2025

Intro to ML

Exploration in Pricing of Playstation Games

I spent way too much time on this.

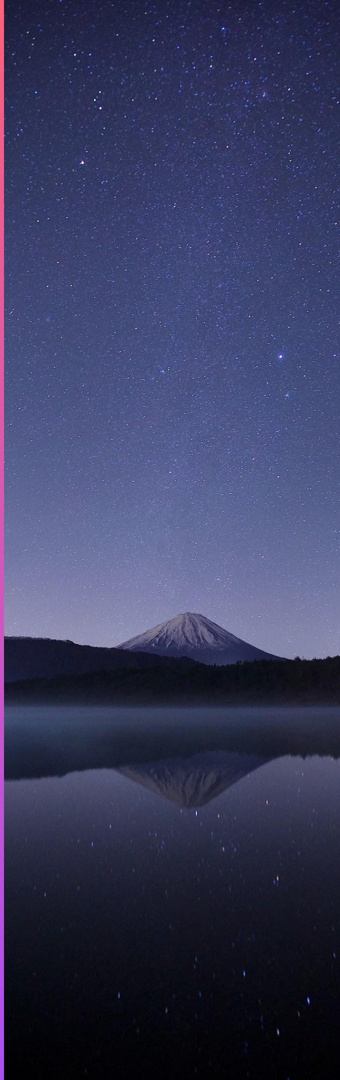
THE PROBLEM

Can we predict the price of Playstation games?

ML Approach

Random Forest Regressor +
Grid Search CV +
Hours Crying over EDA

Problem Solved



THE VISION

Jordan - Regretful Data Scientist

"

This was supposed to be a short and fun project for this final project. It was not. I only have myself to blame - the random forest regressor is actually really easy to use. 10/10 would use again.

EXPLORATORY DATA ANALYSIS

Column Descriptions

`game_name` Title of the game.

`highest_price` The highest recorded price (from PlayStation Store, in EUR).

`release_date` Release date of the game on the specified PlayStation platform.

`genre` Primary and secondary genres (e.g., Action / Adventure).

`publisher` Publishing company or studio responsible for release.

`platform` PlayStation platform (PS3, PS4, PS5).

`metacritic_score` Average critic score (0–100) from Metacritic.

`metacritic_rating_count` Number of critic reviews counted on Metacritic.

`metacritic_user_score` Average user score (0–10) from Metacritic.

`metacritic_user_rating_count` Number of user ratings counted on Metacritic.

`playstation_score` PlayStation Store user score (0–5).

`playstation_rating_count` Total number of PlayStation Store user ratings.

TARGET TO PREDICT

EXPLORATORY DATA ANALYSIS

Column Descriptions

`game_name` Title of the game.

`highest_price` The highest recorded price (from PlayStation Store, in EUR).

`release_date` Release date of the game on the specified PlayStation platform.

`genre` Primary and secondary genres (e.g., Action / Adventure).

`publisher` Publishing company or studio responsible for release.

`platform` PlayStation platform (PS3, PS4, PS5).

`metacritic_score` Average critic score (0–100) from Metacritic.

`metacritic_rating_count` Number of critic reviews counted on Metacritic.

`metacritic_user_score` Average user score (0–10) from Metacritic.

`metacritic_user_rating_count` Number of user ratings counted on Metacritic.

`playstation_score` PlayStation Store user score (0–5).

`playstation_rating_count` Total number of PlayStation Store user ratings.

TARGET TO PREDICT

RIDDLED WITH NULL VALUES

EXPLORATORY DATA ANALYSIS

Column Descriptions

`game_name` Title of the game.

`highest_price` The highest recorded price (from PlayStation Store, in EUR).

`release_date` Release date of the game on the specified PlayStation platform.

`genre` Primary and secondary genres (e.g., Action / Adventure).

`publisher` Publishing company or studio responsible for release.

`platform` PlayStation platform (PS3, PS4, PS5).

`metacritic_score` Average critic score (0–100) from Metacritic.

`metacritic_rating_count` Number of critic reviews counted on Metacritic.

`metacritic_user_score` Average user score (0–10) from Metacritic.

`metacritic_user_rating_count` Number of user ratings counted on Metacritic.

`playstation_score` PlayStation Store user score (0–5).

`playstation_rating_count` Total number of PlayStation Store user ratings.

