三田祭論文

競馬予測における機械学習モデルの比較

三班

小林直生

慶應義塾大学経済学部

概要

近年、ビジネスや日常生活のあらゆるところで機械学習の予測が行われている。競馬予測においても同様に機械学習を用いた予測が行われている。しかし、一言で「機械学習」と言っても、それには多種多様なモデルが存在する。今回は二つのモデルをピックアップし、それぞれ同一のデータを学習させ、実際のレースの予測を算出する。これをもとに、予測に違いが生じるのか、そしてそれはどのような違いなのか、どちらのモデルの予測精度が高いのかを考察していく。

今回ピックアップしたモデルはロジスティック回帰と勾配ブースティングの二つである。この二つのモデルの共通点は教師あり学習であることと、分類タスクが行える点である。競馬予測において、一般的に関心が集まるのは走破タイムなどではなく着順であるため、分類タスクを行えるこの二つを選んだ。今回は「1 位」、「2 位」、「3 位」、「4 位以下」という 4 つのクラスで分類を行う。

データの収集

競馬情報サイトである netkeiba.com から Python による web スクレイピングを用いて、データを収集した。収集したデータは東京競馬場で行われた 2022 年 1 月から 2022 年 6 月までのレースデータと 10 月 29 日、30 日のレースデータである。

ロジスティック回帰による分析

まずはロジスティック回帰の理論的な説明を行う。ロジスティック回帰は、各クラスの事後確率の比の対数を、特徴量と重み係数の線形和によって近似するモデルである。今回は「1 位」、「2 位」、「3 位」、「4 位以下」という 4 つのクラス分けを行う。 3 クラス以上の識別を行うロジスティック回帰を特に多クラスロジスティック回帰と呼ぶ。「1 位」、「2 位」、「3 位」、「4 位以下」のクラスをそれぞれクラスC_1、C_2、C_3、C_4とする。クラスC_i(i=1, … , 4)に対応する重み係数ベクトルw_iと入力ベクトルxの内積をa_iとおく。

$$a_i = w_i^T x$$

入力xが与えられたとき、それがクラスC_iである確率を次のように近似する。

$$p(C_i|x) = \frac{expa_i}{\sum_{l=1}^{4} expa_l}$$

これはソフトマックス関数または正規化指数関数と呼ばれる。$p(C_i|x)$が最大となるクラスを識別結果とする。また、学習データから推定するパラメータは 4 個の重み係数ベクトル$w_i(i = 1, \dots, 4)$である。

2022 年 1 月から 6 月のデータを用いて実際に学習を行う。訓練データとテストデータの割合は 7 対 3 に分けた。目的変数は着順であり、先述の通り「1 位」、「2 位」、「3 位」、「4 位以下」でランク分けを行う。説明変数は「枠番」、「馬番」、「斤量」、「単勝」、「人気」、「年齢」、「体重」、「体重変化」、そしてダミー変数化した「馬名」と「騎手」である。「単勝」とは単勝オッズの値のことである。また「人気」とは人気ランキングの数字である（人気が 1 位であれば「 1 」、2 位であれば「 2 」）。そして馬名と騎手を除いた説明変数のそれぞれの回帰係数は以下の表 1 の通りである。この表から人気や斤量、年齢の係数が大きいことがわかる。

枠番	-1.38×10^{-2}
馬番	-1.19×10^{-2}
斤量	1.84×10^{-2}
単勝	-1.99×10^{-3}
人気	-1.75×10^{-1}
年齢	-1.79×10^{-2}
体重	-3.90×10^{-4}
体重変化	2.20×10^{-3}

