1. INTRODUCTION

1.1 Introduction

Machine learning is one of the applications of artificial intelligence (AI) that provides computers, the ability to learn automatically and improve from experience instead of explicitly programmed. It focuses on developing computer programs that can access data and use it to learn from themselves. The main aim is to allow computers to learn automatically without human intervention and adjust actions accordingly. Cricket had a great deal of attention in sports. The cricket score prediction is a challenging task, which can offer automated prediction about cricket score so that prediction of score can be made effective

1.2 Existing System

ICC (International Cricket Council) maintains all the cricket players and matches records. Even though, those records are not used in an efficient manner for prediction. To maintain the records in an efficient error free manner, the new proposed system is introduced**.**

Disadvantages:

1. Does not generate accurate and efficient results

2. Computation time is extremely high

3. Difficulty in maintenance of cricket match records

4. Lacking accuracy may result in lack of efficient further prediction

1.3 Proposed System

We proposed to develop a system which will help practitioners or normal person to predict ODI cricket score based on some attributes like runs, wickets, overs and so on. So, there is a need for developing a decision system which will help person to predict the cricket score in an easier way, which can offer prediction about the score of the future so that further prediction can be made effectively. This proposed system not only accurately predicts cricket score but also reduces time for prediction. The machine learning algorithms like decision

tree, random forest, Linear Regression , K Nearest Neighbours have proven to be most accurate & reliable and hence, used in this project.

Advantages:

1. Generates accurate and efficient results
2. Computation time is greatly reduced
3. Easy maintenance of matches data or records
4. Reduces manual work
5. Efficient further prediction
6. Automated prediction

1.4. System Requirements

1.4.1 Hardware Requirements:

- System Type : Intel Core i3 or above
- Cache memory : 4MB(Megabyte)
- RAM : 8 gigabytes (GB)
- Bus Speed : 5 GT/s DBI2
- Number of cores : 2
- Number of threads : 4

1.4.2 Software Requirements:

- Operating System : Windows 10 Home, 64-bit Operating System
- Coding Language : Python
- Python distribution : Anaconda, Spyder

2. LITERATURE SURVEY

2.1 Machine Learning

Machine learning is one of the applications of artificial intelligence (AI) that provides computers, the ability to learn automatically and improve from experience instead of explicitly programmed. It focuses on developing computer programs that can access data and use it to learn from themselves. The main aim is to allow computers to learn automatically without human intervention and adjust actions accordingly.

2.2 Some machine learning methods

Machine learning algorithms are often categorized as supervised and unsupervised.

- **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system can provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors to modify the model accordingly.
- In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.
- **Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labelled and unlabelled data for training – typically a small amount of labelled data and a large amount of unlabelled data. The systems that use this method can considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labelled data requires skilled and relevant resources to train it / learn from it. Otherwise, acquiring unlabelled data generally does not require additional resources.

- **Reinforcement machine learning algorithms** is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context to maximize its performance. Simple reward feedback is required for the agent to learn which action is best. This is known as the reinforcement signal.

2.3 Applications of machine learning

1. Virtual Personal Assistants
2. Predictions while Commuting
3. Videos Surveillance
4. Social Media Services
5. Email Spam and Malware Filtering
6. Online Customer Support
7. Search Engine Result Refining
8. Product Recommendations
9. Online Fraud Detection

2.4 Importance of machine learning in Sports

The importance of machine learning in sports is increasing because of its ability to process huge datasets efficiently beyond the range of human capability, and then dependably convert analysis of that data into proper insights that assist predictor in planning and assuming that's is going to happen , which ultimately leads to better outcomes, reduces the costs of prediction, and increases accuracy more. Using these types of advanced analytics, we can provide better information to predict the outcome of the any sport activity.

2.5 Regression

Regression algorithms predict a continuous value based on the input variables. The main goal of regression problems is to estimate a mapping function based on the input and output variables

2.5.1 Machine learning algorithms for regression

Research on data mining has led to the formulation of several data mining algorithms. These algorithms can be directly used on a dataset for creating some models or to draw vital conclusions and inferences from that dataset. Some popular data mining algorithms are Decision tree, Linear Regression, Random Forest, k-Neighbours, Support Vector Machine etc.

1. Decision Tree: Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

The decision rules are generally in form of if-then-else statements. The deeper the tree, the more complex the rules and fitter the model.

2. Linear Regression: Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

Before attempting to fit a linear model to observed data, a modeler should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable *causes* the other (for example, higher SAT scores do not *cause* higher college grades), but that there is some significant association between the two variables. A scatterplot can be a helpful tool in determining the strength of the relationship between two variables. If there appears to be no association between the proposed explanatory and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model. A valuable numerical measure of association between two

variables is the correlation coefficient, which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

A linear regression line has an equation of the form $\boldsymbol{Y = a + bX}$, where $\boldsymbol{X}$ is the explanatory variable and $\boldsymbol{Y}$ is the dependent variable. The slope of the line is $\boldsymbol{b}$, and $\boldsymbol{a}$ is the intercept (the value of $\boldsymbol{y}$ when $\boldsymbol{x} = 0$).

3. Random Forest: Random Forests are an ensemble learning method (also thought of as a form of nearest neighbour predictor) for classification and regression techniques. It builds multiple decision trees and then merges them together in-order to get more accurate and stable predictions. It constructs a number of Decision trees at training time and outputs the class that is the mode of the classes output by individual trees. It also tries to minimize the problems of high variance and high bias by averaging to find a natural balance between the two extremes. Both R and Python have robust packages to implement this algorithm.

4. KNN: KNN algorithm is one of the simplest Regression algorithms and it is one of the most used learning algorithms. Out of all the machine learning algorithms I have come across, KNN algorithm has easily been the simplest to pick up. Despite its simplicity, it has proven to be incredibly effective at certain tasks and even better? It can be used for both classification and regression problems! KNN algorithm is by far more popularly used for classification problems, however. I have seldom seen KNN being implemented on any regression task. My aim here is to illustrate and emphasize how KNN can be equally effective when the target variable is continuous in nature.

The KNN algorithm uses 'feature similarity' to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set Therefore, KNN could and probably should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution data.

The first step is to calculate the distance between the new point and each training point. There are various methods for calculating this distance, of which the most commonly known

methods are – Euclidian, Manhattan (for continuous) and Hamming distance (for categorical).

1. **Euclidean Distance:** Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (y).
2. **Manhattan Distance**: This is the distance between real vectors using the sum of their absolute difference.
3. **Hamming Distance**: It is used for categorical variables. If the value (x) and the value (y) are the same, the distance D will be equal to 0 . Otherwise D=1.

5. Lasso Regression: The acronym "LASSO" stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator. Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the 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.

Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in sparse models with few coefficients; Some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models. On the other hand, L2 regularization (e.g. Ridge regression) *doesn't* result in elimination of coefficients or sparse models. This makes the Lasso far easier to interpret than the Ridge

2.6 Implementation of machine learning using Python

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,

- system scripting.

The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

It is possible to write Python in an Integrated Development Environment, such as Thonny, PyCharm, NetBeans or Eclipse, Anaconda which are particularly useful when managing larger collections of Python files.

Python was designed for its readability.Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

In the older days, people used to perform Machine Learning tasks manually by coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Machine Learning are:

- Numpy
- Scipy
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Pandas
- Matplotlib

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation.

Skikit-learn is one of the most popular Machine Learning libraries for classical Machine Learning algorithms. It is built on top of two basic Python libraries, NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with Machine Learning.

Theano is a popular python library that is used to define, evaluate and optimize mathematical expressions involving multi-dimensional arrays in an efficient manner. It is achieved by optimizing the utilization of CPU and GPU. It is extensively used for unit-testing and self-verification to detect and diagnose different types of errors. Theano is a very powerful library that has been used in large-scale computationally intensive scientific projects for a long time but is simple and approachable enough to be used by individuals for their own projects.

TensorFlow is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensorflow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

Keras is a very popular Machine Learning library for Python. It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best thing about Keras is that it allows for easy and fast prototyping.

