Using Meteorological Data Towards Horse Racing Prediction

Anthony Hein
Princeton University
anhein@princeton.edu

Abstract

In this paper, we explore whether the inclusion of meteorological data can improve predictions over the outcome of a horse race. To answer this question, we annotate a dataset of horse race with weather readings and develop models which leverage these features to predict pairwise winners in a race, which we then resolve into a full predicted ordering over the race. Both the use of such meteorological factors and pairwise prediction approach are novel with respect to the existing literature. We show pressure and temperature to have predictive power and establish the effectiveness of this pairwise prediction approach in its achieved accuracy of 78.89% on predicting the winner of a race.

1 Overview

The historic sport of horse racing, present even in ancient civilizations like that of Greece, is still widely-loved by fans across the world (Bell and Willekes, 2014). In fact, in Hong Kong, the racetrack is the city's largest taxpayer and benefactor, attesting to the popularity of the sport (HKJC, 2021).

In addition to being longstanding, the sport of horse racing is well-documented, as the allure of finding profitability at the racetrack has ensured that there is an abundance of horse racing data.

Yet, despite this abundance of data, there is a lack of academic literature that leverages this data, with only a handful of papers published on the topic in the last decade. However modern technologies are well-suited to analyze these datasets. We have already addressed the popularity of the sport, so it is unlikely that this dearth of literature can be explained by a lack of interest. Additionally, as betting on horse racing continues to flourish and as the available literature supports, horse racing is not a solved problem. In Appendix B, I advance the claim that this lack of literature is due to the proprietary nature of the datasets and controversy surrounding horse racing.

Regardless of the reason, the lack of academic literature on horse racing has left even the most basic questions about the different features determining a horse's performance – and thus the outcome of a horse race – largely unanswered.

Given that horse racing occurs outdoors, a seemingly natural question to ask about horse racing is the following:

> *Does weather have an effect on the outcome of a horse race?*

Towards this question, several articles across non-academic sources attest to the importance of rainfall, barometric pressure, temperature, humidity, and wind towards predicting a horse's performance in a race (CPR, 2021; Punter 2 Pro, 2021; Skymet Weather, 2021; The Whitley Group, 2021). However, these sources do not support their claim with concrete evidence, nor can a well-designed analysis of these claims be found in the academic literature. Therefore, this is left as an open question.

Yet, this question may be of interest to different parties. Foremost, this question may be of interest to jockeys, trainers, and owners, who have a direct stake in the performance of their horse. These parties are interested in information that helps them make decisions about which races to enter. If it is shown that race results are less predictable under non-ideal weather conditions or that races become more dangerous under non-ideal weather conditions, then these parties may abstain from races in such conditions to preserve their budgets and the health of their horses. Furthermore, if it is shown that a knowledge of the weather during a race can improve models that predict the performance of their horse, then these parties will want to use such models to understand what weather they perform poorly under so that they may alter their training regiment.

This question may also be of interest to racecourses, which attract customers by presenting entertaining races. If weather can improve models which predict the outcome of a race, then racecourses may use such models to determine what combination of horses will result in a dead heat that will have their viewers on the edge of their seat. Additionally, knowing which weather conditions make races dangerous for the runners can help a racecourse make wise cancellations before an accident occurs that may disturb their customers.

Next, this question may be of interest to bookmakers and sports bettors, who are involved in gambling on the outcome of a horse race. If using weather in horse racing models improves the performance of such models, then these parties may use such models to modify their beliefs about the outcome of a race. Whether a bookmaker or sports bettor, more accurate beliefs over the outcome of a race would yield more profit.

Still more, this question may be of interest to athletes in other sports. If it is shown that weather has a substantial effect on the outcome of a horse race, then athletes may want to consider how weather may be affecting their own performance and how they may adjust their practice to overcome the elements.

Finally, this question may be of interest to data scientists, purely for the reason that predicting the outcome of a horse race is a timeless and elusive problem that many consider to be "uncrackable", especially under the common practice of handicapping (see Appendix C). Any progress towards solving this problem, especially the discovery of heavily influential latent features, would be an exciting milestone and could provide inspiration on how to approach other challenging problems.

Therefore, given the interest of this question to a diverse array of groups, we will attempt to answer this through an application of data science techniques. In particular, this analysis will focus on determining if weather can improve current machine learning models which attempt to predict the outcome of a race, modeled as a classification problem. If weather has a considerable effect on the outcome of a horse race, then we will observe that machine learning models perform substantially better when they are given access to weather features than when they are not. On the other hand, if the effect of weather on the outcome of a horse race is minimal, then the incorporation of weather features is not expected to improve machine learning models.

The result of our analysis is that models do not show significant nominal improvement when given access to weather features, but nonetheless can be shown to meaningfully use some weather features such as pressure and temperature towards predicting the outcome of a horse race. Therefore, we conclude that some weather features may be able to improve current machine learning models. Additionally, a byproduct of this analysis is a highly-curated dataset of horses races from Ireland, where each race is annotated with four weather readings extracted from a nearby weather station accurate to a half-hour; no other dataset annotating horse races with weather information is known to be available. A second byproduct of this analysis is a novel approach to the horse racing prediction problem that reduces the problem to predicting pairwise winners; using this approach we create a model that achieves an accuracy of 78.89% in predicting the winner of a horse race on a dataset of horse races from Ireland, which thoroughly outperforms the public odds and compares well against the existing literature.

2 Horse Racing

A brief introduction to horse racing is provided in Appendix C for readers who are unfamiliar with the sport.

3 Related Work

As discussed above, the existing literature surrounding horse racing is sparse. Additionally, the majority of this literature is focused on creating a profitable betting strategy, instead of focusing on the prediction problem itself. Several papers try a new machine learning model or include newly engineered features. However, even these models still show predictive power at most 77%, which attests to the complexity of the problem. This has caused some to claim that the problem of predicting the winner of a horse race is "uncrackable" or that the winner of a horse race is nearly a random event (Torné, 2021). Yet, the widely popularized success of sports bettors like Bill Benter and the ability of some models to outperform the public odds reveal that there is room for improvement (Sankari et. al., 2021). Our analysis seeks to establish that further improvement can be achieved by the inclusion of meteorological data in such models and alternative approach to the problem using pairwise predictions. We discuss some of the existing literature revolving around horse racing prediction and the application of meteorological data to sports prediction in the sections to follow.

3.1 Searching for positive returns at the track: A multinomial logit model for handicapping horse races (Bolton and Chapman, 1986)

Some initial work was done towards predicting the winner of a horse race using simple decision rules which considered several features for a horse in a race and predicted whether or not this horse would win the race (Vergin, 1977). However, these were largely unsuccessfully due to the competition between horses in a race that changes the way any given horse would run in isolation (Bolton and Chapman, 1986). Instead, a 1986 paper by Bolton and Chapman omits the implicit assumption that a horse would run the same with or without other horses present by using a multinomial logit model to solve the multiclass classification problem of which horse will be the winner for a race. That is, the input to the model is a feature vector which contains the same set of features across all runners in a race. The model then outputs a probability distribution which is each runner's predicted likelihood of winning the race, from which the predicted winner is selected. For a given horse, there were 10 features in total, including measures such as the lifetime winning percentage of the horse, lifetime winning percentage of the jockey, weight of the horse, and average speed rating. However, a limitation may be the lack of any reference to the weather during the race. Another limitation of this analysis is that the dataset used is extremely small by modern standards, consisting of only 200 races. Perhaps the small dataset explains why, despite this relaxed assumption on the independence of runners during a horse race, Bolton and Chapman's model yields just a 3.1% return when applied to betting.

3.2 Computer Based Horse Race Handicapping and Wagering Systems: A Report (Benter, 1994)

Drawing much influence from the work of Bolton and Chapman, famous sports bettor Bill Benter published a 1994 paper which claimed that computer systems can definitively “beat the races”, with the witness to his claim being the financial success of his own model when used to bet on Hong Kong races over a period of five years (Benter, 1994). During this time, his predictions led him and his team to achieve a wealth which was over forty times their initial capital. This was achieved using a wide array of features involving the horse, jockey, and trainer. While this undoubtedly establishes the proof of concept for successful models, the drawback to Benter's approach is its computational complexity; instead of being neatly output by a single model, predictions are obtained from an amalgamation of several highly-tuned models that require several persons to operate (Benter, 1994). Thus, there is a need to create a model that is simple, yet powerful. We may hope that the inclusion of weather features allows us to achieve similar results with fewer resources, where weather is curiously absent from Benter's otherwise prolific list of features; in a section where the author groups factors that have predictive significance for a race, there is even a group denoted ”preferences which could influence the horse's performance in today's race” under which factors that concern weather would naturally fall, yet are not listed (Benter, 1994). Indeed, our analysis will attempt to show whether or not this omission hinders the author's model.

3.3 A case study using neural networks algorithms: horse racing predictions in Jamaica (Williams and Li, 2008) *and* Horse racing prediction using artificial neural networks (Davoodi and Khanteymoori, 2010)

With the rise of neural networks, we see several authors resurrect this prediction problem in the late 2000s. One example is a 2008 paper by Williams and Li, which uses a neural network architecture that accepts as input a feature vector of a single horse and predicts its finishing time, which is then used to predict the winner of a race. Although this method makes the questionable assumption that horses run independently of each other during a race, the authors boasts an impressive accuracy of 74% on a dataset of horse races in Jamaica (Williams and Li, 2008). Another example by Davoodi and Khanteymoori uses the same architecture towards a dataset of horse races in New York to achieve an accuracy of 77% (Davoodi and Khanteymoori, 2010). Additionally, Davoodi and Khanteymoori are the first authors in the literature to use a measure of weather in their feature vector, though it appears as an extremely limited categorical variable taking on one of a few values (e.g. “clear”, “cloudy”, “showery”). Despite the apparent success of these two papers, we maintain some skepticism about their architecture design decisions due to the extremely small sample size of about 100 races in each analysis; the results learned by these models may not be generalizable if they are learning noise about the limited population of jockeys and horses in their dataset. Of the literature surveyed, these results offer the best accuracy for horse race prediction and they suggest that neural networks present a promising approach.

3.4 Ga Yau: Machine analysis of Hong Kong horse racing data (Torné, 2021)

More recently, the availability of larger datasets, whether by purchase or by webscraping, have permitted the use of more sophisticated neural networks with a richer feature space. One such application of a more sophisticated neural network is due to Torné and was completed just one year prior to this analysis. Following the advice of Bolton, Chapman, and Benter, the neural network architecture that Torné uses accepts an input which has features for all horses, where this input is padded with blanks for races that have fewer runners (Torné, 2021). For each horse, Torné uses thirty-three features that primarily involve the past performances of the horse, the jockey, and the trainer of that horse, where he claims the included features to be directly inspired by a publication on how to pick a winning horse by the Hong Kong Jockey Club (Torné, 2021). This results in 462 nodes in the input layer alone. For his analysis, Torné purchased a large dataset comprised of 12,820 races from Hong Kong; recall that the datasets used in prior analyses did not exceed a few hundred races (Torné, 2021). Torné uses races that occur strictly prior to a cutoff date to train his model, while reserving races that occur on or after this cutoff date for evaluating his model. This choice of a training and testing set ensures that the model, upon receiving a test datapoint, has not trained on any race which occurs in the future with respect to a test datapoint; of course, knowledge of a future race is not representative of the real world. Furthermore, in the event that the underlying distributions governing these races change over time, this division naturally focuses the prediction problem on the most recent distribution, which is incidentally the distribution that is of most interest to all involved parties. Therefore, Torné’s analysis best represents modern machine learning practices and so will be a primary influence of our own analysis.

One critique of Torné’s neural network architecture is that the resulting model ends up having 32,534 trainable parameters, which far exceeds the number of datapoints in his dataset. Therefore, it can be conjectured that this model would be very susceptible to overfitting, and indeed this seems to be observed when his model outperforms the public odds more on the *training* set than on the *testing* set. While a counterargument to this critique may be that the predictive power of the public odds have improved through the years, and recall that the testing set is drawn from more recent races, this counterargument is questionable since the public odds are a feature to Torné’s model, and so the improvement of the public odds should similarly improve the model. Another critique is Torné’s use of placeholder *UNK* tokens when a feature was unavailable, since neural networks do not accept incomplete input (Torné, 2021). Since it is unclear what the effects of this seemingly ad hoc practice may be on the model, we will try to avoid this in favor of design choices we can better explain. Finally, as is the theme in the literature, weather is only indirectly included through features that consider the course the race was run on or the track condition of the race. One of our contributions to the literature will be making weather a more explicit feature drawn from a richer set of values.

3.5 Weather and horse racing: Towards a more objective prediction of the going[1] (Sheridan and Sweeney, 2001)

Shifting our focus to weather in horse racing, Sheridan and Sweeney (2001) aimed to predict the *track condition* in advance of a race using antecedent meteorological conditions. The authors state how important the track condition is to a trainer or owner’s decision whether or not to enter their horse in a given race, since the track condition can “explain the difference between a horse winning and losing a, perhaps very lucrative, race” (Sheridan and Sweeney, 2001). Using a multiple regression analysis, the authors obtain fair success, with multiple correlation coefficients across different racecourses between 0.56 and 0.78. While Sheridan and Sweeney (2001) do not try to predict the outcome of a horse race, their use of a racing dataset

[1] The “going” is a term for the track condition.

and meteorological data from Ireland draws a nice parallel to our own analysis, which will also focus on data from Ireland, though for unrelated reasons. Additionally, its acknowledgement of weather as an explanatory variable for the outcome of a race inspires more interest in our own analysis, where we will attempt to make this connection between weather and horse racing more explicit.

3.6 The use of machine learning in sport outcome prediction: A review (Horvat and Job, 2020) *and* The Effect of Weather in Soccer Results: An Approach Using Machine Learning Techniques (Iskandaryan et al., 2020)

Zooming out to see the bigger picture, we can find two papers in the recent literature that attest to the importance of meteorological conditions towards predicting the outcome of a sporting event. Horvat and Job support the claim that motivates the analysis in this paper when they argue, "Predicting sport outcomes is a complex problem due to the range of uncertainties that can, even during the game, influence the final game outcomes, such as a player's injury, weather conditions, in-game tactical changes, and so on" (Horvat and Job, 2020). Iskandaryan et. al. take this one step further in asking what the effect of weather is on soccer results, using Spanish soccer matches and meteorological data to answer their question (Iskandaryan et al., 2020). Much as our analysis intends to do, for each soccer match in their dataset they find the nearest meteorological station at the time of that match and append its weather data to their soccer dataset. Then, they survey several different classification models to compare the accuracy of models with and without this meteorological data at the task of predicting the outcome of a soccer match. Their results clearly show that the incorporation of meteorological data significantly improves the model's ability to predict the outcome of a soccer match. This method has not been used in the context of horse racing before, allowing us to apply this method to fill this gap.

3.7 Summary of Related Work

In summary, the related work frequently nods to the importance of weather in predicting the outcome of a horse race through the inclusion of features like the track condition in various models but does not use weather features explicitly. Given the acknowledgement of the importance of weather towards the track condition and thereby the race, as well as the result of the importance of weather towards predicting the outcome of a soccer match, it seems plausible that our analysis may reveal the lack of meteorological data to hinder the previous models in the literature. Additionally, with modern machine learning practices that have already demonstrated some success, we are well-suited to conduct this analysis.

4 Methodology

We now outline the approach that will be taken in this paper to answer the research question of whether meteorological data can improve horse racing models. A discussion of the finer details of this approach are deferred to the sections to follow.

First, similar to Iskandaryan et al. (2020), we will collect two large datasets: one containing horse racing information and another containing meteorological data. By matching racetracks to nearby weather stations just as Iskandaryan et al. (2020) have matched stadiums to nearby weather stations, we will supplement each race with weather readings such as temperature, humidity, pressure, and rainfall. In following the advice provided by Benter (1994) and the Hong Kong Jockey Club regarding features used in prediction, we will conduct extensive feature engineering to build out a vector encoding the past performance of a runner at the time of a race. To keep our input vectors short and thus prevent overfitting, as we have suspected to occurred in the analysis by Torné (2021), we will *not* use an architecture that accepts all horses of a race at once and predicts the outcome of the race. Instead, we will use an architecture that accepts two runners from a race at a time and predicts which will finish before the other. The predictions over such pairwise matchups will then be resolved to determine a winner of the overall race using a method of our own design. Such an architecture keeps our feature space small while still avoiding the strong assumption that a horse should run independently of all other horses in a race that several authors in the literature caution against. Finally, we will ablate the trained models by removing features involving weather and comparing the resulting accuracy to determine the value of weather towards this prediction.

Therefore, the novelty of this paper is threefold:

1. The creation of a highly-curated dataset containing over 20,000 horse races from Ireland

which are annotated with temperature, humidity, pressure, and rainfall. To the best of our knowledge, no such dataset is available elsewhere. Additionally, with the exception of that used by Benter (1994), this dataset is the largest used in an analysis of horse racing.

2. A classification model for horse racing that explicitly considers a wide range of meteorological factors, aside from the typical use of extremely limited categorical variables.

3. The approach to train models to predict the pairwise winner over two horses in a race, then aggregating these predictions to predict the winner of a horse race. Reducing the problem of predicting the winner of a race to predicting the winner of a pair of horses is not found in the existing literature.

5 Data

This analysis depends on a dataset which combines horse racing data and meteorological data. In this section, we discuss the process by which we create such a dataset.

5.1 Acquiring Raw Data

An extensive search revealed no available dataset fulfilling the outlined requirements. As is consistent with the discussion of related work, available horse racing datasets had at most a single categorical variable for encoding the weather or the condition of the track, which is loosely a function of the weather. Therefore, it was quickly apparent that this analysis would require the creation of a new dataset by merging available datasets.

