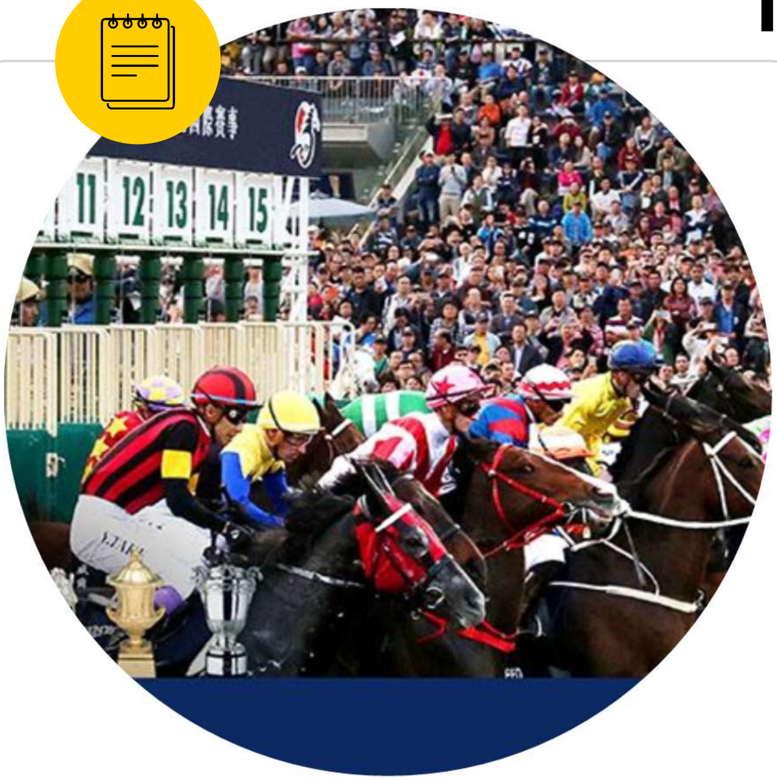
Predicting Horse Racing Results with TensorFlow

LYU 1703

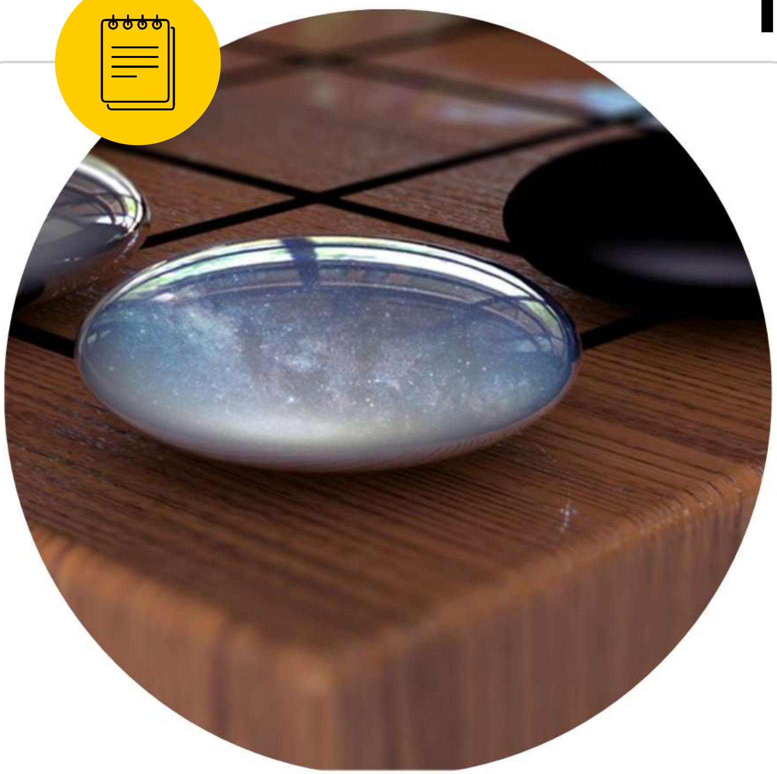
LIU YIDE
WANG ZUOYANG

News



CUHK Professor , Gu Mingao, wins 50 MILLIONS dividend using his "sure-win" statistical strategy.

