iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** IV **Month of publication:** April 2019

DOI: https://doi.org/10.22214/ijraset.2019.4369

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue IV, Apr 2019- Available at www.ijraset.com

Predicting Outcomes of Chess Endgames Using Machine Learning Algorithms

K. Balachandra Reddy[1], P. Yaswanth[2], G. Hari Chandana[3], J. Peddi Raju[4]
[1]*Adhoc Lecturer, Dept of CSE, JNTUACEP, Pulivendula, AP, India*
[2, 3, 4]*Student, Dept of CSE, JNTUACEP, Pulivendula, AP, India*

Abstract: ***The main aim is to predict the outcome of the chess endgames i.e. King-Rook vs. King and King-Rook vs. King-paw endgames. We will find the algorithm that will predict the outcome of these endgames accurately to the great extent. So our goal is the predict the outcome of the game whether the white king will win or not. We will also predict in how many moves white king will be able to win the game. Her we will use classification models. Accuracy is used as the evaluation metric and the best model is chosen based on the accuracy.***
Keywords: Prediction, Classification models, Accuracy

I. INTRODUCTION

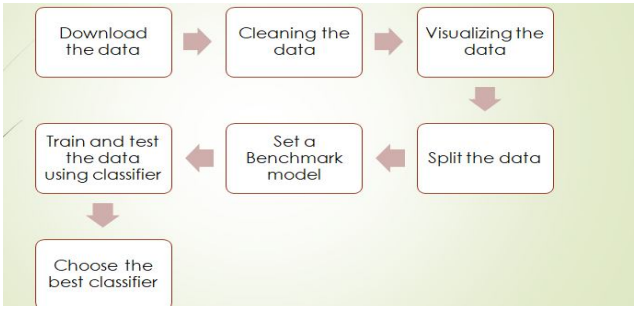
Rook and pawn endgames are the most common type of endgames there is in the game of chess. These endgames take place in about in 80% of all the games. The main idea is checkmate of black king with white king and white rook. This should be done in less than 16 steps. Chess endgames are complex domains which are enumerable. The game theoretic values stored denote whether or not positions are won for either side, or include also depth of wins (number of moves).
There will be mainly two types of outcomes in this game. The outcomes are whether the white king can win the game or cannot win the game. If the white king wins the game we can also predict that in how many moves the white king will win the game.

II. MOTIVATION

We will use this project in the final stage of the chess game i.e. king-rook vs. king and king-rook vs. king-pawn. We have developed this project by using classification algorithms that are involved in machine learning. Once we give the board positions of the white king, white rook, black king, black rook we will be able to predict whether the white king will win the game or not.

III. PROPOSED WORK

By using data mining techniques predicting the outcome of the chess endgames is a time consuming task. So in the proposed system we will use different supervised classification models to find the accuracy given by the each model and finally selects the best model which gives the highest accuracy. Some of the classification models which used are Logistic Regression, Random Forest, Support Vector Machine and Decision Tree. Task flow involved during developing of this application is



IV. METRICS

We will use accuracy score as evaluation metric to predict the outcome of chess end games. It is defined as the number of correct predictions made as a ratio of all predictions made. Accuracy is most common evaluation metric for classification problems.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue IV, Apr 2019- Available at www.ijraset.com

V. DATASET

In our project we consider two types of chess end games one is King-Rook vs. King and other is King-Rook vs. King-Pawn. For King-Rook vs. King: The dataset is comprise of multivariate data and it has both categorical and integer values. There will be total of 28056 instances and the total number of attributes is 6 they are White King File(Column), White King Rank(Row), White Rook File, White Rook Rank, Black King File, Black King Rank, Class(Target attribute).

For King-Rook vs. King-Pawn: The dataset is comprise of multivariate data and it has both categorical and integer values. There will be total of 3196 instances and the total number of attributes is 36 these are from feature value list that appear in every King-Rook vs. King-Pawn end games. Some of the feature values are Bkblk: Black King Block, Stlmt: Stalemate, Wkpos: White King Position …etc. Target attribute is Classes(2): White-Can-Win("won"),White-Cannot-Win ("nowin").

VI. DATA PREPROCESSING

Some of the datasets contain irrelevant information and noisy data. These datasets should be handled properly to get a better result. Data preprocessing includes data cleaning, transformation and dimentionality reduction which convert the raw data into a form that is suitable for further processing.

