Formula 1 Race Predictor

Iram Tazim Hoque
ihoque@purdue.edu
West Lafayette, IN, USA

Zhenhuan Su
su170@purdue.edu
West Lafayette, IN, USA

Jiangqiong Liu
liu3328@purdue.edu
West Lafayette, IN, USA

ABSTRACT

Many private sports-betting companies are using their private algorithms to predict winners. Still, there is no publicly available tool for the general public who wants to participate in sports betting. In particular, we built a website to help sports enthusiasts predict Formula 1 Grand Prix winners. Users get to choose which features or models to use for predictions. Besides, they are able to see the most important factors or features that determine the result. For evaluation, we conducted a quantitative evaluation on the test data and usability testing to test our website with potential users and get their feedback for our tool.

CCS CONCEPTS

• **Computing methodologies** → *Artificial intelligence*; • **Human-centered computing** → *Human computer interaction (HCI)*; *Information visualization.*

KEYWORDS

neural networks, visualization, regression, classification, machine learning, artificial intelligence, sports betting

1 INTRODUCTION

We created an interactive tool to predict Formula 1 Grand Prix winners based on historical data using different attributes from the drivers and the circuits. This is a niche but interesting area of research for F1 racing teams as well as the general public. For F1 teams, they are interested in which potential driver to hire to bolster the team. For the general public, betting on Formula 1 can be both entertaining and profitable. They would like to know which factors are impacting the winners in each race and depending on the factors of the upcoming race, who is more likely to win the race. These are something they may want to know both for the sake of curiosity and if they want to participate in sports betting.

Our tool is available to the public, even if it does not perform quite as well as the closely guarded proprietary tools. Moreover, it is highly interactive, where the winner prediction result depends on the user inputs. We provide many user filters, including what model to choose and other factors related to past races. Users can also see the final output and the most important factors/features that determine the final output.

We worked with data from the year 1983 till the year 2019 to predict the winner of the upcoming races. Ergast Developer API has all the data publicly available[1]. We filtered out unnecessary features from the dataset and ensured we used only the most useful features. We also kept the other features so that the users, if they chose to do so, might use them for the predictions as well through our interactive website.

CS593 Fall '22, Fall 2020, Purdue University
2022. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

For evaluation, we conducted a quantitative evaluation on the test data to verify how well our model did, and the performance metric was accuracy. We also conducted usability testing with potential users to investigate the best way to display explanations to help users better understand how models work and get users' overall satisfaction with the models and interface.

2 MOTIVATION

To our knowledge, currently, there are very few(if any) predictors open for public use. All the sports-betting companies and the F1-racing teams are very secretive about their proprietary data mining tools. To set proper odds, online gambling industries are very particular about the level of precision when determining the likely winner of a race. Also, they want their tools to be a closely guarded secret from their competitors. This ultimately means no such tools are available for the general public to use.

3 RELATED WORK

We studied few papers related to our research. Since, this is a very niche research area, we didn't actually find an exact previous work on predicting f1-racing. Instead we looked at some machine-learning based sports winner prediction type of modelling papers.

The first paper[2] talks about using Artificial Neural Network (ANN) as a tool for sports result prediction. Even though it doesn't really provide any specific algorithm, it gave us insight into general framework of ML based sports-predictor tools.

As we know, there are various types of racing related sports, each with their own unique twists and each vastly different from each other. But, they all share some common concepts(i.e. all sports racing involves some form of competitors competing for the top spots and the probability of a competitor winning the race depends on the race environment, his skills as well as his form). So, we also looked into other specific ML-based racing-sports predicting researches like [4], which describes a ML based Learn-to-Rank approach for Road-cycling race, and [3], which describes a ANN based approach to predict horse-racing results. These papers gave us valuable insight into exactly how to approach these type of problems and how to model and solve these. We learnt that for racing-sports prediction problem, data analysis and feature engineering is more important than model construction. Engineering good feature-sets and producing a good dataset, does more than half the work for solving the whole problem. A good model cannot overcome bad feature-sets and noisy data for this type of problems.

