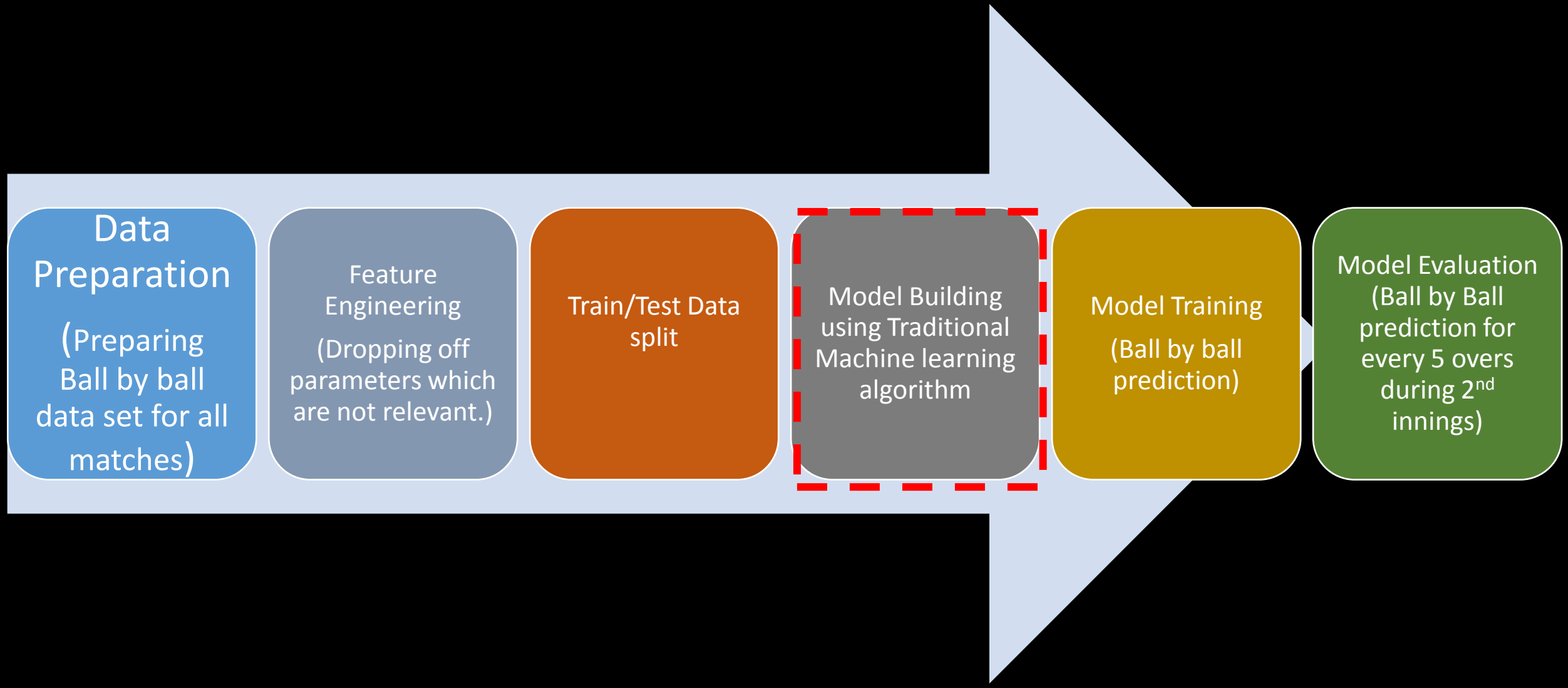
AIML-DEL-MAY-2019-GRP4

IPL Cricket Match Outcome Prediction Using AI Techniques

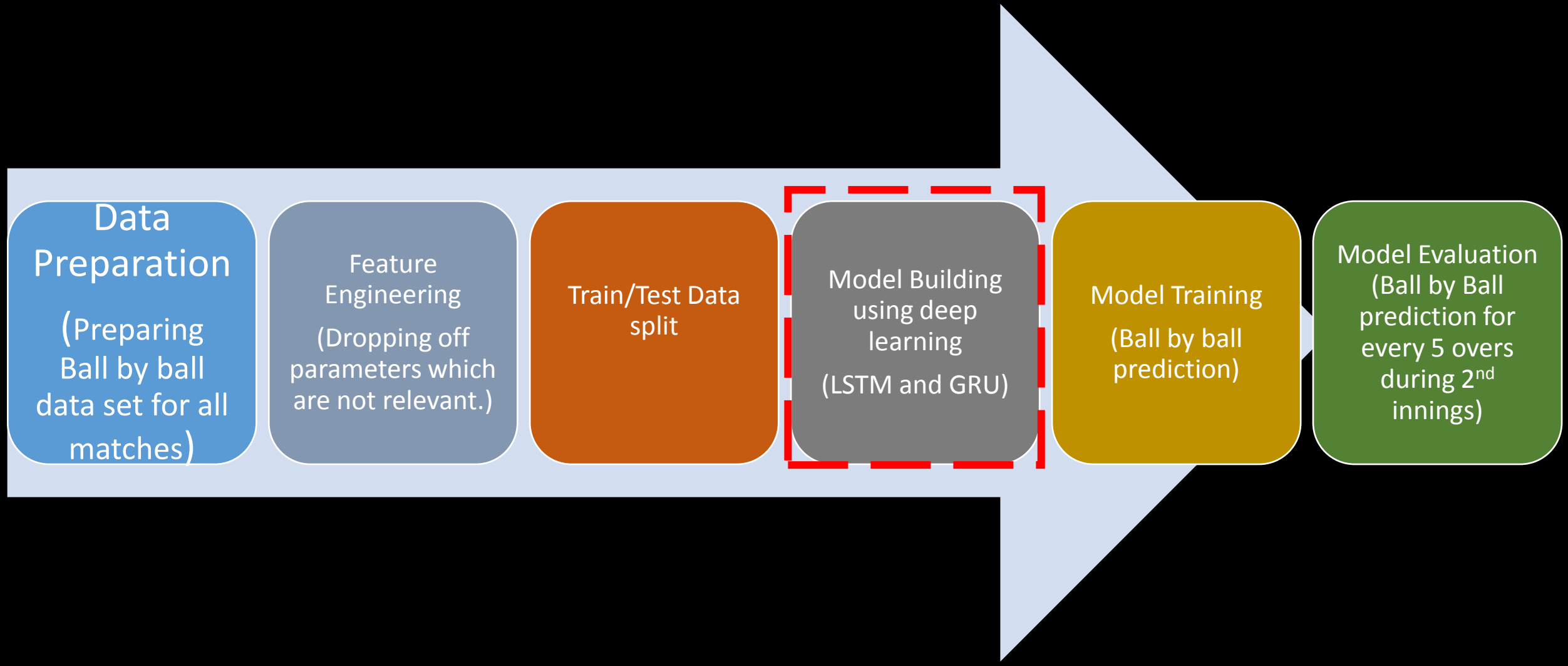
Problem Definition

- Predicting outcome of the IPL Cricket Match by using BALL by BALL prediction during the 2nd inning.

Model processing pipeline using Traditional Machine Learning Algorithm



Model processing pipeline using Deep Learning



Literature Survey

- In most research papers we have found accuracy around 50% whereas we have been able to achieve accuracy close to 78%. Major difference in our approach is that we have used ball by ball prediction during the 2nd innings of each match whereas others have used 1 prediction for the entire match.