News



AlphaGO defeats human world champions at the Chinese ancient game of GO.

Introduction

Motivation

Can we predict the horse racing results, using

○ Machine Learning (specifically, Neural Network) only

✕ instead of statistical inference*

** Professor Gu's work on this topic is NOT PUBLISHED by the time of the presentation.*

Related Work

Few work on related topic is published.

- Williams and Li (2008)
  - Reviewed neural network algorithms. (BP, Quasi Newton, etc.)
  - Predicted horse finishing time of individual horses.
  - Claimed to have great performance (little result data).
- LYU1603
  - Predicted horse finishing time of all horses.
  - Obtained actual net gain with a threshold (>95%)
  - Problem: too high threshold (bet <10 times in a season)

Introduction

Outline

- Background
- Two Approaches
  - Additional information - Weather
  - Divide and Conquer
- Model Architecture
- Results & Discussion
- Conclusion & Future Work
- Q&A

1 Background

Horse Racing Background

- Professional sport to run horse in time
  - Horses are competing in a game for speed.

- Professional & National entertainment events for Hong Kong citizens
  - Over 45% of citizens have betting account.
  - Advanced Pari-mutuel betting.
  - >20 bet types.

Objective

Bets	Meaning
Win	1st in a race
Place	1st, 2nd, 3rd in a race

Table 1: Bets of focus in this project

Objective: Build a prediction model to obtain positive net gain.

Possible ways to model results

Horse racing result is very difficult to model.

- Horse win
  - Predict whether a horse will win
  - Binary classification of win or not
  - Problems:
    - Unevenly distributed dataset (1 win and 13 losses, normally)
    - Cannot model a race
    - Repetitive wins in a race

Possible ways to model results

Horse racing result is very difficult to model.

- Horse ranks
  - Predict ranks of horses in a race
  - Multi-class classification
  - Problems:
    - Races of different horses
    - Ambiguous
    - Repetitive

Horse\Place	1st	2nd	3rd
#1	60%	40%	20%
#2	30%	60%	50%
#3	50%	40%	60%

Possible ways to model results

Horse racing result is very difficult to model.

- Horse finishing time
  - Predict horse finishing time in a race
  - Regression problem
  - Reflect recent horse strength to some extent
  - Problems:
    - Predict finishing time individually
    - But then grouped into a race

2 Approach

Approach

- Additional Information
  - Weather
  - Extract horse racing features
    - Weight difference/ Previous Place

- Divide and Conquer
  - Divide on location
  - Shatin (ST) and Happy Valley (HV)
  - (Extract horse racing features)

