**BLOG ON MY EVALUATION PROJECT- BASEBALL CASE STUDY:-**

**By:- Rishabh Johri**

**INTRODUCTION:-**

I Rishabh Johri want to thank Team Data Trained for giving me an opportunity to work on the Project Base Ball Case Study.

This is my first project Related to Sports. Sports plays a very important role in our life, participation in sports is very important for one's overall self development whether its physically, mentally or for confidence boosting.

I am very excited to show my Knowledge and skills in preparing and presenting project Baseball Case Study, The Project is based on the MLB league which was took place in the Year 2014. I tried to Develop a project which can Predict the Number of wins for a given team in the 2015 season, based on there performance in the year 2014.

Baseball is a bat-and-ball game played between two opposing teams, typically of nine players each, that take turns batting and fielding. The game proceeds when a player on the fielding team, called the pitcher, throws a ball which a player on the batting team tries to hit with a bat. The objective of the offensive team (batting team) is to hit the ball into the field of play, allowing its players to run the bases, having them advance counter-clockwise around four bases to score what are called "runs". The objective of the defensive team (fielding team) is to prevent batters from becoming runners, and to prevent runners' advance around the bases.[2] A run is scored when a runner legally advances around the bases in order and touches home plate (the place where the player started as a batter). The team that scores the most runs by the end of the game is the winner.

A player on the batting team who reaches first base without being called "out" can attempt to advance to subsequent bases as a runner, either immediately or during teammates' turns batting. The fielding team tries to prevent runs by getting batters or runners "out", which forces them out of the field of play. Both the pitcher and fielders have methods of getting the batting team's players out. The opposing teams switch back and forth between batting and fielding; the batting team's turn to bat is over once the fielding team records three outs. One turn batting for each team constitutes an inning. A game is usually composed of nine innings, and the team with the greater number of runs at the end of the game wins. If scores are tied at the end of nine innings, extra innings are usually played. Baseball has no game clock, although most games end in the ninth inning.

Baseball evolved from older bat-and-ball games already being played in England by the mid-18th century. This game was brought by immigrants to North America, where the modern version developed. By the late 19th century, baseball was widely recognized as the national sport of the United States. Baseball is popular in North America and parts of Central and South America, the Caribbean, and East, particularly in Japan, South Korea, and Taiwan.

In the United States and Canada, professional Major League Baseball (MLB) teams are divided into the National League (NL) and American League (AL), each with three divisions: East, West, and Central. The MLB champion is determined by playoffs that culminate in the World Series. The top level of play is similarly split in Japan between the Central and Pacific Leagues and in Cuba between the West League and East League. The World Baseball Classic, organized by the World Baseball Softball Confederation, is the major international competition of the sport and attracts the top national teams from around the world.

In this particular problem, we need to observe and analyse the wins details and try to analyse whether this is win or not in the game.

The steps which we are going to follow arefirst we will import and analyse the dataset, then exploratory data analysis (EDA) , data visualization, data cleaning, pre-processing, model building, model saving and final prediction to check the performance of our model.