Our tool features not only a f1-race predictor model but also web-based frontend for users to interact with our model and use it for prediction. Nowadays, web applications have become is an important platform for people to use on a daily basis. There are already many different applications on the market. It is important for developers to design an application that is easy to use for user

understanding and acceptance, but this is not an easy task. Developers need specifications to guide the design process. Web Site Design Method (WSDM) gives five different design phases[6]:

(1) Mission Statement Specification: identify the function of the web application as well as the target users
(2) Audience modeling: target users are refined into audience classes
(3) Conceptual Design: information and functionality of the web applications are specified at the conceptual level
    - Task and Information Modeling: information and functionality are specified
(4) Navigational design: all navigation possibilities in the web will be designed accordingly
    - Site Structure Design: map structuring the web application into pages
    - Presentation Design: user interface layout of the web applications as well as the look and feel of the web itself
    - Logical Data Design: database schema and the mapping between the conceptual data model and the actual data source are constructed
(5) Implementation design: using all those sets of data models constructed in the design phase previously

We also took a look into papers related to Explainable Artificial Intelligence(XAI)[7], which talks about various XAI element to help users easily understand a model and its working. Now, our tool doesn't exactly have any XAI portion, but this paper still gave us insight into how different UI and interactive elements can be used in our web tool to make it more user-friendly.

4 DESIGN

Our goal for this project is to create a tool for F1 racing prediction. It consists of 2 major portions.

4.1 Machine Learning(Data Mining)

In this section, we explain the whole process of data and feature engineering as well as model construction.

4.1.1 ***Data and Feature Engineering.*** The data provided by Ergast API[1] is divided into multiple different json files for each year of races, constructor standings, driver standings etc. We had to consolidate all these jsons into a single table(Our feature table). We had to do quite a bit of feature analysis and feature engineering for this dataset. Not all of the feature were straightforward. For example, in the original data, the *driver_points* was noting the driver's acquired point during that specific race. But we can't really feed that into our model for prediction because that would be cheating(as its the data about the current race we want the model to predict). So, instead, a more useful feature for the ML model would be that driver's accumulated points until the current race for the current season. That's exactly what we calculated from the dataset and made it into the *driver_points* feature. We also had to consolidate some features into a single feature(i.e. *qualification_time* was calculated from 3 other columns in the dataset). Moreover, we needed to drop some features we felt unnecessary. We had to clean the data to remove any invalid/empty values so that the models won't be impacted by the noisy data. Ultimately we ended up with 15 distinct features in our final cleaned-up dataset which contained race data from year 1983 till year 2019. The feature importance graph for our dataset can be seen in Figure 1.

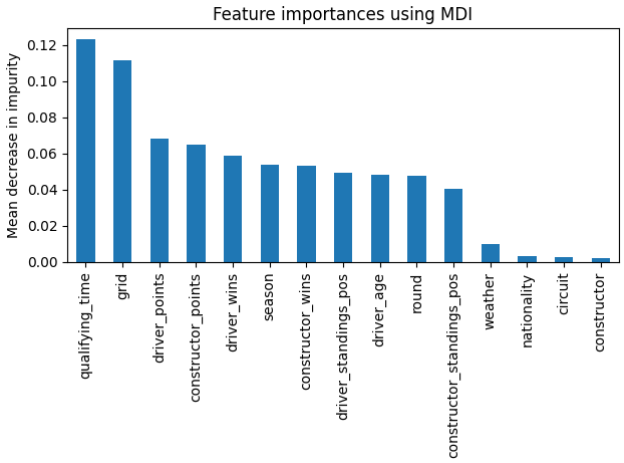

Figure 1: Feature importance of our dataset based on Mean Difference in Impurity(MDI) metric.

4.1.2 ***Model Selection and Design.*** Our feature set is simple enough, so we felt that going into advanced deep-learning is wasteful. So, we limited our model choices to Machine Learning models. We used models mainly from scikit-learn library since they provide ML models in a convenient package. We did try out some deep learning models from pytorch, but for our dataset, they didn't really provide enough accuracy gain and also they were much slower than the basic ML models from scikit. So, we decided not to use those in our tools.