表１：ロジスティック回帰における回帰係数

続いて構築したモデルを 10 月 29 日、10 月 30 日のレースに当てはめる。全 24 レース、316 頭の馬の予想順位と実際の順位を以下の表にまとめた。

	1位	2位	3位	4位
1位予想	6	3	4	11
2位予想	8	7	2	7
3位予想	2	3	4	15
4位以下予想	8	11	14	211

表２：ロジスティック回帰モデルによる予想と実際の結果

勾配ブースティングによる分析

勾配ブースティングの説明を行う。「ブースティング」とは、与えられたデータから決定木分析を行った後に、予測が正しくできなかったデータに重みをつけ、再度、決定木分析を行うという作業を繰り返すことで精度を高める方法である。その中でも勾配ブースティングはデータに重みづけをするのではなく、予測値と実測値の誤差を計算し、誤差を決定木で学習する方法である。ブースティングと同様に誤差に対する学習を繰り返すことで精度を高めていく。

具体的な手順は以下の通りである。

勾配ブースティングの手順

1. 目的変数の平均を計算する
2. 誤差を計算する
3. 決定木を構築する
4. アンサンブルを用いて新たな予測値を求める
5. 再び誤差を計算する
6. 3~5 を繰り返す
7. 最終予測を行う

数式で表すと

1. 定数でモデルを初期化する。

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma)$$

2. モデルの数だけ以下を繰り返す

$$\text{For } m = 1 \text{ to } M$$

1. 擬似残渣を計算する

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n$$

2. ベースモデル$h_m(x)$の擬似残渣を用いてデータセット$\{(x_i, r_{im})\}_{i=1}^{n}$を作成しフィッティングする
3. 次の1次元最適化問題を解いて乗数を計算する

$$\gamma_m = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

4. モデルを更新する

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

3. 最終的なモデル$F_M(x)$を出力する

勾配ブースティングを用いたアルゴリズムとしては XGBoost、Catboost、LightGBM などがある。今回は LightGBM を用いて分析を行う。

LightGBM（Light Gradient Boosting Machine）は「Light（軽い、高速)」なことが特徴である。勾配ブースティングの場合は、誤差を最小化するように“分割”の要素、基準を見つ

けるため、データ量に応じて計算量が増える。1 つ 1 つの決定木の精度をなるべく落とさずに、高速に構築できるようにしたことが LightGBM の最大の特徴である。Light にした工夫としては以下の点があげられる。

①Leaf-wise tree growth

一般的な決定木の場合は、決定木の階層ごとに計算（Level wise）するため、1 つの階層の分岐がすべて終わってから次の階層を計算するが、分岐が必要なくなった要素（＝葉、leaf）については、それ以上は計算しない。

②Histogram based

決定木の分岐をする際に、すべての値をみるのではなく、ヒストグラムをつくって、数値をまとめて分岐させる。

③Gradient-based One-Side Sampling (GOSS)

学習できていない要素を学ぶことを優先するため、誤差が小さいデータは減らし、誤差の大きいデータだけを残すことで学習データの量を減らす。

④Exclusive Feature Bundling (EFB)

異なる特徴量の中でも、まとめても問題がなさそうな特徴量を 1 つにすることで計算量を減らす。

実際に 2022 年 1 月から 6 月までのデータを用いて学習を行う。また、分析をするにあたって、ハイパーパラメータと呼ばれる変数を事前に設定する必要がある。決定木の「葉の数（num_leaves)」や「1 つの葉に含まれる最小データ数（min_data_in_leaf)」、「階層の深さ（max_depth)」などを、モデルの精度と過学習のバランスを考えながらチューニングすることが求められる。今回はハイパーパラメータを自動で設定できる optuna というツールを使い、チューニングを行う。

ハイパーパラメータ	意味
feature_pre_filter	min_child_samples をチューニングする時には False になる
lambda_l1, lambda_l2	正則化パラメータ。過学習を防止する
num_leaves	一つの木に含まれる最大の葉の数
feature_fraction	特徴量の何割を使って一つの木を育てるか
bagging_fraction	データの何割を使って一つの木を育てるか
bagging_freq	バギング（データを水増し）する頻度。0 はバギングしない
min_child_samples	最終的に一つの葉に残るデータの数