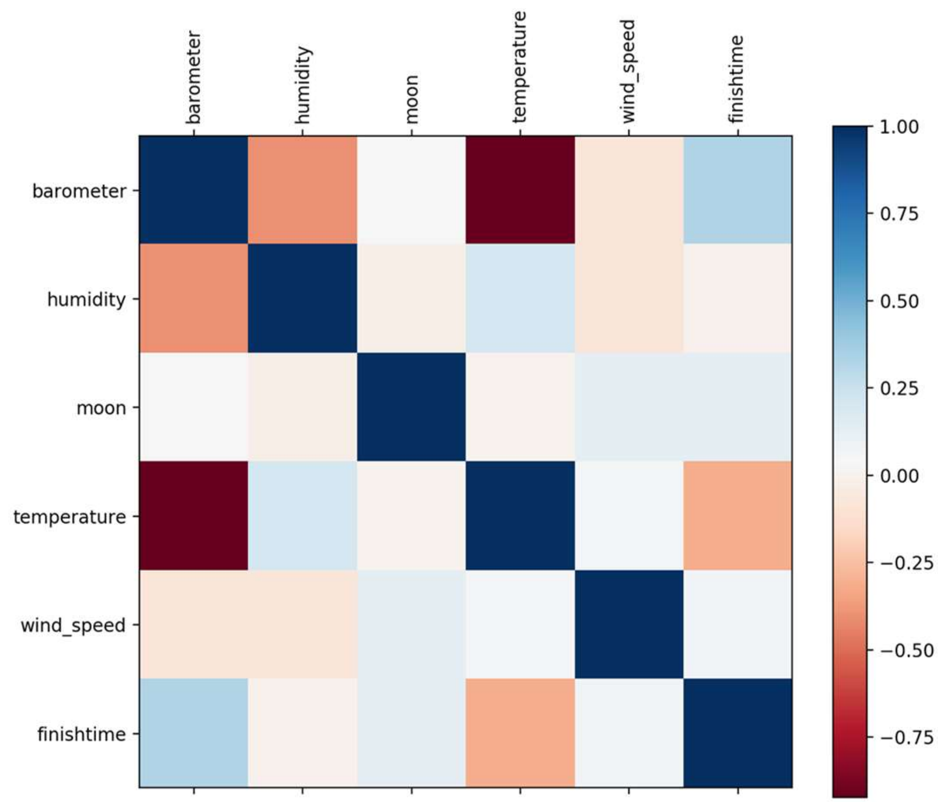
Weather Features

- Horse Performance is influenced by the weather
  - Average performance
  - Individual performance

- Collected Features:
  - Moon phase
  - Wind direction and speed
  - Humidity and weather condition
  - Temperature

Average Performance

- Average horse finishing time can be influenced by weather features
- Temperature ⬆
  Finishing time ⬇

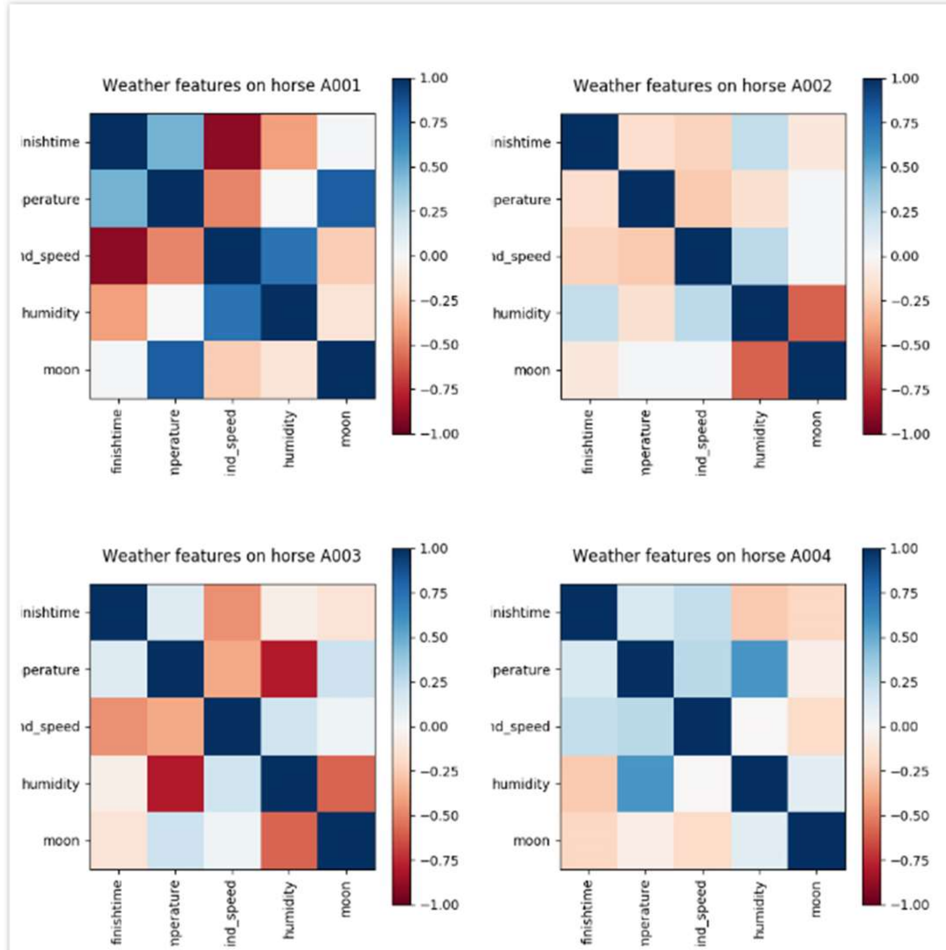


*Finishtime is averaged and normalized by distance to represent horse performances.

