明している。

このような結果から、台風の勢力が一番強いクラスター2 のいずれかの場所を氷投下地点として選ぶのが適当であると判断した。

第 2 項 Mean-Shift Clustering 解析の結果

Mean-Shift Clustering 解析を行った結果を以下の図 11 に示した。

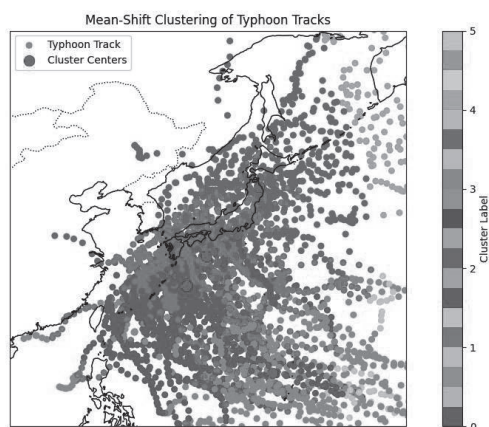


図 11 Mean-Shift Clustering の解析結果

図 11 から Mean-Shift Clustering のクラスター数は 6 であり、図 9 と同様に比較的同クラスターのデータ同士が同緯度帯に位置していることが読み取れる。以下のグラフはクラスターごとの最大風速及び中心気圧を箱ひげ図で示したものである。

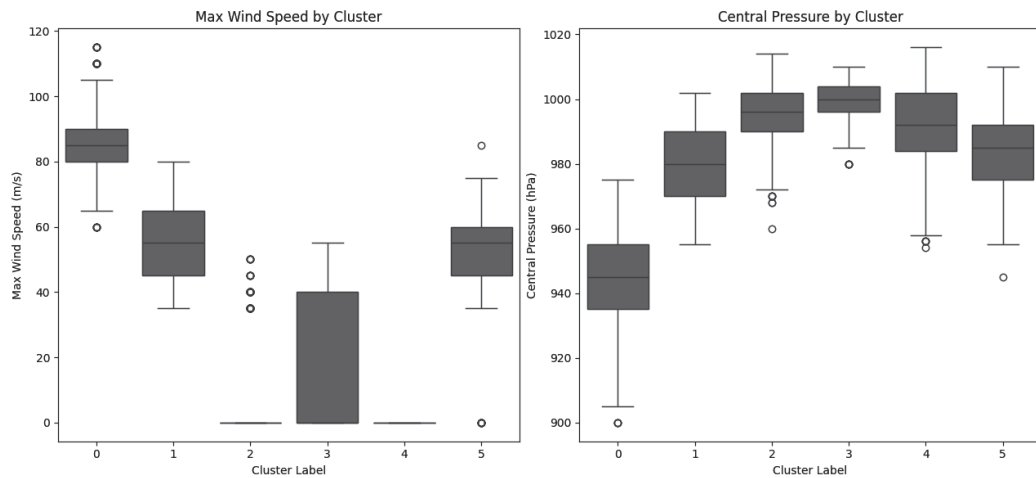


図 12 クラスターごとの箱ひげ図 (Mean-Shift Clustering)

図 12 より、クラスター0 に該当している台風が一番最大風速が強く、中心気圧が低い台風の勢力としては一番強い台風が分布していることがいえる。この結果と前項の結果を比較すると GMM で一番勢力が強いと示された図 9 のクラスター2 と図 11 のクラスター0 と位置関係が類似している。また箱ひげ図で示された四分位範囲も類似している。このことから勢力の強い台風が高確率で通過する点、つまり氷投下予想地点はこの範囲と決定した。また、クラスター0 の最も高密度な点を示す Cluster Center の東経 132 度北緯 27 度であった。

第 5 節 考察

前章の結果と比較すると、最大風速、中心気圧を加味した多次元的なクラスタリング分析により、それぞれのクラスターにどのような勢力のクラスターが存在するかが示すことができるようになった。具体的には、図 11 の勢力が一番強いクラスター0 では図 12 のグラフよりクラスターの最大風速の中央値が 80m/s を超えており、気象庁が発表する台風の分類分けでいう猛烈な台風に属していることがわかる[9]。つまり、今後この範囲に氷が投下されたとき、その効果が十分に発揮されることが期待できる。

