Let’s Predict SpeedRun Times!

ML Project by
Kimberly M.

What is a SpeedRun?

Definitions and Terms

SPEEDRUN.COM

- Speedrunning is the art of completing video games as quickly as possible. Players, known as speedrunners, master every aspect of a game, from its mechanics to its glitches.
- SpeedRuns are commonly submitted to speedruns.com and can have multiple categories like Any% (complete the game at all cost), glitchless (complete the game with no glitches), and more
- WR (World Records) are the number one and top score of a category

The Benefits

- Gamers breaking the game can uncover unintentional bugs and errors in code
- Edge cases in physics engines, collision detection, and level design, providing developers with critical insights for patches and future designs.
- Brings more awareness to indie (independent) games

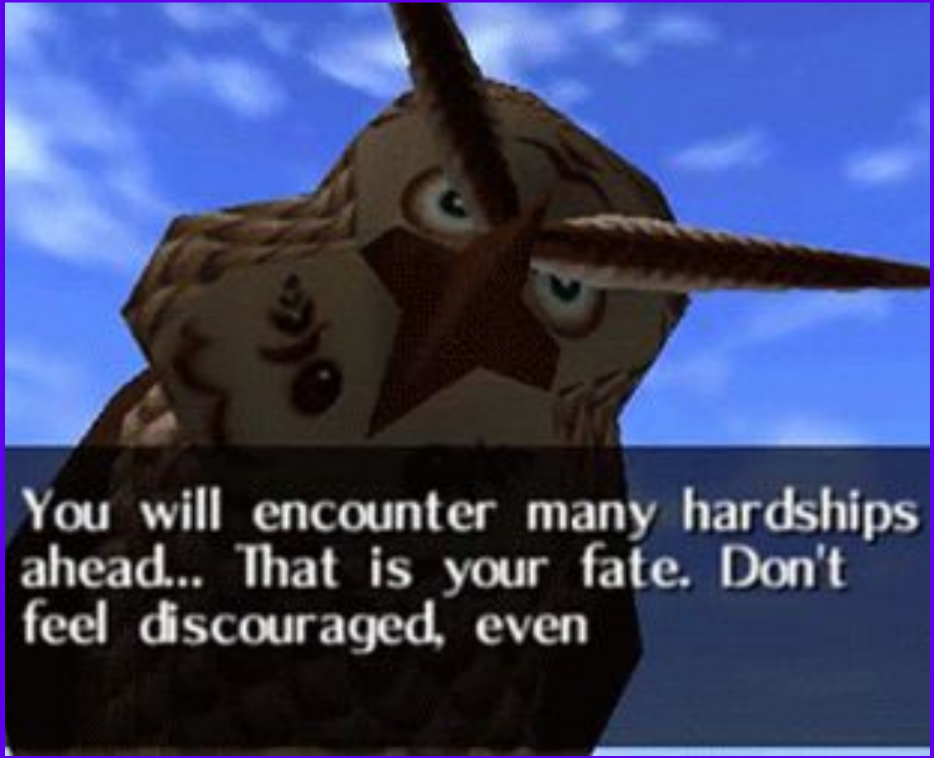
What Problem Do We Solve?

Complexity!

Speedrunning data is complex. Variables range from game type and mechanics to player engagement. Our project tackles this complexity by:

- Predicting world record speedrun times based on game and category features.
- Providing actionable insights for developers and players by identifying key factors that influence performance.

What is the ML approach used?



Finding a Dataset

- Speedruns.com has an API with all their website data but documentation is not updated
- Two Kaggle datasets: alexmerren1 and Matheus Turatti
- Picked the Turatti dataset since the other dataset contained more game metadata than the scores
- Combined the two csv files that were game and score data using the game_id into a singular dataset for the rest of the project

Dataset Overview

- Game Data: Includes metadata like Game_Id, Genres, Platforms, Total_Runs, and Release_Date.
- Category Data: Focuses on speedrun records, including Time_0 (our target), Num_Runs, Players_0, and Record_Date_0.

Exploratory Data Analysis (EDA)

- Missing Data:
  - A heatmap revealed sparse missing values in features like Players_0 and Record_Date_0.
  - Rows with critical missing values were dropped to maintain data integrity.
- Skewness Detection:
  - Features like Total_Runs and the target variable Time_0 showed high skewness.
  - Log transformations normalized these distributions, improving model reliability.
- Feature Relationships:
  - A correlation matrix highlighted key relationships, such as the strong correlation between Num_Runs and Time_0, showing how player engagement affects records.

Feature Engineering

- Taking out most of the game metadata like players and year of release
- Ended up keeping a lot of columns due to EDA and also this dataset not having many features to begin with
- **Categorical Encoding:**
  - Variables like Genres and Platforms were one-hot encoded to make them usable in regression models.
- **Sampling:**
  - To optimize computation, we used 10% of the dataset for initial model training and evaluation.
- **Train-Test Split:**
  - Data was split 80-20 to ensure robust training and testing setups.

Machine Learning Models

- **Linear Regression:**
  - This simple baseline model achieved an RMSE of ~15.2, providing a starting point.
- **Random Forest:**
  - Tuned using GridSearchCV, it achieved an RMSE of ~12.4, outperforming other models.
  - Random Forest's ability to capture non-linear relationships and rank feature importance made it invaluable.
- **Gradient Boosting:**
  - While slightly less accurate than Random Forest, it performed robustly with a mean RMSE of ~13.1 in cross-validation.

Let's look at the results and Demo!

Thank You!