Nuttawut Thuayhanruksa 640631030
Nattawat Wattanawiput 640631031
Sonram Sirirat 640631037

PUBG player skill prediction by Machine Learning Regression Model

Abstract:

In a PUBG game, up to 100 players start in each match Players can be on teams that get ranked at the end of the game based on how many other teams are still alive when they are eliminated. In-game, players can pick up different munitions, revive knocked teammates, drive vehicles, swim, run, shoot, and experience all the consequences such as falling too far or running themselves over and eliminating themselves. To win in PUBG, using boosts item such as energy boost, painkiller, adrenaline and dealing damage are needed because there is high correlation between win place percentage. In regression model, polynomial is high score and low error rate. In solo mode, R2 score is 94.31 and Root Mean Squared Error is 0.0706 with polynomial degree is 2. In team mode, R2 score is 92.63 and Root Mean squared Error is 0.0851 with polynomial degree is 3.

1. Introduction & background

PUBG also known as PlayerUnknown's Battlegrounds is a battle royale shooter that pits 100 players against each other in a struggle for survival. Gather supplies and outwit your opponents to become the last person standing. PUBG is released and developed by KRAFTON, Inc. in 2017. There are 3,236,027 players. High participation of players and sponsors, make PUBG as a successful game tournament with the largest award in the world in 2018. The first major international tournament called "PUBG Global Invitational 2018".

Generally, there are many solutions to win in PUBG such as eliminate all enemies, escape from everything, or stay idle and eliminate the last one enemy but there is only one player or one group of players to win in PUBG. The research will serve three goals:

- To predict win-rate by PUBG mechanics and high-level technic in PUBG dataset.
- To improve model performance.
- To visualize the report to E-sports investor.

Referring to my previous point, the prize pool for PUBG competitions stood at 6.19 million U.S. dollars in 2020. Investing to provide in E-sports player are attracted for E-sports investor. For example, in Buriram united E-sports (Thai professional esports organization) have bought high skilled player to join E-sports team. It's like to transfer window or transfer player in football club which high skilled player will get more performance fee.

It is interesting to research because strategy, tactics and game mechanic have considerations to survive or win in the game. There are several factors to predict percentile of winning placement. The purpose of research to determine the datamining prediction model to predict and find relationship between attributes. Before that, data model requires data preprocessing and exploratory data analyzing to acquaint features of data.

2. Related Concept, Theories and Literature review: Core and Functional or supp

Exploratory data analysis (EDA) from Mohsin Raza [2] to find relationship between each feature which is a suitable model for making use and referring to prediction model because EDA model because it is clearly explained and cover all content. So, we took an example of this model.

Baseline-Linear Regression [3] This model represents to classified the relationship between match, group id by average of win place percentage. R-score in this model is 94.92 which is close to our model but they used only one feature. In my opinion, other features are important as well. Therefore, we choose to use multiple regression and polynomial regression to predict the model

Prediction by regression model [4] Prediction to find the feature importance. The result is killplace, numgroup and walkdistance are the most important but in this model MSE is 0.319 higher than our model. This model did not focus on win place percentage at all.

3. Research & Methodology

Exploratory Data Analysis (EDA)

Data preparation and cleaning method. We stated from data understanding from each feature, which are stand for. After that, we used IQR technique to separate 2 types of players by newbie-player ‘s score and professional-player’s score.

From the previous, there are two types of players. Data outlier detecting or data examine are interesting in data preparation method. In other words, professional-player score should be an outlier by higher score such as, kill, damageDealt score. Conversely, newbie-player will get less score.

Thenceforth, In Exploratory data analysis method (EDA). We looked at relationship between independence feature and target feature. The target feature of dataset is winPlacePec (win-rate ratio) by find a relationship by Univariate analysis, Bivariate analysis, and Multivariate analysis method.

We considered should separate the data by grouping by match type to Solo match type and Team match type, features in each data are difference. Some features are not supposed to be in Solo match type, but Team match type is related such as assists, revives and teamkills.

Data mining prediction by Multiple Linear Regression and Polynomial Regression

1. Multiple Linear Regression

Multiple linear regression has one y and two or more x variables. It is an extension of Simple Linear regression as it takes more than one predictor variable to predict the response variable.

Multiple Linear Regression is one of the important regression algorithms which models the linear relationship between a single dependent continuous variable and more than one independent variable.

