2023 年度

卒業論文

機械学習による競馬予想

指導教員

五島洋行　教授

法政大学

理工学部経営システム工学科

20X4115　中村心大

20X2136　宮崎嵐

学科名	経営システム工	学籍番号	20X4115 20X4136
申請者氏名		中村心大 宮崎嵐	
指導者教員		五島洋行	

論文要旨

論文題名	機械学習による競馬予想

競馬は馬に騎手を乗せ，馬同士を競い合わせる競争競技である．その予想は競馬が馬や騎手の能力，体調，戦略，そしてレース当日の天候やトラックの状態など，他にも多岐にわたる要因が複雑に絡み合う競技であるため，一般的に困難とされている．

その一方で，近年の競馬市場はデジタル化と情報化が進みつつある．競馬情報の提供や投票手段のオンライン化が進んだことで，レースの結果を分析する多くのデータが誰でも簡単に手に入るようになった． こうした状況の中，一般的に困難とされている競馬予想に統計学や機械学習を用いて予測を試みる研究が注目されている．

そこで本研究では，機械学習（LigthGBM）を用いて上位 3 位以内に入る馬を予測する．そして，その予測結果の精度を向上させてより良いモデルを作成し，回収率を計算する．最終的に，単勝・複勝回収率の 100%を超えることを目標とする．

結果として，穴馬に着目したモデルを構築することで，単勝・複勝回収率の 100%を大幅に超えることができた．

目次

1　はじめに

1.1　本研究の意義

競馬は馬に騎手を乗せ，馬同士を競い合わせる競争競技である．そして競馬は国や各地方自治体に よって経営される公営賭博でもある．我が国で行われている競馬は，農林水産省所管の特殊法人である日本中央競馬会（JRA）が主催する「中央競馬」と，地方公共団体が主催する「地方競馬」がある．本研究では日本中央競馬会（JRA）が主催する中央競馬を取り扱う．

中央競馬と地方競馬の違いは，いくつかある．競技場，コースの種類や長さ，規模などである．中央競馬と地方競馬で規模が違うため，競馬自体のレベルの差が出ている．潤沢な資金のある中央競馬では，競走馬の育成，騎手の育成や充実した設備に資金を費やせるため，自然と地方競馬に比べ，中央競馬のレベルが高くなってしまう．そして，最も差が出るのが賞金である．地方競馬で賞金 1 億を超えるレースはほとんどないが，中央競馬の重賞レースでは，賞金が 1 億円を超えることが普通である．このような違いから，強い騎手も中央競馬に集まってしまうため，中央競馬のレベルが高いのが現状の日本の競馬である．

日本競馬の売上高は近年右肩上がりである．コロナウイルス感染症の影響で多くの産業が衰退した．その中でも，サービス産業の活動の活発さを表す第 3 次産業活動指数で，コロナウイルス感染症の影響を最も多く受けたのが「生活娯楽関連サービス」あった．しかしながら、コロナの影響で娯楽業が衰退した中，娯楽業で唯一上昇したのが，「競輪・競馬等の競争場，競技場」である [1]．中央競馬を運営する日本中央競馬会（JRA）の売得金額は，2020 年に，約 2 兆 9000 億円，2021 年に，約 3 兆 900 億円，2022 年に，約 3 兆 2500 億円である[2]．

これらの日本競馬の好調の理由として，元々，電話・インターネット投票を積極的に導入していたことが挙げられる．コロナ禍前の 2019 年でも、中央競馬は約 7 割、地方競馬は約 8 割に達しており、2020 年の地方競馬では、その割合が約 9 割にまでに達している．このように，時間や環境に左右されず，コロナ禍においても，外出せず投票できることを多くのユーザーが認知したため，日本競馬は好調をキープできている．

中央競馬を運営する日本中央競馬会（JRA）は，我々が購入した勝馬投票券の売上げの一部を国庫納付金として納めている．国庫納付金の仕組みは，勝馬投票券の 75%を払戻金に，残りの 25%が控除される．その 25%のうち，10%を国庫に納付している．また，残りの 15%が JRA の運営に充てられ，これにより各事業年度において利益が生じた場合，その額の 50%が国庫に納付される．この国庫納付金は，75%が畜産復興に，25%が社会福祉に活用されている．つまり，中央競馬が盛んになるにつれ，国庫納付金が増え，国庫納付金を通じて，畜産復興や社会福祉に貢献しているのである．

競馬を楽しむ人は競馬の着順を予想し，お金を賭けることで楽しんでいる．この予想に

は様々なアプローチが存在する．各馬の過去の成績や過去のレース映像，調教映像を見て予想する方法や馬の情報を見ずにオッズ理論と呼ばれる，馬券ごとに決められているオッズを用いて予想する方法からオカルト的なものまで様々である．しかし，その予想は競馬が馬や騎手の能力，体調，戦略，そしてレース当日の天候やトラックの状態など，他にも多岐にわたる要因が複雑に絡み合う競技であるため，一般的に困難とされている．

その一方で，近年の競馬市場はデジタル化と情報化が進みつつある．競馬情報の提供や，前述したように投票手段のオンライン化が進んだことで，レースの結果を分析する多くのデータが誰でも簡単に手に入るようになった．さらに，過去の成績データや騎手・馬の状態，天候や競馬場の情報など，これまで専門家しか把握できなかった情報が一般にも広く提供されるようになった．

こうした状況の中，一般的に困難とされている競馬予想に統計学や機械学習を用いて予測を試みる研究が注目されている．これらの手法は多量のデータから有益な情報を引き出し，一定の的中率を達成している．これにより，従来は難解であった競馬の要因や相互関係をより複雑に分析し，予測モデルを構築することが可能になった．しかし，全ての状況・問題点を十分に対応することは難しく，予測の精度向上には未だ課題が残されている．

1.2　本研究の目的

そこで本研究では，機械学習（LightGBM）を用いて競馬の予想を行なう．特に，我々は競馬の予測において 1 位の馬を予測するのではなく，上位 3 位以内に入る馬を予測する．そして，その予測結果の精度を向上させてより良いモデルを作成し，回収率を計算する．そして，単勝・複勝回収率の 100%を超えることを目標とする．

1.3　論文構成

本論文は，全 8 章で構成している．
第 2 章では，本研究の基礎知識，理論について述べる．
第 3 章では，本研究の先行研究について述べる．
第 4 章では，本研究の研究内容について述べる．
第 5 章では，第 4 章の実験から得られた結果を述べる．
第 6 章では，第 4 章の実験から得られた結果を考察し，述べる．
第 7 章では，第 5 章の結果を踏まえ，行った研究内容，結果，考察について述べる．
第 8 章では，第 7 章の結果を踏まえ，行った研究内容，結果，考察について述べる．
第 9 章では，本研究の結論を述べる．

2　基礎理論

2.1　機械学習

2.1.1　機械学習

機械学習 (Machine Learning) は 1960 年頃から人工知能における課題研究として研究されており，データから規則性や判断基準を機械に学習させてそれに基づき未知のものを予想する技術である．基本的な定義としては，「アルゴリズムとして明示的な手順が与えられていないタスクに対して，そのタスクを遂行するモデルを学習データから生成する」である[3].

機械学習の流れとして，まず学習データが入力されると数値データやラベルの集合にデータ化され, 学習アルゴリズムに沿ってモデルが作成される. 次に入力されるデータにこのモデルを適用することで出力結果が得られる．これらの流れを図 1 に示す.

また，以下の表 1 は機械学習の分類についてのものである．表の「特徴量」とは対象の特徴が数値として表されたものであり，教師データとは，機械学習で学習するデータのうち例題と答えに関するデータのことである.

表 1： 機械学習の分類について

	特徴量	教師データ	正しい答え	活用法
教師あり学習	有り	○	有り	株価予想
教師なし学習	有り	×	無し	自動運転 AI
強化学習	有り（試行）	△（間接的）	答えはないが, 評価はあり	将棋 AI

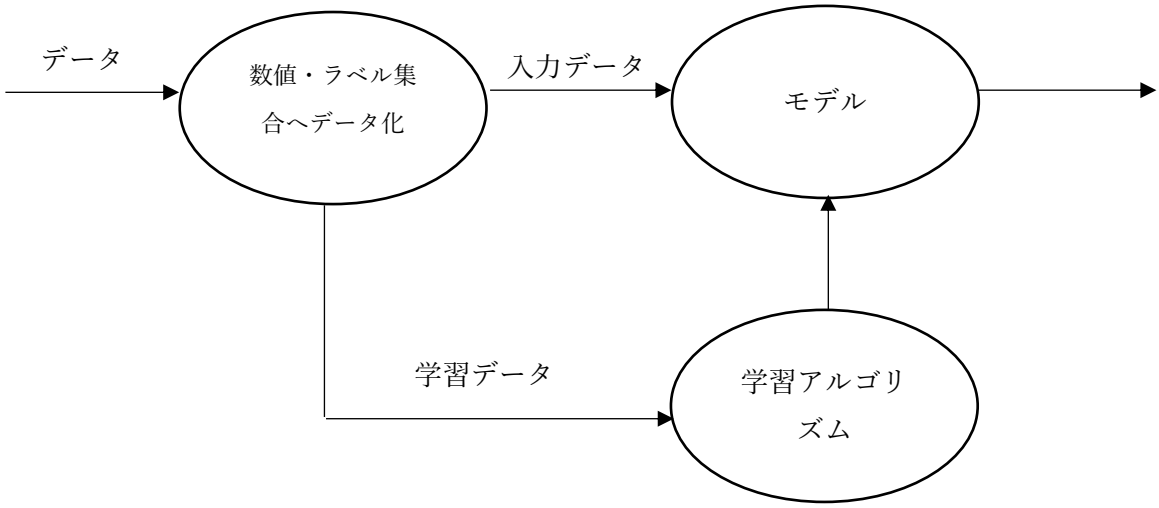


図 1 ：機械学習の基本的な流れ

2.1.2　教師あり学習

教師あり学習とは，既知となった過去の入力に関するデータと出力に関するデータを，機械学習アルゴリズムにあらかじめ与えることで，それらを正解データとして計算する手法である．また教師あり学習には「分類モデル」と「回帰モデル」がある．分類モデルは過去のデータから抽出された工学的特徴の特定のセットに基づいて，項目をクラス分けするものである[4]．また，回帰モデルはある二つの変数の関係を表す式のうち，統計的手法で推測された式により，一方の大きさが他方の大きさをどの程度説明できるかを分析するものである[5]．

2.1.3　教師なし学習

教師なし学習とは，教師あり学習とは異なり，特徴量のみのデータを使用して学習する手法である．この手法は，データの特徴を学習し，データをグループ分けすることが得意である教師なし学習では，入力データの規則性を学習することを目的とする．しかし，教師なし学習は正解のデータが提供されていないため，教師あり学習と比較して精度が低い傾向がある．本研究では，競馬の予測においては教師なし学習の利用は適さないと判断し，採用しないことにした．

2.1.4　強化学習

強化学習（Reinforcement Learning，RL）とは機械が試行錯誤して「価値を最大化するような行動」が何かを学習する機械学習の 1 つの学習手法である．この手法では，はじめに評価を設定し，特定の行動によって評価が与えられ，その結果をもとに繰り返し学習する．強化学習は動的な環境で動作する特性を持っている．ただし，競馬の過去のレース結果のデータから学習させたい本研究においては，強化学習が適していないと判断し，使用しないことにした．

2.2　スクレイピング

2.2.1　スクレイピング

スクレイピング(scraping)という単語は「こすること，削ること」等の意味である．このことから，Web スクレイピングとは，「Web サイトから任意の情報を抽出するコンピュータソフトウェア技術」のことである．ウェブ・クローラーあるいはウェブ・スパイダーと呼ばれることもある．

普段我々がネットサーフィンなどで使用しているGoogle，Yahoo!検索で表示される検索結果は検索エンジンによって決められている．この検索エンジンが，クローラというプログラムを使い，世界中の Web ページ情報を集めている．Web 上のサイトは互いにリンクし合って存在しているため，クローラはそのリンクを辿ることで各Webサイトの情報を集めている．クローラが集めた情報を検索エンジンがデータベースに記録し，その記録とユーザーが入力したキーワード等を照らし合わせて検索結果を表示している．クローラが情報を集めることをクローリングといい，収集した情報を記録することをインデックスという．

Web スクレイピングは，検索エンジンがクローリングからインデックスするまでの流れと密接な関係があり，クローリングは必要不可欠である．クローリングとスクレイピングについて混同してしまいそうであるが，クローリングは Web サイトから HTML や任意の情報を取得する技術・行為であり，スクレイピングとは，取得した HTML から任意の情報を抽出する技術・行為のことを指している．要するに，クローリングしたWebサイトから，ユーザーが欲する特定の情報を抜き出し，見やすいよう整形するのがWebスクレイピングである．使用例として，通販サイト等から商品データを取得し比較や，レストラン等の店舗の口コミを引用すること等が挙げられる．

2.2.2　スクレイピングの手順

まず，Web スクレイピングをするには，スクリプト言語と呼ばれる，プログラムの記述や実行を行いやすいプログラミング言語が使用される．スクリプト言語には，JavaScript や Ruby，PHP 等様々ある．

次に，スクレイピングする際の大まかな手順としては，(1)Web ページの取得，(2)プログラムを作成し指定した情報を取得する，(3)取得したデータを整形し保存または表示するといった 3 段階に分けることができる．

(1)Web ページの取得

まず，Web ページの取得についてである．単純に Web 上のデータを取得する際には，プログラムを自身で組むのであれば，その時に使用するプログラミング言語によってデータを取得するためのライブラリが用意されている場合がある．また，コードおよびネットワークに関する知識のある者であれば，ライブラリのインストール無しにデータを取得するプログラムを組むことも可能である．Webページの取得の際に，Web上にアップされている PNG ファイルやテキストデータを取得できる．

(2)プログラムを作成し指定した情報を取得する

スクレイピングを行う流れは，一般的に，情報の取得，抽出，保存といった流れであ

るが，Web スクレイピングとは，厳密に言えばその中の情報の抽出の行為のことを指す．Web 上から特定の情報を抽出するには，まず，HTML の構造を確認が必要である．

構造の確認が終われば，あとは，特定の要素を抽出するプログラムを考え，作成，保存し，実行すればスクレイピングの完了である．

(3)取得したデータを整形し，保存または表示

この手順は，スクレイピングを行う手順というよりは，スクレイピングを行った後の手順と述べたほうが正しいだろう．Web ページ上で特定の情報を取得すると言っても，その情報の量はユーザーごとで異なる．その日の日経平均株価や，自身の住んでいる地域の天災の情報であるなど，限定的な情報であれば，その日に一度プログラムを実行すればユーザーの目的は達成されたこととなる．

しかし，数日間に亘った株価の動きが知りたい場合や，複数の地域の天災の情報が欲しい場合には，プログラムを定期的に実行するように自動化，または，情報を一覧で表示できるように作成しなければならない．

2.2.3 Web スクレイピングのメリット・デメリット

Web スクレイピングのメリットと言えば，やはり，一度プログラムを組んでしまえば自動的に欲しい情報を取得，保存できることである．サイトから特定の情報を抽出できるため，通販サイトの特定の商品の価格等を取得し商品の比較ができ，定期的に情報を取得していくことで商品の価格の変化を記録することもできる．また，ニュースサイトをスクレイピングすればトップニュースや自分が関心を持っている分野のニュースの見出しを取得することができる．継続的，定期的に情報抽出を行う必要がある場合はコンピュータに Web スクレイピングさせることで自動化させておけば，自分の手で調べる手間が省けることになる．

次に Web スクレイピングのデメリットは，スクレイピングの行い方によっては罪に問われる可能性があることである．自動的に Web サイトへアクセスできるスクレイピング自体に違法性はない．しかし，特定のサイトへ大量アクセスをした場合，アクセス先のサイトがダウンしてしまうことがある．その場合，威力業務妨害に該当し罰せられることになる．また，スクレイピングする Web の記事は著作物に該当することが多い．そのため，著作権の侵害に該当する可能性がある．サイトによっては，スクレイピングで情報を取得すること自体が著作物の複製権侵害にあたる場合がある．また，スクレイピングによって取得したデータを自身の Web ページにそのまま掲載するなど，不特定多数がアクセス可能な環境に保存した場合，著作物の公衆送信権侵害に該当する．ただし，著作権法第 47 条の 7 項には，『著作物は，電子計算機による情報解析を行うことを目的とする場合には，必要と認められる限度において，記録媒体への記録又は翻案を行うことができる．ただし，情報解

析を行う者の用に供するために作成されたデータベースの著作物については，この限りでない.』とある．つまり，解析目的で，必要限度以内であれば，パソコン等に保存・複製しても良いという意味である．Web スクレイピングは効率的に大量のデータを取得できるとても利便性の高い技術だが，配慮すべき点には十分考慮し，適切かつ倫理的に使用することが求められる．

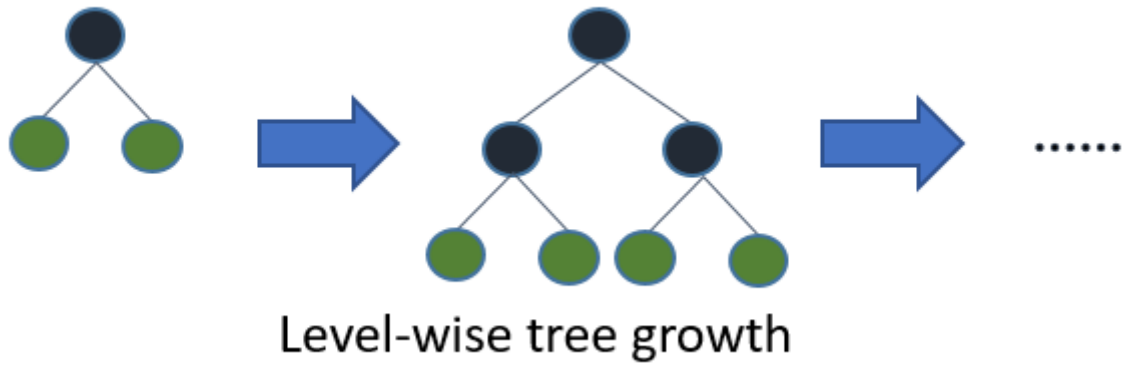
2.3　LightGBM

2.3.1　LightGBM

LightGBM（Light Gradient Boosting Machine）は，機械学習の分野において注目を集める勾配ブースティングフレームワークの一つである．

LightGBM の特徴の一つに，勾配ブースティングの過程において，「Level-leaf」ではなく，「Leaf-wise」を採用している．「Leaf-wise」を採用することで，他の「Level-wise」の決定木アルゴリズムよりも，大規模なデータセットに対しても高速な学習が可能である．「Level-wise」とは，左のノードから順に分割し，その断層の分割が終わり次第，次の断層に行き分割していく決定木の構築の仕方である．それに反し，「Leaf-wise」は，最も損失の小さくなるノードから分割していくことで，決定木の構築を行っている．このような決定木の構築の違いから，「Leaf-wise」の方が，短時間かつ高精度の学習を行うことができる．しかしながら，「Leaf-wise」の方が過学習するのが早いという注意点もある．

また，ヒストグラムベースの学習アルゴリズムを活用していることも特徴の一つとして挙げられる．このアプローチは，データの分割においてヒストグラムを使用しているため，決定木の分岐点を探す手間が短縮され，大規模なデータセットに対しても効率的に計算を行うことが可能である．これらのことから，他の勾配ブースティングフレームワークに比べ，高速かつ精度の高いアルゴリズムであることが言える．「Level-wise」と「Leaf-wise」の学習方法の違いを以下の図 2 で示す．



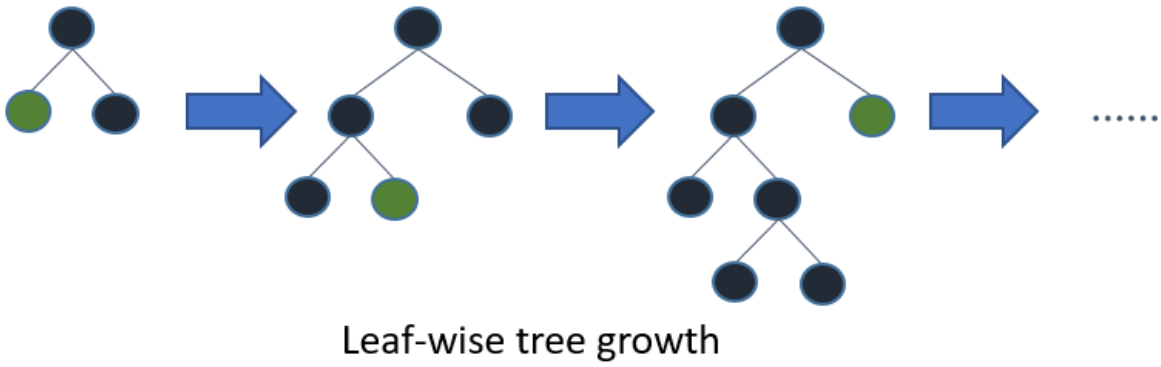


図 2：Level-wise と Leaf-wise
出典：Features — LightGBM 4.1.0.99 documentation

2.3.2　勾配ブースティング

勾配ブースティングとは，機械学習のアンサンブル学習手法の一環であり，複数の弱学習器を組み合わせて強力なモデルを構築する手法である．この手法では，各学習器が前の学習器が誤って分類したデータに焦点を当て，その誤差を最小化するように構築され，目的関数の勾配を利用して効率的かつ高精度な学習が行われる．

2.3.3　アンサンブル学習

アンサンブル学習は，複数の異なる学習モデルを組み合わせて，それぞれのモデルが持つ個別の予測を統合し，より強力で性能の高いモデルを構築する手法である．近年，Kaggle や KDD-Cups など国際的な機械学習競技会でも積極的に用いられて高い性能を発揮している．主なアプローチとして，バギング，ブースティング，スタッキングなどがある．

バギングは，学習データの一部を使用して学習し，最後に結合する方法である．それぞれが別個で計算できるため，並列処理が可能である．代表的な手法としてランダムフォレストがある．

ブースティングは，バギングとは対照的に，弱学習器が直列に構築され，一つ前の学習器が誤ったデータに重要視し，その誤差を修正するように次の学習器を構築する方法である．各学習器は前の学習器の誤差に基づいて重みづけされ，最終的な予測はこれらの学習器を組み合わせることで得られる．重みづけの方法によって，様々な手法がありアダブーストや勾配ブースティングがある．また，代表的な手法として Gradient Boosting や XGBoost，LightGBM がある．

スタッキングは，複数の学習モデルを層状に積み重ねる手法である．最初の層では複数のモデルが独立に学習し，その予測値を特徴量として，次の層のモデルが予測を行う．スタッキングは異なる種類のモデルを組み合わせてモデルを構築するため，精度の高い予測が期待できる手法である．

2.3.4 決定木

決定木は，機械学習の分野で幅広く活用されるモデルであり，説明変数の値によって枝分かれ状に細分化していき，最終的にいくつかのグループに分ける手法である．このモデルは可視化が容易であり，ツリーの構造を通じてデータの意味解釈がしやすいという特徴がある．そのほかにも，分類や回帰にも適用できるため，あらゆる問題に広く対応できるという特徴がある．

決定木には，いくつかのアルゴリズムがあるが CART(Classification and Regression Tree)と呼ばれるものが一般的である．CART は，説明変数の値に対して，条件を「Yes/No」とし，予測を行う方法である．パラメータの値によって，精度の低下や，過学習を起こすことがあるため，最適なパラメータに設定することが重要である．

決定木のプロセスは，すべての特徴量の分割を試し，不純度を計算する．その後，一番不純度の小さい条件を適用し，分割を行い，その分割した領域で，同様の手順を行っていく．不純度を表す代表的な指標に，ジニ不純度や情報エントロピーなどがある．前述した CART と呼ぼれるアルゴリズムでは，ジニ不純度を使用することが多い．

2.4 二値分類問題

二値分類問題は，与えられたデータから予測し，二つのクラスに分類する機械学習の典型的な問題である．学習手順は，データセットをトレーニングデータとテストデータに分割し，様々なアルゴリズムを用いてモデルをトレーニングデータで学習させる．代表的なアルゴリズムとしては，ロジスティック回帰，サポートベクターマシン，ランダムフォレスト，などが挙げられる．

モデル学習が終了した後，テストデータを用いてモデルの性能を評価する．その際，使用される主な評価指標として，正解率（Accuracy），適合率（Precision），再現率(Recall)，および F1 スコアなどの評価指標がある．その他にも，モデルの評価を可視化した ROC 曲線や，それに付随する AUC がある．

これらの評価指標は，以下の混同行列（表 2）を用いて説明する．

表 2：二値分類における混同行列

		モデルの予測	
		Positive	Negative
実際の結果	Positive	TP	FN
	Negative	FP	TN

本研究におけるそれぞれの説明については以下の表 3 にまとめる．

表 3：本研究における各説明

TP（True Positives）	3 着以内と予想した馬が 3 着だった件数と確率（正解）
FN（False Negatives）	3 着以内と予想した馬が着外だった件数と確率（不正解）
FP（False Positives）	着外と予想した馬が 3 着以内だった件数と確率（不正解）
TN（True Negatives）	着外と予想した馬が着外だった件数と確率（正解）

正解率（Accuracy）は，機械学習モデルの評価において最も基本的な指標の一つある．正解率は，全ての予測結果のうち，正しく分類されたサンプルの割合であり，以下の式である．

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

適合率（Precision）は，モデルが 3 着以内と予測した馬のうち，実際に 3 着以内であった馬の割合であり，以下の式である．この値が高いほど性能が高く，間違った分類をしていないことを意味する．

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

再現率(Recall）は，実際に 3 着以内であった馬のうち，モデルが正しく 3 着以内と予測した馬の割合であり，以下の式である．この値が高いほど性能が高く，実際に 3 着以内である馬を正しく予測できたことを意味する．

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

適合率と再現率はそれぞれがトレードオフの関係にあり，適合率の値が高くなると再現率の値が低くなり，再現率の値が高くなると適合率の値が低くなる．そこで，これらのバランスを取る指標が F 値である．

F 値（F1-score）は，正答率と再現率のバランスを取るために使用される指標であり，適合率と再現率の調和平均として計算され，以下の式である．

$$\text{F} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

ROC 曲線（Receiver Operating Characteristic Curve）と AUC（Area Under the Curve）は，異なる閾値（Threshold）における真陽性率と偽陽性率の関係を視覚的に表現し，モデルの総合的な性能を理解するのに寄与する．AUC は 1 に近いほど，モデルの性能が高いとされる．

ROC 曲線（Receiver Operating Characteristic Curve）は，機械学習の二値分類モデルの評価を可視化したものであり，横軸に偽陽性率（FPR），縦軸に陽性率（TPR）がプロットする．左上隅に近いほど理想の形状である．偽陽性率と真陽性率は以下の式で表される．

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

AUC（Area Under the Curve）は ROC 曲線の下の面積を計算したものである．AUC が 1 に近いほど，モデルの性能が高く，0.5 に近いほどモデルがランダムであることを示す．精度の目安は一般的に以下のように分類される．

0.9～1.0： 非常に高い
0.8～0.9： 高い
0.7～0.8： 多少高い
0.6～0.7： 高くない

0.5〜0.6： ランダムに近い

閾値は，モデルが陽性または陰性と判断する基準を指す．閾値を 0.5 と設定した場合，本研究においては，予測精度が 0.5 以上の馬を陽性（3着以内），0.5 未満の馬を陰性（3着以外）と分類とすることになる．異なる閾値では，陽性率と偽陽性率も変化するため，適切な閾値の選択が重要であるとされている．

2.6　ラベルエンコーディング

ラベルエンコーディングとは，機械学習モデルの学習に必要な前処理の一つであり，カテゴリデータを数値に変換する手法である．カテゴリデータとは，文字列やカテゴリ型のデータのことである．機械学習モデルは，数値データは扱うことでできるため，ラベルエンコーディングを行うことで，カテゴリデータを数値データとして扱うことができるようになる．

以下は，天気をラベルエンコーディングした例である．

晴：0
曇：1
小雨：2
雨：3
雪：4

このように，それぞれの天気に整数値を振り当てることで，機械学習モデルが天気のデータを扱えるようになる．

2.7　ハイパーパラメータチューニング

ハイパーパラメータとは機械学習モデルにおける出力結果を決定づける設定値や制限値のことである．以下に具体的なハイパーパラメータをいくつか以下の表 4 に示す．

表 4：ハイパーパラメータとその説明

num_iterations	ブースティングの反復回数，デフォルトは 100．値が大きいほどモデルは複雑になるが，過学習のリスクも増える．
learning_rate	学習率で，各ブースティングステップでの更新の大きさを決める．一般的な値は 0.1，0.05，0.01 など．
num_leaves	一つの木における葉の数．デフォルトは 31．値が大きいほどモデルは複雑になり，過学習しやすくなる．
tree_learner	並列学習の種類を指定．例えば serial，feature，data などがある．
max_depth	木の最大の深さ．過学習を避けるために使われる．デフォルトは-1 で，制限なしを意味する．
min_data_in_leaf	一つの葉が持つデータの最小数．過学習を防ぐために重要なパラメータ．デフォルトは 20．
metrics	モデルの評価指標を示すハイパーパラメータ．mae や rmse など．デフォルトは Objective と紐づく評価指標が設定される．
Objective	機械学習タスクがどんなタスクなのかを決定するハイパーパラメータ．回帰なのか 2 値分類なのか多値分類なのか．デフォルトは regression．
early_stopping_round	イテレーションの強制終了するハイパーパラメータ．num_iterations において early_stopping_round で設定した回数分 Validation データの精度が向上しなければ強制終了する．デフォルトは 0
verbosity	学習の経過の表示を制御するハイパーパラメータ．デフォルトは 1 で 1 回のイテレーションごとに経過が表示される．10 に設定すると 10 回ごとに表示される．−1 に設定すると表示されなくなる．

そして，ハイパーパラメータを調整してモデルの精度の最適化をはかっていくことをハイパーパラメータチューニングと呼ぶ．以下に具体的なハイパーパラメータチューニングをいくつか以下の表 5 に示す．

表 5：ハイパーパラメータチューニングとその説明

リッドサーチ	最適化をはかりたいハイパーパラメータ全ての組み合わせを調べる手法．ハイパーパラメータの多い手法に対しては組み合わせ数が膨大になってしまい現実的ではない．
ランダムサーチ	ハイパーパラメータの組み合わせをランダムに探索的に抽出し，指定した回数分繰り返して最適な組み合わせを探る．試行回数が少ないと最適なハイパーパラメータが見つかりづらい弱点もある．
ベイズ最適化	ベイズ理論を用いたパラメータ最適化手法で，過去のハイパーパラメータの組み合わせの解に基づいて，筋の良さそうな組み合わせの周りを探索していく手法である．

3　先行・類似研究

3.1　機械学習による競馬予想の研究

我々の研究テーマである機械学習による競馬予想は実際にも行われており，日本中央競馬会 (JRA)が提供している「JRA-VAN」や競馬情報サイトの netkeiba.com にて提供されている「的中型人工知能」などがある．またイギリスでも研究が進められており，勝者と敗者を区別するデータに依存した分類モデルが提案されており，イギリス競馬のデータを使用し実証実験も行われている[6]．これらの研究の他にも，機械学習による競馬予想の研究はイベントや個人のブログなどでは盛んに行われている．

3.2　「Favorite-Longshot-Bias（穴馬バイアス)」に関する研究

世界各国の競馬で「各馬の勝つ確率」に「その馬のオッズ」に応じた値（＝期待払戻率）を比較すると，本命馬[1]は値が大きくなり，穴馬[2]は値が小さくなる傾向がある．このような穴馬の過剰な選好を 「Favorite-Longshot-Bias（穴馬バイアス)」と呼ばれており，この現象について数多くの研究がされている[7]．それらの研究の中で，小幡・太宰(2014)の研究に注目する[8]．この研究では，一般の競馬ファンが過剰に大穴馬券を購入するために，その馬券が当たる本来の確率からすると割高なオッズになっており，逆に本命サイドの確率が相対的に高い馬券は割安になっており，そのバイアスは馬券が当たる確率が低いものほど大きくなっていることを実証している．具体的には，馬券の客観確率と主観確率を推計し，オッズが高い馬券ほど，客観確率に対して主観確率が大きくなっていることを実証している．ここでは，配当額が高く，ギャンブル性も高いことから，こうした穴馬バイアスが大きいものとして三連単を挙げており，その支持率を主観確率としていた．一方で，単勝は，そうしたギャンブル性が低く，競馬の知識が豊富な投票者が購入層の中心であるため，穴馬バイアスは小さいとし，その支持率から求めた三連単の確率を客観確率としている．

[1] 本命馬：レースに出走される馬の中で最も強いと思われる馬

[2] 穴馬：能力がわからず人気していない馬

4　研究内容

4.1　開発環境

4.1.1　コードエディタ

Visual Studio Code と JupyterLab を使用.

4.1.2　使用言語・パッケージ

プログラムは Python 3 を使用．パッケージは Python を使用.

4.1.3　ライブラリ・モジュール

（1）requests

HTTP リクエストを簡単に行うためのライブラリ．Web ページの取得などに使用される.

（2）BeautifulSoup (from bs4)

HTML や XML からデータを取り出すためのスクレイピング用のライブラリ．HTML の構造を解析し，データを抽出するのに役立つ.

（3）time

時間に関する機能を提供する標準ライブラリ．スリープ処理に使用されることがある.

（4）LabelEncoder (from sklearn.preprocessing)

カテゴリカルなデータを数値に変換するためのモジュール.

（5）StandardScaler (from sklearn.preprocessing)

データの標準化を行うためのモジュール．特徴量のスケールを揃えるのに使用される.

（6）numpy

数値計算を行うためのライブラリ．主に行列演算や数学的な処理に利用される.

（7）scipy.stats

統計関連の機能を提供する SciPy ライブラリの一部．統計テストや確率分布の関数などが含まれている.

（8）sys

Python のインタプリタや実行環境に関する情報や操作を提供する標準ライブラリ.

（9）roc_curve, roc_auc_score (from sklearn.metrics)

受信者動作特性 (ROC) 曲線や ROC AUC スコアの計算に使用されるモジュール.

（10）matplotlib.pyplot

グラフ描画のためのライブラリ．データの可視化に利用されます.

（11）datetime

日時に関するデータ型や操作を提供する標準ライブラリ.

（12）ast

抽象構文木 (Abstract Syntax Tree) を操作するためのモジュール.

（13）re

正規表現操作を行うための標準ライブラリ.

（14）statistics
統計関連の機能を提供する標準ライブラリ.
（15）optuna.integration.lightgbm (as lgb)
Optuna というハイパーパラメータ最適化フレームワークと LightGBM を統合するためのモジュール.

4.2　使用データ

本研究では，競馬情報サイト「netkeiba.com」[9]のデータベースから，Web スクレイピングによりデータの収集を行う．2016 年～2023 年 11 月 23 日までの中央競馬における全競馬場の全レース出馬情報と払い戻し情報を Web スクレイピングにより csv ファイルとして取得した．取得したデータ数を表 6 に示す.

表 6：取得したデータ数

出馬情報	払い戻し情報	合計
371212（レース数）×26（カラム数） =96515112	27303（レース数）× 7 （カラム数） =191121	96706233

出馬情報のカラムは「race_id」「馬」「騎手」「馬番」「走破時間」「オッズ」「通過順」「着順」「体重」「体重変化」「性」「齢」「斤量」「上がり」「人気」「レース名」「日付」「開催」「クラス」「芝・ダート」「距離」「回り」「馬場」「天気」「場 id」「場名」の 26 個である.
払い戻し情報のカラムは「単勝」「複勝」「馬連」「ワイド」「馬単」「三連複」「三連単」の 7 個である.

「race_id」は「西暦」,「競馬場番号（=場 id)」,「開催何回目か」,「開催何日目か」,「何レース目か」を表していて，レースごとに異なっている.「競馬場番号（=場 id)」については以下の表 7 に示す.

表 7：競馬場番号（=場 id）

01	02	03	04	05	06	07	08	09	10
札幌	函館	福島	新潟	東京	中山	中京	京都	阪神	小倉

4.2.1　出馬データ

実際に競馬情報サイト「netkeiba.com」のデータベースの央競馬における全競馬場の全レース出馬情報の 1 レース分を図 3 に示す．また，その情報を Web スクレイピングして取得した csv ファイルの 1 レース分を図 4 に示す．

結果/払戻　掲示板

着順	枠番	馬番	馬名	性齢	斤量	騎手	タイム	着差	タイム指数	通過	上り	単勝	人気	馬体重	調教タイム	厩舎コメント	備考	調教師	馬主	賞金(万円)
1	5	5	サトミノキラリ	牡2	55	横山武史	1:09.5		**	2-2	34.3	1.2	1	452(-4)				[東] 鈴木伸尋	田代洋己	550.0
2	8	8	ベアゴーゴー	牝2	55	浜中俊	1:09.5	クビ	**	1-1	34.5	4.1	2	454(+2)				[東] 杉浦宏昭	熊木浩	220.0
3	6	6	ハピアーザンエバー	牡2	55	藤岡佑介	1:10.0	2.1/2	**	5-4	34.5	59.9	6	438(-6)				[西] 羽月友彦	田畑利彦	140.0
4	4	4	デビルシズカチャン	牝2	55	ルメール	1:10.2	1.1/2	**	3-3	34.9	16.6	3	450(+2)				[西] 武幸四郎	カカムーチョレーシング	83.0
5	1	1	ウィスピースノー	牝2	55	吉田隼人	1:10.3	1/2	**	8-8	34.5	23.9	5	434(-10)				[西] 今野貞一	水上ふじ子	55.0
6	2	2	ロードスタウト	牡2	55	鮫島克駿	1:10.7	2.1/2	**	5-6	35.1	61.8	7	454(-6)				[西] 中村直也	ロードホースクラブ	
7	3	3	コミックガール	牝2	53	佐々木大	1:10.9	1.1/4	**	5-6	35.3	18.8	4	404(-2)				[東] 上原佑紀	古賀禎彦	
8	7	7	ヒポカンポ	牝2	52	小林勝太	1:11.9	6	**	3-4	36.6	251.3	8	394(-2)				[東] 伊藤圭三	OUMA	

図 3 ：1 レース分の出馬情報

	race_id	馬	騎手	馬番	走破時...	オッズ	通過順	着順	体重
1	202301010101	サトミノキラリ	横山武史	5	1:09.5	1.2	2-2	1	452
2	202301010101	ベアゴーゴー	浜中俊	8	1:09.5	4.1	1-1	2	454
3	202301010101	ハピアーザ...	藤岡佑介	6	1:10.0	59.9	5-4	3	438
4	202301010101	デビルシズ...	ルメール	4	1:10.2	16.6	3-3	4	450
5	202301010101	ウィスピース...	吉田隼人	1	1:10.3	23.9	8-8	5	434
6	202301010101	ロードスタウト	鮫島克駿	2	1:10.7	61.8	5-6	6	454
7	202301010101	コミックガール	佐々木大	3	1:10.9	18.8	5-6	7	404
8	202301010101	ヒポカンポ	小林勝太	7	1:11.9	251.3	3-4	8	394

体重変...	性	齢	斤量	上がり	人気	レース名	日付	開催
-4	牡	2	55	34.3	1	2歳未勝利	2023/07/22	1回札幌1日目
2	牝	2	55	34.5	2	2歳未勝利	2023/07/22	1回札幌1日目
-6	牡	2	55	34.5	6	2歳未勝利	2023/07/22	1回札幌1日目
2	牝	2	55	34.9	3	2歳未勝利	2023/07/22	1回札幌1日目
-10	牝	2	55	34.5	5	2歳未勝利	2023/07/22	1回札幌1日目
-6	牡	2	55	35.1	7	2歳未勝利	2023/07/22	1回札幌1日目
-2	牝	2	53	35.3	4	2歳未勝利	2023/07/22	1回札幌1日目
-2	牝	2	52	36.6	8	2歳未勝利	2023/07/22	1回札幌1日目

クラス	芝・ダート	距離	回り	馬場	天気	場id	場名
2歳未勝利	芝	1200	右	良	晴	0	札幌
2歳未勝利	芝	1200	右	良	晴	0	札幌
2歳未勝利	芝	1200	右	良	晴	0	札幌
2歳未勝利	芝	1200	右	良	晴	0	札幌
2歳未勝利	芝	1200	右	良	晴	0	札幌
2歳未勝利	芝	1200	右	良	晴	0	札幌
2歳未勝利	芝	1200	右	良	晴	0	札幌
2歳未勝利	芝	1200	右	良	晴	0	札幌

図 4 ：Web スクレイピングして取得した csv ファイルの 1 レース分

4.2.2　払戻しデータ

実際に競馬情報サイト「netkeiba.com」のデータベースの央競馬における全競馬場の全レース払い戻し情報の 1 レース分を図 5 に示す．また，その情報を Web スクレイピングして取得した csv ファイルの 1 レース分を図 6 に示す．

払い戻し

単勝	5	120	1
複勝	5 8 6	100 110 310	1 2 6
馬連	5 - 8	170	1

ワイド	5 - 8 5 - 6 6 - 8	120 600 880	1 8 12
馬単	5 → 8	210	1
三連複	5 - 6 - 8	1,120	4
三連単	5 → 8 → 6	2,680	9

図 5　1 レース分の払い戻し情報

	Column1	Column2
1	202301010101	[['5', '120'], ['5', '100', '8', '110', '6', '310'], ['5 - 8', '170'], ['5 - 8', '120', '5 - 6', '600', '6 - 8', '880'], ['5 → 8', '210'], ['5 - 6 - 8', '1,120'], ['5 → 8 → 6', '2,680']]

図 6 ：Web スクレイピングして取得した csv ファイルの 1 レース分

4.3　データの前処理

4.3.1　データの数値化

以下のカラムを数値化させた．

（1）クラス

“クラスに以下の文字が含まれていたら”という条件で数値化した．

‘G1’：10, ‘G2’：9, ‘G3’：8, ‘(L)’：7, ‘オープン’：7, ‘3 勝’：6, ‘1600’：6,‘2 勝’：5, ‘1000’：5, ‘1 勝：4, ‘500’：4, ‘新馬’：3, ‘未勝利’：1, ‘障害’：0

（2）走破時間

走破時間は「1:31.5」のようなフォーマットになっているため，「:」を取り除いて秒数に変換した．「1:31.5」であれば「91.5」秒になる．

（3）通過順

通過順は「10-8-5」のようになっているので，ハイフンを取り除いて割って出した．「10-8-5」であれば，「7.666…」となる．

（4）性別

性別は牡馬：0，牝馬：1，セン馬：2 としている．

（5）芝・ダート

芝・ダートは，芝：0，ダート：1，障害：2 としている．

（6）回り

回りは右：0，左：1，芝：2，直：2 としている．障害レースの場合は「芝」と表現される．

（7）馬場状態

馬場状態は良：0，稍：1，重：2，不：3 としている．

（8）天気

天気は晴：0，曇：1，小：2，雨：3，雪：4 としている．

（9）日付

年，月，日を取り出し，新しい「日付」カラムを作成した．

4.3.2　エンコーディング

学習に使わないデータはラベルエンコーディングした．実際にエンコーディングするのは「’馬’, ‘騎手’, ‘レース名’, ‘開催’, ‘場名’」である．

4.3.3　直近 5 レースのデータの情報を追加

予測の精度を上げるためには，予測するためのデータ量を増やす必要がある．そこで直近 5 レース分のデータを追加した．「馬番」「騎手」「斤量」「オッズ」「体重」「体重変化」「上がり」「通過順」「着順」「距離」「クラス」「走破時間」「芝・ダート」「天気」「馬場」「日付差」[3]「距離差」[4]のデータを直近 5 レース分追加した．

4.3.4　騎手の勝率のデータを追加

「騎手」ごとに勝率を計算して，「騎手の勝率」カラムを作成した．これにより騎手の過去の実績を特徴量として取り込んだ．

4.3.5　斤量の平均値のデータを追加

過去 5 レースの斤量の平均値を計算して，「平均斤量」カラムを作成した．これにより馬の過去の斤量の傾向を特徴量として取り込んだ．

以下の表 8 に前処理を行ったデータの csv ファイルから，本研究で用いるカラムとその説明についてまとめたものを示す．

---

[3] 日付差：レース間隔

[4] 距離差：前走からの距離変化

表 8：本研究で用いるカラムと説明

race_id	上記で説明
馬	馬名
騎手（1，2，3，4，5）	騎手名
馬番（1，2，3，4，5）	各レースにおける各場につけられる番号
走破時間（1，2，3，4，5）	スタートからゴールまでのタイム
オッズ（1，2，3，4，5）	その馬のオッズ
通過順（1，2，3，4，5）	各コーナーを通過した順位
着順（1，2，3，4，5）	ゴールした際の順位
体重（1，2，3，4，5）	馬の体重
体重変化（1，2，3，4，5）	全レースからの体重変化
性	馬の性別
齢	馬の年齢
斤量（1，2，3，4，5）	馬がレースで背負うおもり
上がり（1，2，3，4，5）	残り 600m からゴールまでのタイム
人気	馬の人気
レース名	各レースの名前
日付（1，2，3，4，5）	各レースの開催日
開催	開催における詳細
クラス（1，2，3，4，5）	馬の年齢と収得賞金の額によるレースの格付け
芝・ダート（1，2，3，4，5）	馬が走る地面の状態（芝生・砂地）
距離（1，2，3，4，5）	馬が走る距離
回り	馬が走るルート
馬場（1，2，3，4，5）	馬が走る地面の湿潤度合
天気（1，2，3，4，5）	天気
場 id	上記で説明（表４）
場名	開催される競馬場
距離差（1，2，3，4,）	前走からの距離変化
日付差（1，2，3，4,）	レースの間隔
平均斤量	上記で説明（3.3.5）
騎手の勝率	上記で説明（3.3.4）

（(1，2，3，4，5)，(1，2，3，4,）はそのカラムの直近 5 レースを追加する意味．例えば，日付（1，2，3，4，5）なら日付１，日付２，…，日付５のように五つのカラムになる．）

4.4　モデルの構築

本研究では，回収率を向上させたいため，馬券に絡む可能性が高い３着以内の馬を予想する．ただ，順位を目的変数には用いない．なぜなら，馬券に絡まない 4 着以下の馬の順位を当てることができたとしても，回収率には関係ないため，我々の研究目的にはそぐわない．そこで，３着以内の馬を True，４着以下を False とした二値分類問題を解くことにする．

今回使用するアルゴリズムは LightGBM である．データセットを訓練データとテストデータに分割し，訓練データだけを使用し，モデルの学習を行う．分割の割合としては，訓練データを７割，テストデータを３割とする．

使用する特徴量は，表８に示したカラムから，「着順」，「オッズ」，「人気」，「上がり」，「走破時間」，「通過順」を除いたものとする．また，ターゲットについては，予測対象である「着順」とする．

着順に関しては，着順はレースの結果であり，その馬の速さを表す指標ではない．例えば，クラスの高いＧ１で低い着順の馬よりも，クラスの低い未勝利戦などで，好成績を残している馬の方が速いとみなされてしまう．そのため，特徴量から着順を除く．人気，オッズに関しては，予想結果に影響を与えすぎるため，特徴量から除く．あがり，走破時間，通過順に関しては，予測時には使えないデータのため，特徴量から除く．

その後，テストデータを使用し，モデルの評価を行う．また，各特徴量の重要度を取得する．

4.5　モデルの評価

モデルの評価については，正解率（Accuracy），適合率（Precision），再現率(Recall)，F1 スコア，AUC の評価指標を使用する．混同行列やぞれぞれの評価を以下の表９に示す．

表 9：本研究のモデルの混同行列と各評価指標の値

		モデルの予測	
		Positive	Negative
実際の結果	True	11.79	10.53
	False	12.37	65.31

正解率	77.10%
適合率	48.80%
再現率	52.82%
F1 スコア	0.51
AUC	0.80

また，各特徴量の重要度を以下の図 7 で示す．

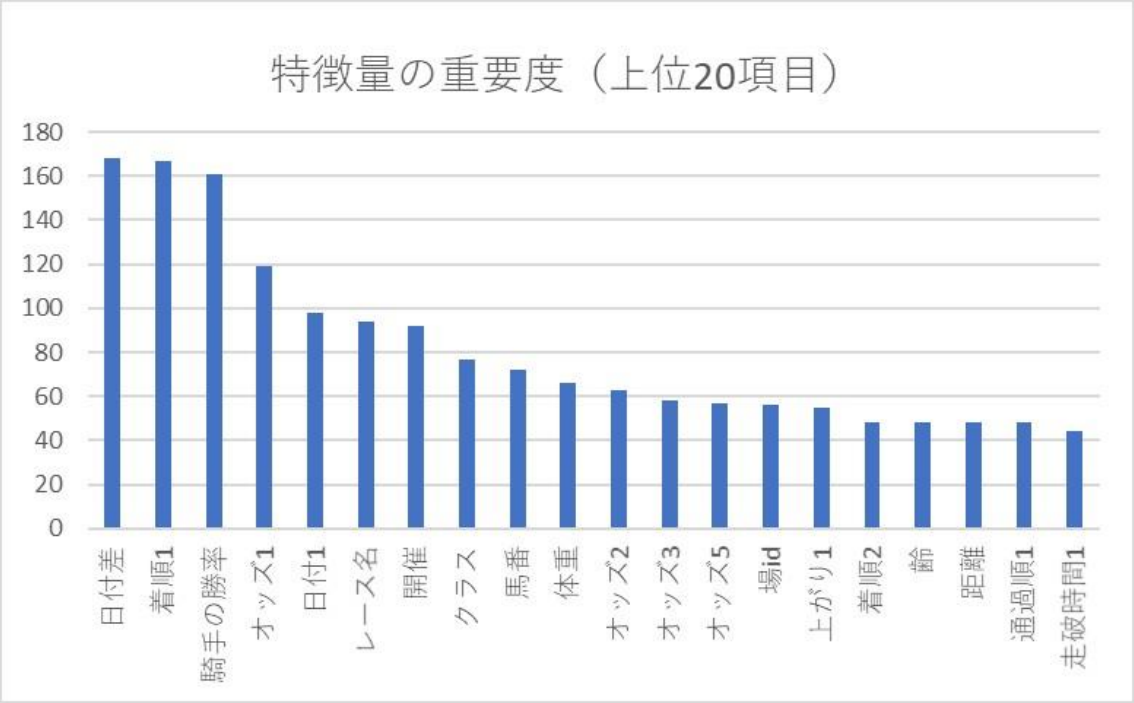


図 7：各特徴量の重要度（上位２０項目）

4.6　回収率の計算

回収率の計算において，まず初めに 2016 年から 2023 年 10 月 23 日までの払戻しデータ（図４）の読み込みを行う．その後，予測結果が設定した閾値以上の確率で予測された馬を抽出し，賭ける馬の決定を行う．賭ける馬が単勝，複勝した場合，それぞれの回収金額を，払戻しデータを基に計算する．回収金額を投資金額で割ることで回収率を計算する．投資金額は，予測精度に関係なく，一定とする．

$$回収率 = \frac{回収金額}{投資金額} \times 100$$

5　結果

二値分類の結果と払戻し情報を用いた回収率の結果は以下の表 10 のようになった.

表 10：各閾値に対する回収率と Betting Horses

閾値	0.5	0.6	0.7	0.8
単勝回収率	71.79%	82.10%	99.31%	142.94%
複勝回収率	69.24%	77.03%	92.28%	127.91%
Betting Horses	18699	13875	8929	4610

6　考察

閾値を変えることで，回収率が高くなり，目標としていた回収率 100%を超えることができた．しかしながら，閾値 0.7 以下では，回収率が 100%を超えていない．各閾値で回収率 100%を超えるためにも，まずモデルの予測精度の向上を行っていきたい.

7　研究内容 2

そこで，上述の実験を踏まえ，以下の実験を行う.

7.1　提案手法

回収率の向上を目的とし，ハイパーパラメータチューニングを行い，回収率の計算を行う．投資金額は，予測精度に関係なく，一定とする.

7.2　結果

ハイパーパラメータチューニング後の混同行列，ぞれぞれの評価や回収率を以下の表 11, 12 に示す.

表 11：パラメータチューニング後の混同行列と各評価指標の値

		モデルの予測	
		Positive	Negative
実際の結果	True	16.29	5.98
	False	20.08	57.65

正解率	73.94%
適合率	44.79%
再現率	73.15%
F1 スコア	0.56
AUC	0.82

表 12：パラメータチューニング後の各閾値に対する回収率と Betting Horses

閾値	0.5	0.6	0.7	0.8
単勝回収率	71.79%	82.10%	99.31%	124.88%
複勝回収率	69.24%	77.03%	92.28%	112.77%
Betting Horses	18699	13875	8929	5557

7.3　考察

回収率の向上を目的として，パラメータチューニングを行い，予測精度を向上することができた．しかしながら，モデルの予測精度は向上したが，回収率は下がる結果となった．回収率が下がった理由として，Betting Horses[5]が増えたことにより投資金額が高くなったこと，予測精度の高い馬のオッズが低いことの二つが挙げられる．このことから，回収率の向上するためには，モデルの予測精度の向上よりも，予測精度やオッズを考慮し，賭け方の工夫が必要だと考えられる．予測精度やオッズを考慮した賭け方については，今後の課題としたい．

[5] Betting Horses：それぞれの閾値を超える馬の数

8　研究内容 3

そこで，上記の結果を踏まえ，以下の研究を行う．

8.1　提案手法

以下の研究では，穴馬に着目し，研究を行う．上記までの研究では，3 着以内に入る馬を”1”とし，4着以降を”0”とし，二値分類問題として予測を行った．しかしながら，我々が 3 着以内と予測した馬の多くは，人気があり，オッズが低いことから，利益が低く，回収率の向上にはつながらなかった．そこで，穴馬に着目し，オッズという条件をつけ，二値分類問題を解くことにする．

具体的には，オッズが 15 倍以上で 3 着以内の馬を”1”，オッズが 15 倍未満かつ 3 着以内の馬，4 着以降の馬を”0”とする．また，使用するアルゴリズムは上記同様の，LightGBM である．

8.2　結果

8.1 で提案したモデルでの，混同行列，それぞれの評価や回収率を以下の表 13，14 に示す．

表 13：穴馬に着目したモデルの混同行列と各評価指標の値

		モデルの予測	
		Positive	Negative
実際の結果	True	0.63	96.09
	False	0.08	3.20

正解率	96.72%
適合率	88.73%
再現率	16.45%
F1 スコア	0.28
AUC	0.75

表 14：穴馬に着目したモデルの各閾値に対する回収率と Betting Horses

閾値	0.5	0.6	0.7	0.8
単勝回収率	131.82%	239.90%	505.66%	727.62%
複勝回収率	107.16%	193.17%	376.57%	586.34%
Betting Horses	6434	2341	797	445

8.3 考察

穴馬に着目し，オッズを条件に加えることで，大幅に回収率を向上することができた．このような結果になった理由として，オッズが高い馬を当てることができたことが挙げられる．実際に，我々のモデルでは，単勝オッズが 130 倍を超える馬を当てることができている．穴馬に着目したモデルのため，モデル自体の評価は低いが，1 回でも穴馬を当てることができれば，回収率を向上することができると分かる．

しかしながら，Betting Horses は少ないため，継続的にモデルを使用することは難しく，また，穴馬が 3 着以内に入ることは，そもそも少ないため，このモデルの使用は，リスクが高いと考える．

9 おわりに

9.1 まとめと更なる課題

本研究では，機械学習（LightGBM）を用いて，3着以内に入る馬の予測を行い，回収率の計算を行った．単勝・複勝回収率の 100%越えを目標とし，パラメータチューニングの試行や穴馬に着目したモデルの提案を行った．

パラメータチューニング前のモデルでは，閾値を 0.72 以上に設定することで，単勝・複勝回収率の 100%越えを達成することができた．しかしながら，モデルによって適切な閾値は異なるが，一般的に閾値は 0.5 とされているため，閾値を 0.5 に設定した際にも，単勝・複勝回収率が 100%を超えるために，モデル精度の向上を試みた．

モデル精度の向上を目的とし，パラメータチューニングを行った．パラメータチューニングを行うことで．F 値が向上し．モデルの総合的な精度を向上することができた．しかしながら，我々の予想と反し，モデル精度の向上を行ったが，各閾値での単勝・複勝回収率は下がってしまった．各閾値でのそれぞれの回収率が下がってしまった要因として，モデル精度の向上に伴い．Betting Horses が増え．投資金額が高くなってしまったこと，予測精度の高い馬が，人気があり，オッズが低いことが挙げられた．

そこで，我々は，先行研究でも挙げられていた穴馬に着目し，オッズを考慮したモデルの提案を行った，研究内容 1,2 では，3 着以内に入る馬を”1”，4 着以下の馬を”0”として，二値分類問題を解いたが，オッズを考慮したモデルでは，オッズが 15 倍以上で 3 着以内の

馬を”1”，オッズが 15 倍未満かつ 3 着以内の馬，4 着以降の馬を”0”とし，二値分類問題を解いた．

結果として，研究内容 1 では達成できなかった，閾値 0.72 未満での単勝・複勝回収率の 100%越えを達成することができた．このような結果になった理由として，穴馬に着目したため，オッズの高い馬を当てることができたことが挙げられる．しかしながら，研究内容 1，2 と比べ，Betting Horses が極端に下がってしまっていることや，穴馬が 3 着以内に入るレースが少ないことが懸念として挙げられる．

今後の展望としては，賭け式を単勝と複勝の 2 通りに絞っていたが，単勝や複勝よりもオッズの高い 3 連複や 3 連単などの賭け式でも賭けることができるようなモデルの構築や賭け方を提案していきたいと考えている．また，回収率を維持しつつ，Betting Horses を増やすためにも，穴馬と，オッズは低いが 3 着以内に入る確率の高い安定力のある馬を掛け合わせたモデルにも挑戦していきたいと考えている．

参考文献

[1] 経済産業省 大臣官房 調査統計グループ 経済解析室．“コロナ禍の逆風下でも絶好調！！；近年最高の活況となった 2020 年の「競輪・競馬等の競走場，競技団」”．経済産業省．2021-03-02.
https://www.meti.go.jp/statistics/toppage/report/minikaisetsu/hitokoto_kako/20210302hitokoto.html, (参照 2023-12-21)

[2] JRA．“成長推移”．JRA．更新日不明.
https://jra.jp/company/about/outline/growth/pdf/g_22_01.pdf, （参照 2023-12-21）

[3] 荒木雅弘．フリーソフトではじめる機械学習入門．森北出版

[4] 阿部悟．“機械学習の 2 つの壁「分類モデルの選定」と「過学習」への対処法”.
MONOist．2020-04-07.
https://monoist.itmedia.co.jp/mn/articles/2004/07/news017.html, （参照 2023-12-05）

[5] 日本リスク・データ・バンク株式会社（RDB）．“Riskpedia（信用リスク用語集）”.
日本リスク・データ・バンク株式会社（RDB），更新日不明.
https://www.riskdatabank.co.jp/rdb/riskpedia/314/, （参照 2023-12-05）

[6] Stefan Lessmann (2009):” Identifying winners of competitive events: A SVM-based classification model for horserace prediction”, p.1

[7] 芦谷政浩(2010).“「穴馬への過剰な選好 (longshot bias)」に関するサーベイ”．pp1-18.

[8] 小幡　績 ・太宰　北斗(2014)．“競馬とプロスペクト理論：微小確率の過大評価の実証分析”．pp1-17

[9] netkeiba．情報源不明．データベース．https://db.netkeiba.com/race/，（最終入手日

2023-11-23)