Individual Performance

- Individual horse has different performances in different weather

- Weather is closely correlated to both average and individual performances.



*Finishtime normalized by distance to represent horse performances.

Why Divide and Conquer

- Two racecourses: Sha Tin and Happy Valley;
- Previous studies show some patterns;
- Tuning sub-models to optimize in the future.

Divide and Conquer By Location

- Split the data set into two subsets;
- Build and train NN models based on both subsets;
- Predict separately on both models and combine.

Win odds?

- Odds is closely related to the prediction by intuition.
- However LYU 1603 chose to exclude this feature.
- Compare models with odds and without odds to figure it out.

3

Configuration

Structures and settings of the models

Layer and batch size

- Commonly used structures are used for this semester;
- Number of layers: 2;
- Batch size: 128;
- We assume this network configuration is representative.

Train & Test data set

- Need to be comparable to LYU 1603 and 1604;
- Train data: 2011 - 2014;
- Test data: 2015 - 2016.

Number of training steps

To search for a best number of training steps, a simple experiment is conducted.

Number of Steps	Noodds_noweather	noodds_weather	odds_noweather	odds_weather
10k	4.025	3.603	4.347	3.263
100k	4.291	4.697	4.819	3.668
1m	5.192	5.221	5.088	4.281

TABLE 3.1: Experiments on the number of training steps

Evaluation Standard

- Loss: Mean-square-error between predicted and actual finishing time
- Accuracy_win: Accuracy of correct win bets
- Accuracy_place: Accuracy of correct place bets
- Net gain: Overall profits of all bets

4 Results & Discussion

Results

Models	Model 000	Model 001	Model 010	Model 011	Model 100	Model 101	Model 110	Model 111
Loss	515.2	461.2	556.4	417.7	583/ 575	527/ 536	629/ 577	652/ 589
Accuracy _win	0.08367	0.07029	0.08090	0.10742	0.08798/ 0.08014	0.07725/ 0.07292	0.08155/ 0.09028	0.07940/ 0.06944
Accuracy _place	0.42926	0.41954	0.47547	0.47789	0.44277/ 0.43902	0.43419/ 0.46766	0.44778/ 0.47052	0.4542/ 0.47685
Net gain	-1087	-991	-1378	-568	37/ -1005	-1088/ -1579	655/ -917	339/ -1724

- Notation: three binary digits representing divided/undivided, odds/no odds and weather/no weather.
- For the divided models, the first values refer to Sha Tin and the second refer to Happy Valley.

Results

Models	Model 000	Model 001	Model 010	Model 011	Model 100	Model 101	Model 110	Model 111
Loss	515.2	461.2	556.4	417.7	583/ 575	527/ 536	629/ 577	652/ 589
Accuracy _win	0.08367	0.07029	0.08090	0.10742	0.08798/ 0.08014	0.07725/ 0.07292	0.08155/ 0.09028	0.07940/ 0.06944
Accuracy _place	0.42926	0.41954	0.47547	0.47789	0.44277/ 0.43902	0.43419/ 0.46766	0.44778/ 0.47052	0.4542/ 0.47685
Net gain	-1087	-991	-1378	-568	37/ -1005	-1088/ -1579	655/ -917	339/ -1724

- Loss:
  - Weather features reduce prediction loss.
  - Win odds increases prediction loss.
  - Dividing the dataset will increase prediction loss.

Results

Models	Model 000	Model 001	Model 010	Model 011	Model 100	Model 101	Model 110	Model 111
Loss	515.2	461.2	556.4	417.7	583/ 575	527/ 536	629/ 577	652/ 589
Accuracy _win	0.08367	0.07029	0.08090	0.10742	0.08798/ 0.08014	0.07725/ 0.07292	0.08155/ 0.09028	0.07940/ 0.06944
Accuracy _place	0.42926	0.41954	0.47547	0.47789	0.44277/ 0.43902	0.43419/ 0.46766	0.44778/ 0.47052	0.4542/ 0.47685
Net gain	-1087	-991	-1378	-568	37/ -1005	-1088/ -1579	655/ -917	339/ -1724

- Accuracy:
  - Weather features reduce prediction accuracy.
  - Win odds affects prediction accuracy unclearly.
  - Dividing the dataset does not affect prediction accuracy significantly.