Assumptions for Multiple Linear Regression:

1. A linear relationship should exist between the Target and predictor variables.
2. The regression residuals must be normally distributed.
3. MLR assumes little or no multicollinearity (correlation between the independent variable) in data.

$$Y_i = \beta_0 + \ \beta_1 X_{i1} + \ \beta_2 X_{i2} + \cdots + \ \beta_{p-1} X_{i,\ p-1} + \ \varepsilon_i$$

Y = Dependent variable / Target variable
β_0 = Intercept of the regression line
$\beta_1, \beta_{2,} \ldots \beta_n$ = Slope of the regression lime which tells whether the line is increasing or decreasing
$X_1, X_2, \ldots X_n$ = Independent variables / Predictor variables**
ε = Error

2. Polynomial Regression

Polynomial regression is a special case of linear regression where we fit a polynomial equation on the data with a curvilinear relationship between the target variable and the independent variables.

In a curvilinear relationship, the value of the target variable changes in a non-uniform manner with respect to the predictor (s)

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \varepsilon$$

Y = Dependent variable / Target variable
β_0 = Intercept of the regression line
$\beta_1, \beta_{2,} \ldots \beta_n$ = Slope of the regression lime which tells whether the line is increasing or decreasing
ε = Error

The number of higher-order terms increases with the increasing value of n, and hence the equation becomes more complicated.

3. Regression Evaluation Metrics

Evaluation metrics for regression problems as below:

Coefficient of determination (R2 score) is used to evaluate the performance of a linear regression model. It is the amount of the variation in the output dependent attribute which is predictable from the input independent variable(s). It is used to check how well-observed results are reproduced by the model, depending on the ratio of total deviation of results described by the model.

$$R2 = 1 - SSres/SStot$$

Mean Absolute Error (MAE): the mean of the absolute value of the errors is the easiest to understand, because it's the average error.

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Mean Squared Error (MSE): the mean of the squared errors is more popular than Mean Absolute Error (MAE), because the MSE “punishes” larger errors, which tends to be useful in the real world.

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_j\right)^2$$

Root Mean Squared Error (RMSE): the square root of the mean of the squared errors is even more popular than MSE, because RMSE is interpretable in the "y" units.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_j\right)^2}$$

4. Dataset and Features

Data description

We use Kaggle datasets [1] for PUBG players. This dataset contains a detailed listing table with 29 raw input columns/features. The PUBG dataset contains a total of 1,111,742 listings/rows. we split the dataset into train:test with a ratio of 7:3 for each dataset to run a machine learning model.

Features Description

Features are chosen only if they are likely to correlate with winPlacePerc feature. Therefore, we dropped features like Id, matchId, groupId.

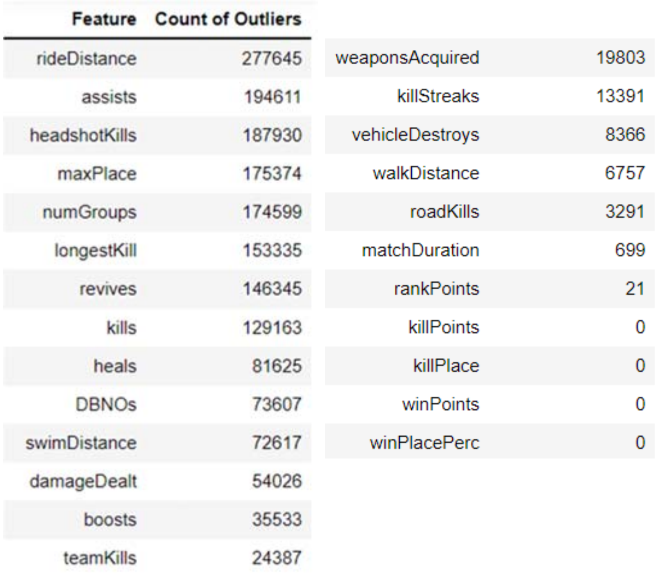
- Categorical features: matchType: Identifying the game mode (Solo, Duo, Squad).
- Continuous features:
    - DBNOs: Number of enemy players knocked.
    - assists: Number of enemies that players damaged and killed by teammates.
    - boosts: Number of boost items used.
    - damageDealt: Total damage dealt not include self-damage.
    - headshotKills: Number of enemy players killed with headshots.
    - heals: Number of healing items used. Items
    - killPlace: Ranking in a match of the number of enemy players killed.
    - killPoints: Kills-based external ranking of players.
    - killStreaks: Max number of enemy players killed in a short amount of time.
    - kills: Number of enemy players killed.
    - longestKill: longest distance between player and player killed at time of death.
    - matchDuration: Duration of match in seconds.
    - rankPoints: A ranking system that where the player is rank after the game finish.
    - revives: Number of times this player revived teammates.
    - rideDistance: Total distance travelled in vehicles measured in meters.
    - roadKills: Number of kills while in a vehicle.
    - swimDistance: Total distance traveled by swimming measured in meters.
    - teamKills: Number of times this player killed a teammate.
    - vehicleDestroys: Number of vehicles destroyed.
    - walkDistance: Total distance traveled on foot measured in meters.
    - weaponsAcquired: Number of weapons picked up.
    - winPoints: Win-based external ranking of players.
    - numGroups: Number of groups within the match.
    - maxPlace: Worst placement in the match.
    - winPlacePerc: Percentile winning placement.