This race predictor problem can be modeled either as a classification problem(Highest probability class is the winner, 2nd highest is 2nd and so on) or a regression problem. We decided on keeping both options as choices for the users to select. The models we selected for regression are:

(1) Linear Regression
(2) SVM Regression
(3) Neural Network Regression

Our classification models are:

(1) SVM classifier
(2) Neural Network Classifier
(3) Random Forest Classifier

4.1.3 ***Hyperparameter Tuning.*** We did hyperparameter tuning for each of these models(using validation data). We used a 60 : 20 : 20 split for train, validation and test data. We used 10-fold cross-validation during the hyperparameter tuning phase. Our hyperparameter tuning graphs can be seen in Figure 2.

4.1.4 ***Recent 10 years of data for training.*** From our domain knowledge, we know that F1 races change their rules and regulations drastically every few years which changes the dynamics of the races dramatically. Therefore, we thought that using only the

(a) Random Forest Classifier (b) Neural Network Classifier (c) SVM Classifier

(d) Neural Network Regressor (e)SVM Regressor

Figure 2: Hyperparameter Tuning Graphs for 5 of our models excluding linear regressor model since it doesn't have any tunable hyperparameter

past few(10 years) years of data(from the year we are predicting) for training can be helpful in increasing the accuracy of the model and our preliminary test results verified this. So, we chose to keep this(partial year range selection for training data) as a feature of our tool.

4.2 Human-AI Interaction(Web app for user interaction)

We plan to(already in progress) create a web app to give users the opportunity to interact with the model and see the output visualizations in a beautiful and easy-to-understand way. We are using Python Flask as the backend framework. We are also using Bootstrap as our CSS framework. We considered the possibility to use MaterializeCSS as our CSS framework but ultimately decided on Bootstrap because of its huge customizability and the vast amount of resources available online.

4.2.1 ***High Level Design.*** Our entire process of designing the UI is based on the five stages of WSDN[6].

(1) Mission Statement Specification: Our target audience is people who like F1 racing and will bet on it. Since we are using different AI models, this requires users to have some understanding of machine learning. The function of this web app is to give predictions of a winner in multiple drivers.
(2) Audience modeling: Although we require users to have some understanding of AI, each user has a different level of understanding. Some users know all the models, but some users know only one or two of them. Our UI should accommodate both types of users.
(3) Conceptual Design: The user needs to participate in the whole process, from selecting a model, training the model, and finally getting the prediction results
    - Task and Information Modeling: In each step the user's input is requested, and accordingly the UI has to explain to the user what input is required
(4) Navigational design
    - Site Structure Design: In our design, we separate the whole process into 3 pages: introducing different models, training one model, and predicting of winner
    - Presentation Design: The articulation of the three pages must be tight and smooth. How the UI presents different data to the user should be simple and easy to understand. For example, use some charts or explanatory marks.
    - Logical Data Design: Some data can be displayed directly on the page, but some minor data can be hidden temporarily and displayed when needed
(5) Implementation design: see details in next section

4.2.2 ***Implement.*** In this section, we will discuss the details of UI and how we apply the design to the real web app.

- In Figure 3, we present the bar graph for the best accuracy of the six different AI models. The main function of this page is to give an overview of all the models that we use and give the user choices to look at some secondary data. Models in 2 categories(regression and classification) are with overlapping names, so I separate them into 2 colors(apply on categories, columns, and buttons). The best parameters are displayed when the user's mouse hovers over a model

column. Also, only when the user clicks the button, they can see the hyperparameter tuning graph for that model. As mentioned in 4.2.1, some users know AI models very well, but some users don't have much background in AI. For the later type, they tend to skip all the parts they don't understand. We just need to show them east restrictive and must-know data(what models we have). Users like being given the choice of which models to view the best parameters and if they want to see some content or not. If we show all the data at once, the user might be confused by all the data and feel forced to look at it. At last, the "train model" button informs the user the next step is to train the model.