TARGET TO PREDICT

USEFUL WHEN TRANSFORMED

FEATURE ENGINEERING

RELEASE_DATE

- → Release date of the game
- → Helps to control for costs increasing over time
- → Transformed from datetime to a single year value
- → Ranges from ~2008-2025

GENRE

- → Genre of the game (action, adventure, puzzle, etc.)
- → Helps to predict costs that may be associated with trending game genres (ex: action/adventure)
- → Transformed into boolean columns - one per genre
- → This broke me a little bit, not going to lie.

PUBLISHER

- → Publishing company of the game
- → This single field was engineered into four separate fields:
  - Large publisher
  - Non-large publisher
  - Indie
  - Triple A (AAA)

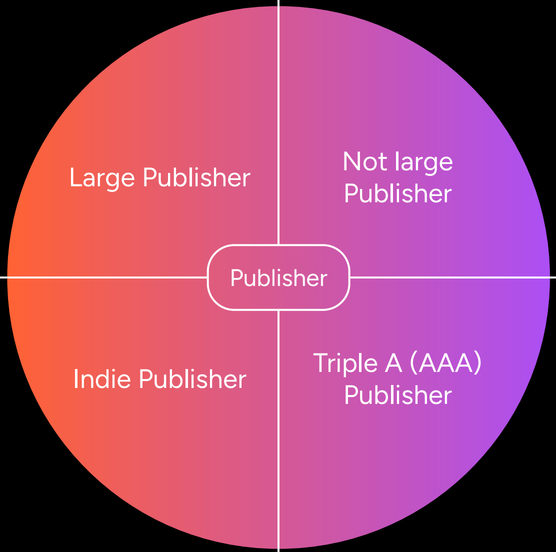
PLATFORM

- → Playstation platform that a game was released on (PS4, PS5, PS4 AND PS5)
- → Helps to predict costs that may be related to the popularity/lack thereof of a platform
- → Was way easier to implement than genre or publisher.

PUBLISHER feature engineering

- → Has published more than 25 games to Playstation platforms for the duration of dataset
- → Proxy to evaluate if the publisher is a large company, therefore having the resources to produce a higher costing game

- → Has published fewer than 25 games to Playstation platforms for the duration of the dataset
- → Encompases all publishers that do not fall into the “large publisher” category



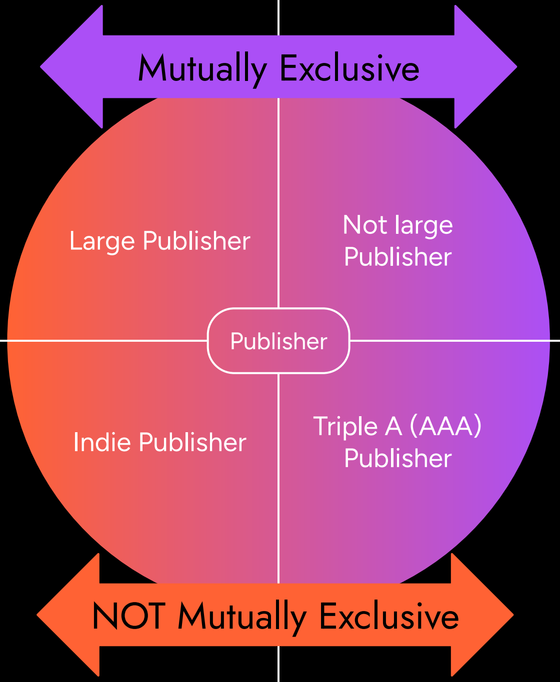
- → A company that has been identified to support indie game development
- → Indie games are typically created by smaller teams with fewer resources for video game production

- → A company that has been identified as a triple A (AAA) developer/publisher
- → These companies typically have extensive resources for video game production

PUBLISHER feature engineering

- → Has published more than 25 games to Playstation platforms for the duration of dataset
- → Proxy to evaluate if the publisher is a large company, therefore having the resources to produce a higher costing game

- → Has published fewer than 25 games to Playstation platforms for the duration of the dataset
- → Encompases all publishers that do not fall into the "large publisher" category



- → A company that has been identified to support indie game development
- → Indie games are typically created by smaller teams with fewer resources for video game production

