Horse-Racing Tipster Reliability Prediction

By

Anoop Kanigolla

Chakshu Bansal

Mehula Awadhiya

Sai Varaprasad Khandavilli

Introduction

A tipster is someone who regularly provides information (tips) on the likely outcomes of sporting events on internet sites or special betting places.

In the past tips were bartered for and traded but nowadays, thanks largely to the Internet and premium rate telephone lines, they are usually exchanged for money, and many tipsters operate websites. Some of them are free and some require subscription.

A tip in gambling is a bet suggested by a third party who is perceived to be more knowledgeable about that subject than the bookmaker who sets the initial odds. (A bookmaker will vary his odds according to the amount of money wagered but must start with a blank book and himself set an initial price to encourage betting.) Thus, a tip is not even regarded by the tipster as a certainty but that the bookmaker has set a price too low (or too high) from what the true risk is: it is a form of financial derivative, since the tipster himself risks none of his own money but sells his expert knowledge to others to try to "beat the bookie".

The Tipster must overcome the profit margin integrated into sports betting odds by bookmakers trading teams and then also obtain an additional edge to deliver profit over the long term.

Tipsters are sometimes insiders of a particular sport able to provide bettors with information not publicly available. There are other tipsters who provide equally respectable results through analysis of commonly accessible information.

Some tipsters use statistical based estimations about the outcome of a game and compare this estimation with the bookmaker's odds. If there is a gap between the estimate odds and the bookmaker’s odds, the tipster is said to identify "value", and a person who bets on such odds when they perceive not a certainty but a "gap in the book" is said to be a "value bettor". When value is found, the tipster is recommending the bettor to place a bet.

A tip that is a racing certainty, that is, almost completely certain to be true, is also called a nap and tipsters in newspapers will tend to indicate the "nap".

In the past tipping was mostly associated with horse racing but can apply to any sport that has odds offered on it. The relaxed cultural attitude towards gambling in the UK[1] is increasingly resulting in a gambling element being promoted alongside sport coverage in the media.

Problem Definition and Motivation

The traditional approach in attempting to make a profit from horse-racing, using machine learning techniques, is to use systems involving dozens and dozens of variables. These systems include the following types of variables:

Horse - Name, Sex, Age, Pedigree, Weight, Speed over various distances, race data with finishing times and positions - etc. Trainer info. Jockey info. Track info - Track, track conditions - etc.

Finding, compiling, maintaining, and updating this data is a massive task for the individual. Unless you have access to a database of such data - where would you even start?

The tipsters use their skill to study the horses and make a prediction - that they think a particular horse will win a particular race. We take those tipsters predictions and put them through a machine learning algorithm (Microsoft Azure) asking it to predict a 'win' or 'lose' based upon the tipster's performance history.

Data Description

The data set for this classification problem comes from the Horse Racing – Tipster Bets sample data set collection on Kaggle.

A general idea of the data can be gained by looking at the columns and their unique values.

UID - Unique ID. Not used in the actual system. The dataset is housed in an access database - the UID keeps all the bets in order. You may find a use for it!

ID - Each tipsters bets are kept in date and time order with the ID being incremented for each new bet, for each tipster.

Tipster - The name of the tipster. Each tipsters' bets are in order, followed by the next tipster's bets

Date - The date of the race. Previous experiments showed, to us anyway, that the date was not important, that it was the ID number that showed the system that each tipsters bets are a linear list. The date is still used as it has never been shown to affect results in a negative way.

Track - The name of the track. Some tipsters favor some tracks more than others and this affects their profit margin.

Horse - The name of the horse.

Bet Type - Is the bet a 'Win' bet or an 'Each Way' bet.

Odds - The odds that the tipster presenting the bet say they got for the bet. When you place a bet, you rarely get the predicted odds. Would the system be better served by lowering the odds by 10% to 20% which would be more realistic?

Result - Did the bet Win or Lose.