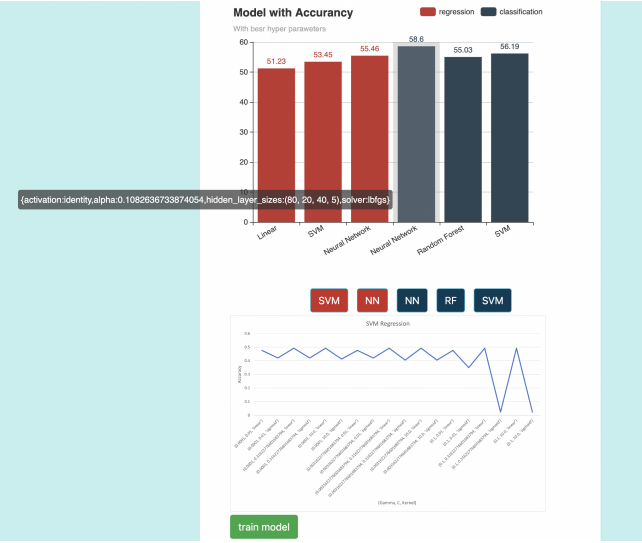

Figure 3: UI: page1 bar graph

- Figure 4 5 6 show the three parts of training model. In the first figure, the user can choose which model they want to use, and they can input the required hyperparameters. Each hyperparameter has a question mark icon to show what is it and what format it needs. For users who don't know what to input, they can choose the "base model" button to use the best hyperparameters for this mode as mentioned on page 1. In the feature figure, the user has 2 options: use all the features or select features they don't want in the model training. The "feature importance" button leads to the graph of all feature importance. It gives users a very clear view of the relationship between all the features.
  In the year figure, we also provide 2 options for the user: they can train the model with data from the year range or just all data.
- Figure 7 shows the prediction page. The question mark icons give explanations for this page and all features. We also give 2 rules to help user to understand how to predict the winner. For each feature, we input the default value so that it can save the users some time to input. After the user clicks prediction, we will give the result of it. At last, the "start from beginning" button directs the user to page 1 so that they can choose a different model and retrain it.

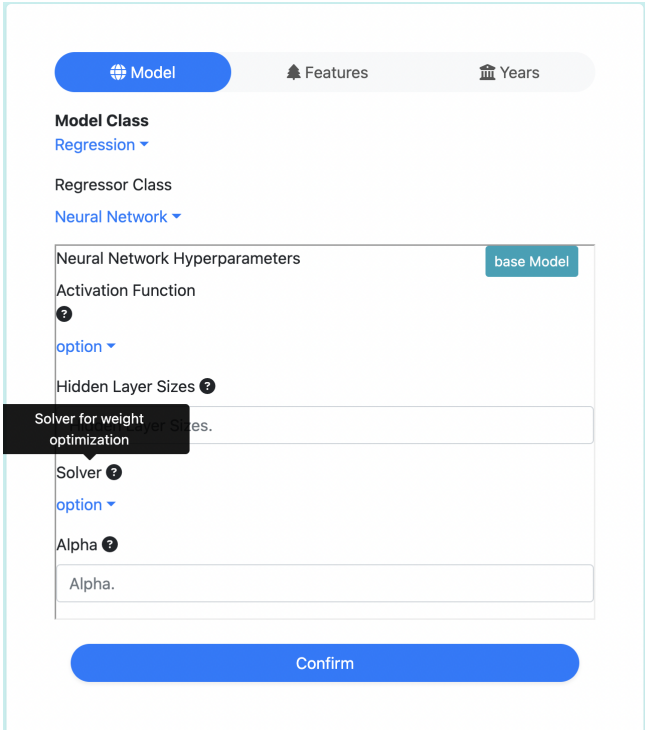

Figure 4: UI: page2 model

In conclusion, the main idea of the UI design is to give the user confidence in the whole process and let them think they have control of everything. We try our best to decrease the effect of the black box problem[5]. With enough explanation, transparent AI processing, and freedom of choice, the prediction result should be more trustworthy.