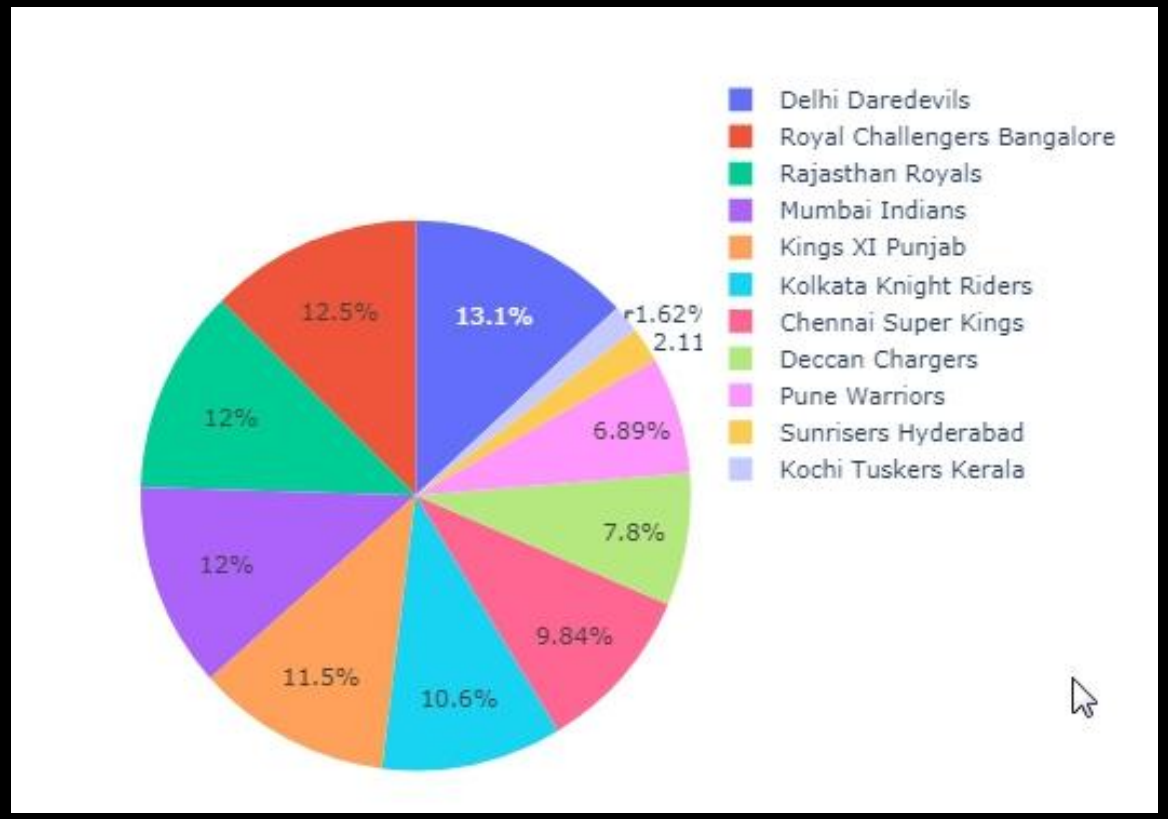
- In another research we analyzed there are factors like winning of a toss , weather conditions, batting first, fielding first and home ground plays an important role in winning.

- In another research we studied about the off the field scenarios of the match, players and the team on the bases of ticket sales patterns and their merchandise changes and prediction of the same by using machine learning algorithm.

