

Preidykcja Czasów Okrążenia w Formule 1

Autorzy: Milena Oberzig, Tobiasz Adamczyk



Cel projektu i definicja problemu

- Cel: Stworzenie systemu predykcyjnego estymującego czasy okrążeń w wyścigach Formuły 1 na podstawie telemetrii, wydarzeń na torze i stanu opon
- Zmienna docelowa: Przewidywanie czasu okrążenia byłoby zbyt skomplikowane z powodów różnic między torami (wahania od 70s do 110s), zamiast tego przewidujemy stosunek do najlepszego czasu z kwalifikacji (w miarę stałe pomiędzy torami)
- Potencjalne trudności: błędy kierowców, różne strategie, do wielu danych nie mamy dostępu



Zbiór danych

- Dane zostały pobrane za pomocą biblioteki FastF1, obejmują one 23 wyścigi z sezonu 2025 (wyścig w Las Vegas odrzucony), łącznie 24 226 okrążeń
- Po usunięciu okrążeń, w których czasy nie zostały zdefiniowane, podczas których była wywieszona żółta flaga (mocno zaburza czas okrążenia) oraz pierwszego okrążenia w każdym wyścigu (wolniejsze z powodu startu z miejsca) pozostało 21 541 okrążen



EDA

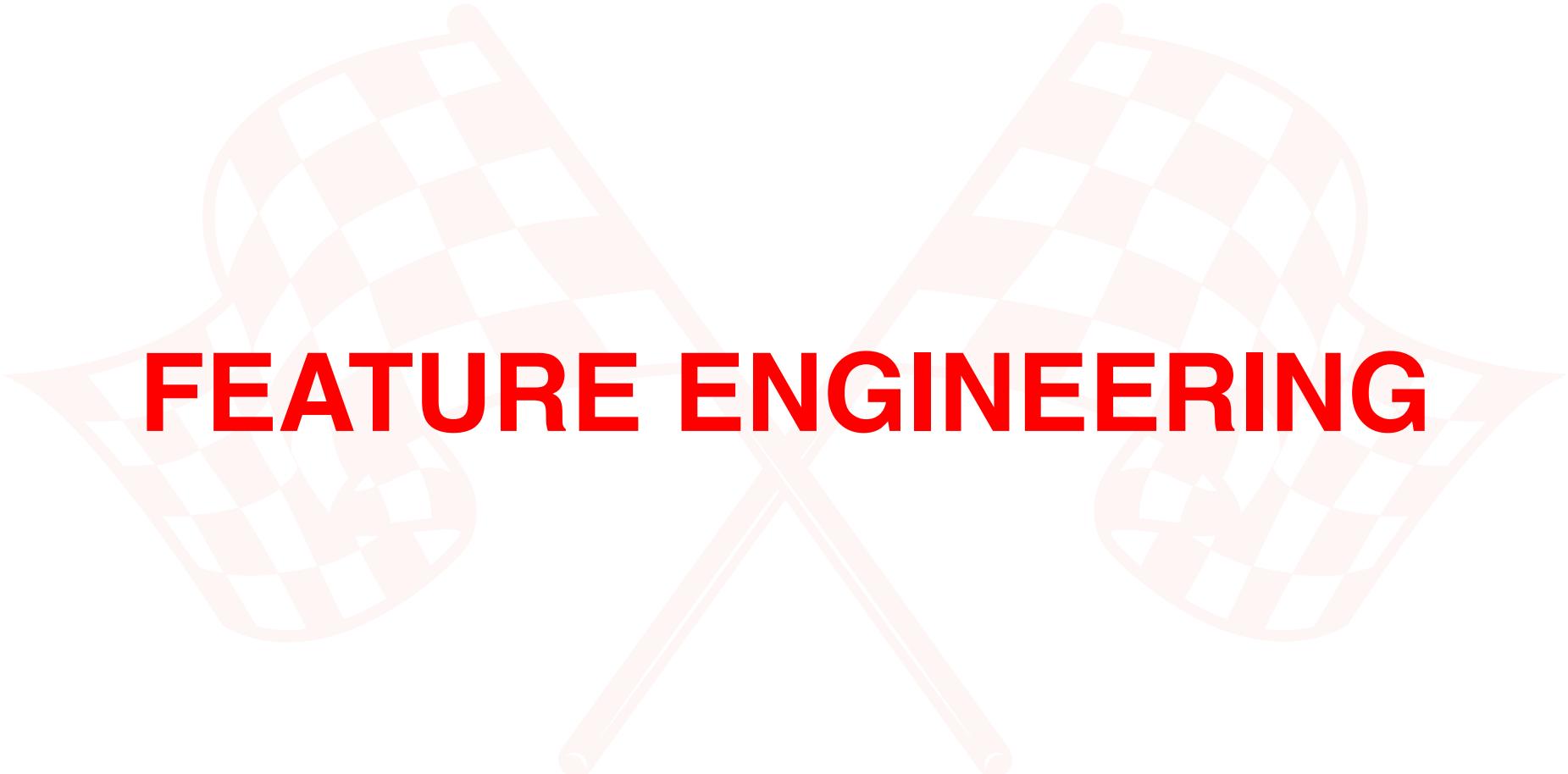
- Są duże różnica pomiędzy wyścigami, więc porównywanie wprost nie ma sensu - zamieniamy target na laptimes/qualibest żeby mieć skalę toru.
- Analiza percentylów i ucięcie outlierów
- Sprawdzenie korelacji zmiennych: widzimy że zależności mogą różnić się między torami, więc sprawdzamy też korelacje w obrębie rundy i medianę i średnią po rundach.
- Baseline (średnia): MAE 2.7 s
 - Obserwacje:
 - Im później w wyścigu, tym szybsze są okrążenia - Fuellevel i target mocno zależne
 - SpeedSt i Windspeed nie mają stabilnego wpływu pomiędzy rundami mimo że globalnie mają mocną zależność z targetem



