2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA) | 979-8-3503-0426-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCUBEA58933.2023.10392224

F1 Race Winner Predictor

[1] Priya Shelke, [2] Riddhi Mirajkar, [3]Anurag Pande, [4]Sruja^{n} Kale [5]Yash Paralikar
[1,2] Faculty, Vishwakarma Institute of Information Technology, Pune
[3,4,5]B.Tech Information Technology, Vishwakarma Institute of Information Technology, Pune
[1]Priya.shelke@viit.ac.in , [2]Riddhi.mirajkar@viit.ac.in, [3]yash.22010118@viit.ac.in, [4] anurag.22010621@viit.ac.in,
[5] srujan.22011200@viit.ac.in

***Abstract*— With the help of machine learning, a winning Formula One (F1) race prediction model is what this project hopes to create. The model is trained using historical data from F1 races, such as lap times, sector times, qualification times, and information about the drivers and teams. To make the predictions we will be using Python and Support Vector Machines (SVM). The outcome of this initiative could be quite interesting for F1 fans and could influence wagering and other related activity. With this project we aim to create a machine-learning model that can forecast an F1 race winner based on a variety of inputs or display the effectiveness of SVMs by comparing predicted values and actual values.**

Keywords*— *Formula One, F1 race, machine learning, neural network, prediction, data collection.

I. Introduction

Formula One (F1) is a high-speed, high-tech sport that attracts millions of fans worldwide. With the advent of machine learning and data science, it is now possible to predict the winner of an F1 race with a high degree of accuracy. In this research paper, we developed a machine learning model which can accurately predict the winner of an F1 race using classification techniques.

We used the approach described in the article titled "Formula 1 Race Predictor" by Mark Nagelberg as a basis for our project. This article proposes a machine learning model that predicts the winner of an F1 race based on various data inputs, including lap times, sector times, qualifying times, track characteristics, weather conditions, and driver and team information.

For our project we are using python and the Scikit- learn library. It provides a wide range of tools for data mining, data analysis, and machine learning. Scikit-learn is built on top of NumPy, SciPy, and matplotlib, which are other popular Python libraries for scientific computing. Scikit-learn is widely used in industry and academia for a variety of applications, including NLP, computer vision, finance, and even biology. It is a powerful and flexible tool for machine learning, and its popularity is due in part to its user-friendly interface and extensive documentation.

Furthermore, for our project we our using Support Vector Machines (SVMs). SVMs are a type of learning algorithm that are used for classification, regression, and other tasks. They work by finding the best hyperplane that separates the data points of different classes in a feature space.

II. LITERATURE SURVEY

Here we have referred some state-of-the-art papers in this domain.

A paper [1] from a book about trends in Machine Learning looks at cost-effective alternative options for Formula 1 racing teams. They performed some research on the current methods of data collection, analysis and prediction. It was discovered that a big portion of the league's racing firms require a cheap, effective, and automated data interpretation method. The need for powerful prediction software grows, just as the modern trend behind F1 increases.

The research of Léon Sobrie [2] shows use of tree-based models in similar predictions. Their paper shows 3 different analysis, the first focuses on top 3 finishers and high performances in F1 races. The second analysis focuses on actual completion of the race, further helping make the races safer and minimize human and technical errors. The final analysis comprises the qualifying ability for the race to make sure the teams have 2 drivers at the start

Another paper[3] talks about making predictions in F1 or motor racing by using Artificial Neural Network(ANN). It is divided into two parts, the first part explains what an ANN is and gives an in-depth study of its many features, functions and layers. The second part focuses more on the implementation and methodologies such as data collection, comparison and much more. Some points in paper also talk about minimizing errors by regularization.

Speaking of regularization in Machine learning techniques, a research paper[4] in 2018 also goes in depth about the various methods of regularization. Not every method or technique is going to work in every situation, hence specific methods need to be applied for specific models. The model the paper primarily focused on was ANN. It used libraries from Tensorflow, and performed an analytical comparison between each algorithm and method.

Getting back to Formula 1, an interesting research paper[5] shows the significant importance of pit stops. It focuses less on predicting results and more on strategizing the use of these pit stops. In their paper they have presented their own Virtual Strategy Engineer (VSE) and described how it improves on the pre-existing methods like solving quadratic optimization problems or studying race simulations. Finally they have provided an 'example race' to show their VSE.

Since our project is using SVMs we have referred some papers which purely focus on use and features of SVM. A survey [6] teaches us about the use of SVMs in learning, classification, regression and forecasting. They have summarized almost everything there is to know about SVMs and even talked about the various parameters and how to optimize them. They have also provided information about new types of SVM, like FSVM and TSVM. Their primary focus is in the area of mobile multimedia. In the end, the paper discusses more about the future direction of SVM and its improvements.

979-8-3503-0426-8/23/$31.00 ©2023 IEEE

We also looked at papers with previously mentioned FSVM[7] and TSVM[8]. These papers provide detailed explanation on their respective subjects. SVMs have been used in many other prediction models such as predicting quality of soil[9], predicting diseases like diabetes in patients[10], school student performances[11] and even early detection of Covid-19 in patients[12]. We also looked at papers more focused on using SVM in predicting races like horse races[13].

