Betting Odds Calculation with Machine Learning

LYU 2102
NAM Man Leung
supervised by Prof. Michael Lyu
Outline

• Introduction
• Background Knowledge
• Data Preparation
• Methodology
• Experiment and Result
• Conclusion
Introduction - Motivation

• Revenue from horse racing is approximately HK$280 billion in 2020-2021 despite the economic downturn caused by the coronavirus pandemic.
• Win odds enhance prediction accuracy as shown in previous FYP[1][2].
• Win odds keep changing before the start of the race.
• Use machine learning methods to resemble the effect of winning odds in horse racing prediction.
Introduction - Objective

• Apply statistical models (rating systems) to evaluate the performance of horses
• Apply techniques in natural language processing for winning horse classification
• Reproduce the effect of variable win odds from the Hong Kong Jockey Club in horse racing prediction by invariable features
Background Knowledge – Rating System

Glicko Rating System[3]

- Rating
  - Performance of a horse
- Rating deviation
  - Reliability of a horse’s rating
  - a low value of rating deviation indicates that the horse joins races frequently and the rating is more reliable
  - uncertainty of a horse’s ability reduces because more information is obtained when the horse joins more races
Background Knowledge – Rating System

TrueSkill Rating System[4]

• Rating
  • Performance of a horse

• Rating deviation
  • Reliability of a horse’s rating

• Multiple horse environment
  • Assume outcome of each race is a permutation of multiple horses
  • Allow horses to have the same rank
Background Knowledge – Rating System

Elo-MMR Rating System[5]

- Rating
  - Performance of a horse
- Rating deviation
  - Reliability of a horse’s rating
- Multiple horse environment
  - Assume outcome of each race is a permutation of multiple horses
  - Assume horses have distinct ranks
- Incentive compatible
  - Horses’ ratings should not have opposite changes to their performance
Background Knowledge – Transformer

Self Attention mechanism[6]
- Features in the sequence interact with each other
- Assign weights to features according to the relative importance
- Decide dependency relationships between features of the sequence
Data Preparation - Collection

• Write web crawlers by using BeautifulSoup library in python
• Collect data on the HKJC official websites
• Obtain 9191 race records in our dataset dated from June 2008 to October 2021
• Obtain horse records of 6642 horses which participated in those 9191 race records
Data Preparation – Feature Analysis

• By Bayes’ formula,
  \[ \Pr( Y = \text{win} \mid X = x_1, x_2, ...) = \frac{\Pr(X = x_1, x_2, ... \mid Y = \text{win}) \Pr( Y = \text{win})}{\Pr( X = x_1, x_2, ... )} \]

• Likelihood estimation in machine learning is simplified by assuming that features are conditional independent.

• We observe the likelihood \( \Pr( X = x \mid Y = \text{win} ) \) one by one.
Data Preparation – Feature Analysis (Age)

- Most winning horses are aged between 5 and 6
Data Preparation – Feature Analysis (Draw)

• Most winning horses have smaller draw number.
Data Preparation – Feature Analysis (Origin)

• Most winning horses come from Australia or New Zealand
Data Preparation – Feature Analysis (Color)

• Most winning horses have skin color Bay
Data Preparation – Feature Analysis (Sex)

- Most winning horses are of sex Gelding.
Data Preparation – Feature Analysis (Numerical features)

- Frequency of 1\textsuperscript{st} place has a significant correlation with the frequency of 2\textsuperscript{nd} place and 3\textsuperscript{rd} place which are 0.4500 and 0.4468 respectively.

- Rating systems are applicable in prediction.
Data Preparation – Feature Analysis (Numerical features)

• A positive correlation of 0.4291 between the win odds and the place

• Winning odds help the prediction of horse racing result
Data Preparation – Data Imputation

• A small part of horse data about those retired horses is missing in our data set
• do data imputation on our dataset by using the k nearest neighbors method
• Invoke the KNN Imputer from Scikit Learn library to impute the missing values
Data Preparation – Data Encoding

- Input of our neural network models must be numerical but some of our data are categorical
- One hot encoding[7]
  - dimension of our input will be increased drastically
  - requires extra memory and more computational time in training
- Ordinal Encoding scheme[7]
  - a unique integer means a category
  - dimension of the data is the same as the original
- invoke the Ordinal Encoder from the Scikit Learn Library
Data Preparation – Normalization

• z-score normalization[8]
• Values of all variables are recomputed into the same scale
• the same scale of all variables balances the focus of error minimization in the weight correction algorithm

\[ x_i' = \frac{x_i - \bar{x}_i}{\sigma_i} \]
Data Preparation – Rating Generation

• Ratings mentioned before do not exist on the HKJC websites
• need to calculate those ratings with the information provided by our dataset