Tipster Active - Is the tipster active - true or false

	uid	id	tipster	date	track	horse	bet_type	odds	result	tipsteractive
1	1	1	Tipster A	2015-07-24	Ascot	Fredricka	Win	8	Lose	true
2	2	2	Tipster A	2015-07-24	Thirsk	Spend A Penny	Win	4.5	Lose	true
3	3	3	Tipster A	2015-07-24	York	Straightothepoint	Win	7	Lose	true
4	4	4	Tipster A	2015-07-24	Newmarket	Miss Inga Sock	Win	5	Lose	true
5	5	5	Tipster A	2015-07-25	Ascot	Peril	Win	4.33	Win	true
6	6	6	Tipster A	2015-07-25	York	Aldreth	Win	6	Lose	true
7	7	7	Tipster A	2015-07-25	Newcastle	Niceonemyson	Win	6	Lose	true
8	8	8	Tipster A	2015-07-25	Lingfield	Brandon Castle	Win	6	Lose	true
9	9	9	Tipster A	2015-07-26	Carlisle	Sands Time	Win	5.5	Lose	true
10	10	10	Tipster A	2015-07-26	Pontefract	Ad Dabaran	Win	2	Lose	true
11	11	11	Tipster A	2015-07-26	Uttoxeter	Milgen Bay	Each Way	10	Lose	true
12	12	12	Tipster A	2015-08-01	Thirsk	Gleese The Devil	Win	8	Lose	true
13	13	13	Tipster A	2015-08-01	Hamilton	Especial	Win	6	Lose	true
14	14	14	Tipster A	2015-08-02	Chepstow	Indian Affair	Win	6	Lose	true

Metrics for Model Evaluation

Various measures are used to evaluate the performance of the chosen models:

• **Feature weights:** Indicates the model's key features for generating predictions.

• **Confusion matrix:** Displays a grid of true and false predictions versus actual values.

• **Accuracy score:** Indicates the model's overall accuracy for both the training and test sets.

• **ROC Curve:** Shows a model's diagnostic ability by combining true positives rate (TPR) and false positive rate (FPR) for various class prediction thresholds (For example, churn thresholds of 10%, 50%, or 90% result in a prediction of churn)

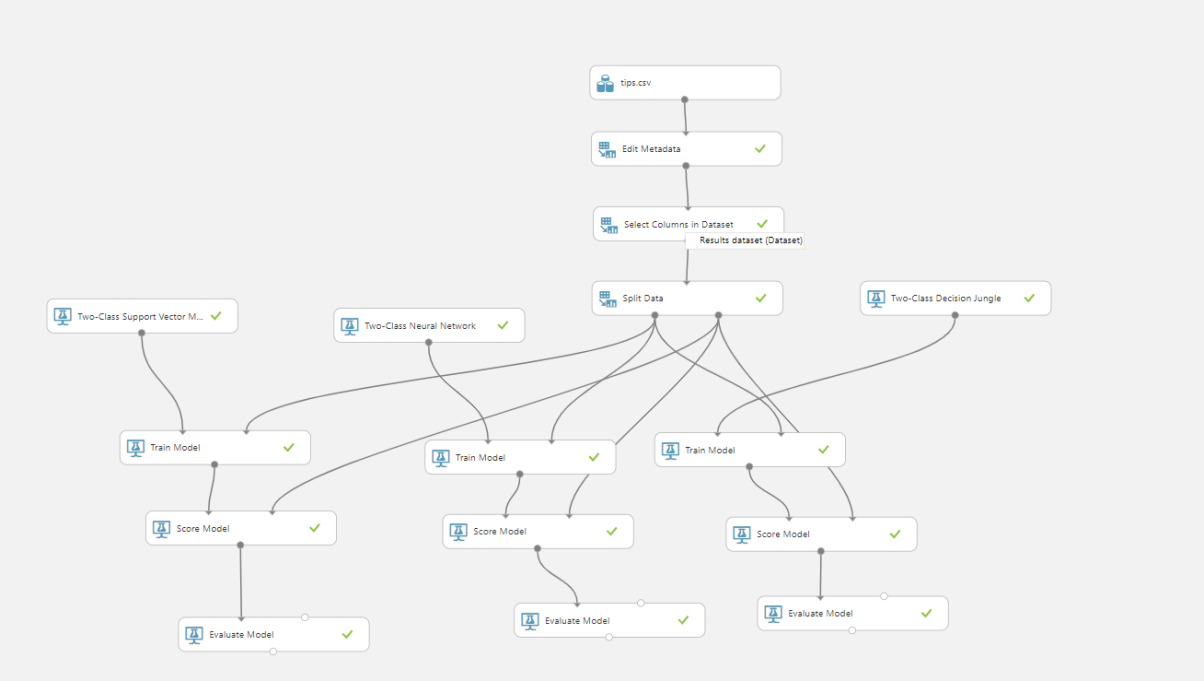
• **AUC (for ROC):** Indicates the model's overall separability between classes associated with the ROC curve.