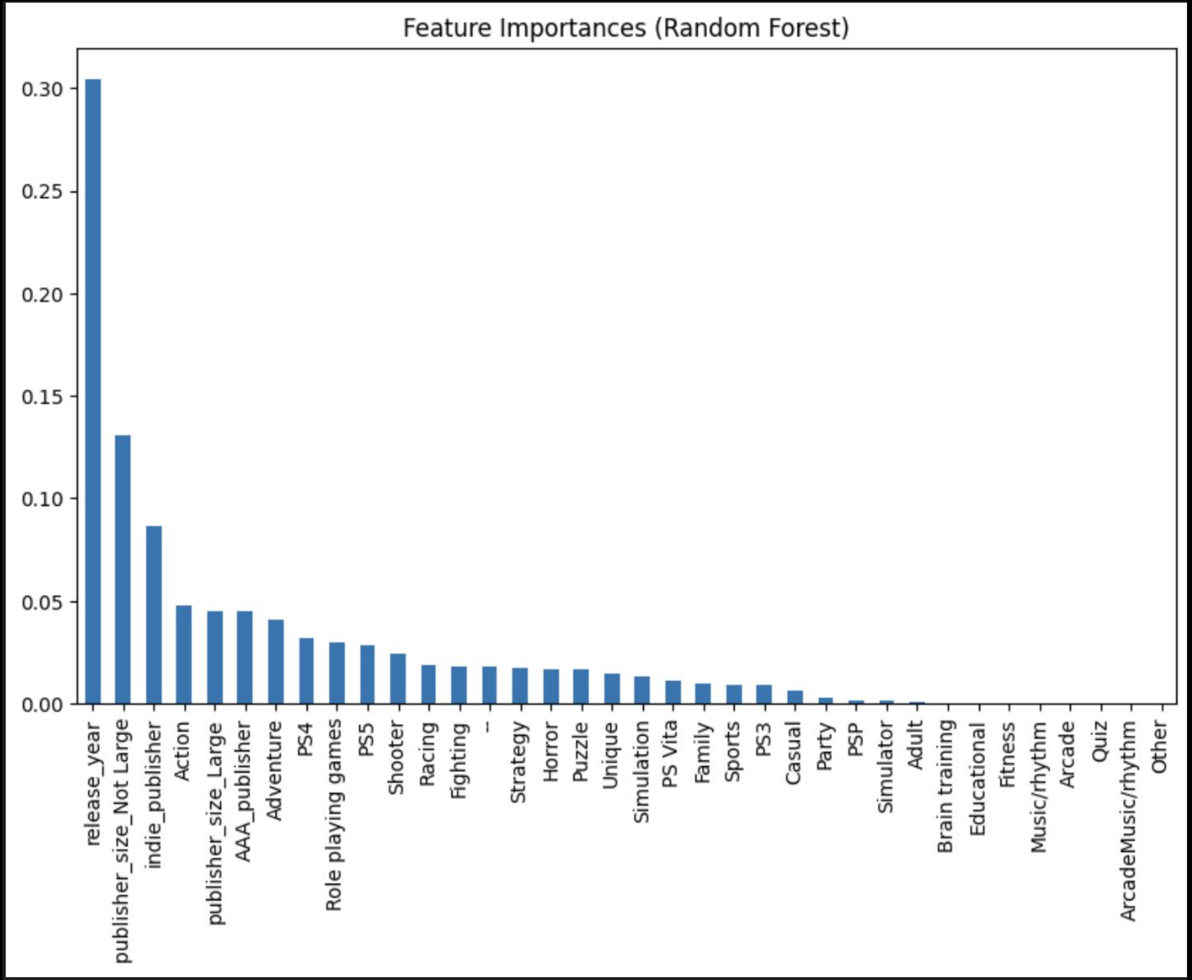
- → A company that has been identified as a triple A (AAA) developer/publisher
- → These companies typically have extensive resources for video game production

Feature Importance

using empty RandomForestRegressor

Top Features

- Release_year
- Not Large Publisher
- Indie Publisher
- Action genre
- Large Publisher
- AAA_Publisher