5.1.1 Acquiring Horse Racing Data

First, we discuss the selection of horse racing data. Although horse racing occurs worldwide, the most prolific data is available from horse racing in Europe and Hong Kong. Notably, due to the proprietary nature of these datasets, horse racing data in the United States is heavily guarded behind a paywall and so is inaccessible for the scope of this analysis.[2] There are two substantial datasets of horse racing data from Hong Kong available on Kaggle. However, neither contains the start time of each race, which means that any analysis with these datasets would only be able to attach *daily* weather data to each race, where we might otherwise prefer to attach more fine-grained *hourly* weather data to each race.

Finally, we find Nikolay Kashavkin's dataset on horse racing, which contains an assortment of races from several different countries and was obtained by scraping publicly available results across several websites (Kashavkin, 2020). Most importantly, this data includes the time of each race, the racetrack of each race, the finishing position of each horse in each race, and the public odds on each horse in each race. The time and racetrack of each race is required to annotate the race with accurate weather information. The position of each horse is required because this is the target for prediction. Finally, the public odds are important because they serve as a baseline. This dataset spans 30 years of horse racing from 1990 to 2020 and so can serve as a source for historical analysis as well as modern analysis.

One drawback of this dataset is that it does not have the finishing *time* of each horse, but rather only the finishing time of the first place finisher for each race and then the distance from each horse to its predecessor, with the units of distance being the length of a horse. Using these two quantities, we can estimate the finishing time of *any* horse in a race, and so this does not hinder the usability of this Kashavkin's dataset. A discussion of how the finishing time of a horse was estimated is included as Appendix D.

A complete description of the format of this dataset can be found in Appendix E.

5.1.2 Acquiring Meteorological Data

Now, we discuss the selection of meteorological data. To keep the analysis tractable and avoid inconsistent weather-recording practices across different countries, we decide to focus on only one country. As shown in Figure 1, within the horse racing dataset, the most races occur in Great Britain, Ireland, France, the United States, and South Africa. So we want to find meteorological data for one of these countries. Also recall that we want this meteorological data to be at the most granular level possible; weather conditions several hours from the start of a race may not be representative of the weather conditions during the race.

Although an early version of this project at-

[2] As a point of interest, a developer who created an application for scraping horse racing data in the United States was sent a "cease and desist" letter and forced to remove his application from GitHub (source: https://www.thoroughbreddailynews.com/getting-from-cease-and-desist-to-come-work-with-us/).

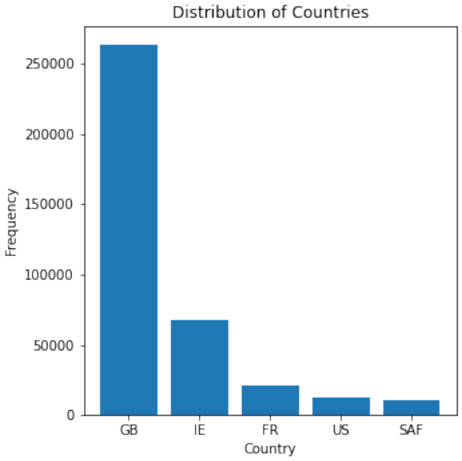

Figure 1: Distribution of countries within the dataset by (Kashavkin, 2020).

tempted to conduct this analysis for horse racing in Great Britain, the meteorological data from Great Britain ultimately proved to be difficult to work with. A discussion of this attempt is included as Appendix F.

Therefore, we shifted our focus to horse racing in Ireland. To this end, we found the website of the Met Eireann, the Irish Meteorological Service, which offers datasets of weather in Ireland measured hourly at several stations (Met Eireann). Sheridan and Sweeney (2001) from the related work actually use the same meteorological service towards predicting the track condition at a horse race. This website offers one file of data per station. Furthermore, although each station in the Met Eireann database offers a slightly different set of measurements, all stations record temperature, humidity, pressure, and rainfall, which are claimed to be the most important by various non-academic sources. We additionally download a file of station metadata which records the opening year, closing year, and latitudinal and longitudinal coordinates of each station. A complete description of the format of this dataset can be found in Appendix G.

5.2 Scrubbing Data

Having selected the raw datasets used in each case, we proceed to cleaning the data. In particular, this largely concerns removing rows with values that do not make sense given their context or rows with missing values. The horse racing dataset underwent extensive data scrubbing, a technical discussion of which is left to Appendix H.

One important point to discuss here is missing values. After correcting the entries we could in the horse racing dataset, scrubbing still required us to drop around 7% of the starting horse racing dataset, so we may be concerned about injecting bias into the analysis. The bias that would be injected due to dropping these entries depends on the reason that these values in these entries were missing. This can be problematic if this reason pertains to attributes of the horse or the race the horse ran in. For example, suppose that missing values exclusively occurred in races where there was a strongly projected winner, perhaps because the record-keeper assumed that no one would want or need these values since the outcome seemed obvious at the time. In this case, the missing information is the very valuable input the record-keeper necessarily possessed at one point which explained *why* the outcome of the race was so predictable. Clearly, if this were the case, then the data left after dropping entries with missing values would very much be a function of the race itself, rather than a random sample of the true distribution of races, as we would hope it to be, thus hindering the generalizability or trustworthiness of the results.

However, two more innocuous explanations for this missing data is that data may be missing because:

1. The respective race occurred several decades ago such that the record-keepers back then didn't have the tools, language, or conventions to collect clean data. Or, clean data has since been lost over the years due to wear and tear.

2. The race involves a high number of runners such that the limited attention of a small number of record-keepers trying to keep up with the fast-paced racing day prevents them from collecting all the data related to a race.

In the case of the former explanation and under the additional assumption that the distribution of races and horses does not change substantially throughout the years, sampling more heavily from some years than others does not change the distribution we end up with. If that assumption does not hold and the distribution of races and horses changes throughout the years, this is still not a problem because the ultimate goal is to predict future races instead of past races, and so sampling more from more recent years helps achieve this goal. In the case of the latter explanation, the dataset we end up using for training and testing will be biased to have a greater proportion of races with fewer runners than is actually observed in practice, and so predicting the outcome may be slightly easier. However, since we will measure the performance of any models we train against baselines rather than absolute values, this is not actually an advantage and if our model achieves good performance with respect to the baseline this can still be impressive regardless of the exact composition of our dataset.

Having shown these two intuitive explanations for missing data to be innocuous, we can proceed by testing these hypotheses. Indeed, as Figure 2 shows, both hypotheses can be supported by the data. That is, the data shows that the average number of missing values per horse per race decreases over time and the average number of missing values per horse per race increases with the number of runners in a race. Echoing the previous explanation, this can be interpreted to mean that the quality of the data improves when there are fewer runners to worry about and when technological advances permit better collection and preservation of this data. In conclusion, the means by which we have scrubbed the dataset to remove entries is not expected to invalidate the results of this analysis.

On the other hand, the meteorological dataset was well-formed, and so scrubbing this dataset simply involved dropping the few records which contained missing data.

5.3 Combining Datasets

Having scrubbed the datasets, we can now proceed to combine the racing data and meteorological data. Recall, metadata available from the Met Eireann provided the latitudinal and longitudinal coordinates for each of the weather stations available, as well as the open year and close year of each station.

No such equivalent exists for the tracks present in the horse racing dataset; instead, for each race, regarding positional information, we only had the *name* of the track on which the race occurred and the country code of this track. Therefore, we had to use Google's Geocoding API to infer the position of each of the tracks. To automate the process, we programmed a client in Python to send a query to the API with the following format:

“`<racetrack name>` racetrack”

After receiving tentative geographic coordinates for each racetrack, we manually reviewed these results to confirm their accuracy and filled in any racetracks containing erroneous or null results.

With these latitudinal and longitudinal coordinates, it was now possible to calculate the distance between any racetrack and any weather station using the haversine formula. This is an alternative to using Euclidean distance and is meant to be more accurate, at the cost of added computational complexity. If the number of point-to-point distances necessary for a future analysis makes the haversine formula a bottleneck, we would recommend the use of Euclidean distances.

Next, we faced a design decision of whether, for a given race, to select meteorological data from the station that was open and nearest in proximity to the race or from the station that had the most recent reading with respect to the starting time of a race. We decided to optimize over distance, similar to what was done in the related literature (Sheridan and Sweeney, 2001; Iskandaryan et al., 2020). However, using the alternate approach of taking the most recent reading (and breaking ties by distance) often yields the same result anyways, due to the fact that the meteorological data was synced to the same time intervals.

Therefore, to annotate a race with weather, we first found the open weather station closest to the track then found the most recent weather reading. Some care had to be taken to convert all times to the same timezone. Additionally, to keep the computation tractable, we had to precompute a list of the nearest weather stations for each track. Then, after identifying the nearest open station, we used binary search within the list of weather readings for that station, leveraging the fact that these weather readings were sorted by date and time.

Before proceeding, we can inspect the “goodness” of the weather readings we have annotated the races with. First, for each race, we can measure the distance between the track this race is run on and the weather station being used. A histogram of these distances is displayed in Figure 3. The closest reading is taken only 3.79 kilometers from the race, approximately the distance from the Princeton Bendheim Center for Statistics and Machine Learning to Princeton Junction. The median distance to a station is 32.5 kilometers, approximately the distance from Princeton to Trenton and back. Finally, the farthest reading is taken from a station which is 99.9 kilometers from the race; this is approximately the distance from Princeton to Philadelphia and back. To further visualize the distances over which we are accessing meteorological

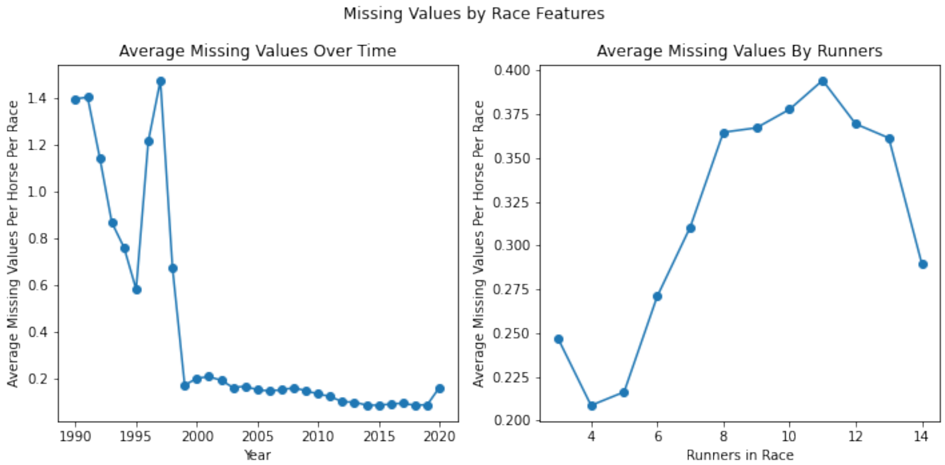

Figure 2: (Left) It is shown that the average missing values per horse per race generally decrease over time. Such a trend can be explained by the racecourses obtaining better equipment for data collection as the technology advances. (Right) It is shown that the average missing values per horse per race generally increases when there are more runners in a race. Such a trend can be explained by the difficulty of collecting all the desired data as there are more moving parts in a race.

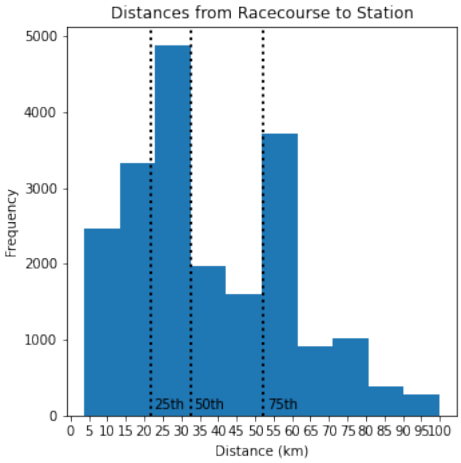

Figure 3: Distribution of distances from racecoures to the weather stations from which the readings will be used. Dotted black lines mark the quartiles of the distribution. Fifty percent of weather readings are taken from a station that is 32.5 kilometers away. The distribution is right skewed, meaning that few races use weather data from extremely distant stations.

data from, we can plot racecourses and weather stations on a map and draw edges between a racecourse and a weather station if some race run on that racecourse is annotated with meteorological data from that weather station. A random sample of twenty of the twenty-eight racecourses were selected and the plot constructed exactly as explained is presented as Figure 4. There seems to be a shortage of weather stations in the southwest corner of Ireland, which perhaps explains the long distances observed in Figure 3, and so these readings may be slightly noisy. However, on the east cost near Dublin, we see that the racecourses are very near to the weather stations, and so we expect these readings to be highly accurate. Although the weather readings are imperfect, even approximate measurements of the temperature, humidity, pressure, and rainfall at the time the race occurs are still a significant improvement over that previously used in the existing literature.

Secondly, for each race, we can measure the time between the race and the weather reading. The result of this is overwhelmingly positive, with 98.6% of the weather readings having been recorded within a half-hour of the race. Given our use of hourly data, we may suspect that the only reason a weather reading may not have been recorded within a half-hour of the race is if the selected station was experiencing a power outage or was undergoing maintenance during this time. Given the small number of races which did not meet the criterion of

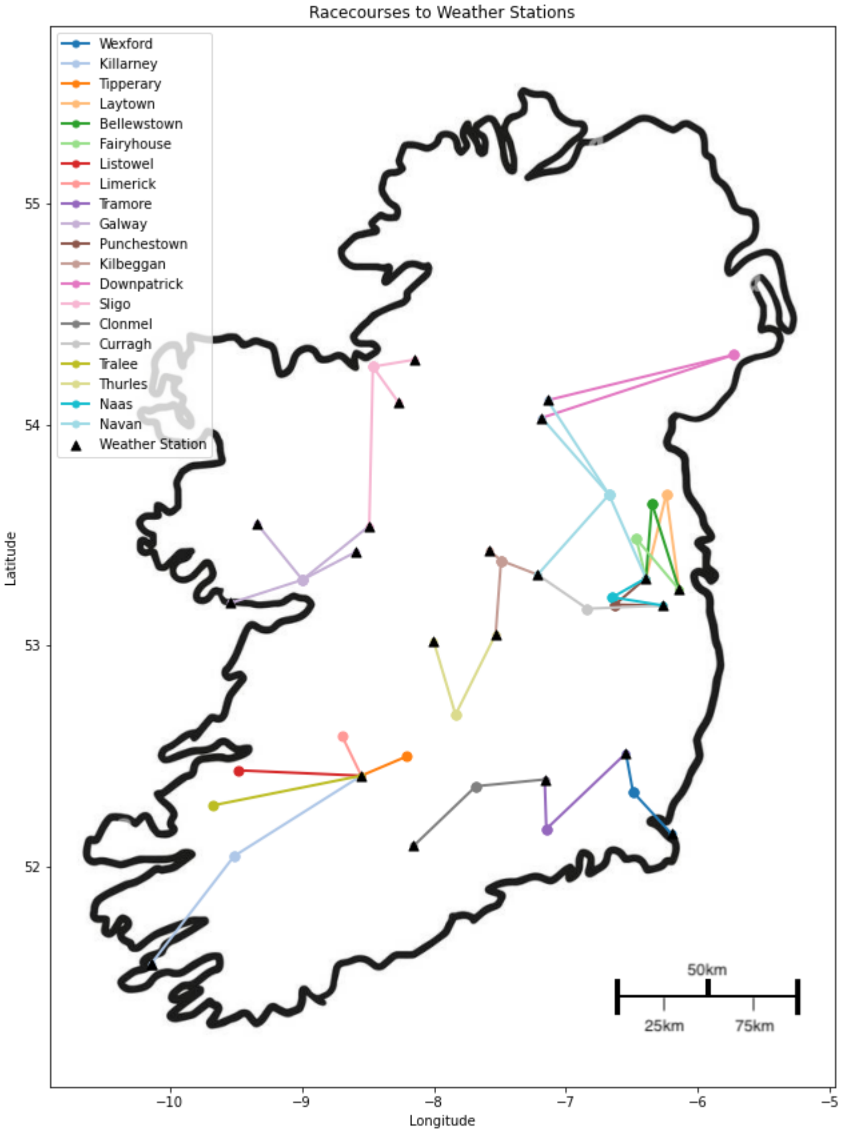

Figure 4: Graph of a random sample of twenty out of the twenty-eight racecourses and weather stations from which races run on these racecourses extract meteorological data. A racecourse may take data from several weather stations since closer weather stations may shut down forcing a racecourse to take data from a farther weather station or a closer weather station may open so that a racecourse can get more accurate readings. This graph is *approximately* to scale.

having weather data accurate to the half-hour, we decided to drop such races from out dataset (1.4%).

The "goodness" of the collected weather readings will be defended further in Section 6 as we conduct exploratory data analysis.

5.4 Summary of Data

The process of acquiring, scrubbing, and combining horse racing and meteorological dataset has been explained in depth. The result of this is a novel dataset of non-hurdle horse races from Ireland containing three to fourteen runners with temperature, humidity, pressure, and rainfall readings. These datasets are made available on the accompanying GitHub, detailed in Appendix A.

6 Exploratory Data Analysis

Having constructed our dataset, we pause prior to analysis to conduct some exploratory data analysis on our dataset. First, we construct a pairplot over the dataset of runners to identify which features may be correlated with the finishing position of a horse and visualize the distribution of columns we hope to use as features. This is included as Figure 5. As expected, the decimal odds and RPR of a horse – both ratings intended to measure the "goodness" of a horse – are the most correlated with the finishing position of a horse, and correlated with each other. That is, a horse that is favored by the public odds will be similarly favored by the racing post ratings. Additionally, we see that the handicap and the public odds are related; again this makes sense since horses that are favored by the public odds are handicapped with metal weights to even out the race. The last observation from this plot is that the distribution of positions reveals that there are fewer races with a larger number of horses. For this reason, we must be careful to nuance our results since it may be inherently easier to predict on races with fewer horses.