下の表が実際にチューニングを行った結果である。

ハイパーパラメータ	
feature_pre_filter	False
lambda_l1	6.97×10^{-6}
lambda_l2	2.00×10^{-2}
num_leaves	9
feature_fraction	0.680
bagging_fraction	1.0
bagging_freq	0
min_child_samples	20

このパラメータをもとに勾配ブースティングモデルを構築し、ロジスティック回帰のときと同様に 10 月 29 日と 10 月 30 日のレースに当てはめる。予想順位と実際の順位の表は次の通りである。

	1位	2位	3位	4位以下
1位予想	3	3	3	15
2位予想	6	1	1	16
3位予想	0	4	6	14
4位以下予想	15	16	14	199

結果の考察

今回は二つのモデルの予測結果がどれほど異なっているのか、そしてどちらの予測精度が優れているのかを考察していく。まずは二つのモデルの予測結果の違いを見ていく。10 月 29 日、30 日のレースでの二つのモデルの 1 位から 3 位までの予測を表にまとめた。ロジスティック回帰による予想順位と勾配ブースティングによる予想順位が一致している箇所をマーカーで黄色く塗りつぶした。

10 月 29 日のレースの予想

（数字は馬番、ロジ＝ロジスティック回帰の予想、勾配＝勾配ブースティングの予想）

	レース 1		レース 2		レース 3		レース 4	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	9	1	8	10	3	3	4	11
2位予想	10	9	1	11	2	2	10	7
3位予想	1	10	9	8	7	7	13	14
	レース 5		レース 6		レース 7		レース 8	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	7	7	9	12	6	13	4	9
2位予想	8	15	8	1	11	4	10	6
3位予想	2	11	6	8	5	9	8	3
	レース 9		レース１０		レース１１		レース１２	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	11	12	10	12	5	7	4	4
2位予想	14	14	6	5	3	2	3	13
3位予想	6	9	3	9	10	4	16	16

10 月 30 日のレースの予想

	レース 1		レース 2		レース 3		レース 4	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	13	13	9	8	7	6	5	12
2位予想	8	8	5	9	5	7	4	7
3位予想	4	2	8	5	6	5	11	10
	レース 5		レース 6		レース 7		レース 8	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	8	7	2	16	4	6	11	4
2位予想	6	12	3	2	6	4	6	2
3位予想	5	4	7	7	10	10	1	8
	レース 9		レース１０		レース１１		レース１２	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	5	13	10	6	8	9	4	13
2位予想	12	12	13	13	7	6	12	4
3位予想	10	1	16	16	4	3	6	11

24 レースの 1 位から 3 位それぞれの予想において、ロジスティック回帰による予測と勾配ブースティングによる予測が一致している割合は 0.194(14/72)である。

続いて 3 位圏内に一致している馬が何頭あるかを下の表で示した。マーカーの色の基準は次のとおりである。

3つ一致	
2つ一致	
1つ一致	

10 月 29 日のレースの予想

	レース 1		レース 2		レース 3		レース 4	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	9	1	8	10	3	3	4	11
2位予想	10	9	1	11	2	2	10	7
3位予想	1	10	9	8	7	7	13	14
	レース 5		レース 6		レース 7		レース 8	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	7	7	9	12	6	13	4	9
2位予想	8	15	8	1	11	4	10	6
3位予想	2	11	6	8	5	9	8	3
	レース 9		レース 1 0		レース 1 1		レース 1 2	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	11	12	10	12	5	7	4	4
2位予想	14	14	6	5	3	2	3	13
3位予想	6	9	3	9	10	4	16	16

10 月 30 日のレースの予想

	レース 1		レース 2		レース 3		レース 4	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	13	13	9	8	7	6	5	12
2位予想	8	8	5	9	5	7	4	7
3位予想	4	2	8	5	6	5	11	10
	レース 5		レース 6		レース 7		レース 8	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	8	7	2	16	4	6	11	4
2位予想	6	12	3	2	6	4	6	2
3位予想	5	4	7	7	10	10	1	8
	レース 9		レース 1 0		レース 1 1		レース 1 2	
	ロジ	勾配	ロジ	勾配	ロジ	勾配	ロジ	勾配
1位予想	5	13	10	6	8	9	4	13
2位予想	12	12	13	13	7	6	12	4
3位予想	10	1	16	16	4	3	6	11