• **Precision-Recall-Curve:** Compares the false positive rate (FPR) and false negative rate (FNR) for different thresholds of class predictions to demonstrate diagnostic competence. It's good for data sets with a lot of class imbalances (negative values overrepresented), because it concentrates on accuracy and recall, which aren't affected by the quantity of genuine negatives, hence it eliminates the problem.

• **F1 Score:** Calculates the harmonic mean of precision and recall and so assesses the trade-off between the two.

• **AUC (for PRC):** Indicates the model's overall separability between classes as measured by the Precision-Recall curve.

Predictive Model



In the beginning we tested out two models and measured their performance by several metrics. Those models have been optimized in a later step by tuning their hyperparameters.

The models used include:

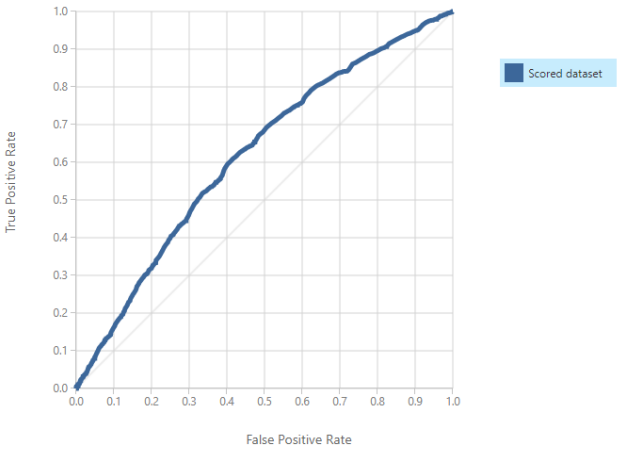
Logistic Regression: Logistic regression is a well-known statistical technique that is used for modeling many kinds of problems. This algorithm is a supervised learning method; therefore, you must provide a dataset that already contains the outcomes to train the model.

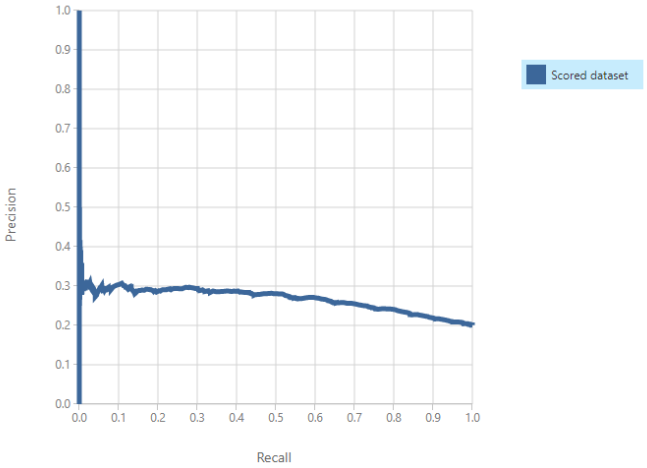
Neural Network: Despite the fact that the data set is minimal and that neural networks typically require a large amount of training data to have useful prediction capabilities, a rudimentary neural network is used to compare the two approaches.

Support Vector Mechanism*: Support vector machines (SVMs) are a well-researched class of supervised learning methods. This implementation is suited to prediction of two possible outcomes, based on either continuous or categorical variables.