Having looked at the dataset at a glance, we will now zoom in and show several intuitive relationships in the data.

6.1 Public Odds Decently Predict the Winner of a Horse Race

First, we want to make sure that the public odds make sense. Since the public odds are the most readily digested piece of information available to all parties involved in horse racing and represent the collective beliefs of the masses, we hope that the public odds may serve as a baseline for our analysis, similar to Torné (2021). If this baseline performed abysmally on the dataset, we would have to carefully consider whether we trusted the dataset or whether there may have been an error in the public odds or race outcomes during its compilation.

Fortunately, the public odds do indeed serve as a suitable baseline, achieving an accuracy of 37.1% across the entire dataset and even larger accuracy when conditioned on races with fewer runners. This well exceeds the performance of an alternative baseline which simply picks a horse uniformly at random as its prediction, thus achieving accuracy 1 / (# number of runners). The accuracy of this baseline as the number of runners in the race varies is included as Figure 6.

6.2 Dangerous Track Conditions Occur when it Rains

Now, we wish to check that rainfall data we have collected from weather stations decently explains the track conditions observed during the race. If this were not the case we again may be skeptical that there could be errors in the dataset.

First, we will reduce the twelve categorical track conditions found in our dataset into four more digestible levels, which are firm, good, soft, and heavy (in increasing order of danger). That is, it is least risky to run on a firm track and most risky to run on a heavy track where the ground is comparable to mud. Then, we will split our dataset by whether or not we have recorded it to be raining during this race and plot a bar graph to visualize the proportion of each type of track condition. This bar graph is included as Figure 7. This graph shows that, conditioned on it raining, the track condition is more likely to be soft or heavy and less likely to be firm or good, which supports our claim. This is especially apparent at the extremes where 22% of races without rain are run on firm ground while only 10% of races with rain are run on firm ground. At the other end, only 11% of races without rain are run on heavy ground while 19% of races with rain are run on heavy ground.

The reader may be curious how any race where it is raining can be run on firm ground. While we maintain the possibility that some of our weather readings are inaccurate, this can also be explained by the fact that constant and careful maintenance of the track by members of the racecourse mitigates the erosion caused by rain. So, it is possible

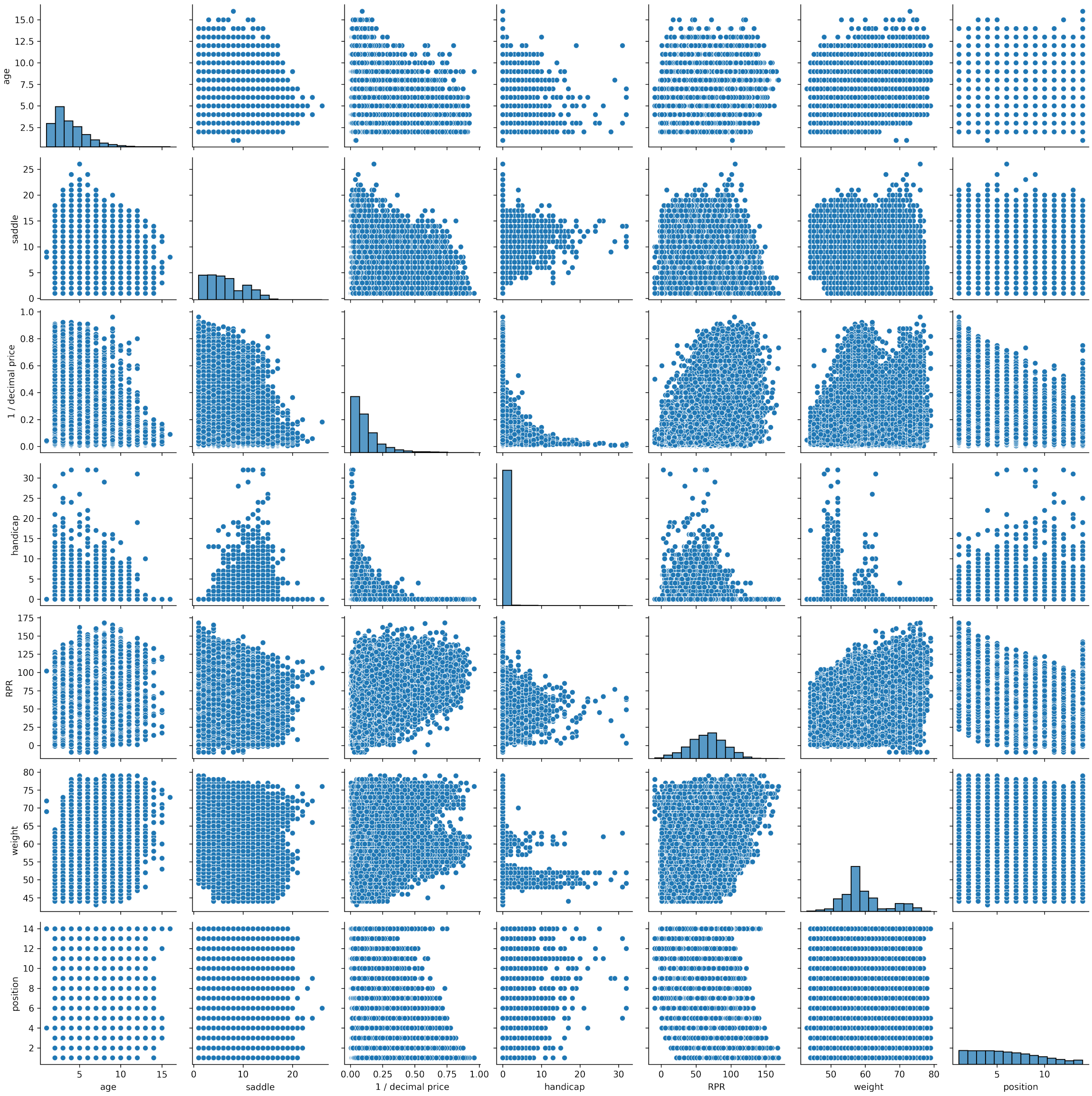

Figure 5: A pairplot showing the distribution of some attributes of a horse as well as the joint distribution of these attributes with other attributes, most notably the finishing position of a horse. This shows that the inverse of the decimal odds and RPR are most correlated with the finishing position, as expected.

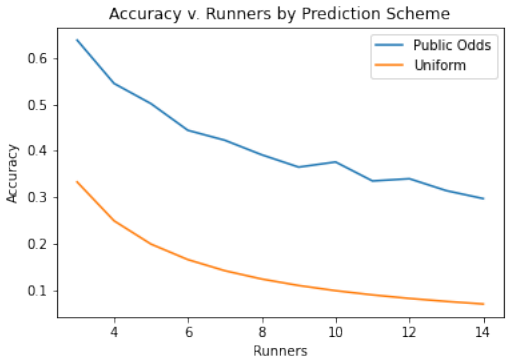

Figure 6: A baseline predictor which selects the horse with the most favorable public odds achieves fair accuracy, especially when used to predict the winner for races with fewer runners. This baseline vastly outperforms the baseline which selects a horse uniformly at random.

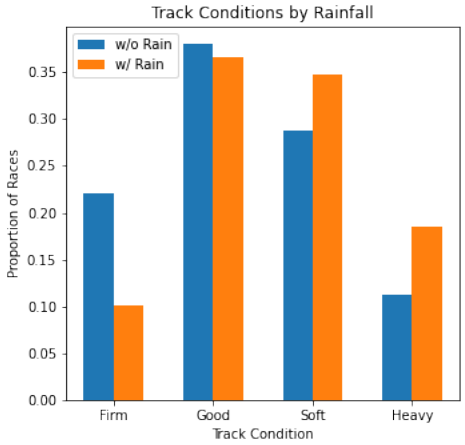

Figure 7: The proportion of races with each of four simplified track conditions, with color used to split the dataset into those races with and without rain.

that softer track conditions may only prevail when the rain is substantial or persistent. In any case, this preliminary result inspires confidence in our dataset.

6.3 Projected Winners Place Higher Under Ideal Weather Conditions

Next, we want to test the hypothesis that projected winners – i.e. those horses selected by the public odds – will place better under ideal weather conditions. This is in part inspired by a claim found on a blog post that "when the weather is very harsh, some horses will win by pure chance, beating better and faster horses" (The Whitley Group, 2021). The exact definition of ideal weather conditions is deferred to Appendix I, though is essentially a race having a moderate temperature, moderate humidity, high pressure, and no rainfall.

Accordingly, we can split our dataset into two types of races – races under ideal weather conditions and races under non-ideal weather conditions – and measure whether there is a significant difference between the placement of project winners in each dataset.

The results of several two-sample t-tests for comparing population means are included in Table 1 and a visualization of the underlying distributions are included as Figure 8. In all cases, we find that the projected winners perform better on average when the weather conditions are ideal versus when they are non-ideal, and furthermore that these results are statistically significant. This can also be interpreted to mean that under non-ideal weather conditions, "longshots", loosely defined as horses not projected to win, are more likely to win. This makes sense, since we would expect non-ideal weather conditions to inject variance into the race and thus even out the playing field for the runners involved.

6.4 More Horses Fail to Finish Under Non-Ideal Weather Conditions

A horse may fail to finish a race due to a catastrophic event in which they get injured. Also, a horse may fail to finish a race if their jockey pulls up on the reigns prior to the completion of a race if the jockey deems it unsafe or unwise to continue running. In both cases, we can easily see how non-ideal weather conditions may make each situation more likely since we expect non-ideal weather conditions to push horses to their physical limits, make the track unfavorable, or obscure visibility. Us-

Measure	Ideal Weather	Non-Ideal Weather	t-stat	p-value
Average position of horse with best odds	2.88	2.98	-3.18	0.001
Average position of horse with 2^{nd} best odds	3.63	3.73	-2.85	0.004
Average position of horse with 3^{rd} best odds	4.16	4.36	-5.11	$3.17 \cdot 10^{-7}$

Table 1: On average, horses with better odds place better (i.e. obtain a nominally lower finishing position) under ideal weather conditions than under non-ideal weather conditions. These results hold with statistical significance at $\alpha = 0.05$ using a two-sample t-test with samples of size 10520 (ideal weather) and 9681 (non-ideal weather).

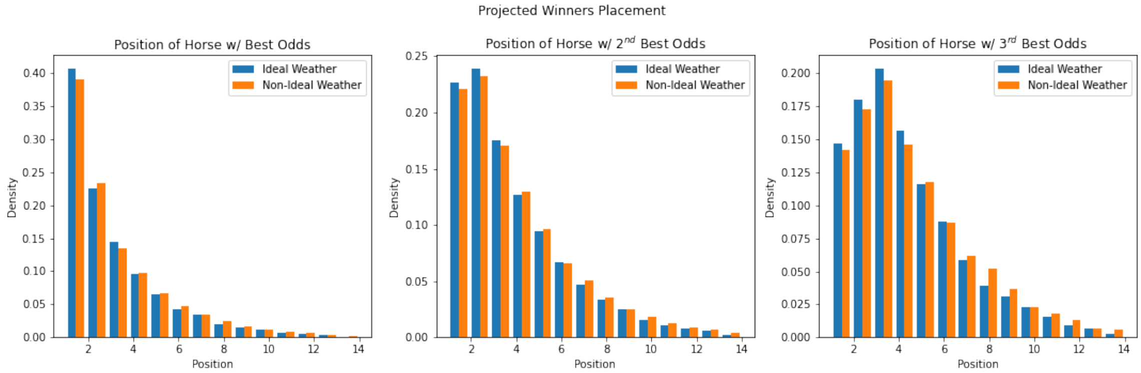

Figure 8: The distributions of finishing positions for the top three projected winners, with color used to split the dataset into those races with ideal weather conditions and those races with non-ideal weather conditions. As can be seen, under ideal weather conditions the distribution has more mass to the left than under non-ideal weather conditions in all cases.

ing the same definition of ideal weather conditions and the same partition of the dataset, we can test this hypothesis that more horses fail to finish under non-ideal weather conditions.

We find that during ideal weather conditions, on average 0.142 horses fail to finish per race while during non-ideal weather conditions, on average 0.155 horses fail to finish per race. Although this mildly supports our hypothesis that more horses fail to finish during non-ideal weather conditions, the associated t-stat of -1.46 yields an insignificant p-value of 0.144. One possible explanation we can have as to why this result is not more pronounced is that jockeys may pull up on the reigns after the first three finishers cross the line just to give the horse a break, even when there is no imminent danger in finishing the race.

Nonetheless, we are able to pick out a particularly interesting datapoint, which is the Boolavogue Mares Maiden Hurdle occuring on June 7^{th}, 2017. The weather information we have collected reveals that it was raining heavily during this race, with 3.1 millimeters of rain in the past hour, and that the recorded humidity was extremely high. Of the fourteen runners, only six finished, with seven jockeys pulling up and one horse falling prior to the finish line. Although this singular example cannot sustain the hypothesis posited by this section, it is interesting to be able to explain this otherwise confusing race outcome using the obtained meteorological data.

6.5 Winning Times are Higher Under Non-Ideal Weather Conditions

As one last check to the validity of the dataset, we may expect winning times to be higher – i.e. horses to run slower – under non-ideal weather conditions. To be more concrete, it is easy to see that when the track beneath the horses is soft they will not be able to push off of the ground as hard and thus cannot propel themselves as far. Alternatively, given non-ideal weather conditions, the horse and jockey may adjust their speed to ensure that they do not overexert themselves or otherwise injure themselves.

In Appendix J we detail how to obtain a "normalized winning time" to allow for comparison even given that races are run on different distances. Then, we can proceed by doing a two-sample t-tests on the mean normalized winning time across races with ideal weather conditions and the mean normalized winning time across races with non-ideal weather conditions. The mean normalized winning time of races with ideal weather conditions is -0.12 (recall, this is a normalized value so it can be negative). The mean normalized winning time of races with non-ideal weather conditions is 0.13. This

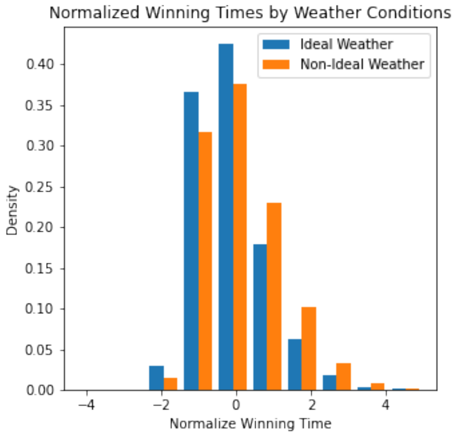

Figure 9: The distribution of normalized winning times, with color used to split the dataset into those races with ideal weather conditions and those races with non-ideal weather conditions. Winning time are higher under non-ideal weather conditions, showing that horses run slower in races run under non-ideal weather conditions.

yields a t-statistic of -17.20 ($p \approx 0$) which is *extremely* statistically significant at $\alpha = 0.05$. The distribution of the normalized winning times split by weather conditions shown in Figure 9 furthers the claim that there is a clear increase in winning times when weather conditions are non-ideal.

6.6 Summary of Exploratory Data Analysis

It has been shown that the public odds indeed serve as a reasonable baseline for our analysis. Additionally, we are able to support, with a high degree of statistical significance, several intuitive hypotheses that we would expect to hold in our dataset if the attached meteorological data was accurate. Several of these findings can be summed up as having shown that there is considerably more variance in races with non-ideal weather conditions than in races with ideal weather conditions. In other words, as the weather conditions stray further from what is normal or expected, the public odds seem to yield poorer results. At a high level, this seems to support the importance of weather towards the prediction of the outcome of a horse race, though we will continue investigating this question by building out machine learning models in the following sections.

7 Featurization

Having defended the suitability of our dataset through an exploratory data analysis that supports several intuitive hypotheses about the data, we now seek to develop models which use this data to predict the outcome of a race. A prerequisite to such analysis is the featurization of the data to make it more readily consumable by a machine learning model.

7.1 Horse Featurization

First, we would like to featurize the horse dataset, which the reader will recall has rows which are runners in a race. This featurization should enable us to preserve some identity of the horses, especially their affinity for different race or weather conditions. Intuitive proxies that use the horse name, jockey name, trainer name, or past performance of a horse in different conditions are infeasible with our dataset, as described in Appendix K.

Instead, we use the past performance of a jockey as a proxy for horse identity. Trainers and owners select which jockey will ride their horse, often using the same jockey across different horses they own. Thus, we would expect a jockey to ride horses that follow similar training regiments, have similar diets, run in the same class of races, and are approximately the same speed. Even if a jockey's past performances is only loosely correlated with the past performances of a horse, this choice still seems reasonable since jockeys are influential to the outcome of a race and are likewise affected by weather conditions. Therefore, a model may learn that some *jockeys* perform better than others under certain weather conditions. Clearly, if a model is able to leverage the fact that some jockeys perform better than others under certain weather conditions, then this still answers our research of whether meteorological data can be used to improve predictions over a horse race.

