***Baseball Case Study Article***

1. **Problem definition**

This dataset utilizes data from 2014 Major League Baseball seasons in order to develop an algorithm that **predicts the number of wins** for a given team in the 2015 season based on several different indicators of success. There are **16 different features that will be used** **as the inputs** to the machine learning and the **output will be a value that represents the number of wins**.

1. **Data Analysis**

Totally there are 16 different features in this dataset as input. **W(Wins)** which is present in the first column in the image below is the **label**.

It is continuous data present in the label. So we have to build a **regression algorithm model**.

We use the Wikipedia link **(https://en.wikipedia.org/wiki/Baseball\_statistics)** to know the actual meaning of the short forms present in the features headings.

I will let you know few of the important features full forms below.

\***R** - Runs scored: times reached home plate legally and safely

\***2B** - [Double](https://en.wikipedia.org/wiki/Double_(baseball)): hits on which the batter reaches second base safely without the contribution of a fielding error

\***3B** - [Triple](https://en.wikipedia.org/wiki/Triple_(baseball)): hits on which the batter reaches third base safely without the contribution of a fielding error

\***HR** - [Home runs](https://en.wikipedia.org/wiki/Home_run): hits on which the batter successfully touched all four bases, without the contribution of a fielding error

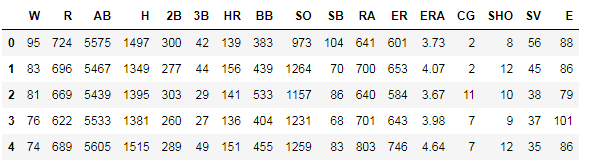
\***RA** - [Run average](https://en.wikipedia.org/wiki/Run_average): number of runs allowed times nine divided by innings pitched

\***ER** - [Earned run](https://en.wikipedia.org/wiki/Earned_run): number of runs that did not occur as a result of errors or passed balls

\* **SHO** - [Shutout](https://en.wikipedia.org/wiki/Shutout_(baseball)): number of complete games pitched with no runs allowed

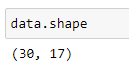
\* **E** - [Errors](https://en.wikipedia.org/wiki/Error_(baseball_statistics)): number of times a fielder fails to make a play he should have made with common effort, and the offense benefits as a result

\*\*Please visit the link mentioned above for all the full forms present for each feature.

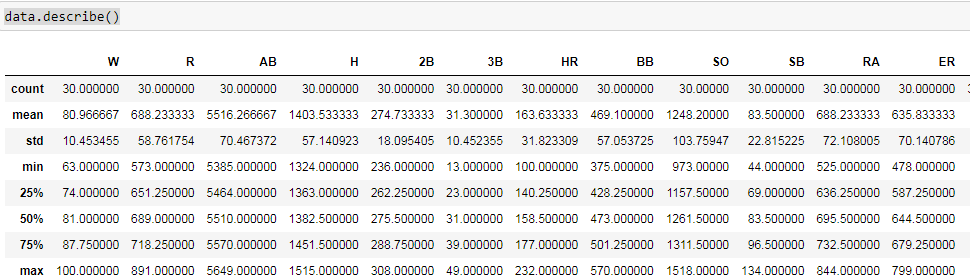


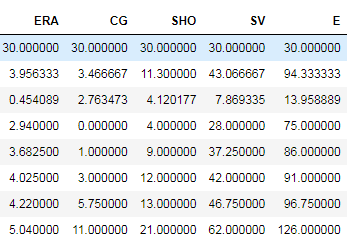
We also see that all the columns present are in **numerical values**, so we **need not use any encoding techniques** to convert the data into machine learning language.

The shape of the dataset is that we have **30 rows and 17 columns in total**.



When we describe the dataset, we find out that there are **no null values** in any of the columns.





The mean and standard deviations for all of the columns also looks decent enough and now we can go ahead for more of visualizations and EDA.

