www.ijecs.in

International Journal Of Engineering And Computer Science
Volume 10 Issue 11 November 2021, Page No. 25431-25441
ISSN: 2319-7242 DOI: 10.18535/ijecs/v10i11.4635

Implementation of Classification Algorithms in Neo4j using IPL data

Surajit Medhi[1], Hemanta K. Baruah[2]
[1]Department of Computer Science, Gauhati University, Assam, India
[2]Gahuhati University, Assam, India
surajitmdh@gmail.com

Abstract: The main objective of this paper is to implement the classifications algorithms in Neo4j graph database using cypher query language. For implementing the classification algorithm, we have used Indian Premier League (IPL) dataset to predict the winner of the matches using some different features. The IPL is the most popular T20 cricket league in the world. The prediction models are based on the city where the matches were played, winner of the toss and decision of the toss. In this paper we have implemented Naïve Bayes and K-Nearest Neighbors (KNN) classification algorithms using cypher query language. Different classifiers are used to predict the outcome of different games like football, volleyball, cricket etc, using python and R. In this paper we shall use cypher query language. We shall also compare and analysis the results which are given by Naïve Bayes and K-Nearest Neighbors algorithms to predict the winner of the matches.

Keywords: Neo4j, IPL, cypher query language, Naïve Bayes, KNN

INTRODUCTION:

Cricket is a popular game in India as well as in other countries. The game was started as a test match format and test match is played for five days and each team has chance to bat two times as well as for bowling. After that one day cricket is introduced and it is a 50 over game which is played in day and day-night format. Then T20 Cricket is introduced and now a day it gets more popularity among all three formats. IPL is a popular T20 cricket league all over the world. In India during March or April and May, IPL is conducted by BCCI every year. The first season was started in Delhi during 2008. BCCI vice president Lalit Modi was the main person who introduced the IPL and also described the revenue system, prize money, tournament format and number of teams and venues. Indian Premier League is now a day's very popular cricket game all over the world. In IPL there are different teams and each team has Indian batsmen and bowlers as well as overseas players. Each team has played in their home ground as well as outside the home ground. In 2009 IPL was held in South Africa. In 2014 some of the matches were played in UAE and the complete edition of 2020 IPL, also known as IPL 13 was held in UAE because of covid-19 pandemic.

The information Graph Databases is NOSQL databases and they use graph data structure to represent the data as nodes and edges. Key-value databases, column-family stores databases, document-oriented databases and graph databases are the different types of NOSQL [1]. DEX, Infinite graph, Neo4j, Orient DB, Titan are the different types of graph databases are developed to store and retrieve as a graph structure over a traditional relational databases[2]. In our works we have used Neo4j graph database. Many of the Researchers are used R, Python language as well as Matlab and WEKA software. But in these technologies we do not need to write the overall program to implement the different classification algorithms. In our work we have written the code using cypher query language for implementing Naïve Bayes and KNN classification algorithm.

Neo4j [3] is the open source graph database and support labeled property graph model. For storing data, Neo4j uses nodes, edges and properties. It provides index - free adjacency that means each node is a pointer. A labled property graph model contains nodes and relationships with properties and node can be labled with more than one label. In this model, let's say we want to model a employee department relationship then we have to create different nodes for employees and departments and connect them with a relation works_in with edges. For each employee and department we have set the properties as attributes and also for works_in relation. Neo4j provides visualization of all information as a graph with contains nodes and edges with properties. Cypher query language is a query language which is used by Neo4j to store, retrieve, manipulating and searching information.

LITERATURE REVIEW:

Several researchers have used different machine learning algorithms to predict and analysis the winner of the cricket matches. Lots of prediction models are developed using different classification models to predict the winner of the IPL matches.

S Agrawal et al. [4] used three different machine learning algorithms to predict the winner of a match. Gagana S .cc [5] have used Recurrent Neural Network (RNN) and Hidden Markov Model(HMM) to build the predictive model for outcome of IPL matches. The model provided a good prediction accuracy. Prince Kansal et al. [6] proposed several prediction models using different data mining algorithms such as Naïve Bayes, Decision Tree and Multilayer Perceptor(MLP) to predict player selection in IPL based on players past performance. The best accuracy among all other algorithm was given by MLP. Sasank et al. [7] dynamically predits the result in the second innings of an IPL match using the batsman and bowler ratings. Chellapilla et al. [8] developed different approaches to predict of IPL season 9. Amal Chaminda Kaluarachichi et al. [9] examined that classification is the best method to predict the winner of a match. M G Jhanwar .cc [10] used the K-Nearest Neighbor algorithm to predict the winner of matches. The authors also create a model about the players using their past data as well as current form. Harsit Barot et al. [11] used the Support Vector machine (SVM), Decision Tree, Random Forest, Logistic Regression algorithms to predict the winner of a match. They used some crucial factors like team form, batsman and bowler ratings etc. logistic Regression gave the best accuracy rate. Stylianos Kampakis .cc [12] have used machine learning models to predict the outcome of English County twenty over cricket matches.