It is much more feasible to featurize using the past performance of a jockey rather than the past performance of a horse. This is primarily due to the fact that the population of jockeys is much smaller than the population of horses, with only 2,660 unique jockeys compared to 50,291 unique horses. As a result, on average, a jockey has more prior races than the horse on which it is running. Figure 11 shows the distribution of races run per jockey and shows that, although there are still a number of jockeys with a single race (25% of all jockeys), there are a greater proportion of jockeys with substantially more races that make past performance features possible. Using the past performance of a jockey allows us to achieve the de-

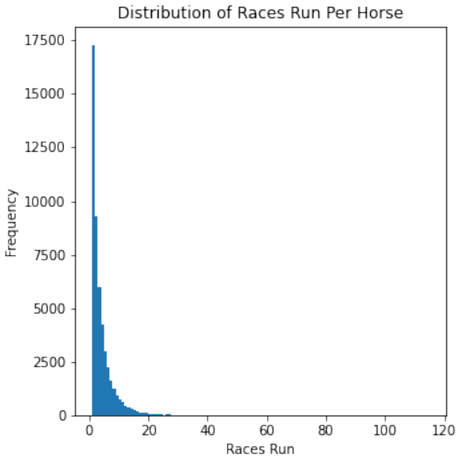

Figure 10: The distribution of races run per horse. Most horses in the dataset run very few races, which makes using the past performances of a horse as features infeasible since these past performances frequently won't exist.

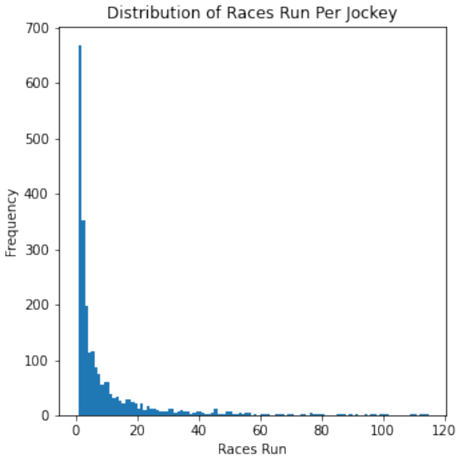

Figure 11: The distribution of races run per jockey. This should be compared to Figure 10, where it will become clear that there is a greater proportion of the histogram at larger values of races run. Note that this distribution has been clipped at 115 to match the scale of Figure 10, omitting 7% of jockeys which have run a number of races exceeding 115.

sired features while conserving most of the data whereas using the past performance of a horse towards the same features results in a substantially thinner dataset. Therefore, we proceed with this featurization.

An enumeration of the features that we have engineered based on the past performance of the jockey is shown in Table 2. again punting on the definition of similar weather conditions once more. These features are largely based on those used by Torné (2021), which in turn is based on professional advice from the Hong Kong Jockey Club. All features involving weather are novel to this project and most directly pertain to our research question.

As some extra commentary to motivate these features, `d_last_race` and `d_first_race` encode the freshness and seniority of this jockey respectively. The three unconditioned previous race features encode past performance in the most general sense. Finally, the conditioned previous race features allow a potential model to select whichever high-level race attributes or weather conditions are the most important, where we will be particularly interested with respect to our research question to see if those involving weather conditions are selected.

Finally, we will concatenate features involving past performance of the jockey with the available horse attributes outlined in Table 3 to create our final feature vector for a runner in a race, totaling 72 dimensions.

7.2 Race Featurization

Now, we turn the discussion to how we featurized the dataset containing metadata on each race, such as the distance the race is run on, what course the race occurs at, and the weather under which the race is run.

Most of these variables are categorical variables with few levels that can easily be one-hot encoded, where this one-hot encoding is important for linear models that otherwise would be unable to learn relationships between the levels. Unless explicitly mentioned, this is the approach taken to featurize such variables in this dataset. Some extra care is taken to featurize the datetime of the race, explained in Appendix M.

Next, we wish to featurize the weather features. We chose to featurize the temperature by placing a temperature in one of five bins. The choice of 5

Feature	Description
d_last_race	Number of days since the jockey's last race.
d_first_race	Number of days since the jockey's first race.
prev_[1/2/3]_position	Finishing position of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race.
prev_[1/2/3]_finishing_time_ratio	Finishing time ratio of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race.
prev_[1/2/3]_global_finishing_time_ratio	Global finishing time ratio of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race.
prev_[1/2/3]_[position/finishing_time_ratio]_course	Finishing position/ finishing time ratio of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race on the same course.
prev_[1/2/3]_[position/finishing_time_ratio]_metric	Finishing position/ finishing time ratio of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race on the same distance.
prev_[1/2/3]_[position/finishing_time_ratio]_ncond	Finishing position/ finishing time ratio of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race on the same track condition.
prev_[1/2/3]_[position/finishing_time_ratio]_runners	Finishing position/ finishing time ratio of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race with the same number of runners.
prev_[1/2/3]_[position/finishing_time_ratio]_month	Finishing position/ finishing time ratio of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race in the same month.
prev_[1/2/3]_[position/finishing_time_ratio]_temp	Finishing position/ finishing time ratio of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race in a similar temperature.
prev_[1/2/3]_[position/finishing_time_ratio]_msl	Finishing position/ finishing time ratio of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race in a similar barometric pressure.
prev_[1/2/3]_[position/finishing_time_ratio]_rain	Finishing position/ finishing time ratio of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race in a similar rainfall.
prev_[1/2/3]_[position/finishing_time_ratio]_rhum	Finishing position/ finishing time ratio of jockey in their $1^{st}/2^{nd}/3^{rd}$ most previous race in a similar humidity.

Table 2: All the features related to the past performance of a jockey in a race. Brackets and forward slashes are used to group together features which are similar, meaning that there are actually 65 such features altogether. A technical definition of the "(global) finishing time ratio" is left to Appendix L while the interpretation of similar weather is provided in Section 7.2

Feature	Description
age	The age of the horse in years at the start of the race.
saddle	The saddle number of the horse.
decimalPrice	The *reciprocal* of the decimal odds for a horse (larger means more favored to win).
isFav	Boolean whether the horse has the (weakly) largest decimal odds of all horses in the race.
outHandicap	Amount of additional weight (possibly zero) placed on the horse.
RPR	Rating by racecourse officials, similar to the odds except not determined by the public.
weight	Weight of the horse (in kilograms).

Table 3: All the features related to a horse in a race. There are 7 altogether.

bins is intentional and is meant to mimic the five words we commonly use to describe a temperature, namely "cold", "cool", "just right", "warm", and "hot". We set the bins to be of equal width within the range of temperatures. We similarly follow this convention of 5 levels to featurize the pressure and humidity, with the semantic interpretation again being "extremely low", "low", "just right", "high", and "extremely high". Finally, we featurize the amount of rainfall by making a binary variable of whether it is raining or not, since the boundary on the distribution of rainfall posed by no rainfall makes this variable unlike the other weather variables, which more naturally follow a normal distribution.

The distributions resulting from this binning strategy are shown in Figure 12. These distributions mostly match our intuition, where bins towards the center usually have more datapoints than bins at the extremes, with the exception being pressure where there is a long left tail.

Finally, with this idea of binned weather conditions, we can define what it means for a race to be run under similar weather conditions as another. A race under a similar temperature as another race will have a temperature reading belonging to the same bin; the definition follows similarly for pressure, humidity, and rainfall. This definition of similar weather conditions is motivated over alternatives in Appendix N

We now reiterate how this featurization is novel with respect to the prior literature's treatment of weather. Firstly, even though the weather features are binned for use in the definition of similar weather, we still maintain the numerical values separately for potential use. Secondly, by having three weather features take one of five bins and the last weather feature take on one of two bins, this yields a total of 250 distinct weather combinations, which vastly outnumbers the levels on the categorical weather features used elsewhere, and so our definition of similar weather is sufficiently expressive. Still more, our weather variables are the direct result of objective numerical readings, rather than subjective observations (e.g. "cloudy" used in Davoodi and Khanteymoori (2010)). Lastly, we make explicit the relationship between a jockey's performance and the weather during a race.

7.3 Prediction Target

At this point, we have feature vectors for each runner in a race and for each race itself. Now, we consider what the target for prediction in our analysis will be, which will in turn motivate how to

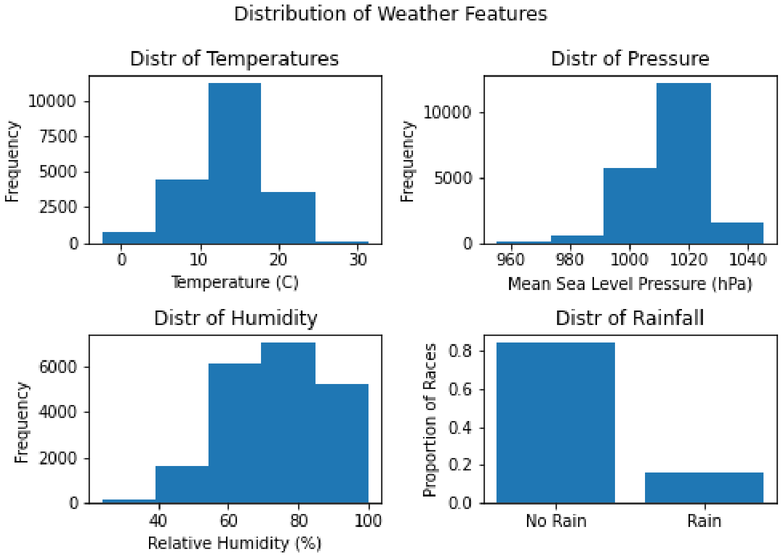

Figure 12: The distribution of the included weather features, visualized by the bins used to featurize them.

combine these available vectors to create input vectors for our prospective model. The input to our model will be the concatenation of feature vectors derived in Section 7.1 for a *pair* of horses in a race. Then, the target of prediction will be the earlier finisher of the pair, referred to as the "pairwise winner". Since this represents a novel approach, in Appendix O we explain why the two most popular existing approaches were deemed unsuitable for our purposes. In the paragraphs to follow, we motivate and explain our own approach.

Again, our approach will predict which runner will finish first given a pair of horses in a given race. Foremost, this addresses a concern of large, possibly padded, variable-length input vectors since the input to this model will only require the concatenation of two vectors for runners in a race. Next, this addresses our concern of a shortage of data since the number of pairs within a race is quadratic in the number of runners. Additionally, this approach is extremely flexible to instances where some runners in a race have feature vectors and other runners in this race saw their feature vectors dropped due to missing values. In this case, these runners can still be used to create pairs as input to the model. Predictions over these pairs are still meaningful since that allow us to infer *some* ordering on runners in a race, even if we cannot make claims about runners for which we do not have feature vectors. For example, suppose that for a race between runners X, Y, and Z, the feature vector for runner Z is absent but we have predicted X to finish before Y. Then, even without any knowledge of Z, we know that Y probably will not win, which is valuable information. Still more, this approach relaxes the strong assumption that horses run completely independently of one another since we are considering horses pairwise. Note that under this approach there is still a *looser* assumption that two horses X and Y under consideration will run independently of a third horse Z; perhaps horse Z horse motivates horse X to run faster than horse Y, where it otherwise would not have. However, if we perform this prediction across several pairs and aggregate the results, then we can loosen this independence assumption even further, and so this is not a severe limitation. Finally, this addresses our concern of simplicity since it presents a binary classification problem: the output is either that the first horse will finish before the second horse (represented by a target of 1) or that the second horse will finish before the first horse (represented by a target of 0).

This approach still allows us to answer our research question: if meteorological information is important to the outcome of a race, then the inclusion of such information will permit a model to achieve higher accuracy at predicting pairwise winners, compared to a model which does not have such information.

To further motivate this approach, consider how such a model may be used downstream to predict the winner of a race with full information. This can be done by using the model to predict on each unique pair of horses in the race and aggregating the output pairwise winners to select a winner of the *entire* race. Note, we are not claiming that the model has to be trained or tested on races with full information – it has been argued that this is not necessary. This will be revisited in Section 8 and is included here only to justify the suitability of this approach.

Some additional technical details of this approach are discussed in Appendix P.

7.4 Summary of Featurization

In summary, the result of featurization is an extensive dataset of ordered pairs of runners in a race. To be exact, this pipeline results in 1,143,824 such vectors that span 18,591 unique races, of which full information is available for 4,584 races. These input vectors are of length 144, with an equal number of dimensions contributed by the first and second runners and primarily encoding the horse's attributes or jockey's past performance. The target for these input vectors is a binary value which is 1 if the first horse finished before the second horse and 0 otherwise. This approach still allows us to predict the winner of a race and more importantly, answer our research question of whether meteorological data can improve predictions over the outcome of horse races.

8 Analysis

As discussed in Section 4, the main analysis of this paper will build out several classification models that, given a pair of horses in a race, will predict the pairwise winner. We will then perform an ablation experiment by removing features which concern past performance of the runner under similar weather conditions from the dataset and retraining each model. Finally, we will compare the performance across the models with and without these features involving weather to answer the research question posed by this paper.

8.1 Train-Dev-Test Split

First, we split the dataset into a training set, development set, and testing set. We will use the training set to train models, the development set to select appropriate hyperparameters, and the testing set only once to evaluate the final model. We will select our training set to be about 70% of all data, the development set to be about 20% of all data, and the testing set to be about 10% of all data.

The normal approach to achieving this split of the initial dataset is to draw simple random samples of the desired size. However, such an approach is unsuitable for our dataset, given how the dataset is featurized to include past performance. This is more fully explained with examples in Appendix Q.

Instead, we will split the dataset chronologically, as recommended by Torné (2021). In other words, we will split the dataset such that all races in the training set come chronologically before those in the development set, and all races in the development set come chronologically before those in the testing set. We used the desired sizes mentioned earlier to motivate the choice of cutoffs in this process. The resulting training set contains races from September 14^{th}, 1996 to July 27^{th}, 2016. The resulting development set contains races from July 28^{th}, 2016 to July 23^{rd}, 2019. Finally, the resulting test set contains races from July 24^{th}, 2019 to December 5^{th}, 2020. This translates to 800,666 paired inputs in the training set, 228,766 paired inputs in the development set, and 114,392 paired inputs in the testing set.

In Appendix R we discuss the possibility that the resulting sets may come from slightly different underlying distributions, but ultimately determine that this is still the best way to achieve our split.

8.2 Model Evaluation

Now that we have divided our featurized dataset into a training set, development set, and testing set, we are ready to begin creating machine learning models. Prior to our ablation experiments, we may ask the question whether a model with these weather features can do well in general. If the answer to this question is in the affirmative, then it makes sense to proceed with the ablation experiment, since any change in the performance of the model has meaningful consequences. If the answer to this question is in the negative, it is less interesting whether or not this model does better with weather features, since the model performs poorly overall. Poor model performance would suggest that there are other more important latent features, and so undermines the focus of this exploration, instead highly motivating a pivot in our analysis.

Note that false positives and false negatives have no particular semantic interpretation by how we constructed our featurized dataset. Therefore, the discussion to follow will primarily focus on accuracy as an evaluation metric.

Hyperparameter	Value
Epochs	25
Batch Size	256
Learning Rate	0.02
Optimizer	Adam
# Hidden Layers	1
# Hidden Nodes / Layer	150

Table 4: Chosen hyperparameters for the neural network that uses *all* features to predict pairwise winners.

To demonstrate that a model can be performant, we will train a neural network. This choice is because the presence of non-linear activation functions and hidden layers make neural networks highly-expressive, such that if any model is able to achieve high performance at this task, a neural network probably can. Conversely, if a neural network cannot achieve high performance at this task, *no* model can likely achieve high performance at this task.

To train this neural network, we will use the popular Python library `PyTorch`.[3] The input size is fixed at the length of an input vector (144) and the output size is fixed at one node, since we have intentionally set this up to be a binary classification task. Additionally, we will fix the nodes to use the ReLU activation function, with the exception of the output node which uses a sigmoid activation function to produce a probability in $[0, 1]$. Finally, we fix the loss function to be the binary cross entropy loss. Then, we search over the remaining hyperparameters, which include the number of epochs, the batch size, the learning rate, the optimizer, the number of hidden layers, and the number of nodes in each hidden layer. The suitability of hyperparameters is determined by the model's performance on the training set and validation set.

After an extensive search over more than eighty different models, we select the hyperparameters shown in Table 4. These hyperparameters were chosen because they yield high performance on both the training and development set while still maintaining some amount of simplicity in their architecture – containing only one hidden layer and a number of nodes on the order of the dimension of the input vector. Furthermore, the suitability of the chosen number of epochs is evidenced by the fact that the loss function tapers off, suggesting that further training would only cause the model to learn noise in the dataset. Note that for each combination of hyperparameters, we averaged the results across five models trained under these hyperparameters to account for the randomness in the initial configuration of the neural network.

To interpret our results, we will introduce a baseline which predicts the pairwise winner to be the horse with the more favorable public odds. Each comparison to this baseline that will follow in the next few paragraphs is shown in Figure 13, where higher values demonstrate better predictions in all subplots.

This model achieves 95.02% accuracy on the training set and 94.68% accuracy on the development set. Since the accuracy on these two datasets is roughly equal, this suggests that the model is *not* substantially overfitting; the slightly worse performance on unseen data is expected. The public odds baseline achieves an accuracy of 67.67% on the training set and 68.85% on the development set. Therefore, the accuracy of the neural network represents a 40.42% increase with respect to the baseline on the training set and a 37.52% increase with respect to the baseline on the development set. These results are shown in the leftmost subplot of Figure 13.

To further compare the trained neural network against the public odds baseline, we can consider downstream prediction tasks, such as predicting the winner of a horse race, the first two finishers in a horse race, or the first three finishers in a horse race, or even the full ordering of a horse race. It is obvious how to extract these predictions from the public odds: simply sort the horses in a race by their public odds and select the appropriate number of horses. However, it is non-obvious how to extract this prediction from our model, which only predicts pairwise winners. Our approach to using pairwise predictions towards downstream prediction tasks is discussed in detail in Appendix S, where the general idea is that we will predict on all pairs of horses and sum the results.

