

**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**

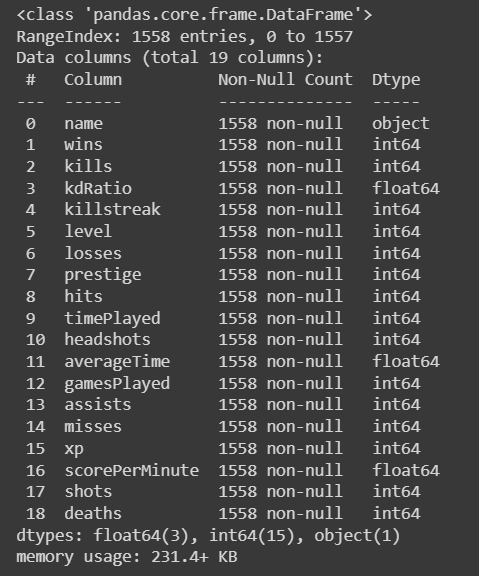
**Problem Description:** Implementation of Linear Regression model with a dataset. The Dataset must be multivariant. At the basis of other parameters, we have to predict another parameter.

**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.

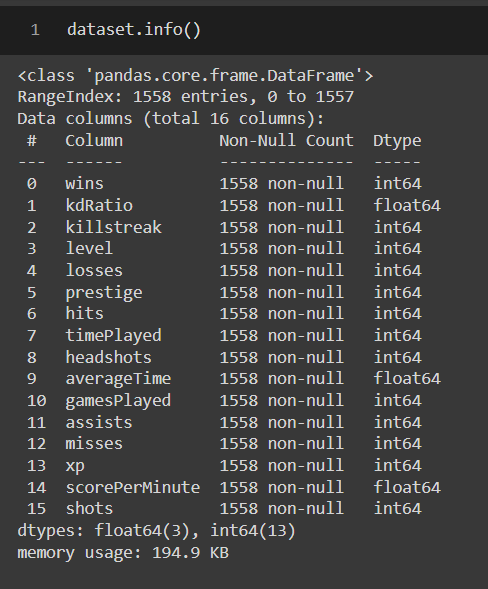
For this approach, I’m going to use a game(Call of Duty) dataset which is about (1558, 19) in size . But we will only use (1558,15) data for independent(X) axis and (1558,) data for dependent(y) axis.

Dataset: [Call of Duty](https://www.kaggle.com/datasets/aishahakami/call-of-duty-players)

**Dataset Info:**



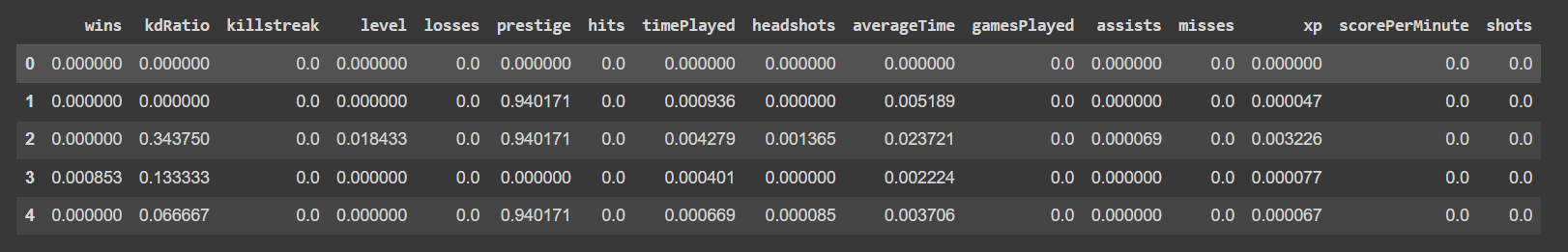
**Processed Data:**

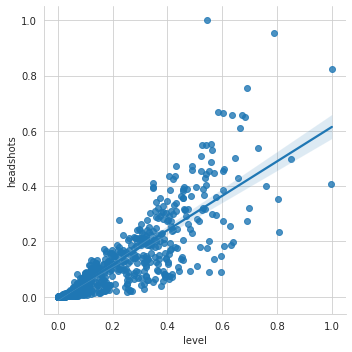
for x axis, we will use 'wins', 'kdRatio', 'killstreak', 'level', 'losses', 'prestige' ,'hits', 'timePlayed', 'averageTime', 'gamesPlayed', 'assists', 'misses', 'xp', 'scorePerMinute', 'shots' – data

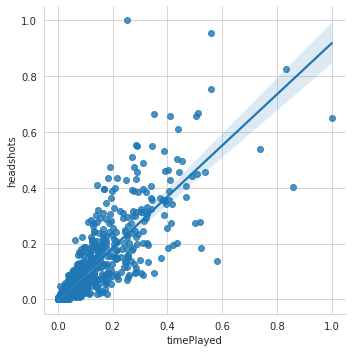
and for y-axis I will use “headShots” data. So basically I’m going to predict the Headshot values.