Data cleaning

Identify the null in each column. There is 1 missing value then we drop missing value. There are 1,111,741 rows in dataset.

Detect outliers

In this method by using Inter Quartile Range (IQR), To detect outliers. Any value, which is beyond the range of -1.5 x IQR to 1.5 x IQR is treated as outliers.



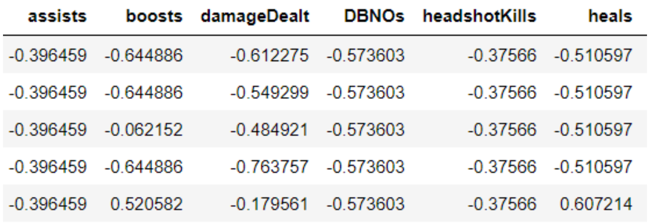
Feature	Count of Outliers
rideDistance	277645
assists	194611
headshotKills	187930
maxPlace	175374
numGroups	174599
longestKill	153335
revives	146345
kills	129163
heals	81625
DBNOs	73607
swimDistance	72617
damageDealt	54026
boosts	35533
teamKills	24387
weaponsAcquired	19803
killStreaks	13391
vehicleDestroys	8366
walkDistance	6757
roadKills	3291
matchDuration	699
rankPoints	21
killPoints	0
killPlace	0
winPoints	0
winPlacePerc	0

Figure 1 Count of outliers for each feature.

In this case. We are dealing with game datasets. There is only one player or one team to win the game with almost 100 players then we do nothing with outlier.

Data Transformation

We use Scaling transformation "Z-score normalization" to numerical data and ignore categorical data.



assists	boosts	damageDealt	DBNOs	headshotKills	heals
-0.396459	-0.644886	-0.612275	-0.573603	-0.37566	-0.510597
-0.396459	-0.644886	-0.549299	-0.573603	-0.37566	-0.510597
-0.396459	-0.062152	-0.484921	-0.573603	-0.37566	-0.510597
-0.396459	-0.644886	-0.763757	-0.573603	-0.37566	-0.510597
-0.396459	0.520582	-0.179561	-0.573603	-0.37566	0.607214

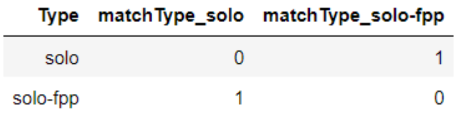
Figure 2 Some of the results after normalization.

Data preparation

We considered should separate the data by grouping by match type. Because in solo match some of features are not related. For example, assists, revives and teamKills. The numGroups in Solo match type just tell us how many players within match, so we considered drop it.

Solo match type

One hot encoding creates new (binary) columns, indicating the presence of each possible value from the original data. After encoding there are 2 match types.

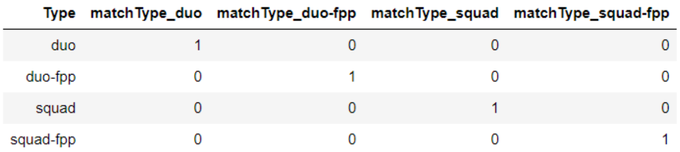


Type	matchType_solo	matchType_solo-fpp
solo	0	1
solo-fpp	1	0

Figure 3 One hot encoding solo match type.

Team match type

One hot encoding creates new (binary) columns, indicating the presence of each possible value from the original data. After encoding there are 4 match types.



Type	matchType_duo	matchType_duo-fpp	matchType_squad	matchType_squad-fpp
duo	1	0	0	0
duo-fpp	0	1	0	0
squad	0	0	1	0
squad-fpp	0	0	0	1

Figure 4 One hot encoding team match type.

After that, we split PUBG dataset to solo match type dataset (Solo mode) and team match type dataset (Team mode) with each one hot encoding. We will use 2 datasets to run machine learning model in the next part.