Results

Models	Model 000	Model 001	Model 010	Model 011	Model 100	Model 101	Model 110	Model 111
Loss	515.2	461.2	556.4	417.7	583/ 575	527/ 536	629/ 577	652/ 589
Accuracy _win	0.08367	0.07029	0.08090	0.10742	0.08798/ 0.08014	0.07725/ 0.07292	0.08155/ 0.09028	0.07940/ 0.06944
Accuracy _place	0.42926	0.41954	0.47547	0.47789	0.44277/ 0.43902	0.43419/ 0.46766	0.44778/ 0.47052	0.4542/ 0.47685
Net gain	-1087	-991	-1378	-568	37/ -1005	-1088/ -1579	655/ -917	339/ -1724

- Net gain:
  - Weather features increase net gain this time.
  - No obvious patterns shown for win odds or dividing the data.
  - Races in Sha Tin are much more predictable than those in Happy Valley.

Results

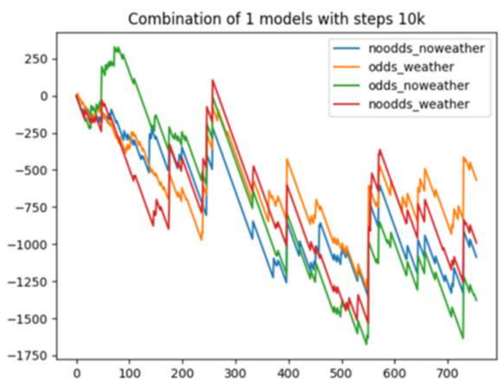
Models	Model 000	Model 001	Model 010	Model 011	Model 100	Model 101	Model 110	Model 111
Loss	515.2	461.2	556.4	417.7	583/ 575	527/ 536	629/ 577	652/ 589
Accuracy _win	0.08367	0.07029	0.08090	0.10742	0.08798/ 0.08014	0.07725/ 0.07292	0.08155/ 0.09028	0.07940/ 0.06944
Accuracy _place	0.42926	0.41954	0.47547	0.47789	0.44277/ 0.43902	0.43419/ 0.46766	0.44778/ 0.47052	0.4542/ 0.47685
Net gain	-1087	-991	-1378	-568	37/ -1005	-1088/ -1579	655/ -917	339/ -1724

- Decrease in loss ≠ Increase in accuracy.
- Higher accuracy ≠ higher net gain (because of win odds).

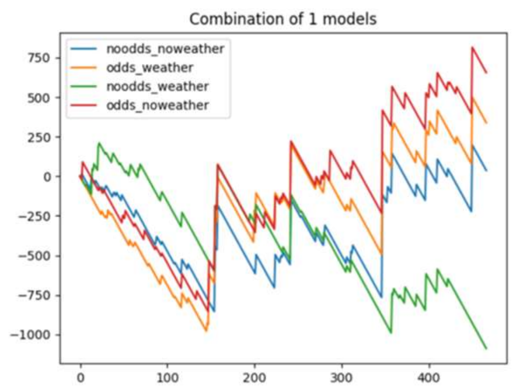
- Net gain is low because we bet on all the horses the predictions suggest.
- To increase net gain, more strategies need to be applied.