METHODOLOGY:

3.1 Dataset collection

The dataset was collected from kaggle website [13]. The dataset contains two files namely deliveres.csv and matches.csv. We have used the matches.csv file. This file contains the summary of each IPL match from the year 2008 to 2019 which includes playing team, the city, winner of the toss, decision of the toss, winner of the match and more.

3.2 Data preprocessing and feature selection

The dataset contains the following attributes-

Id, season, city, date, team1, team2, toss_winner, toss_decision, result, dl_applied, winner, win by run, win by wicket, player of match, venue, umpire1, umpire2, umpire3.

From the above attributes we have taken season, city, team1, team2, toss_winner, toss_decision, result and winner.

In our work, we have considered the following features-

Home city – Every IPL team has home ground and they always want to take advantages from the home city because they know the ground properly and also take the home crowd advantages. In Home ground home teams are supported by lots of crowd.

Toss_winner- Winner of the toss is also an important feature because home team knows the pitch better than the other teams.

Toss_decision- Decision of the toss is also an important feature because according to the pitch and weather condition, every team will decide whether they choose batting or bowling.

In 2009, IPL was hosted by South Africa so that we have deleted the matches which are played in 2009. After that in 2014 some of the matches are played in UAE and we have deleted the matches also. We have deleted the matches which are not played in their home city as well as home ground. Therefore, in our work we are not considered some of the data from our dataset

From the year 2008 to 2010 eight teams were played in IPL. They are -

Chennai Super King (CSK), Kolkata Knight Riders (KKR), Rajasthan Royals (RR), Mumbai Indians (MI), Deccan Chargers (DC), Kings XI Punjab (KXIP), Royal Challengers Bangalore (RCB), Delhi Daredevils (DD). In 2011, two new teams were added to the IPL namely Pune Warriors and Kochi Tuskers Kerela but in 2012 Kochi Tuskers Kerela was not part of the IPL and Pune warriors team also was not part of the IPL from 2014 so that we have deleted the Kochi Tuskers Kerela's and Pune Warriors matches. CSK and RR were released by BCCI in the year 2016 and two new teams were added namely Rising Pune Supergiant (RPS) and Gujrat Lions (GL). But again CSK and RR were joined the IPL in the year 2018. RPS and GL were released by BCCI. So we are not considered the RPS and GL matches. IPL 13 was held in UAE so that we are not considered the IPL 13 matches. Overall we have considered the eight IPL teams. Deccan Charges team was renamed as Sunrisers Hyderabad (SRH) in 2013 and Delhi Daredevils was renamed as Delhi Capitals (DC) in 2019. Finally we have considered the matches from 2008 to 2019 season except the matches which are played in South Africa and some matches are played in UAE in 2014. In our dataset team Deccan Chargers is renamed as Sunrisers Hyderabad (SRH) and Delhi Daredevils is renamed as Delhi Capitals (DC).

3.2 Naïve Bayes Classification Algorithm in Neo4j Graph Database

3.2.1 Naïve Bayes Classifier using percentage split:

Naïve Bayes classifier consists of different classification algorithms based on Bayes Theorem. Here we have implemented the Naïve Bayes classifier based on Bayes theorem. Bayes theorem is defined as P(A|B)=P(B|A).P(A)/P(B). For implementing this algorithm we have used the IPL dataset as mentioned above. We have used different steps for implementing this algorithm in Neo4j graph database using cypher query language.

Step1: We have divided our dataset into two parts-

i) All the matches from season 2008 to 2017 for training.

ii) All the matches from season 2018 to 2019 for testing.

Step2: We have considered eight teams in our work so that we divide the matches for every team with their opposition team.

Step3: We have implemented Naïve Bayes algorithm for every team's matches with their opposition team.