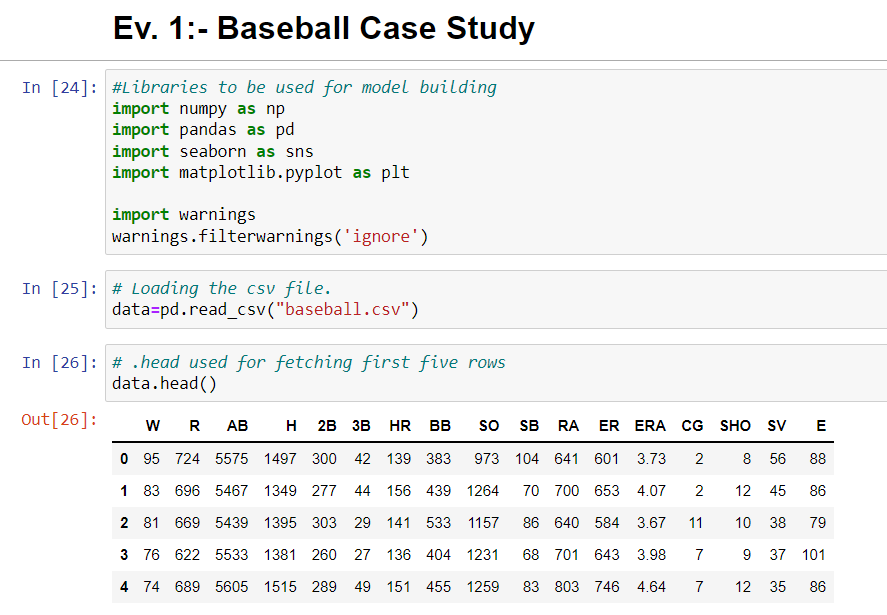
**2. ANALYSING DATASET:-**

Let’s first import the dataset. We have features in the dataset. We have to observe and analyse the data. Although, as a Data Scientist we must procure lot of information regarding the subject which can help us build a model which can provide us best accuracy score. But there can be some columns which may or may not affect our model prediction, we must observe those columns closely and if required, we must drop them also, for maintaining efficiency and accuracy.

For observing the Target column “W” and checking the data type of target column to decide which kind of problem algorithms we have to implement in that, suitable parameters we have to go through also.

After observing the dataset, we got to know that our target column “W” is a numerical datatype column, so this is a Regression Problem and we have to use all regression algorithms while building our model.

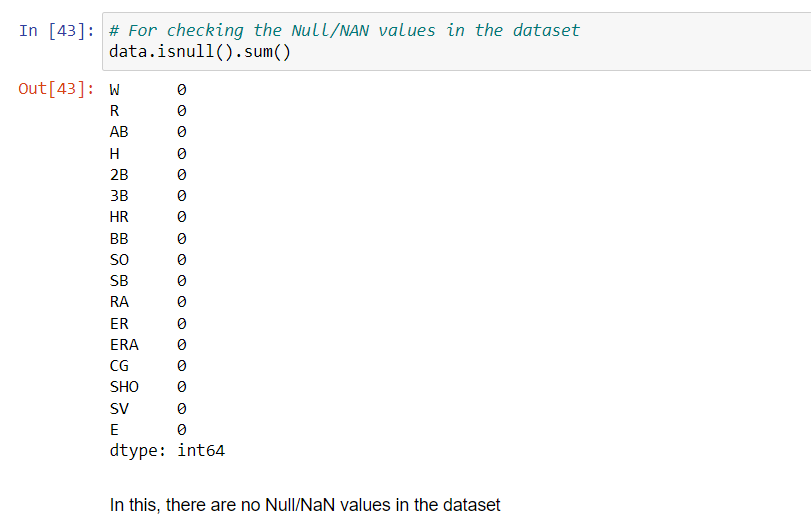
Now, for importing dataset and fetching records from it:-



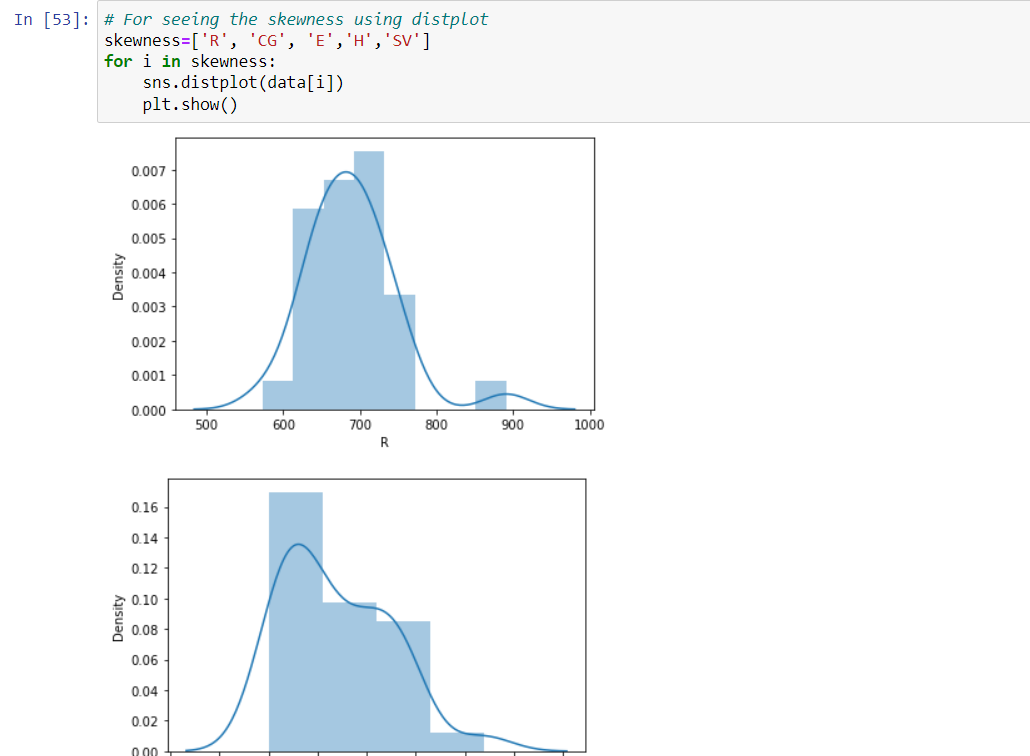
**Data Preparation and Cleaning:-**

* Firstly, we need to do some analysis like checking shape, nunique, value counts, info, datatype etc.
* After looking and observing the value counts if we find any unnecessary columns in the dataset, we can drop those columns.
* Presently I found some columns which needs to be removed:-data1.drop(["RA", "ER","ERA"], axis=1, inplace=True)

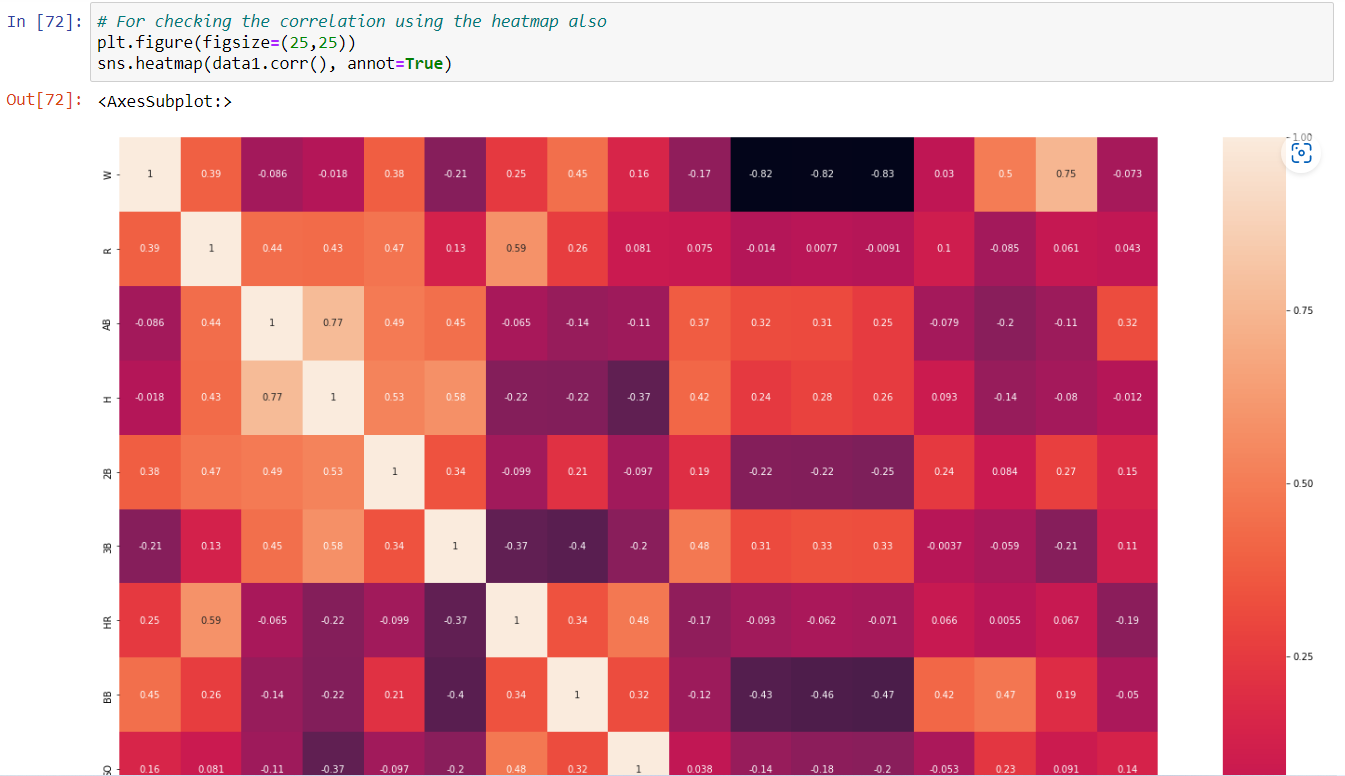
**For checking the Null values in the dataset:-**

