

A decorative graphic on the left side of the slide, consisting of a network of thin, light green lines and small circles, resembling a circuit board or a data network. The lines are vertical and horizontal, with some diagonal connections, and the circles are placed at various points along these lines.

CLASS PROJECTS – BIG DATA 2018

CCBD PROJECT TIMELINES

- Mid Review
 - October 26, 29, 30
 - All team members must be present for presentation.
 - Take feedback on project
 - Presentation format is shared on Google Drive.
- Final Project Submission
 - November 27-December 1
 - Schedule will be put up, you can select your slot.

AVAILABLE PROJECTS

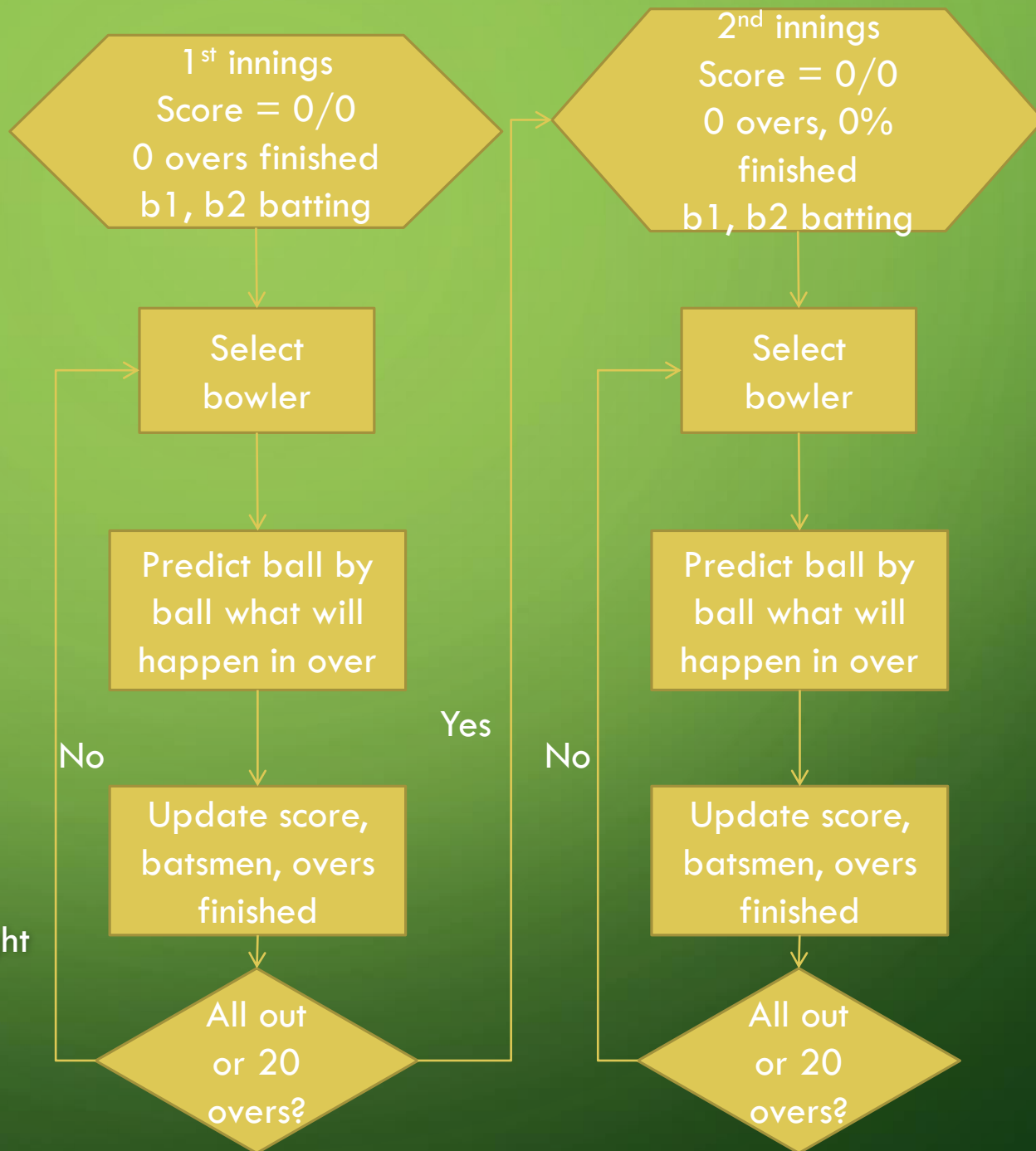
- Analysis based
 - IPL Analysis
- Coding based
 - Building a distributed key value store

A decorative graphic on the left side of the slide, consisting of a network of thin, light green lines and small circles, resembling a circuit board or a stylized tree structure, set against a dark green gradient background.