Further exploration of SVM and its many uses, as well as comparisons with other models were studied in this paper [14]. The paper mentions how SVMs have more advantages than other algorithms in cases such as analyzing problems theoretically using concepts from computational learning theory.

Finally, we needed to look at the specific parameters for our SVM. There are 4 parameters that are typically used: Linear, polynomial, RBF and sigmoid. A research paper [15] published by IEEE gives us proper insight on how each parameter and kernel works and how they are tuned for optimal results. It also goes in-depth about the multiple uses of SVM in our current technological age.

III. Methodology

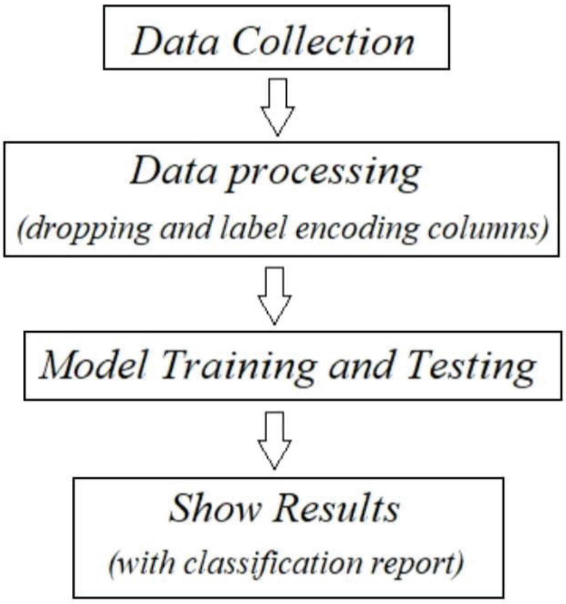
Data Collection
Data processing
(dropping and label encoding columns)
Model Training and Testing
Show Results
(with classification report)

Fig. 1. Step-by-step process

Our proposed methodology for developing an F1 race winner predictor using machine learning includes the following steps:

Data collection: We collected our data from the Ergast API. Ergast was created to make it easier for developers to access and use Formula 1 data, such as driver and team statistics, race results, and lap times. Ergast has become a popular resource for developers who are interested in working with Formula 1 data. It is used by a wide range of applications and websites, from fan-made data visualizations to professional sports analytics platforms.

Data preprocessing: We then preprocess the collected data by cleaning and transforming it into a format that is suitable for analysis. This may include droping columns from the files taken from Ergast or changing datatypes(mostly from string to int)

We have kept data relating to past races, like qualifying drivers, constructors, circuits, time, grid position and constructors. Total 20 features are currently used in the model. These 20 features are still however subject to changes. There are a total of 819 rows in the final dataframe.

Model working: For selecting the model we have options like Random Forest Classifier, SVM and neural network. We chose the SVM model and will train it by using the APIs provided by Scikit-learn.

For SVM, the hyperplane equation for a linearly separable data set is:

$$w.x + b = 0$$

where w is the weight vector, x is the input vector, and b is the bias

The model will then be trained and tested, and once the model is ready we ask it to print the output results. For training and testing, the data is split into two sets for model: training and testing. We took the most widely used approach of using 20% of the data for testing and rest 80% for training.

Model evaluation: We can evaluate the performance of the developed model using various metrics, such as accuracy, precision, recall, and F1 score. There is a very simple function provided by the Scikit-learn library which gives us a complete classification report and compares the predicted values and actual values.

The kernel function in SVM will dictate the shape of the decision boundary in the high-dimensional feature space. For our project, we received the most satisfactory outputs from linear and polynomial functions and performed further comparisons between the two.

The linear kernel is a simple kernel function commonly used in Support Vector Machines (SVMs) that can handle linearly separable data. The kernel function takes the dot product of the input vectors x and x' and adds a bias term b, resulting in a scalar value:

$$K(x, x') = x . x' + b$$

where x and x' are input vectors and . represents the dot product.

The polynomial kernel takes the dot product of two vectors, x and y, and raises the result to the dth power. The value of α and c are determined during the training phase of the SVM algorithm, which involves finding the optimal decision boundary that separates the input data points into their respective classes. The degree of the polynomial, d, is also a hyperparameter that can be tuned to improve the accuracy of the SVM classifier.

The polynomial kernel for a Support Vector Machine (SVM) is defined as:

$$K(x, y) = (\alpha\ x\text{^}T\ y + c)\text{^}d$$

where:

- x and y are input data points
- α is a hyperparameter that determines the influence of each training example on the decision boundary

- c is a constant that determines the offset of the decision boundary from the origin
- d is the degree of the polynomial used for the kernel function

Results: The output of the model tells us whether or not the driver finished with points in the race. Moreover it should also tell us if it had a ‘podium finish’(finished 1st,2nd or 3rd). Since SVM works best with floats and integers, the output gives us a number :

1 means podium finish

2 means points finish

3 means no points or did not finish.