第 5 章 機械学習を用いたシミュレーション分析

第 1 節 本調査の概要

前章までは過去のデータをもとに一番台風が強い威力を維持したまま通過する可能性の高い地点を見つけ、そこを今後も台風が通過すると仮定して調査を行った。本調査では過去のデータから今後どのような進路を通る台風があるのかをシミュレーションし、そこからも一つの氷投下地点の候補を見つけた。そのために、第 4 章で用いた 4 つの特微量の内どの条件が台風の氷投下地点を決定する上で最も重要度が高いのかを求め、その特微量に着目して過去の上陸データから機械学習を用いてシミュレーションを行い分析した。またシミュレーションの結果と第 4 章の結果を比較して、氷投下予想地点を決定した。

第 2 節 調査目的

本調査は過去のデータを用いた機械学習を用いた予測型のシミュレーションを実行し分析することで、未来における氷投下予想地点を決定することを目的とした。

第 3 節 調査方法

第 1 項 ランダムフォレストを用いた重要度の高い特微量の決定

第 4 章の解析に用いた 4 つの特微量の内、どの条件が台風の氷投下地点として重要度の高い条件なのかを求めるためにランダムフォレストを用いた。重要度の高い条件を求めることで、一番注目すべき指標がわかり、第 2 項に行うシミュレーションを効率的に実行することが可能になるため今回この手法を用いた。ランダムフォレストとは、機械学習におけるアンサンブル学習の一種で、複数の決定木を構築し予測を行う手法のことだ。今回はデータを訓練データとテストデータで 7:3 の割合で分割し、ランダムフォレスト分類器として 100 本の決定木を構築した。分類した訓練データを用いてモデルを学習させ、テストデータを用いて `accuracy_score` 関数でモデルの精度を評価した。今回目的変数は最大風速が 50KT の範囲の長径と設定し、このモデルの精度評価を 4 つの特微量それぞれで行い、学習したモデルから各特微量の重要度を抽出し、グラフにまとめた。用いたコードは以下のとおりである。

第 2 項 GBR を用いたシミュレーション

前項で求めた一番重要度の高い特微量に着目して、Gradient Boosting Regression (GBR) を用いてシミュレーションを行った。GBR とは、決定木を基本学習器とするアンサンブル学習アルゴリズムである。GBR は前の学習器の誤差を次の学習器が修正する形で学習を進め、台風のような複雑な非線形関係もとらえることができるため今回この手法を用いた。第 4 章の 4 つの特微量の内、重要度が高い特微量として選ばれなかった 3 つの条件

とそれに加えて台風の半径を特徴量に加えて、一番重要度の高い特徴量が予測される台風においてどのような分布になるのかを示した。シミュレーションの流れとしては前項のランダムフォレストと同じで、データを訓練用とテスト用に分割し、訓練用のデータからモデルを学習させ、テストデータを用いてモデルの評価を行った。今回のシミュレーションでは、決定木の本数を 100、モデルがどの程度パラメータを更新するかを表す学習率を 0.1、モデルの複雑さを制御するパラメータの一つである最大深さを 3 として、データセットの 80%を訓練用、20%をテストデータとして分割し、モデルの学習を行った。また今回のモデルの評価を行う際、MAE（平均絶対誤差）を用いた。MAE とは予測値と実際の値の差の平均値を表し、モデルの予測精度を直感的に理解しやすくした指標である。

この結果を可視化するために Cartopy ライブラリを用いて、シミュレーション結果をプロットした。この時すべての点はプロットせず、最大風速 64 ノット以上の気象庁から強い台風として分類される台風[9]のみを図で表した。これは氷を投下するのはそもそも甚大な被害を招く可能性のある強い台風のみであるため、今回はすべての点をプロットせず強い台風のみとした。各地点の色は予測された最大風速を表し、coolwarm カラーマップを使用して視覚化した。これにより、最大風速の地理的分布と氷投下候補地点の関係を直感的に把握することができる。

またシミュレーション後の結果を先ほどと同様に、Mean-Shift Clustering でクラスタリング分析を行った。この結果をカラーマップと比較してどのクラスターに今後勢力の強い台風が通過するのか決定した。

