VictorVis

An Exploration of
eSports Statistics

Preparation Phase

What is VictorVis?

VictorVis is a project utilizing Machine Learning through Supervised data. With this data we want to determine how well we can predict a player's rating as our hypothesis is that higher rated players will help predict winners.

Hypothesis

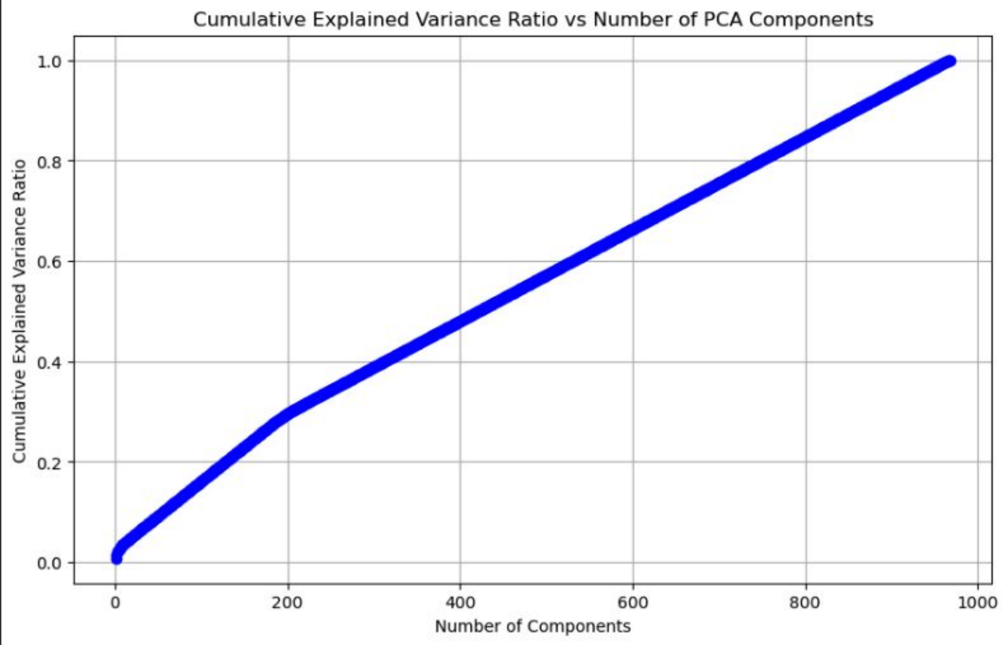
'We believe through the utilization of our large dataset, that we would successfully be able to predict whether or not a solo player can work well within a team for Counter-Strike competitions based off of their rating/predicted rating.'

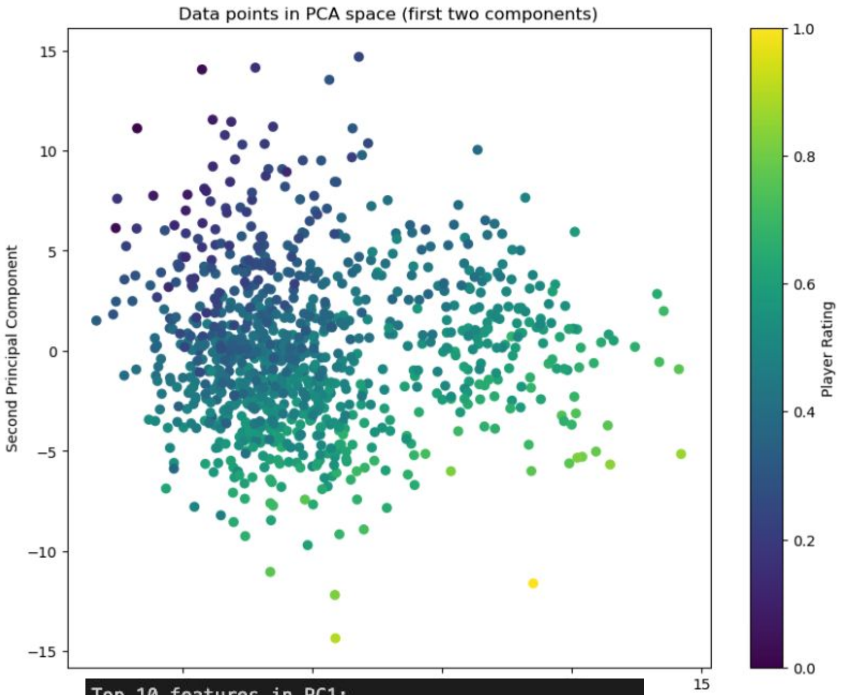
The Dataset

Writing out a script, we accessed data held within HLTV.org player stats pages - a website that records the statistics of solo and team players in the environment of Counter-Strike.

Splicer Results

A total of 967+ players and their stats were gathered into one CSV file/Dataframe. Each player had 72 data points gathered.