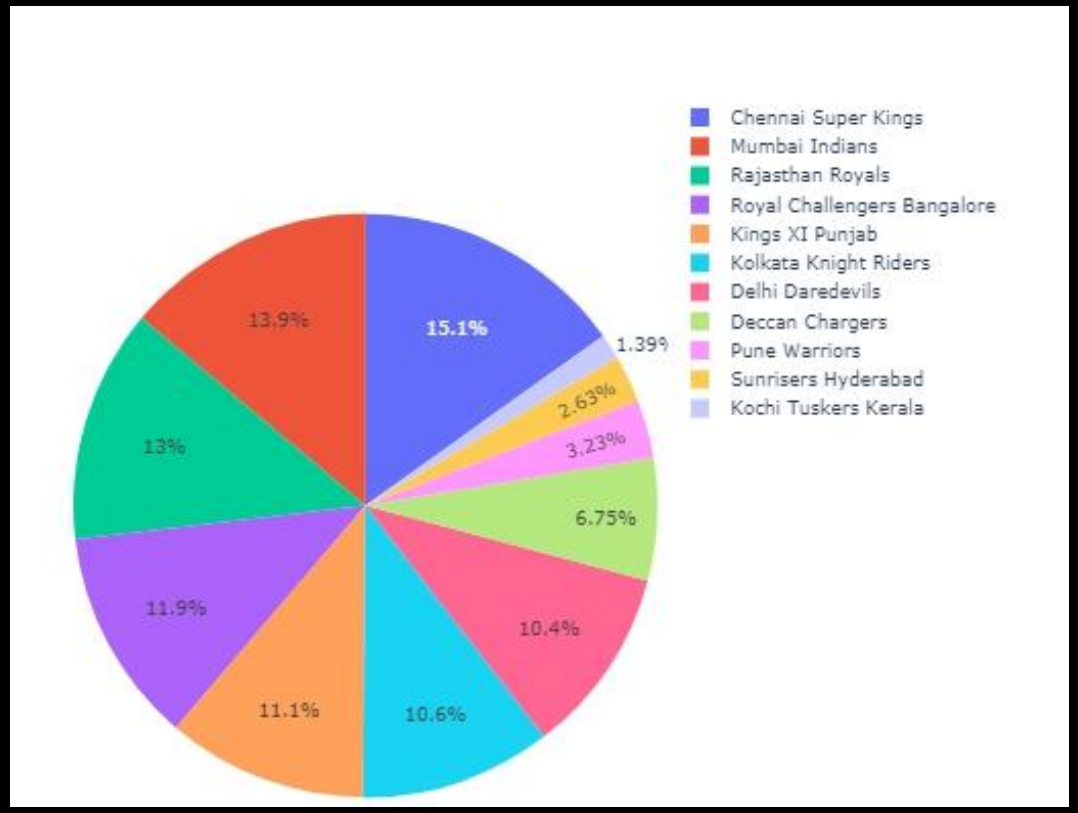
Data Description and EDA

- Data Collection
  - Data set from different sources were examined for this study.
  - Data set for all IPL matches(Season 2008-2013) were considered for this project work.
  - Total 378 matches were considered for this study.
- Pre-processing
  - Data cleaning (filtered out highly correlated attributes).
  - Feature engineering of the data ball by ball for each match manually.
    - Automated this process by making macros to speed up this process (~70.0% time efficient)

