**ANÁLISE PREDITIVA DO DESEMPENHO NA FÓRMULA 1 COM MACHINE LEARNING**

Luciano Franzoi Filho

Unicesumar – Centro Universitário Cesumar

Curso de Engenharia de Software – Disciplina: Arquitetura de Software

**Resumo**

Este artigo apresenta uma análise preditiva baseada em aprendizado de máquina sobre dados históricos da Fórmula 1, com foco em prever a pontuação dos pilotos em corridas específicas. O dataset utilizado foi disponibilizado pelo Kaggle, cobrindo o período de 1950 a 2020. A metodologia implementada inclui autenticação via API, download automatizado, análise exploratória (EDA), pré-processamento, e dois fluxos distintos de treinamento de modelos: um manual com cinco algoritmos de regressão (Gradient Boosting, Random Forest, AdaBoost, LightGBM e SVR) e outro automatizado utilizando a biblioteca PyCaret. Os modelos Gradient Boosting e Random Forest obtiveram os melhores resultados em termos de R². A análise com SHAP permitiu interpretar a importância de cada variável na previsão da pontuação. Os gráficos gerados destacaram padrões históricos relevantes, como o domínio da Mercedes na era híbrida, a consistência da Ferrari ao longo das décadas e a influência da idade dos pilotos no desempenho. A aplicação de ciência de dados neste contexto esportivo demonstra o potencial de tecnologias de machine learning para revelar tendências, apoiar decisões estratégicas e expandir a compreensão sobre a performance no automobilismo.

**Palavras-chave**: Fórmula 1; Machine Learning; Análise de Dados; Regressão; Data Science.

**1. Introdução**

A Fórmula 1 é considerada o ápice do automobilismo mundial, combinando tecnologia, engenharia de ponta, estratégia e habilidade dos pilotos. Com mais de sete décadas de história, o esporte acumula uma imensa quantidade de dados, tornando-se um campo fértil para análise com técnicas de ciência de dados.

Este artigo tem como objetivo explorar, modelar e interpretar dados históricos do Campeonato Mundial de Fórmula 1 (1950–2020) utilizando algoritmos de aprendizado de máquina. A análise visa prever a pontuação dos pilotos em corridas e identificar os principais fatores que influenciam o desempenho esportivo.

**2. Fundamentação Teórica**

O uso de aprendizado de máquina (machine learning) em análises esportivas tem crescido significativamente nos últimos anos, principalmente devido à disponibilidade de dados detalhados e ferramentas computacionais acessíveis. Em especial, algoritmos de regressão supervisionada têm sido amplamente empregados para tarefas que envolvem previsão de valores numéricos, como pontuações de jogadores, tempo de voltas, ou distância percorrida.

A regressão supervisionada parte do princípio de que existe uma relação entre um conjunto de variáveis independentes (features) e uma variável dependente (target). No contexto deste trabalho, a variável alvo é a pontuação obtida pelo piloto em uma corrida, e as variáveis independentes incluem dados como idade do piloto, equipe (construtor), posição de largada, status da corrida (abandono, finalizado, etc.), tempo da volta mais rápida e resultados de classificação.

Dentre os algoritmos usados neste projeto estão: Random Forest, Gradient Boosting, LightGBM, AdaBoost e SVR. Todos esses modelos são conhecidos por sua robustez e boa performance em tarefas de regressão, especialmente quando bem ajustados com técnicas como GridSearchCV e validação cruzada. Tais algoritmos também foram amplamente discutidos por Géron (2019), que destaca o Gradient Boosting como um dos métodos mais eficazes quando se trata de dados tabulares com múltiplas variáveis correlacionadas.

A análise também utilizou a técnica SHAP (SHapley Additive exPlanations), que é baseada em teoria dos jogos e permite quantificar o impacto de cada feature nas previsões do modelo. Isso contribui para a interpretabilidade dos resultados, um aspecto essencial em aplicações reais onde é importante compreender por que um modelo tomou determinada decisão. Lundberg e Lee (2017) demonstraram que SHAP fornece uma base matemática sólida para explicações consistentes, sendo altamente recomendada para análise de modelos de machine learning.