5. Exploratory data analysis (EDA)

5.1 Univariate analysis

5.1.1 damageDealt

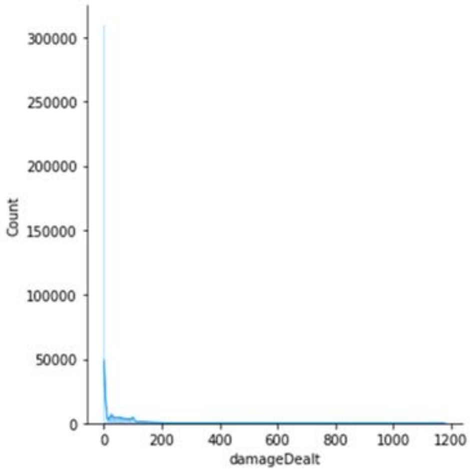


Figure 5 Distribution of damage dealt with other players.

A distribution of how much damage, players that don't kill anyone, can inflict on their enemies. We can see that most player don't deal out too much, this is most likely all the new players trying to figure out the controls and getting to know the game while they continually get beaten up the more experience players.

5.1.2 Kills and killStreaks

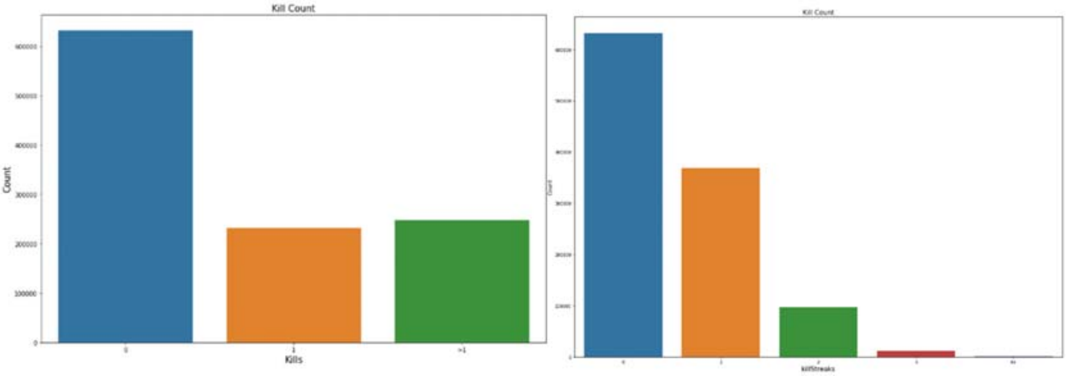


Figure 6 Show the number of eliminating players separated by kill count.

Charts of kill Count and killStreaks. Show the number of eliminate some players, most of players got zero for kill Count and killStreaks.

5.2 Bivariate analysis

5.2.1 winPlacePerc and damageDealt

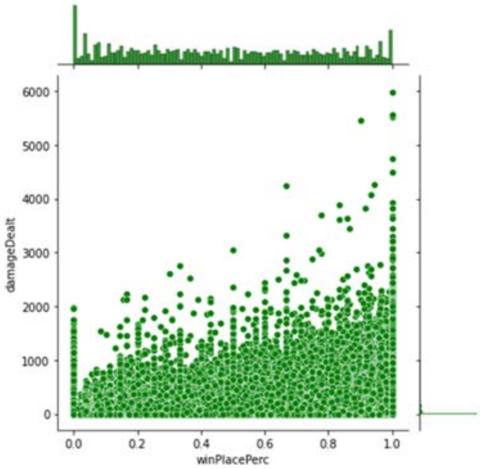


Figure 7 Correlation between win chance and damage dealt.

There is a reasonable correlation here with the damage we deal out to enemy players and winPlacePerc.

5.2.2 winPlacePerc and Total distance travelled

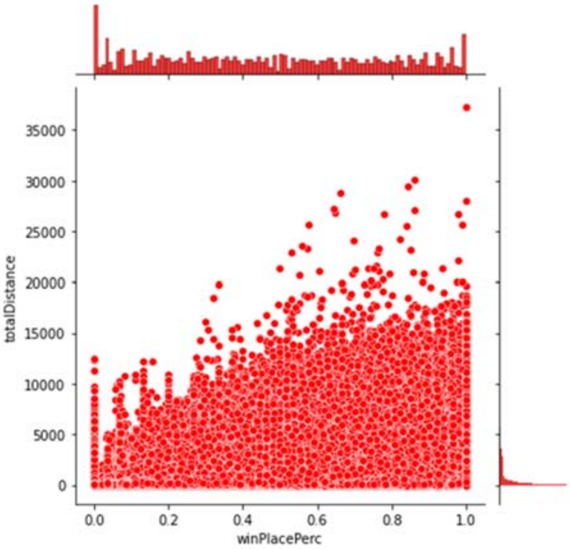


Figure 8 Correlation between win chance and travel distance.