Data Visualization

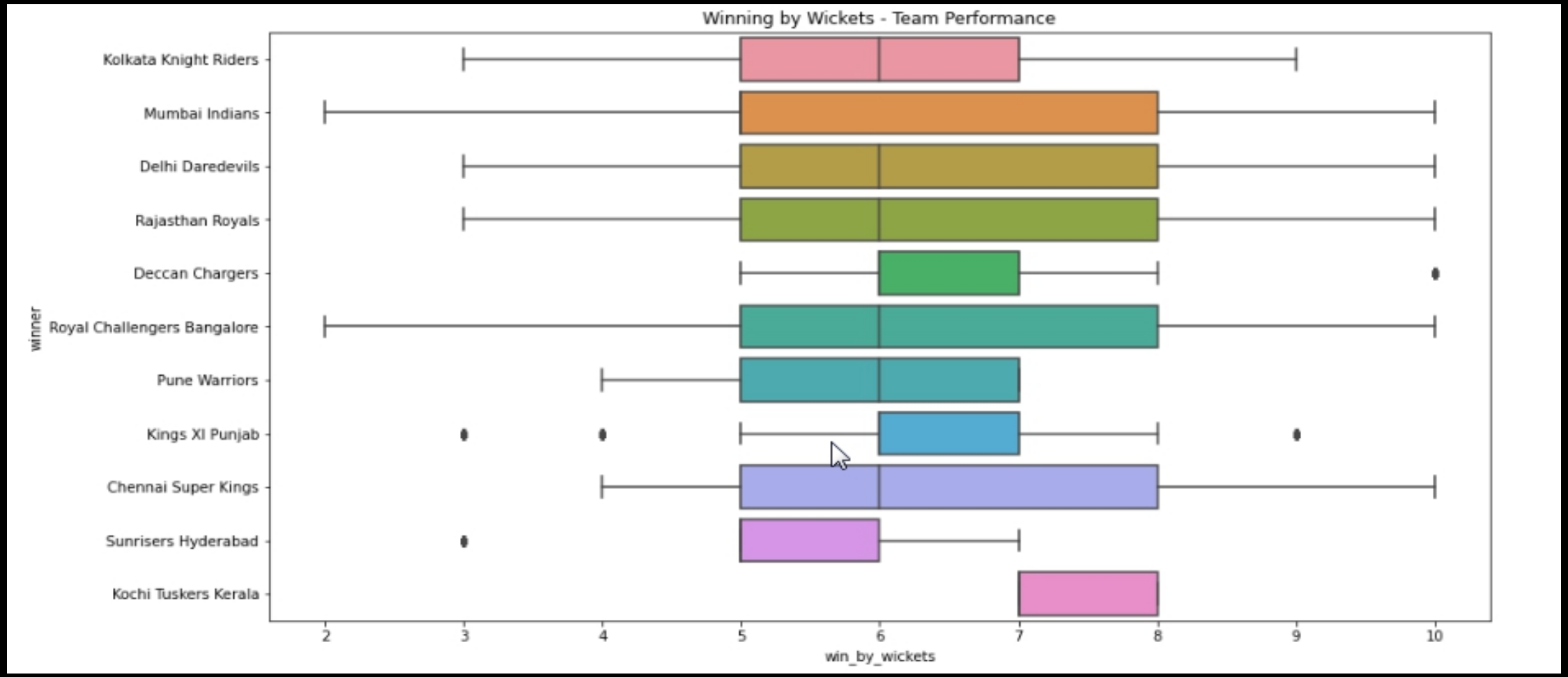


Playing Team wise distribution



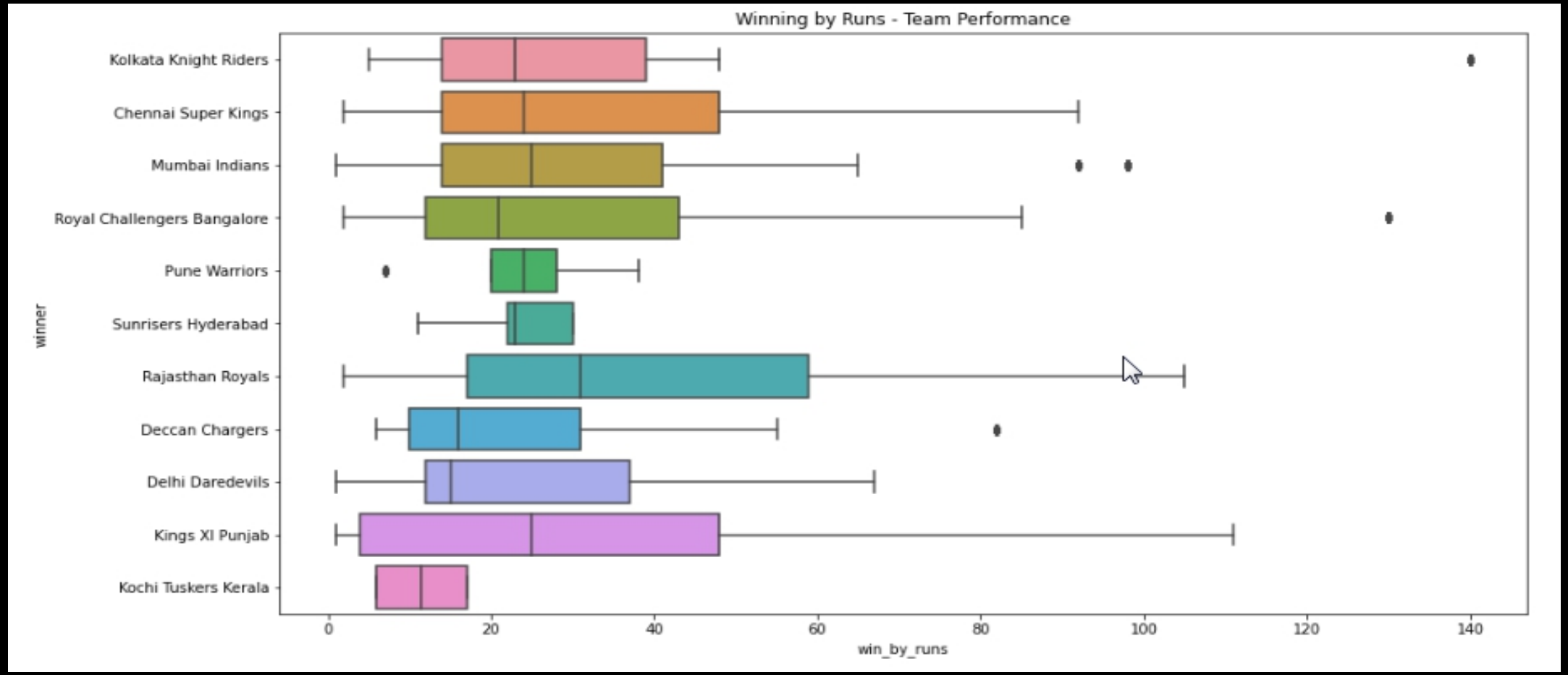
Winning Team wise distribution

Data Visualization