Metodologia ewaluacji

- Podczas feature engineeringu istotność cech sprawdzana jest na testowym zbiorze danych zawierającym 4 ostatnie wyścigi
- Finalna ewaluacja przeprowadzana metodą Leave One Group Out, trenujemy model na wszystkich wyścigach poza jednym i testujemy na nim. Robimy tak dla każdego wyścigu a finalny wynik to średnia błędu z każdego okrążenia
- W treningu modelu minimalizujemy Mean Squared Error, aby karać model za większe błędy
- Do oceny modelu używamy metryki Mean Absolute Error wyrażonej w sekundach, która jest najłatwiejsza do interpretacji i pokazania skali błędu





FEATURE ENGINEERING

Kodowanie zmiennych kategorycznych

- Dodawanie osobnej kolumny dla każdego kierowcy i zespołu stworzyłoby łącznie 27 nowych kolumn, zamiast tego zamieniamy te wartości na reprezentujące rzeczywisty potencjał kierowców i zespołów
- Dla każdego kierowcy tworzymy kolumnę DriverPower. Jest to mediana targetu z danych treningowych. To samo robimy z zespołami tworząc kolumnę TeamPace
- Po tych zmianach MAE wynosi już tylko 1.03s



Dodanie charakterystyk torów

- Każdy tor ma inną charakterystykę asfaltu i długość okrążenia (co oznacza że zużycie opon po 20 okrążeniach jest inne na różnych torach)
- Dodanie danych o długości nitki, cechach asfaltu i charakterystyce zakrętów pozwala modelowi bardziej zrozumieć fizykę stojącą za czasami okrążeń
- Dodane cechy: długość toru, trakcja, szorstkość asaltu, ewolucja toru, obciążenie opon, siły boczne w zakręcie, docisk, twardość mieszanki (C1 – C5)
- MAE po tych zmianach wyniosło 0.84s