There is a reasonably strong correlation with the total distance travelled and winning, although most of this correlation may just be due to the strong correlation with walking distance and winPlacePerc. However, one interesting item to note is that it looks like the person that travelled the longest distance didn't win, when they travelled over 41 kms is a single match.

5.2.3 winPlacePerc and boosts

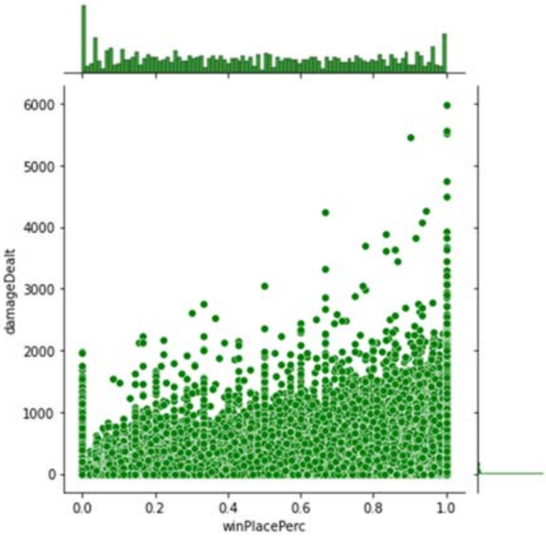


Figure 6 Correlation between win chance and boosts.

There is a reasonably correlation with boosts and winning. However, one interesting item to note is that it looks like the person that travelled the longest distance didn't win, when they travelled over 41 kms is a single match.

5.2.4 Heals and Boosts

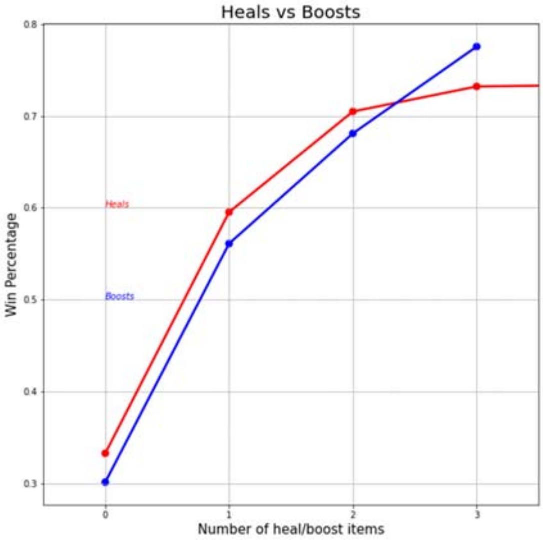


Figure 7 Correlation between heat and boost items.

Here we can see how the heal items and boost items are used compared to each other. This seems to indicate that using a few healing items increase your chance of winning, but you need to use more boosts to actually achieve a change of wining.

5.3 Multivariate analysis

Correlation between all features

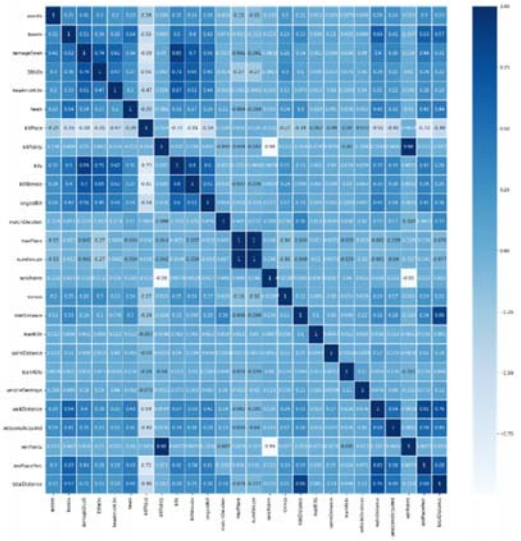


Figure 8 Features correlation (blue – higher, white – lower)

1. According to the color bar we can find the correlation between different features.
2. If correlation is positive, one variable increases with other.
3. If correlation is negative, as one variable increases, the other decreases.
4. if correlation is 1, it means that either the variables are same, or they are almost same.

6. Methods

Prediction player skill in Solo mode

Data set: PUBG_solo_dataset.csv

Model Building: 1. Multiple Linear Regression 2. Polynomial Regression (degree = 2, 3, 4)

Train and Test split: Separate Training set 70%, Testing set 30% at Random State = 0.