Winning Team – Wicket wise

Data Visualization

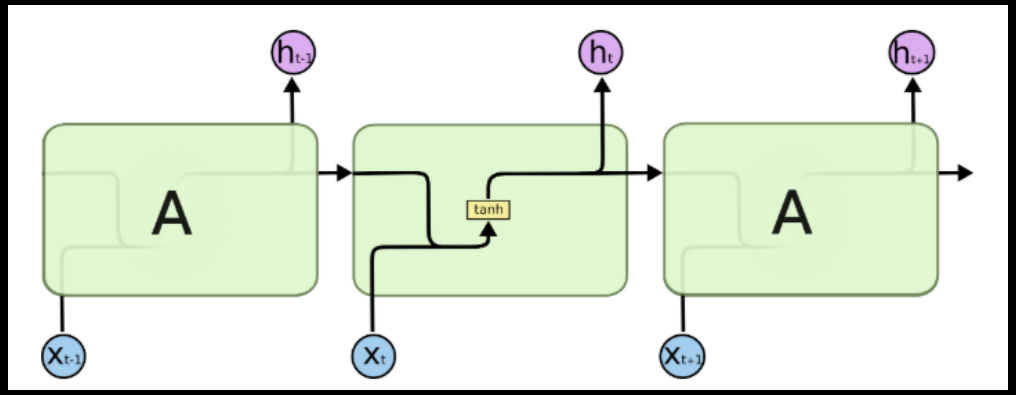


Winning Team – Run wise

Architectures of models – Vanilla LSTM

Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies.