Experiments

1. Two Class Support Vector Machine



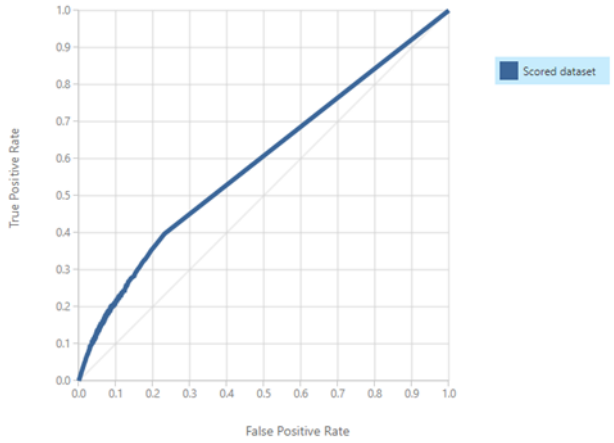


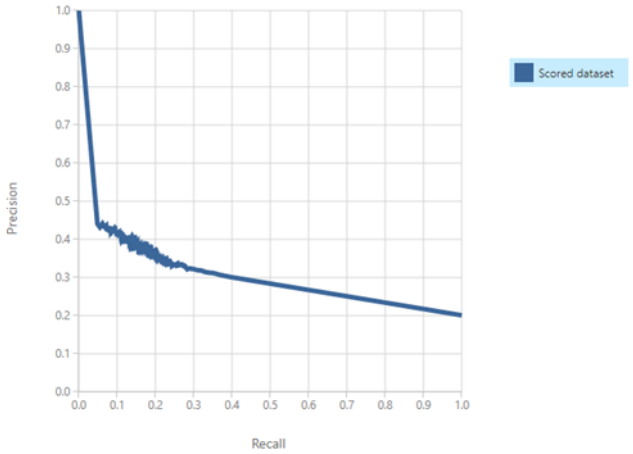
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
86	2219	0.788	0.291	0.5	0.619

False Positive	True Negative	Recall	F1 Score
210	8959	0.037	0.066

Positive Label	Negative Label
Win	Lose

2. Two Class Neural Network

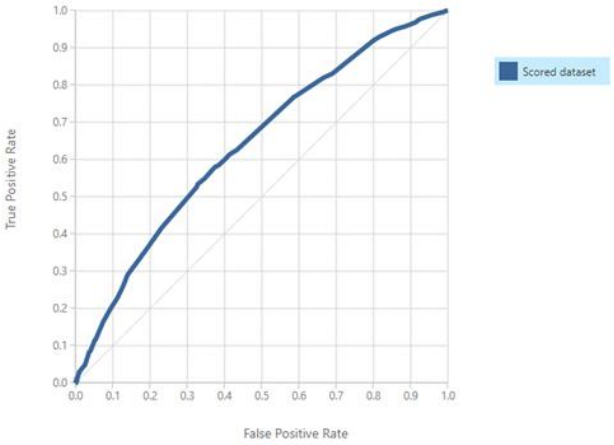


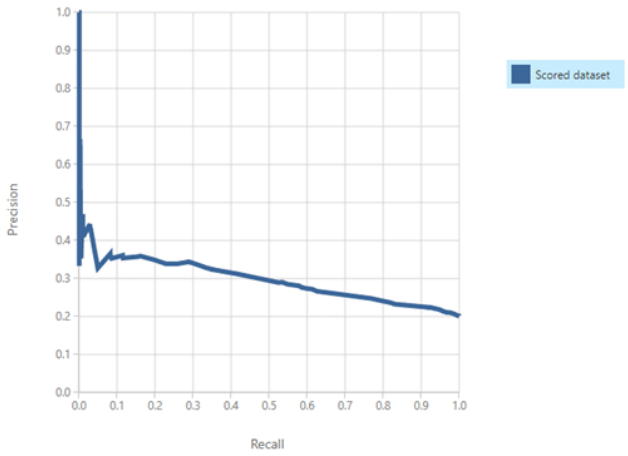


True Positive	False Negative
390	1915
False Positive	**True Negative**
645	8524
Positive Label	**Negative Label**
Win	Lose