Explored Data

Through exploration of the data, we were able to determine that stats like kd_ratio,kills_per_round, and kast provided the best statistics to make rating predictions rather than utilizing all 75+ columns.

```
Top 10 features in PC1:
sniping_rounds_with_kills_percentage    0.206995
sniping_kills_per_round                 0.205665
sniping_multi_kill_rounds               0.205606
sniping_score                           0.204642
sniping_kills_percentage                0.204071
opening_success                         0.202675
sniping_opening_kills_per_round         0.200080
kd_ratio                                0.191628
dpr                                     0.174935
deaths_per_round                        0.174935
```

Feature Selection

Using various methods in feature selection, we were able to determine the best correlative features that interact with the rating to create robust models to predict player rating.

Mutual Information Scores:

kd_ratio	0.874832
kpr	0.740670
kills_per_round	0.711247
firepower_score	0.543491
firepower_rounds_with_kill	0.536227

- **Columns Kept:** We retain several columns for model training based on their relevance which was discoverd through our ExploreData notebook.
- **Action:** Print the shape of the DataFrame after selecting the required features to verify the subset size.

```python
# List of columns selected via feature selection
columns_to_keep = [
    # Numeric features
    'kd_ratio', 'firepower_damage_per_round_win', 'kills_per_round',
    'firepower_score', 'impact', 'trading_damage_per_kill', 'kast',
    'entrying_support_rounds', 'utility_time_opponent_flashed_per_round',

    # Categorical features
    'team',

    # Target variable
    'rating'
]

# Keep selected columns plus player_name and real_name (for reference)
df = df[columns_to_keep + ['player_name', 'real_name']]
print(f"DataFrame shape after selecting columns: {df.shape}")
```

Python

Model Phase

Model Results- LinearRegression

- Features utilized here are determined in ExploreData.ipynb
- kd_ratio
- kpr
- firepower_score
- firepower_rounds_with_kill
- impact

Features Used for the Model:

y:['rating']
X:~~dataframename~~.drop(columns= 'rating')

The used dataframes: team_players_featured and solo_players_featured

Linear Regression Model: Shayne (linearregression.ipynb)

LinearRegression Cont.

According to the Linear Regression results. . .

The MSE for both the team players data and solo players data are relatively low.

The r2 data is also notable high.

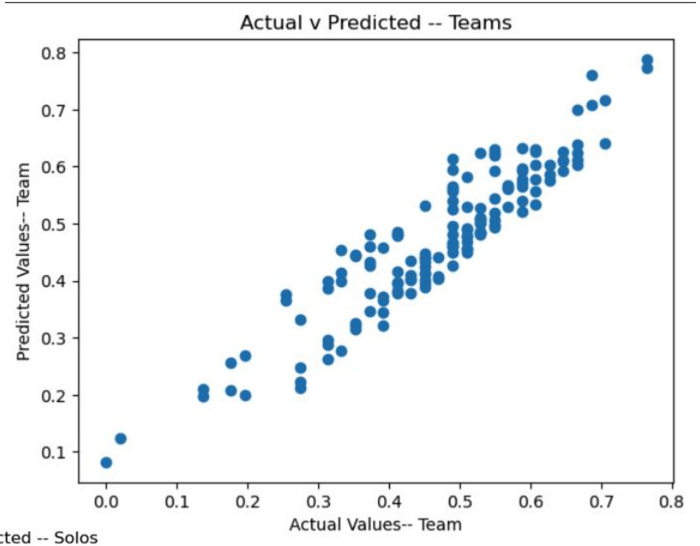
Both factors determine Linear Regression ***could*** be a good model to use.

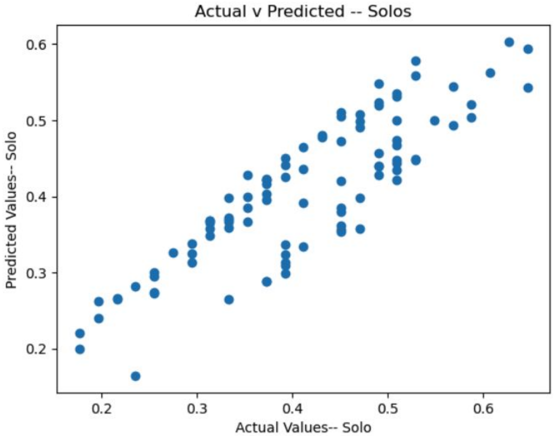
```
display(team_mse)
display(solo_mse)
```

```
0.002770219007218261

0.003122449752702039
```

```
display(team_r2)
display(solo_r2)
```

```
0.852311615262563

0.7412622079256567
```





Model Selection

Model Evaluation Results:

1. **Linear Regression**
   - **MSE:** 0.0020
   - **MAE:** 0.0374
   - **R-squared:** 0.8853