Step4: For example we take CSK and KKR matches then we use the following steps for training-

i) We have considered the features city, toss_winner and toss decision and our target value is winner of the matches.

ii) Then we find how many matches are played between CSK and KKR and how many matches are won by CSK and KKR. After that we find the winning probability of CSK with the help of matches won by CSK divided with total number of matches. Same formula is used for KKR also.

iii) After that we have done expansion with the different features. First we have taken the city feature. City feature consist of two values namely Chennai and Kolkata and then we find how many matches are played in Chennai and Kolkata. After that we find how many matches are won by CSK and KKR in Chennai and also find the winning probability of CSK and KKR with the help of matches won by CSK which are played in Chennai divided by total number of matches won by CSK and the matches won by KKR which are played in Chennai divided by total number of matches won by KKR respectively. Same formula is used for the matches that are played in Kolkata. For toss_winner feature, we find how many tosses are won by CSK and KKR. After that we find how many matches are won by CSK or KKR when the tosses are won by CSK. And then we find the winning probability of CSK and KKR with the help of matches won by CSK when the tosses are won by CSK divided by total number of matches won by CSK and matches won by KKR when the tosses are won by CSK divided by total number of matches won by KKR respectively. Same formula is used for the matches when KKR won the tosses. Same technique is used for toss decision feature also.

Step5: After that we have used the matches which are played in season 2018 and 2019 for testing-

i) For testing we have considered each and every match played between CSK and KKR.

ii) Then we find the winning probability of CSK and KKR using their training values.

iii) After that we have compared the value of CSK and KKR and if Probability value of CSK is greater than KKR then result goes to CSK.

Step6: We have used same technique for every matches played between different teams.

3.2.2 Naïve Bayes classifier using K-fold cross validation:

In our work we have used 4-fold cross validation for implementing Naïve Bayes classifier in Neo4j graph database using cypher query language. In first fold we have used the matches that are played in season 2008 to 2017 for training and the matches which are played in season 2018 to 2019 for testing. In second fold we have used the matches that are played in season 2008, 2010 to 2016 and 2019 in training and the matches which are played in season 2017 and 2019 for testing. In third fold we have used the matches that are played in season 2011 to 2019 for training and the matches that are played in season 2008 and 2010 for testing. In fourth fold we have used the matches that are played in season 2008 to 2014 and 2017 to 2019 for training and the matches that are played in season 2015 and 2016 for testing. Same method for cross validation is used for each and every match.

3.3 K-Nearest Neighbor Classification Algorithm in Neo4j Graph Database:

3.3.1 K-Nearest Neighbor Classifier using percentage split:

In our work we have implemented the KNN classification algorithm in Neo4j using cypher query language. For implementation the KNN we have divided our dataset in two parts namely training and testing. For training and testing we have considered the matches that are played in season 2008 to 2017 and in season 2018 to 2019 respectively. In our model we have used the KNN algorithm for every match played between each other team. For example we consider the matches which are played between CSK and KKR. We have done binary encoding for the different features to implement the KNN classification. After that we find the Euclidian distance for every match played in training part with the matches played in testing part. The formula for Euclidian distance is √ ((x1-x2) ^2+ (y1-y2) ^2+ (z1-z2) ^2) where x1,y1 and z1 are used the value of the features in training part of the matches and x2,y2and z2 are used the value of the features in testing part. After that we sort the distances. We consider the value of k is 3 for every match. After that according to Euclidian distance we get the target value and then compare the target value with the target value of the testing part. Then we find the accuracy of the winner prediction. These steps are repeated for each and every match.

3.3.2 K-Nearest Neighbor Classifier using K-fold cross validation:

We have considered the 4-fold cross validation for implementing the KNN classification algorithm. Same training and testing part of matches for k-fold cross validation which is used in Naïve Bayes classifier for K-fold cross validation in KNN classification algorithm. Then we implement the KNN classification algorithm for training and testing for every match. For every fold we find the Euclidian distance between the matches that are in training part and testing part. After that we find accuracy for every fold matches and the combine the result of all 4 folds. Then we get the final accuracy of the prediction of the winner of the matches.