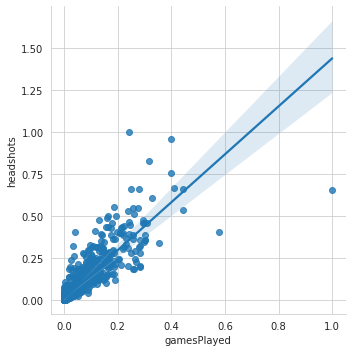
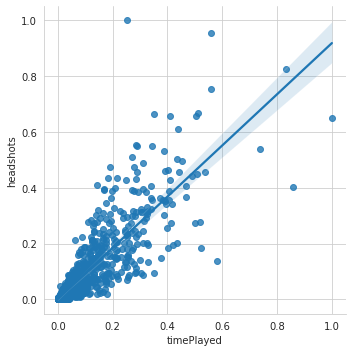
But before implementation, I’ve to normalize those data for better prediction.

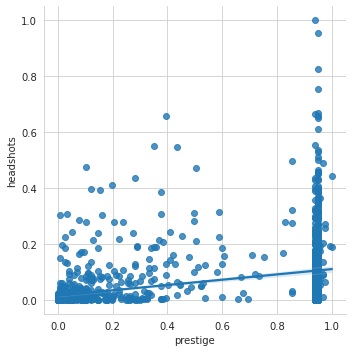
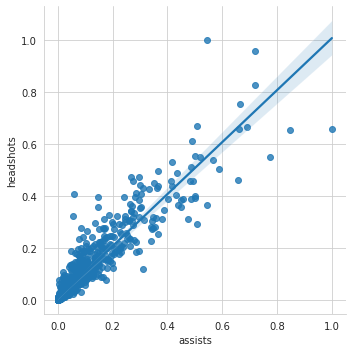
**After Normalizing Data:**

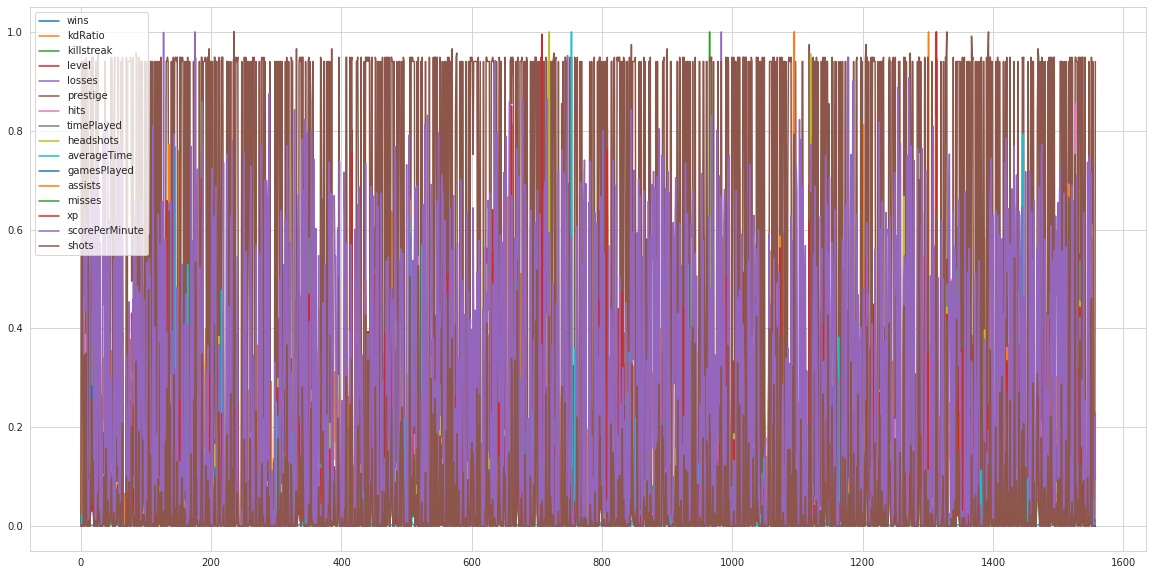


**Plot Some Data:**



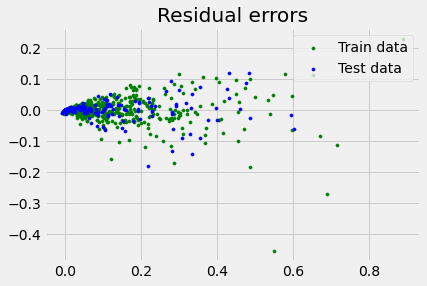


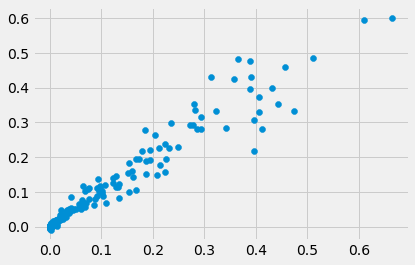




**Tools & Languages:**

* Language – Python
* IDE: Google Collab

**Predictions Graph:**

****

**Source Code:**  [SKLearn](https://colab.research.google.com/drive/1WIUqM6R4ud5VJvxP0SMDUjLKZfhtCZ2n?authuser=1#scrollTo=xANDGlCSHyHc), [Manual](https://colab.research.google.com/drive/14_ysP-95yz-uwEgNtWAb9KUYksGDsStc?authuser=1#scrollTo=CxQiYbOnRT62), [Manual(2)](https://colab.research.google.com/drive/1vxV4xNkbQQqyGGy66aFKiW5hKNmMO-Ve?authuser=1#scrollTo=M-pCZ5Npu_du)

**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.