第 4 節 調査結果

第 1 項 ランダムフォレストを用いた結果

以下の図は、ランダムフォレストを用いた結果である。

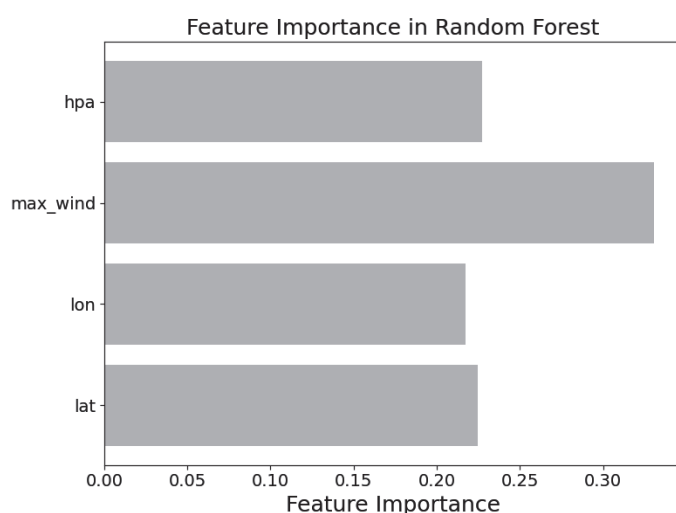


図 13 ランダムフォレストを用いた結果

図 13 から 4 つの特徴量の内、最も重要度が高いと判断されたのは最大風速であった。ま

た重要度が一番低かったのは経度であった。最大風速が最も重要であると判断されたのは、比較的大きい値と比較的小さい値との差が歴然としていて台風の威力を決定する上でこの差が重要であると判断されたからだと考えた。また、経度の重要度が一番低かったのは、今回のデータが日本に上陸する台風しか考慮していないため、経度の幅が限定されてこのような結果になったのではないかと考えた。

この結果から、台風の勢力の大小を決める上で一番重要な条件は最大風速であると分かり、この後の GBR を用いたシミュレーションでは最大風速に注目して行った。

第2項 GBR を用いたシミュレーション結果

前項の結果から、今回シミュレーションを行うのは最大風速の値であり、特徴量として用いるのは緯度、経度、中心気圧、台風の半径の4つとする。この特徴量と第3節第2項の条件でシミュレーションを行った結果シミュレーション結果は以下の図のようになった。

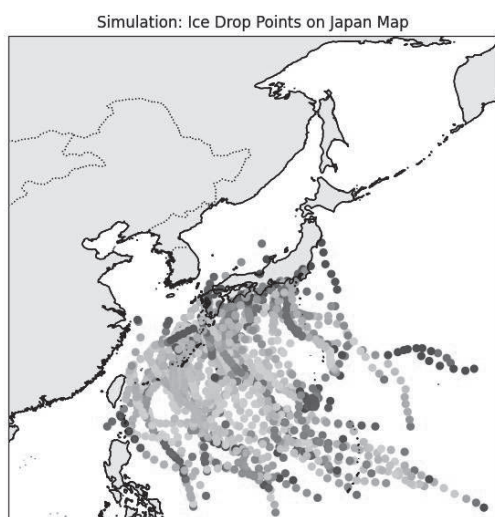


図 14-1 シミュレーション結果（勢力の強い台風のみ記載）

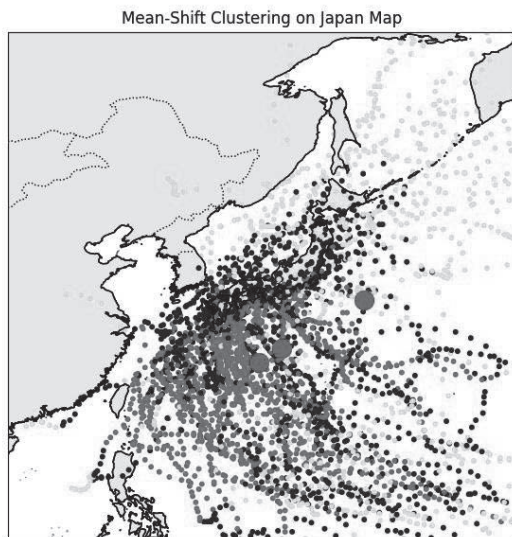


図 14-2 シミュレーション結果 (Mean-Shift Clustering)

シミュレーション結果は図 14 のようになり、またモデルの評価を行う際に用いた MAE の値は 2.20 となった。つまり、モデルの予測が平均して 2.20 ノットだけ実際の値から外れているということであり、データの範囲は 0 ノットから 95 ノットであることから考えると誤差は非常に小さいといえる。また、図 14-1 と 14-2 とを比較すると、最大風速の大きい勢力の強いクラスターが特定の 1 クラスターと非常に位置関係が類似していることから、このクラスターに今後勢力の強い台風が到達する可能性が高いことがいえる。なお、このクラスターの一番高密度な点である Cluster Center は北緯 27 度、東経 134 度であった。