The first step in data preprocessing is read the dataset, it is done by using read_csv(). And the shape of the dataset will helps to understand briefly about the size of the data. For King-Rook vs. King:

```
# know the shape of the data
df.shape
```

```
(28056, 7)
```

For King-Rook vs. King-Pawn

```
df.shape
```

```
(3196, 37)
```

First five instances of the dataset For King-Rook vs. King:

```
# using head() print the first five columns of the data
df.head()
```

	wkf	wkr	wrf	wrr	bkf	bkr	class
0	a	1	b	3	c	2	draw
1	a	1	c	1	c	2	draw
2	a	1	c	1	d	1	draw
3	a	1	c	1	d	2	draw
4	a	1	c	2	c	1	draw

For King-Rook vs. King-Pawn:

```
df.head()
```

	bkblk	bknwy	bkon8	bkona	bkspr	bkxbq	bkxcr	bkxwp	blxwp	bxqsq	...	spcop	stlmt	thrsk	wkcti	wkna8	wknck	wkovl	wkpos	wtoeg	class
0	f	f	f	f	f	f	f	f	f	f	...	f	f	f	f	f	f	t	t	n	won
1	f	f	f	f	t	f	f	f	f	f	...	f	f	f	f	f	f	t	t	n	won
2	f	f	f	f	t	f	t	f	f	f	...	f	f	f	f	f	f	t	t	n	won
3	f	f	f	f	f	f	f	f	t	f	...	f	f	f	f	f	f	t	t	n	won
4	f	f	f	f	f	f	f	f	f	f	...	f	f	f	f	f	f	t	t	n	won

5 rows x 37 columns

To find any missing values present in the dataset we will use info(). For King-Rook vs. King:

```
# using info() find the information of the attributes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28056 entries, 0 to 28055
Data columns (total 7 columns):
wkf      28056 non-null object
wkr      28056 non-null int64
wrf      28056 non-null object
wrr      28056 non-null int64
bkf      28056 non-null object
bkr      28056 non-null int64
class    28056 non-null object
dtypes: int64(3), object(4)
memory usage: 1.5+ MB
```

For King-Rook vs. King-Pawn:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3196 entries, 0 to 3195
Data columns (total 37 columns):
bkblk    3196 non-null object
bknwy    3196 non-null object
bkon8    3196 non-null object
bkona    3196 non-null object
bkspr    3196 non-null object
bkxbq    3196 non-null object
bkxcr    3196 non-null object
bkxwp    3196 non-null object
blxwp    3196 non-null object
bxqsq    3196 non-null object
cntxt    3196 non-null object
dsopp    3196 non-null object
dwipd    3196 non-null object
hdchk    3196 non-null object
katri    3196 non-null object
mulch    3196 non-null object
qxmsq    3196 non-null object
r2ar8    3196 non-null object
reskd    3196 non-null object
reskr    3196 non-null object
rimmx    3196 non-null object
rkxwp    3196 non-null object
rxmsq    3196 non-null object
simpl    3196 non-null object
skach    3196 non-null object
skewr    3196 non-null object
```

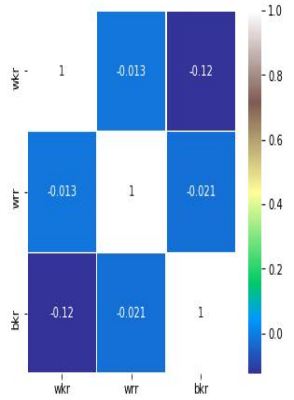
There is no missing or null values present both the datasets.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue IV, Apr 2019- Available at www.ijraset.com

VII. DATA VISUALIZAION

Data Visualization provides an important suite of tools for gaining a qualitative understanding .This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers. By using visualization we will easily find the correlation between different attributes in the dataset.

```
# we will now plot the heatmap to know about the correaltion between the attributes.
# import matplotlib.pyplot and seaborn
import matplotlib.pyplot as plt
import seaborn as sns
#heatmap to find the correlation.
sns.heatmap(df.corr(),annot=True,cmap='terrain',linewidths=0.1)
fig=plt.gcf()
fig.set_size_inches(5,5)
plt.show()
```



The above heat map shows the correlation between different attributes in the dataset.

VIII. MODELLING

In order to model the data we will first divide the entire dataset into training and testing datasets. 70 percent of the data comes under training data and the rest of the data goes into testing data set.

IX. ALGORITHMS AND TECHNIQUES

For the King-Rook vs. King dataset we apply three apply three supervised classification algorithms LogisticRegression (Bench mark model), Random Forest, Support Vector Machine.

For the King-Rook vs. King-Pawn dataset the supervised classification algorithms applied are: Logistic Regression (Benchmark model), Random Forest, Decision Tree.

1) *Logistic Regression:* Logistic regression is one of the supervised classification algorithm used for binary classification. It gives a discrete binary outcome between 0 and 1. This algorithm works by measuring the relationship between the dependent variable and one or more independent variables Advantage: Outputs have a nice probabilistic interpretation and the algorithm can be regularized to avoiding overfitting. Disadvantage: Logistic regression tends to underperform when there are multiple or non linear decision boundaries.