Feature Importance

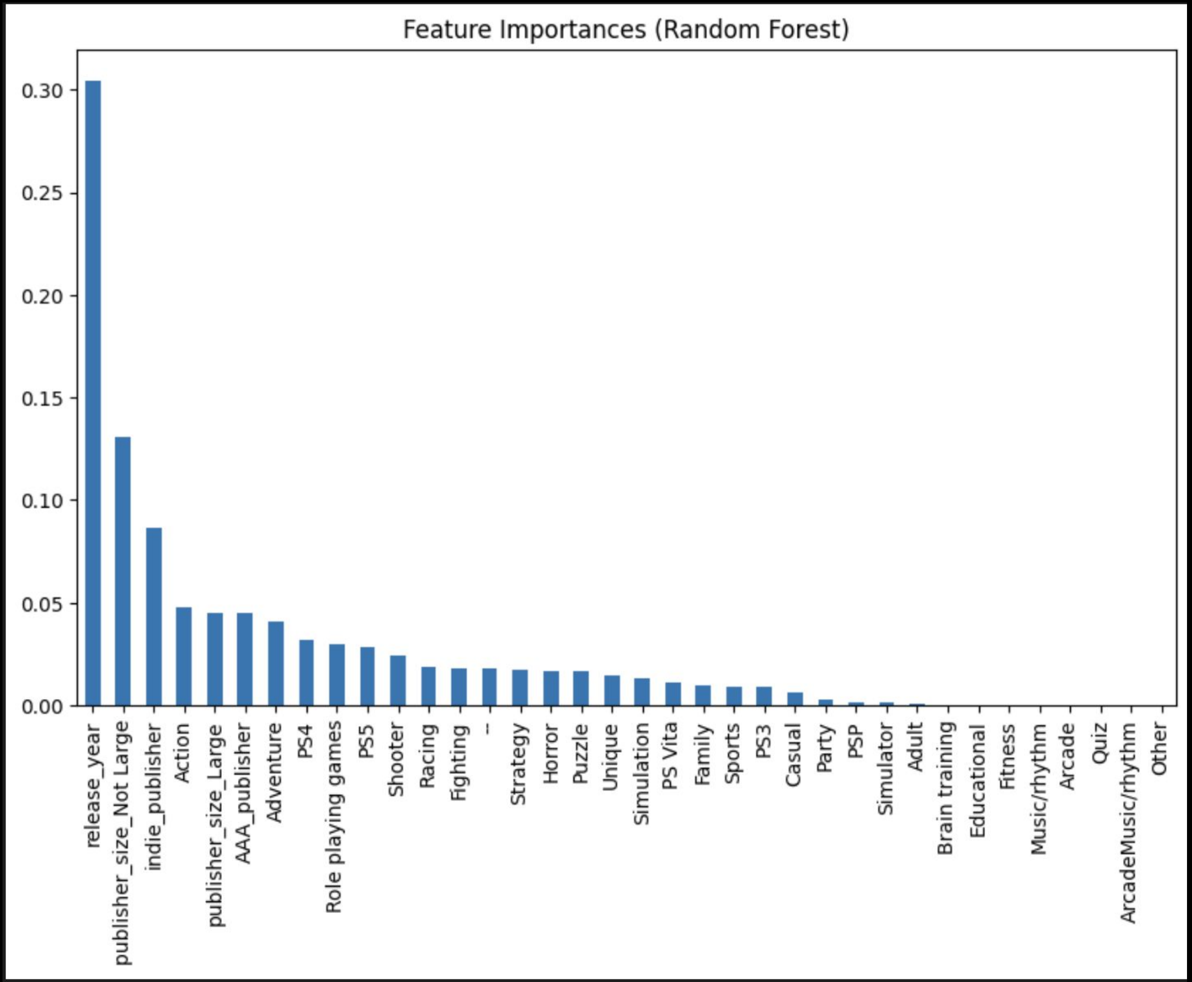
using empty RandomForestRegressor

Top Features

- Release_year
- Not Large Publisher
- Indie Publisher
- Action genre
- Large Publisher
- AAA_Publisher

Takeaways

- Release_year = super important
- Publisher type is a strong predictor of game cost
- Certain genres are stronger predictors than others



Baseline model findings

Only can go up from here - I hope!

```
from sklearn.metrics import mean_absolute_error, r2_score

y_pred = rf.predict(X_test_features)
mae = mean_absolute_error(y_test, y_pred)
print("MAE:", round(mae,2))

r2 = r2_score(y_test, y_pred)
print("R²:", round(r2,2))
```