As shown in the second subplot of Figure 13, when using this approach, our model predicts the winner of a race with 82.72% accuracy on the training set and 83.43% accuracy on the development set. The public odds baseline achieves an accuracy of 36.36% on the training set and 37.25% on the development set. Therefore, the accuracy of the neural network represents a 127.50% increase with respect to the baseline on the training set and a

[3] `https://pytorch.org/`

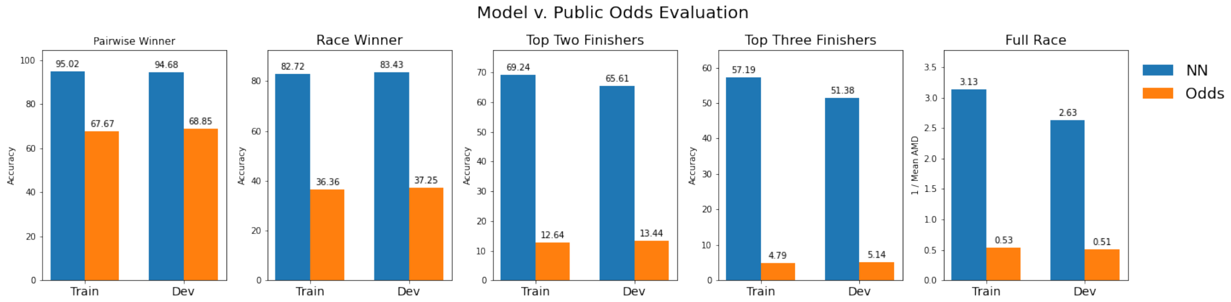

Figure 13: A comparison of the neural network model to the public odds by several different metrics.

123.97% increase with respect to the baseline on the development set. We can also compare this to values found across the literature since predicting the winner is a common task unlike predicting pairwise winners. Also using a neural network, Williams and Li (2008) is able to achieve an accuracy of 74% on a small dataset of Jamaican horses. Similarly, Davoodi and Khanteymoori (2010) is able to achieve an accuracy of 77% on a small dataset of New York horse races. Our results represent a fair improvement over these, though an exact comparison is not possible since the underlying datasets are different and may be inherently easier or harder to predict. In any case, our results represent a higher degree of confidence given a prediction over a much larger set involving thousands of races, as opposed to a few hundred. A cleaner comparison can be made to Torné (2021), since this analysis also includes the public odds baseline. In Torné (2021), the public odds baseline achieves an accuracy of 7.4% on the training set, whereas his trained neural network achieves an accuracy of 14.9% on the training set, representing a 101.35% increase. Therefore, although the dataset used in Torné (2021) appears inherently more difficult than our own, we are able to achieve much better accuracy proportional to the public odds baseline. Altogether, there is strong evidence that this model's performance is comparable to, or even outperforms, the other models found in the literature.

Still more, another way to evaluate our model may be to try and predict the top two finishers in a race (where order matters).[4] For this task, the trained neural network is able to achieve an accuracy of 69.24% on the training set and 65.61% on the development set. Instead, using the public odds attains an accuracy of 12.64% and 13.44% on the training and development set respectively. These results are shown in the third subplot of Figure 13. In a similar way, we can use our model to predict the top three finishers in a race (where order matters).[5] For this task, the trained neural network is able to achieve an accuracy of 57.19% on the training set and 51.38% on the development set. Instead, using the public odds attains an accuracy of 4.79% and 5.14% on the training and development set respectively. This is shown in the fourth subplot of Figure 13. In both cases, it is shown that our favorable results on the ability to predict a race winner generalize to further positions in the ordering.

Finally, we will evaluate our model on its prediction of the full ordering of a horse race. Since this is an exceptionally hard task, we will introduce a different evaluation metric as opposed to the binary evaluation metrics used before. We refer to the designed evaluation metric as the "average Manhattan distance" (AMD) of a predicted ordering on a race, and detail this metric in Appendix T. The general interpretation is that the AMD is the average difference between a horse's true and predicted finishing position, so that it is roughly in the range [0, # of runners]. A lower AMD represents a predicted ordering that is more similar to the true ordering and a high AMD represents a predicted ordering that is that is less similar to the true ordering. Conversely, the inverse of the AMD is a measure of goodness, and so is what is plotted in the rightmost plot of Figure 13 to provide visual symmetry. Using this evaluation metric, we derive that the models' predictions have an inverse mean AMD of 3.13 on the training set and 2.63 on the development set. Using our public odds baseline yields an inverse mean AMD of 0.53 on the training set and 0.51 on the development set. Once again, our model surpasses

[4] This task is known as an "exacta" bet in horse racing.

[5] This task is known as a "trifecta" bet in horse racing.

the baseline.

As one final comparison that illustrates the success of our neural network over the public odds baseline and is especially of interest to an audience which engages in gambling on horse racing, we can plot the profit of a simple betting scheme that uses the neural network versus the public odds. This is shown as Figure 14, where it is clear that a bettor using the neural network is able to achieve much greater profit than a bettor using the public odds.

As these tests have shown, the trained neural network model is highly performant with respect to the public odds baseline and even other models found across the literature, thus answering the question posed by this section. This high performance not only allows us to continue with our ablation experiment, but also establishes the novel method of using pairwise winners as a viable approach. Furthermore, in comparing this against the literature, there is some evidence to suggest that this method is better than alternatives, and future work may want to apply this to other datasets or other settings.

8.3 Ablation Experiment

Now, we conduct our ablation experiment to determine whether meteorological data is responsible for our high performance. In this experiment, we will train several models using the set of all features (including weather features) then train several more models using all features *except* weather features and compare their performance at the task of predicting the pairwise winner.

The models we have chosen to use in this experiment are a neural network, logistic regression model, random forest model, extra trees model, and decision tree model. The neural network will be created as before using `PyTorch` and the others will be created using `scikit-learn`.[6] Although other classification models exist, such as k-nearest neighbors and support vector machines, we leave them to future work on the basis that k-nearest neighbors suffers from the curse of dimensionality when dealing with long input vectors and support vector machines are slow due to their cubic asymptotic running time.

For each model, we will perform a grid search over a predefined set of hyperparameters, similar to work done by Iskandaryan et al. (2020). Although having to define valid hyperparameters in advance serves as a limitation of this approach, our available computational resources prevent a more thorough search over the hyperparameters. To allow each model to learn the dataset to the fullest extent, we will this grid search separately when using weather features and when not using weather features. After selecting the optimal hyperparameters through grid search, we will report the model performance on the training and development sets.

The results of this experiment are shown in Table 5. The table suggests that the models that leverage weather features perform roughly the same as the models which do not leverage weather features. We will discuss these results by model type.

8.3.1 Neural Networks

First, we will discuss the results of this experiment when using neural networks. Interestingly, we find that the chosen neural network which does *not* use weather features performs slightly better than the neural network which uses weather features on both the training and development set. Perhaps one explanation for these results is that the model with weather features is placing most of its weight on weather features while not leveraging non-weather features. That is, perhaps the model with weather features is obtaining the advertised accuracy with the weather features alone. This would be an interesting result because this would suggest that combining the model trained without weather data and the model trained with weather data, thus taking the union of their disjoint knowledge, would result in an even stronger model. Alternatively, this may suggest that more epochs are required for a model to simultaneously leverage both weather and non-weather features, though such computational resources are not available to test this.

A full exploration of this requires us to open up the neural network and observe its weights. This is a notoriously difficult task, earning neural networks their reputation for being a “black box”. Instead, we can indirectly observe the weights by feeding the neural network hand-crafted input and recording the output of the neural network. To determine the importance of a feature, we start with input representing logically equivalent horses then modify only this feature across both horses and observe the change in the output. We will use two different levels of the intensity for the modification of this feature, denote “small” modification and “large” modification. We expect the change in the output of the model to be proportional to the importance of

[6] `https://scikit-learn.org/`

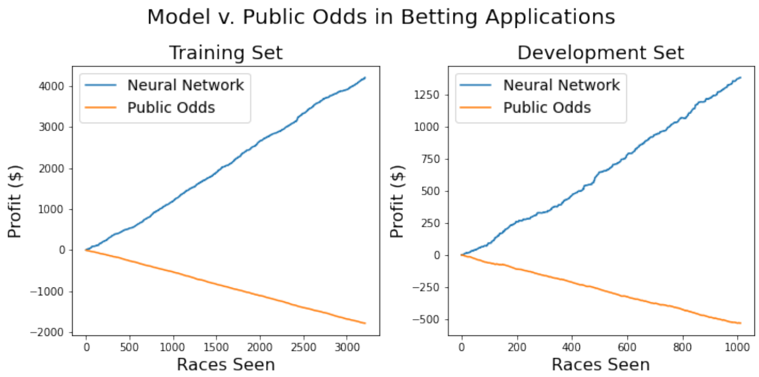

Figure 14: The profit of a bettor over time, with color used to differentiate between a bettor using the neural network or the bettor using the public odds to select a race winner. The bettor makes one dollar bets on the winner of a race in the training set or development set in chronological order. Nuance is added to these results in Appendix U.

			w/o Weather			**w/ Weather**		
Model	**Hyperparameters**	**Search Space**	**Opt**	**Train**	**Dev**	**Opt**	**Train**	**Dev**
NN	*	*	*	95.28	95.13	*	95.02	94.68
LR	solver	newton-cg, lbfgs, liblinear	lbfgs	94.55	94.31	lbfgs	94.55	94.31
	penalty	none, l1, l2, elasticnet	none			l2		
	C	0.01, 0.1, 10, 100	100			100		
RF	n_estimators	10, 50, 100, 200	200	96.95	85.78	200	99.34	84.76
	max_features	auto, sqrt	sqrt			auto		
	min_samples_leaf	1, 5, 10, 50	10			5		
ET	n_estimators	10, 50, 100, 200	200	99.16	83.59	200	99.59	82.89
	max_features	auto, sqrt	sqrt			sqrt		
	min_samples_leaf	1, 5, 10, 50	5			5		
DT	criterion	gini, entropy	gini	92.55	90.44	gini	92.56	90.48
	splitter	best, random	best			best		
	max_depth	5, 10, 20	10			10		
	min_samples_split	2, 5, 10	5			10		
	min_samples_leaf	1, 4, 8	8			4		

Table 5: This table captures the results of the ablation experiment. Note that the hyperparameters searched over for the neural network models are those outlined in Table 4, not shown here for brevity. The optimal hyperparameters for the neural network with weather features are shown in Table 4. The optimal hyperparameters for the neural network without weather features are the same as Table 4 except with 500 hidden nodes instead of 150 hidden nodes. Models with weather data perform roughly the same as models without weather data.

the feature. The technical details of this approach are left to Appendix V.

The results of this experiment are shown in Table 6. We will interpret the results of the columns labeled “Avg Rank”, which is a measure of the importance of the group of features, with lower ranks indicating more important groups. First, note that as the modification level increases, the different in predictions monotonically increase, which captures our intuition that the model is able to better discern between two horses as they appear more different. The horse attributes are the most important features in the input, which again matches our intuition since our public odds baseline is part of this group. More interestingly, we see that the neural network which uses weather generally places more importance on the past performances which are conditioned on similar weather conditions than those that which are not, as indicated by low average ranks on these groups of features compared to the other groups of feature. In particular, pressure and humidity seem particularly important to the model’s prediction. Of the weather features, rainfall seems to be the least important, yet is still more important than most non-weather features.

To determine if these results are due to noise, we can inspect how each model uses non-weather features. Both models place relatively low importance on past performances on the same course or same track condition, as indicated by high average ranks in each. Both models also place relatively high importance on past performances during the same month or with the same number of runners, as indicated by low average ranks. Finally, both models place moderate importance on past performances over the same racing distance or unconditioned past performances, as indicated by the average ranks lying between those of the aforementioned groups. Since these models exhibit similar trends among non-weather features, we can conclude that these results are probably *not* due to noise. If the models were learning noise, then it is unlikely that this relationship would hold across both models, given that these models receive a different set of features.

Therefore, the neural network with weather places more importance on the weather features, while secondarily learning the same trends over non-weather features as the neural network without weather. These results support the earlier claim that the model with weather features is using the weather features in a meaningful way and using them more heavily than the non-weather features. So, the neural network without weather data and the neural network with weather data have each learned how to leverage mostly disjoint features and so are equally useful towards the prediction problem. Future work may attempt to combine the knowledge of these models to create an even more performant model.

8.3.2 Logistic Regression

Moving on to the next model, we will now discuss the results of this experiment using the logistic regression model. The chosen logistic regression model with weather features and the chosen logistic regression model without weather features perform almost identically. Additionally, the difference in the performance on the training set and development set is slim, meaning that we can have high confidence that neither model has substantially overfit to the training set and has instead learned actual relations in the data. Again, we want to explore whether or not the logistic regression model is leveraging the available weather data.

Unlike the neural network, we can easily gain insight into how our logistic regression models are approaching the prediction problem by simply observing the weights they have learned. Since there are too many features to enumerate explicitly, we will again group these features and look at high-level trends among these groups.

The first row of Figure 15 plots the distribution of feature importances by group for each model, where we hold out groups that involve weather to allow for a more direct comparison of how each model uses non-weather features. In both models, horse attributes are the most important. This trend was also seen in the neural network models and again makes sense since the public odds are a baseline and the horse is most directly responsible for running the race. Next, the second-most important group to each model is the group of features that concern previous performance during the same month, which suggests that there is a time of the year when the jockey predictably performs better, possibly due to extended training during this time. In both models, features concerning distance and course have roughly the same performance. The models disagree slightly on the importance of past performances in races with the same number of runners and unconditioned past performances. The model that does not use weather places slightly more importance on past performances involving

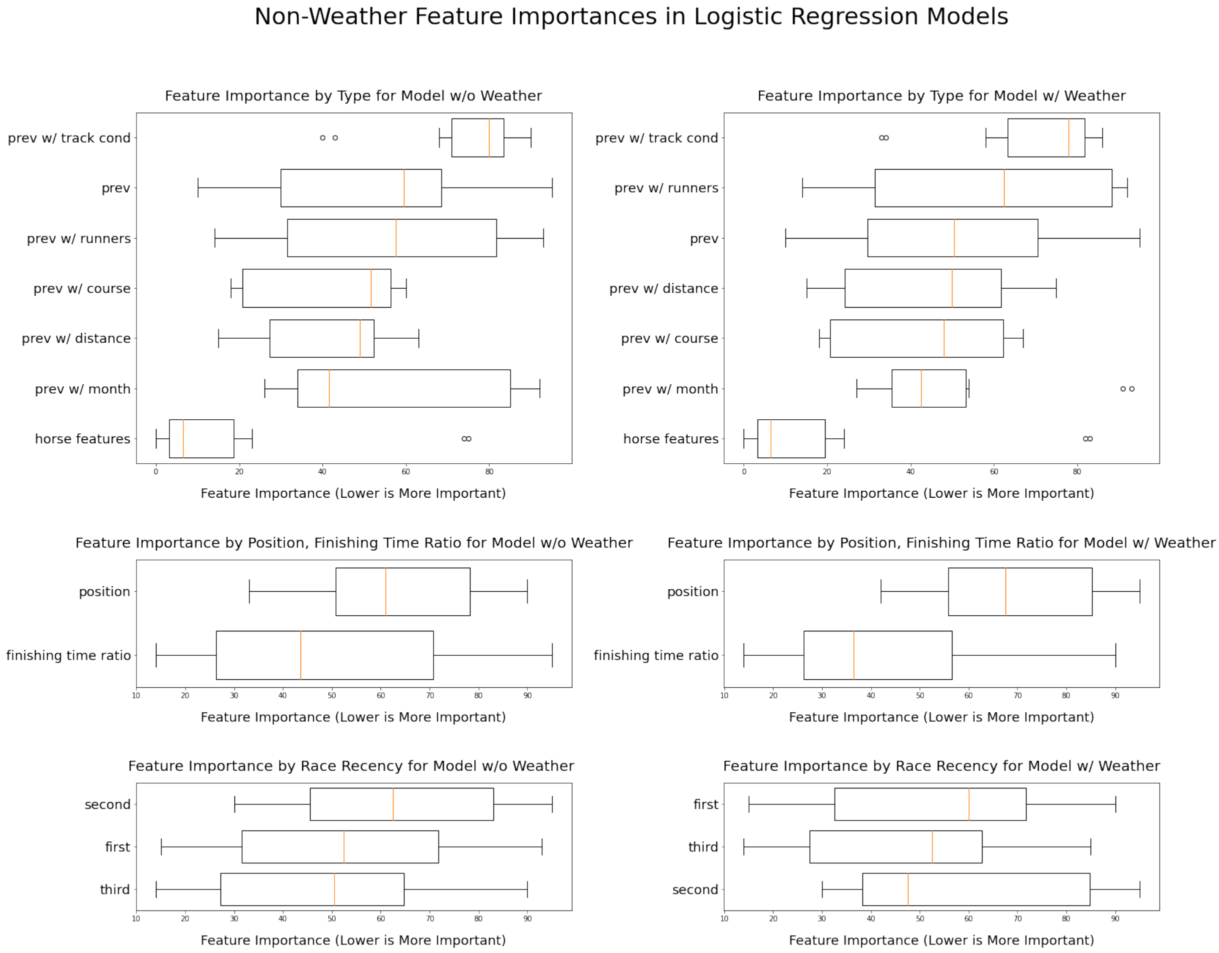

Figure 15: Each boxplot shows the distribution of feature importances for some group of features for each of the logistic regression models. Groups are sorted by their median feature importance from bottom to top. Feature importance is taken to be the rank of the feature's absolute weight in the sorted order of feature absolute weights. Lower feature importances correspond to more important features.