第 5 節 考察

図 15 のシミュレーション結果の分布から、強い台風が通過すると予測されて分布されたほとんどの点が図 11 のクラスター 0 の位置に分布していることがわかり、このことから第 4 章で導いた氷投下地点は将来的な予測から見ても有力であると考えられる。しかし、今回のシミュレーションでは一つの特徴量についてのみしか考慮できておらず、シミュレーションの精度には課題が残った。

前章及び本章で示した結果から、台風は北緯 20 度から 30 度あたりで一番勢力を増すことがわかり、この地点は台風が北東へ進行方向を変える変曲点なのではないかと仮説を立てた。この仮説が成立したとき、変曲点で勢力が最大になる主な理由として考えたのは、台風発生時、そのあたりの海水温度が高くなり、台風のエネルギー源である海水の熱量が増え、雲がより発生しやすくなるからではないかということだ。

この仮説が正しいかどうか考察するため、具体的事例として今回、令和元年に上陸した台風 19 号を取り上げた。この台風は第 1 章第 1 節でも取り上げた通り、甚大な被害を及ぼした台風として知られている。以下は海洋研究開発機構 (JAMSTEC) が発表した平常時 (2019/11/7) と台風 19 号発生時 (2019/10/4) それぞれの平均海水温との差を示した図

である[10]。

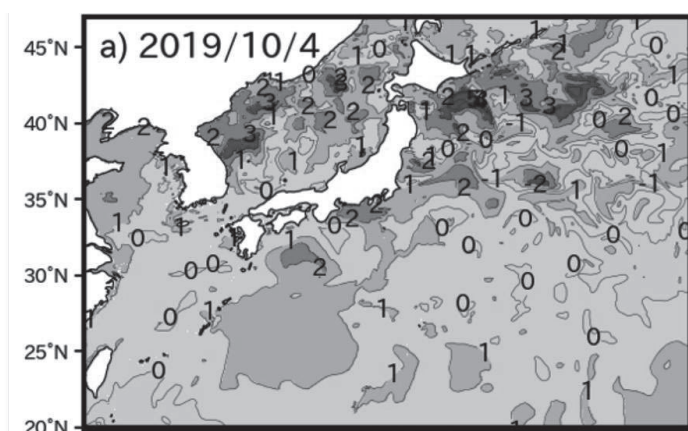


図 15-1 台風 19 号発生時の海水温の分布

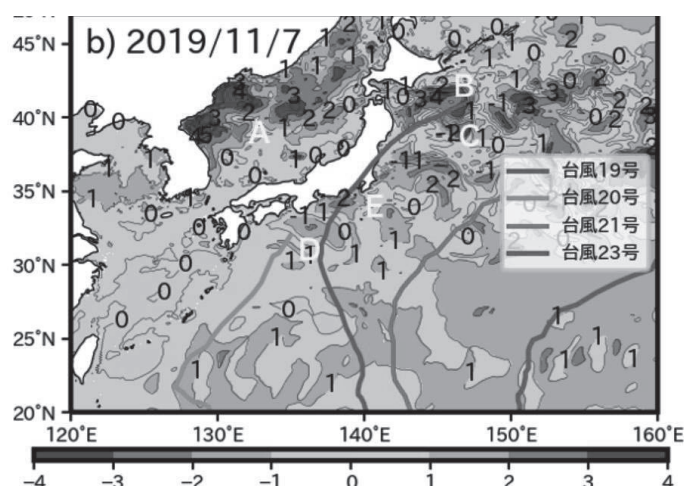


図 15-2 平常時の海水温の分布

図 15-1 を見ると、台風 19 号発生時点で広範囲で海水温が平均海水温よりも高いことがわかる。特に図 15-2 の台風 19 号の経路図の北東へと進路を変える変曲点周辺（地点 D 付近）では著しく平均海水温度より高いことがわかる。つまり、変曲点あたりで勢力を増すのは海水温度が関係しているということがこの件ではいえる。しかし、確実に変曲点と海水温度との関係性を示すことは難しく、検討の余地が見られる。今回は台風の経路を地点の連続性とみて分析を行ったが、今後は線形として分析していく必要があるといえる。

第 6 章 結論

第 1 節 結論

過去の上陸台風のデータから多種類のクラスタリング分析を行った結果、台風の経路は基本的に曲線系であり、北緯 10 度から 30 度の範囲で台風の勢力は最大になった。また図 11 のクラスター 0 が過去のデータにおいてもそこから予想されたシミュレーション結果においても勢力が最大になるクラスターであり、このクラスターを氷投下予想地点と決定した。これにより、氷投下地点の大まかな予測が可能となり、タイフーンショット構想の実現、つまり台風の弱体化の効果が期待できる。