✓ 0.0s

```
MAE: 10.38
R²: 0.31
```

Baselines

10.38

Mean Absolute Error

0.31

R-Squared

Outlier Analysis

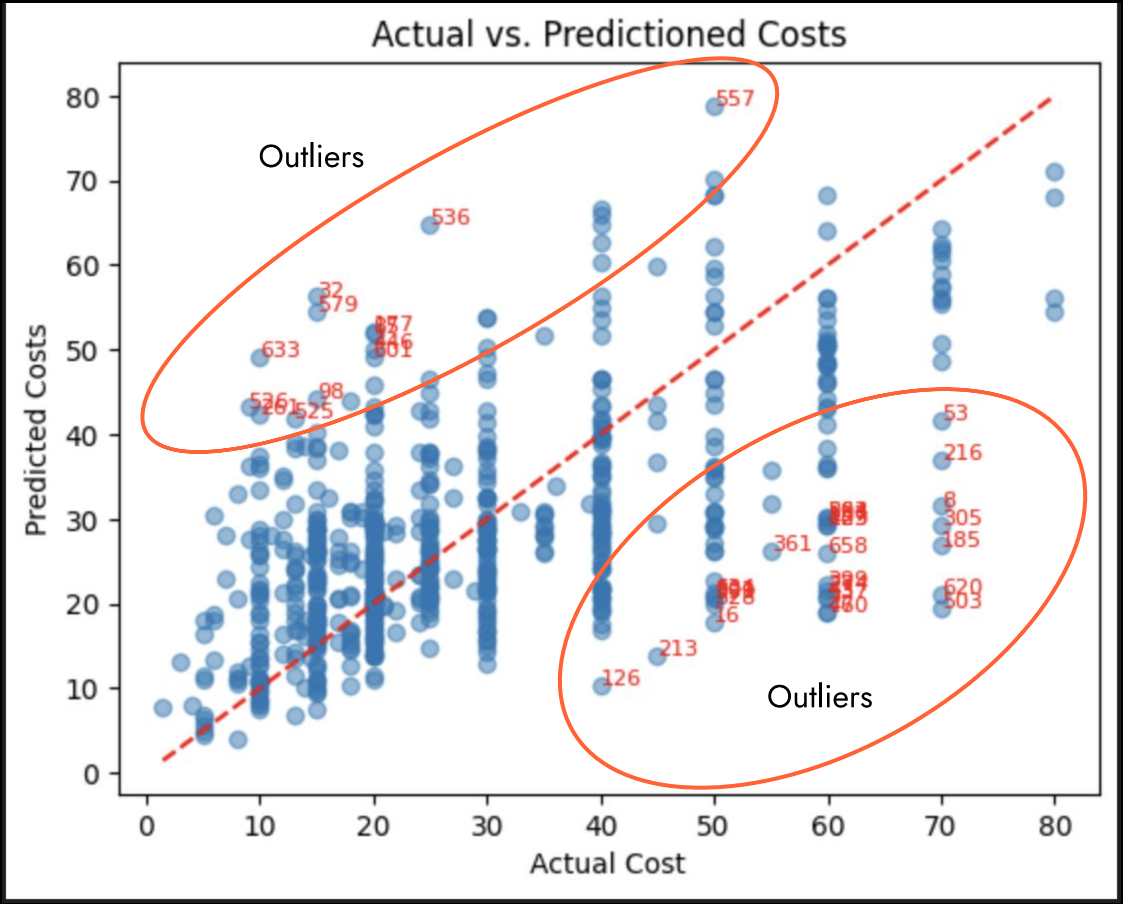
using empty RandomForestRegressor

Reasons for outliers

- Highest price value was invalid (ex: created in error)
- Demos of games were priced at 0
- Games have been re-published or ported to newer platforms (ex: Grand Theft Auto/Skyrim have been re-published and have high costs)

Bottom Line

- Invalid records (demos) were removed
- Valid records remain



Hyperparameter Tuning

Please let this work

Identified Parameters

20

Max Depth

log2

Max Features

1

Min Samples Leaf

5

Min Samples Split

250

N Estimators

```python
# Start to tune hyperparameters
# Use GridSearchCV to evaluate params
param_grid = {
    'n_estimators': [10,50,100,250,500],
    'min_samples_split': [2,5,10,20,50],
    'max_depth': [None,1,5,10,20],
    'max_features': ["sqrt","log2",None],
    'min_samples_leaf': [1,5,10,20]
}

pretuned_rf = RandomForestRegressor()

grid = GridSearchCV(pretuned_rf, param_grid, cv=3, scoring='r2')
grid.fit(X_train, y_train)

print("Best params: ",grid.best_params_)
print("Best accuracy: ",grid.best_score_)
```

Tuned Model

using RandomForestRegressor

Performance

- MAE of 9.91
- R-Squared of 0.401

Bottom Line

- Not great, but does show evidence of predictive power

```python
tuned_rf = RandomForestRegressor(
    n_estimators=250,
    max_depth=20,
    min_samples_split=5,
    max_features='log2',
    min_samples_leaf=1
)
tuned_rf.fit(X_train_features, y_train)

y_pred = tuned_rf.predict(X_test_features)

mae = mean_absolute_error(y_test, y_pred)
print("MAE:", round(mae,2))

r2 = r2_score(y_test, y_pred)
print("R²:", round(r2,3))
```