```
# import LogisticRegression from sk.model_selection
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
# fit the training data
lr.fit(X_train, y_train)
# predict the training data
pred = lr.predict(X_train)
# find the accuracy score of training data
print('\nLogistic Regression - Train accuracy', round(accuracy_score(y_train, pred), 3))

pred = lr.predict(X_test)
# find the accuracy score of testing data
print('\nLogistic Regression - Test accuracy', round(accuracy_score(y_test, pred), 3))
```

2) *Random Forest:* Random Forest is a predictive modeling algorithm which is used for both classification and regression tasks. It works well with default hyper parameter. It can be used to rank the importance of variables in a regression or classification problem. Advantage: Reduction in overfitting by averaging several trees, there is significantly lower risk of overfitting Disadvantage: It takes more time to train the samples.

```
# import RandomForestClassifier from sklear.ensemble
from sklearn.ensemble import RandomForestClassifier
rf_fit = RandomForestClassifier(n_estimators=1000,random_state=1)
# fit the training data
rf_fit.fit(X_train, y_train)

# find the accuracy score of trained data
print('\nRandom Forest - Train accuracy', round(accuracy_score(y_train, rf_fit.predict(X_train)), 3))

 # find the accuracy score of  testing data
print('\nRandom Forest - Test accuracy', round(accuracy_score(y_test, rf_fit.predict(X_test)), 3))
```

3) *Support Vector Machine:* Support vector Machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space which can be used for classification, regression or other tasks like outlier detection. Advantage: It works very well with clear margin of separation and it is effective in high dimensional spaces. Disadvantage: It doesn't perform well when we have large dataset because the required training time is higher.

```
# import svm from sklearn
from sklearn import svm # Support vector machine

rbfSVM = svm.SVC(kernel='rbf', C=1, gamma=0.1)
#fit the training data
rbfSVM.fit(X_train, y_train)
# find the accuracy of trained data
pred = rbfSVM.predict(X_train)
print('\nrbf SVM - Train accuracy: ', round(accuracy_score(pred, y_train), 3))
# find the accuracy score of testing data
pred = rbfSVM.predict(X_test)
print('\nrbf SVM - Test accuracy: ', round(accuracy_score(pred, y_test), 3))
```

4) *Decision Tree Algorithm:* It is one the most popular machine learning algorithm used for both classification and regression tasks and it is often mimic the human level thinking so it is very simple to understand the data and make some good interpretations Advantage: Decision Trees are able to generate understand the rules and performing classification without requiring much computation. Disadvantage: Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.

```
# Decision Tree Classifier
from sklearn.tree import DecisionTreeClassifier

# parameters selected for DT classifier
# criterion='gini', max_depth=5, min_samples_split=2, min_samples_leaf=1
dt_fit = DecisionTreeClassifier(criterion='gini', max_depth=5, min_samples_split=2, min_samples_leaf=1, random_state=42)
dt_fit.fit(X_train, y_train)


from sklearn.metrics import accuracy_score, classification_report
print("\nDecision Tree - Train accuracy\n\n", round(accuracy_score(y_train, dt_fit.predict(X_train)), 3))

