 **Baseball case study and predicting wins with Python.**

**"Baseball is like a poker game. Nobody wants to quit when he's losing; nobody wants you to quit when you're ahead." - Jackie Robinson**

**A batter catcher and umpire during a baseball game

Description automatically generated**

**Overview :Baseball Game.**

**Baseball** is a bat and ball game played between two opposing teams who take turns batting and fielding. The game proceeds when a player on the , 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 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 is to prevent batters from becoming runners, and to prevent runners. 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.

**The Dataset: Major league Baseball Seasons (2014).**

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.

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

-- Output: Number of predicted wins (W)

-- General Approach: For predicting the number of wins a baseball team will attain -- based on the given input features, a linear regression approach is neccessary.

**NOTE:** All the code is executed using jupyter notebook for Python.

The overview of the steps we are going to take:

1. Problem Definition
2. Data Analysis
3. EDA Concluding Remarks
4. Pre-processing Pipeline
5. Building Machine Learning Models
6. Concluding Remarks

Lets’ begin!

**Step 1: Problem Definition:**

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.

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

-- Output: Number of predicted wins (W)

This is also considered as regression problem since the output is continuous.

**Step 2: Data Analysis**

Importing all the required libraries and performing the data analysis on the dataset which is loaded from the local CSV file into your python program.Let’s use the read\_csv method of

Pandas library.

A screenshot of a cell phone

Description automatically generated

After reading the dataset, if you open cus dataframe in jupyter notebook, you should see the columns /features as shown below:

A screenshot of a cell phone

Description automatically generated

The first column(W) is the output/label and all the other columns are features which are used to predict the dependent variable/label/target/output variable.

Check for the null or missing values in the dataset and treat the missing values accordingly.

A screenshot of a cell phone

Description automatically generated

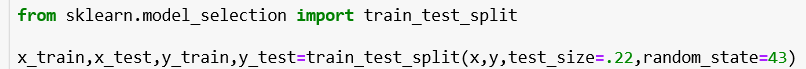
The heatmap from seaborn library is used here to visualize the null values.From the above picture ,you can observe that there are no null values present in this dataset.

**Step:3 EDA concluding remarks.**

Check for the correlation and feature dependency on the label so that we can build a strong model with optimum r^2 score(the r^2 score is considered as one of the metric to visualize the model performance.)

**Step 4: Pre-processing pipelines.**

After dropping several columns which are not helpful for the analysis,the feature set is pre-processed and brought down to the common scale to decrease the complexity in building the machine learning models.



The dataset is split into training and testing using sci-kit learn.

**Step 5: Building Machine Learning models.**

As the problem comes under Regression,here in the dataset ,performing various regression techniques such as LinearRegression,AdaBoostRegressor,GradientBoostRegressor,Lasso,RandomForestRegressor,Ridge.

**Step 6: Concluding Remarks.**

Here, in this dataset by considering the classification metrics like r^2 score,mean\_squared\_error,mean\_absolute\_error,coeff\_,intercept\_,LinearRegression has outrun other models and performed best with the accuracy of 94%.

A screenshot of text

Description automatically generated

After fitting the best model, the predictions can be saved into a csv file with the help of (to\_csv) method from pandas and the model can be saved into hard drive with the help of pickle library of python.