   Interpretation:
   The Linear Regression model performs reasonably well with an R-squared of 0.8853, indicating that it explains about 88.53% of the variance in the target variable. The error metrics, MSE and MAE, are within an acceptable range but are slightly higher compared to the other models.
2. **Random Forest**
   - **MSE:** 0.0018
   - **MAE:** 0.0314
   - **R-squared:** 0.9003

   Interpretation:
   The Random Forest model performs the best out of the three models, with the lowest MSE (0.0018) and MAE (0.0314), indicating more accurate predictions. The R-squared value of 0.9003 shows that it captures about 90.03% of the variance in the target variable, making it the strongest performer.
3. **XGBoost**
   - **MSE:** 0.0021
   - **MAE:** 0.0344
   - **R-squared:** 0.8804

   Interpretation:
   The XGBoost model performs similarly to Linear Regression, with an R-squared of 0.8804, slightly lower than Random Forest. While it does not outperform Random Forest, XGBoost remains a strong model with relatively low MSE and MAE values.

Prediction Results Before Optimization (Random Forest)

Top 10 Most Accurate Predictions:

player_name	real_name	team	actual_rating	predicted_rating	rating_difference	abs_difference
AdreN	Dauren Kystaubayev	no team	0.333333	0.333137	-0.000196	0.000196
SEMINTE	Valentin Bodea	no team	0.274510	0.274314	-0.000196	0.000196
f0rest	Patrik Lindberg	no team	0.568627	0.568039	-0.000588	0.000588
regali	Iulian Harjău	entropiq	0.666667	0.667255	0.000588	0.000588
eraa	Sean Knutsson	cph wolves	0.509804	0.508824	-0.000980	0.000980
interz	Timofey Yakushin	cloud9	0.352941	0.351569	-0.001373	0.001373
oskarish	Oskar Stenborowski	no team	0.313725	0.312157	-0.001569	0.001569
asap	Tyson Paterson	rooster	0.647059	0.649216	0.002157	0.002157
tarik	Tarik Celik	no team	0.431373	0.429216	-0.002157	0.002157
innocent	Paweł Mocek	rebels	0.352941	0.355490	0.002549	0.002549

Optimization

We performed iterative optimization on both models:

1. Baseline Models:
    - Initial performance was strong, with Random Forest achieving an R^2 of 0.9003 and XGBoost achieving 0.8804 on the test set.
2. Basic Tuning:
    - We used RandomizedSearchCV to explore different hyperparameters.
    - This included tuning parameters like number of estimators, max depth, and learning rate.
3. Advanced Tuning:
    - We further refined the models with more specific hyperparameter ranges.
    - Additional parameters like min_samples_split, max_features, and subsample were tuned.

RandomForest/XGBoost cont.

```
RandomForest:
  Baseline   - MSE: 0.0018, R2: 0.9003, MAE: 0.0314, CV MSE: 0.0016
  Optimized - MSE: 0.0018, R2: 0.8962, MAE: 0.0321, CV MSE: 0.0016
  Improvement - MSE: -4.13%, R2: -0.46%, MAE: -2.29%, CV MSE: 2.00%

XGBoost:
  Baseline   - MSE: 0.0021, R2: 0.8804, MAE: 0.0344, CV MSE: 0.0018
  Optimized - MSE: 0.0017, R2: 0.9025, MAE: 0.0302, CV MSE: 0.0016
  Improvement - MSE: 18.45%, R2: 2.51%, MAE: 12.18%, CV MSE: 11.98%
```

After optimization:

- Random Forest:
  - Slight decrease in performance (R^2 from 0.9003 to 0.8962)
  - This suggests the baseline model was already well-tuned for our data.
- XGBoost:
  - Significant improvement (R^2 from 0.8804 to 0.9025)
  - 18.45% reduction in Mean Squared Error
  - 12.18% improvement in Mean Absolute Error

XGBoost emerged as our best-performing model after optimization.

Extra Phase

Encountered Challenges

1) We weren’t granted access to the API we originally desired.
    a) We created scrapers to gather data instead.
    b) We were granted access to the API on September 25th (4:29 AM), a full week after we requested access.
2) 50%+ of data columns were objects.
    a) We created a loop that would fix those columns to int/float values.
3) Age Value accidentally replaced by ‘23’ for everyone without realizing until last minute
    a) We removed this data column, it will be used in the future to help with further progress

Future Plans pt.1

If provided more time we would likely be able to. . .

- Create a Streamlit App where we could predict a player's rating after inputting data.
- Create an extension that could cobble together the best fantasy eSport team
- Expand VictorVis into covering stats from games other than Counter-Strike

Future Plans pt.2

- Utilizing the Age and Country columns for future interpretation/analysis.
- Run Live Predictions in the model/future presentations.

QUESTIONS?

VICTORYVISION

ESPORTS DONE EASY