5 EVALUATION

We conducted two types of evaluations for our tool. Firstly, a quantitative evaluation of our models. Secondly, a user study of our tool as a whole too see how users perceive the usefulness of the tool. These two types of evaluations are discussed below.

5.1 Quantitative Evaluation and Results

5.1.1 ***Performance Measure.*** We split our dataset into training and test subsets on a 80 : 20 split for evaluation. The test dataset only contained placement of each driver on a specific season, specific round race. We couldn't run the built-in performance metric function from scikit on that data because those functions would look at the data row by row and see if the position of the racers match, which in our case would mean only 1 winner and about 15-20 loser per race. So, even if our tool actually do predict the 1 winner correctly but predicts the ordering of the other 15-20 racers, the reported accuracy would be horrible and won't be indicative of what we want to achieve. So, we had to create our own evaluation function which would look at the data race by race and see if the models can predict the winner correctly and performance metric would be indicative of how many percent of the races our tool

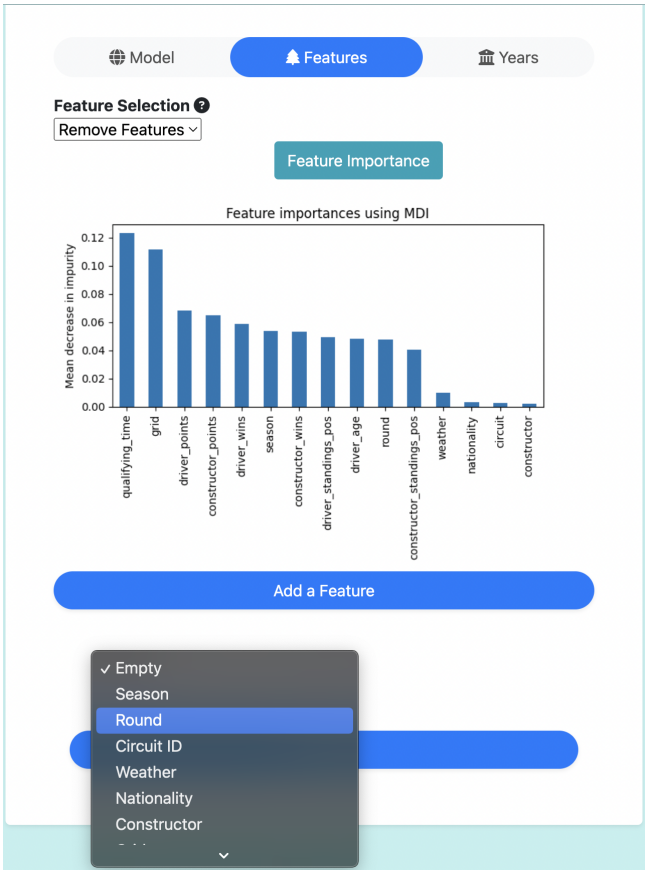

Figure 5: UI: page2 feature

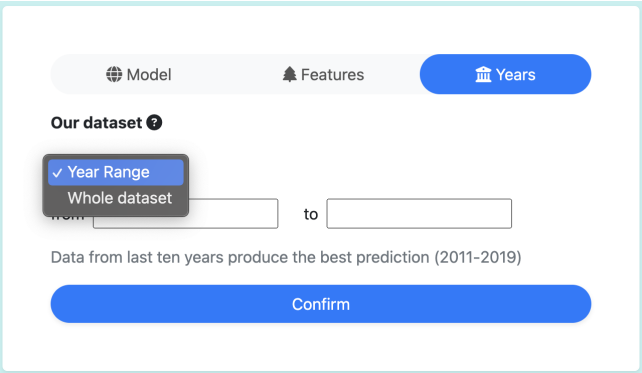

Figure 6: UI: page2 year

predict the winner for correctly. The performance metric we used is *Accuracy*. So, a reported accuracy of 55% by our evaluation function would mean, our model had accurately predicted the winner of 55% of all the races in the test data.