IPL ANALYSIS – CLASS PROJECT

OVERVIEW

- Given
 - 2 cricket teams
 - Batting order
 - Bowling order
 - Who bats first
- Predict the score
- How?
 - 2 approaches
 - Flowchart at right
 - Decision Tree



PREDICTING OUTCOME OF OVER

- Predict ball by ball
 - Ignore extras for now
 - Handle wickets separately
 - Ignore stage of the game
 - 1st innings – how many overs are complete
 - 2nd innings – overs complete, % target complete
- We need to calculate $p(0, 1, 2, 3, 4, 6 \text{ runs scored})$, for each ball
- This depends upon
 - Who is batting
 - Who is bowling

ESTIMATING PROBABILITIES

- Maximum Likelihood Estimator
 - $P(\text{event}) = (\text{no. of times event occurred}) / (\text{total number of events})$
- Example
 - Suppose McCleneghan bowls 40 balls to Shikhar Dhawan
 - 4 times Shikhar Dhawan hits a 4
 - $P(\text{hitting 4}) = 0.1$

SIMPLE APPROACH (MAY NOT WORK)

1. Go over all the games
2. If we find a game where
 1. Shikhar Dhawan is batting
 2. McCleneghan bowling
 3. 4 runs are scored
3. Add 1 to the number of times 4 is hit
4. Calculate probabilities

PROBLEM WITH SIMPLE APPROACH

- Many times, in IPL, we may need combinations we haven't seen before or have seen rarely
 - If it's a rare event, probability calculation may not be accurate

SOLUTION

- Cluster (group together) similar batsmen into say 10 groups
- Also cluster similar bowlers
- Calculate probability of n runs when
 - Batsman of group A is batting
 - Bowler of group X is bowling

CLUSTERING DETAILS

- Download from cricinfo.com
 - Player vs player statistics for all T20s in 2016
 - Player profiles
- Use these player profiles to cluster batsmen, bowlers
- From the player vs player statistics, calculate the probability of runs being scored in over

APPROACH TO COMPUTING #RUNS SCORED PER BALL

- Determine batsman-bowler combination (or batsman cluster-bowler cluster combination)
 - Use cluster if enough data is not there about specific batsman bowler
- Generate a random number
- Look up cumulative probability distribution of runs scored to see where the random number falls
 - Use that as the total runs scored.

- Example

- If random number val on first ball = 0.91 then look up table below.
- 0.85-1.0 is a 6, so a 6 is scored

Runs	P(Runs)	CumProb(Runs)
0	0.1	0.1
1	0.2	0.3
2	0.2	0.5
3	0.1	0.6
4	0.25	0.85
6	0.15	1.0

PLAYER VS PLAYER INFO

Get data from cricsheets

Compute player vs player profiles by
parsing the sheets.

Maybe write a MR program or a python
program – left to you.

Sri Lanka tour of England and Ireland, Only T20I: England v Sri Lanka at The Oval, May 20, 2014

[Scorecard](#) | [Commentary](#) | [Wickets](#) | [Partnership table](#) | **[Player v player table](#)** | [Over comparison](#) | [Career averages](#) | [Report](#) | [Articles \(7\)](#) | [Photos \(16\)](#) | [Videos \(5\)](#) | [Hawk-Eye](#) | [Wagon wheel](#) | [Manhattan](#) | [Worm](#) | [Run rate graph](#) | [Player v player graph](#) | [Partnership graph](#) | [Scoring shots graph](#) | [Wickets pie](#) | [Extras pie](#)

Sri Lanka in England T20I Match England v Sri Lanka



T20I no. 401 | 2014 season

Played at [Kennington Oval](#), London

Sri Lanka won by 9 runs

20 May 2014 - day/night match (20-over match)

[1st innings](#) | [2nd innings](#)

[Expand All](#)  [Collapse All](#) 

Sri Lanka - 1st innings

TM Dilshan - 1st innings

v Bowler	0s	1s	2s	3s	4s	5s	6s	7+	Dismissal	Runs	Balls	SR
JE Root	0	1	0	0	1	0	0	0		5	2	250.00
CR Woakes	1	1	0	0	1	0	0	0		5	3	166.66
HF Gurney	3	0	0	0	1	0	0	0	bowled	4	4	100.00