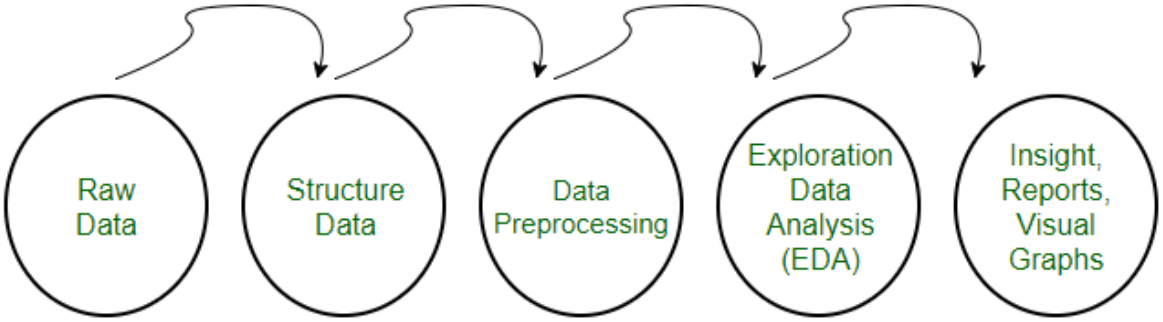
PyTorch is a popular open-source Machine Learning library for Python based on Torch, which is an open-source Machine Learning library which is implemented in C with a wrapper in Lua. It has an extensive choice of tools and libraries that supports on Computer Vision, Natural Language Processing(NLP) and many more ML programs. It allows developers to perform computations on Tensors with GPU acceleration and also helps in creating computational graphs.

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

Matpoltlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, histogram, error charts, bar chats, etc.

Data Pre-processing

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.



Need of Data Pre-processing

For achieving better results from the applied model in Machine Learning projects the format of the data must be in a proper manner. Some specified Machine Learning model needs information in a specified format. For example, Random Forest algorithm does not support null values, therefore, to execute random forest algorithm null values have to be managed from the original raw data set.
Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one dataset, and best out of them is chosen.

2.7 Machine learning products

Machine learning (ML) is one of the intelligent methodologies that have shown promising results in the domains of classification and prediction. One of the expanding areas necessitating good predictive accuracy is sport prediction, due to the large monetary amounts involved in betting.

In addition, club managers and owners are striving for classification models so that they can understand and formulate strategies needed to win matches. These models are based on numerous factors involved in the games, such as the results of historical matches, player performance indicators, and opposition information.

This paper provides a critical analysis of the literature in ML, focusing on the application of Artificial Neural Network (ANN) to sport results prediction. In doing so, we identify the learning methodologies utilised, data sources, appropriate means of model evaluation, and specific challenges of predicting sport results. This then leads us to propose a novel sport prediction framework through which ML can be used as a learning strategy. Our research will hopefully be informative and of use to those performing future research in this application area.

Data mining is widely used in the medical field such as prediction of heart disease since it is a multidisciplinary field. Using data mining researchers are developing various techniques in-order to predict the heart diseases with high accuracy.Large no. of research work is carried out for medical diagnosis for various diseases

Prediction of heart disease using k-nearest neighbor and particle swarm optimization was introduced Jabber MA[3]. Feature subset selection is used to solve this problem. Feature selection improved accuracy and reduced the running time. Before feature subset selection accuracy obtained is 75%. PSO search filters the number of features and selects the features which contribute more to the classification. By applying KNN with PSO accuracy improved to 100%.

Prediction of risk score for heart disease using associative classification and hybrid feature subset selection was obtained by Jabbar Akhil[5] used Feature selection as a pre-processing step in used to reduce dimensionality, removing irrelevant data and increasing accuracy and improves comprehensibility. Associative classification is a recent and rewarding technique that applies the methodology of association into classification and achieves high classification accuracy. Most associative classification algorithms adopt exhaustive search algorithms like in Apriori, and generate huge no. of rules from which a set of high quality of rules are chosen to construct efficient classifier.

An Integrated Decision Support System Based on ANN and Fuzzy_AHP for Heart Failure Risk Prediction byOluwarotimi Williams Samuela, Grace MojisolaAsogbona, Arun Kumar Sangaiahc, Peng Fanga and Guanglin Lia [6]. Fuzzy analytic hierarchy process (Fuzzy_AHP) technique was used to compute the global weights for the attributes based on their individual contribution with an accuracy of 91.10%, which is 4.40% higher in comparison to that of the conventional ANN method.

3. SYSTEM ANALYSIS

3.1 Scope of the project

The scope of this system is to maintain different cricket matches details/history in datasets, train the model using the large quantity of data present in datasets and predict the score based on related information and parameters. This project not only predict the score

also analyse the accuracy level with R square value to know how much percentage this model suits for this data.

3.2 Analysis

The dataset used in this work is **odi** dataset which is obtained from ESPN (Entertain and Sports Programming Network) Machine Learning repository. The dataset contains 15 attributes which are used to predict the Cricket score such as

1. mid -Each match is given a unique number
2. date -When the match happened
3. venue -Stadium where match is being played
4. bat_team -Batting team name
5. bowl_team -Bowling team name
6. batsman -Batsman name who faced that ball
7. bowler -Bowler who bowled that ball
8. runs -Total runs scored by team at that instance
9. wickets -Total wickets fallen at that instance
10. overs -Total overs bowled at that instance
11. runs_last_5 -Total runs scored in last 5 overs
12. wickets_last_5 -Total wickets that fell in last 5 overs
13. striker -max (runs scored by striker, runs scored by non-striker)
14. non-striker -min (runs scored by striker, runs scored by non-striker)
15. total - Total runs scored by batting team after first innings

The dataset contains 1188 matches , 350899(columns),15(rows)

The dataset is converted into a csv (comma separated values) file.

mid	date	venue	bat_team	bowl_tear	batsman	bowler	runs	wickets	overs	runs_last_5	wickets_last_5	striker	non-strike	total
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	DT Johnston	0	0	0.1	0	0	0	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	DT Johnston	0	0	0.2	0	0	0	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	DT Johnston	4	0	0.3	4	0	0	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	DT Johnston	6	0	0.4	6	0	0	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	DT Johnston	6	0	0.5	6	0	0	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	DT Johnston	6	0	0.6	6	0	0	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	EC Joyce	D Langford-Smit	6	0	1.1	6	0	0	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	EC Joyce	D Langford-Smit	6	0	1.2	6	0	0	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	EC Joyce	D Langford-Smit	6	0	1.3	6	0	0	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	EC Joyce	D Langford-Smit	7	0	1.3	7	0	0	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	EC Joyce	D Langford-Smit	8	0	1.4	8	0	1	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	D Langford-Smit	8	0	1.5	8	0	1	0	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	D Langford-Smit	9	0	1.6	9	0	1	1	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	DT Johnston	10	0	2	10	0	1	1	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	DT Johnston	10	0	2.1	10	0	1	1	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	DT Johnston	11	0	2.2	11	0	1	1	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	EC Joyce	DT Johnston	11	0	2.3	11	0	1	1	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	EC Joyce	DT Johnston	12	0	2.4	12	0	2	1	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	DT Johnston	12	0	2.5	12	0	2	1	301
1	13-06-2006	Civil Service Cricket Clu	England	Ireland	ME Trescothick	DT Johnston	16	0	2.6	16	0	5	2	301

3.3 Data Pre-processing

Before feeding data to an algorithm we must apply transformations to our data which is referred as pre-processing. By performing pre-processing, the raw data which is not feasible for analysis is converted into clean data. In-order to achieve better results using a model in Machine Learning, data format must be in a proper manner. The data should be in a particular format for different algorithms. For example, if we consider Random Forest algorithm it does not support null values. So that those null values must be managed using raw data.