第 2 節 今後の展望

本研究は過去の上陸台風のデータをもとに一番勢力の強い台風が高確率で通過する範囲を求めその範囲を氷投下が予想される有力候補と決定した。しかし、実際にタイフーンショット構想を実用化するにあたってはこの候補地点に必ずしも通るとは限らない。そのため、今後の研究では今回決定した候補地点を参考にしてもっと高精度なシミュレーションを行いこの候補地点に一体どれぐらいの確率で通過するのかを求め最適地点の吟味を行いたい。その際、第 5 章第 5 節でも示した通り台風を線形として扱い、海水温度との関係性を調べたい。また、高精度なシミュレーションを行った場合はシミュレーション結果の統計分析も行いたい。

また、今回の結果を踏まえてタイフーンショット構想実現に向けた氷の最適な投下地点だけではなく、投下する氷の量や無人航空機の飛行場所についても吟味したい。

謝辞

本研究を進めるにあたって、ご指導を頂いた本校教員山本拓弥先生、林兵馬先生に深く感謝いたします。

参考文献

- [1]国土交通省 | 令和元年台風 19 号による被害等 | 2019 |
https://www.mlit.go.jp/river/shinngikai_blog/shaseishin/kasenbunkakai/shouiinkai/kikouhendou_suigai/1/pdf/11_R1T19niyoruhigai.pdf?utm_source=chatgpt.com
- [2]環境省 | IPCC 第 4 次評価報告書 | 2007 |
<https://www.env.go.jp/earth/ipcc/4th/ar4syr.pdf>
- [3]気象庁 | 台風への備え | 2021 | https://www.jma-net.go.jp/miyako/topix/typhoon_miyako.html
- [4] 横浜国立大学 | Typhoonshot 構想 |
2022 | <https://typhoonshot.ynu.ac.jp/aboutts.html>
- [5]気象庁 | 台風の一生 | 2011 | <https://www.jma.go.jp/jma/kishou/known/typhoon/1-2.html>
- [6]National Oceanic & Atmospheric Administration | Project STORMFURY | 2019 |
https://www.aoml.noaa.gov/hrd/hrd_sub/sfury.html
- [7]MBS 毎日放送 | 2050 年には台風の脅威を恵みに変える！？ | 2024 |
<https://www.mbs.jp/news/column/scene/article/2024/07/101359.shtml>
- [8]気象庁 | 過去の台風資料 |
2017 | <https://www.data.jma.go.jp/yoho/typhoon/index.html>
- [9] 気象庁 | 台風情報 | 2021 |
https://www.data.jma.go.jp/multi/cyclone/cyclone_capl原因.html?lang=jp
- [10]海洋研究開発機構 | 最近の水温と海面高度の状況 (2019/10) 台風 19 号が近づく |
<https://www.jamstec.go.jp/aplinfokowatch/?p=9408>

付録

○階級分け無しのコード

```
import pandas as pd
import plotly.express as px

df = pd.read_excel('上陸台風ビッグデータ.xlsx')

fig = px.scatter_mapbox(
    df,
    lat='緯度',
    lon='経度',
    hover_data=['最大風速'],
    zoom=3,
    height=700,
    width=1500
)
fig.update_layout(mapbox_style='open-street-map')

fig.update_layout(margin={"r": 0, "t": 0, "l": 0, "b": 0})

fig.show()
```

○階級分けありのコード

```
import pandas as pd
import plotly.express as px

df = pd.read_excel('上陸台風ビッグデータ.xlsx')

def categorize_wind_speed(speed):
    if speed < 33:
        return '低'
    elif 33 <= speed < 85:
        return '中'
    else:
        return '高'

df['風速階級'] = df['max_wind'].apply(categorize_wind_speed)

color_discrete_map = {
```

```

        '低': 'blue',
        '中': 'yellow',
        '高': 'red'
    }

fig = px.scatter_mapbox(
    df,
    lat='lat',
    lon='lon',
    color='風速階級',
    hover_data=['max_wind'],
    color_discrete_map=color_discrete_map,
    zoom=3,
    height=700,
    width=1500
)

fig.update_layout(mapbox_style='open-street-map')

fig.update_layout(margin={"r": 0, "t": 0, "l": 0, "b": 0})

fig.show()