4. RESULT AND DISCUSSION

4.1 Result of Naïve Bayes Classifier using percentage split:

CSK MATCHES			**MI MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
CSK_DC	67	56	MI_CSK	50	42.85714
CSK_KXIP	75		MI_DC	0	
CSK_KKR	50		MI_KXIP	25	
CSK_MI	50		MI_KKR	75	
CSK_RCB	75		MI_RCB	75	
CSK_RR	75		MI_RR	25	
CSK_SRH	0		MI_SRH	50	
DC MATCHES			**RCB MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
DC_CSK	67	48.85714	RCB_CSK	75	66.71429
DC_KXIP	75		RCB_DC	75	
DC_KKR	25		RCB_KXIP	25	
DC_MI	0		RCB_KKR	75	
DC_RCB	75		RCB_MI	75	
DC_RR	50		RCB_RR	67	
DC_SRH	50		RCB_SRH	75	
KXIP MATCHES			**RR MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
KXIP_CSK	75	53.57143	RR_CSK	75	48.14286
KXIP_DC	75		RR_DC	50	
KXIP_KKR	50		RR_KXIP	75	
KXIP_MI	25		RR_KKR	20	
KXIP_RCB	25		RR_MI	25	
KXIP_RR	75		RR_RCB	67	
KXIP_SRH	50		RR_SRH	25	
KKR MATCHES			**SRH MATCHES**		

Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
KKR_CSK	50	47.85714	SRH_CSK	0	41.42857
KKR_DC	25		SRH_DC	50	
KKR_KXIP	50		SRH_KXIP	50	
KKR_MI	75		SRH_KKR	40	
KKR_RCB	75		SRH_MI	50	
KKR_RR	20		SRH_RCB	75	
KKR_SRH	40		SRH_RR	25	

4.1 Result of Naïve Bayes Classifier using k-fold cross validation:

CSK MATCHES			**MI MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
CSK_DC	54	58.57143	MI_CSK	55	60.42857
CSK_KXIP	59		MI_DC	67	
CSK_KKR	64		MI_KXIP	53	
CSK_MI	55		MI_KKR	81	
CSK_RCB	58		MI_RCB	64	
CSK_RR	67		MI_RR	45	
CSK_SRH	53		MI_SRH	58	
DC MATCHES			**RCB MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
DC_CSK	54	55.28571	RCB_CSK	58	55.42857
DC_KXIP	46		RCB_DC	48	
DC_KKR	58		RCB_KXIP	39	
DC_MI	67		RCB_KKR	77	
DC_RCB	48		RCB_MI	64	
DC_RR	56		RCB_RR	40	
DC_SRH	58		RCB_SRH	62	
KXIP MATCHES			**RR MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
KXIP_CSK	59	54.57143	RR_CSK	67	51
KXIP_DC	46		RR_DC	56	
KXIP_KKR	58		RR_KXIP	69	
KXIP_MI	53		RR_KKR	48	
KXIP_RCB	39		RR_MI	45	
KXIP_RR	69		RR_RCB	40	
KXIP_SRH	58		RR_SRH	32	
KKR MATCHES			**SRH MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
KKR_CSK	64	64.71429	SRH_CSK	53	55.42857
KKR_DC	58		SRH_DC	58	
KKR_KXIP	58		SRH_KXIP	58	
KKR_MI	81		SRH_KKR	67	
KKR_RCB	77		SRH_MI	58	
KKR_RR	48		SRH_RCB	62	
KKR_SRH	67		SRH_RR	32	