5.1.2 ***Results***. The highest accuracy we achieved on our dataset was 58.6% and the best performance model was *Neural Network Classifier*. In general, the classifier models performed better than the

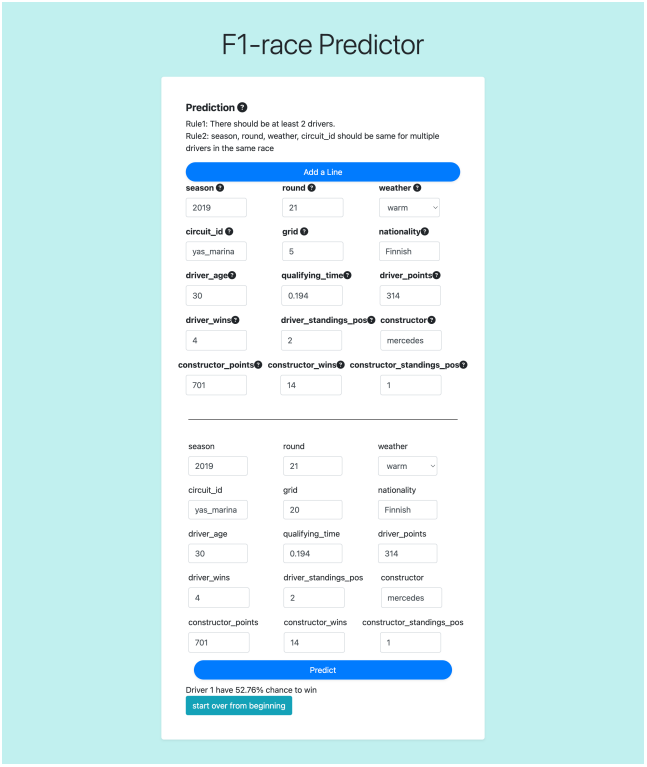

Figure 7: UI: page 3 prediction

regressor models. The performance measure of all 6 of our models can be seen in the Figure. 8

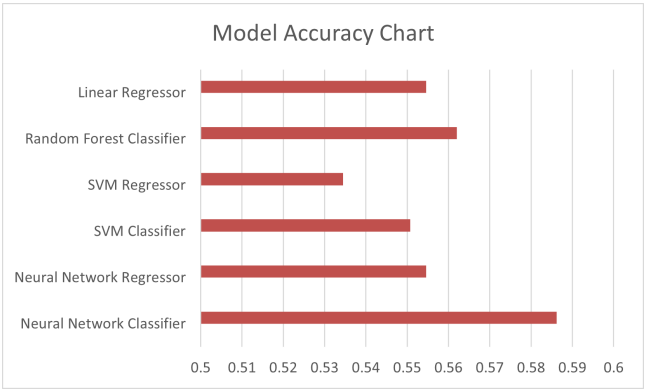

Figure 8: Accuracy of our 6 models, each with their best combination of hyperparameters

As mentioned before, we hypothesized that, using only the past few years of data instead of the whole data can possibly capture the recent trends better and may provide higher accuracy. So, we also made a comparison, for the 3 classifier models(since they performed better overall than the regressor models) between the accuracy when the whole dataset was used vs when only the recent 10 years of data was used for training. This comparison can be seen in the

Figure. 9. we can see clearly that all the models performed better when past 10 years of race data was used for training proving our hypothesis correct.

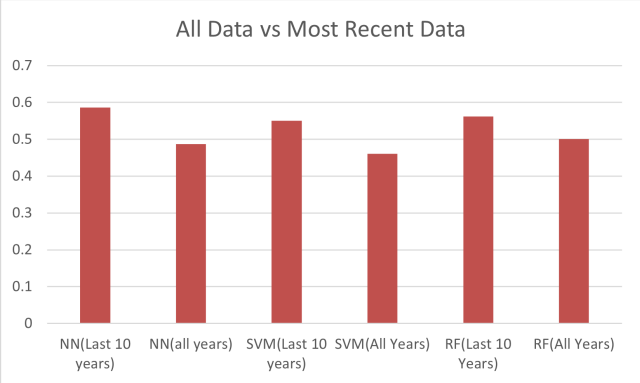

Figure 9: Performance comparison of whole dataset-trained models and partial dataset-trained models

5.2 User Study Evaluation Methodology

Prior to the user study reported below, we conducted a pilot study with 2 participants to evaluate the study design and the test setup. Below, we only report the final experiment.

