UNIVERSITY OF ASIA PACIFIC

Department of Computer Science & Engineering

Course Title – Artificial Intelligence and Expert Systems Lab.

Course Code – CSE-404.

Project – Implementation of Multivariable Linear Regression Using A Public Dataset

SUBMITTED BY

Shawan Das.

ID – **19101020**

Section – **A**1

SUBMITTED TO

Dr. Nasima Begum

University of Asia Pacific

Date of Submission – 25–10–2022

Problem Title: Implement Multivariable Linear Regression Using a Public Dataset

Problem Description: Implementation of Linear Regression model with a dataset. The Dataset must be multivariant. At the basis of other parameters, we have to predict another parameter.

Objective: There are several approach in Machine Learning to predict a data at the basis of other data. In this project we are going to implement "Linear Regression"- model to predict data.

For this approach, I'm going to use a game(Call of Duty) dataset which is about (1558, 19) in size . But we will only use (1558,15) data for independent(X) axis and (1558,) data for dependent(y) axis.

Dataset: Call of Duty

Dataset Info:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1558 entries, 0 to 1557
Data columns (total 19 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   name            1558 non-null   object
 1   wins            1558 non-null   int64
 2   kills           1558 non-null   int64
 3   kdRatio         1558 non-null   float64
 4   killstreak      1558 non-null   int64
 5   level           1558 non-null   int64
 6   losses          1558 non-null   int64
 7   prestige        1558 non-null   int64
 8   hits            1558 non-null   int64
 9   timePlayed      1558 non-null   int64
 10  headshots       1558 non-null   int64
 11  averageTime     1558 non-null   float64
 12  gamesPlayed     1558 non-null   int64
 13  assists         1558 non-null   int64
 14  misses          1558 non-null   int64
 15  xp              1558 non-null   int64
 16  scorePerMinute  1558 non-null   float64
 17  shots           1558 non-null   int64
 18  deaths          1558 non-null   int64
dtypes: float64(3), int64(15), object(1)
memory usage: 231.4+ KB
```

Processed Data:

```
1   dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1558 entries, 0 to 1557
Data columns (total 16 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   wins            1558 non-null   int64
 1   kdRatio         1558 non-null   float64
 2   killstreak      1558 non-null   int64
 3   level           1558 non-null   int64
 4   losses          1558 non-null   int64
 5   prestige        1558 non-null   int64
 6   hits            1558 non-null   int64
 7   timePlayed      1558 non-null   int64
 8   headshots       1558 non-null   int64
 9   averageTime     1558 non-null   float64
 10  gamesPlayed     1558 non-null   int64
 11  assists         1558 non-null   int64
 12  misses          1558 non-null   int64
 13  xp              1558 non-null   int64
 14  scorePerMinute  1558 non-null   float64
 15  shots           1558 non-null   int64
dtypes: float64(3), int64(13)
memory usage: 194.9 KB
```

for x axis, we will use 'wins', 'kdRatio', 'killstreak', 'level', 'losses', 'prestige' ,'hits', 'timePlayed', 'averageTime', 'gamesPlayed', 'assists', 'misses', 'xp', 'scorePerMinute', 'shots' – data

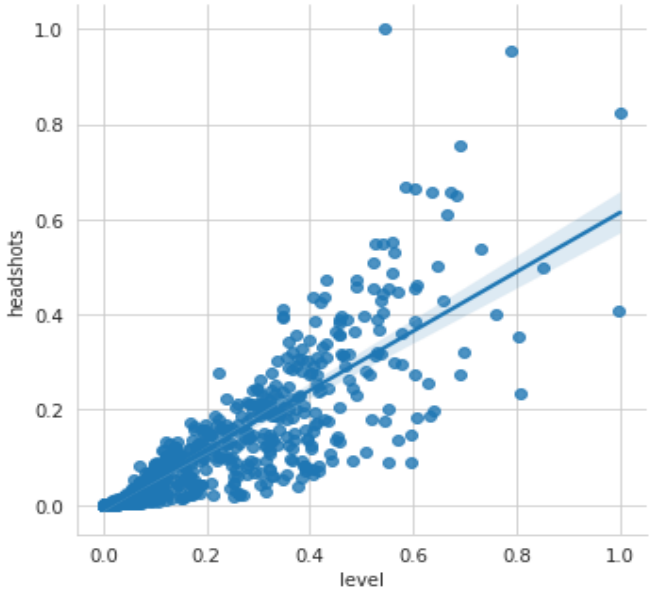
and for y-axis I will use “headShots” data. So basically I’m going to predict the Headshot values.

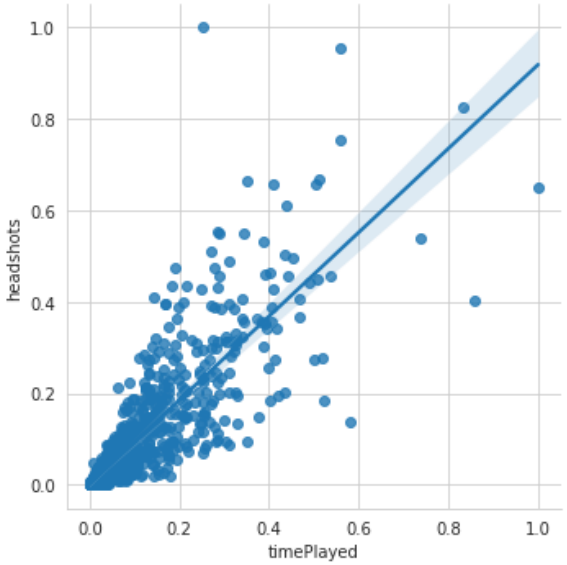
But before implementation, I’ve to normalize those data for better prediction.

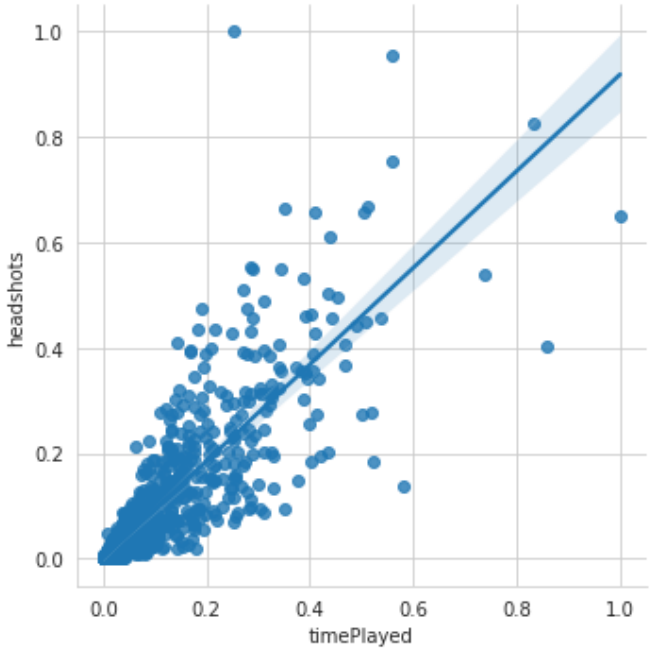
After Normalizing Data:

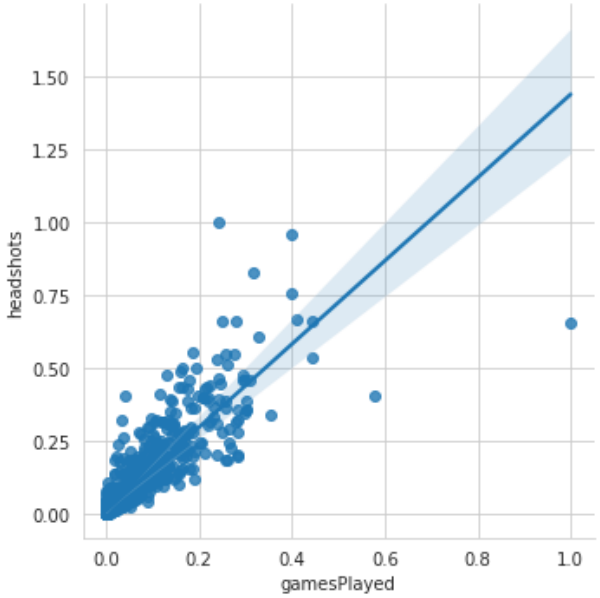
	wins	kdRatio	killstreak	level	losses	prestige	hits	timePlayed	headshots	averageTime	gamesPlayed	assists	misses	xp	scorePerMinute	shots
0	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.0	0.0
1	0.000000	0.000000	0.0	0.000000	0.0	0.940171	0.0	0.000936	0.000000	0.005189	0.0	0.000000	0.0	0.000047	0.0	0.0
