Baseball Wins Prediction with Machine Learning Models

horizontal line

# 

# Introduction

**Problem Definition:**

Major League Baseball (MLB) is regarded as the highest level of professional baseball in the world. It is also considered one of the most popular international sporting events. A lot of research work has been conducted on constructing models for predicting the outcome of MLB matches. The accuracy in predicting the results of baseball games dependents greatly on the size of available datasets. Therefore, models built using machine learning methods are useful for predicting the outcomes (win/loss) of MLB matches. It is also very important to compare the differences between the models with respect to their performance. In this project, the match data of 30 teams from the 2014 MLB season is utilised for building machine learning models that would predict the number of wins for a given team in the following MLB season. Based on the outcome of comparing the prediction accuracies of the models, the best one from among them will be finally picked and tuned further to improve its prediction accuracy.

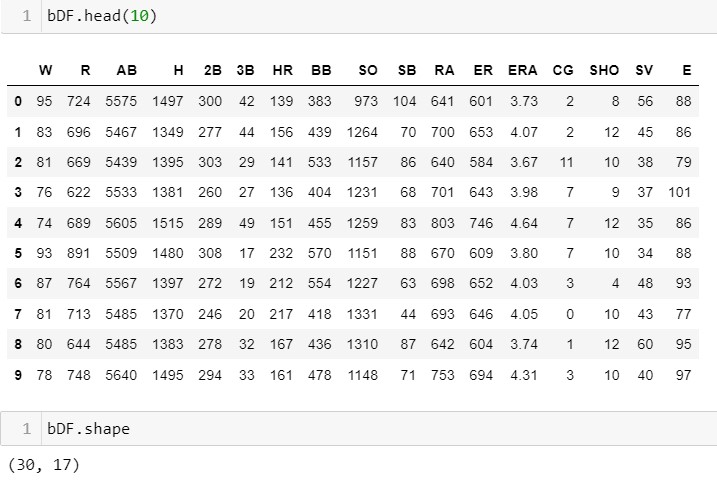
**Executive Summary:**

In this project, a dataset was provided with the details regarding team performance and various batting, pitching and baserunning statistics from 2014 Major League Baseball season. The task is 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.

The Dataset was first cleaned, the various feature columns were analysed, and then based on strength of correlation and ANOVA f-score values, the feature columns were selected that would best predict the Target variable, to train and test machine learning models.

The Baseball dataset from MLB 2014 was worked with to build a predictive model that best predicts the number of wins for a team in the 2015 season of MLB. Several regression models were trained and fitted with a part of the dataset and then tested with a different part of the dataset. The model that performed with the best prediction accuracy, lowest root mean squared error, and best cross validation score was then selected and tuned further with hyper parameter tuning techniques.

**About the Dataset:**

****

The given dataset consists of 17 columns and 30 rows.

This dataset utilizes data from 2014 Major League Baseball season 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.

**The Independent Feature columns are:**

**Runs R**: number of times a player crosses home plate

**At Bats AB**: plate appearances, not including bases on balls, being hit by pitch, sacrifices, interference, or obstruction

**Hits**: reaching base because of a batted, fair ball without error by the defence

**Doubles**: hits on which the batter reaches second base safely without the contribution of a fielding error

**Triples**: hits on which the batter reaches third base safely without the contribution of a fielding error

**Homeruns**: hits on which the batter successfully touched all four bases, without the contribution of a fielding error

**Walks**: times pitching four balls, allowing the batter to take first base / hitter not swinging at four pitches called out of the strike zone and awarded first base.

**Strikeouts**: number of batters who received strike three

**Stolen Bases**: number of bases advanced by the runner while the ball is in the possession of the defense

**Runs Allowed**: the number of runs scored against a pitcher. This includes earned runs and unearned runs.

**Earned Runs**: number of runs that did not occur as a result of errors or passed balls

**Earned Run Average (ERA)**: the average number of earned runs allowed by a pitcher per nine innings

**Shutouts**: number of complete games pitched with no runs allowed

**Saves**: Number of games where the pitcher enters a game led by the pitcher's team, finishes the game without surrendering the lead, is not the winning pitcher, and either (a) the lead was three runs or fewer when the pitcher entered the game; (b) the potential tying run was on base, at bat, or on deck; or (c) the pitcher pitched three or more innings

**Complete Games**: number of games where player was the only pitcher for their team

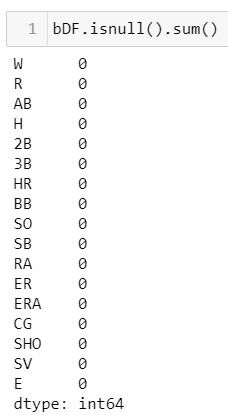
**Errors**: number of times a fielder fails to make a play he should have made with common effort, and the offense benefits as a result

**The Target Variable to predict is given in the column:**

**W:** Number of predicted wins

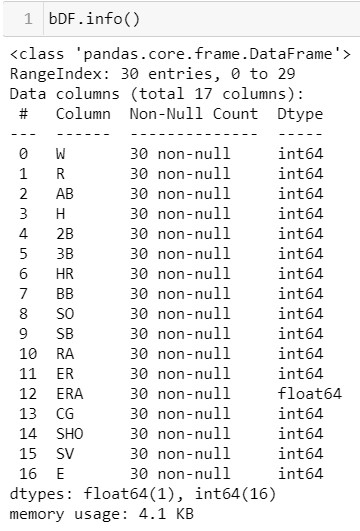
**Data Cleaning:**

Upon inspecting all the columns in the dataframe, it is observed there are no null values / values missing from any of the columns in the dataframe..



**Exploratory Data Analysis**

**Getting the basic summary and statistical information of the data.**

****

All columns contain continuous type of data

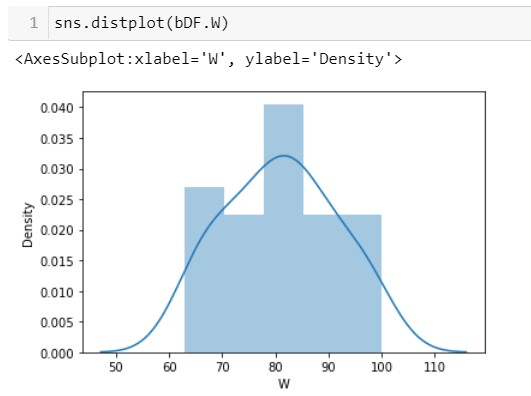


mean and 50% of all columns are similar. difference between 75% and max in columns like E,SV,SHO,SB etc is considerable indicating presence of outliers.

**This is a Regression Problem since the Target variable / Label column ("W") has Continuous type of Data.**

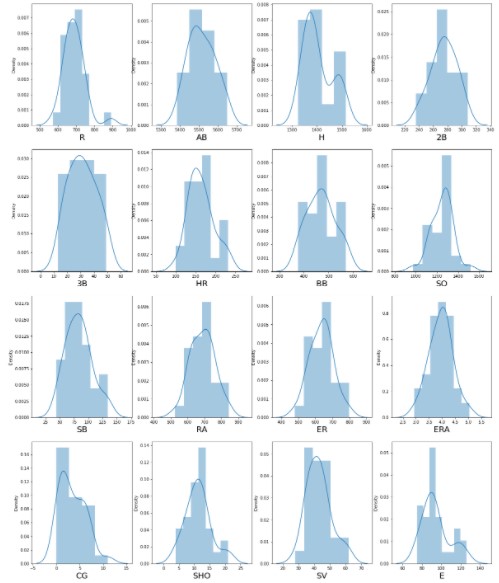
**Univariate Analysis**

**Analyzing the Target Variable**

****

From the graph above it is observed that the W data forms a continuous Normal distribution with mean of 80.966.

**Analyzing the Feature Columns**

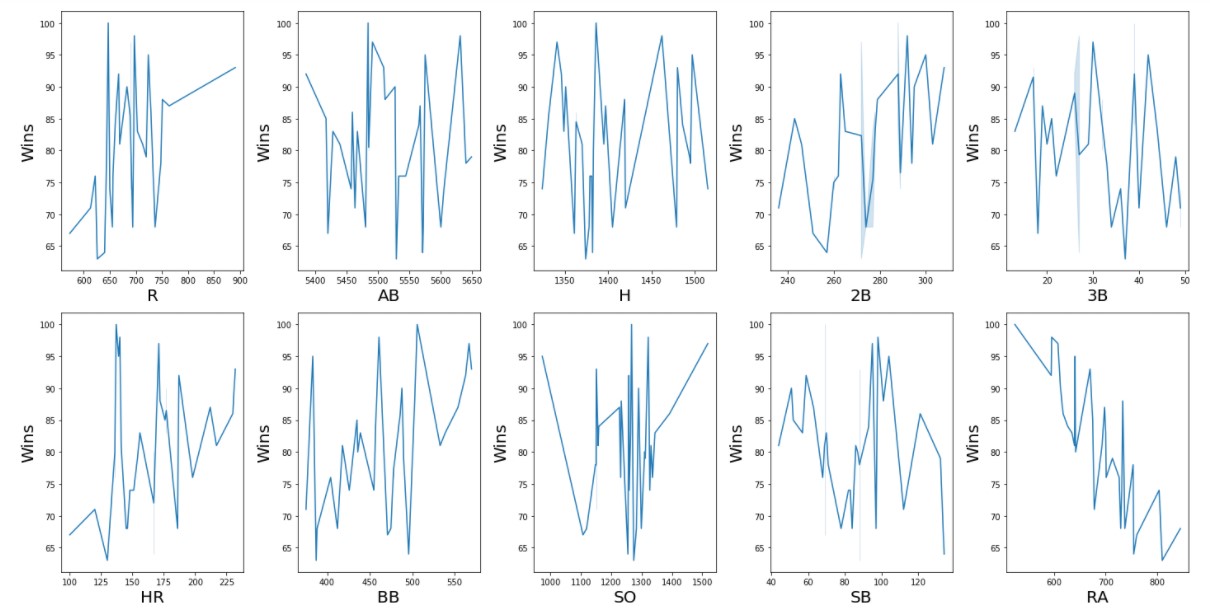
****

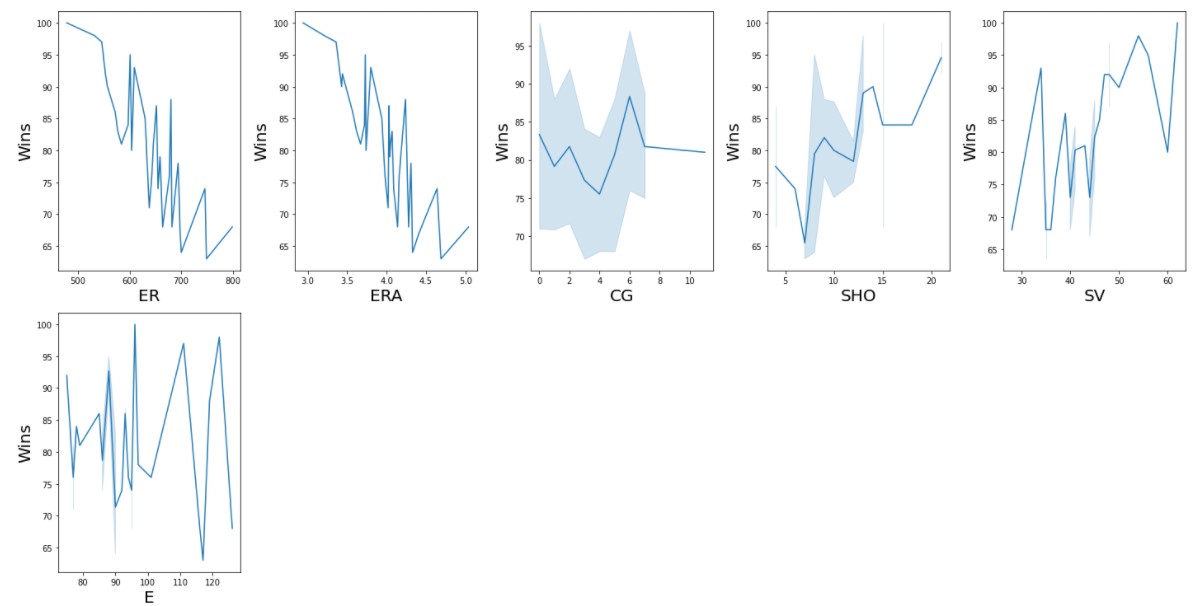
**Upon analyzing the Feature Columns, following observations are made:**

* It can be observed from above graphs that data is mostly normally distributed.
* Data in columns like R, CG, E, SV,H are skewed.

**Bivariate Analysis**

### Interpreting Relationship between Dependent Variable and Independent Variables

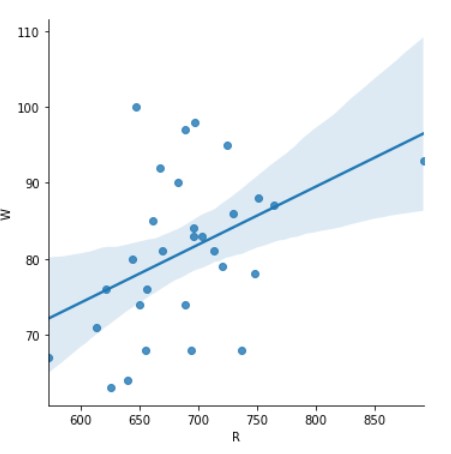
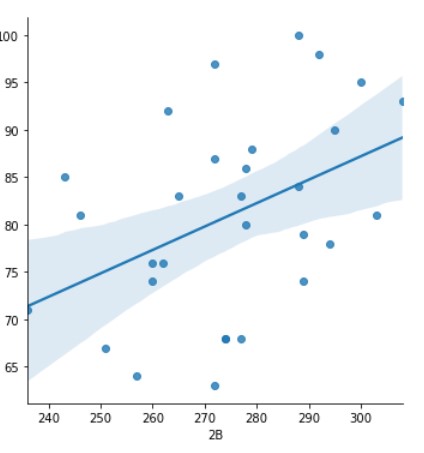


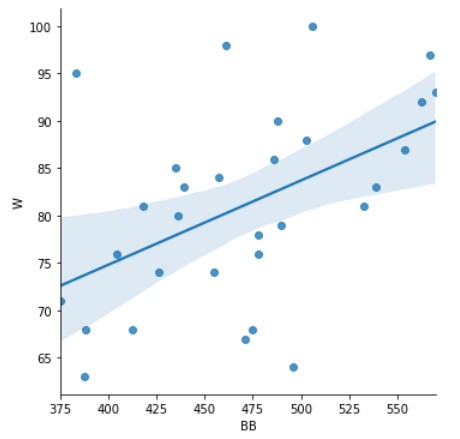
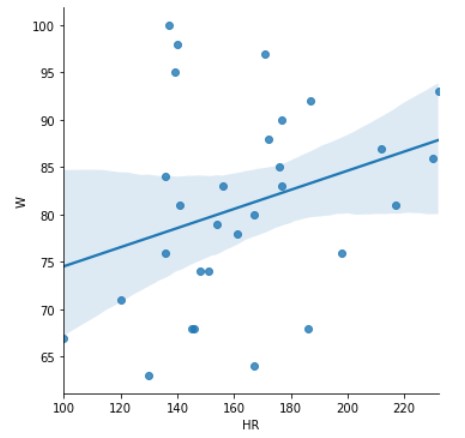


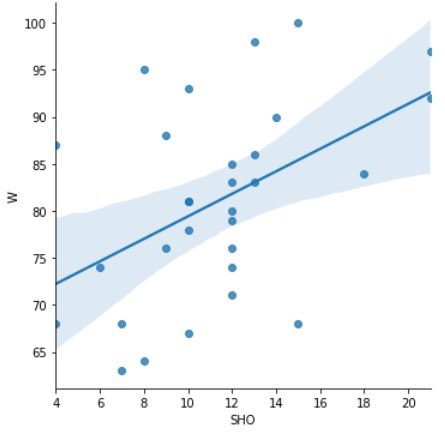
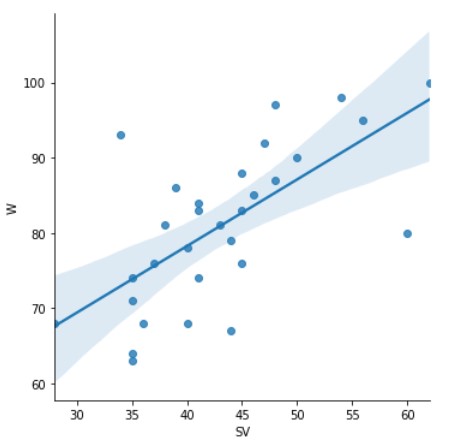
**Following observations can be made from above graphs:**

* It can be observed that Runs have a positive linear relationship with Win.
* Doubles have a positive linear relationship with Win.
* Homeruns have a positive linear relationship with Win
* Base on balls has a positive linear relationship with Win
* Save has a positive linear relationship with Win
* Shutouts have a positive linear relationship with Win
* Runs on Average have a negative linear relationship with W
* Earned Runs have a negative linear relationship with W
* Earned Runs Average has a negative linear relationship with W

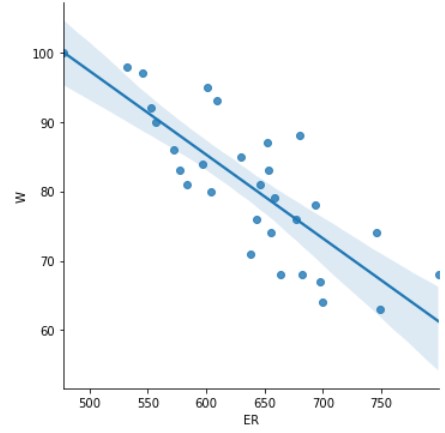
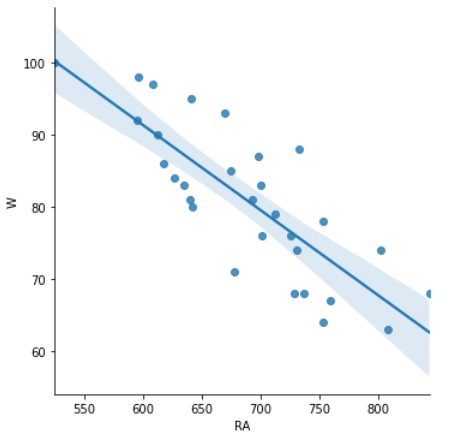
**These relations can be better understood when visualized using lmplots**

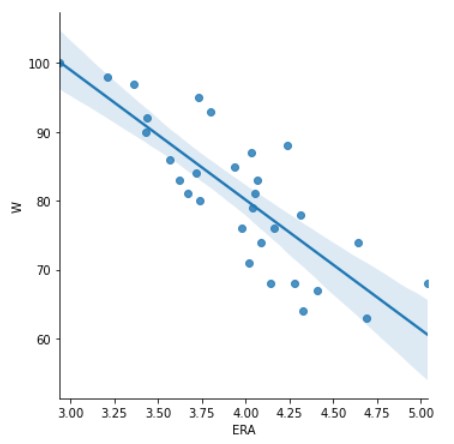
**Positive Relationships:  
 **

****

****

**Negative Relationships:**

****

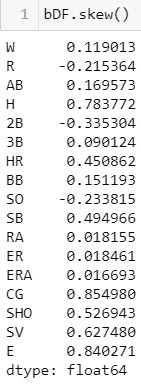
****

### Checking for Outliers in Features.

****

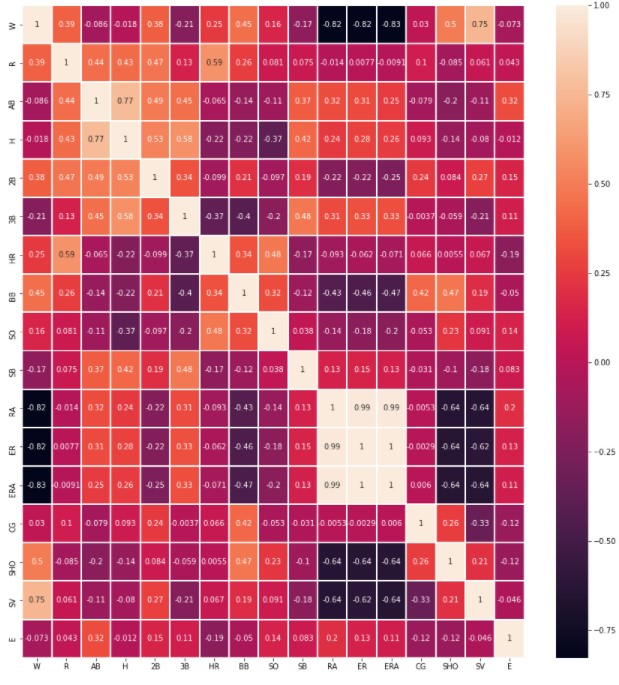
From the above boxplots, it is observed that Columns like SHO, SV, ERA and E have outliers present. Outliers, although not many in number, still greatly impact the skewness of the data distributions and therefore need to be removed.

**Checking for skewness in data distributions**

****

There is moderate skewness in E, CG, H, and SV. Rest of the Data distributions are symmetric.

**Finding the correlations**

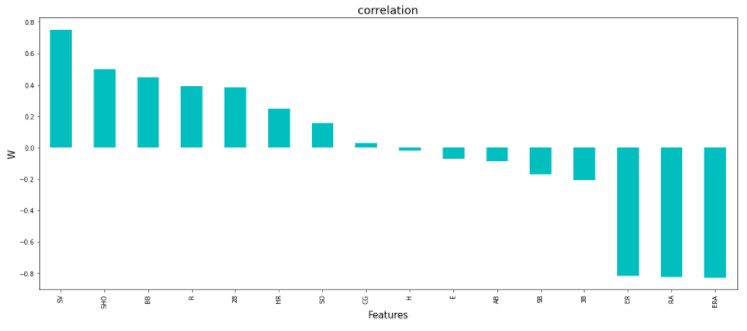
****

From the above heat map of the correlations between the columns of the dataframe,

It is observed that there is a very high correlation between features ER,ERA,RA.

ERA is calculated using the formula: ER\*9/Innings pitched, factors like 'Innings pitched' are not available as columns in the data. This clearly explains why ERA and ER are correlated.

**Visualizing correlation of feature columns with label column.**

****

From analysing the graph above, it is observed that SV has the highest positive correlation with W followed by SHO and BB. While, ER,ERA and RA have the highest negative correlation wiht W. H has the weakest correlation with W.

**Data Pre-Processing**

**Checking for Outliers in columns**

****

It is observed that Columns like SHO, SV, ERA and E have outliers present.

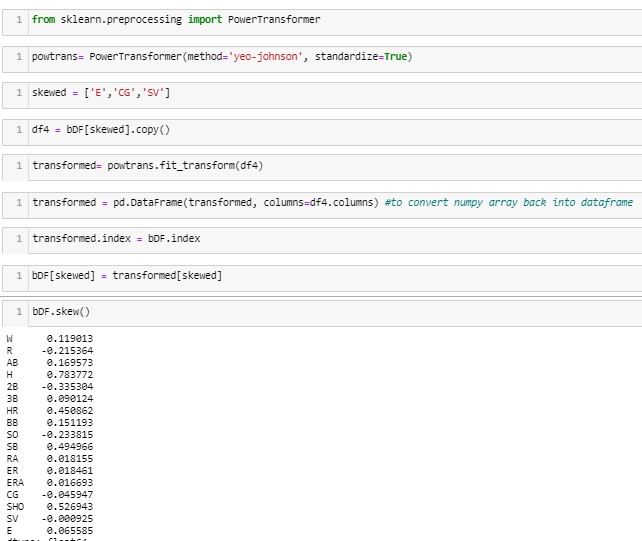
The method used here for outlier removal is the Z score method.

The outcome of outlier removal using the Z score method was a significant reduction in the presence of outliers in the feature columns. However, in the process, the dataset lost 3% of the total data originally available. Fortunately, this loss would not have any impact on the training and testing of the models nor their final prediction accuracy.

### Normalizing Data Distribution using Power Transformer

The Yeo-Johnson power transformer method is used to transform the values of the columns whose data distributions are skewed. The optimal parameter for stabilizing variance and minimizing skewness is estimated through maximum likelihood.

Using the code below the data distribution was normalised.



It is observed that Skewness has been greatly reduced.

Next step is to select the best features which would build the most accurate Machine Learning Models to predict the target variable.

### Checking for Multicollinearity using Variance Inflation Factor

Variance inflation factor measures how much the variance of an independent variable is influenced / inflated, by its interaction/correlation with other independent variables. Variance inflation factors allow a quick measure of how much a variable is contributing to the standard error in the regression.