	NN w/o Weather			**NN w/ Weather**		
	Avg Diff in Output (x100)			**Avg Diff in Output (x100)**		
Group	**Small Mod**	**Large Mod**	**Avg Rank**	**Small Mod**	**Large Mod**	**Avg Rank**
Horse Attributes	40.8	40.8	1.0	38.2	38.3	1.0
Past Perf, Unconditioned	1.66	2.01	4.5	1.19	1.48	7.0
Past Perf w/ Course	1.47	2.10	5.5	0.90	1.13	10.0
Past Perf w/ Distance	1.42	2.52	5.0	1.12	1.30	8.5
Past Perf w/ Track Condition	1.61	1.95	5.5	0.83	0.84	11.0
Past Perf w/ Runners	1.94	2.22	3.0	1.52	1.65	5.0
Past Perf w/ Month	1.55	2.80	3.5	1.56	2.07	3.0
Past Perf w/ Temperature	-	-	-	1.22	1.66	5.5
Past Perf w/ Pressure	-	-	-	1.36	2.83	3.0
Past Perf w/ Rain	-	-	-	0.98	1.83	7.0
Past Perf w/ Humidity	-	-	-	1.18	2.10	5.0

Table 6: This table captures the average differences between the model's prediction on near-identical horses with modifications only to one group of features and the model's prediction on exactly identical horses, at two levels of modification. Larger differences represent higher feature importances. The average rank is the average position of the two values immediately to the left in sorted descending order in their respective columns. Therefore, the average rank is an estimate how important a group of features is to the model's prediction, with lower average ranks corresponding to higher feature importances.

the same number of runners than the model which uses weather and less importance on the unconditioned past performances than the model which uses weather. Finally, both models agree that past performance on the same track condition is not important towards this prediction problem. Since the same trends are seen across each model, we can be fairly certain that they represent the true nature of the prediction problem.

The second and third row of Figure 15 similarly show consistent trends across the two logistic regression models that is indicative of true learning and so we defer their discussion to Appendix W.

Having established that both logistic regression models have learned valuable information about the data, we now want to see how the logistic regression model trained with weather leverages the weather features. Figure 16 is identical to the top-right plot of Figure 15 except that it reintroduces the features that involve weather. This figure reveals past performances under the same pressure appear to be the most important to the model of all features involving past performances. The next important weather group is past performances under the same temperature, though several other groups lie between this and past performances under the same pressure. Past performance under the same humidity and past performance under the same rainfall appear to be relatively unimportant to the model.

Overall, it is shown that the logistic regression model likely meaningfully leverages pressure and

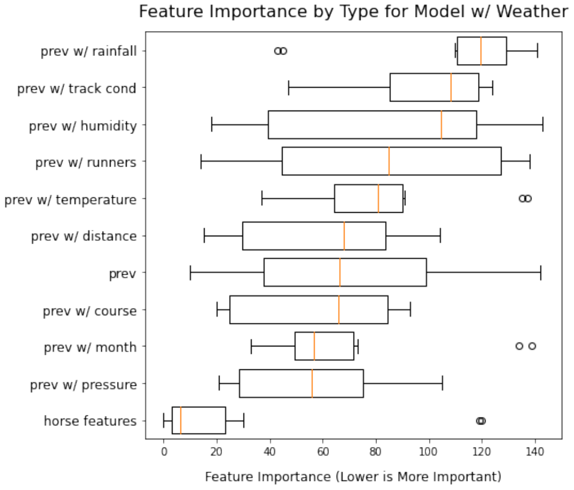

Figure 16: Each boxplot shows the distribution of feature importances for some group of features for the logistic regression model trained with weather. Groups are sorted by their median feature importance from bottom to top. Feature importance is taken to be the rank of the feature's absolute weight in the sorted order of feature absolute weights. Lower feature importances correspond to more important features.

temperature towards the prediction problem. Therefore, under more ideal training circumstances (e.g. a larger hyperparamter search or proactive weather collection), it is expected that the inclusion of such would bolster the model with respect to the baseline which lacks weather features.

8.3.3 Random Forest, Extra Trees, Decision Tree

The ablation experiment captured in Table 5 reveals that the random forest, extra trees, and decision tree models have all overfit to the training data. Therefore, despite nonetheless outperforming the public odds baseline, they offer relatively weak explanatory power compared to the two models we have already analyzed. For this reason, we defer a discussion of these models to Appendix X.

8.4 Summary of Analysis

In summary, through this analysis we have first demonstrated the success of our approach at creating a model for the pairwise winner prediction task that thoroughly outperforms the public odds baseline. Furthermore, in aggregating the predictions on all pairs of horses in a race, we are able to demonstrate success in downstream prediction tasks, such as predicting the winner of a race.

Given this performance, we conducted an ablation experiment by removing those features involving weather to determine if this performance was due to the inclusion of weather data. Although the ablation experiment revealed that models with weather data and models without weather performed similarly, a deeper inspection of the resulting neural network and logistic regression models reveals some evidence that weather features, in particular pressure and temperature, are meaningfully leveraged in these models to learn actual relationships in the data. This corroborates some of the non-academic sources we have come across. In fact, The Whitley Group (2021) claims that barometric pressure is the chief meteorological factor that may affect a horse's performance in a race, rather than rainfall as is often suspected. Additionally, CPR (2021) details how excessive heat may affect horses differently, such that some horses suffer significantly more discomfort than others. On the other hand, neither of these models benefit from the use of rainfall data, which opposes claims about the importance of rain to horse racing prediction made by Punter 2 Pro (2021) and Skymet Weather (2021). One explanation for this may be that rainfall, since it is more easily observable than pressure for instance, is already considered by trainers and owners such that a sort of "adverse selection" has already taken place, making the pool of horses more uniform in their ability to run under the current rainfall conditions. This "adverse selection" may also explain why the track condition is lightly used by models, since track condition is related to rainfall.

Therefore, the slim differences in the performances between models with these features and models without these features may be attributed to the fact that the imperfect process of training these models have caused them to learn a disjoint set of features that achieve similar predictive power rather than leveraging the full feature space. This seems especially plausible given the high performance of the model, after which it seems difficult to earn additional performance regardless of the features used due to an argument of the law of diminishing returns.

Nonetheless, since the inclusion of weather features cannot fully explain the success of our model, which is speculated to be slightly better than those models found in the prior literature, we instead turn to our approach as an explanation for the success of our model. Recall, our approach is novel in the amount of data used and reduction of the task to predicting the pairwise winner.

Finally, we evaluate our neural network with weather against the held-out testing set to achieve an accuracy of 94.56% at predicting the pairwise winner and an accuracy of 78.89% at predicting the race winner, both of which are slightly below the accuracy on the development set but nonetheless thoroughly outperform the baseline and compare favorably to existing models across the literature.

9 Conclusion

We are able to make a successful neural network that achieves 78.89% accuracy at predicting the winner of a horse race using a dataset of races from Ireland. This neural network, among other models used in our analysis, is novel in its thorough consideration of meteorological conditions at the time and location of a race, whereas such consideration is largely absent from the existing literature except when included in the form of a categorical variable taking on one of only a handful of levels. This is enabled by our construction of a highly-curated set of over 20,00 horse races occurring in Ireland annotated with the temperature, pressure, humidity, and rainfall near the time of the race, based on data from Kashavkin (2020) and Met Eireann. This dataset is made publicly available as discussed in Appendix A and is one of our main contributions.

Although ablating the model to prevent it from using these features related to weather did not significantly change its nominal performance, there is nonetheless evidence to support the conclusion that at least pressure and temperature are important towards determining the outcome of a horse race. Both of these meteorological factors are posited to be important by non-academic sources but not acknowledged in the academic literature, thus making this confirmation one of our contributions to the literature.

Additionally, we present a unique approach to predicting the outcome of a horse race in reducing the problem to predicting the winner among a pair of horses then aggregating such results using simple heuristics. This approach is chosen for its seamless data augmentation, flexibility in handling missing data, and architectural simplicity. The success of this approach is witnessed by the high accuracy reported above, in combination with the observation that the inclusion of meteorological data alone cannot explain this performance.

In summary, in response to our research question, our results suggest that weather factors such as pressure and temperature may improve machine learning models which predict the outcome of a horse race. This may be of interest to jockeys, trainers, owners, racecourse managers, and sports bettors who seek to employ models to more selectively enter horses in a race, schedule more entertaining races, or find profit at the racetrack. This is also of casual interest to data scientists and athletes who wish to apply these findings to different disciplines.

10 Future Work

The results of this project inspire several directions for future work. The most promising of which is a direct comparison between pairwise winner prediction and other approaches, such as predicting the winner given an entire race or predicting whether a given horse will be a winner. Future work on the same dataset will allow us to directly compare performance of these two approaches. Should the results of these experiments show that pairwise winner prediction is a strong approach, the next step might be to apply pairwise winner prediction to other disciplines entirely where there is a simultaneous contest between more than two entities. For example, predicting pairwise winners to resolve a race may be useful to researchers that work with auto racing, track and field, or even political elections. To understand why each of these fields is important, consider in the case of sports events such as the first two, the ability to predict the winner may help schedule close races, which are more entertaining for viewers to watch. In the case of political elections, campaign managers are highly motivated to predict the outcome of the race so that they may reroute resources to increase their own chances.

Additionally, we discuss a few direct extensions of our own research, should we desire to continue exploring this research question. Firstly, the same as was done with jockey past performances may be done with trainer past performances, under the same assumption that horses with the same trainer undergo the same training regiment and so can be expected to perform similarly. This may be helpful because the pool of trainers is even slimmer than the pool of jockeys, such that each trainer has an even richer record of past performances than a jockey. We chose not explore this here since we suspect that the correlation between a trainer and the performance of a horse in a race may be *looser* than that between a jockey and the performance of a horse because the trainer is not directly involved at the time of the race. Additionally, a direct extension of our own research may divide races into classes based on their weather and determine if models with weather features do better on certain classes than others compared to models without weather features. A more targeted analysis, rather than predicting over all weather conditions, may reveal some particularly susceptible class of races. Lastly, a direct extension of our own research is to use our existing infrastructure towards different datasets. Financial limitations prevented us from exploring American horse races, which may reveal different trends than that of Irish horse races due to different practices in meteorological record-keeping or different climates.

Acknowledgments

I extend my gratitude to Professor Jonathan Hanke and Sulin Liu of Princeton University who offered invaluable feedback at various stages of the project.

References

Sinclair Bell and Carolyn Willekes. 2014. Horse racing and chariot racing. *The Oxford Handbook of Animals in Classical Thought and Life*, pages 478–90.

William Benter. 1994. Computer based horse race handicapping and wagering systems: a report. In *Efficiency of racetrack betting markets*, pages 183–198. World Scientific.

Ruth N Bolton and Randall G Chapman. 1986. Searching for positive returns at the track: A multinomial logit model for handicapping horse races. *Management Science*, 32(8):1040–1060.

CPR. 2021. Racing in the heat.

Elnaz Davoodi and Ali Reza Khanteymoori. 2010. Horse racing prediction using artificial neural networks. *Recent Advances in Neural Networks, Fuzzy Systems & Evolutionary Computing*, 2010:155–160.

HKJC. 2021. Hkjc betting services.

Tomislav Horvat and Josip Job. 2020. The use of machine learning in sport outcome prediction: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(5):e1380.

Ditsuhi Iskandaryan, Francisco Ramos, Denny Asarias Palinggi, and Sergio Trilles. 2020. The effect of weather in soccer results: An approach using machine learning techniques. *Applied Sciences*, 10(19):6750.

Nikolay Kashavkin. 2020. Horse racing data from 1990.

Met Eireann. Historical data.

Met Office. 2019. Met office midas open: Uk land surface stations data (1853-current).

Punter 2 Pro. 2021. The impact of weather on horse racing the going.

Sankari et. al. 2021. Can machine learning make horse races a winning proposition?

Marie Sheridan and John Sweeney. 2001. Weather and horse racing: Towards a more objective prediction of the going. *Weather*, 56(2):48–55.

Skymet Weather. 2021. How does the weather conditions impact horse racing?

The Whitley Group. 2021. Tips for horse racing: How weather can affect a race.

Olaf Torné. 2021. Ga yau: Machine analysis of hong kong horse racing data.

Roger C Vergin. 1977. An investigation of decision rules for thoroughbred race horse wagering. *Interfaces*, 8(1):34–45.

Janett Williams and Yan Li. 2008. A case study using neural networks algorithms: horse racing predictions in jamaica. In *Proceedings of the International Conference on Artificial Intelligence (ICAI 2008)*, pages 16–22. CSREA Press.

A Code Availability

All code and data is publicly available at https://github.com/AnthonyHein/SML310Project. Additionally, an accompanying web application for this project is found at http://horseracing.anthonyhein.com/.

B Lack of Horse Racing Academic Literature

We advance the claim that the lack of academic literature revolving around horse racing is due to the following two reasons:

- **Proprietary nature of the datasets**: The entities best equipped to create and maintain such datasets are the racetracks themselves, who serve as bookmakers to sports bettors visiting the racetrack. As the profit extracted by the racetrack is proportional to the amount wagered, and thereby the disagreement in the belief's of the sports bettors, the racetrack is incentivized *not* to share this data. In other words, as the knowledge a sports bettor holds about a given horse race approaches perfect information, the bookmaker may extract less revenue from the sports bettor in expectation.

- **Controversy surrounding horse racing**: With jockeys, trainers, and owners all having a stake in the outcome of a race, an exorbitant amount of pressure is put on horses to perform well. This pressure is sometimes exerted in the form of premature training, rigorous schedules that omit leisure time or social interactions, constant caged travel to different racetracks, and the use of performance enhancing drugs. Such controversy may discourage academics from exploring the data, since doing so may implicitly condone such malicious behavior by jockeys, trainers, and owners. Although we do not condone such behavior, this paper will apply data science and machine learning techniques towards horse racing data to answer an interesting research questions. Furthermore, to recognize the unfortunate circumstances of many racing horses, we have made a donation to the Thoroughbred Retirement Foundation.[7]

C Horse Racing

In this section, we will present the reader with a casual knowledge of horse racing. This primarily defines the common terminology of the sport, the influence of which can be found within the datasets, related work, and the analysis of this paper itself, and so is unavoidable.

A typical race is between fourteen or fewer horses (sometimes called *runners*). Each horse has a jockey who is meant to maneuver the horse around and between other horses as well as pace the horse to prevent early fatigue or signal the horse to speed up. Both the horse and jockey will be wearing the same color, and this color will be different from other horses and jockeys to make it easy to identify the horse during the race. Additionally, the garments worn by the horse and jockey will bear the *saddle* number for this horse, which is used as an abbreviated way to identify a horse during a race and also is the position of the horse when they lineup at the starting gate, with lower numbers meaning that the horse is closer to the inside of the track. This means that, all else being equal, horses with lower saddle numbers run a shorter distance. During a race, spectators and announcers typically use the unit of a "length", which is the length of a horse from their nose to their tail, to describe how much a horse is trailing its predecessor by.

Races are held on a racecourse/racetrack, each of which is slightly different. The condition of the track – i.e. how firm or slippery the track is – is often referred to as the *going*. Additionally, the distance of the race varies across racecourses.

Many fans of the sport enjoy betting on the outcome of a race. The most popular bet that is usually offered is a *win* bet, where the bettor selects a single horse that they believe will place first in the race and wins if and only if this horse indeed places first.

Towards this purpose, the racecourse will maintain the public odds for each horse in a race as they collect wagers. The easiest odds to interpret, and the odds that are used throughout this analysis, are *decimal odds* / *decimal prices*. The amount you would earn on a *win* wager for a given horse is just the amount of the wager times the *decimal price*. For example, a decimal price of 2 means that you would double your money. On the other hand, a decimal price that is very nearly 1 means that this horse is highly projected to win such that you would simply earn back your wager plus some small amount. The inverse of the decimal odds induce an un-normalized probability distribution for which horse will win the race. The managers of the racecourse may place a *handicap* on a horse by adding extra metal weights on a horse if the horse would otherwise be projected to win with high probability to encourage more people to bet and more accurately simulate a random event. Finally, other metrics meant to capture a horses ability include racing post ratings (RPR), topspeed (TR), and official rating (OR).