Accuracy	Precision	Threshold	AUC
0.777	0.377	0.5	0.599
Recall	**F1 Score**		
0.169	0.234		

3. Two Class Decision Jungle





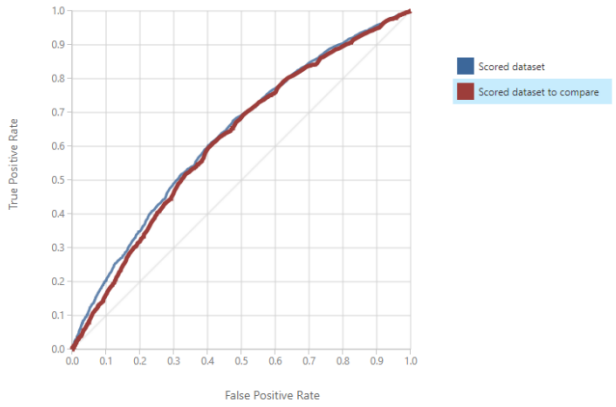
True Positive	False Negative
0	2305
False Positive	**True Negative**
0	9169
Positive Label	**Negative Label**
Win	Lose

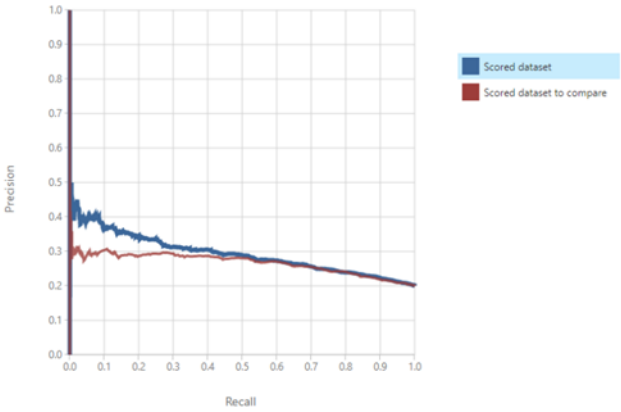
Accuracy	Precision	Threshold	AUC
0.799	1.000	0.5	0.639
Recall	**F1 Score**		
0.000	0.000		

Model Optimization/Hyperparameter Tuning

We are frequently unaware of the appropriate hyperparameter settings that will produce the best model output. As a result, we applied Hyperparameter tweaking, which improved precision and accuracy, as seen below.

1. Two Class Support Vector Machine (Optimized)

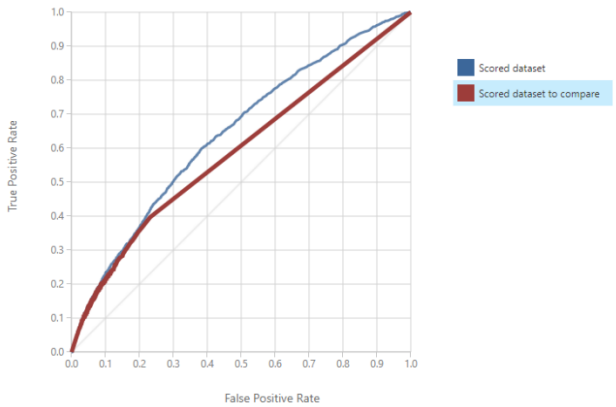


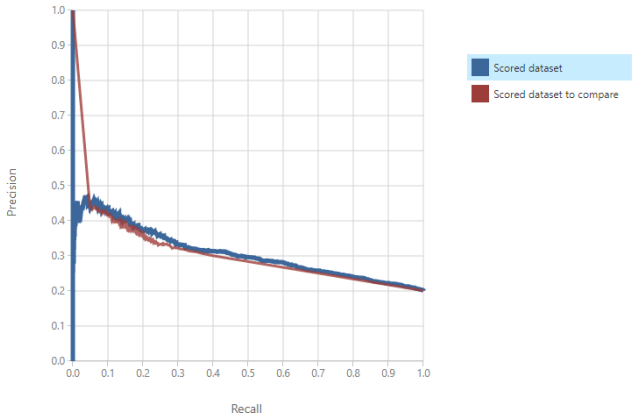


True Positive	False Negative
86	2219
False Positive	**True Negative**
210	8959
Positive Label	**Negative Label**
Win	Lose