3.3.1 Missing values

Filling missing values is one of the pre-processing techniques. The missing values in the dataset is represented as '?' but it a non-standard missing value and it must be converted into a standard missing value NaN. So that pandas can detect the missing values. The fig1 below is a heatmap representing the missing values. In these graph missing values are not presented. In our dataset there are no missing values

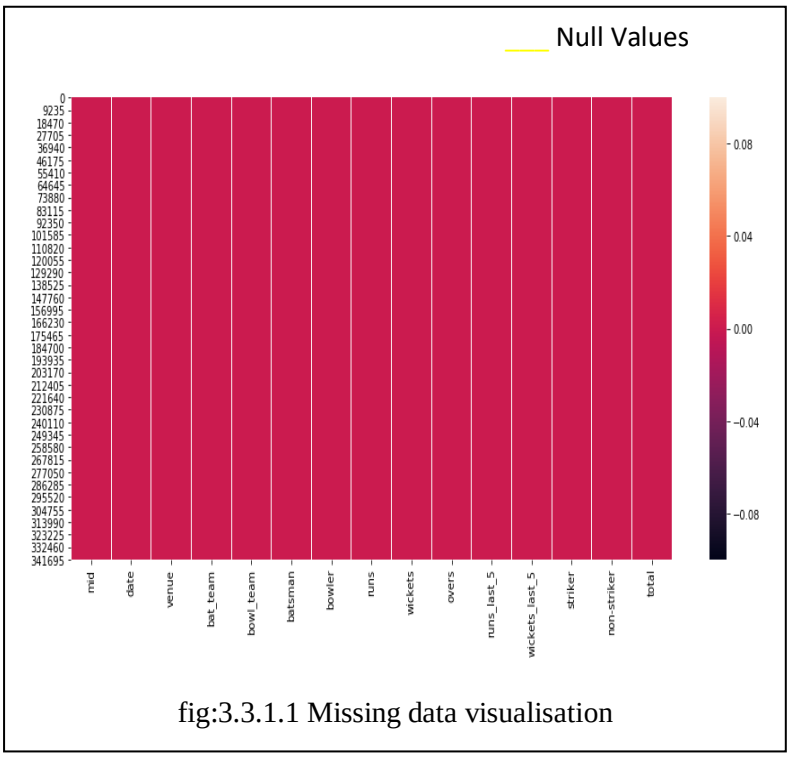


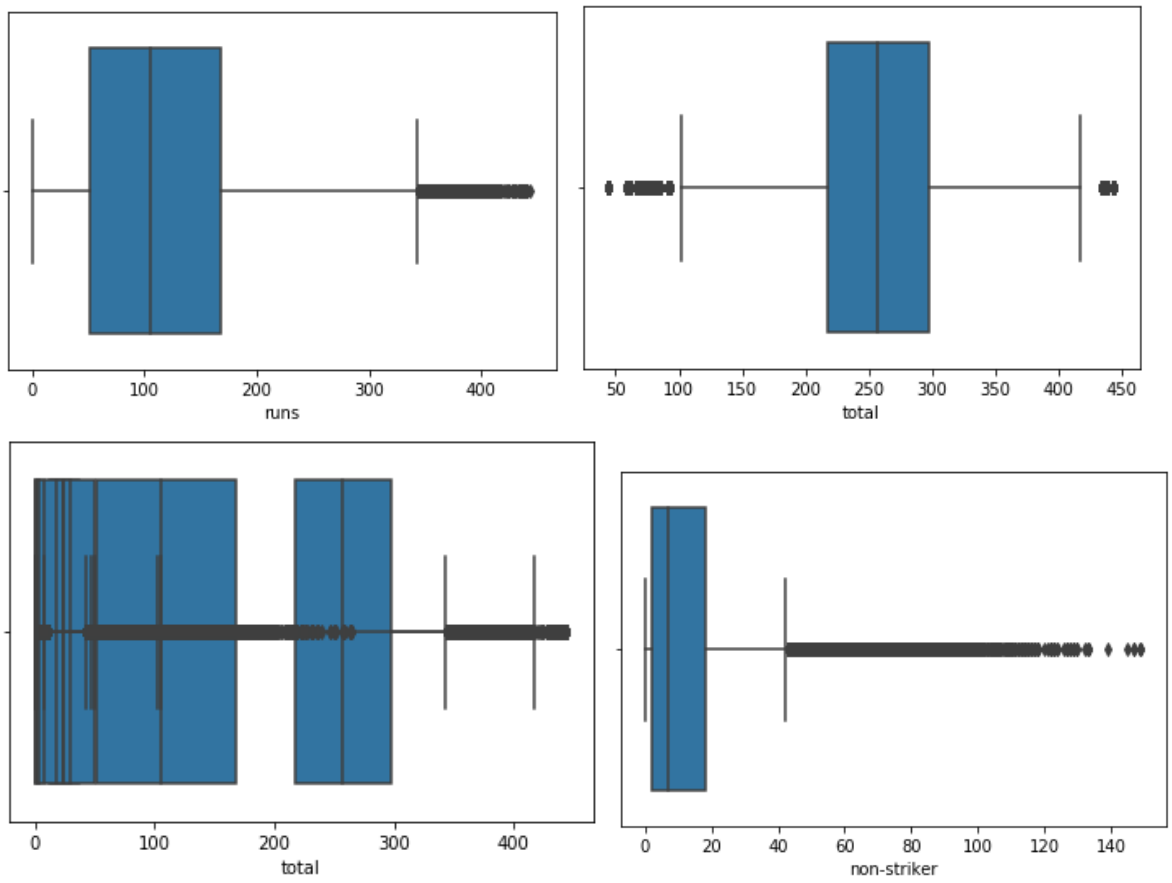
fig:3.3.1.1 Missing data visualisation

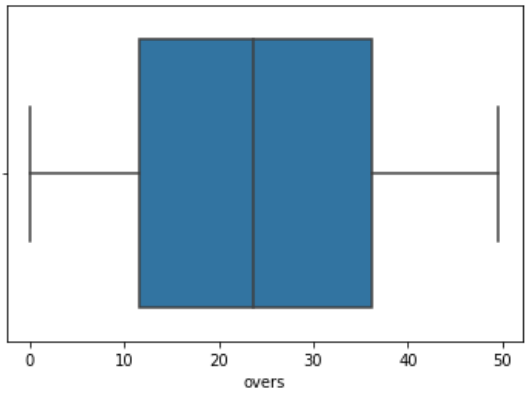
3.3.2 Outliers

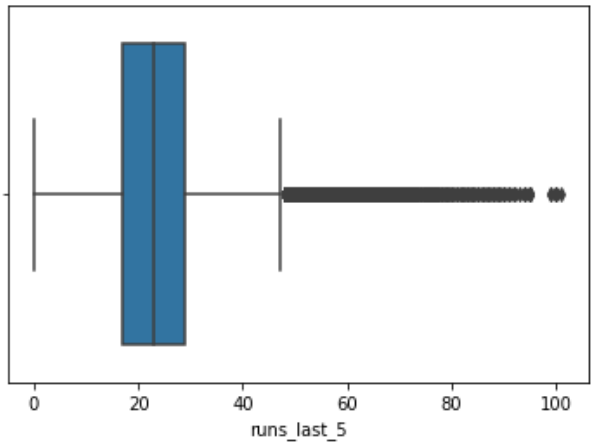
An outlier is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution error. The analysis of outlier data is referred to as outlier analysis or outlier mining.
Most data mining methods discard outliers noise or exceptions, however, in some applications such as fraud detection, the rare events can be more interesting than the more regularly occurring one and hence, the outlier analysis becomes important in such case.

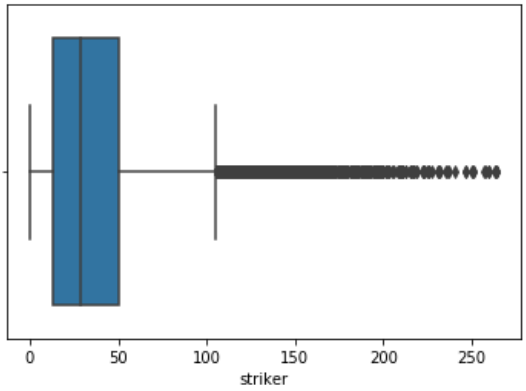
Clustering based outlier detection using distance to the closest cluster:
In the K-Means clustering technique, each cluster has a mean value. Objects belong to the cluster whose mean value is closest to it. To identify the Outlier, firstly we need to initialize the threshold value such that any distance of any data point greater than it from its nearest cluster identifies it as an outlier for our purpose. Then we need to find the distance of the test data to each cluster mean. Now, if the distance between the test data and the closest cluster to it is greater than the threshold value then we will classify the test data as an outlier.