4.3 Result of K-Nearest Neighbor Classifier using percentage split:

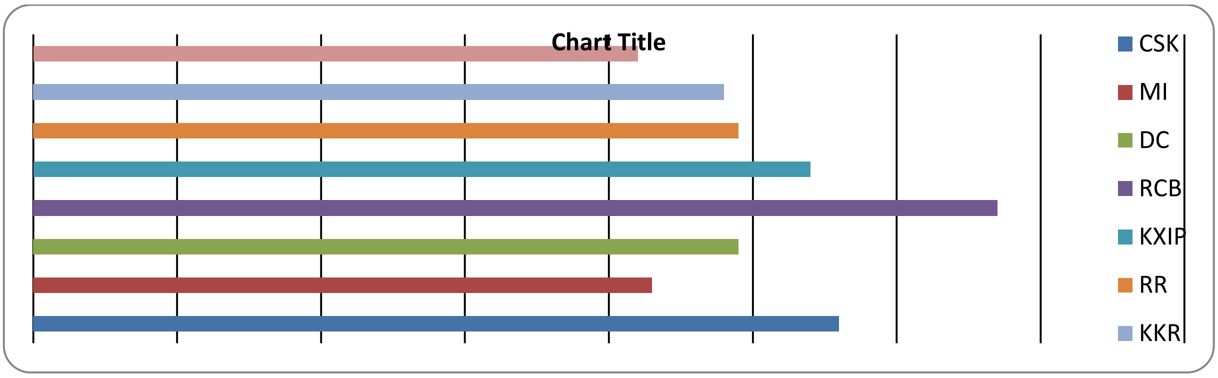
CSK MATCHES			MI MATCHES		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
CSK_DC	56	56.71429	MI_CSK	47	43.28571
CSK_KXIP	59		MI_DC	25	
CSK_KKR	67		MI_KXIP	33	
CSK_MI	47		MI_KKR	67	
CSK_RCB	59		MI_RCB	47	
CSK_RR	50		MI_RR	25	
CSK_SRH	59		MI_SRH	59	
DC MATCHES			**RCB MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
DC_CSK	56	46.28571	RCB_CSK	59	56.85714
DC_KXIP	42		RCB_DC	50	
DC_KKR	42		RCB_KXIP	25	
DC_MI	25		RCB_KKR	75	
DC_RCB	50		RCB_MI	47	
DC_RR	50		RCB_RR	67	
DC_SRH	59		RCB_SRH	75	
KXIP MATCHES			**RR MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
KXIP_CSK	67	47.85714	RR_CSK	50	47.42857
KXIP_DC	42		RR_DC	50	
KXIP_KKR	42		RR_KXIP	76	
KXIP_MI	33		RR_KKR	47	
KXIP_RCB	25		RR_MI	25	
KXIP_RR	76		RR_RCB	67	
KXIP_SRH	50		RR_SRH	17	
KKR MATCHES			**SRH MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
KKR_CSK	67	55.28571	SRH_CSK	59	52.28571
KKR_DC	42		SRH_DC	59	
KKR_KXIP	42		SRH_KXIP	50	
KKR_MI	67		SRH_KKR	47	
KKR_RCB	75		SRH_MI	59	
KKR_RR	47		SRH_RCB	75	
KKR_SRH	47		SRH_RR	17	

4.4 Result of K-Nearest Neighbor Classifier using k-fold cross validation:

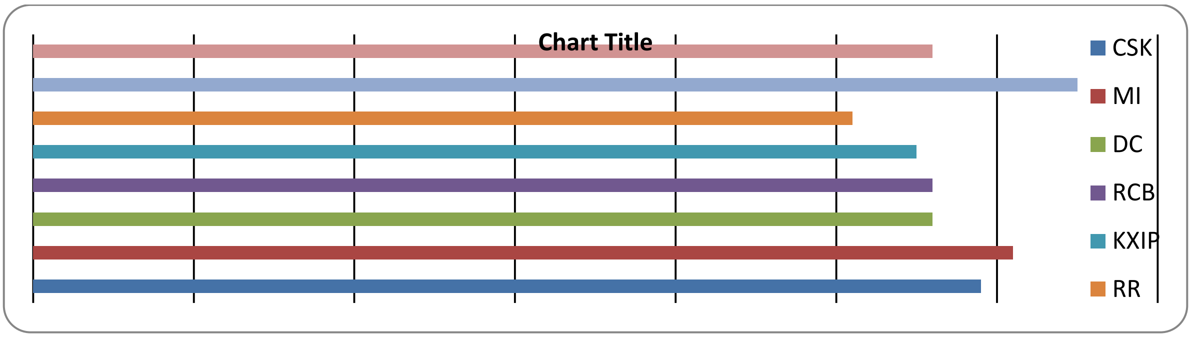
CSK MATCHES			MI MATCHES		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
CSK_DC	74	68.85714	MI_CSK	70	63.42857
CSK_KXIP	59		MI_DC	72	
CSK_KKR	74		MI_KXIP	53	
CSK_MI	70		MI_KKR	77	
CSK_RCB	64		MI_RCB	64	
CSK_RR	88		MI_RR	50	
CSK_SRH	53		MI_SRH	58	
DC MATCHES			**RCB MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
DC_CSK	74	64	RCB_CSK	64	58.71429
DC_KXIP	60		RCB_DC	64	
DC_KKR	64		RCB_KXIP	43	
DC_MI	72		RCB_KKR	72	
DC_RCB	64		RCB_MI	64	
DC_RR	56		RCB_RR	27	
DC_SRH	58		RCB_SRH	77	
KXIP MATCHES			**RR MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
KXIP_CSK	59	58.42857	RR_CSK	88	58.85714
KXIP_DC	60		RR_DC	56	
KXIP_KKR	67		RR_KXIP	69	
KXIP_MI	53		RR_KKR	65	
KXIP_RCB	43		RR_MI	50	
KXIP_RR	69		RR_RCB	27	
KXIP_SRH	58		RR_SRH	57	
KKR MATCHES			**SRH MATCHES**		
Matches	Accuracy%	Average accuracy%	Matches	Accuracy%	Average accuracy%
KKR_CSK	74	68.71429	SRH_CSK	53	60.42857
KKR_DC	64		SRH_DC	58	
KKR_KXIP	67		SRH_KXIP	58	
KKR_MI	77		SRH_KKR	62	
KKR_RCB	72		SRH_MI	58	
KKR_RR	65		SRH_RCB	77	
KKR_SRH	62		SRH_RR	57	