D Estimating Finishing Times

For each race, we have the finishing time of the first place finisher and the distance from each horse to its predecessor, with the units of distance being the length of a horse. Recall, from the provided overview of horse racing in Appendix C, that it is common for those involved in the sport to use "lengths" instead of actual finishing times. These lengths are derived from the finishing times themselves, which means that we would be able to recover these finishing times by reversing the calculations, given the conversion factor of seconds per length. While, this conversion factor is subjective and varies depending on the individual entering the data, we can estimate it by using known race attributes using the following equation:

$$\gamma \text{ (seconds / length)} = [\text{winning time (s) / dist of the race (m)}] \times [\text{avg length of horse (m) / one length}]$$

Then, the finishing time of any horse is calculated as the following:

$$\text{finishing time} = \text{winning time} + (\text{lengths behind winner}) \times \gamma$$

E Kashavkin's Horse Racing Dataset

For each year $y \in \{1990, 1991, ..., 2019, 2020\}$, the dataset by (Kashavkin, 2020) contains a file `horses_y.csv` and `races_y.csv`. Each file of the type `horses_y.csv` has rows which are horses and contain the following columns (most important are bolded):

- **`rid` - race ID**
- **`horseName` - horse's name**

[7] https://www.trfinc.org/

- **age - horse's age**
- **saddle - saddle # where horse starts**
- **decimalPrice - inverse of the decimal odds**
- isFav - indicator variable if horse was favored to win
- trainerName - horse's trainer's name
- **jockeyName - horse's jockey's name**
- **position - finishing position of horse, 40 if they did not finish**
- positionL - how far a horse has finished from the preceding horse, measured in lengths
- **dist - how far a horse has finished from a winner, measured in lengths**
- weightSt - horse's weight in stones
- weightLb - horse's weight in pounds
- overWeight - overweight code
- outHandicap - horse's handicap weight
- headGear - horse's head gear code
- **RPR - racing post rating**
- TR - topspeed
- OR - official rating
- father - horse's father's name
- mother - horse's mother's name
- gfather - horse's grandfather's name
- runners - total runners in race
- margin - sum of decimalPrices for the race
- **weight - horse's weight in kilograms**
- reswin - indicator variable if horse won or not
- resplace - indicator variable if horse placed or not

Each file of the type races_y.csv has rows which are races and contain the following columns (most important are bolded):

- **rid - race ID**
- **course - course of the race**
- **time - time of the race in hh:mm format, London TZ**
- **date - date of the race**
- title - title of the race
- rclass - race class
- band - band
- ages - ages allowed
- distance - distance
- condition - surface condition
- hurdles - hurdles, their type and amount
- prizes - places prizes
- **winningTime - best time shown**
- prize - prizes total (sum of prizes column)
- **metric - distance in meters**
- **countryCode - country code of the racecourse**
- **ncond - condition type (created from condition feature)**
- class - class type (created from rclass feature)

F Great Britain Meteorological Data

When searching for meteorological data for Great Britain, we found the Met Office Integrated Data Archive System (MIDAS) for land surface station data which offers hourly weather at several stations (Met Office, 2019). Accordingly, we using Google's Geocoding API to annotate over 60 racecoures with latitudinal and longitudinal coordinates, then proceeded to check these annotations manually. Next, we downloaded a spreadsheet detailing the opening year, closing year, and latitudinal and longitudinal coordinates of each weather station in Great Britain from MIDAS. Then, we ran an algorithm to determine, for each race occurring on some racecourse at some date and time, the closest open weather station. Due to the format of MIDAS, where each station maintains weather data in separate files for each year and separate files for each type of reading, the set of files identified by our algorithm exceeded one-thousand files in size which would need to be manually downloaded. Additionally, due to the sparse nature of this data, where a file commonly had a large proportion of missing values, additional files would need to be re-downloaded after inspection. Therefore, after a prolonged foray into Great Britain's meteorological record-keeping, we ultimately halted work on this front.

G Met Eireann Meteorological Data

We maintain Irish meteorological data in a file called `weather_all.csv`, which contains rows which are weather readings with the following columns:

- `date` - the date and time of this reading
- `temp` - air temperature (°C)
- `msl` - mean sea level pressure (hPa)
- `rain` - precipitation amount (mm)
- `rhum` - relative humidity (%)
- `station number` -

Additionally, metadata is provided in a file called `ireland_stations_metadata`, which contains rows which are weather stations with the following columns (most important are bolded):

- `County` - county of weather station
- **`Station Number` - station number of weather station**
- `name` - name of weather station
- `Height (m)` - altitude of weather station
- `Easting` - x-coordinate in geographic Cartesian coordinates
- `Northing` - y-coordinate in geographic Cartesian coordinates
- **`Latitude` - latitude of weather station**
- **`Longitude` - longitude of weather station**
- **`Open Year` - opening year of weather station**
- **`Close Year` - closing year of weather station**

H Scrubbing Horse Racing Data

For the file containing metadata on races, we first trimmed this dataset by removing all races that occurred in a country other than Ireland, having decided this to be the focus of our analysis. We additionally dropped rows according to the following conditions:

- **Remove a race if it has hurdles**: Hurdles were removed because they occured in too many categorical levels and could not easily be translated to a numerical feature space such that it was not expected to be helpful towards predicting the outcome of a race. The vast majority of races did not have hurdles, which makes the effect of dropping these negligible on the size of our dataset. Of course, this means that our results only apply to those races without hurdles.
- **Remove a race if it has a non-positive winning time**: Recall, the winning time is the time of the first place finisher in the race. Obviously, a non-positive winning time does not make sense given this definition, and there is no clear way to infer this value. Additionally, the winning time is important to downstream analysis since it is used to calculate other finishing times, so we may not leave an invalid value here.
- **Remove a race if it has fewer than three or more than fourteen runners**: This is the range of runners in a typical horse race. Races outside of this range usually represent special events that are not representative of typical races. More practically, given that our analysis will predict on pairs of horses within a race, an excessive amount of runners makes this computation intractable since the number of pairs is quadratic in the number of runners.

Having trimmed this dataset, we took a few more steps to clean it, which included:

- **Correcting the date**: Sometimes, the column for the date on which the race occurred contained an extraneous suffix of "00:00", which made it difficult to parse this column into a datetime object in Python, which was necessary towards our goal of annotating races with weather readings. We removed all such suffixes.

- **Correct the racetrack**: Some racetracks appeared with several country codes between parentheses appended to the end of them (e.g. "Dundalk (AUS) (GB) (US)"), which confused Google's Geocoding API. We removed all such country codes.

For the file containing information on each horse that runs in a race, we dropped rows according to the following conditions:

- **Remove a horse if it has a position which is non-positive**: The position of a horse is used either directly or indirectly as the target for some of the classification models in our analysis, so we cannot leave rows with an invalid position. Due to the possibility of ties or non-finishers, this is impossible to infer.
- **Remove a horse if it has an age which is non-positive**: We suspect that the age of the horse may be an important factor in determining its performance in a race, as we may expect younger more sprightly horses to perform better. Since age is an absolute measure, rather than a rating which is measured relative to other horses, it is not clear how to impute this value by means of k-nearest neighbors or similar methods.
- **Remove a horse if it has a saddle which is non-positive**: Horses with a lower saddle number have to run a slightly shorter distance, though traditionally suffer the disadvantage of being unable to pace themselves very well at the start of the race. Due to this tradeoff, we want to incorporate this as a feature in our model.
- **Remove a horse if it has an RPR which is non-positive**: RPR is the racing post rating and is a calculated by racecourse officials, instead of a result of public bets, that is intended to measure how "good" a horse is suspected to be. Therefore, it seems helpful to have this measure in addition to the public odds. More practically, very few entries are lacking an RPR value and so it is not costly to drop those which are missing this.
- **Remove a horse if its distance to its predecessor is null**: This information is required since we will use a calculation involving these distances along with the winning time of the first place finisher to estimate the finishing time of a horse. In turn, the finishing time of a horse is suspected to be an important feature in the model.
- **Remove a horse if it lacks the name of its jockey**: Knowing the jockey is important since jockeys differ in skill level. A novice horse mounted by a veteran jockey may be able to outperform a veteran horse mounted by a novice jockey. More practically, we can suspect that several engineered features involving the jockey will be helpful in downstream analysis.
- **Remove a horse if it lacks the name of its trainer**: This case is the same as the above, except that the trainer does not influence the second-by-second performance during a race as much as the workout regiment that the horse undergoes in preparation of a race. Although we do not end up using the trainer in our analysis, we nonetheless have this data in case it is of future interest.

Note that these are not the only places where we find invalid data among entries for horses. Null values can also be found in columns meant to record the father, mother, and grandfather of a horse. Although it seems plausible that there could be a correlation between the performance of a horse and the performance of its father, mother, or grandfather such that we can use the performance of the horse's father, mother, or grandfather as a feature, only 25% of horses in our dataset had a father also in the same dataset, 17% of horses in our dataset had a mother also in the same dataset, and 8% of horses in our dataset had a grandfather also in the same dataset. This means that any such feature would be exceedingly sparse and likely unhelpful, so we will ignore these columns altogether. Nonetheless, we will note that it may be an interesting question for future research to determine how well the past performance of a horse's father, mother, or grandfather can predict the debut performance of a horse.

Furthermore, null values can be found in columns meant to record TR and OR: various ratings for a horse. We will likewise ignore these columns because the markedly less sparse RPR and decimal odds are meant to perform the similar function of providing easily digestible summary statistics for a horse and so can be expected to come from a similar distribution as TR and OR.

Indeed, we can show the correlation between RPR and TR and the correlation between RPR and OR to be relatively high, at 0.63 and 0.77 respectively. Therefore, rather than drop entries based on this and thereby make our dataset much smaller, we will assume most of the information otherwise provided by TR and OR to be captured by RPR and decimal odds.

In an effort to conserve data, we took steps to correct invalid entries where possible. For example, in some instances where a horse was listed with a non-positive or null age, we were able to correct this by finding entries for the same horse in a different race where this horse *was* listed with a valid age, finding the amount of time between these races, and adding/subtracting the appropriate amount of time to obtain the correct age. As another example, we could take a similar approach of finding different races containing the same horse to infer the father, mother, or grandfather of a horse where this data was otherwise missing.

Additionally, during data scrubbing, we repurposed a column called `dist`. This column was particularly confusing when using the dataset, as it was meant to capture the distance a given horse finished behind the *second* place finisher of that race. Then, horses that finished in first or second would have null values in this column. It was unclear why the original author of the dataset included this as a column, since finishing in first place and second place are not the same from any involved party's perspective, such that it seems important to know how far a horse is from the *first* place finisher of a race, more than the *second* place finisher. For example, it could be the case that the first place horse won by a longshot then all horses to follow came in at roughly the same time; this situation is not captured very well under this definition. Therefore, we recalculated this column to be the distance of a given horse to the first place finisher by iterating on the distance of a horse to its predecessor within a race. One difficulty of this was translating horse racing jargon, as the community has come to use body parts to represent fractions of lengths; for example, a "neck" is about a quarter of a length. Note, the distance of the first place finisher to itself is trivially 0, rather than null.

As a small point, we fixed some instances of non-consecutive finishing positions within the dataset. An example of such an instance is a race with eight runners having horses that finished in positions 1 through 4 and then in positions 6 through 9, without any horse finishing in position 5. We chalk up such instances to bad data entry.

Finally, we made some design decisions in how to handle entries for horses that did not finish, which were marked in the dataset by having a position of 40. If we were to view each record in the horse dataset as a data structure, the cleanest solution would be to make columns such as the aforementioned `dist`, the calculated finishing time, and the positions *optionals* that may take on some value or may be null, then append a Boolean encoding whether the horse finished or not. However, the inability of machine learning models to gracefully handle null values makes this a non-solution. Furthermore, dropping all races where there are non-finishers may inject an intolerable bias, since it creates the implicit assumption that all horses will finish a race; this is a poor assumption and indeed part of predicting the outcome of a race is predicting whether a horse will finish or not. Therefore, the chosen solution was to keep a position of 40 and let the distance from all non-finishers to the first place finisher be the distance of the last finisher plus 30 lengths (where 30 lengths is chosen because it is a typical constant used in the horse racing community to mean that a horse is trailing by a lot). Note that the finishing time is calculated from the distance of a horse to the first place finisher, so encoded in this is the fact that the finishing time of a non-finisher will be relatively bad. Basically, this modification pretends that all non-finishers did indeed finish the race and just performed extremely poorly, as if they crawled to the finish line. In any case, this avoids the complicated relationships that optionals would enforce over the columns.

After individually cleaning the file of race information and the file of horse information, we took the rows belonging to the intersection of the races represented by each file. That is, for each race in the race file we have the full information about the horses present and for each horse in the horse file we have the full information about the race it ran in.

I Ideal Weather Conditions

Related to featurization of the dataset which is discussed in detail in Section 7, the numerical values for temperature, humidity, and pressure can each be placed into one of five equal bins while rainfall is binarized. Then, a race that is run under ideal weather conditions will have all of:

- A recorded temperature that belongs to one of the middle three bins for this variable. Such a choice is motivated by the posited difficulties of running in either extreme cold or extreme heat where one's body must expend more energy to maintain a comfortable temperature.
- A recorded humidity that belongs to one of the middle three bins for this variable. Such a choice is motivated by the posited difficulties of running in extremely low humidity where one's body dries up or extremely high humidity where the inability for sweat to evaporate traps excess heat.
- A recorded pressure that belongs to one of the top two bins for this variable. Unlike the previous variables, higher pressure makes it strictly easier to run, as it is correlated with the abundance of oxygen. On the other hand, low pressure represents a lack of oxygen. Tangentially, the difficulty of running in low pressure explains the popularity of high altitude training regiments (where pressure decreases with altitude).
- No rainfall recorded during this race. Rainfall decreases the firmness of the track and decreases visibility, both of which make it more difficult to run.

Note that when we split the dataset by this definition, we are left with 10520 races run under ideal weather conditions and 9681 races run under non-ideal weather conditions with nearly the same distribution of runners.

J Normalized Winning Time

The "normalized winning time" is meant to be a metric to measure how fast a winning time is with respect to the global population of winning times for races *on the same distance*. That is, since races occur on different distances, we cannot use the winning time directly. To achieve a "normalized winning time", we will group the dataset of races by their distances then compute the mean and standard deviation of the winning times for each distance. Finally, we will normalize the winning time of each individual race by subtracting off the mean of the winning times for races on the same distance and dividing by the standard deviation of the winning times for races on the same distance. Now, all "normalized winning times" are drawn from the standard normal distribution and so are more readily comparable, despite their underlying races being run on different distances.

K Featurization Non-Solutions

K.1 Categorical Variables

One possible approach to predicting the outcome of a race is to leave the horse name, jockey name, and trainer name as categorical variables and simply append the race metadata. As best practice would encourage, such categorical variables would then have to be unraveled into several one-hot encoded columns before being fed into a machine learning model. Then, we would hope that a machine learning model can learn which values of these categorical variables are "better" than others through observing when these one-hot encoded columns are active and what the result of the race was. We would also hope that a machine learning model can learn which horses are better than others under certain weather conditions by observing the weather in the race metadata, observing the runners in the race, and then observing the outcome of the race. For example, perhaps the horse with the name "Chief Little Hawk" performs better than the horse with the name "King's Return" when it is raining during the race but not when it is dry during the race. Then, such a model can be expected to be weaker if weather data was removed and so this would allow us to answer our question of whether meteorological data can improve predictions on the outcome of a race.

However, upon observing that there are 50,291 unique horses, 2,660 unique jockeys, and 1,774 unique trainers, it is clear that this is not feasible since the resulting vectors would be too large for the available computational power and number of datapoints. Therefore, the horse name, jockey name, and trainer name cannot be used as categorical variables.

K.2 Horse Past Performance

Although using the horse name, jockey name, and trainer name as categorical variables seems infeasible, we would still like to preserve the identity of a runner in some way, as is suggested to be important by other analyses in the existing literature and required to answer our research question. Otherwise, it is not clear how a model would be able to learn that horse A is better than horse B under certain weather conditions that are relevant to the current race when the identities of horse A and horse B are stripped off.

Therefore, a second approach to this prediction problem attempts to reinstate some form of identity by encoding this as a horse's past performance rather than a horse's name. In other words, we may attempt to replace an instance of the horse name "Chief Little Hawk" in a race that occurs on June 10^{th}, 2020 with the finishing position of "Chief Little Hawk" in the most previous races this horse ran prior to this date. This is very similar to the features used across models in the literature, especially the features used by Torné (2021). Furthermore, consider how we can modify this feature to be the finishing position of this horse in the most previous race it ran prior to this data *that was under the same weather conditions as the current race*. Note, Section 7.2 introduces the idea of similar weather conditions across two races. Also note that we may not consider races that occurred *after* the given race in question because this contains information not available at the time at which prediction is most important, namely, before the given race in question occurs. This featurization scheme recovers the ability to consider the weather and how it affects individual horses when predicting on the outcome of a horse race, at intended. Unfortunately, a deeper look at the distribution of horses in the dataset reveals that very few horses run enough races to accumulate these features. This is shown in Figure 10 where the distribution is heavily right-skewed, with 50% of the horses in the dataset running two or fewer races and 34% of the horses in the dataset running only a single race. It is easy to see that any horse which is running in its debut race will not have these features available and so this entry will have to be dropped, since machine learning models do not accept null values. However, this problem is not unique to horses running in their debut race; since we will want to capture the past performance of a horse in similar track and weather conditions, this will often necessitate a horse having run several – at least more than one – previous races. Therefore, for each horse this would require dropping several of this horse's first few races, if they even exist. Using this featurization scheme with a modest set of features involving past performances of a horse results in 98.9% of the featurized dataset containing null values, and thus being dropped. So, we are not able to use the past performances of a horse as our features.

L Finishing Time Ratio

Here, we provide two definitions related to the past performance features used in our model:

Definition 1 (Finishing Time Ratio)**.** Define the finishing time ratio of a jockey in a race as the ratio of the jockey's finishing time in that race to the winning time in that race. That is, if the jockey wins the race, then the finishing time is 1. If the jockey loses the race, the winning time is some number strictly greater than 1. Intuitively, this represents how close to winning a jockey was in a given race, with values nearest to 1 meaning that the jockey was closer to winning.

Definition 2 (Global Finishing Time Ratio)**.** Define the global finishing time ratio of a jockey in a race as the ratio of the jockey's finishing time in that race to the fastest recorded winning time across all prior races run on the same distance. It is important to only consider races on the same distance, since races run on farther distances will necessarily have longer finishing times. Additionally, it is important to only consider prior races, else this uses information not available at the time of prediction. Even if the jockey wins the race, the global finishing time ratio may still be greater than 1 unless, in winning the race, they have set a new record for that distance within the dataset. Intuitively, this represents how fast the jockey was in a race compared to *all* jockeys running on the same distance, not just those jockeys in the given race.

M Featurizing Datetimes

To make datetimes more digestable as a feature, we do two things to encode these. First, we encode the month, which implicitly uses the assumption that weather is similar/constant throughout a month, which seems fair given how it is common to talk about the weather at the granularity of a month. Then, we extract the year and leave it as its own variable. This helps a potential model capture any trends over time. In summary, we propose that anything a model may learn from a datetime is in fact some baseline knowledge about the month of the datetime plus some perturbation introduced by the year. To make this more concrete, this is similar to claiming that December is very cold, but gets warmer as the years go on because of global warming.