5.2.1 ***Research Questions and Hypotheses.*** Given the complexity of Machine Learning models and the number of required inputs from users in the Model, Features, Years, and Prediction sections, we added explanations for inputs in the interface to help users better understand how models work. Since we were unsure about the best way to display these explanations, we designed an experiment to evaluate and investigate whether an interface design with question mark icons could help users better understand the models. We hypothesized that it takes users less time to make a winner prediction using question mark icons showing additional information on hover.

5.2.2 ***Participants.*** We recruited 12 participants(7 males and 5 females) who are currently studying Computer Science at Purdue University. We pre-selected participants according to the criteria that they need to have some background in Machine Learning because the interactive tool we designed for F1 race prediction requires some basic understanding of different Machine Learning models.

We gathered information on participants' demographic data, background in Machine Learning, and sports betting experience using an introductory questionnaire. Because we pre-selected our participants, in general, our participants have studied ML or have done projects with ML before, and the average years of participants' ML experience are 2.5. 17% of them have done sports betting, and only 1 participant has betted on an F1 race.

5.2.3 ***Study Design.*** We did a between-subjects experiment with 2 designs. One design shows directly on the page the information about each required input for models. The other design uses question mark icons to show information only when the user hovers over icons. We chose a between-subjects design because the difference between these two designs is very subtle, and using a within-subjects design would definitely make subjects' reactions to one design contaminated by exposure to the other design.

We randomly assigned 6 participants to each design and asked participants to predict a driver with the highest winning probability among several drivers using random order of sections. In order to keep participants involved in the situation and get an idea of what they were thinking while completing the task, we asked the participants to think aloud. Besides these, we noted the time each participant used to complete the task.

After the experiment, the participants had to fill out a closing questionnaire that asked them to judge their understanding of each section on a scale of 1 to 5(easy to difficult). The questionnaire also asked the participants what they liked and disliked about the interface and what they would change if they could redesign it.

5.3 User Study Evaluation Results

5.3.1 ***Task Completion Time.*** Since the sample size is 6 for each design, and the data sets follow a normal distribution, we used an unpaired t-test as an inference statistic to determine if there is a significant difference between the means of two groups. The mean time of the group with question mark icons is about 2 minutes and 11 seconds, and the mean time for the other group is around 2 minutes and 50 seconds. We chose the significance level to be 0.05. The t-score, or the ratio of the difference between the mean of the two sample sets and the variation that exists within the sample sets, is 4.3348. The two tailed P value equals 0.0015. Therefore, by conventional criteria, this difference is considered to be very statistically significant. We can conclude that it takes less time to make a winner prediction using an interface design with question mark icons.

5.3.2 ***User Feedback.*** We asked our participants to think aloud and give continuous feedback about the user interface. For the design with explanations directly showing on the interface, some participants stated that the page looked a little messy with the amount of text and felt they were forced to read them. One participant mentioned that the explanations were useful during the first round of prediction, but he would not like to read them afterward. For the design with question mark icons, some participants said the information provided was helpful when they got confused.

From the closing questionnaire completed by participants after the experiment, there is not much difference between these two designs regarding understanding the inputs needed in each section. Most participants thought the added explanations were helpful for filling out the inputs for models regardless of the way of showing them in the interface. People with less experience appreciated the explanations more than people familiar with different models. Since they all had learned Machine Learning, most of them stated they trusted the winner prediction given by this tool. However, most of them were not sure if other people interested in Formula 1 race betting would like to use this tool because they did not have experience in sports betting.

6 DISCUSSION

Due to time constraints of this project, we could only dedicate limited amount of time to the data,feature and model engineering of the project, leading to quite mediocre accuracy of the final model. More advanced data analysis to engineer the features better would likely lead to a overall better performing model.

