CricAI

"An Outcome Prediction System for Cricket"

Project Guide:
Er. Rajeev Kumar
Assistant Professor CSED,
NIT Hamirpur

Submitted By:
Amandeep Prasad (14MI507)
Rajat Patial (14MI512)
Pritish Chouhan (14MI516)
Aditya Thakur (14MI521)
Jalaz Kumar (14MI528)
Aarti Ramoul (14MI547)

Contents

- Introduction
- Methodology and Approach
- Dataset Collection
- Multi-Layer Perceptron Network
  - Applications
  - Mathematical Formulas
  - Advantages
  - Disadvantages
- Decision Trees
  - Mathematical Formulas
  - Advantages
  - Disadvantages
- Support Vector Machines
  - Data Prepration for SVM
- Results and Observations
- Tools Used
- Snapshots of Project
- Refrences

Introduction

CricAI is a tool based on emerged results, which can be used to predict the outcome of any ODI match given the concerned factors as inputs.

This software of ours can be of real value to the cricketers, support staff of teams and cricket analysts in terms of analyzing the future game in advance and working towards maximizing their chances of victory.

Introduction

We make use of 3 different classification models:

- Multilevel Perceptron Classifier
- Decision Tree Classifier
- SVM classifier

We also make observations regarding the performance

measures of all the 3 classifiers & try to comprehend that information in tabular format, for the ease of doing comparative analysis.

Methodology and Approach

The whole approach we took has a base assumption that all the future matches are somehow relatable to the historic data & no setbacks are possible.

Our approach is divided into 3 phases:

- Data collection
- Training of data for the chosen model
- Evaluating accuracy and Performance

Dataset Collection

- Dataset comprises of all the ODI matches from Jan 5, 1971 to Oct 29, 2017. A total of 3933 ODIs match results were scrapped.
- Continuous dataset is converted into a categorical dataset, using dummy variables.
- Dataset is divided into two parts:

✓ Test data

✓ Training data

- Training Dataset Size: 5620 0

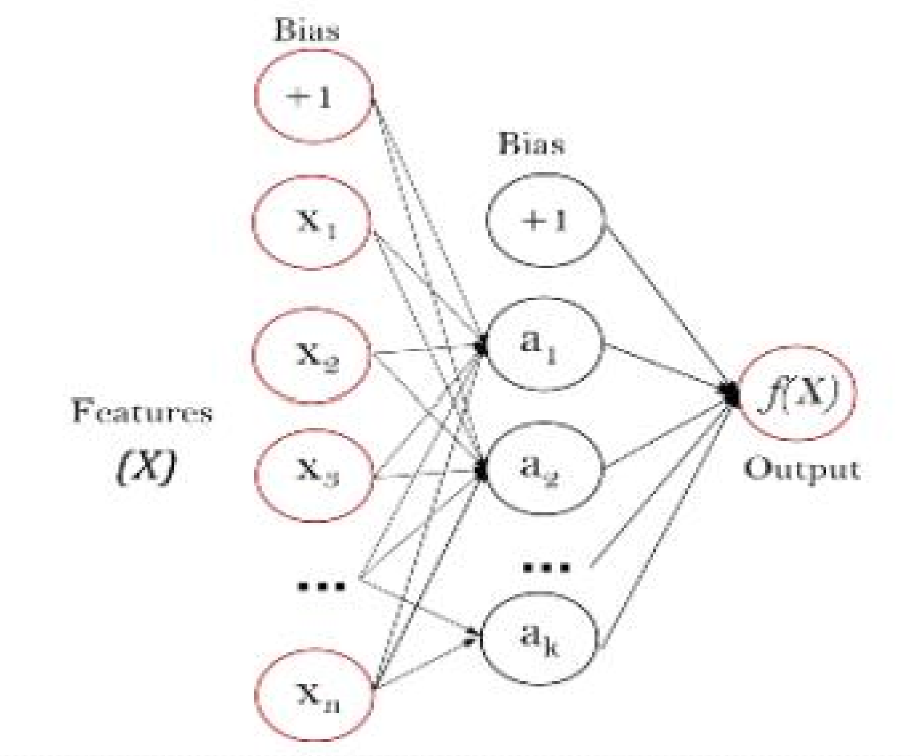
  Testing Dataset Size: 1874

Multi-Layer Perceptron Network

- A multilayer Perceptron (MLP) is a class of feed forward artificial neural network.
- An MLP consists of at least three layers of nodes.

  Except for the input nodes, each node is a neuron that uses a nonlinear activation function.
- MLP utilizes a supervised learning technique called back propagation for training.

Multi-Layer Perceptron Network



Multilayer Perceptron with 1 hidden layer

Multi-Layer Perceptron Network

- The MLP consists of three or more layers of nonlinearly-activating nodes making it a deep neural network.
- Since MLPs are fully connected, each node in one layer connects with a certain weight to every node in the following layer.