<table>
<thead>
<tr>
<th>contest_index</th>
<th>rating_mu</th>
<th>rating_sig</th>
<th>perf_score</th>
<th>place</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
<td>1400</td>
<td>174</td>
<td>1381</td>
<td>9</td>
</tr>
<tr>
<td>107</td>
<td>1384</td>
<td>133</td>
<td>1365</td>
<td>11</td>
</tr>
<tr>
<td>162</td>
<td>1481</td>
<td>114</td>
<td>1724</td>
<td>5</td>
</tr>
<tr>
<td>199</td>
<td>1443</td>
<td>102</td>
<td>1370</td>
<td>9</td>
</tr>
<tr>
<td>275</td>
<td>1463</td>
<td>95</td>
<td>1511</td>
<td>7</td>
</tr>
<tr>
<td>328</td>
<td>1482</td>
<td>90</td>
<td>1540</td>
<td>7</td>
</tr>
<tr>
<td>351</td>
<td>1491</td>
<td>87</td>
<td>1517</td>
<td>6</td>
</tr>
<tr>
<td>386</td>
<td>1522</td>
<td>85</td>
<td>1639</td>
<td>6</td>
</tr>
<tr>
<td>417</td>
<td>1543</td>
<td>84</td>
<td>1620</td>
<td>4</td>
</tr>
<tr>
<td>446</td>
<td>1563</td>
<td>82</td>
<td>1642</td>
<td>6</td>
</tr>
<tr>
<td>468</td>
<td>1527</td>
<td>82</td>
<td>1370</td>
<td>10</td>
</tr>
<tr>
<td>533</td>
<td>1562</td>
<td>81</td>
<td>1700</td>
<td>2</td>
</tr>
<tr>
<td>592</td>
<td>1610</td>
<td>81</td>
<td>1833</td>
<td>1</td>
</tr>
<tr>
<td>632</td>
<td>1613</td>
<td>81</td>
<td>1623</td>
<td>5</td>
</tr>
<tr>
<td>687</td>
<td>1626</td>
<td>80</td>
<td>1677</td>
<td>6</td>
</tr>
</tbody>
</table>
Methodology - Overview

• Rating systems estimate the relative skill level of horses based on their historical performance

• Self attention mechanism captures the dependencies between horses

• Multiclass classification on place
  • The winning horse number is the output

• Transformer classification model including ratings in the feature list
Methodology - Evaluation

• Accuracy
  • Accurate prediction about the winner
• Betting simulation
  • Net gain
  • Bet $10 for each race in test data

• Combining transformer architecture and ratings give a better result
  • Multilayer perceptron with ratings
  • Transformer without ratings
  • Transformer with ratings
Methodology – Multilayer perceptron

1. **Input layer**
2. **3 linear hidden layer**
3. **Dropout layer**
4. **Output layer**

- Relu Activation function
- Cross-Entropy Loss Function
- Stochastic gradient descent
Methodology – Transformer

1. Input layer
2. Word embedding layer
3. Position embedding layer
4. Transformer encoder
5. 2 linear hidden layers
6. Dropout layer
7. Output layer

- Relu Activation function
- Cross-Entropy Loss Function
- Stochastic gradient descent
Experiment and Result – Input Data

• Training data:
  • races from 22 June 2008 to 6 December 2020.
  • 8500 races

• Testing data:
  • all races from 9 December 2020 to 17 October 2021.
  • 688 races
## Experiment and Result – Input Data

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Venue</td>
<td>Location of the race</td>
</tr>
<tr>
<td>Horse_class</td>
<td>Class of the horses</td>
</tr>
<tr>
<td></td>
<td>Stronger horses compete in high race class</td>
</tr>
<tr>
<td>Distance</td>
<td>The distance of the race</td>
</tr>
<tr>
<td>Going</td>
<td>Condition of the lane</td>
</tr>
<tr>
<td>Course_track</td>
<td>The lane of the race</td>
</tr>
<tr>
<td>Course_track_code</td>
<td>Description about the lane</td>
</tr>
<tr>
<td>Horse_i_number</td>
<td>The horse number in the race</td>
</tr>
<tr>
<td>Horse_i_name</td>
<td>The name of horse</td>
</tr>
<tr>
<td>Horse_i_jockey</td>
<td>The name of jockey</td>
</tr>
<tr>
<td>Horse_i_trainer</td>
<td>The name of trainer</td>
</tr>
<tr>
<td>Horse_i_declared_weight</td>
<td>The weight of horse</td>
</tr>
<tr>
<td>Horse_i_origin</td>
<td>The place of birth</td>
</tr>
<tr>
<td>Horse_i_age</td>
<td>The age of horse</td>
</tr>
<tr>
<td>Horse_i_color</td>
<td>The color of skin</td>
</tr>
<tr>
<td>Horse_i_sex</td>
<td>The gender of horse</td>
</tr>
<tr>
<td>Horse_i_1st_place_frequency</td>
<td>The frequency of getting 1st place</td>
</tr>
<tr>
<td>Horse_i_total_race</td>
<td>The total count of horse's participation</td>
</tr>
<tr>
<td>Horse_i_rating</td>
<td>The rating of the horse</td>
</tr>
</tbody>
</table>

Repeat 14 times
Experiment and Result – Accuracy

• Multilayer perceptron with ratings
• The model with Glicko ratings reaches the highest test accuracy of 20.4%
• The accuracy of the model with Glicko ratings fluctuates in a larger range than that with Elo-MMR and TrueSkill
Experiment and Result – Accuracy

- Transformer without ratings
- the best performance of this model is having 19.2% before overfitting
Experiment and Result – Accuracy

- Transformer with ratings
- The transformer model with Elo-MMR ratings has the highest test accuracy of 21.4% among the other models
Experiment and Result – Betting Simulation

• Multilayer perceptron with ratings
• all three models perform better than random betting
• Multilayer perceptron with Elo-MMR ratings has the best performance
  • highest net gain : -13%
Experiment and Result – Betting Simulation

• Transformer without ratings
• The net gain is -4% after betting on all 688 races in our test data
• Better than the multilayer perceptron
Experiment and Result – Betting Simulation

• Transformer with ratings
• Positive net gain of 3% to 6% after betting on 688 races in the test data
• Transformer with Elo-MMR ratings has the best performance
Conclusion

• Win odds in the feature list have the effect of enhancing the accuracy and net gain
• Exclude the win odds from the feature list this time
• Resemble the effect of the winning odds by combining rating systems and the transformer architecture
• the best case of our models is the transformer with Elo-MMR ratings
  • the highest test accuracy : 21.4%
  • a positive net gain : 6%
Thank you!
References


References