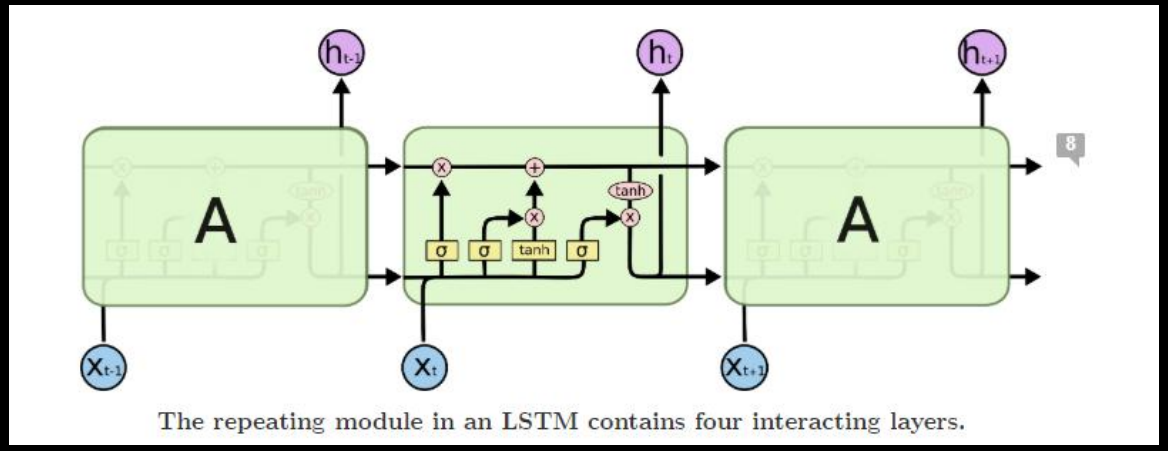
- They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in following work.
- They work tremendously well on a large variety of problems, and are now widely used.
- LSTMs are explicitly designed to avoid the long-term dependency problem.
- Remembering information for long periods of time is practically their default behavior, not something they struggle to learn.
- In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.



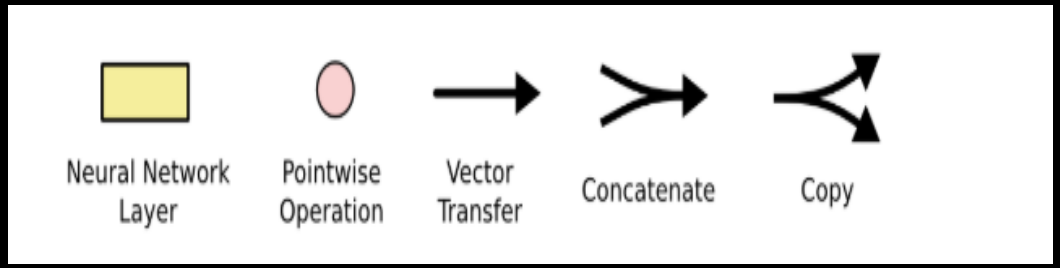
- LSTMs also have this chain like structure, but the repeating module has a different structure.
- Instead of having a single neural network layer, there are four, interacting in a very special way.

Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Architectures of models – Vanilla LSTM



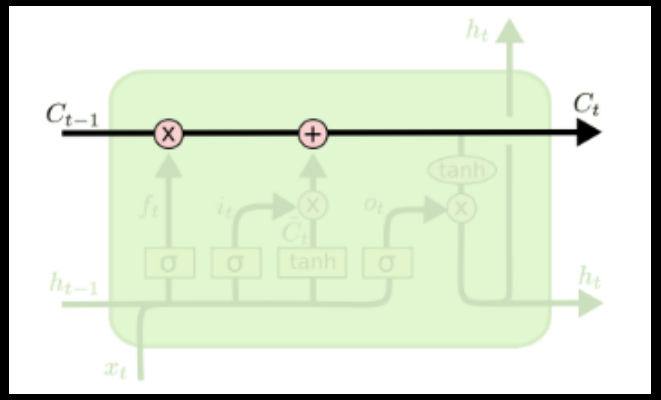
The repeating module in an LSTM contains four interacting layers.