**3. Metodologia**

A metodologia do projeto foi organizada em etapas bem definidas, visando garantir a reprodutibilidade e clareza na execução do pipeline. Abaixo, detalho como cada etapa foi desenvolvida:

1. **Autenticação e Download**: Utilizamos a API oficial do Kaggle com um arquivo de credenciais (kaggle.json), por meio da função autenticar\_kaggle(). Com isso, o script dataset\_downloader.py realiza automaticamente o download e extração dos arquivos CSV do dataset de Fórmula 1.
2. **Tratamento e Consolidação**: A função ajustar\_datasets() relaciona tabelas principais como drivers, races, results, constructors, entre outras. Também converte tempos de volta para milissegundos e calcula a idade dos pilotos com base na data de nascimento e data da corrida.
3. **Análise Exploratória (EDA)**: Com o eda\_service.py, usamos bibliotecas como AutoViz e ydata\_profiling para entender a distribuição dos dados, identificar valores nulos e encontrar correlações relevantes.
4. **Pré-processamento**: O data\_preprocessor.py trata valores ausentes, aplica StandardScaler em variáveis numéricas e OneHotEncoder para variáveis categóricas. Utilizamos o ColumnTransformer para aplicar transformações combinadas no conjunto de dados, conforme boas práticas descritas por Pedregosa et al. (2011) no desenvolvimento da biblioteca Scikit-learn.
5. **Treinamento de Modelos Manuais**: Em model\_service.py, a função train\_model() treina cinco modelos distintos com validação cruzada e busca de hiperparâmetros via GridSearchCV. Os resultados são armazenados e comparados com base em métricas como R², MAE e RMSE.
6. **Treinamento com PyCaret**: Também integramos ao pipeline a biblioteca PyCaret, por meio da função train\_with\_pycaret(). Esta função automatiza o processo de configuração, comparação e seleção do melhor modelo de regressão, salvando-o para uso posterior. A comparação entre modelos é feita automaticamente pelo PyCaret, que retorna o modelo com melhor desempenho baseado em múltiplas métricas de avaliação.
7. **Geração de Relatórios e Visualizações**: A função gerar\_graficos\_relatorios() plota gráficos como Top Pilotos, Status de Abandono e SHAP values. Estes são salvos para análise posterior. Isso segue práticas comuns em projetos de ciência de dados, como documentado por Brownlee (2020).
8. **Execução Geral**: O Main.py conecta todos os módulos, permitindo executar todo o pipeline de ponta a ponta com um único comando. O pipeline contempla dois fluxos paralelos de modelagem: um manual (com análise SHAP) e outro automatizado via PyCaret, aumentando a robustez da avaliação e seleção do melhor modelo.

**4. Coleta e Tratamento dos Dados**

**4.1 Obtenção do Dataset**

Utilizando a API oficial do Kaggle, o dataset foi baixado e armazenado localmente. As tabelas principais incluem: races.csv, results.csv, drivers.csv, constructors.csv, circuits.csv, qualifying.csv e status.csv.

**4.2 Pré-processamento**

As seguintes transformações foram aplicadas:

* Substituição de valores nulos (\N → NaN)
* Conversão de tempos para milissegundos
* Criação de nova variável: idade do piloto
* Ajuste de pontuações com base em regras da FIA
* Normalização com StandardScaler e codificação com OneHotEncoder

As colunas mais relevantes para a modelagem foram selecionadas: posição, pontos, idade, construtor, status da corrida, tempo de volta mais rápida, entre outras.

**5. Análise Exploratória de Dados (EDA)**

Utilizando AutoViz e ydata\_profiling, foram gerados relatórios sobre:

* Distribuição de variáveis contínuas e categóricas
* Correlações entre idade e desempenho
* Frequência de abandonos por tipo de falha

**[INSERIR FIGURA 1: Status Mais Comuns em Abandonos]**

O gráfico mostra que a maior parte dos abandonos ocorre por problemas no motor, seguido por colisões e falhas na transmissão.