MDKJ Perera - 1st innings

v Bowler	0s	1s	2s	3s	4s	5s	6s	7+	Dismissal	Runs	Balls	SR
JE Root	3	0	0	0	1	0	0	0		4	4	100.00
CR Woakes	2	0	0	0	0	0	1	0		6	3	200.00
CJ Jordan	3	0	0	0	0	0	0	0	caught	0	3	0.00

PLAYER PROFILE

<http://www.espncriinfo.com/ci/content/player/50710.html>

Full name Kumar Chokshanada Sangakkara

Born October 27, 1977, Matale

Current age 37 years 302 days

Major teams Sri Lanka, Asia XI, Central Province, Colombo District Cricket Association, Deccan Chargers, Durham, ICC World XI, Jamaica Tallawahs, Kandurata, Kandurata Maroons, Kings XI Punjab, Marylebone Cricket Club, Nondescripts Cricket Club, Sunrisers Hyderabad, Warwickshire

Playing role Wicketkeeper batsman

Batting style Left-hand bat

Bowling style Right-arm offbreak

Fielding position Wicketkeeper



Like <17k

Batting and fielding averages

	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	4s	6s	Ct	St
Tests	134	233	17	12400	319	57.40	22882	54.19	38	52	1491	51	182	20
ODIs	404	380	41	14234	169	41.98	18048	78.86	25	93	1385	88	402	99
T20Is	56	53	9	1382	78	31.40	1156	119.55	0	8	139	20	25	20
First-class	235	387	28	18134	319	50.51				53	77		352	33
List A	509	481	51	18389	169	42.76				35	115		508	124
Twenty20	164	158	17	4214	94	29.88	3355	125.60	0	24	457	76	99	41

Bowling averages

	Mat	Inns	Balls	Runs	Wkts	BBI	BBM	Ave	Econ	SR	4w	5w	10
Tests	134	4	84	49	0	-	-	-	3.50	-	0	0	0
ODIs	404	-	-	-	-	-	-	-	-	-	-	-	-
T20Is	56	-	-	-	-	-	-	-	-	-	-	-	-
First-class	235		246	150	1	1/13		150.00	3.65	246.0	0	0	
List A	509	-	-	-	-	-	-	-	-	-	-	-	-
Twenty20	164	-	-	-	-	-	-	-	-	-	-	-	-

Career statistics

WICKETS

- From statistics, calculate probability of batsman being out
- Example
 - Suppose A&B are batting
 - X,Y,Z are bowling
 - $P(\text{A out when X bowling}) = 0.04$
 - $P(\text{B out when Y bowling}) = 0.06$
 - $P(\text{A out when Z bowling}) = 0.08$
- Fall of wickets
 - Ball 1: X bowls to A; $p(\text{A is not out}) = 0.96$
 - ...
 - Ball 6: X bowls; $p(\text{A is not out}) = 0.78$
 - Ball 7: Y bowls to B; $p(\text{B is not out}) = 0.94$
 - Ball 12: Y bowls; $p(\text{B is not out}) = 0.69$
 - Ball 13: Z bowls to A; $p(\text{A is not out}) = 0.78 \times 0.92 = 0.72$
 - Ball 17: Z bowls; $p(\text{A is not out}) = 0.51$
 - Ball 18: Z bowls; $p(\text{A is not out}) = 0.47 < 0.5$
 - Assume wicket falls on 18th ball

APPROACH 2: USING A DECISION TREE

- For each over of all the matches in the past
 - Based on the two batsman cluster playing, the bowler cluster, and their stats (avg score, strike rate, wickets, economy rate), ground information,
 - Train a decision tree model using Spark MLlib(inputs are the parameters, outputs – runs scored and wicket taken)
 - <https://spark.apache.org/docs/latest/ml-classification-regression.html#decision-trees>
- For the given match, iterate for all the overs of the match
 - Using the decision tree model built earlier
 - Generate the number of runs scored and wickets lost. Assume that batsman get out in first come first out basis
 - Stop if all wickets are taken
- Compute total score in all overs. Repeat for second innings. Compare scores against actual scores of the match.