Informacje o pozycji względem innych bolidów

Położenie bolidu na torze względem innych wskazuje na dużo rzeczy. Bolid mniej niż sekundę przed kierowcą daje mu DRS oraz tunel aerodynamiczny. Położenie blisko innego bolidu może oznaczać, że kierowcy będą próbować się wyprzedzać (co znacznie wpływa na czas). Czas mierzony jest na początku każdego okrążenia, dla pierwszego i ostatniego kierowcy wartości GapAhead i GapBehind wynoszą po 100s (bo tor przed lub za nimi jest pusty)



Dodanie telemetrii

- Dane telemetryczne z biblioteki FastF1 są mierzone z częstotliwością 9 razy na sekundę. Przeciętny wyścig trwa 1.5h, więc $90 \times 60 \times 23 \times 20 \times 9$ to około 22 mln wierszy
- Najpierw dane agregujemy wyciągając następujące kolumny: Średnia prędkość w zakrętach, maksymalna prędkość, średnie obroty silnika, zmiany biegów, średni czas z gazem w podłodze, średni czas z naciśniętym hamulcem
- Te zmienne przekładają się na kilka rzeczy: walka z innym kierowcą, mapowanie silnika etc.
- Problem: Awarie czujników
- Rozwiązanie: IterativeImputer (albo i nie ale o tym zaraz)
- MAE po dodaniu telemetrii: 0.74s



Ewaluacja modeli

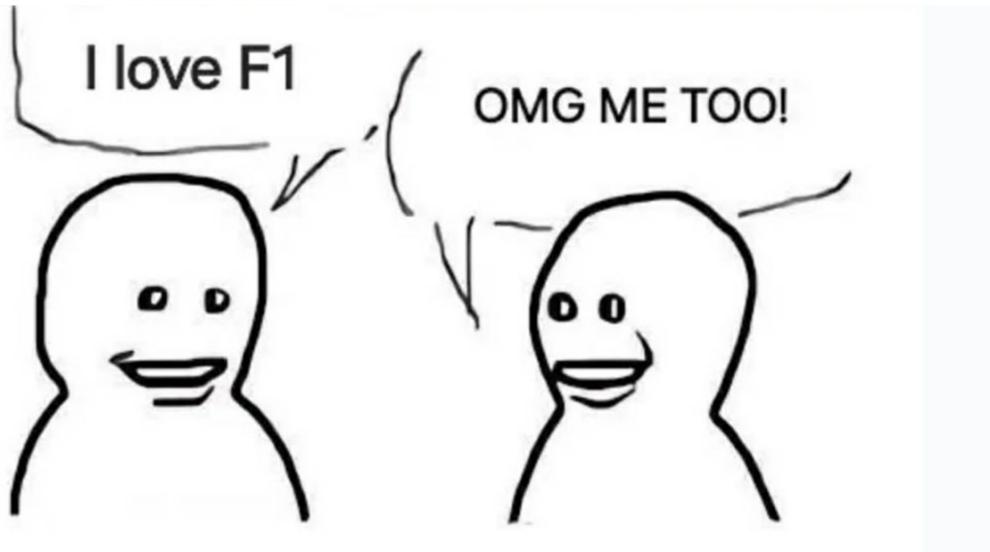
- Używamy Leave-One-Group-Out - dla każdej rundy trenujemy na wszystkich rundach poza nią, testujemy na tej która pozostała
- Sprawdzamy: RandomForestRegressor, XGBRegressor, GradientBoostingRegressor
- Używamy GridSearchCV do dobrania parametrów
- Wyniki:

RandomForest: CV MAE (s) = 1.577 s, params = {'model__max_depth': None, 'model__n_estimators': 100}

XGB: MAE (s) = 1.301 s, params = {'model__learning_rate': 0.05, 'model__max_depth': 3, 'model__n_estimators': 100}

GradientBoostingRegressor: MAE (s) = 1.254 s, params = {'model__learning_rate': 0.05, 'model__max_depth': 3, 'model__n_estimators': 100}

- Najlepszy był GradientBoostingRegressor z MAE = 1.3 , w porównaniu do baseline MAE= 2.1



$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$



Dziękujemy za
uwagę!