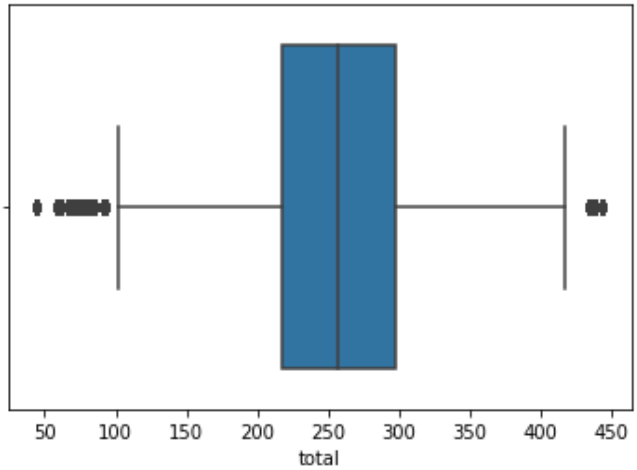
Fig: Before outliers removed:

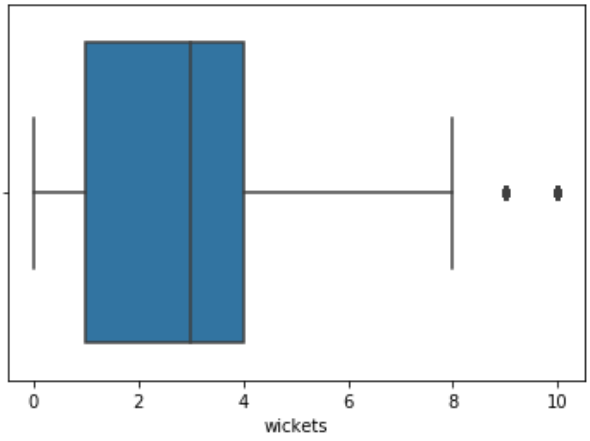



0
10
20
30
40
50
overs


0
20
40
60
80
100
runs_last_5


0
50
100
150
200
250
striker


50
100
150
200
250
300
350
400
450
total


0
2
4
6
8
10
wickets

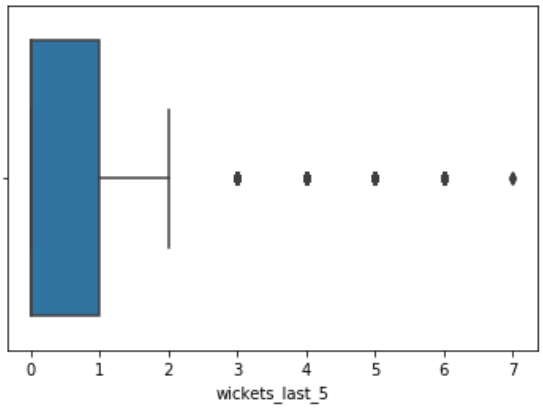
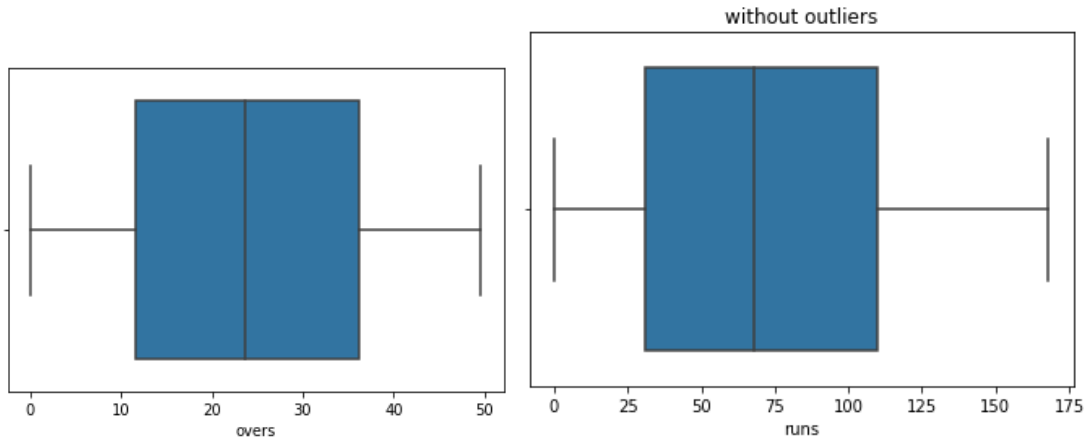
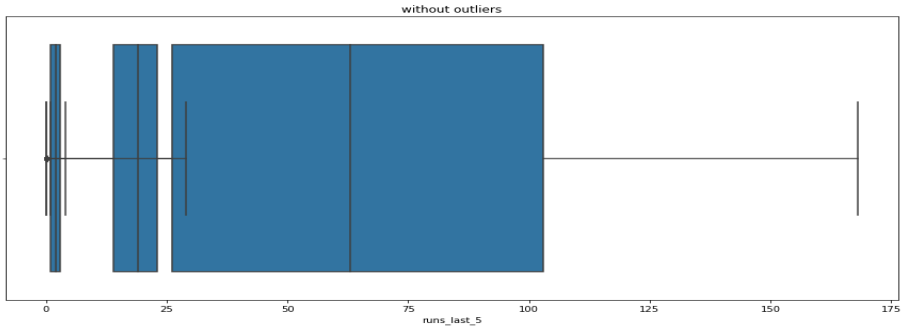
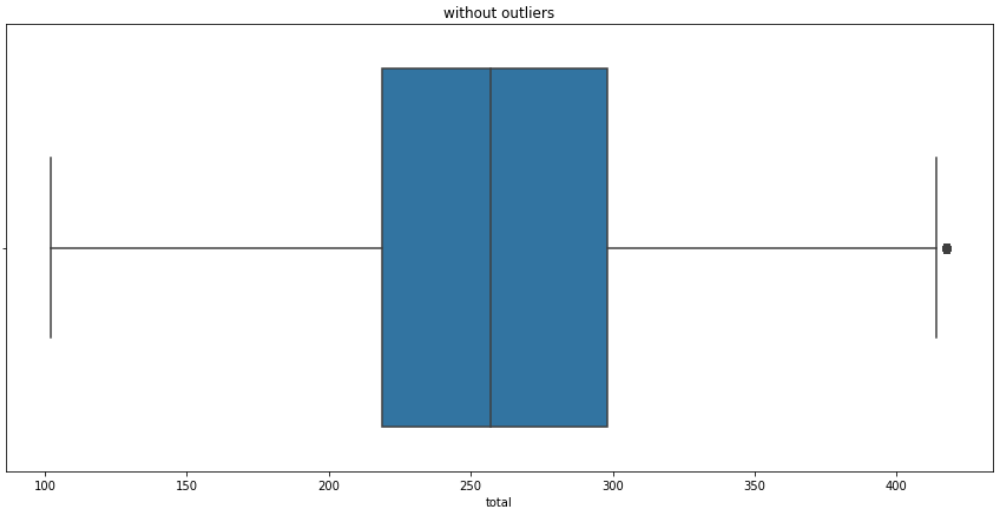
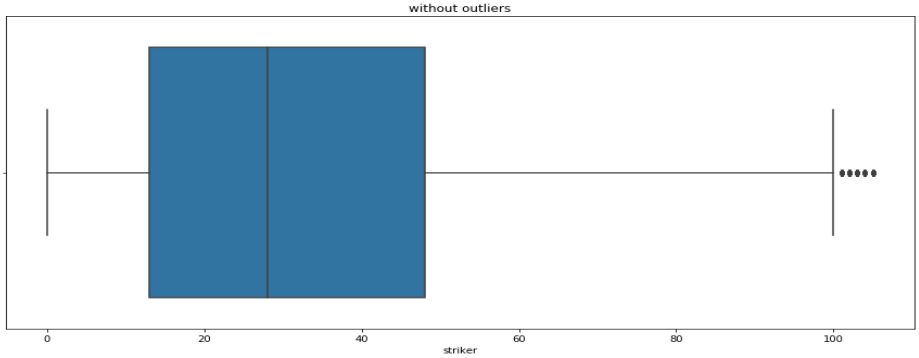