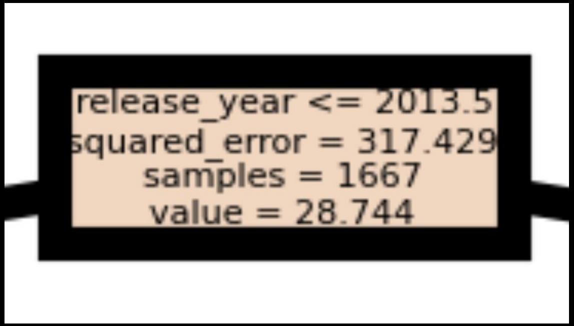
70] ✓ 0.2s

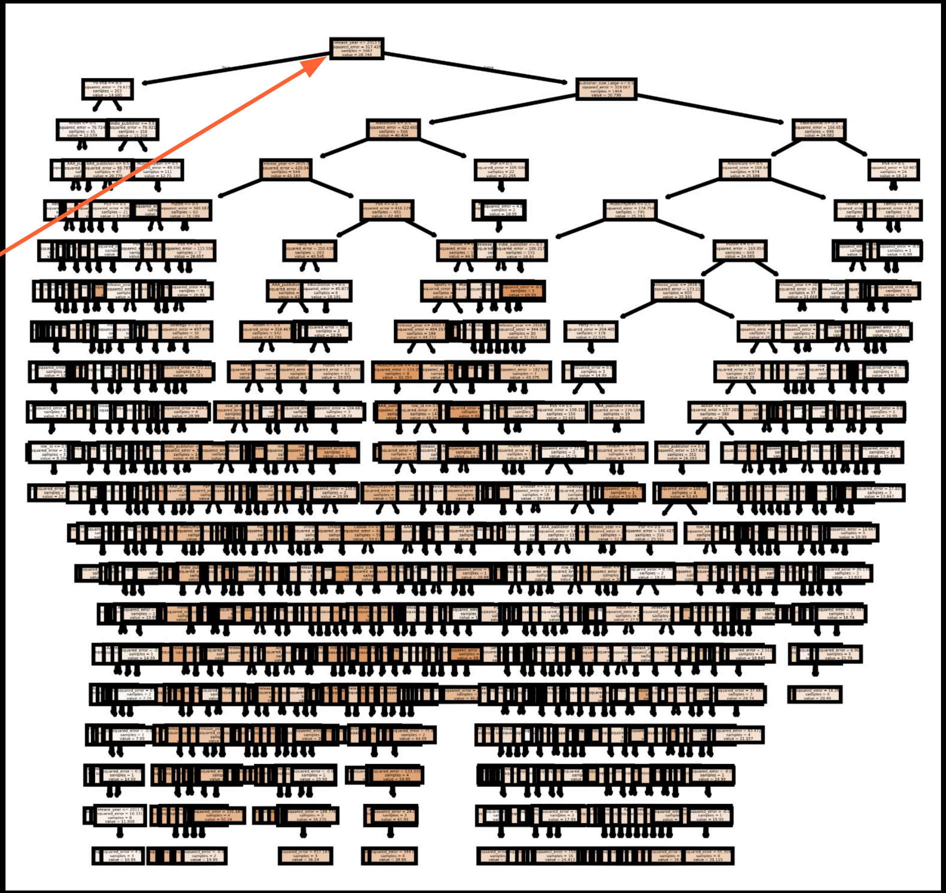
```
MAE: 9.91
R²: 0.401
```

Example Tree

from RandomForestRegressor

Root Node





Opportunities for future improvement

Adjust prices for inflation

This may decrease dependence on release_year and would allow for other features potentially to show more predictive power

Feature engineer genre

Certain genres were found to have predictive power while others did not. Doing additional feature engineering might help to elevate differences in cost by genre.

Capturing game re-releases

As discussed, games can be re-released on new platforms. This creates inflated costs on historical games. Finding a way to capture games that have been re-released would control for some of that unexplained variability.

Including game developer feature

The dataset used for this project included the publisher of a game, but it did not include the company that developed said game. Including this element could help to further parse out differences in costs that may be driven by a particular game development studio.

THANK YOU

No questions at this time. I’m tired.

https://github.com/PoofyOddish/intro-to-ml-final-project