Accuracy	Precision	Threshold	AUC
0.788	0.291	0.5	0.619
Recall	**F1 Score**		
0.037	0.066		

2. Two Class Neural Network (Optimized)

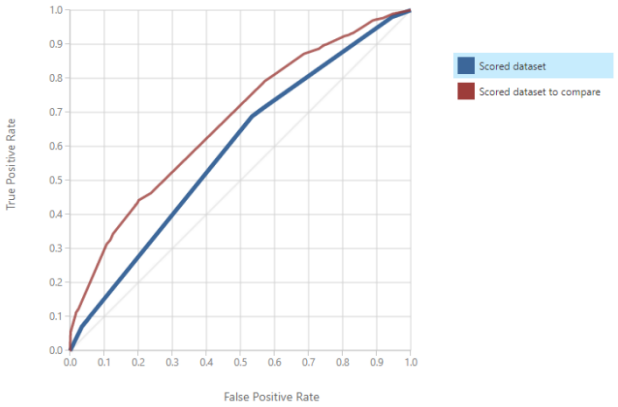


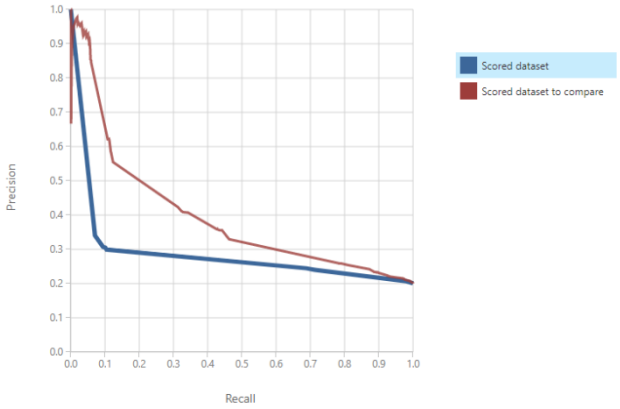


True Positive	False Negative
390	1915
False Positive	**True Negative**
645	8524
Positive Label	**Negative Label**
Win	Lose

Accuracy	Precision	Threshold	AUC
0.777	0.377	0.5	0.599
Recall	**F1 Score**		
0.169	0.234		

3. Two Class Decision Jungle (Optimized)





True Positive	False Negative
0	2305
False Positive	**True Negative**
0	9169
Positive Label	**Negative Label**
Win	Lose

Accuracy	Precision	Threshold	AUC
0.799	1.000	0.5	0.588
Recall	**F1 Score**		
0.000	0.000		

Comparing Models

Model	Accuracy	Precision	Recall	F1 Score	AUC
Two Class SVM	0.788	0.291	0.037	0.066	0.619
Two Class SVM (Optimized)	0.788	0.291	0.037	0.066	0.619
Two Class Neural Network	0.777	0.377	0.169	0.234	0.599
Two Class Neural Network (Optimized)	0.777	0.377	0.169	0.234	0.599
Two Class Decision Jungle	0.799	1.000	0.000	0.000	0.639
Two Class Decision Jungle (Optimized)	0.799	1.000	0.000	0.000	0.672

Conclusion

Based on the above findings, the Two Class Decision Jungle model after optimization has the best accuracy (0.799) on the test set.

Although there are a significant number of rows, the final numbers portray the biased nature of the dataset. This is highly due to the Tipsteractive column.

The bets from inactive tipsters are critical to performance and therefore highly affect the final numbers.

Recommendations:

Additional optimization efforts should be made in order to attain a higher score and, as a result, boost prediction power and hence increase user profits.

A plan can be created to retain tipsters as most tipsters are not opted for due to their inactive status, resulting in an increased false negative rate.

By attaining data from additional horse-racing betting sources, apply survival analysis technique, a technique to uncover scenarios that are not tracked, to estimate profits.