Results

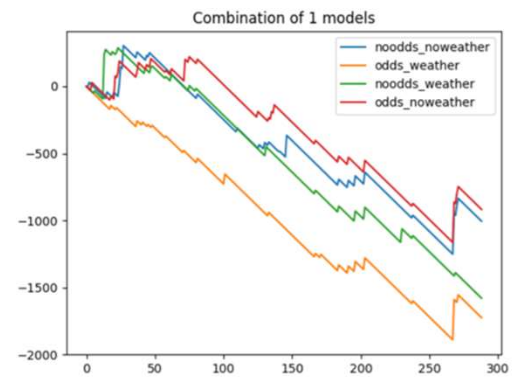
Figures



Unvided



Shatin



Happy Valley

- Average Net gain:

-1006

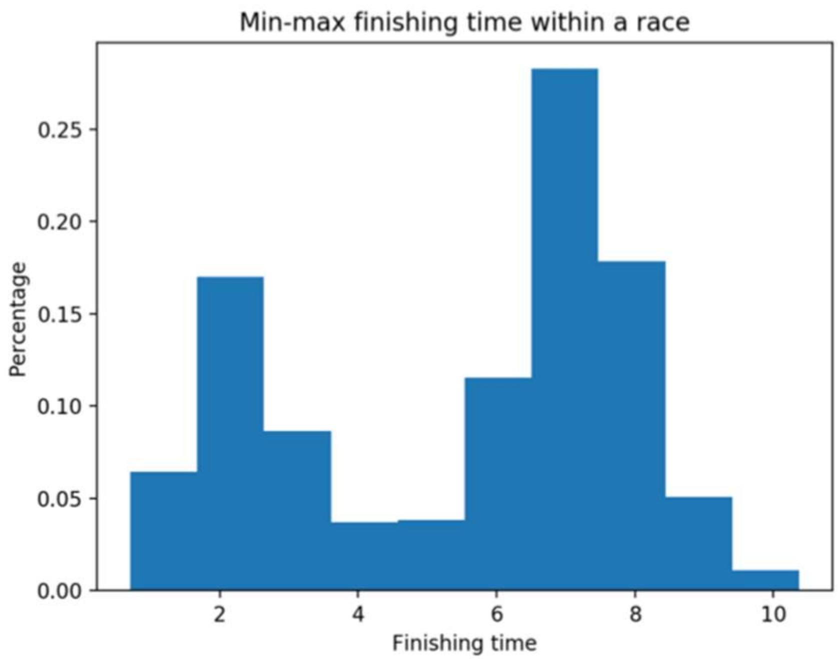
-1306

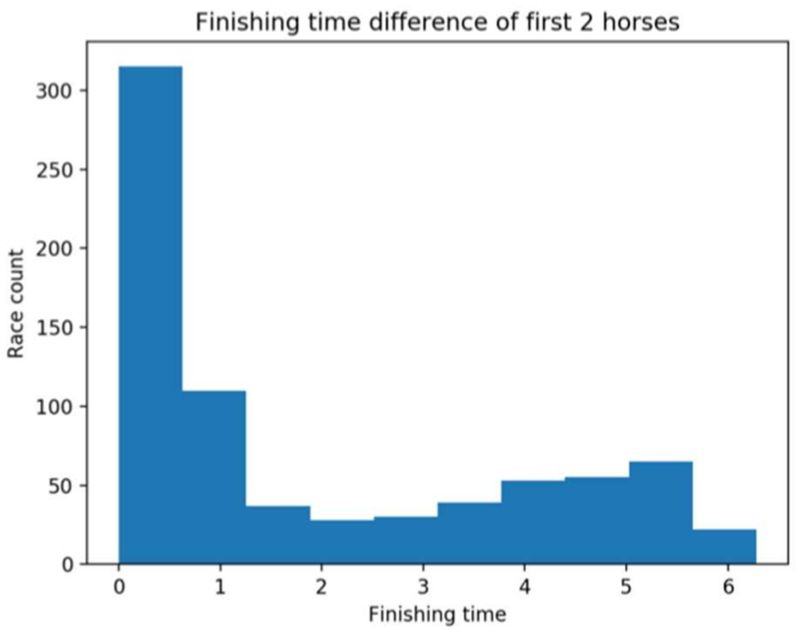
-14.25

Why?