Skit learn Model: **from** sklearn.linear_model **import** LinearRegression,

from sklearn.preprocessing **import** PolynomialFeatures

Regression Evaluation Metrics

	Multiple Linear Regression	Polynomial Degree = 2	Polynomial Degree = 3	Polynomial Degree = 4
Coefficient of determination (R2 score)	88.66	94.31	95.31	95.73
Mean Absolute Error (MAE)	0.0705	0.048	0.0453	0.3874
Mean Squared Error (MSE)	0.0099	0.0054	0.0058	1989.30
Root Mean Squared Error (RMSE)	0.0997	0.0706	0.0764	44.602

Predictions from our Model

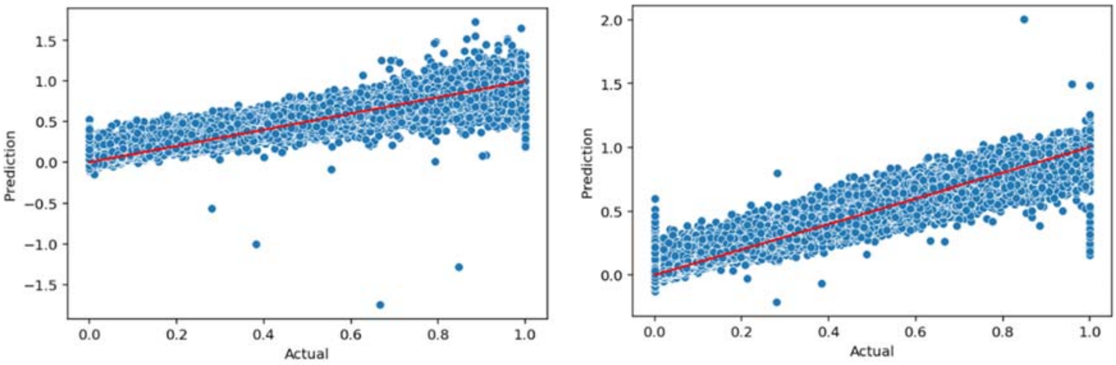
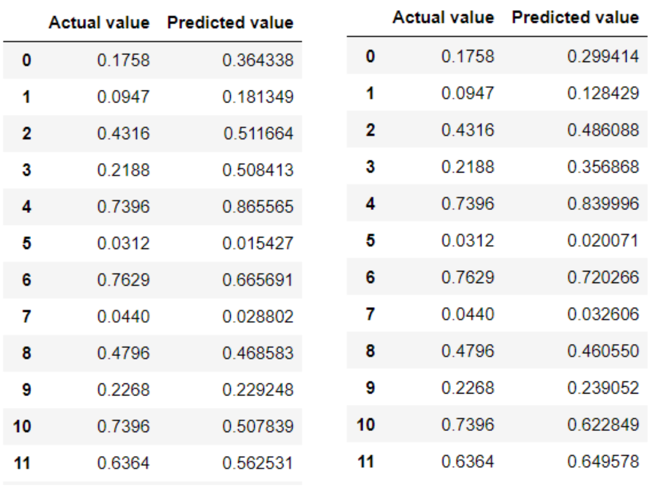


Figure 9 Multiple Linear Regression (Left), Polynomial Regression degree = 2 (Right)

Random sample test: Actual value vs Predicted value



	Actual value	Predicted value
0	0.1758	0.364338
1	0.0947	0.181349
2	0.4316	0.511664
3	0.2188	0.508413
4	0.7396	0.865565
5	0.0312	0.015427
6	0.7629	0.665691
7	0.0440	0.028802
8	0.4796	0.468583
9	0.2268	0.229248
10	0.7396	0.507839
11	0.6364	0.562531

	Actual value	Predicted value
0	0.1758	0.299414
1	0.0947	0.128429
2	0.4316	0.486088
3	0.2188	0.356868
4	0.7396	0.839996
5	0.0312	0.020071
6	0.7629	0.720266
7	0.0440	0.032606
8	0.4796	0.460550
9	0.2268	0.239052
10	0.7396	0.622849
11	0.6364	0.649578