print("\nDecision Tree - Test accuracy", round(accuracy_score(y_test, dt_fit.predict(X_test)), 3))
```

X. RESULTS

The accuracy scores given by the classification algorithms are: For King-Rook vs. King end game

Model name	Accuracy
Logistic Regression	0.358
SVM	0.654
Random Forest	0.737

For King-Rook vs. King-Pawn end game

Model name	Accuracy
Logistic Regression	0.951
Decision Tree	0.932
Random Forest	0.977

From the above results we conclude that Random Forest algorithm gives the best accuracy score in both the end games and it is the best algorithm when compared with the others.

XI. REFINEMENT

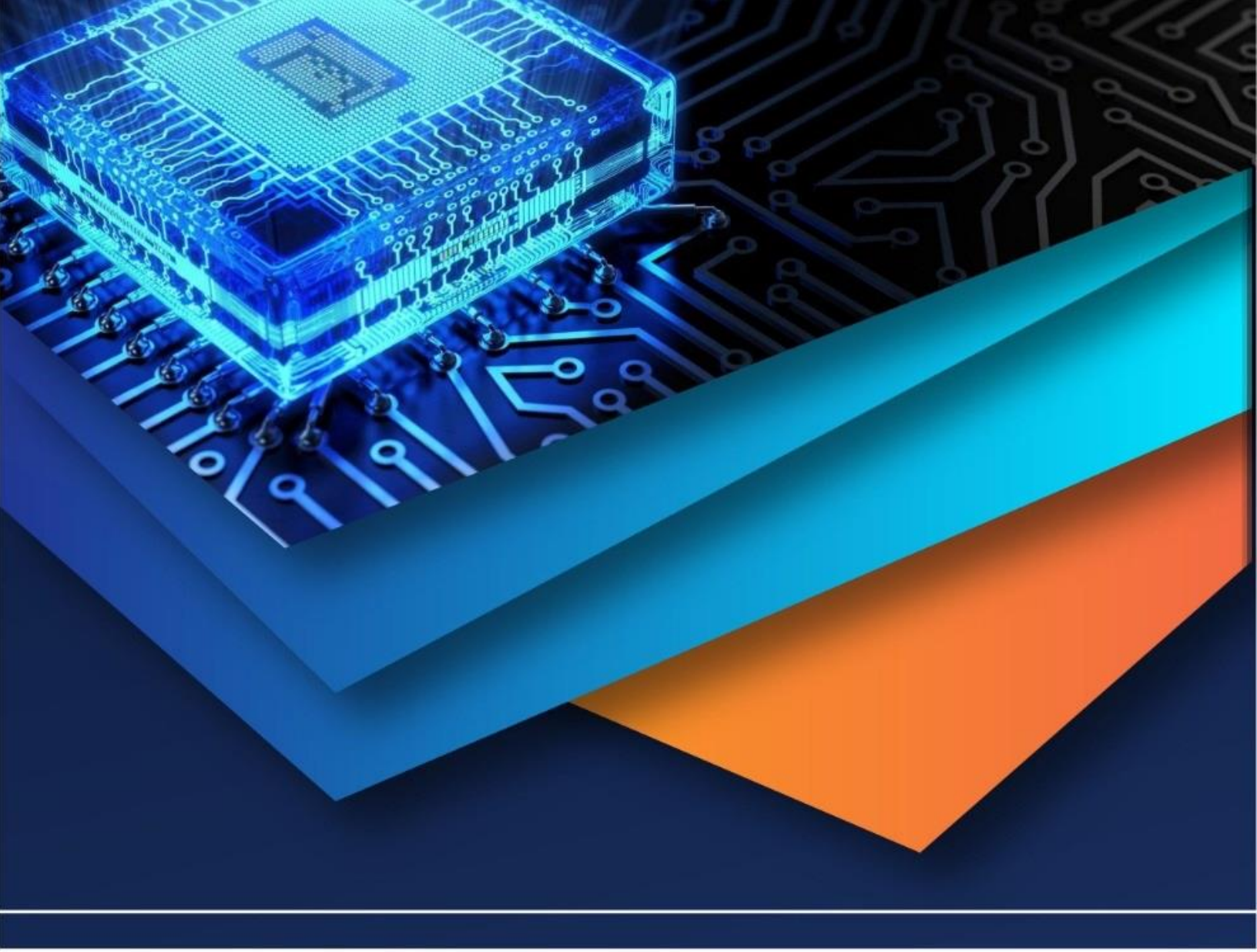
We will refine the models by using GridSearchCV. And we will tune the parameters are 'n_estimators'= [450,500] and 'criterion'= ['gini','entropy']. After tuning the final accuracies are: For King-Rook vs. King : 0.7379 For King-Rook vs. King-Pawn : 0.9896

XII. CONCLUSION

By using machine learning algorithms we can easily predict the outcomes of chess endgames. These models are trained with the dataset and gives the accuracy score as results. From all the algorithms Random Forest algorithm gives the best accuracy score in King-Rook vs. King and King-Rook vs. King-Pawn end games so it is selected as best algorithm from the others.
This application can be further developed for other endgames like Tic-Tac-Toe endgame and many more.

REFERENCES

[1] https://archive.ics.uci.edu/ml/datasets/Chess+%28King-Rook+vs.+King%29
[2] https://archive.ics.uci.edu/ml/datasets/Chess+%28King-Rook+vs.+King-Pawn%29
[3] https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd
[4] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/
[5] https://www.ritchieng.com/machine-learning-decision-trees/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY