**Baseball Case Study**

*By Kunal chand*

**1. Problem Definition.**

This dataset utilizes data from the 2014 Major League Baseball seasons 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. 16 different features will be used as the inputs to the machine learning and the output will be a value that represents the number of wins.

**-- Input features:** Runs, At Bats, Hits, Doubles, Triples, Homeruns, Walks, Strikeouts, Stolen Bases, Runs Allowed, Earned Runs, Earned Run Average (ERA), Shutouts, Saves, Complete Games and Errors

**-- Output:** Number of predicted wins (W)

To understand the meaning of the column, follow the link given below to understand the baseball

Attribute Information:

1. R – Runs scored: number of times a player crosses home plate.

2. AB – At bat: plate appearances, not including bases on balls, being hit by pitch, sacrifices, interference, or obstruction.

3. H – Hit: reaching base because of a batted, fair ball without error by the defense.

4.2B – Double: hits on which the batter reaches second base safely without the contribution of a fielding error

5.3B – Triple: hits on which the batter reaches third base safely without the contribution of a fielding error

6. HR – Home runs: hits on which the batter successfully touched all four bases, without the contribution of a fielding error

7. BB – Base on balls (also called a "walk"): hitter not swinging at four pitches called out of the strike zone and awarded first base.

8. PA/SO – Plate appearances per strikeout: number of times a batter strikes out to their plate appearance

9. SB – Stolen base: number of bases advanced by the runner while the ball is in the possession of the defense

10. RA – Run average: number of runs allowed times nine divided by innings pitched

11. ER – Earned run: number of runs that did not occur as a result of errors or passed balls

12. ERA – Earned run average: total number of earned runs (see "ER" above), multiplied by 9, divided by innings pitched

13. CG – Complete game: number of games where the player was the only pitcher for their team

14. SHO – Shutout: number of complete games pitched with no runs allowed

15. SV – Save: 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

16. E – 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.

**2. Data Analysis**.

Data exploration is the first step in data analysis and typically involves summarizing the main characteristics of a data set, including its size, accuracy, initial patterns in the data, and other attributes. It is commonly conducted by data analysts using visual analytics tools, but it can also be done in more advanced statistical software, Python. Before it can analyze data collected by multiple data sources and stored in data warehouses, an organization must know how many cases are in a data set, what variables are included, how many missing values there are, and what general hypotheses the data is likely to support. An initial exploration of the data set can help answer these questions by familiarizing analysts with the data with

Which they are working. We divided the data 8:2 for Training and Testing purposes respectively.

**3. EDA Concluding Remark.**

As for any basic model building, we have to understand the type of target variable; the data of the target variable is continued or classified.

Data Analysis is always the difficult part, for better understanding different kinds of bar plots, distribution plots are created with the target Column for finding the insights of the dataset we have.

Analytical Modeling always starts with the target variable we have, and in that case, our target variable is W(wins), for that, we create some box plots with the target variable to understand which feature columns help to learn the model best and which feature columns reduce the accuracy of the model.

And after finding the relation and correlation with the target variable we choose either Regression Model or Classification Model. Here in this problem, our target feature column continues so we build our Machine Learning model on Regression.

# EDA Process:

In choosing the right features to feed into our model, we want to capture the input variables most strongly related to the target variable.

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 30 entries, 0 to 29

Data columns (total 17 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 W 30 non-null int64

1 R 30 non-null int64

2 AB 30 non-null int64

3 H 30 non-null int64

4 2B 30 non-null int64

5 3B 30 non-null int64

6 HR 30 non-null int64

7 BB 30 non-null int64

8 SO 30 non-null int64

9 SB 30 non-null int64

10 RA 30 non-null int64

11 ER 30 non-null int64

12 ERA 30 non-null float64

13 CG 30 non-null int64

14 SHO 30 non-null int64

15 SV 30 non-null int64

16 E 30 non-null int64

dtypes: float64(1), int64(16)

memory usage: 4.1 KB

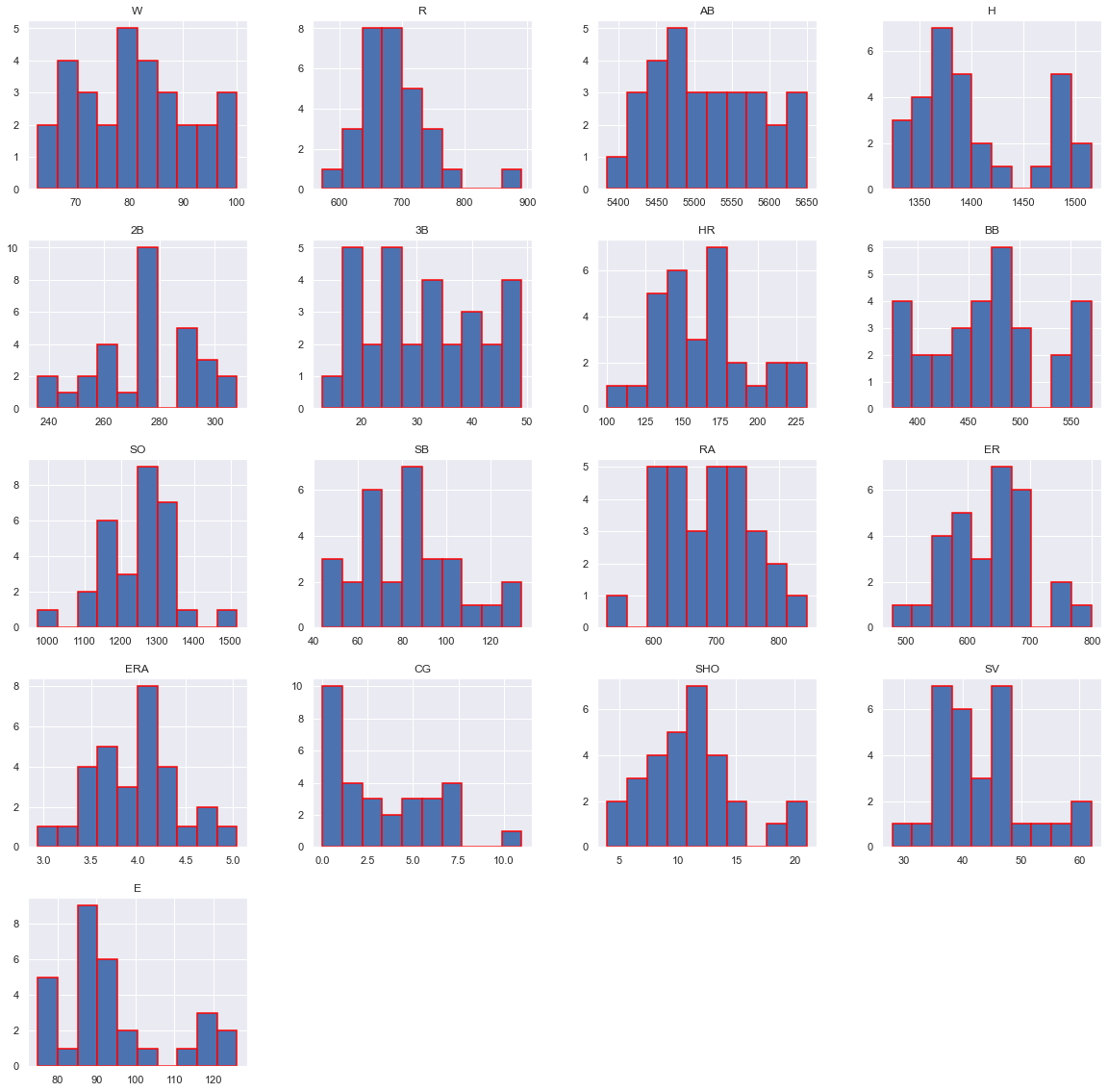
observations:

1. from the above code we can easily find that there are no nan values are present in the dataset.

**Visualization**

Data visualization is the process of translating large data sets and metrics into charts, graphs, and other visuals. The resulting visual representation of data makes it easier to identify and share real-time trends, outliers, and new insights about the information represented in the data

# Histograms of all the features columns



# CORRELATION BETWEEN THE COLUMNS:

# download (2).png

# download (3).png

# Correlation: From the above result it is clear that some columns make positive correlations and some make negative correlations.

Positive correlation columns are:

1. R

2. H

3.2B

4. HR

5. BB

6. SO

7. CG

8. SHO

9. SV

Negative correlation columns are:

1. AB

2.3B

3. SB

4. RA

5. ER

6. ERA

7. E

The positively correlated columns have a great impact on the target column while the negatively correlated have less or zero impact on the target column.

# Plotting of Description of the dataset.

# download (4).png

**From the above plotting, we can easily look at the mean,std-deviation, min, and max values of each column, it helps in further data cleaning.**

Total no of rows in the dataset:

1. 29 rows
2. 17 columns

R:

1.mean= 688.233333 2.std= 58.761754 3.min= 573 4.max= 891

AB:

1.mean= 5516 2.std= 70.46 3.min= 5385 4.max= 5649

H:

1.mean= 1403 2.std= 57.14 3.min= 1324 4.max= 1515

2B:

1.mean= 274.733 2.std= 18.09 3.min= 236 4.max= 308

3B:

1.mean= 301 2.std= 10.45 3.min= 13 4.max= 49

HR:

1.mean= 163 2.std= 31.82 3.min= 100 4.max= 232

BB:

1.mean= 469.1 2.std= 57 3.min= 375 4.max= 570

SO:

1.mean=1248 2.std= 103 3.min= 973 4.max= 1518

SB:

1.mean= 83.50 2.std= 22.81 3.min= 44 4.max= 134

RA:

1.mean= 688.23 2.std= 72.10 3.min= 525 4.max= 844

ER:

1.mean= 635 2.std= 70 3.min= 478 4.max= 799

ERA:

1.mean=3.95 2.std= .45 3.min= 2.94 4.max= 5.04

CG:

1.mean= 3.46 2.std= 2.76 3.min= 0000 4.max= 11

SHO:

1.mean= 11.3 2.std= 4.12 3.min= 4.0 4.max= 21

SV:

1.mean=1248 2.std= 43 3.min= 28 4.max= 68

E:

1.mean=94.33 2.std= 13.95 3.min= 75 4.max= 126

# Now use subplot and distribution plot to check data are normalized or not.

# download (5).png

# Observations:

# 1. Some data is normally distributed and some are not.

# 2. Data is skewed towards the right.

# 3. outliers are present as data is out of the normalized curve.

# Plotting of positive and negative correlated columns.

# download (6).png

# Observations:

# 1. SV,BB,SHO,R,2B,HR,SO and CG are making positive relations.

# 2. E, SB,3B, and RA make a negative correlation.

# Regplot: This method is used to plot data and a linear regression model fit. (POSITIVE Relation)

# SV and W

# download (22).png

# BB and W

# download (15).png

# SHO and W

# download (21).png

# R AND W

# download (9).png

# 2B AND W

# download (12).png

# HR AND W

# download (14).png

# SO AND W

# download (16).png

# Regplot: This method is used to plot data and a linear regression model fit. (Negative Relation)

# E AND W

# download (23).png

# SB AND W

# download (17).png

# 3B AND W

# download (13).png

# RA AND W

# download (18).png

# For finding out the positive and negative relation we use a regression plot for insights into the data.

# Detecting outliers

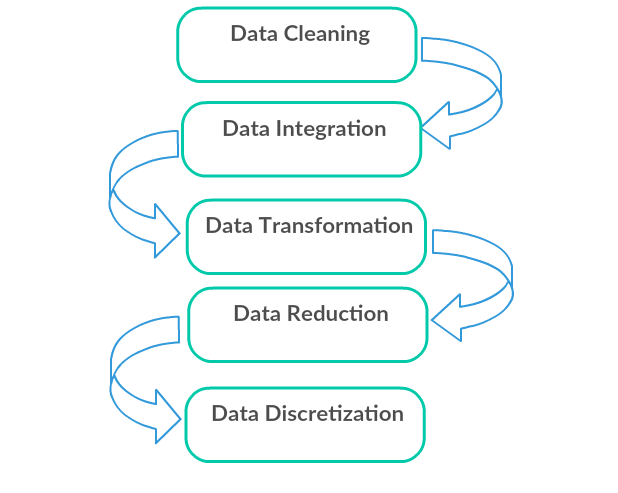
# download (7).png

# download (8).png

# Outliers are present in the dataset.

**4. Pre-Processing Pipeline.**

Data pre-processing can refer to the manipulation or dropping of data before it is used to ensure or enhance performance, and is an important step in the data mining process.



1. Data Cleaning: First we clean the data which have no use in prediction like the ID column, and then we drop the data which has a high no of missing percentages.
2. Data Integration: then we do some EDA process for finding out the meaning full insights of the data.
3. Data transformation is the process of changing the format, structure, or values of data; we use a labeled encoder for coding the object data into integer data.
4. Data Reduction: it is the process of finding the most correlated columns, and combining them because the machine does not understand which feature columns impact the most on accuracy.
5. Data discretization converts a large number of data values into smaller once, so that data evaluation and data management becomes very easy, using box plots is makes a clear understanding of the data.

**5. Building Machine Learning Models**.

After analyzing the dataset, and doing pre-processing.

Find the correlation between the columns with target columns and delete the non-related feature columns.

We observe that the target column is skewed so we remove the skewness of the target column because normal data gives better results when we make the M.L model.

The target column is continuous type so we start work on Regression models building.

* Testing of Identified Approaches (Algorithms)

1. Linear Regression
2. Regurgitation:

Lasso & Ridge Regression

1. Ensemble techniques

Decision Tree Regression

Random forest Regression

1. Gradient Boosting Regression
2. Support vector machine
3. K-nearest Neighbour Regression

**Linear Regression Model**

• Linear Regression is a machine learning algorithm based on supervised learning.

• It performs a regression task. Regression models a target prediction value based on independent variables.

• It is mostly used for finding out the relationship between variables and forecasting.

# download (24).png

**0.9321945078346161**

**Training r2\_score is:- 91.28813591198644**

**Testing r2\_score is:- 93.21945078346161**

**Observations:**

1. This Linear Regression Performs with 93% accuracy for predicting total wins.
2. We use the best-fit line and we can easily see that most of the points are fall on the line.

**Regularization:**

# 1. Lasso

# ****Lasso regression**** is a type of ****linear regression****that uses shrinkage. Shrinkage is where data values are shrunk towards a central point as they mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

# download (25).png

**At cv:- 4**

**Cross-validation score is:- 75.31384350380506**

**R2\_score is :- 92.16206706388459**

**Observations:**

**R2 score and cv are good.**

# 2. Ridge

# Ridge regression is a model tuning method that is used to analyze any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

# download (26).png

**At cv:- 4**

**R2 Score: 94.07505578566887**

**Cross Val Score: 63.97423569357152**

# Observations:

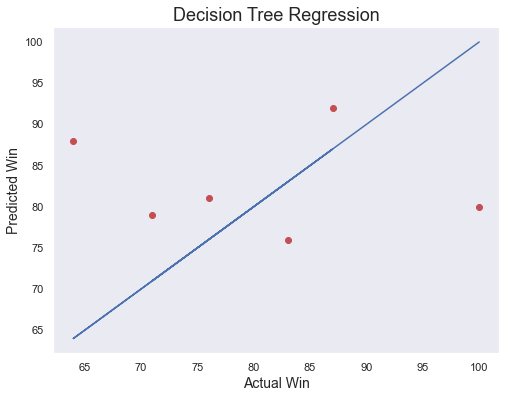
# Rsults are good, from lasso but cv is not up to the mark

# Ensemble Techniques:

**Decision Tree Regressor**

A decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches, each representing values for the attribute tested. Leaf node (e.g., present or not-present) represents a decision on the numerical target. The topmost decision node in a tree that corresponds to the best predictor is called the **root node**. Decision trees can handle both categorical and numerical data.



**At cv:- 2**

**R2 Score: 79.77389516957862**

**Cross Val Score: -71.22898240460206**

**Observations:**

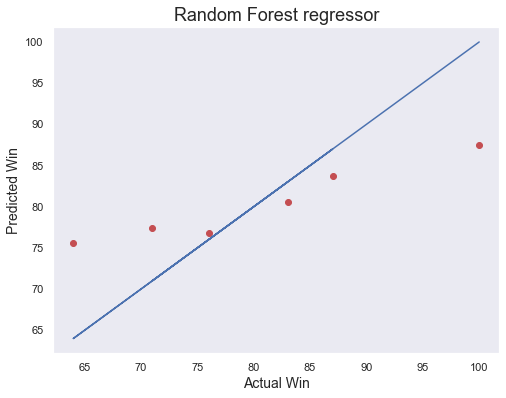
1. This Decision Tree Regression Performs with 79% accuracy for predicting total wins.
2. After predicting and plotting the predicted data on the best fit line we observe that DTR is not so accurate.

**Random Forest Regression Model**

1. A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as bagging.

2. Bagging, in the Random Forest method, involves training each decision tree on a different data sample where sampling is done with replacement.

3. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.



**At cv:- 2**

**R2 Score: 69.43427201096264**

**Cross Val Score: 2.166800965403237**

Observations:

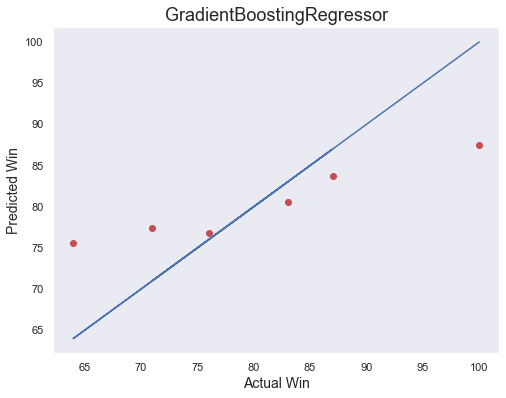
1. RFR performs not that well.

2. CV at2 is giving good results.

3. After predicting and plotting the predicted data on the best fit line we observe that RFR is not so accurate.

# Gradient Boosting Regression

# Gradient boosting is a machine learning technique for regression, classification, and other tasks, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. When a decision tree is a weak learner, the resulting algorithm is called gradient boosted trees, which usually outperforms random forest. It builds the model in a stage-wise fashion as other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.



**At cv:- 4**

**R2 Score: 77.40068588353249**

**Cross Val Score: 44.4393604685385**

Observations:

1. GBR performs better and gives optimum results.
2. After predicting and plotting the predicted data on the best fit line we observe that GBR is accurate.

# Support Vector Regression

SVMs or Support Vector Machines are one of the most popular and widely used algorithms for dealing with classification problems in machine learning. However, the use of SVMs in regression is not very well documented. This algorithm acknowledges the presence of non-linearity in the data and provides a proficient prediction model.



**At cv:- 4**

**R2 Score: 91.24164073439381**

**Cross Val Score: 61.242675480230105**

Observations:

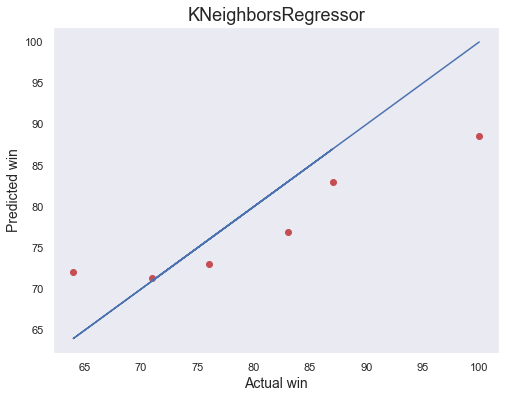
1. SVR performs better and gives optimum results.

2. After predicting and plotting the predicted data on the best fit line we observe that SVR is accurate.

# K-nearest Neighbors Regression

KNN regression is a non-parametric method that, intuitively, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighborhood*.* The size of the neighborhood needs to be set by the analyst or can be chosen using cross-validation (we will see this later) to select the size that minimizes the mean-squared error.

While the method is quite appealing, it quickly becomes impractical when the dimension increases, i.e., when there are many independent variables.



**At cv:- 4**

**R2 Score: 68.56737062431972**

**Cross Val Score: 0.940943503841435**

Observations:

1. KNN performs not well and gives no proper results.

2. After predicting and plotting the predicted data on the best fit line we observe that KNN is far behind from remaining algorithms.

**6. Concluding Remarks.**

So, our Aim is achieved as we have successfully ticked all our parameters as mentioned in our Aim Column. It is seen over time as the most effective attribute in predicting the SV and that the Lasso Regression is the most effective model for our Dataset with an

R2 score is **92.16.**

The best model is Lasso Regression. Since the difference between the percentages score of cross-validation and R2\_score is optimum.

At cv:- 4

Cross-validation score is:- 75.31384350380506

R2\_score is:- 92.16206706388459

That's it! We reached the end of our exercise.

Throughout this kernel, we put in practice many of the strategies for predicting the prices of the Wins. We philosophized about the variables, we analyzed 'Wins' alone and with the most correlated variables, we dealt with missing data and outliers, we tested some of the fundamental statistical assumptions. That's a lot of work that Python helped us make easier**.**

**Thank you**