0
1
2
3
4
5
6
7
wickets_last_5

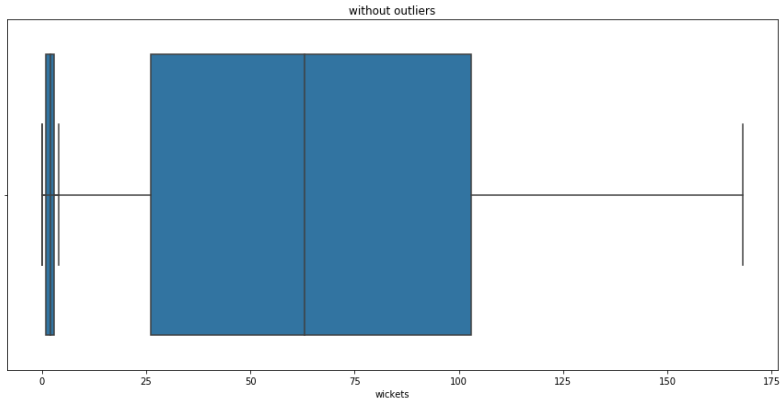
Fig:After Outliers removed:

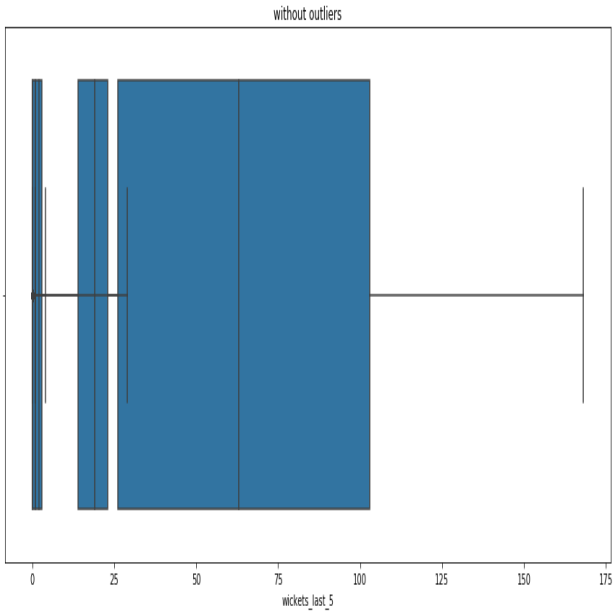












3.3.3 Feature scaling

It is nothing but data normalization in data processing used to standardize the range of independent variables. This feature is extremely useful in data pre-processing step.

The StandardScaler will assume that the data is normally distributed within each attribute and will scale them in such a way that the distribution is now centred on 0, with a standard deviation is of 1.

The mean and standard deviation are calculated for the feature and then the feature is scaled based on:

y_i–mean(y)/stdev(y)

y_i represents the values of attribute y

3.3.4 Correlation coefficient method

We can find dependency between two attributes p and q using Correlation coefficient method using the formula.

$r_{p,q}= \sum(p_i-\overline{p})(q_i-\overline{q})/n\sigma_p\sigma_q =\sum(p_i\ q_i)-n\overline{pq}/\ n\sigma_p\sigma_q$

n is the total number of patterns, p_i and q_i are respective values of p and q attributes in patternsi, $\overline{p}$and $\overline{q}$ are respective mean values of p and q attributes, σ_p,$\sigma_q$are respective standard deviations values of p and q attributes. Generally, $-1\leq r_{p,q}\leq +1$. If $r_{p,q}< 0$, then p and q are negatively correlated. If $r_{p,q}=0$, then p and q are independent attributes and there is no correlation between them. If $r_{p,q}> 0$, then p and q are positively correlated.

We can drop the attributes that are having correlation coefficient value as 0 as it indicates that the variables are independent with respect to the prediction attribute.

In-order to choose n value for knn algorithm we use the error rate. We choose the k value where the error rate is low.

Principle Component Analysis

It is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing *n* observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