****

**For checking the skewness levels:-**



**For checking the correlation among the dataset:-**



**3.EDA Conclusions:**

* I have checked for NaN values and I found there was no NaN /missing values in the dataset.
* I have extracted the necessary features from existing features to get better accuracy and dropped the old columns to avoid multicollinearity. If we will keep the old columns as it is, then they will act as duplicates in the model.
* I have also dropped the unnecessary columns.
* I have used both matplotlib and seaborn to visualize the data.
* To get better insight on the features, I have used distplot, barplot, regplot, stripplot and boxplot methods, since all columns were numerical I have used all numeric plots.

**For Checking and Removing Outliers:-**



* Now, for the removal of outliers,I have sued this below commands:-

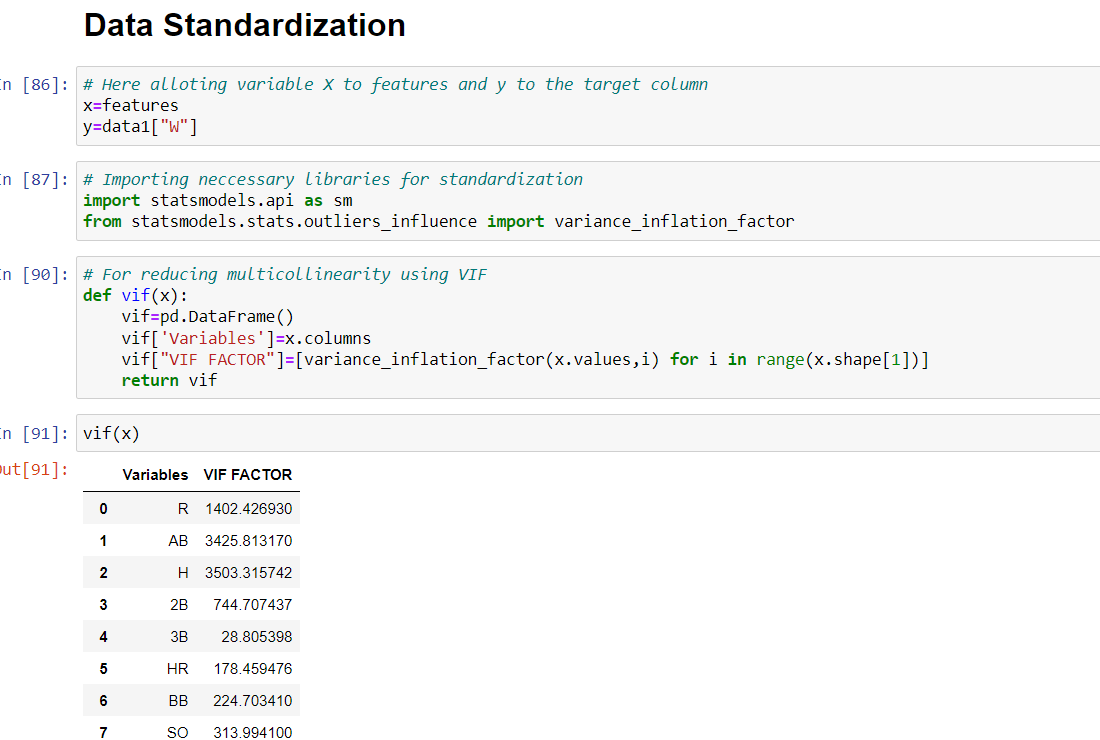
**from scipy.stats import zscore**

**zsc=np.abs(zscore(data))**

**threshold=3**

**print(np.where(zsc>3))**

**For Splitting the data and checking Multicollinearity issue:-**



* **Also, I have scaled the data using MinMax Scaler method**:-

**from sklearn.preprocessing import MinMaxScaler**

**Sca=MinMaxScaler()**

**Sca.fit(data1)**

**Sd=Sca.transform(data1)**

**Sd**

**USED PCA FOR DIMENSION REDUCTIONS:-**

from sklearn.decomposition import PCA

pca=PCA(n\_components=7)

X1=Sd[:,1:]

Y1=Sd[:,0]

pca.fit(X1)

x2=pca.transform(X1)

dataf=pd.DataFrame(data=x2)

dataf

**4.Preprocessing:-**

As a first step I have separated the dependent and independent features.

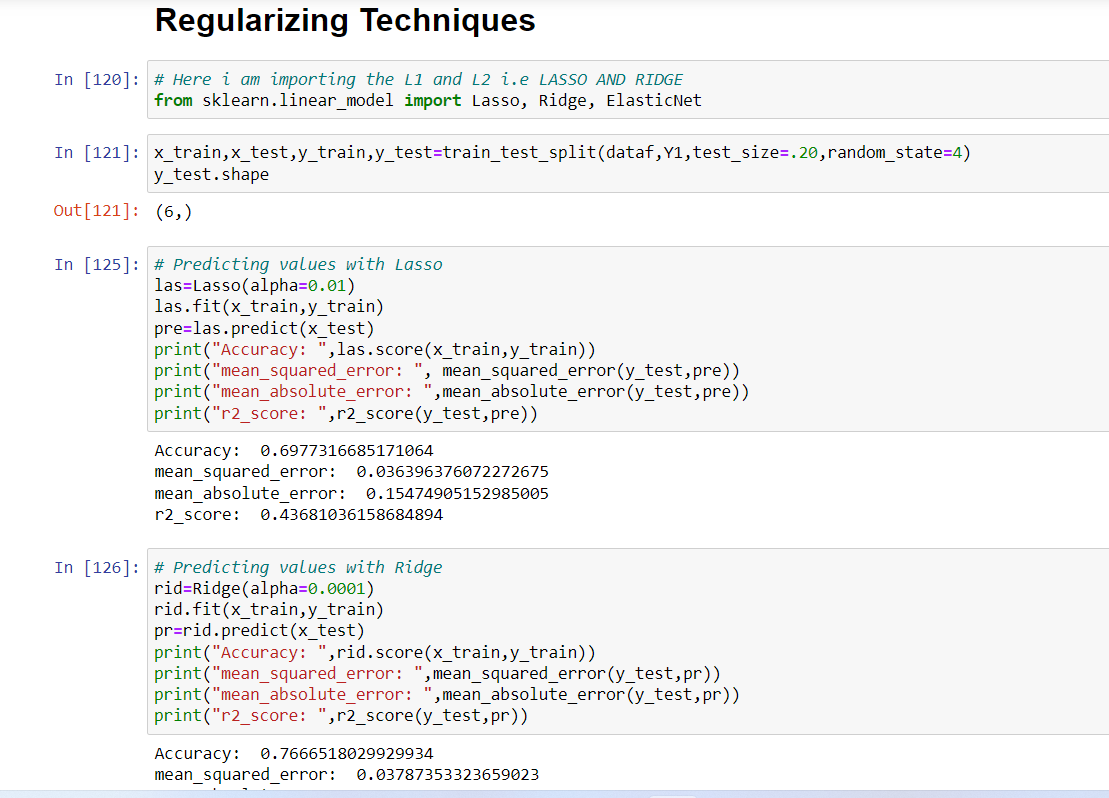
* I have taken x as features and y as dependent/ Target feature.
* Then scaling will be done on my features to get the same range in all the columns. If I don’t scale my independent columns, then there is a chance that my model may get biased. So In this particular case I have used Standard scaler and MinMax Scaler as I have removed all outliers and skewness from the dataset it is good to use standard scaler.

**5. Building ML Models:-**

1. **For checking R2 score with Linear Regression:-**



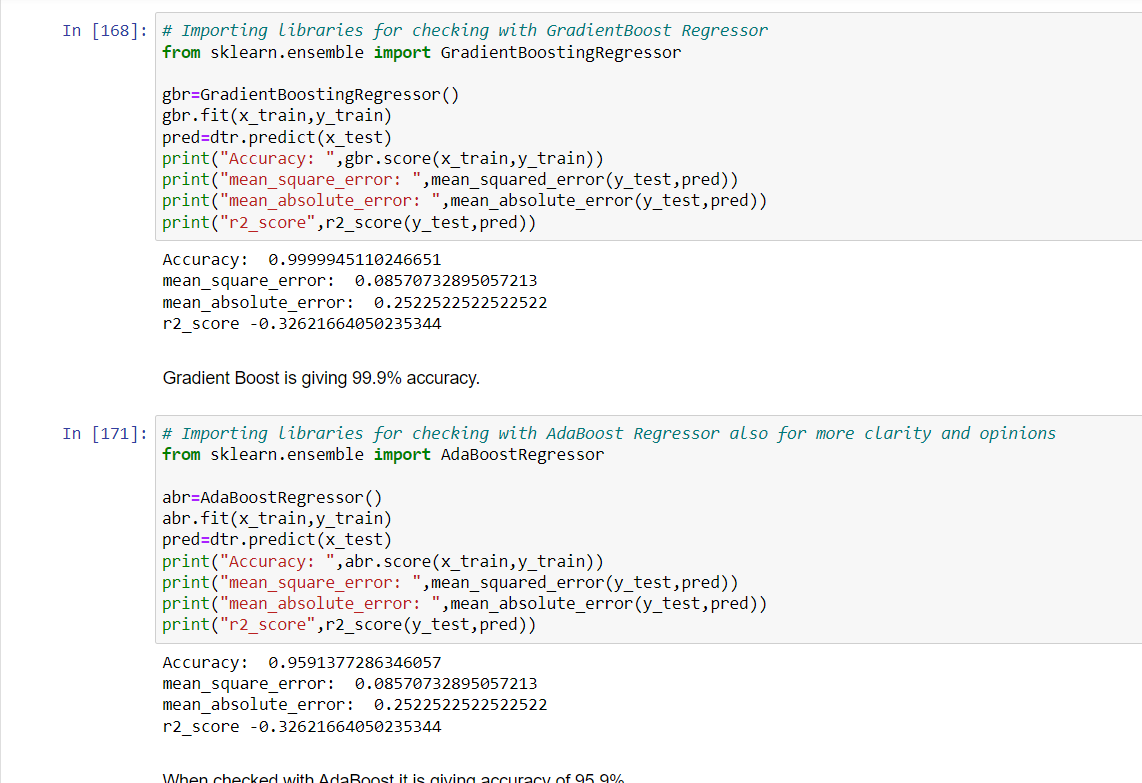
**Regularization Techniques:-**



**Now, checking R2 score with other Algorithms:-**

* **Random Forest Regressor.**
* **Decision Tree Regressor.**
* **Gradient Boosting Regressor.**
* **AdaBoost Regressor.**





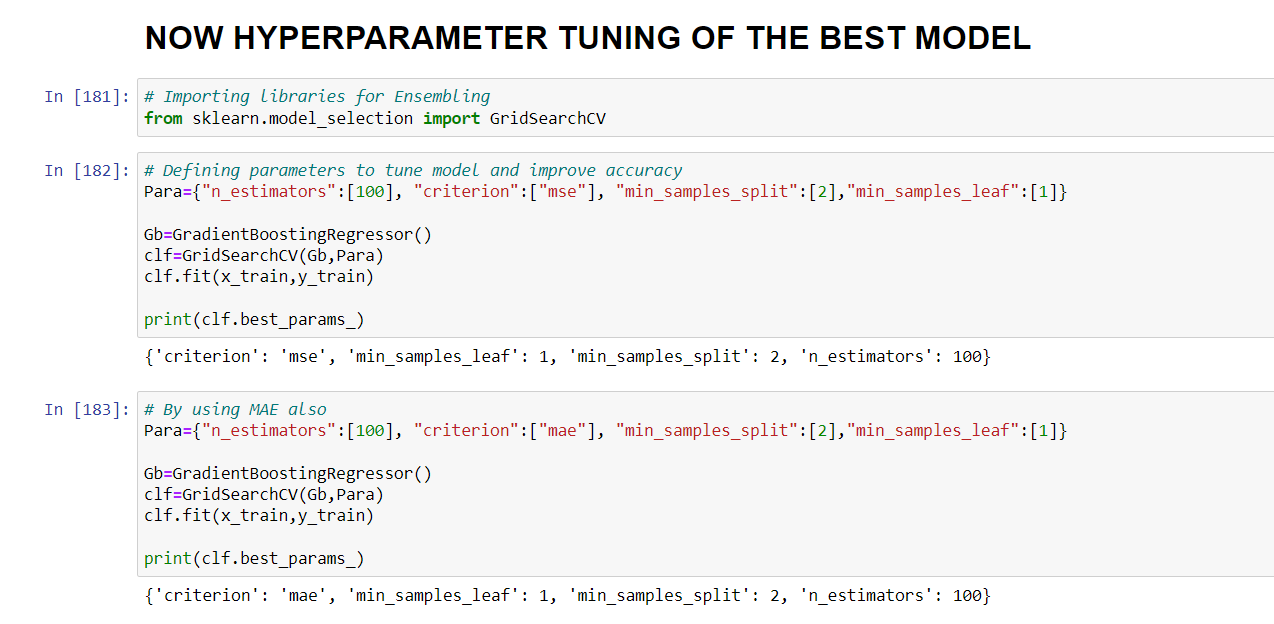
**CROSS VALIDATION PHASE:-**

**Here, I have checked Cv score of all the above mentioned models as well:-**

from sklearn.model\_selection import cross\_val\_score

* print(cross\_val\_score(lr,x,y,cv=5).mean())
* print(cross\_val\_score(dtr,x,y,cv=5).mean())
* print(cross\_val\_score(rf,x,y,cv=5).mean())
* print(cross\_val\_score(gbr,x,y,cv=5).mean())
* print(cross\_val\_score(abr,x,y,cv=5).mean())