2	0.000000	0.343750	0.0	0.018433	0.0	0.940171	0.0	0.004279	0.001365	0.023721	0.0	0.000069	0.0	0.003226	0.0	0.0
3	0.000853	0.133333	0.0	0.000000	0.0	0.000000	0.0	0.000401	0.000000	0.002224	0.0	0.000000	0.0	0.000077	0.0	0.0
4	0.000000	0.066667	0.0	0.000000	0.0	0.940171	0.0	0.000669	0.000085	0.003706	0.0	0.000000	0.0	0.000067	0.0	0.0

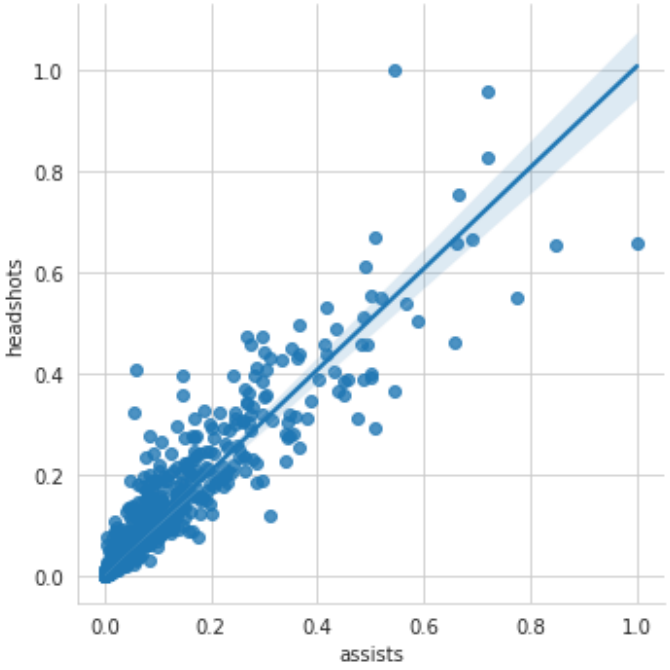
Plot Some Data:

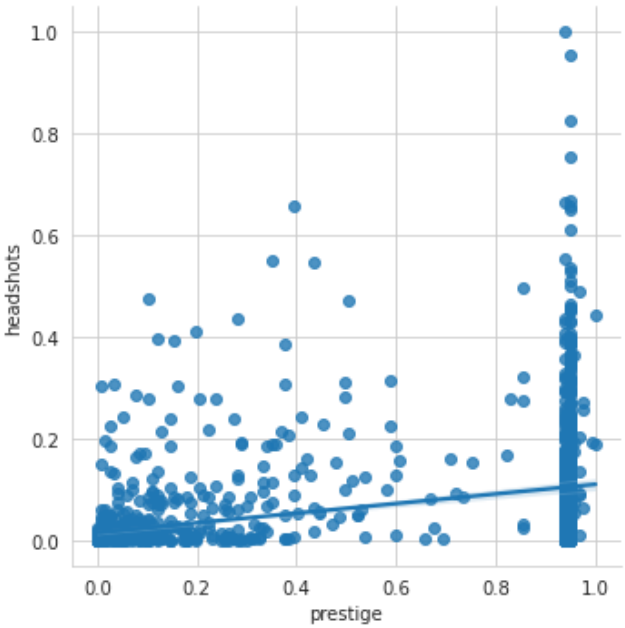


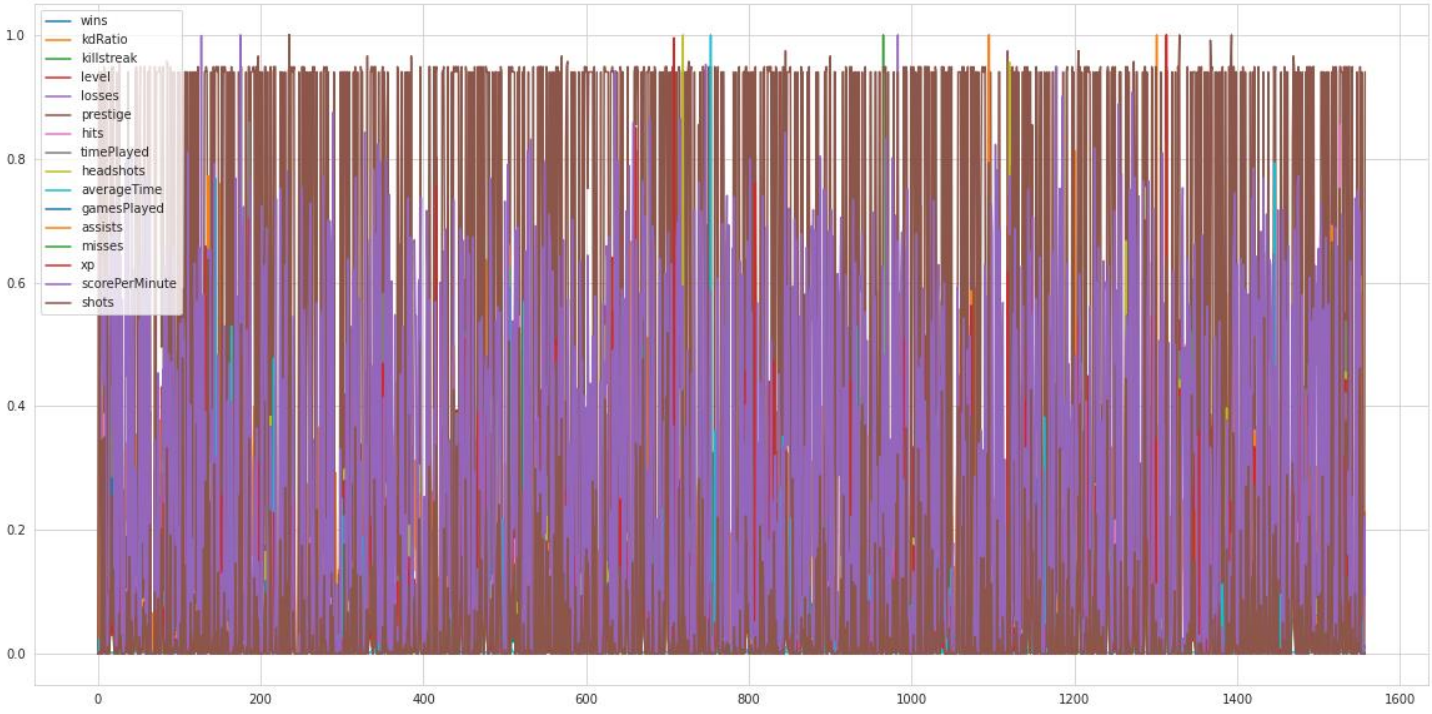








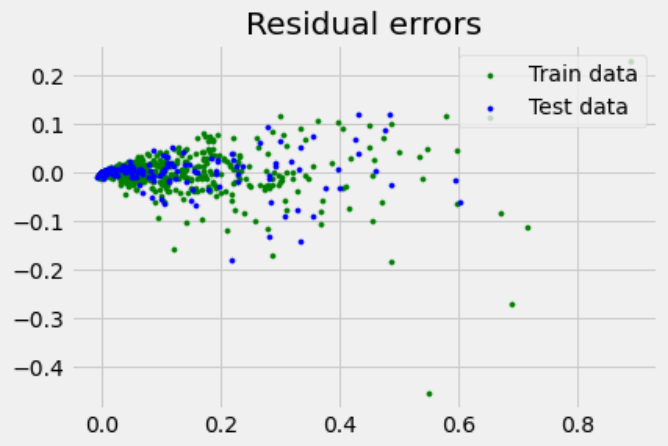


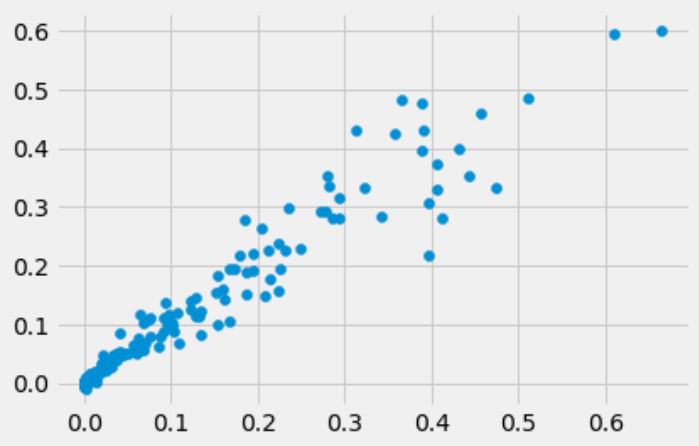


Tools & Languages:

- Language – Python
- IDE: Google Collab

Predictions Graph:





Source Code: SKLearn, Manual, Manual(2)

Challenges & Conclusion:

SkLearn implementation part was easier than the Manual approach. SkLearn model provided around 95.04%. But the manual part is incomplete because of some space errors.