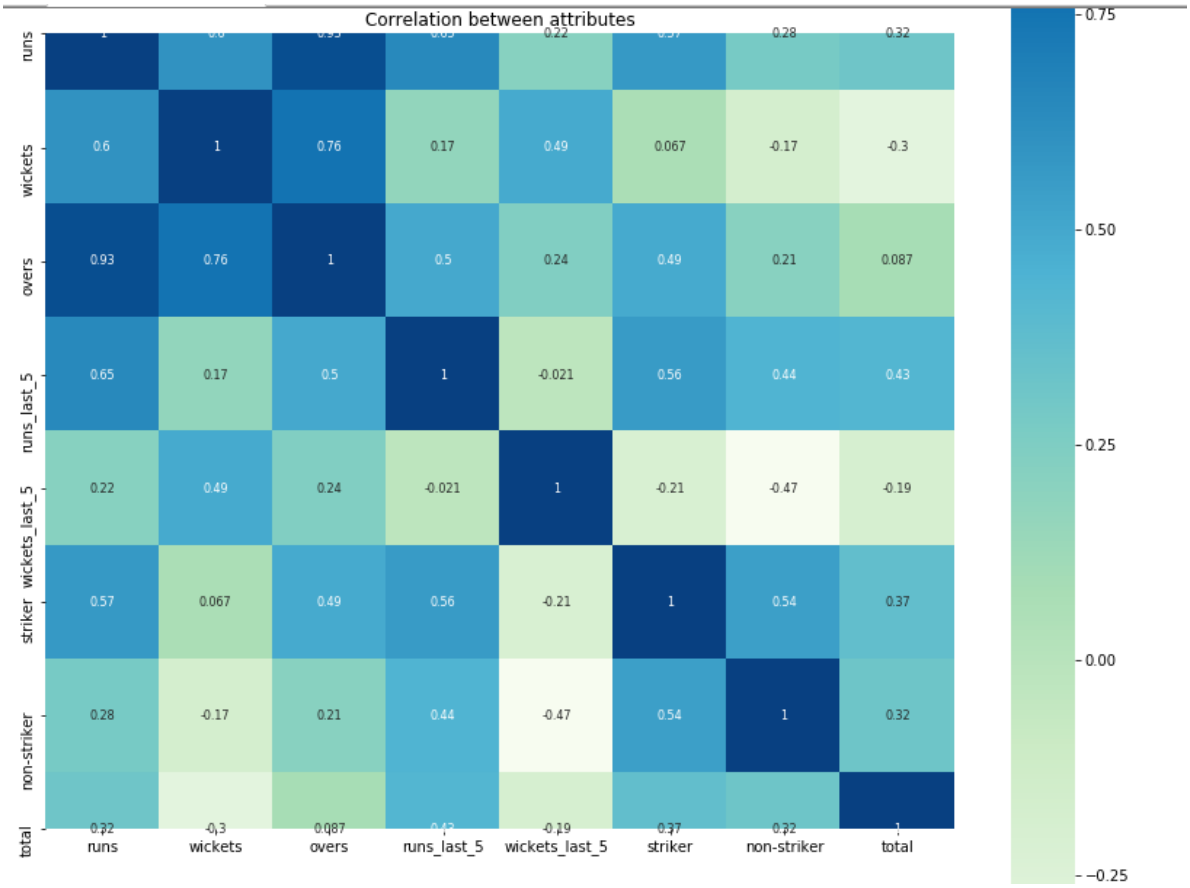


Fig: Heat map before removal of correlated attributes

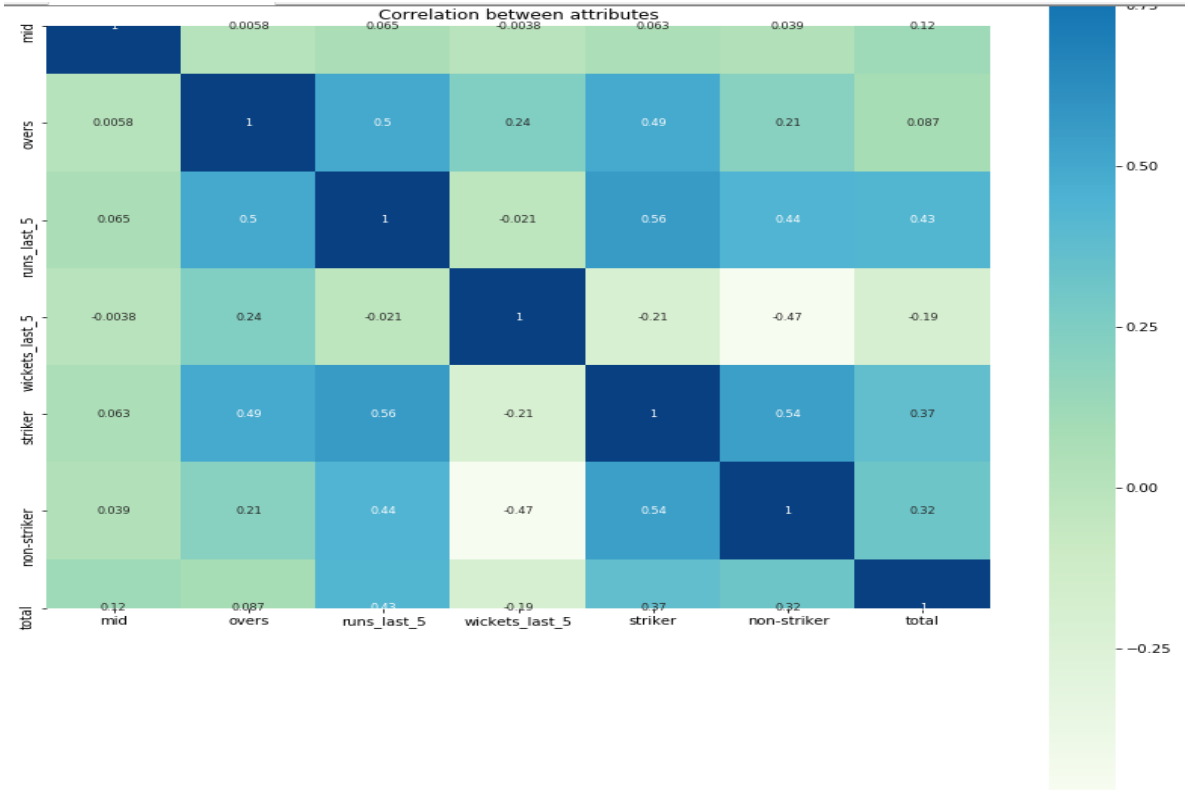


Fig: After removal of correlated attributes

3.5 Implementation code

```
import matplotlib.pyplot as plt
import seaborn as sns
def custom_accuracy(y_test,y_pred,thresold):
    right = 0
    l = len(y_pred)
    for i in range(0,l):
        if(abs(y_pred[i]-y_test[i]) <= thresold):
            right += 1
    return ((right/l)*100)
# Importing the dataset
import pandas as pd
dataset = pd.read_csv('data/odi.csv')
X = dataset.iloc[:,[7,8,9,12,13]].values     #input features
y = dataset.iloc[:, 14].values               #output featues
# Splitting the dataset into the Training set(75%) and Test set(25%)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)


# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
from sklearn.decomposition import PCA
```

```
    # Let's say, components = 2 \n
pca = PCA()
X_train=pca.fit_transform(X_train)
X_test=pca.transform(X_test)
#null values are not exist in the dataset, if exist replace with mean value
'''print(dataset.isnull().sum()) #checking for missing values or null values
#dataset.wickets=dataset.wickets.fillna(dataset.wickets.mean())'''

'''for col in dataset.columns:  #value counts
    print(col.value_count())'''
#FINDING OUTLAYERS (unrelated data to my dataset...so remove them)
sns.boxplot(x=dataset['runs'])
sns.boxplot(x=dataset['wickets'])
sns.boxplot(x=dataset['overs'])
sns.boxplot(x=dataset['runs_last_5'])
sns.boxplot(x=dataset['wickets_last_5'])
sns.boxplot(x=dataset['striker'])
sns.boxplot(x=dataset['non-striker'])
sns.boxplot(x=dataset['total'])

#DELETING OUTLIERS FROM DATASET
def remove_outliers(dataset,col_name):
    q1=dataset[col_name].quantile(0.25)
    q3=dataset[col_name].quantile(0.75)
    iqr=q3-q1
```

```
    fence_low=q1-1.5*iqr
    fence_high=q3+1.5*iqr
    dataset_out=dataset.loc[(dataset[col_name]>fence_low)&(dataset[col_name]<fence_high)]
    return dataset_out
l=['runs','wickets','overs','runs_last_5','wickets_last_5','striker','non-striker','total']
for i in l:
    outliers_removed = remove_outliers(dataset,i)
print(outliers_removed)

#dropeddata.to_csv('dropeddata.csv',index=False)
outliers_removed.to_csv('outliers_removed.csv',index=False)


#r square= 88.95 and custome is 86.42
from sklearn.neighbors import KNeighborsRegressor
neigh = KNeighborsRegressor(n_neighbors=1)
neigh.fit(X_train, y_train)
y_pred = neigh.predict(X_test)
score9= neigh.score(X_train,y_train)*100
print("R Squre value:",score9)
print("Custome accuracy for KNeighborsRegressor:",custom_accuracy(y_test,y_pred,20))
# Testing with a custom input
import numpy as np
new_prediction = neigh.predict(sc.transform(np.array([[100,0,13,50,50]])))
print("Prediction score:" , new_prediction)
```

```
models=['RandomForestRegression','LinearRegression','Losso','GaussianNB','DecisionTreeRegressor','KNeighborsRegression','SupportVectorMachine']

acc_score=[0.77,0.43,0.27,0.37,0.78,0.87,0.49]

plt.rcParams['figure.figsize']=(15,7)

plt.bar(models,acc_score,color=['green','pink','cyan','skyblue','orange','lime','blue'])

plt.ylabel("Accurate scores")

plt.title("Which model is the best accurate for inbalenced data")

plt.show()
```