4.5 Result of Naïve Bayes and KNN classification algorithm in bar diagram:

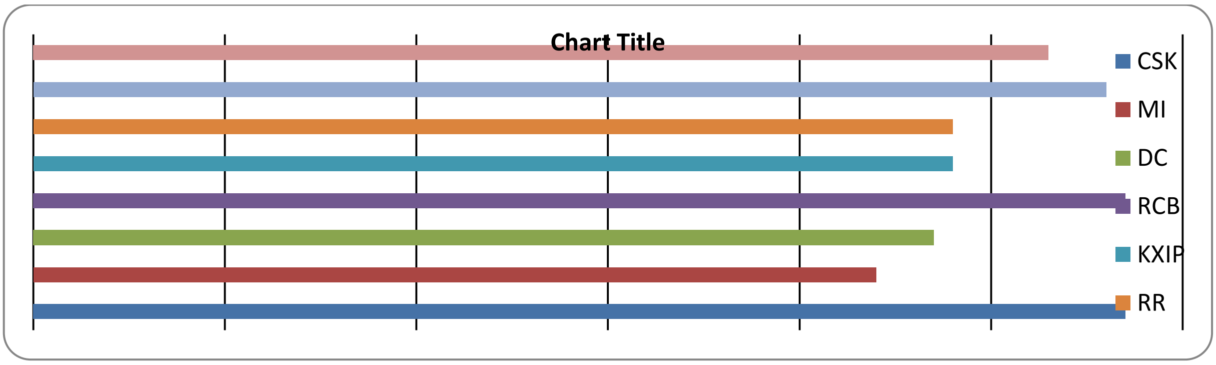
4.5.1 Result of Naïve Bayes Classifier using percentage split:



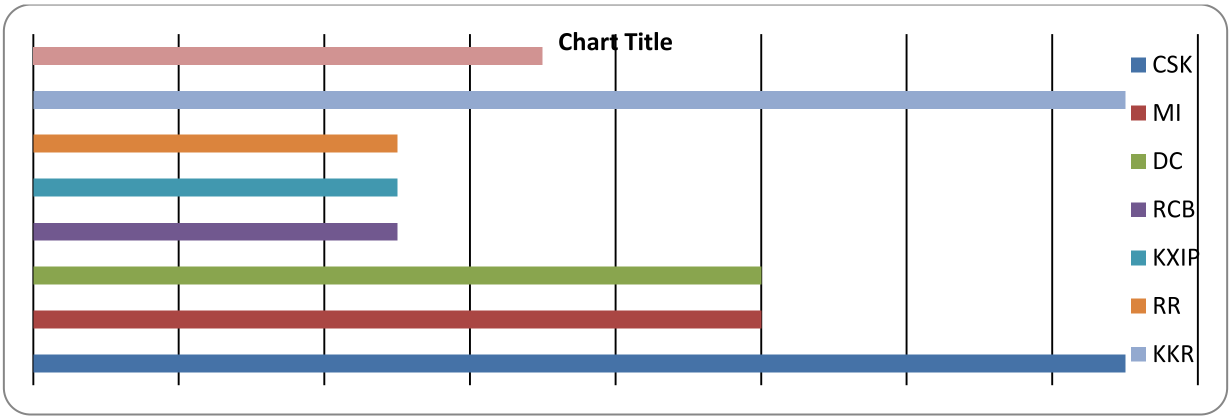
4.5.2 Result of Naïve Bayes Classifier using k-fold cross validation:



4.5.3 Result of K-Nearest Neighbor Classifier using percentage split:



4.5.4 Result of K-Nearest Neighbor Classifier using k-fold cross validation:



4.6 Comparision of Naïve Bayes and KNN classification algorithm:

In this part we have compared the winning accuracy of different teams using Naïve Bayes and KNN classification algorithm in the following table:

TEAM	Win accuracy % using NB percentage split	Win accuracy % using NB k-fold cross validation	Win accuracy % using KNN percentage split	Win accuracy % using KNN k-fold cross validation
CSK	56	59	57	69
MI	43	61	44	64
DC	49	56	47	64
RCB	67	56	57	59
KXIP	54	55	48	59
RR	49	51	48	59
KKR	48	65	56	69
SRH	42	56	53	61

Table: Comparision of Naïve Bayes and KNN classification algorithm

From the above table we can say that for every team except RCB, KNN classification algorithm using k-fold cross validation gives the better accuracy. For RCB, NB classification using percentage spilt gives the better result.

CONCLUSION AND FUTURE WORK

In our work we have implemented the Naïve Bayes and KNN classification algorithm using IPL dataset in Neo4j graph database. We have used cypher query language to implement the algorithms. We have used percentage split and k-fold cross validation techniques to implement the algorithms and all these techniques are implemented by cypher query language. Percentgae spilt and k-fold cross validation tencniques are used deifferently in different algorithms and used for different combinations in all eight teams played in IPL from 2008 to 2019. After that we have compared the results of winning percentage of all eight team that are given by Naïve Bayes and KNN classification algorithms using different techniques. The KNN classification algorithm using k-fold cross validation has given us better result for all teams except RCB. From the above results IPL fan followers shall predict the winner of the matches. In future we shall implement these algorithms with different features like batting and bowling performance for individual player using cypher query language in Neo4j graph database. And also we shall implement the different classifer like SVM, Decision tree using cypher query language in Neo4j.

REFERENCES

[1] Gabriele Pozzani, “Introduction to NoSQL “profs.sci.univr.it/~pozzani/attachments/nosql%20 %2001%20-%20introduction.pdf, April 24, 2013.

[2] R. Angles and C. Gutierrez,” Survey of Database Models”, ACM Comput. Surv .40(1):1–39, 2008.

[3] www.neo4j.com

[4] Shilpi Agrawala, Suraj Pal Singh, Jayash Sharma, “Predicting Results of Indian Premier League T-20 Matches using Machine Learning”, 2018 8th International Conference on Communication Systems and Network Technologies (CSNT).

[5] Gagana and K Paramesha, “ A Perspective on Analyzing IPL Match Results using Machine Learning”, International Journal for Scientific Research & Development| Vol. 7, Issue 03, 2019.

[6] Prince Kansal, Pankaj Kumar, Himanshu Arya and Aditya Methaila, “Player Valuation in Indian Premier League Auction using Data Mining Technique”, IEEE, 2014.

[7] Sasank Viswanadha, Kaustubh Sivalenka, Madan Gopal Jhawar, and Vikram Pudi, “Dynamic winner prediction in Twenty20 Cricket: Based on Relative Team Strength”

DSAC, Kohli Centre on Intelligent Systems, IIIT Hyderabad, India.

[8] Chellapilla Deep Prakash, C. Patvardhan,C. Vasantha," Data Analytics based Deep Mayo Predictor for IPL-9". International Journal of Computer Applications (0975 – 8887) Volume 152 – No.6, October 2016.

[9] Amal Kaluarachchi, Aparna S. Varde, CricAI: A classification based tool to predict the outcome in ODI cricket**,** Conference: IEEE Information and Automation for Sustainability (ICIAFs),2010.

[10] Madan Gopal Jhawar, Vikram Pudi, Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (*ECML-PKDD 2016*).

[11] Harshit Barot, Arya Kothari, Pramod Bide, Bhavya Ahir, Romit Kankaria, Analysis and Prediction for the Indian Premier League, 2020 International Conference for Emerging Technology (INCET)
Belgaum, India. Jun 5-7, 2020

[12] Stylianos Kampakis, William Thomas, "Using Machine Learning to Predict the Outcome of English
County twenty over Cricket Matches", Proceedings of the 5th International Conference on Mathematics in Sport Loughborough University, U.K. 29 June – 1 July 2015

[13] www.kaggle.com