Multi-Layer Perceptron Network

- Applications:

- ✓ MLPs are useful in research for their ability to solve problems stochastically, which often allows approximate solutions for extremely complex problems like fitness approximation.
- ✓ MLPs are universal function approximators.
- ✓ Applications in diverse fields such as speech recognition, image recognition, and machine translation software, but thereafter faced strong competition from much simpler and support vector machines.

Multi-Layer Perceptron Network

- Advantages:

✓ It is capable to run non-linear models.

✓ MLP Classifier uses Back propagation so, it continuously learns and improvise itself.

✓ Capability to learn models in real-time using partial fitting.

Multi-Layer Perceptron Network

- Disadvantages:

✓ Highly sensitive to feature scaling.

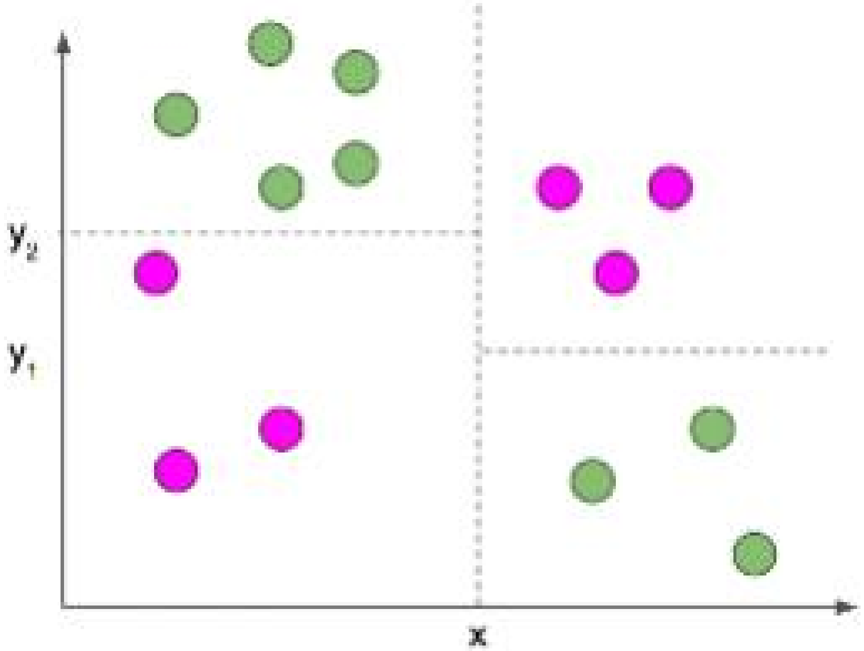
✓ It uses a black box model, results may be more difficult to interpret.

✓ MLP requires tuning a number of hyper parameters such as the number of hidden neurons, layers, and iterations.

Decision Trees

- Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receives an answer, a follow-up question is asked until a conclusion about the class label of the record is reached.
- Decision trees can be constructed from a given set of attributes.
- The decision tree inducing algorithm must provide a method for specifying the test condition for different attribute types as well as an objective measure for evaluating the goodness of each test condition.

Decision Trees



Decision Tree

Decision Trees

- To determine how well a test condition performs, we need to compare the degree of impurity of the parent before splitting with degree of the impurity of the child nodes after splitting.
- The measurements of node impurity/purity are:

✓ Gini Index

✓ Entropy

✓ Misclassification Error

- A stop condition is also needed to terminate the tree-growing process

Decision Trees

- Advantages:

- ✓ Quite Simple to understand, interpret and visualize
- ✓ Able to handle both numeric as well as categorical data and also multi-output problems.
- ✓ Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by boolean logic.

Decision Trees

- Disadvantages:

✓ Creation of over-complex trees that do not generalize the data well. Overfitting is a problem in Decision Tree.