**[INSERIR FIGURA 2: Evolução de Pontos por Temporada (2010–2024)]**

Evidencia-se o domínio da Mercedes na era híbrida (2014–2020) e o crescimento da Red Bull.

**[INSERIR FIGURA 3: Relação entre Idade do Piloto e Posição Final]**

Existe uma leve correlação positiva entre experiência (idade) e melhor desempenho.

OU

A análise exploratória de dados (EDA) foi essencial para entender a estrutura do dataset e identificar padrões ocultos. Usando AutoViz, ydata\_profiling e D-Tale, geramos relatórios interativos e gráficos descritivos.

Análises incluíram:

* **Distribuição de Pontos**: Observamos que a maioria dos pilotos marca poucos pontos, o que é consistente com a natureza competitiva da F1.
* **Status da Corrida**: Classificamos os tipos de abandono, como falha de motor, colisão e desistência. Esses status impactam diretamente na pontuação.
* **Idade x Desempenho**: Um gráfico de dispersão revelou correlação leve entre idade e melhor colocação final, sugerindo que experiência pode ser um diferencial.
* **Top Equipes**: Ferrari, McLaren, Mercedes e Red Bull são consistentemente as mais bem pontuadas, o que valida os dados históricos.

Essas observações foram essenciais para validar a qualidade dos dados e orientar a escolha de variáveis no modelo.

**6. Modelagem Preditiva**

Cinco modelos de regressão foram treinados para prever pontuações:

1. **Gradient Boosting Regressor**
   * R²: 0.83
   * Melhor desempenho geral
2. **Random Forest Regressor**
   * R²: 0.82
   * Alta robustez e interpretabilidade
3. **AdaBoost Regressor**
   * R²: 0.74
   * Desempenho intermediário
4. **LightGBM Regressor**
   * R² competitivo, mas inferior aos anteriores
5. **Support Vector Regressor (SVR)**
   * R²: 0.02
   * Baixa capacidade de generalização neste caso

**[INSERIR FIGURA 4: Comparação de Modelos de Regressão]**

O gráfico evidencia o domínio dos modelos baseados em árvores de decisão.

Ou

Nesta etapa, aplicamos cinco modelos supervisionados de regressão, cada um com suas particularidades:

* **Gradient Boosting Regressor**: Ideal para lidar com dados heterogêneos. Utiliza uma série de árvores de decisão fracas para formar um modelo forte, corrigindo os erros das etapas anteriores. Essa abordagem foi inicialmente proposta por Friedman (2001) e se tornou um padrão-ouro para competições de dados.
* **Random Forest Regressor**: Combina múltiplas árvores de decisão de forma paralela. Mostrou-se eficaz e rápido. Originalmente descrito por Breiman (2001), é amplamente usado em benchmarks de regressão.
* **LightGBM**: Usado por sua eficiência em grandes volumes de dados. Trabalha com histogramas para acelerar o processo. Desenvolvido por Ke et al. (2017), LightGBM é uma escolha popular em competições do Kaggle.
* **AdaBoost Regressor**: Foca nas instâncias mal classificadas em cada iteração. Boa performance, mas inferior ao Random Forest. Introduzido por Freund e Schapire (1997), é um dos algoritmos de boosting mais antigos.
* **SVR (Support Vector Regressor)**: Embora promissor teoricamente, teve desempenho fraco neste caso (R² de 0.02). SVR é eficiente em espaços de alta dimensionalidade, mas sensível à escala e parametrização.

Todos os modelos foram treinados com validação cruzada k-fold (k=5) e hiperparâmetros otimizados via GridSearchCV. As métricas principais para comparação foram:

* R² (coeficiente de determinação)
* MAE (erro absoluto médio)
* RMSE (raiz do erro quadrático médio)

As melhores performances foram obtidas com Gradient Boosting e Random Forest, ambos com R² superior a 0.82.