1. **EDA**

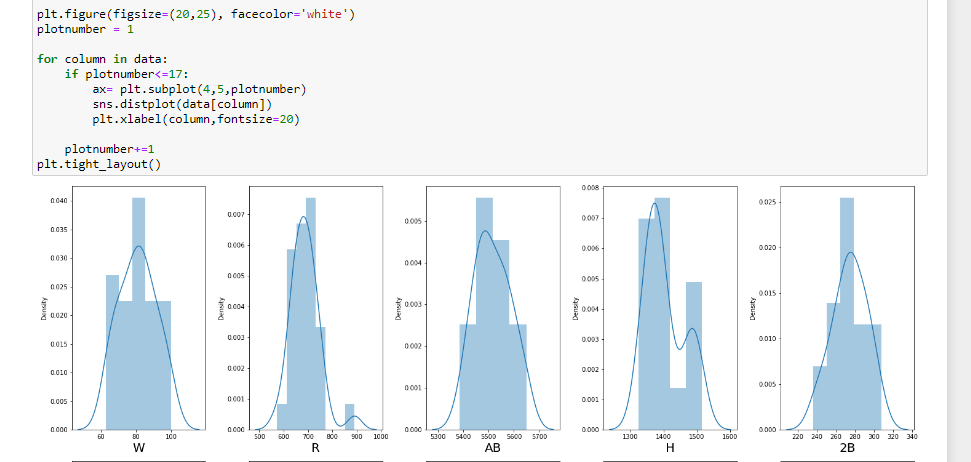
**We will do two types of analysis on the dataset. First we will have a Univariate analysis** **on the dataset and later we will go for bivariate analysis**.

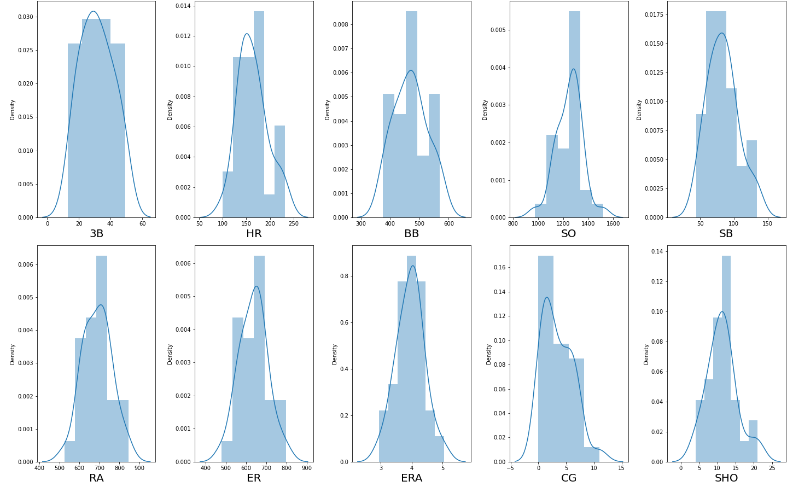
**Univariate analysis:** So now we plot graphs for all the features to dig deep into the visualization part and see how our data is distributed.

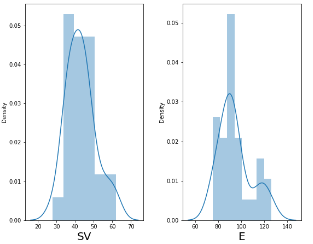
We use **distribution plots** to see if the graphs are normally distributed for each of the feature.

We will see if the graphs are bell shaped curve and we will get to know if we have **outliers** also present in the dataset.

If the graphs are **right skewed or left skewed, it means that there are outliers present in the dataset**. So we will plot the graphs and it is shown below.







So we can conclude on the above visualization that **there isn’t skewness in any of the features present.**

And we see that all the **graphs are normally distributed and there is no outliers present in our dataset**.

The **distribution plot looks decent enough** and we can go ahead and proceed for further processing of the machine learning model.

Before that we will do our **Bivariate analysis**:

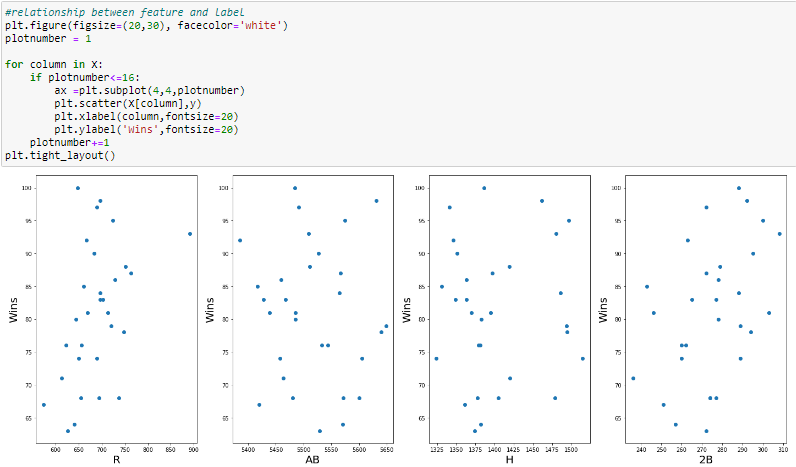
We will use **scatter plot** here to distinguish any relationship between the features and the label.

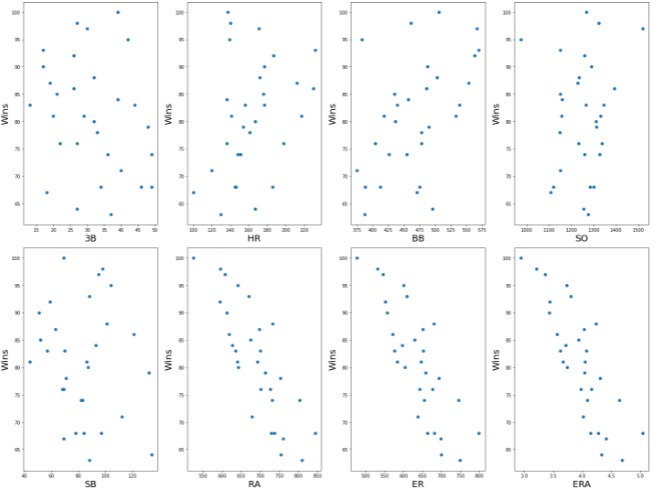
First we will divide the data into two variables X and y.

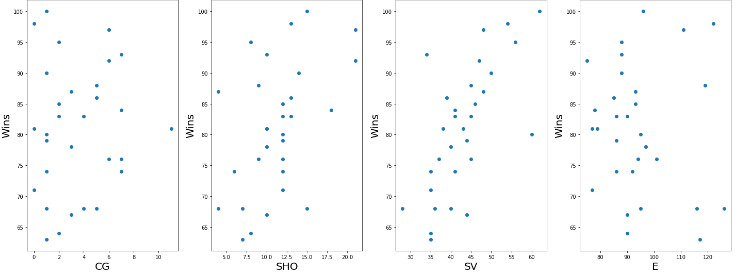


**X is the feature data and y is the label data.**

So we will apply **scatter plot** to the dataset and check the relationship between the feature data and label data. Visualising the plots, we can come to a conclusion on the features selection and will be useful for further processing and building the machine learning model.







So we have the above scatterplot which visualises the relationship between the features and that of the label. We see that the **features RA, ER and ERA plot against the wins shows similar trends.**

There seems to be **collinearity between some of the features present in this dataset**. So we will go ahead check on multi colleanirity in the preprocessing stage.

1. **Pre-processing Pipeline**

Now, we need to do a little bit of pre-processing before we start building and training our model.

As our EDA showed us that there is a possibility of having multi colleanirity between the features.

We will use the **Variance inflation factor** method and check the values for each of the feature and see whether we can drop any of the unwanted columns which has multi collinearity.

Variance inflation factor gives us values for each feature by comparing each of the feature against another feature for the whole dataset.

Some take 5 as the standard value and some take 10 as the standard value. **If the vif values is less than 5 or 10** as to how one understands the data, then they go ahead with building the machine learning model.

Before we apply the vif, we have to scale our data using standard scaler.

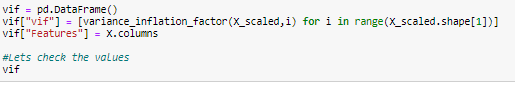
Standardization is another **scaling technique** where the **values are centered around the mean with a unit standard deviation**. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

We can import standard scaler using **“from sklearn.preprocessing import StandardScaler”**



The above expression will transform the given data into standard data for further processing.

So now we will go ahead and import vif using **“from statsmodels.stats.outliers\_influence import variance\_inflation\_factor”**

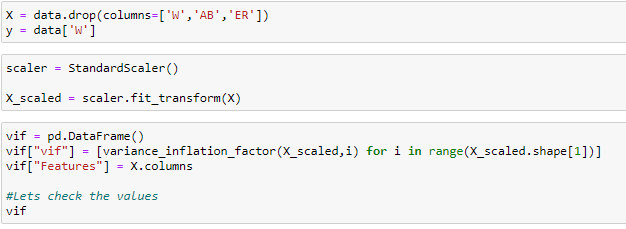


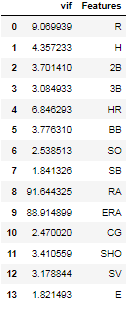
**We will get the vif values as below image**



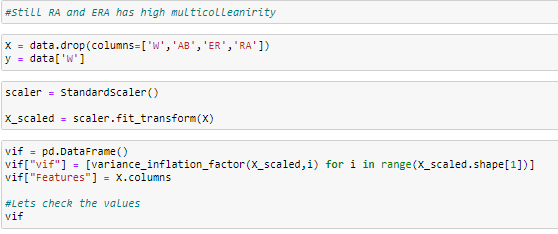
So as we can see above, few of the values are so high than that of the standard, so we will go ahead and drop the features which has highest values and check the vif values again.

So I dropped 2 features here that is **ER and AB**. So now I will scale the data again and check the **new updated vif values**.





Here we see a **much better vif values than before**. But again we have **RA and ERA with high vif value.** So we will **drop one that has highest vif value** of the two that is RA. And check the vif value again



So now we check the vif values and we get as shown below.



So we have **much better vif values** and we have **removed 3 columns which are multicollinear** and we are **done with the preprocessing stage**.

All the features present can be fed into our machine learning model which will be used in predicting our label.

1. **Building Machine Learning Models**

So our preprocessing stage is completed. Now we go ahead and start training our model.

Initially we will divide the data set into train and test data. 25% of the data is used as test data in my model.



And since this is a regression model, I have used linear regression model first up to train the data. Linear regression model can be imported **“from sklearn.linear\_model import LinearRegression”**



So we have trained our model here using linear regression by giving 75% data as training data and 25% as test data.

We will check the regression score of our training dataset.

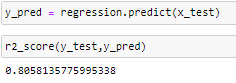


We have a score of 95% on our training dataset. Now lets check on test data.



Here we have a score of 80% on our test dataset which is good for our model.

**To evaluate the performance of our linear regression model, we have R^2 value. It is the amount of the variation in the output dependent attribute which is predictable from the input independent variable(s).**



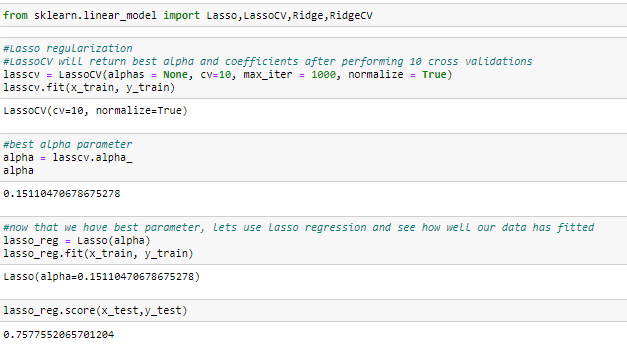
**So we have R2 value as 80.5% which is really good for our model.**

**We have built our linear regression model above which has a R2 score of 80.5%.**

* **Regularization**

We will use regularization method and apply in this linear regression model and check if our model can be modified for a better result.

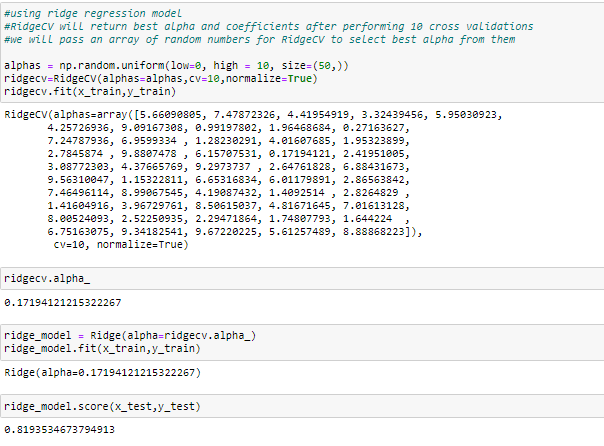
2 types of regularization we have, one is **LASSO** and another is **RIDGE.**



Here we have imported the **lassoCV** and applied on our test data, we have trained the lasso model and our score has been reduced than our linear regression model. We have our alpha value also in the above image.

We have **LASSO regression score as 75.77%.**

So now we train for our RIDGE regression model and check if we can get a better score than our linear regression model.



So we have our **RIDGE model score as 81.9%** which is much better than our linear regression model. So Regularization has helped in our model and our test score of our dataset has been increased.

1. **Concluding Remarks**

**So I conclude by saying that we have built a linear regression model and Ridge regression model with scores of around 80% and 81% respectively. This model is now capable of predicting the number of wins for a given team in the 2015 season based on several different indicators of success as per the problem statement.**

**Our model is capable of giving the accuracy of 80% which is really good. Problem statement has been achieved in our model.**

**Thank you,**

**Akshay D**