Given the constraints of this project, we followed the waterfall process to develop the interactive tool. We thought about the user needs, implemented the functionalities, and did a user study afterward. We thought we had included all functionalities that users might want, but the user study showed it otherwise. For example, they would like to be able to revisit the model accuracy graph while filling out the inputs for other sections, and we did not have a button for it. An agile development approach to incorporate user feedback and modify our tool iteratively would make our tool cover more user needs.

Most of our participants did not have prior experience with sports betting, even though they all liked watching sports games. We had limited access to participants with both a Machine Learning background and sports betting experiences because we recruited participants only on the Purdue campus, and in the United States, most states require residents to be at least 21 years old to take part in sports betting. According to Forbes, most online gambling formats require bettors to be 21 or older to bet with real money. This applies to sports betting and online casino gambling in almost every legal market. It would be worthwhile to conduct usability testing with more participants who have done sports betting on other online platforms so that we could make a comparison between our tool and those platforms to better understand the usefulness of our tool.

We specifically designed our tool with the target users of people who have a basic understanding of different regression and classification models, and we evaluated our tool with people who had some Machine Learning background before. It would be interesting to experiment with the novice and test if our tool is good enough for the novice to understand how winner predictions are made. Most F1 fans in the real world do not have a Machine Leaning background, and they likely know nothing about models. Future work could include using different model visualizations to improve the explainability and interpretability of the tool so that it could cater to more F1 fans.

7 CONCLUSIONS

Formula 1 Grand Prix winner prediction is a niche and interesting area of research. Many private sports-betting companies have their private algorithms for predicting winners. We think an interactive tool available for the general public who wants to participate in sports betting would be helpful. We developed a tool for people with basic knowledge of Machine Learning models. Users can choose which features or models to use for predictions and see the most important factors or features that determine the result. We performed a quantitative evaluation on the test data. We also conducted usability testing to test our tool with potential users and get their feedback. Future research should include people with different ML and sports betting backgrounds and improve the explainability and interpretability of the tool to cater to more F1 fans.

REFERENCES

[1] 2022. Ergast Developer API. *Retrieved from* (2022). http://ergast.com/mrd/

[2] Rory P. Bunker and Fadi Thabtah. 2019. A Machine Learning Framework for Sport Result Prediction. *Applied computing informatics* 15.1 (2019), 27–33. https://www.sciencedirect.com/science/article/pii/S2210832717301485

[3] Alireza. Davoodi, Elnaz Khanteymoori. 2010. Horse racing prediction using artificial neural networks. *Recent Adv. Neural Netw. Fuzzy Syst.Evol. Comput.* (2010), 155–160. https://www.researchgate.net/publication/228847950_Horse_racing_prediction_using_artificial_neural_networks

[4] Leonid et al. Kholkine. 2021. A Learn-to-Rank Approach for Predicting Road Cycling Race Outcomes. *Frontiers in sports and active living* 3 (2021), 714107–714107. https://pubmed.ncbi.nlm.nih.gov/34693282/

[5] Hui Wen et al. Loh. 2022. Application of Explainable Artificial Intelligence for Healthcare: A Systematic Review of the Last Decade (2011–2022). *Computer methods and programs in biomedicine* 226 (2022), 107161–107161. https://www.researchgate.net/publication/363863038_Application_of_Explainable_Artificial_Intelligence_for_Healthcare_A_Systematic_Review_of_the_Last_Decade_2011-2022

[6] Matthew Wee. Mubin, Siti Azreena Poh. 2019. Web Application Design Methodology: A Review. Journal of Advanced Research in Dynamical and Control Systems. 10 (2019), 14–19. https://www.researchgate.net/publication/339252719_Web_Application_Design_Methodology_A_Review

[7] Mengchen Liu Jun Zhu Shixia Liu, Xiting Wang. 2017. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics* 1 (2017), 48–56. https://doi.org/10.1016/j.visinf.2017.01.006