**7. Interpretação dos Resultados**

Utilizando SHAP Values, foi possível analisar a influência de cada variável nas previsões do modelo.

**[INSERIR FIGURA 5: Gráfico SHAP das variáveis mais impactantes]**

Fatores como posição de largada, construtor e tempo de volta rápida foram os mais relevantes.

Além disso, outros gráficos complementares destacam:

* **[INSERIR FIGURA 6: Top 20 Pilotos com Mais Pontos]**
* **[INSERIR FIGURA 7: Top 10 Equipes com Mais Pontos]**
* **[INSERIR FIGURA 8: Top 10 Pilotos com Mais Pontos]**

Lewis Hamilton, Michael Schumacher e Sebastian Vettel lideram os rankings. Ferrari, McLaren e Mercedes dominam entre as equipes.

**8. Conclusão**

Este estudo demonstrou a eficiência de modelos de aprendizado de máquina aplicados à análise esportiva, especialmente no contexto da Fórmula 1. Ao longo da pesquisa, foi possível extrair insights relevantes a partir de dados históricos, entender os principais fatores que influenciam o desempenho dos pilotos e prever pontuações com alto grau de acurácia.

A análise revelou que variáveis como equipe, posição de largada, idade do piloto e status final da corrida têm impacto direto na pontuação. Gráficos como SHAP Values permitiram uma visão clara da contribuição de cada feature.

Recomenda-se, como trabalho futuro, a incorporação de variáveis contextuais como condições climáticas, estratégias de pit-stop e tipo de circuito (rua ou fechado). Também é possível explorar redes neurais recorrentes para prever desempenho ao longo de múltiplas corridas sequenciais. Além disso, explorar dados adicionais de sensores em tempo real, como telemetria dos carros, pode trazer avanços significativos nas previsões.

Por fim, este projeto contribui para o avanço do uso de inteligência artificial no esporte, fornecendo um modelo replicável que pode ser aplicado a outras competições ou modalidades esportivas.

**Referências**

* Bunker, R., & Thabtah, F. (2019). A machine learning framework for sport result prediction. *Applied Computing and Informatics*, 15(1), 27-33.
* Towards Data Science. (2020). Formula 1 Race Predictor. https://medium.com/towards-data-science/formula-1-race-predictor-5d4bfae887da
* Brownlee, J. (2020). *Machine Learning Mastery With Python*. Machine Learning Mastery.
* Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
* Lundberg, S.M., & Lee, S.I. (2017). A unified approach to interpreting model predictions. *NeurIPS*.
* Ke, G. et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. *NeurIPS*.
* Rao, R. (2020). Formula 1 World Championship 1950–2020. *Kaggle Dataset*.
* Grandi, T. (2020). Formula 1 Winner Prediction. *GitHub*.
* Nigro, V. (2020). Formula 1 Analysis with Python. *Medium*.
* Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32.
* Friedman, J.H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*.
* Freund, Y., & Schapire, R.E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*.
* Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
* CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. p. 785-794.
* FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, v. 55, n. 1, p. 119-139, 1997.
* LI, Y.; WANG, X. LightGBM: A highly efficient gradient boosting decision tree. Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017. p. 3146-3154.
* MDPI. A review of gradient boosting methods for predictive modeling. Diagnostics. Disponível em: <https://www.mdpi.com/2075-4418/11/9/1714>. Acesso em: 30 mar. 2025.
* RAO, R. Formula 1 World Championship 1950-2020. Disponível em: <https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020>. Acesso em: 30 mar. 2025.
* SCIENCEDIRECT. A survey of gradient boosting models in predictive analytics. Journal of Machine Learning Research. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0020025521002875>. Acesso em: 30 mar. 2025.
* SPRINGER. A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review. Disponível em: <https://link.springer.com/article/10.1007/s10462-020-09896-5>. Acesso em: 30 mar. 2025.