それぞれのレースの 3 位圏内の 3 つの予測のうち、3 つとも一致しているレースは 5 レース、2 つが一致しているレースは 4 つ、一つが一致しているレースは 6 つとなった。

また、表 1、表 2 のデータをもとにそれぞれの適合率を比較していく。適合率とは予測したデータのうち、実測値と一致するデータの割合のことである。下の表は 1 位、2 位、3 位それぞれの適合率をモデル別でまとめたものである。

	ロジスティック回帰	勾配ブースティング
1位予想の適合率	0.250	0.125
2位予想の適合率	0.292	0.042
3位予想の適合率	0.167	0.250

各順位予測の適合率をみていくと、ロジスティック回帰と勾配ブースティングでそれぞれ1位予想が0.250、0.125、2位予想が0.292、0.042、3位予想が0.167、0.250となり、1位予想と2位予想の適合率はロジスティック回帰が上回り、3位予想の適合率は勾配ブースティングが上回る結果となった。1位から3位までの各適合率の平均を計算すると、ロジスティック回帰が0.236で勾配ブースティングが0.139となる

また、3位以上と予想し、実際に3位以上になる割合は以下の表の通りである。

	ロジスティック回帰	勾配ブースティング
3位以上予想の適合率	0.542	0.375

これらの結果から総じてロジスティック回帰の予測精度の方が勾配ブースティングの予測精度に比べ、高かったといえる。

今後に向けて

今回、二つのモデルでの予測結果の違いを観測することはできた。しかし、なぜそのような差異が生じたのかという原因までは追求できなかった。今後の研究で行なっていきたいことは三つある。一つ目はサンプル数の増大である。今回、東京競馬場における2022年1月から6月までのデータをもとにモデルを構築し、10月29日、30日のレースを予測した。その結果、適合率を算出する際の母数が24となり、二つのモデルの優劣を比較する上では物足りないサンプル数となったため、改善していきたい。

二つ目は様々な買い方での回収率をシミュレーションすることである。複勝、三連単、三連複など様々な馬券の買い方からモデルの比較を行うことで、二つのモデルの特徴の違いがより明確に見えてくる可能性がある。このような試行錯誤によって、機械学習モデルの違いが予測にどのような違いを与えるのか、なぜそのような違いが生じるのかを明確にしていきたい。

三つ目は回収率向上に向けた戦略策定である。今回はあくまでもモデルの比較がテーマであったため、回収率を最大化する方法の模索などは行わなかった。しかし、競馬予測において関心が集まるのは、やはり収益である。そのため、モデルの性質の違いを踏まえた上で、回収率が上がるような戦略を策定することを最終目標としていきたい。

参考文献

書籍

・速水悟　(2016)『事例＋演習で学ぶ機械学習　ビジネスを支えるデータ活用のしくみ』　森北出版
・金森敬文　(2018)『Python で学ぶ統計的機械学習』　オーム社

WEB 上の資料

・netkeiba.com　https://www.netkeiba.com/?rf=logo
最終閲覧日 2022.11.25
・野村総合研究所　『～用語解説～　LightGBM』
https://www.nri.com/jp/knowledge/glossary/lst/alphabet/light_gbm
最終閲覧日 2022.11.25
・Preferred Networks　『optuna ハイパーパラメータ自動最適化フレームワーク』
https://www.preferred.jp/ja/projects/optuna/
最終閲覧日 2022.11.25
・『競馬予想で始める機械学習』
https://zenn.dev/dijzpeb/books/848d4d8e47001193f3fb
最終閲覧日 2022.11.25
・@kuroitu　『勾配ブースティング決定木ってなんぞや』
https://qiita.com/kuroitu/items/57425380546f7b9ed91c
最終閲覧日　2022.11.25