**HYPER PARAMETER TUNING:-**

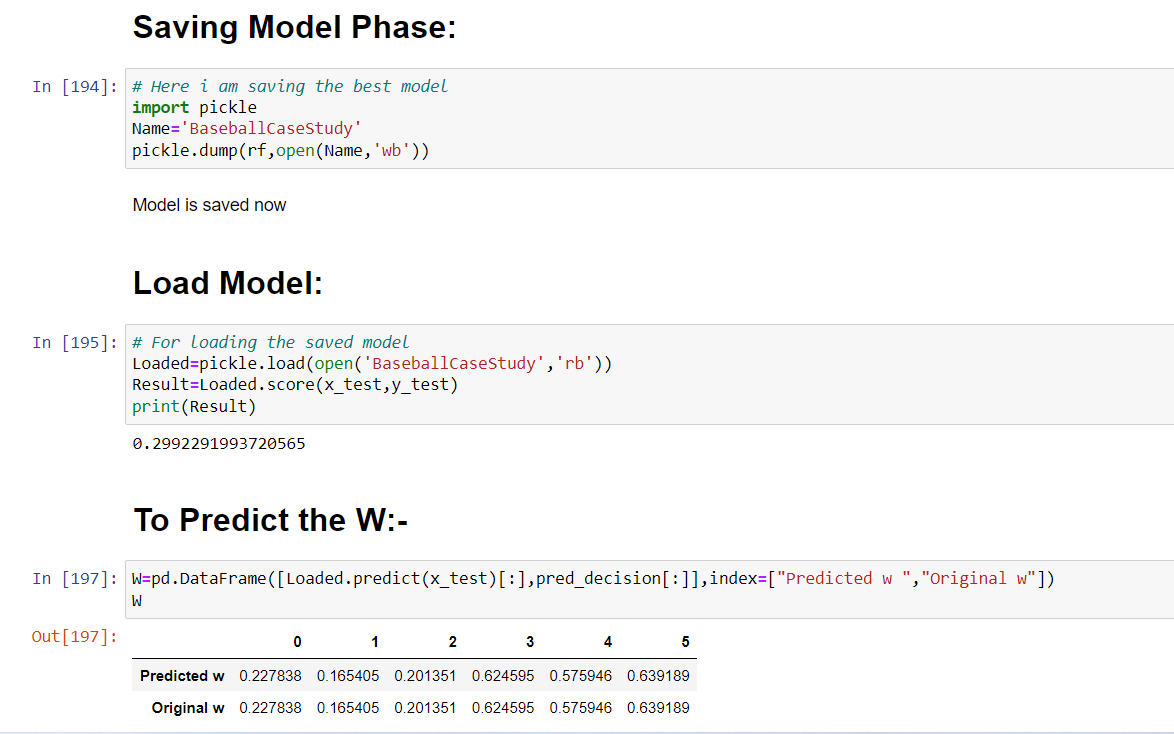


After this, my model’s R2 Score has been increased to 29.9 % as seen below:-



* After tuning the model, the R2 score is improved by almost 1.5 % which means that the default parameters used by the model were giving the best accuracy, And the model is now ready with best score performance ahead.

**Now, after all observations and findings Saving the Model and Loading Predictions:-**



**6. Conclusion:-**

* This particular problem needs a good understanding of data, and in this problem, Feature Engineering is the one of the most important step which actually helped in our model performance.
* Also, we can find the way we have handled numerical data and also how we build different machine learning Algorithms on the same dataset.
* Using Hyper Parameter Tuning, we have tried to improve our best model’s R2 Score, which to an extent we achieved after hyperparameter tuning our final model showed improvement of almost 1.5 %.
* So, With the help of this Machine Learning Model, I am able to predict the number of Wins and the actual and predicted values also at the time of Loading Model.

Thanks