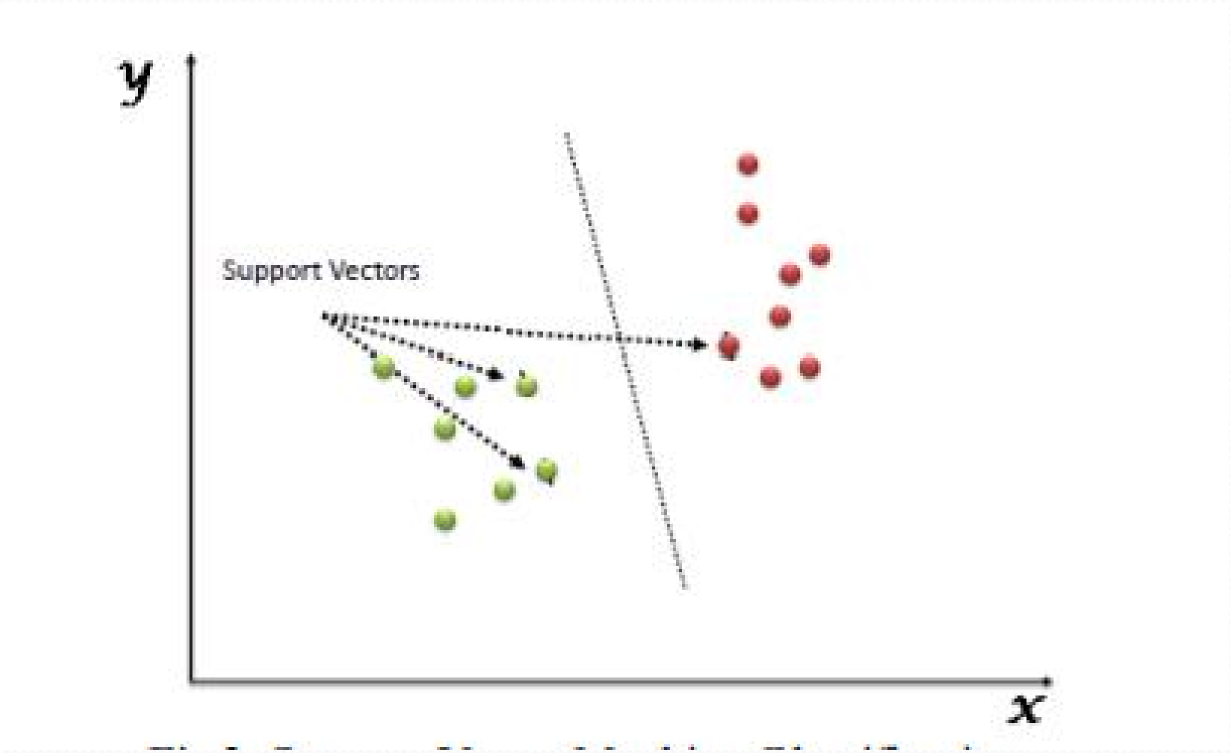
✓ Decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

✓ For the trivial cases, where some classes dominate over all others creation of biased Decision Tree usually takes place.

“Support Vector Machine” (SVM)

- “Support Vector Machine” (SVM) is a supervised machine learning models with associated learning algorithms that analyze data used for classification and regression analysis.
- Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other.
- An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

“Support Vector Machine” (SVM)



Support Vector Machine Classification

“Support Vector Machine” (SVM)

- DATA PREPARATION FOR SVM:

✓ Numerical Inputs: SVM assumes that your inputs are numeric.

✓ Binary Classification: Basic SVM as described in this post is intended for binary (two class) classification problems.

Results and Observations

- Performance Measures:

✓ **Accuracy Score:** This compares the actual outcomes with the predicted outcomes of our Classifier for a given input dataset.

✓ **Precision-Recall:** In information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned.

✓ **Precision Score:** This is defined as the number of true positives (Tp) over the number of true positives plus the number of false positives (Fp).

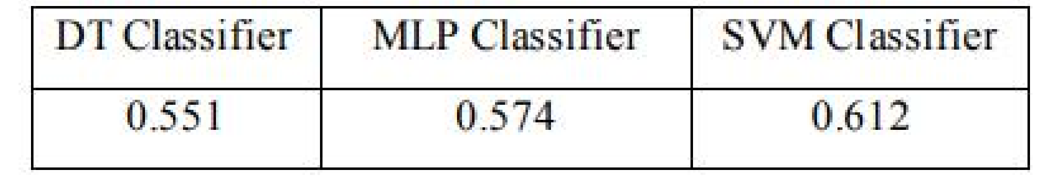
✓ **Recall Score:** This is defined as the number of true positives (Tp) over the number of true positives plus the number of false negatives (Fn).

✓ **F1 Score:** This is defined as the interpreted as a weighted average of the precision and recall.

✓ **Average Precision Score:** This summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold.

Results and Observations

- Comparative Analysis:

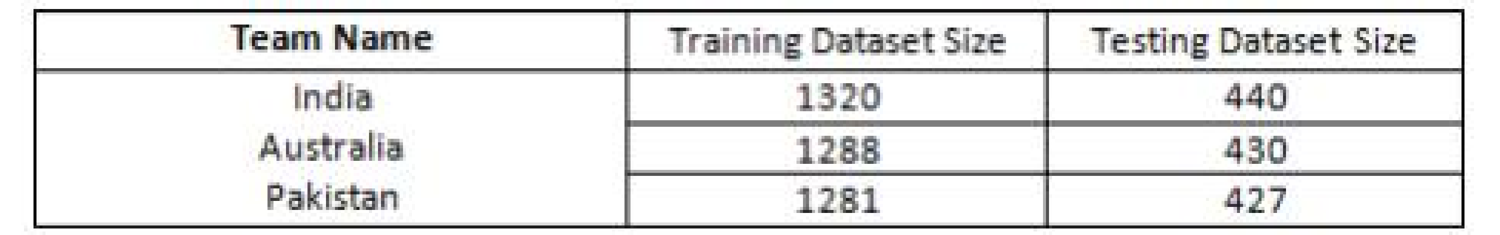


DT Classifier	MLP Classifier	SVM Classifier
0.551	0.574	0.612

Accuracy Score of all 3 models

Results and Observations

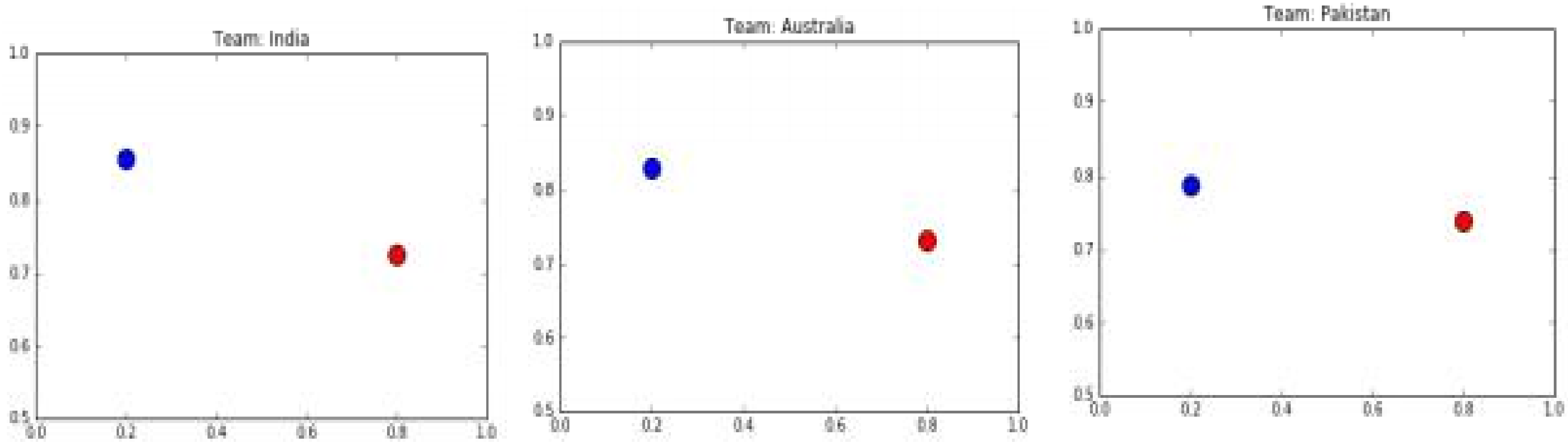
- We selected 3 teams: India, Australia and Pakistan randomly and separated the match records of these 3 teams to obtain the performance measure for them separately.



Team Name	Training Dataset Size	Testing Dataset Size
India	1320	440
Australia	1288	430
Pakistan	1281	427

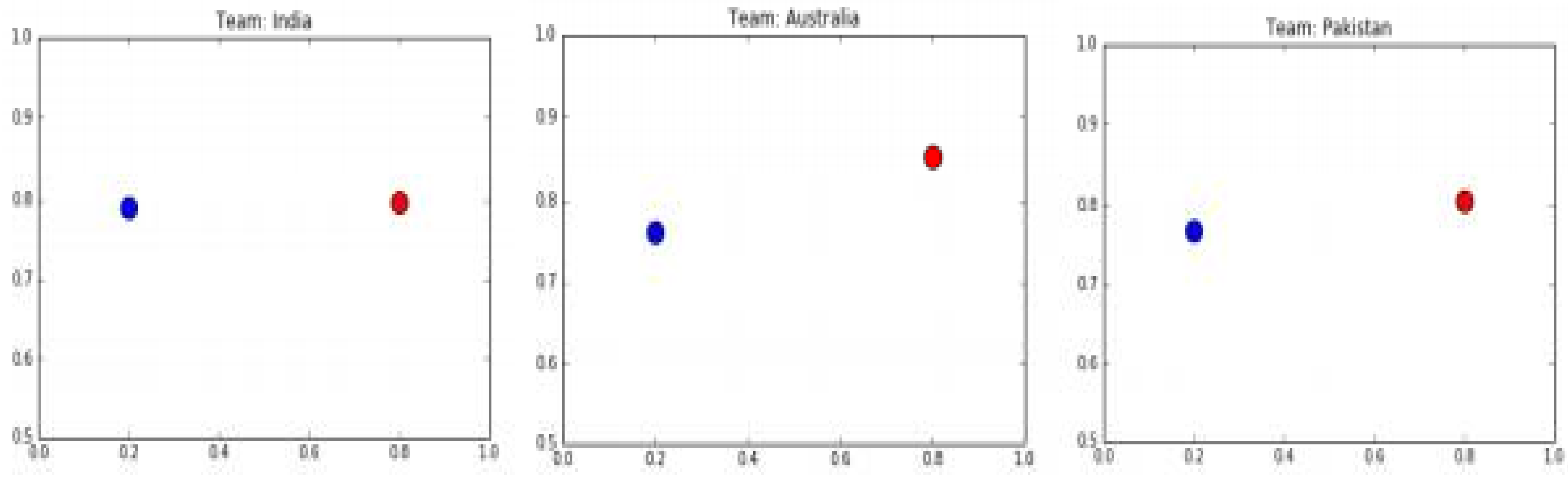
Splitted Data for Observations

Results and Observations



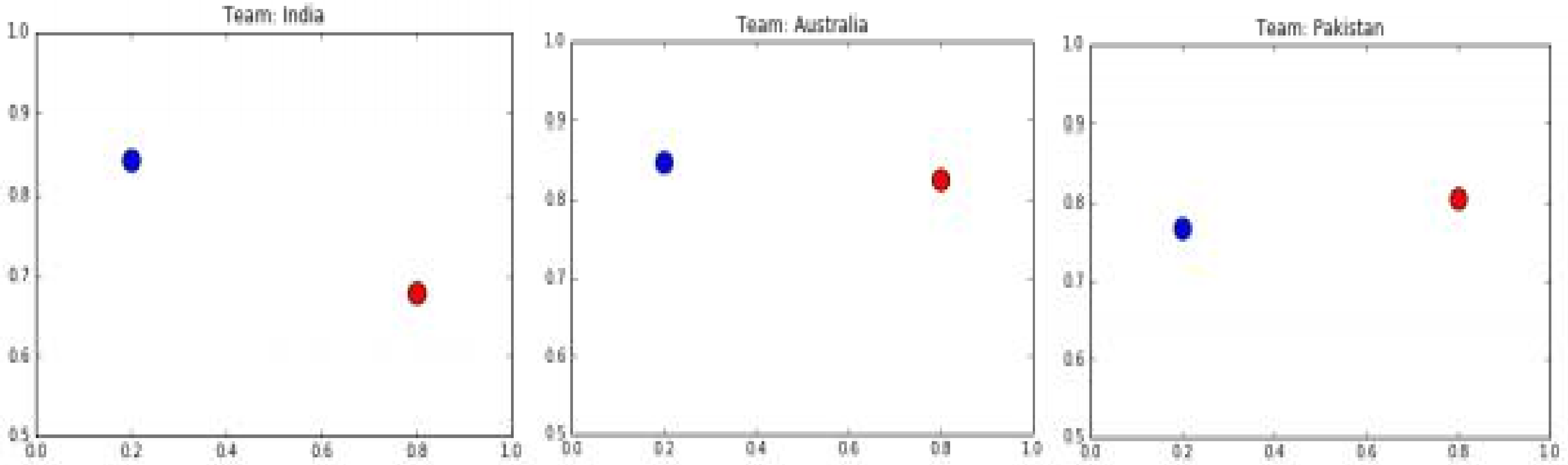
Recall-Precision Score Value for DT Classifier

Results and Observations



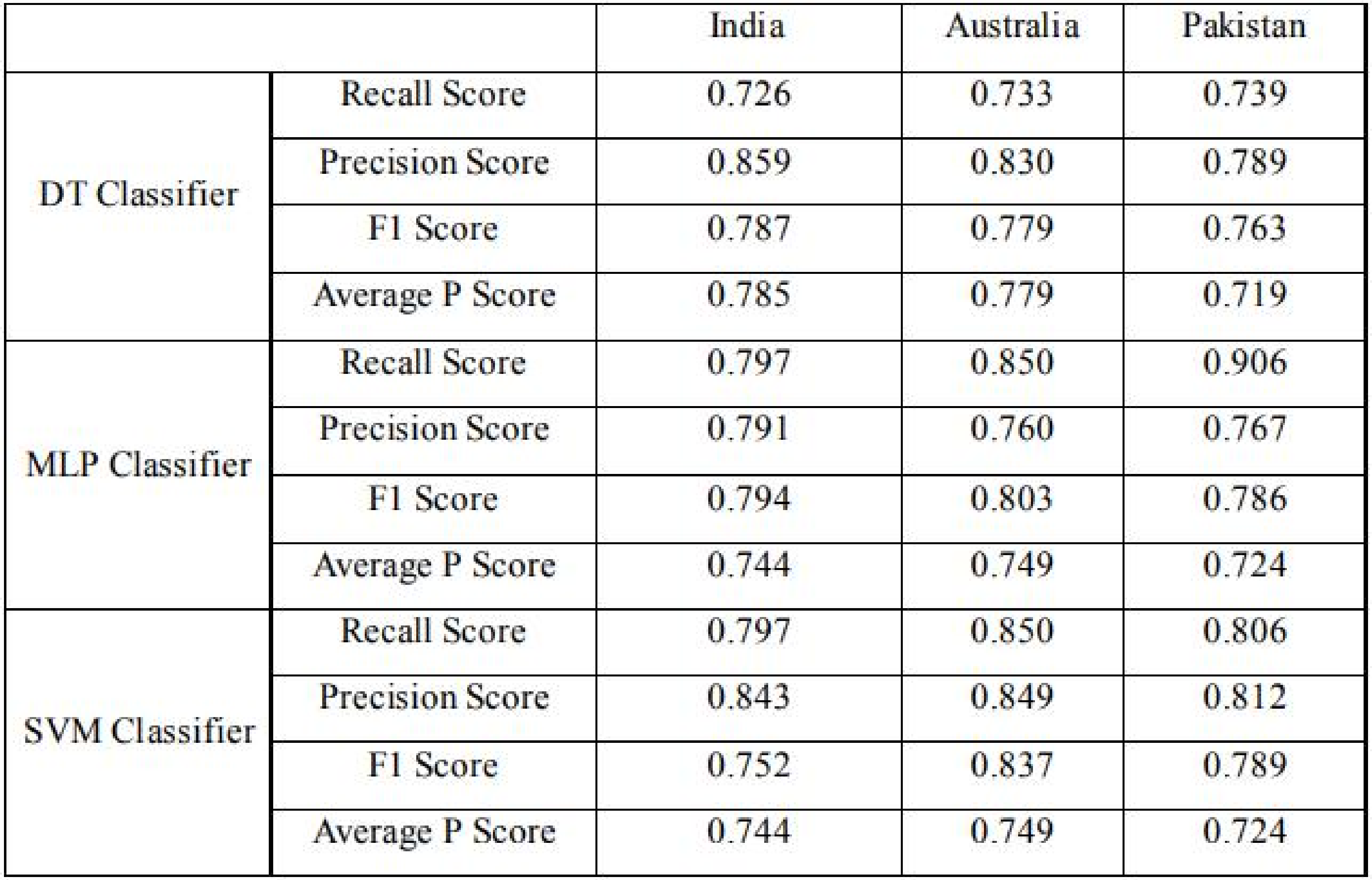
Recall-Precision Score Value for MLP Classifier

Results and Observations



Recall-Precision Score Value for SVM Classifier

Results and Observations



		India	Australia	Pakistan
DT Classifier	Recall Score	0.726	0.733	0.739
	Precision Score	0.859	0.830	0.789
	F1 Score	0.787	0.779	0.763
	Average P Score	0.785	0.779	0.719
MLP Classifier	Recall Score	0.797	0.850	0.906
	Precision Score	0.791	0.760	0.767
	F1 Score	0.794	0.803	0.786
	Average P Score	0.744	0.749	0.724
SVM Classifier	Recall Score	0.797	0.850	0.806
	Precision Score	0.843	0.849	0.812
	F1 Score	0.752	0.837	0.789
	Average P Score	0.744	0.749	0.724

Observed Value of Performance Measures

Tools Used

- 1. **Data Set**

✓ 1.1. Pandas

✓ 1.2. Beautiful Soup

✓ 1.3. UrlLib

- 2. **Project Development**

✓ 2.1. Scikit-learn

✓ 2.2. Jupyter IPython Notebook

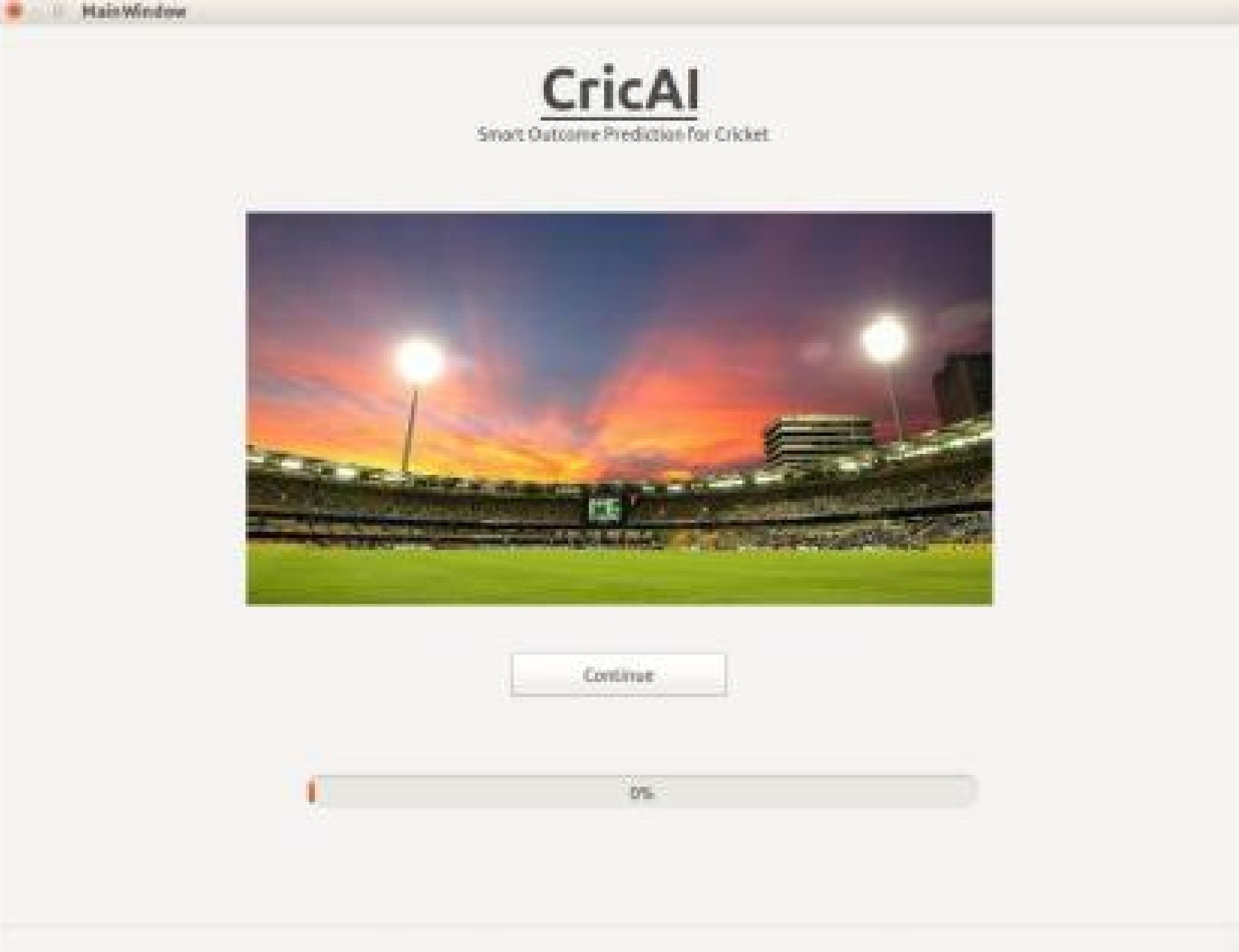
✓ 2.3. Git & GitHub

- 3. **UI Development**

✓ 3.1. PyQT5

Python was used as the main scripting language

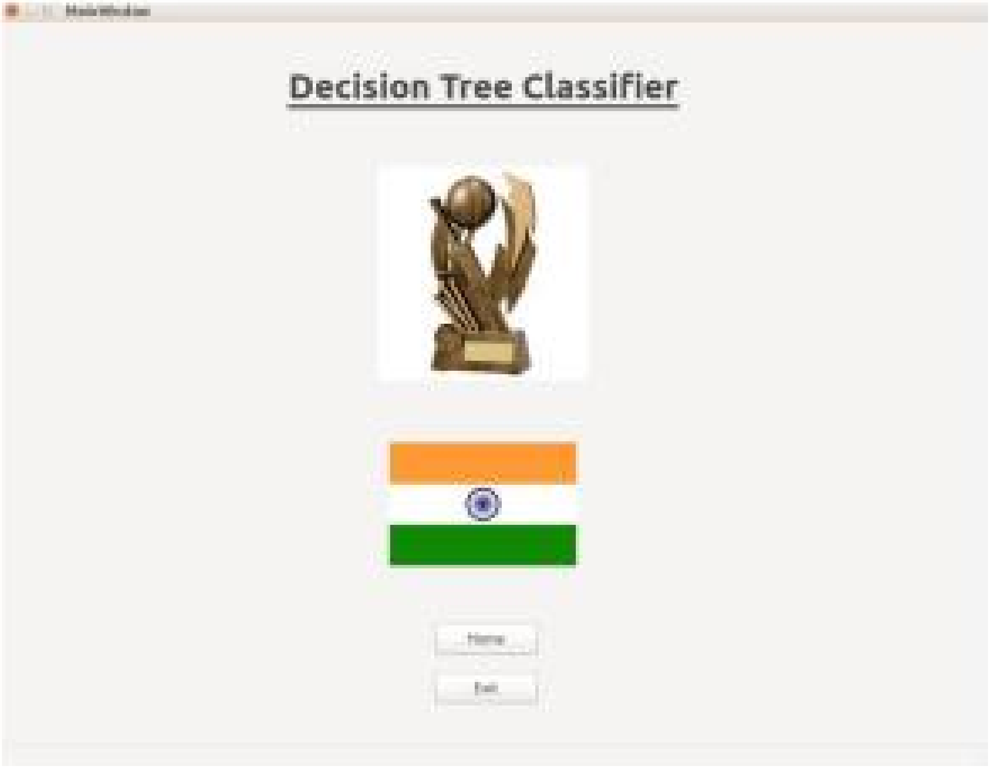
Project Snapshots



CricAI

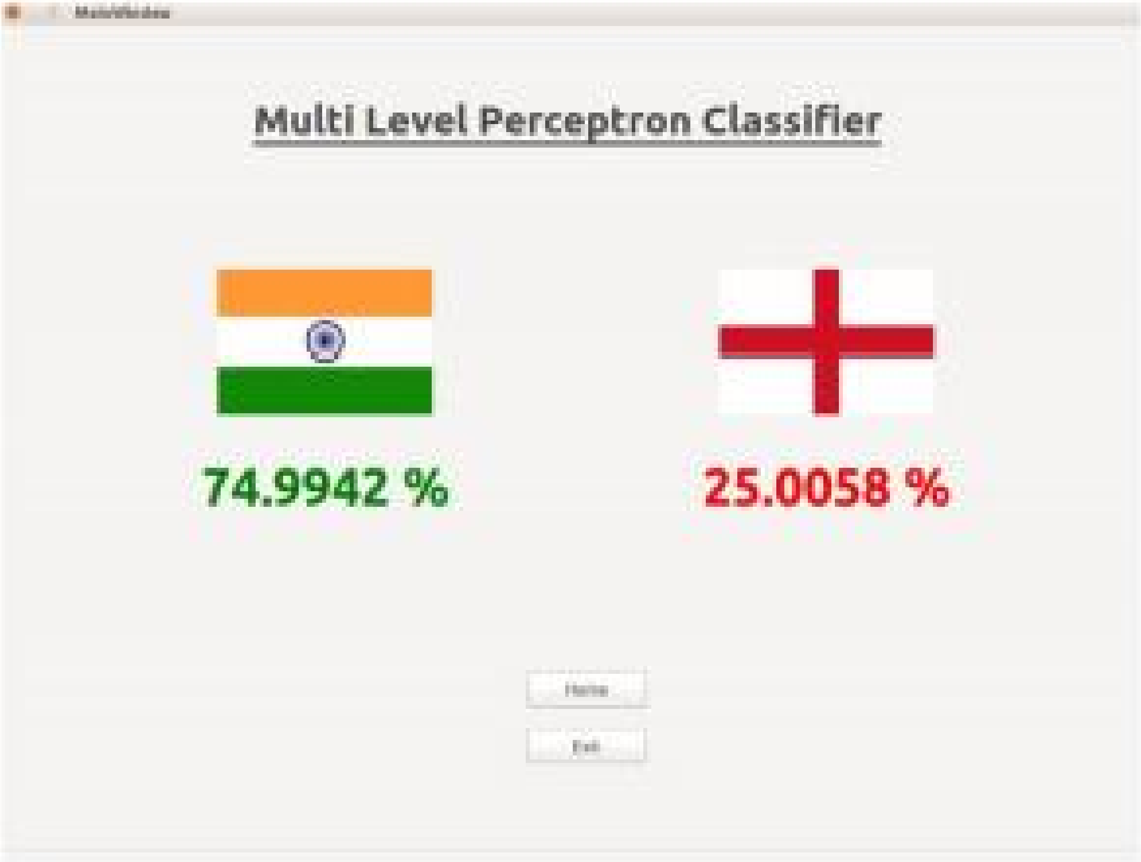
Smart Outcome Prediction for Cricket

Project Snapshots



DT Result

Project Snapshots



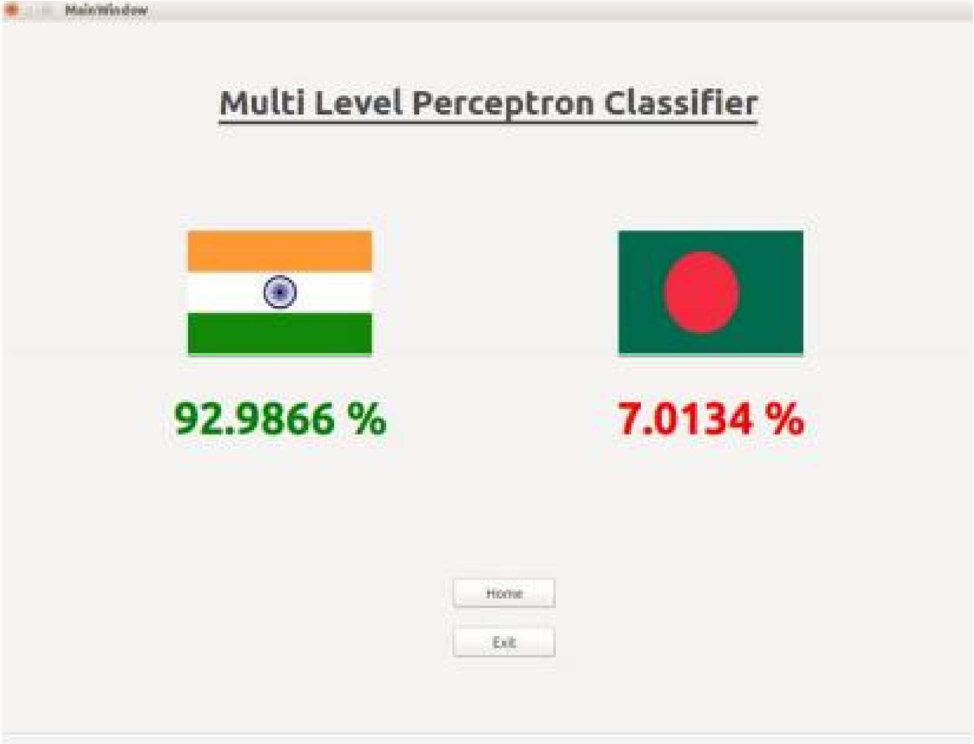
Multi Level Perceptron Classifier

74.9942 %

25.0058 %

MLP Result

Project Snapshots



Multi Level Perceptron Classifier

92.9866 %

7.0134 %

MLP Prediction

REFERENCES

- ESPN Cricinfo, http://www.stats.espncricinfo.com
- Scikit learn, http://scikit-learn.org/stable/index.html
- Wikipedia Foundation https://en.wikipedia.org
- Medium.com https://medium.com/machine-learning-101/
- Parag Shah and Mitesh Shah, ”Predicting ODI Cricket Result”. Journal of Tourism, Hospitality and Sports, 2312-5179, Vol.5, 2015.