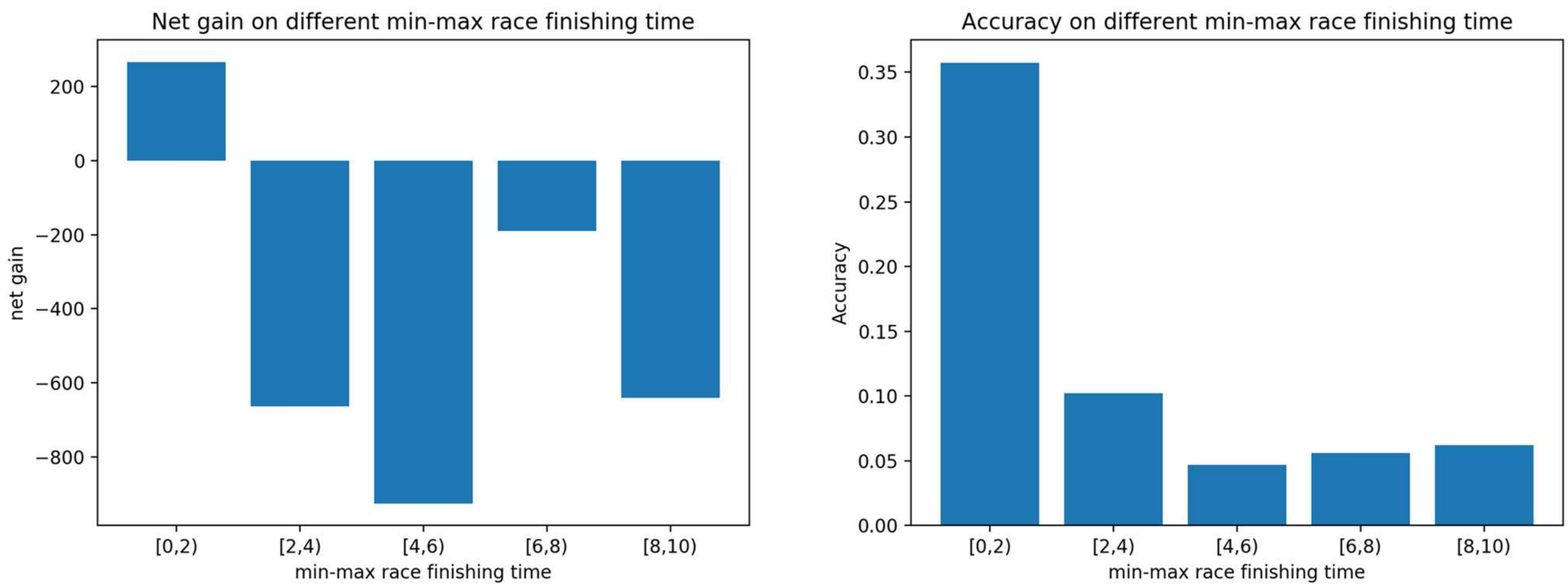
- Using Loss to evaluate a model hardly works
  - Finishing time is predicted individually
  - yet grouped together in a race
  - Loss is too simple to model the prediction results
- Confidence/Trend matters
  - imply the relative horse performance
  - Help lessen being influenced by randomness