STEP 1

- Load player data into HDFS, cluster batsmen, bowlers into groups
 - Spark MLlib: <http://spark.apache.org/docs/latest/mllib-clustering.html#k-means>
- Due date
 - Show by October 30th
 - What are the clustering criteria?
 - How many clusters did you get?
 - -1 mark for each day late

STEP 2

- Load group vs group statistics, simulate match using approach 1
- Due date
 - November 12th
 - You should show us one IPL 2018 match of your choice
 - We will ask you to simulate a random IPL 2018 match
 - Show comparison for any 10 matches of your choices to demonstrate accuracy.
 - -1 mark for each day late

STEP 3

- Simulate match score using a decision tree.
- Due date
 - November 27th – December 1
 - You should show us one IPL 2018 match of your choice
 - We will ask you to simulate a random IPL 2018 match
 - Show comparison for any 10 matches of your choices to demonstrate accuracy.
 - Compare accuracy with previous Approach 1
 - What parameters did you use for the decision tree? How does it affect accuracy?
 - -1 mark for each day late

A decorative graphic on the left side of the slide, consisting of a network of thin, light green lines and small circles, resembling a circuit board or a data network topology. The lines are vertical and horizontal, with some diagonal connections, and the circles are placed at various points along these lines.

BUILDING A DISTRIBUTED KEY VALUE STORE

BIG DATA 2018— CLASS PROJECT



OBJECTIVE

- Get a practical insight into building a distributed key-value store
- 
- 
- 

WHAT IS A KEY-VALUE STORE?

- Simply put a store that is capable of storing data indexed by a key
 - Key is a string of characters
 - Value is a string of characters
 - However, the value is a JSON object

SO WHAT IS DISTRIBUTED

- The data is stored across multiple servers
- Each server is responsible for part of the keys
 - It will be the primary server for a part of the keys
 - It will also be a backup server for one of the primaries

WHAT COMPONENTS WILL BE BUILT/USED

- Zookeeper for storing configuration information
 - Address of the active servers
 - Key to server mapping
- Client
 - will take requests from the user
 - Types of requests supported
 - Put (char *key, char *value);
 - Get(char *key)
 - Client will contact zookeeper to determine address of the master server
 - Contact the master server to get key-server mapping
 - Determine whether the cluster is up.
 - Use key-server mapping to connect to the server that is responsible for the keys

SERVER INITIATIALIZATION

- Check if there is a master
 - Use zookeeper for this
- If no master
 - Declare self as master
 - Set cluster status to INITIALIZING
 - Wait x minutes for other servers to come up
 - Set cluster status to READY
 - Send start signal to all other servers
 - Send key-range and replicas to each server
 - Also send server id back. This is a sequence #
 - Send total #servers.
- If there is a master
 - Register with master
 - Wait until start signal
 - Store configuration data on which key-range server is responsible for.

SERVER – NORMAL OPERATION

- Clients will send requests to the server
 - Server will determine request type – put, get
 - Server will determine if it can process the request or the request has to be serviced by other servers
 - For self served requests – it will process the request and send back status of response
 - Remote server – respond with error message.(ERR_KEY_NOT_RESPONSIBLE)
- Server storage
 - Data will be stored in memory and not in any file. Management of memory is left to you.

SERVER - REPLICATION

- On coming up, will register with zookeeper it's address and the set of keys it is responsible for.
 - On writing request to the local storage, the data is also queued to be written to a remote server.
 - Remote server = $(\text{current_server id} + 1) \% \text{total \#servers}$.

HANDLING SERVER FAILURE

- Client tries connecting to server with key.
- On server failure, connects to master to get new list of keys-server mapping.
- Talks to the replica to retrieve data
- When server comes back up
 - The data which was written on the replica should be synced back

SCHEDULE

Date	Deliverable	Details
October 30	Basic client-server communication for the commands and implement get/put. Setup zookeeper for master server.	Set up a client and server and establish communication between these. Have the client send get/put/getmultiple messages and get a response. Single client server with get/put implemented. Provide server side script that can check contents of the storage. Server must store information in zookeeper and client must use it for <u>query</u>
November 12	Implement replication across servers and get/put across multiple servers	
November 27- Dec 1	Demonstrate high availability	On a server going down, client must still be able to retrieve information from a replica.

FINAL DEMO

- Must be shown on a cluster of at least 3 nodes.
 - Assume that data is split fixed across 3 nodes.

IMPORTANT DATES

Date	Deliverable
October 26, 29-30	CCBD Project (Mid Review) (10 marks)
October 30	Class Project Step 1 (5 marks)
November 12	Class Project Step 2 (5 marks)
November 27-Dec 1	Final Project Evaluation (20 marks), Step 3 of class project (20 marks)