N Similar Weather Conditions

In our analysis, a race is run under similar weather conditions as another race if their weather values belong to the same bins. Consider an alternative to this idea of races run under similar weather, which would be to find races with the *most* similar conditions to the given race, perhaps measured by Euclidean distance of vectors containing the weather values. We argue that this would not yield good results since, for a given jockey, the race run under the most similar weather conditions to the given race may be arbitrarily different if that jockey simply has not run a race which is intuitively similar. As an example which illustrates the failure mode of this alternative, consider a race that is run in highly non-ideal weather conditions. Now, suppose that there is a veteran jockey who has run previous races in such non-ideal weather conditions. Then, the race by this veteran jockey which has the most similar weather conditions *accurately* reflects the conditions of the current race. However, suppose that there is also a very novice jockey who has not run previous races in such non-ideal weather conditions. Then, the race by this novice jockey which has most similar conditions will not accurately reflect the conditions of the current race, and instead may reflect very ideal weather conditions if that is all this novice jockey has run in. Therefore, this alternative definition of similar weather conditions will consider vastly different types of races for each of these jockeys. On the other hand, the proposed method to use a race with the same binned values will ensure that only truly similar races are considered. Therefore, binning yields the most consistent definition of similar weather.

O Prediction Target Non-Solutions

O.1 Predicting if a Single Horse is a Winner

Some prior literature uses models which predict whether a given horse will win, such that the input to the model contains features of the horse, optionally concatenated with features of the race, without information about any other horses in the race. This is notably done by Williams and Li (2008) and Davoodi and Khanteymoori (2010). As mentioned in Section 3, this approach makes a very strong assumption that a horse in a race runs entirely independent of other horses in a race, which we believe would not hold given the competition during a race as runners try to pace with respect to each other. Additionally, we believe that this assumption could fail because it doesn't consider the skill level of other horses in a race; perhaps a runner has done well in the past but recently moved to a different division such that the competitors have a higher skill level and so this horse cannot be expected to perform as well anymore. Nonetheless, one advantage of this approach is that it presents a binary classification problem, which is simpler than multi-class classification alternatives. Another advantage of this approach is that we can create well-formed inputs to the model even if we are missing information for *some* runners in a race. For this reason, we desire an approach that presents these advantages without making strong assumptions.

O.2 Predicting the Winner Given an Entire Race

Also popular within the literature is an approach which tries to predict the outcome of an entire race given all the entrants to the race, explored by Bolton and Chapman (1986) and Torné (2021). One disadvantage of this approach is that, since models are inflexible to variable-length input, this requires races with fewer than the maximum number of runners to be padded to the desired length. Such padding complicates the structure of the input and places the burden on the model to learn this structure. Another disadvantage of this approach is that it creates very wide input vectors, since the resulting vector is of length (max horses) × (features per horse). In our case these numbers would be 14 and 72 respectively, resulting in input vectors with 1,008 dimensions. A final disadvantage of this approach is that this strictly requires feature vectors for *all* horses in a race; even if we have perfect information on 13 of the 14 runners in a race, we still cannot make a well-formed input vector to such a model. For reference, after scrubbing and featurizing our data, only 4,584 races have *complete* feature vectors for *all* runners. Consider the tension that arises from simultaneously having larger input vectors coupled with fewer datapoints; per the prior discussion we would have fewer than 5 data points per feature, making this extremely susceptible to overfitting (as is speculated to have occurred in Torné (2021)). However, an advantage of this approach is that it allows the model to consider all runners simultaneously so that the runners can be measured up against each other rather than considered isolation. We desire an approach that presents this advantage without the high likelihood of overfitting.

P Pairwise Winner Approach Technical Details

There are four small technical details of the approach of predicting pairwise winners.

First, to enforce the fact that the order of the horses within the input vector matters, we will create one input vector for each *ordered* pair of horses in a race in our dataset. Specifically, for two horses A and B that run in a race, we will have both input vectors [A;B] and [B;A]. Accordingly, the target vector will be flipped for these two rows.

Secondly, we need to handle cases where horses finish in the same position due to a tie. One approach may be to make the target 0.5 so that it lies between 0 and 1. Ultimately, for the sake of simplicity, we chose to maintain the binary nature of our target vector by having the target across *both* orderings of a tied pair be 0 so that it appears that each runner won the pairwise matchup exactly once and lost the pairwise matchup exactly once. We may hope that this implicit contradiction drives a model towards indifference on the outcome, which would be correct in the truest sense.

Thirdly, although we have featurized the racing dataset, we will not end up using these features. Note that all the interesting features of a race are already encoded in the past performance of a jockey. The omission of race metadata from the input vector keep the dimensionality of the feature vector tractable and represents preemptive feature selection. In other words, the input vector will *only* have the concatenated feature vectors for the two runners.

Fourthly, we additionally preprocess the data by scaling all features to be in the interval $[0, 1]$. This ensures that the magnitude of weights on features should be directly comparable to each other.

Q Train-Dev-Test Split Nonsolution

Here, we demonstrate why using a simple random sample to achieve our split of the dataset is not suitable for purposes. Consider that, due to the use of past performance in our feature vectors, a feature for one input may be a label for another input. Therefore, we must be careful that the model does not see labels for inputs in the development or testing set while it is training. Otherwise, these datapoints may be trivially solvable purely because the model has inadvertently accessed information from the future, without actually learning anything.

To understand this concretely, consider that instead of the jockey's performance in their last three races, our feature vectors contained the jockey's performance in their last t races, for some large t. Note that everything to follow is still true for $t = 3$, but this failure mode is exaggerated as t increases. Now, suppose that drawing these datasets at random resulted in a split such that the jockey's i^{th} race was in the training set and their $(i-1)^{th}$ race was in the development or testing set. Then the model will see the jockey's i^{th} race during training and thus see t past performances, of which the $(i-1)^{th}$ race is included; let these past performances be the vector $\vec{x}$. Now, at testing time, the model will be asked to predict on the jockey's $(i-1)^{th}$ race. But, consider that the t past performances to the jockey's $(i-1)^{th}$ race will look almost identical to the t past performances of the jockey's i^{th} race, except prepended with some value p and lacking the last value. To be precise, for $\vec{x'}$ a vector of the past performances of the jockey's $(i-1)^{th}$ race and $\vec{x}$ a vector of the past performances of the jockey's i^{th} race, we have:

$$\vec{x'} = [p; \vec{x}_{0:t-1}]$$

Additionally, note that the outcome of the $(i-1)^{th}$ race is exactly $\vec{x}_t$. Therefore, a sufficiently expressive model can pattern match on the last entries of a vector then lookup the appropriate datapoint in the training set and return the correct label, $\vec{x}_t$, with high probability. This model simply takes advantage of the format of the dataset, without learning anything about horse racing and so this situation should be avoided.

R Distribution of Horse Racing Over Time

It is possible that the distribution of our data changes over time as the distribution of horses and jockeys changes over time. Then, the training set, development set, and testing set may come from slightly different distributions. Technological advances are a common reason that a distribution may shift over time. However, while technology may be used off the racetrack in the healthcare of horses, determination of odds, and handicapping, it is not present during the physical race, and so we may suspect its influence on the distribution to be minimal. Additionally, the nostalgia of horse racing may push back against substantial changes into the sport, though we cannot offer any evidence for this claim. Even if the distribution did shift over time such that the training set and testing set came from slightly different distributions, those involved in the sport care most about predicting future races, such that it betrays the usefulness of this analysis to do otherwise.

S Aggregating Pairwise Predictions

To use our model towards downstream prediction tasks, such as predicting the winner, top two finishers, top three finishers, or full ordering of a race, we will need to feed our model several pairs in a race and somehow aggregate the results. Ideally, the model, after feeding it each unique pair of horses in a race, would output a complete and non-contradictory set of pairwise winners that can be aggregated into a *full* ordering of the race. However, we do not expect these predictions to be non-contradictory, since this is not something we enforce during training. Therefore, we will anticipate such contradictions and instead use the a simple heuristic to create an ordering. Before we explain the heuristic, we introduce a definition:

Definition 3 (Aggregate Likelihood)**.** Let the function that the model computes over an input vector be f, where the image of the function is the interval $[0, 1]$ (by virtue of the sigmoid function). Additionally, let $[h; h']$ represent the input vector which corresponds to the *ordered* pair of horses h and h'. Then for a race r with horses H_r, among which is horse h, the *aggregate likelihood* of horse h in race r, denoted $\text{AL}(h, r)$ is

$$\text{AL}(h,r) = \sum_{h' \in H_r \setminus h} f(h,h') + \sum_{h' \in H_r \setminus h} (1 - f(h',h))$$

To understand this equation, consider that by construction of our featurized dataset, for an input vector $[h; h']$, the model computes the likelihood that horse h will finish earlier than horse h'. Equivalently, the model computes the likelihood that horse h' will finish after horse h, one minus which is the likelihood that horse h' will finish earlier than horse h. Therefore, for all possible ordered pairs, we are summing the likelihood that horse h will finish earlier than another horse in the race.

Now, a natural way to receive an ordering from our model on races that have full information is to calculate the aggregate likelihood for each horse then sort horses by their aggregate likelihood in descending order. To understand why this works, consider what a perfect model would output using this scheme. A perfect model would predict the actual winner of a race to finish earlier than all other horses in a race, thus achieving the highest aggregate likelihood. Continuing, a perfect model would predict the second-place finisher of a race to finish earlier than all horses in a race except for one, thus achieving the second-highest aggregate likelihood. This trend continues all the way to the last-place finisher of a race, which would be predicted to finish earlier than *none* of the other horses in the race, and so achieve the lowest aggregate likelihood.

Unlike the previous idea of relaxing pairwise winners into a full ordering, this is metric is resistant to contradictions, since it is simply additive. Taken to an extreme, for horses h_1, h_2, it could be the case that $f(h_1, h_2) = 1$ and $f(h_2, h_1) = 1$, a clear contradiction, yet the aggregate likelihood can still be computed with ease.

Thus, we have argued for the suitability and robustness of using the approach of sorting horses in descending order by their aggregate likelihood. We use this approach for downstream prediction tasks.

T Average Manhattan Distance

Here, we introduce a more lenient evaluation metric for evaluating the prediction of a full ordering of a race as opposed to using the binary evaluation metric which declares an output ordering to either be entirely correct or entirely incorrect.

To understand why we may want to be more lenient, consider that, for example, in a race with fourteen runners, it is still extremely favorable to be able to correctly predict the finishing position of twelve runners. Under a binary evaluation metric, correctly predicting the finishing position of zero runners and correctly predicting the finishing position of all but two runners would be punished equally. This harshness acts against our intuitive understanding of "goodness".

The metric we will use to evaluate the full predicted ordering over a race will be the average absolute difference between a runner's predicted and actual finishing position, where this average is taken over all runners in a race. At risk of overloading terminology, will to refer to this evaluation metric as the "average Manhattan distance" (AMD) of a prediction on a race. We illustrate this idea in the following example:

Example T.1 (Average Manhattan Distance)**.** Consider a race where horses actually finish in the order

Achilles, Beowulf, Constantine, Democritus

and the model predicts horses to finish in the order

Constantine, Beowulf, Achilles, Democritus

Then, the absolute difference between each runner's predicted and actual finishing position (referred to as "Manhattan distance") is captured in the table below:

Runner	Manhattan Distance
Achilles	2
Beowulf	0
Constantine	2
Democritus	0

Therefore, the average Manhattan distance of this prediction on this race is $(2+0+2+0)/4 = 1$.

A model performs better if the mean AMD across its predicted orders is *lower*, since this means that horses are nearer to their correct position, on average. Alternatively, a model performs better if the inverse of the mean AMD is *higher*.

U Nuancing Model Application to Betting

Here, we add nuance to the results of Figure 14. Primarily, since the public odds fluctuate very near to the start of a race, a bettor which wants to use the neural network or public odds would need to be well-connected to the network so that they can submit their bets with the most up-to-date knowledge just prior to the start of the race. Additionally, bets are not atomic, since an individual's bet changes the public odds. This is the reason we have approximated this with one dollar bets, though this is still imperfect. Still more, it is unlikely that an individual will be able to bet on as many races as are simulated because this would require an individual to travel all across Ireland to different racecourses, or even be at two different races at the same time. Finally, note that the races simulated span several years, such that this is certainly not a sustainable source of income. Therefore, while it is clear that the neural network outperforms the public odds baseline in this simulation, these results are nuanced.

V Neural Network Experiment Technical Details

We describe the process by which we created hand-crafted input to feed into the neural network to indirectly observe the feature importances. First, we randomly sampled 10% of the rows in the training data. The point of drawing a random sample from our training data is so that the data looks similar to that which the model was trained on, rather than us arbitrarily making input vectors that may not look like data drawn from the implicit distribution. Next, since each row represents a pair of horses, we set one horse in this pair to have identical attributes to the other horse in this pair, therefore producing rows where both composite horses have the exact same characteristics. When we run these hand-crafted inputs through the model, we get the model's prediction on inputs that represent logically equivalent horses. A perfect model would output a value near 0.50 on such inputs to represent its uncertainty since the horses are identical; this is generally observed since the average output of our model is around 0.62 on such inputs.

Next, for each group of features that is of interest to us (e.g. features involving past performance under the same temperature), we can change those and only those dimensions in our otherwise logically equivalent horses. We will make one horse have favorable values in these dimensions and the other horse have unfavorable values in these dimensions. We use an intuitive understanding of the features to determine favorability of different values. For example, when looking at a feature which is the finishing position of a horse in a previous race, we assume that lower finishing positions are more favorable whereas higher finishing positions are less favorable. Now, we can compare the output of the model on this modified input where the horses are equal *except* in one group of features to the output of the model on input where the horses are entirely equal. The importance of this group of features to the model's prediction is revealed by how much the output has changed.

We will use two levels of modification, referred to as "small" and "large" modifications, which represent increasing differences between the favorable value injected into one horse's vector and the unfavorable value injected into the other horse's vector. Therefore, for "small" modifications, the horses are still roughly equal, even at the modified feature. On the other hand, for "high" modifications, one horse has values that are substantially more favorable than the other horse. To be precise, we have chosen that for "small" modifications, one horse's modified features are set to the $1/3$-quantile value when ordered by favorability and the other horse's modified features are set to the $2/3$-quantile value when ordered by favorability. For "large" modifications, one horse's modified features are set to the $1/5$-quantile value and the other horse's modified features are set to the $4/5$-quantile value, driving a larger gap between the horses's favorability in these features.

In essence, by modifying exactly one group of features at a time and comparing it against the baseline where all features are exactly equal, we are able to attribute any change in the output to the modifications over this group of features, and thus gain insight into their importances.

W Logistic Regression Feature Trends Extended Discussion

This appendix further discusses Figure 15 as part of our interpretation of the results of the ablation experiment with the logistic regression model.

In the second row of Figure 15, we have grouped past performance features into those that encode a past finishing position and those that encode a past finishing time ratio and similarly compared across models. The goal of this comparison is again to establish that each model has learned non-noisy trends. In both models, features involving finishing time ratios are strongly more important than features involving the finishing positions. Intuitively this makes sense since the finishing time ratio, due to its continuous nature, is a more fine-grained metric of a horse's performance in a race, as opposed to the finishing position which only takes discrete values. To understand this by means of example, consider that a horse may finish in second place, losing by a millisecond or losing by several tens of seconds. In the former case, the finishing time ratio would be approximately 1, accurately capturing the intuitive "goodness" of this result for the horse under consideration, whereas in the latter case, the finishing time ratio would be much larger than 1, accurately capturing the intuitive "badness" of this result for the horse under consideration. However, the finishing position is the same in both cases and thus lacks this additional descriptive information. Overall, since this trend is seen in both models and matches our intuition, we gain more confidence that the logistic regression model has learned the data well. Furthermore, we confirm that finishing time ratios are more important than finishing positions, should we want to be more selective in our features during a future analysis.

Finally, in the third row of Figure 15 we have grouped past performance features by whether they encode the first, second, or third past race for some category. As before, we can hope that the models learn the same trends. However, in this case, the trends are not as uniform across the models, with the models placing different importances on the recency of the past performances. Unlike the prior discussion, there is also not an intuitive answer as to whether more recent or less recent races should be more important; on the one hand a horse is probably in a more similar physical shape as their most recent race but looking farther back into the horse's history may reveal information that is not captured by other variables like the public odds. Therefore, these results may be explained by the fact that both more recent and less recent races provide similar information, such that a learned model does not need to strictly prefer one over another and this minor discrepancy does not detract from our conclusion that model logistic regression models have learned the data well.

X Random Forest, Extra Trees, Decision Tree

Here, we highlight the results of using the random forest, extra trees, and decision tree models in the ablation experiment captured in Table 5. We have chosen to group these models together under an abbreviated discussion because these models all exhibit substantial overfitting. This overfitting is evidenced by a substantially larger accuracy on the training set than on the development set. For this reason, these models can be expected to have learned noise in the dataset that does not generalize to the full distribution over which the training set is implicitly drawn from. Therefore, an exploration into how these models make predictions would not be insightful to our analysis. Note that this overfitting has occurred despite conscious efforts to regularize the model, where these efforts are evidenced by our inclusion of parameters like `n_estimators`, `max_depth`, `min_samples_split`, `max_features`, and `min_samples_leaf`, which all impose some restrictions on the structure of these models. Searching over more aggressive values for these parameters may decrease the amount of overfitting, but the available computational resources do not permit this at this time. Ignoring overfitting, nominally, the random forest model and extra trees model do better without weather while the decision tree model does better with weather. Since the decision tree is the model that overfits the least out of these three and has performed marginally better with weather features, this gives some evidence that weather features are helpful towards the model.