3.6 Result Analysis

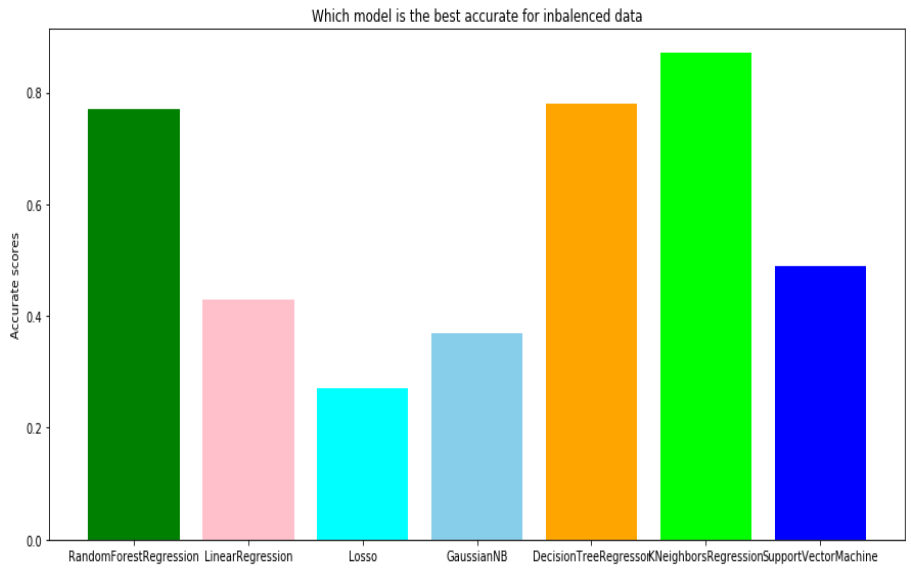


Fig 3.6.1: comparison of 7 algorithms after PCA

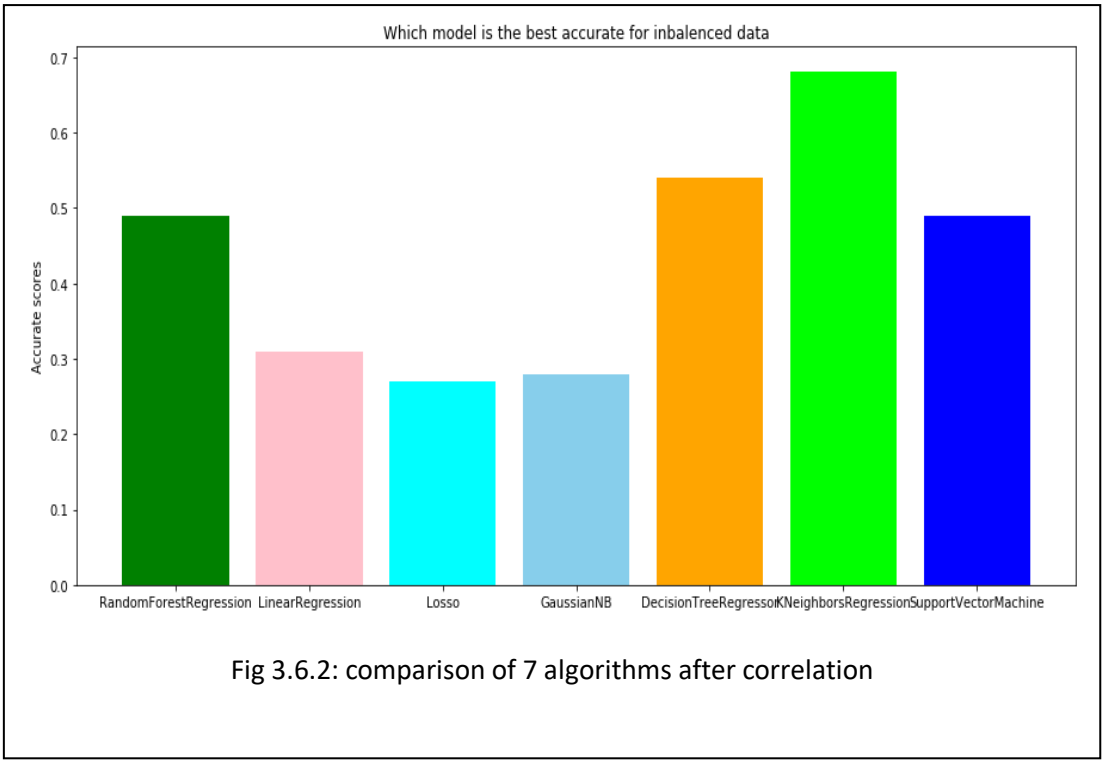


Fig 3.6.2: comparison of 7 algorithms after correlation

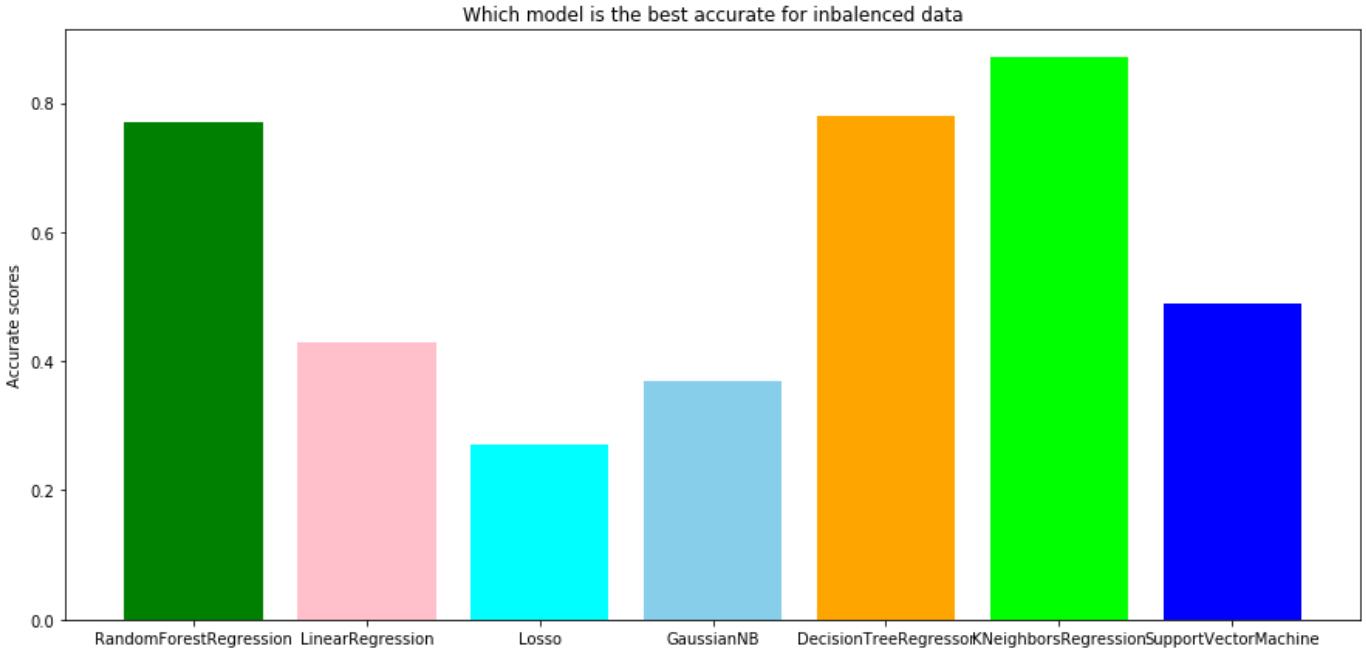


Fig 3.6.3: Comparison of all 7 Algorithms accuracy

4.SCREEN SHOTS

HOME PAGE

Fig.4.1 Output screen for Home page

PREDICT SCORE PAGE

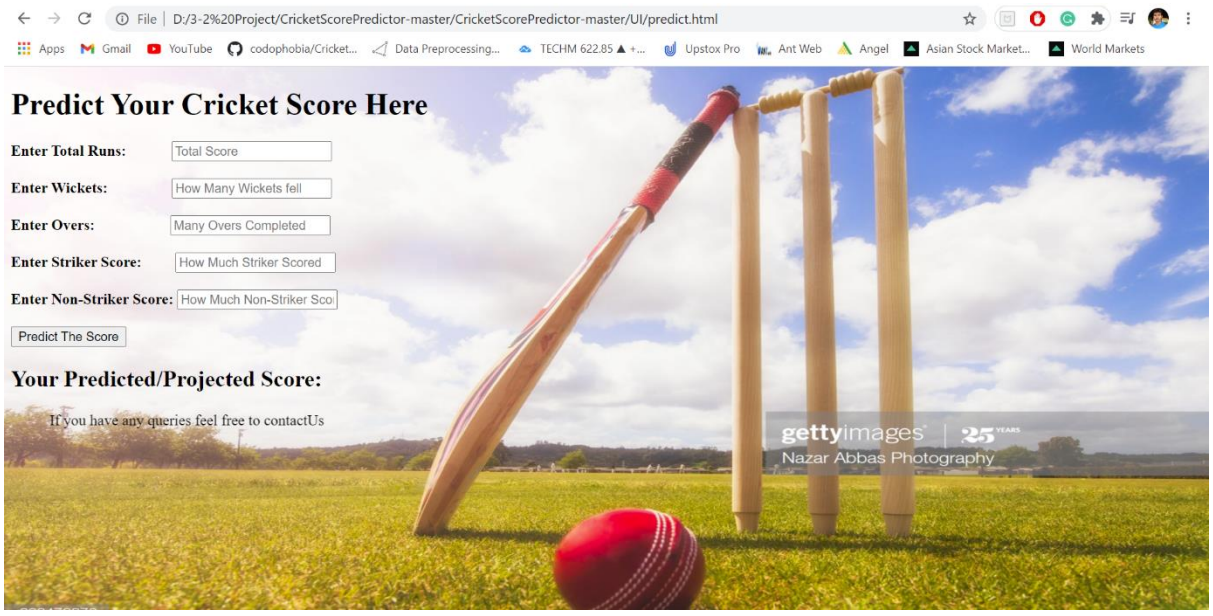


Fig.4.2 screen for predict score page

SCORE PREDICTION

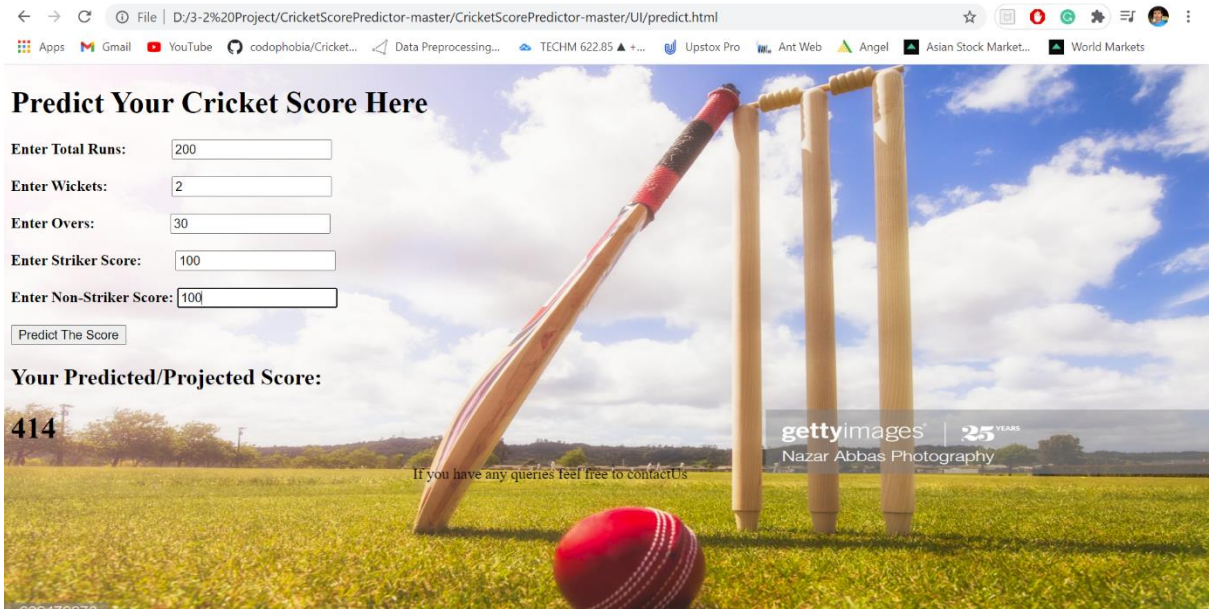


Fig.4.3 Output screen for predicted score page

PREDICTION PROCESS PAGE

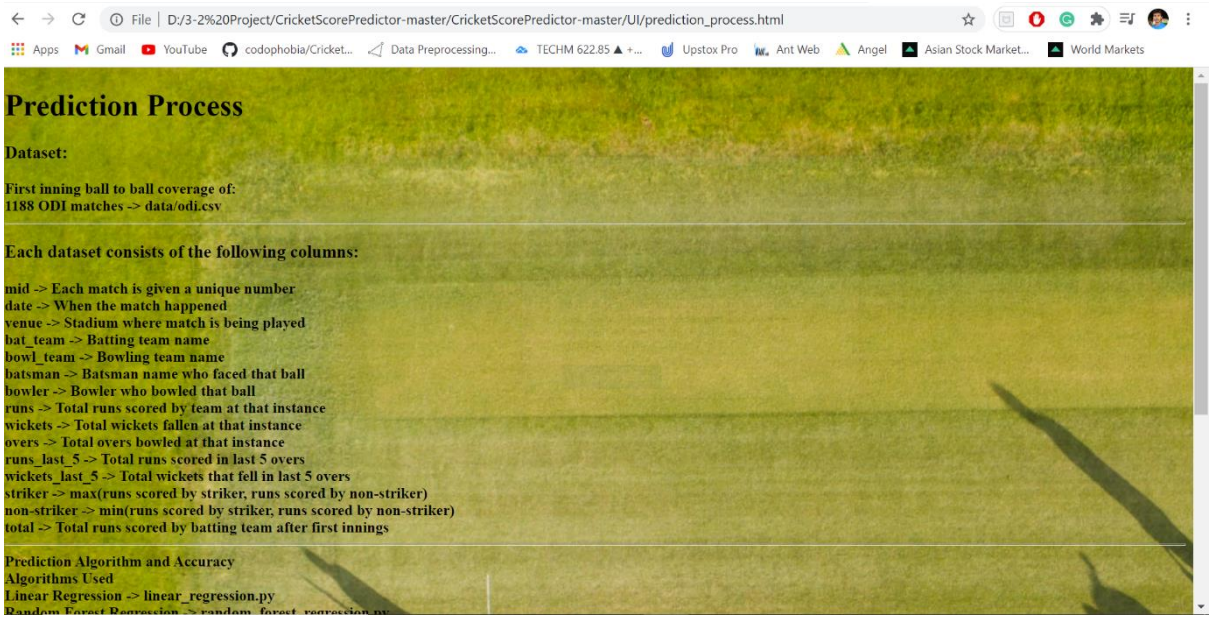


Fig.4.4 Output screen for Prediction process page

5. CONCLUSION

We have used 7 algorithms like Decision Trees, Random Forests, Support Vector Machine, KNN, Linear Regression , Lasso and GussianNB in-order to predict to predict the accurate cricket score. The accuracy varies for different algorithms. The accuracy for Decision tree algorithm is 78. The accuracy for Random Forest algorithm is 77. The accuracy for Support vector machine algorithm is 49. The accuracy for KNN algorithm is 87. The accuracy for linear regression algorithm is 43. The accuracy for Lasso algorithm is 27. The accuracy for GaussianNB algorithm is 37.

The highest accuracy is given when we have used KNN algorithm which is nearly 87.2%.

6. FUTURE SCOPE

- This project further can be developed as Android app or iOS app to predict the cricket score