Bet on best predicted races



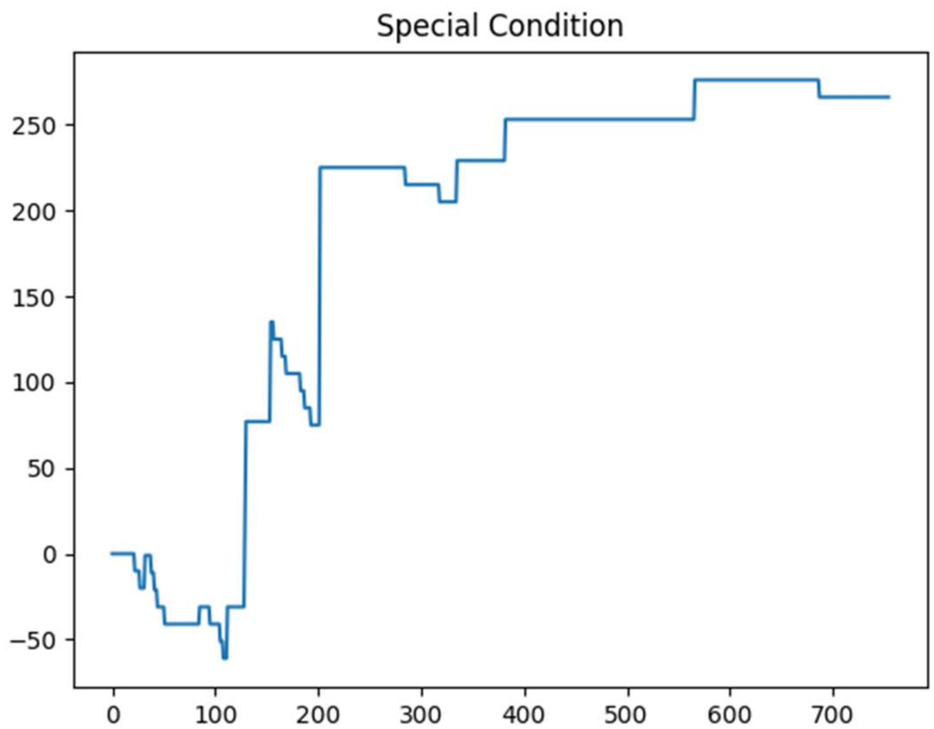


Bet on best predicted races



Net gain & Accuracy in different time intervals on training set (undivided)

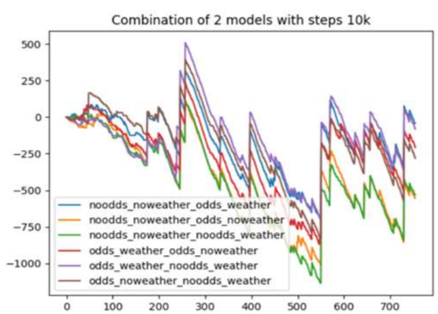
Bet on best predicted races



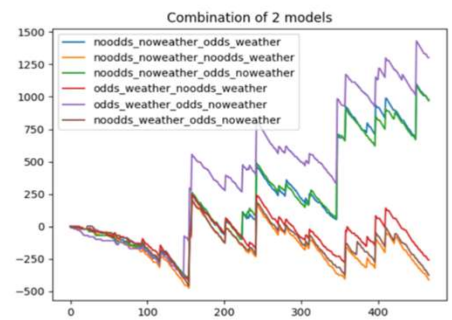
Using strategy on test set (undivided)

Confidence

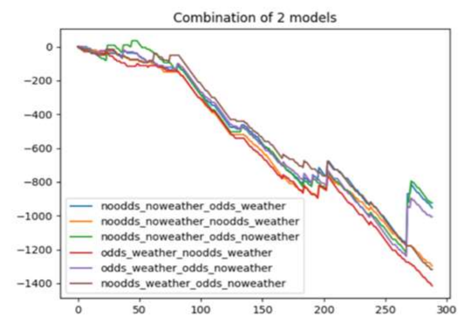
Combination of 2 models



Unvided



Shatin



Happy Valley

- Average Net gain:

  -530.8 -1030 325.32

- Average Net gain (Previously):

  -1006 -1306 -14.25

Future outlook

- Explore the best way to predict the results
- Build a more solid regressor in use

Future Outlook
Directions <u>In Progress</u>

- Investigate in depth on the relations between Loss(MSE) and our goal.
  - Models trained with 1m steps. (Overfit, increasing loss)
  - Models with regularizations (e.g. dropout) to minimize MSE

- Use average finishing time to regularize finishing time in a race
  - Combine our understandings on horse racing and model design
  - Test error (MSE) ≈ 0.59

Future Outlook
Goal

- Build a more solid system
  - Maybe Shatin racecourse
  - Maybe average finishing time

- Deploy models to train on individual horse records
  - Similar to markov chain
  - Where future state depends on current state (& past in this case)
  - Inspired by Prof. Gu wengao in STAT department

- Try other bets

5 Summary

Summary

- Horse racing prediction is not a traditional machine learning problem;
- Loss, accuracy and net gain are less related to each other than we expected;
- However, divide-and-conquer and apply the idea of confidence help improve the prediction.

Q & A