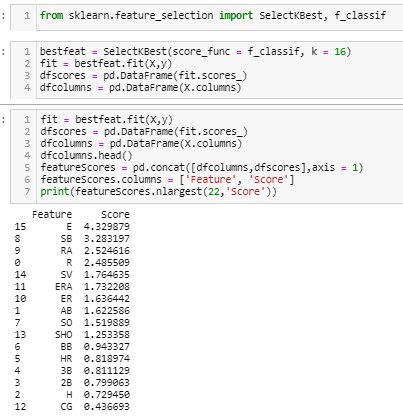


It is found that R,ER and ERA have the highest multi collinearity amongst all the features.

### 

### Selecting Kbest Features

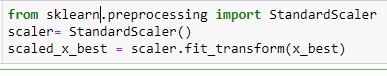
Based on the respective ANOVA f-score values, the feature columns are selected that would best predict the Target variable, to train and test machine learning models.



Upon analyzing the scores of each column, it is decided that the columns with the lowest scores, as well as the highly collinear column 'ERA' will be dropped.

**Feature scaling**

Scaling the values in the feature columns using StandardScaler inorder to normalize the range of data.



### 

**Regression Model Building**

**Finding the Best Random State**

The best random state has to be determined, which will then decide the splitting of data into train and test indices in the most optimal way, that yields maximum model prediction accuracy.

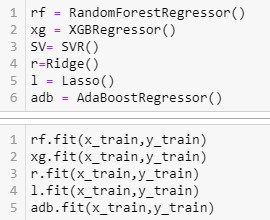


**Creating Train-Test split based on random state obtained above:**

****

### 

### Training the Models

****

**Analyzing Model Accuracies**

### Ridge Model Accuracy

The trained Ridge Regression Model shows

R2 score of 0.7400

Mean Squared Error of 17.929

Root Mean Squared Error of 4.234

Cross validation score of 0.5539

### Lasso Model Accuracy

The trained Lasso Regression Model shows

R2 score of 0.9535

Mean Squared Error of 3.2052

Root Mean Squared Error of 1.7903

Cross validation score of 0.7529

### Random Forest Regressor Model Accuracy

The trained Random Forest Regression Model shows

R2 score of 0.6880

Mean Squared Error of 21.515

Root Mean Squared Error of 4.6385

Cross validation score of 0.4709

**XGB Regression Model Accuracy**

The trained XGB Regression Model shows

R2 score of 0.6933

Mean Squared Error of 21.155

Root Mean Squared Error of 4.5995

Cross validation score of 0.3924

### AdaBoost Regression Model Accuracy

The trained AdaBoost Regressor Model shows

R2 score of 0.4975

Mean Squared Error of 34.6588

Root Mean Squared Error of 5.887

Cross validation score of 0.5049

Since, the dataset available to work with is extremely small, it is observed that most of the machine learning models have performed fairly poorly, except for Lasso which has displayed the best R2 score and cross validation score, along with having the lowest mean squared error. Lasso Model does shrinkage and variable selection simultaneously for better prediction and model interpretation and prevents model overfitting.

### Based on comparing Accuracy Score results with Cross Validation results, it is determined that Lasso is the best model. It also has the lowest Root Mean Squared Error score.[¶](http://localhost:8888/notebooks/InsuranceFraud_proj.ipynb#Based-on-the-above-graph-and-roc_auc_scores,XGB-Classifier-is-the-best-model-for-the-dataset,-with-AUC-=-0.97-and-roc_auc_score-=-0.9121)

### 

### Hyper Parameter Tuning

GridSearchCV is used for Hyper Parameter Tuning of the XGB Classifier model.

Based on the input parameter values and after fitting the train datasets,

The Lasso Regression Model was further tuned based on the parameter values yielded from GridsearchCV.

The Tuned Lasso Regression Model displayed an accuracy of 95.35%

### Concluding Remarks

In conclusion, Lasso Regression Model is able to correctly predict the number of wins for a team in the following MLB tournament with great accuracy.

The dataset had very limited data which is problematic as models show greater stability when the dataset is of a good size. Therefore, Lasso Model works best in this case as it has a penality factor to determine the total number of features to be retained, thereby preventing model overfitting to a great length. It gives best estimators that have lower variance. Therefore, this model has greater predicting power than all the other models. Using GridSearchCV the optimal penalty factor was determined which helped the model generalise the data samples with greater accuracy.