- In future we can try other measures and other machine learning techniques for better comparison on results
- This analysis will help the sports sector industries to analyse and predict the score before cricket match complete

7. REFERENCES

[1] www.espn.com/odi.csv

[2] “M. Ahmad, A. Doud, L. Wang, H. Hong, October 2016, Prediction of rising stars in cricket”.

[3] “P. Somaskandhan, G. Wijesinghe, L. Bashitha, 2017, Identifying optimal set of attributes that impose high impact on end results of cricket match using machine learning.” [3] “M. Rehman, O. Shamim, S. Ismail, October 2018, an analysis of Bangladesh One day international cricket: Machine learning approach”.

[4] “A. Tripathi, J. Vanker, B. Vaje, V. Varekar, Cricket Score Prediction system using clustering algorithm.

[5] M. Bailey and S. Clarke,” Predicting the Match Outcome in One Day International Cricket Matches, while the Game is in Progress”, Journal of sports science & medicine, vol. 05, no. 04, pp. 480-487, 2006.

[6] N. Pathak and H. Wadhwa,”Applications of Modern Classification Techniques to Predict the Outcome of ODI Cricket”, Procedia Computer Science, vol. 87, pp. 55-60, 2016.

[7] P. Satao, A. Tripathi, J. Vankar, B. Vaje and V. Varekar, ”Cricket Score Prediction System (CSPS) Using Clustering Algorithm”, International Journal of Current Engineering and Scientific Research, vol. 03, no. 04, pp. 43-46, 2016.

[8]https://web.archive.org/web/20130120040151/http://www.icccricket.com/match_zone/historical_r anking.php