IV. Classification Report

The scikit-learn classification report provides several performance metrics for a classification model, including precision, recall, and F1-score. The formulae for these metrics are:

Precision: Precision is the ratio of true positives (TP) to the total number of positive predictions (TP + false positives, FP). It represents the accuracy of positive predictions made by the model.

$$Precision = TP / (TP + FP)$$

Recall: Recall is the ratio of true positives (TP) to the total number of actual positive instances (TP + false negatives, FN). It represents the ability of the model to identify positive instances.

$$Recall = TP / (TP + FN)$$

F1-score: The F1-score is the harmonic mean of precision and recall, and provides a balanced measure of the model's performance.

$$F1\text{-}score = 2 * (precision * recall) / (precision + recall)$$

Support: The support is the number of instances in each class.

The scikit-learn classification report provides these metrics for each class in the classification problem, as well as an overall weighted average.

The below table compares the classification reports for the linear and polynomial kernels.

TABLE I. Comparing precision

	Precision		
	Podium	Points	No points
Linear	1	0.9	1
Polynomial	0.98	0.96	0.94

TABLE II. Comparing Recall

	Recall		
	Podium	Points	No Points
Linear	0.95	1	0.97
Polynomial	0.99	0.91	0.97

TABLE III. Comparing f1-score

	f1-score			Accuracy
	Podium	Points	No points	
Linear	0.98	0.95	0.98	0.97
Polynomial	0.98	0.93	0.95	0.96

V. Conclusion

In this research paper, we have presented a machine learning approach to predict the winner of an F1 race based on historical race data taken from Ergast API.

With the help of Scikit-learn library we were able to use the SVM model to predict the race winners and classify them based on finishing positions. The classification report shows accuracy of over 97% which only increases the mode we improve the data and features.

Overall, our research shows that machine learning can be an effective tool for predicting the winner of an F1 race. The models we have developed can be used to provide valuable insights to race teams and enthusiasts alike, helping them to make more informed decisions when it comes to predicting race outcomes. As such, we believe that our work has the potential to make a significant contribution to the field of F1 racing and machine learning.

References

[1] Kumar, M. & Preethi, N.. (2023). Formula One Race Analysis Using Machine Learning.

[2] Sobrie, Léon. "SIFTING THROUGH THE NOISE IN FORMULA ONE: PREDICTIVE PERFORMANCE OF TREE-BASED MODELS."

[3] Stoppels, Eloy. Predicting race results using artificial neural networks. MS thesis. University of Twente, 2017.

[4] Ismoilov, Nusrat & Jang, Sung-Bong. (2018). A Comparison of Regularization Techniques in Deep Neural Networks. Symmetry. 10. 648. 10.3390/sym10110648.

[5] Heilmeier, Alexander, et al. "Virtual strategy engineer: Using artificial neural networks for making race strategy decisions in circuit motorsport." Applied Sciences 10.21 (2020)

[6] Wang, Huibing, et al. "Research survey on support vector machine." 10th EAI International Conference on Mobile Multimedia Communications. 2017.

[7] Batuwita, Rukshan & Palade, Vasile. (2010). FSVM-CIL: Fuzzy support vector machines for class imbalance learning. Fuzzy Systems, IEEE Transactions on. 18. 558 - 571. 10.1109/TFUZZ.2010.2042721.

[8] Singla, Manisha, Debdas Ghosh, and K. K. Shukla. "pin-TSVM: A Robust Transductive Support Vector Machine and its Application to the Detection of COVID-19 Infected Patients." Neural Processing Letters 53.6 (2021).

[9] Niu, Yan, and Shenglan Ye. "Data Prediction Based on Support Vector Machine (SVM)—Taking Soil Quality Improvement Test Soil Organic Matter as an Example." IOP Conference Series: Earth and Environmental Science. Vol. 295. No. 2. IOP Publishing, 2019.

[10] Yu, Wei, et al. "Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes." BMC medical informatics and decision making 10.1 (2010): 1-7.

[11] Naicker, Nalindren, Timothy Adeliyi, and Jeanette Wing. "Linear support vector machines for prediction of student performance in school-based education." Mathematical Problems in Engineering 2020 (2020): 1-7.

[12] Guhathakurata S, Kundu S, Chakraborty A, Banerjee JS. A novel approach to predict COVID-19 using support vector machine. Data Science for COVID-19. 2021:351–64. doi: 10.1016/B978-0-12-824536-1.00014-9. Epub 2021 May 21. PMCID: PMC8137961.

[13] Lessmann, Stefan & Sung, Ming-Chien & Johnson, Johnnie. (2009). Identifying winners of competitive events: A SVM-based

classification model for horserace prediction. European Journal of Operational Research. 196. 569-577. 10.1016/j.ejor.2008.03.018.

[14] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428.

[15] A. Patle and D. S. Chouhan, "SVM kernel functions for classification," 2013 International Conference on Advances in Technology and Engineering (ICATE), Mumbai, India, 2013, pp. 1-9, doi: 10.1109/ICAdTE.2013.6524743.