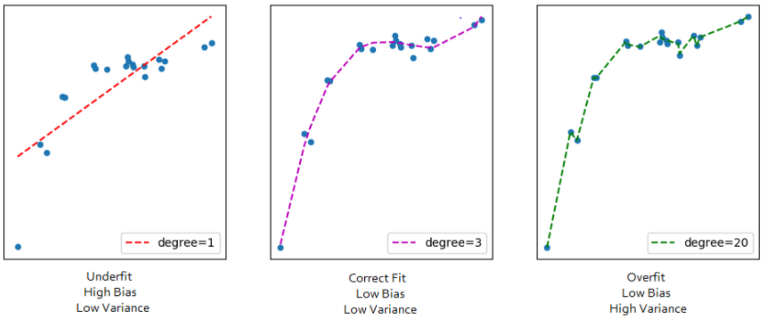
Figure 10 Multiple Linear Regression (Left), Polynomial Regression (Right)

Conclusion

In this data set Polynomial regression get all result better than Multiple linear regression in term of Regression Evaluation Metrics and Random sample test: Actual value vs Predicted value.

For degree of Polynomial regression, we decide to use degree = 2 that make to model balance (See at best of RMSE) and not over fitting if degree = 1 the result similar likely Multiple linear regression (under fitting).

For PUBG_Solo_Mode_dataset we choose model to prediction is Polynomial regression with degree =2 and developing for this project in the future we will solving problem in result of prediction that value get over 1.00 and under 0.00



Underfit
High Bias
Low Variance

Correct Fit
Low Bias
Low Variance

Overfit
Low Bias
High Variance

Figure 11 Overfitting and Underfitting with Machine Learning Algorithms

Prediction player skill in Team mode

Data set: PUBG_Team_dataset.csv

Model Building: 1. Multiple Linear Regression, 2. Polynomial Regression(degree=1,2,3)

Train and Test split: Separate Traing set 70%, Testing set 30% at Random State = 0.

Skit learn Model: **from** sklearn.linear_model **import** LinearRegression,

from sklearn.preprocessing **import** PolynomialFeatures.

Regression Evaluation Metrics

	Multiple Linear Regression	Polynomial degree=1	Polynomial degree=2	Polynomial degree=3
Coefficient of determination (R2 score)	83.90	83.90	91.26	92.63
Mean Absolute Error (MAE)	0.0909	0.0909	0.0670	0.062
Mean Squared Error (MSE)	0.0153	0.0153	0.0084	0.072
Root Mean Squared Error (RMSE)	0.1238	0.1238	0.0914	0.0851

Predictions from our Model

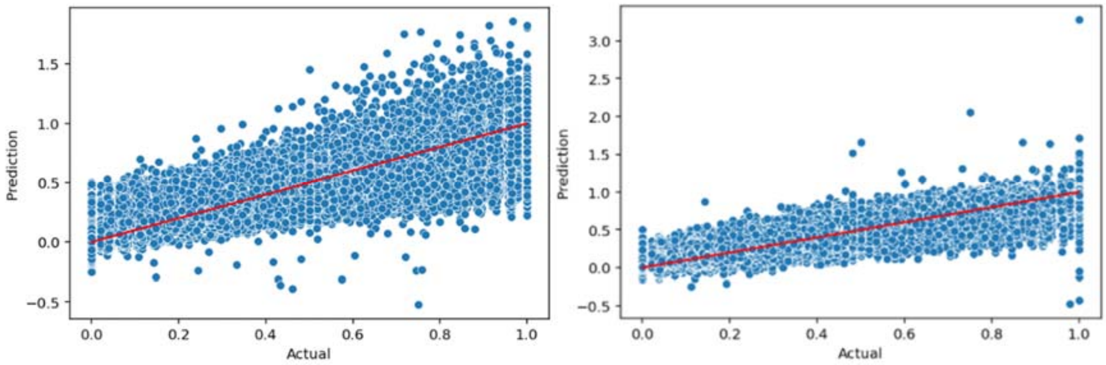
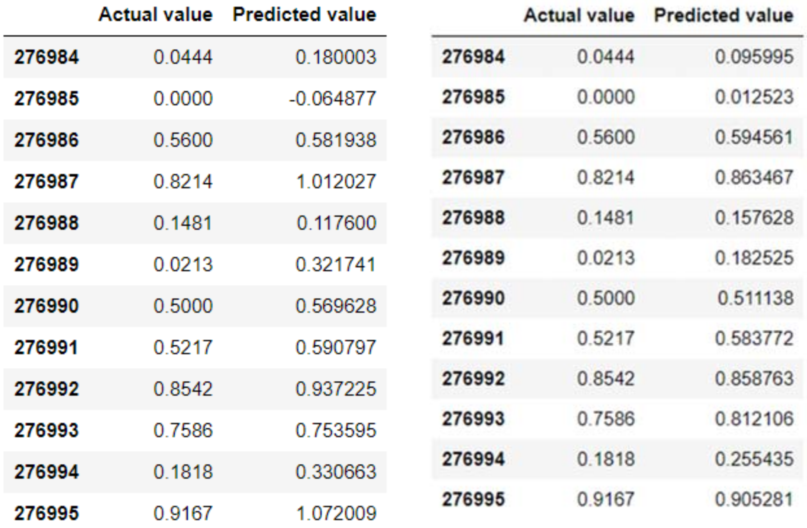