```

○最小サンプル数の最適化のコード

```

from sklearn.cluster import DBSCAN
from sklearn.metrics import silhouette_score
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_excel('上陸台風ビッグデータ.xlsx')
X = df[['lon', 'lat']].values

minPts_values = range(2, 300)
scores = []

for minPts in minPts_values:
    dbscan = DBSCAN(eps=4, min_samples=minPts)
    clusters = dbscan.fit_predict(X)
    if len(set(clusters)) > 1:

```

```

        score = silhouette_score(X, clusters)
        scores.append(score)
    else:
        scores.append(-1)

best_minPts = minPts_values[np.argmax(scores)]
best_score = max(scores)

plt.figure(figsize=(10, 6))
plt.plot(minPts_values, scores, marker='o')
plt.xlabel('minPts')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score vs minPts')
plt.grid()
plt.axvline(x=best_minPts, color='r', linestyle='--', label=f'Best minPts:
{best_minPts}')
plt.legend()
plt.show()

print(f'最適な minPts: {best_minPts}, Silhouette スコア: {best_score}')

```

○eps の最適化コード

```

import pandas as pd
import numpy as np
from sklearn.neighbors import NearestNeighbors
import matplotlib.pyplot as plt

file_path = '上陸台風ビッグデータ.xlsx'
df = pd.read_excel(file_path)

X = df[['lon', 'lat']].values

min_samples = 38
neighbors = NearestNeighbors(n_neighbors=min_samples)
neighbors_fit = neighbors.fit(X)
distances, indices = neighbors_fit.kneighbors(X)

distances = np.sort(distances[:, min_samples-1], axis=0)

plt.figure(figsize=(10, 6))

```

```
plt.plot(distances)
plt.title("K-distance plot")
plt.xlabel("Points")
plt.ylabel(f"{min_samples}-th Nearest Neighbor Distance")
plt.grid()
plt.show()
```

○DBSCAN のコード

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN

file_path = '上陸台風ビッグデータ.xlsx'
typhoon_data = pd.read_excel(file_path)
data = typhoon_data[['lat', 'lon']]

dbscan = DBSCAN(eps=4.5, min_samples=38)
labels = dbscan.fit_predict(data)

plt.scatter(data['lon'], data['lat'], c=labels, cmap='viridis')
plt.title('DBSCAN Clustering')
plt.xlabel('longitude')
plt.ylabel('latitude')
plt.show()
```

○エルボー法およびシルエットスコア算出のコード

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import StandardScaler

file_path = '上陸台風ビッグデータ.xlsx'
typhoon_data = pd.read_excel(file_path)
data = typhoon_data[['緯度', '経度']]

scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)
```

```

wcss = []
silhouette_scores = []
range_n_clusters = range(2, 11)

for n_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=n_clusters, init='k-means++', max_iter=300,
n_init=10, random_state=0)
    kmeans.fit(scaled_data)
    wcss.append(kmeans.inertia_)

    cluster_labels = kmeans.predict(scaled_data)
    silhouette_avg = silhouette_score(scaled_data, cluster_labels)
    silhouette_scores.append(silhouette_avg)

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(range_n_clusters, wcss, marker='o')
plt.title('Elbow Method (Scaled Data)')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')

plt.subplot(1, 2, 2)
plt.plot(range_n_clusters, silhouette_scores, marker='o')
plt.title('Silhouette Score Method (Scaled Data)')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette Score')

plt.tight_layout()
plt.show()

optimal_n_clusters = silhouette_scores.index(max(silhouette_scores)) + 2

kmeans = KMeans(n_clusters=optimal_n_clusters, init='k-means++', max_iter=300,
n_init=10, random_state=0)
kmeans.fit(scaled_data)
labels = kmeans.predict(scaled_data)

centers = kmeans.cluster_centers_

```

```

plt.scatter(data['longitude'], data['latitude'], c=labels, s=50, cmap='viridis',
label='Typhoon Path')
plt.scatter(centers[:, 1], centers[:, 0], c='red', s=200, alpha=0.75,
marker='X', label='Ice Drop Points')
plt.title('Typhoon Path and Ice Drop Points')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend()
plt.show()

```

○K-means のコード

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

file_path = '上陸台風ビッグデータ.xlsx'
typhoon_data = pd.read_excel(file_path)

latitude = typhoon_data['lat']
longitude = typhoon_data['lon']
data = typhoon_data[['lat', 'lon']]

kmeans = KMeans(n_clusters=3)
kmeans.fit(data)
labels = kmeans.predict(data)

centers = kmeans.cluster_centers_

plt.scatter(longitude, latitude, c=labels, s=50, cmap='viridis', label='Typhoon
Path')
plt.scatter(centers[:, 1], centers[:, 0], c='red', s=200, alpha=0.75,
marker='X', label='Ice Drop Points')
plt.title('Typhoon Path and Ice Drop Points')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend()
plt.show()

```

○BIC スコアを算出するコード

```

import pandas as pd
import numpy as np
from sklearn.mixture import GaussianMixture
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns

file_path = "上陸台風ビッグデータ.xlsx"
data = pd.read_excel(file_path)

columns_to_use = ['緯度', '経度', '最大風速', '中心気圧']
df = data[columns_to_use]

scaler = StandardScaler()
df_scaled = scaler.fit_transform(df)

n_components_range = range(1, 10)

bic_scores = []

for n_components in n_components_range:
    gmm = GaussianMixture(n_components=n_components, random_state=42)
    gmm.fit(df_scaled)
    bic_scores.append(gmm.bic(df_scaled))

plt.figure(figsize=(8, 5))
plt.plot(n_components_range, bic_scores, marker='o')
plt.title("BIC Scores for Different Numbers of Clusters")
plt.xlabel("Number of Clusters")
plt.ylabel("BIC Score")
plt.show()

delta_bic = np.diff(bic_scores)

plt.figure(figsize=(8, 5))
plt.plot(n_components_range[1:], delta_bic, marker='o')
plt.axhline(y=0, color='gray', linestyle='--')
plt.title("Delta BIC for Different Numbers of Clusters")
plt.xlabel("Number of Clusters")
plt.ylabel("Delta BIC (Change in BIC)")

```

```
plt.show()
```

```
optimal_clusters = n_components_range[np.argmax(delta_bic) + 1]  
print(f"最適なクラスター数は: {optimal_clusters}")
```

○GMM のコード

```
import pandas as pd  
import numpy as np  
from sklearn.mixture import GaussianMixture  
from sklearn.preprocessing import StandardScaler  
import matplotlib.pyplot as plt  
import cartopy.crs as ccrs  
import cartopy.feature as cfeature
```

```
file_path = "上陸台風ビッグデータ.xlsx"  
data = pd.read_excel(file_path)
```

```
columns_to_use = ['緯度', '経度', '最大風速', '中心気圧']  
df = data[columns_to_use]
```

```
scaler = StandardScaler()  
df_scaled = scaler.fit_transform(df)
```

```
n_clusters = 5  
gmm = GaussianMixture(n_components=n_clusters, random_state=42)  
labels = gmm.fit_predict(df_scaled)
```

```
data['Cluster'] = labels
```

```
fig, ax = plt.subplots(figsize=(10, 10), subplot_kw={'projection':  
ccrs.PlateCarree()}))
```

```
ax.set_extent([110, 160, 10, 60], crs=ccrs.PlateCarree())
```

```
ax.coastlines(resolution='10m')
```

```
ax.add_feature(cfeature.BORDERS, linestyle=':')  
ax.add_feature(cfeature.LAND, edgecolor='black')
```

```
scatter = ax.scatter(data['lon'], data['lat'], c=data['Cluster'], cmap='viridis',
```

```
s=30)
```

```
plt.colorbar(scatter, ax=ax, label='Cluster')
```

```
plt.title("GMM Clustering on Japan Map")
```

```
plt.xlabel("Longitude")
```

```
plt.ylabel("Latitude")
```

```
plt.show()
```

○最大風速の箱ひげ図

```
import pandas as pd
```

```
import numpy as np
```

```
from sklearn.mixture import GaussianMixture
```

```
from sklearn.preprocessing import StandardScaler
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
file_path = "上陸台風ビッグデータ.xlsx"
```

```
data = pd.read_excel(file_path)
```

```
columns_to_use = ['経度', '緯度', '最大風速', '中心気圧']
```

```
df = data[columns_to_use]
```

```
scaler = StandardScaler()
```

```
df_scaled = scaler.fit_transform(df)
```

```
n_clusters = 5
```

```
gmm = GaussianMixture(n_components=n_clusters, random_state=42)
```

```
labels = gmm.fit_predict(df_scaled)
```

```
data['Cluster'] = labels
```

```
plt.figure(figsize=(10, 6))
```

```
sns.boxplot(x='Cluster', y='max_wind', data=data, palette='viridis')
```

```
plt.title("Boxplot of Maximum Wind Speed by Cluster")
```

```
plt.xlabel("Cluster")
```

```
plt.ylabel("Maximum Wind Speed (kt)")
```

```
plt.show()
```

○中心気圧の箱ひげ図

```
import pandas as pd
import numpy as np
from sklearn.mixture import GaussianMixture
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns

file_path = "上陸台風ビッグデータ.xlsx"
data = pd.read_excel(file_path)

columns_to_use = ['lat', 'lon', 'max_wind', 'hpa']
df = data[columns_to_use]

scaler = StandardScaler()
df_scaled = scaler.fit_transform(df)

n_clusters = 5
gmm = GaussianMixture(n_components=n_clusters, random_state=42)
labels = gmm.fit_predict(df_scaled)

data['Cluster'] = labels

plt.figure(figsize=(10, 6))
sns.boxplot(x='Cluster', y='hpa', data=data, palette='viridis')
plt.title("Boxplot of Central Pressure by Cluster")
plt.xlabel("Cluster")
plt.ylabel("Central Pressure(hpa)")
plt.show()
```

○Mean-Shift Clustering のコード

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
from sklearn.cluster import MeanShift, estimate_bandwidth
from sklearn.preprocessing import StandardScaler

df = pd.read_excel('上陸台風ビッグデータ.xlsx')
```

```

features = df[['経度', '緯度', '最大風速', '中心気圧']]

scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)

bandwidth = 1.0

mean_shift = MeanShift(bandwidth=bandwidth, bin_seeding=True)
mean_shift.fit(features_scaled)

labels = mean_shift.labels_
cluster_centers = scaler.inverse_transform(mean_shift.cluster_centers_)

df['cluster_label'] = labels

plt.figure(figsize=(10, 8))
ax = plt.axes(projection=ccrs.PlateCarree())
ax.set_extent([110, 160, 10, 60], crs=ccrs.PlateCarree())

ax.add_feature(cfeature.COASTLINE)
ax.add_feature(cfeature.BORDERS, linestyle=':')

scatter = ax.scatter(df['lon'], df['lat'], c=labels, cmap='tab20', s=30,
transform=ccrs.PlateCarree(), label='Typhoon Track')

ax.scatter(cluster_centers[:, 1], cluster_centers[:, 0], c='red', s=100,
transform=ccrs.PlateCarree(), label='Cluster Centers')

plt.title('Mean-Shift Clustering of Typhoon Tracks')
plt.legend()
plt.colorbar(scatter, ax=ax, orientation='vertical', label='Cluster Label')
plt.show()

```

○ランダムフォレストのコード

```

import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

```

```

from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

X = df[['経度', '緯度', '最大風速', '中心気圧']]
y = df['50KT 長径']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

importances = rf.feature_importances_
feature_names = X.columns

plt.figure(figsize=(8, 6))
plt.barh(feature_names, importances, color='skyblue')
plt.xlabel('Feature Importance')
plt.title('Feature Importance in Random Forest')
plt.show()

```

○GBR のコード

```

import pandas as pd
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error

df = pd.read_excel('上陸台風ビッグデータ.xlsx')

X = df[['hpa', 'lat', 'lon', 'radius']]
y = df['max_wind']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

gbr = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1,

```

```

max_depth=3, random_state=42)
gbr.fit(X_train, y_train)

y_pred = gbr.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)

simulation_data = df.copy()
simulation_data['predicted_wind_speed'] = gbr.predict(simulation_data[['hpa',
'lat', 'lon', 'radius']])

threshold = 64
ice_drop_points = simulation_data[simulation_data['predicted_wind_speed'] >=
threshold]

import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature

fig, ax = plt.subplots(figsize=(10, 10), subplot_kw={'projection':
ccrs.PlateCarree()})

ax.set_extent([110, 160, 10, 60], crs=ccrs.PlateCarree())

ax.coastlines(resolution='10m')

ax.add_feature(cfeature.BORDERS, linestyle=':')
ax.add_feature(cfeature.LAND, edgecolor='black')

scatter = ax.scatter(ice_drop_points['lon'], ice_drop_points['lat'],
c=ice_drop_points['predicted_wind_speed'], cmap='coolwarm',
s=30)

plt.title('Simulation: Ice Drop Points on Japan Map')
plt.xlabel('Longitude')
plt.ylabel('Latitude')

plt.show()

```

○MAEを算出するコード

```
import pandas as pd
```

```

from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error

df = pd.read_excel('上陸台風ビッグデータ.xlsx')

X = df[['hpa', 'lat', 'lon', 'radius']]
y = df['max_wind']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

gbr = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, max_depth=3,
random_state=42)
gbr.fit(X_train, y_train)

y_pred = gbr.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
print(f'平均絶対誤差 (MAE): {mae:.2f}')

```