Figure 11 Multiple Linear Regression (Left), Polynomial Regression degree = 3 (Right)

Random sample test: Actual value vs Predicted value



	Actual value	Predicted value
276984	0.0444	0.180003
276985	0.0000	-0.064877
276986	0.5600	0.581938
276987	0.8214	1.012027
276988	0.1481	0.117600
276989	0.0213	0.321741
276990	0.5000	0.569628
276991	0.5217	0.590797
276992	0.8542	0.937225
276993	0.7586	0.753595
276994	0.1818	0.330663
276995	0.9167	1.072009

	Actual value	Predicted value
276984	0.0444	0.095995
276985	0.0000	0.012523
276986	0.5600	0.594561
276987	0.8214	0.863467
276988	0.1481	0.157628
276989	0.0213	0.182525
276990	0.5000	0.511138
276991	0.5217	0.583772
276992	0.8542	0.858763
276993	0.7586	0.812106
276994	0.1818	0.255435
276995	0.9167	0.905281

Figure 12 Multiple Linear Regression (Left), Polynomial Regression (Right)

Conclusion

In this data set Polynomial regression get all result better than Multiple linear regression in term of Regression Evaluation Metrics and Random sample test: Actual value vs Predicted value.

For degree of Polynomial regression, we used degree = 3 that make to model balance (See at RMSE) and not over fitting, if degree = 1 the result similar with Multiple linear regression (under fitting)

For PUBG_Team_Mode_dataset we choose model to prediction is Polynomial regression with degree =3 and developing for this project in the future we will solving problem in result of prediction that value get over 1.00 and under 0.00.

7. Summary

From EDA table show the relationship between 2 attributes. It indicates that many attributes such a walk distance, total distance, Boosts, Damage dealt, Kills have high correlation between win rate percentage (winplaceperc). Therefore, appropriate strategy of playing PUBG is travelling by foot and always rotation to others location. In the case of the fight boosts item is needed for increase character efficiency. For example, energy drink to increase character moving movement speed and reloading speed, painkiller to automate regeneration the wound and Adrenaline to perfectly increase character efficiency. Also, killing the enemy with one fight (without escape) is the one of best strategy to win PUBG. However, reviving teammate is not a good idea because reviving will interrupt the fight.

From Polynomial regression. It indicates that win prediction rate for all attributes. For investors who interested in PUBG E-sport team. Investment in Solo player is the good way because prediction rate in Solo mode is higher accuracy than Team mode because in the model R2 score in solo mode is higher than team mode. Therefore, strategy is easier and more precise as is evident from Regression Evaluation Metrics in root mean square error in solo mode is less than team mode. However, investment should depend on the prize of type of competition. For example, PUBG team mode has higher price of reward than solo mode in every year If investor prefer high risk and high return investing in team mode is good choice but strategy to win planner should follow the strategy from EDA table.

Regression Evaluation Metrics

	Polynomial in solo mode	**Polynomial in team mode**
Coefficient of determination (R2 score)	94.31	92.63
Mean Absolute Error (MAE)	0.048	0.062
Mean Squared Error (MSE)	0.0054	0.072
Root Mean Squared Error (RMSE)	0.0706	0.0851

8. Reference

[1] PUBG_Dataset, Mohsin Raza. Machine Learning Engineers at Moses Technologies. Retrieved 30-05-2021. Available from: https://www.kaggle.com/razamh/pubg-dataset

[2] PUBG Exploratory Data Analysis (EDA), Mohsin Raza. Machine Learning Engineers at Moses Technologies. Retrieved 30-05-2021.Avalible from: https://www.kaggle.com/razamh/pubg-exploratory-data-analysis-eda

[3] Baseline-Linear Regression, Yan Sun, Learner at UC San Diego Retrieved 16-12-2018. Available from: https://www.kaggle.com/yansun1996/baseline-linearregression?scriptVersionId=8500777

[4] PUBG_Predictions, DANIIL LEVIN Retrieved 6-5-2021. Available from: https://www.kaggle.com/dzlevin/pubg-predictions