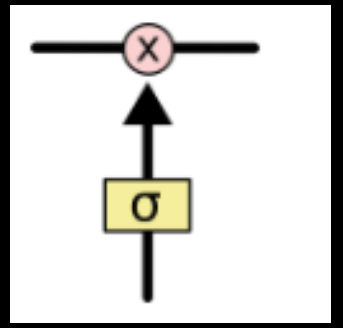
- In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others.
- The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers.
- Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

Core Idea Behind LSTMs

- The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.
- The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged.



- The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.
- Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.



- The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through.
- A value of zero means "let nothing through," while a value of one means "let everything through!"
- An LSTM has three of these gates, to protect and control the cell state.

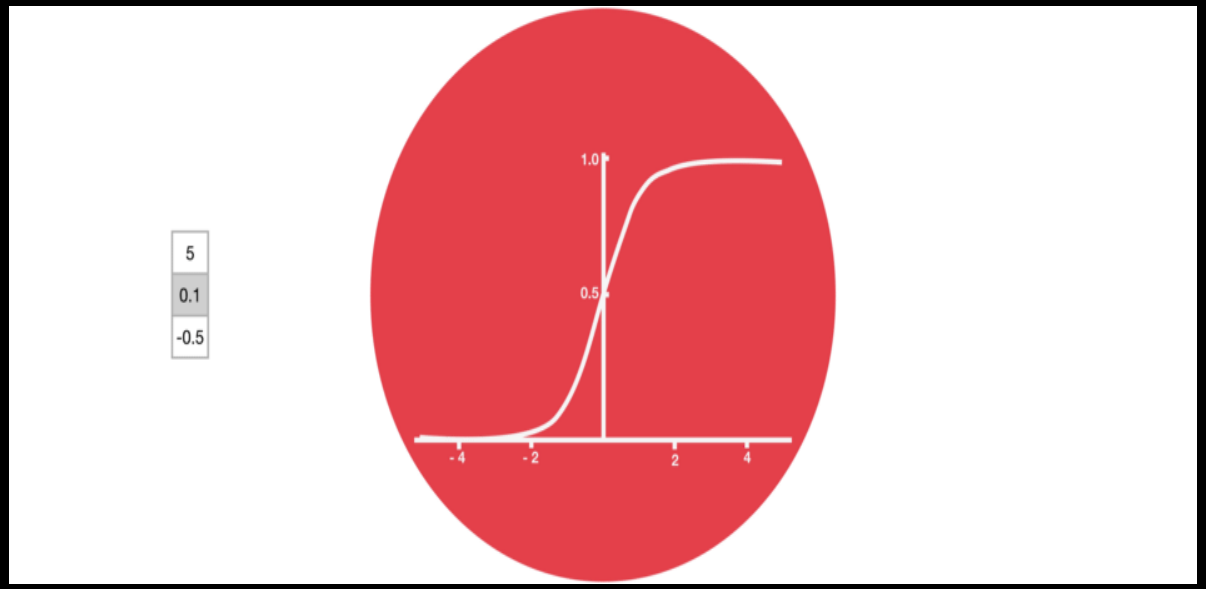
Core Concept of LSTM

- The core concept of LSTM’s are the cell state, and it’s various gates.
- The cell state act as a transport highway that transfers relative information all the way down the sequence chain.
- As the cell state goes on its journey, information get’s added or removed to the cell state via gates.
- The gates are different neural networks that decide which information is allowed on the cell state.
- The gates can learn what information is relevant to keep or forget during training.

Core Concept of LSTM

Sigmoid

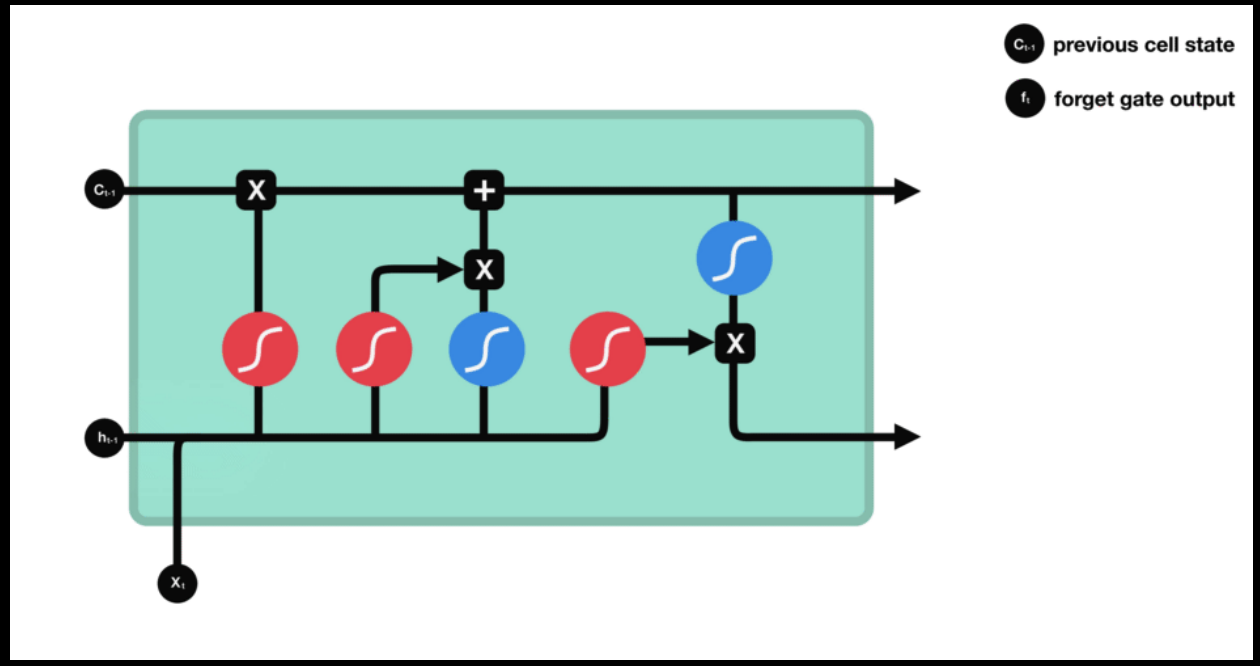
- Gates contains sigmoid activations.
- A sigmoid activation is similar to the tanh activation.
- Instead of squishing values between -1 and 1, it squishes values between 0 and 1.
- That is helpful to update or forget data because any number getting multiplied by 0 is 0, causing values to disappears or be “forgotten.”



Core Concept of LSTM

Forget Gate

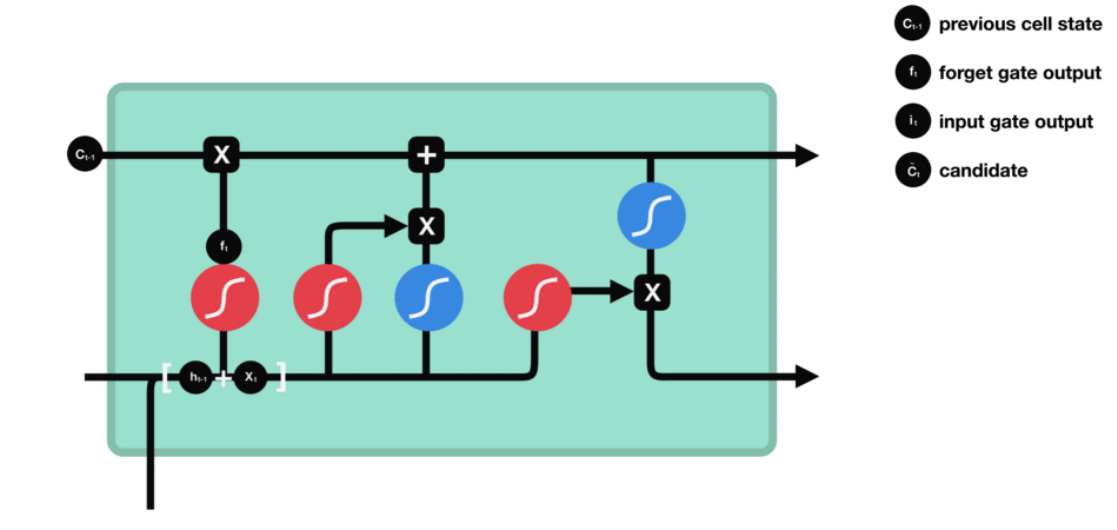
- First, we have the forget gate. This gate decides what information should be thrown away or kept.
- Information from the previous hidden state and information from the current input is passed through the sigmoid function.
- Values come out between 0 and 1. The closer to 0 means to forget, and the closer to 1 means to keep.



Core Concept of LSTM

Input Gate

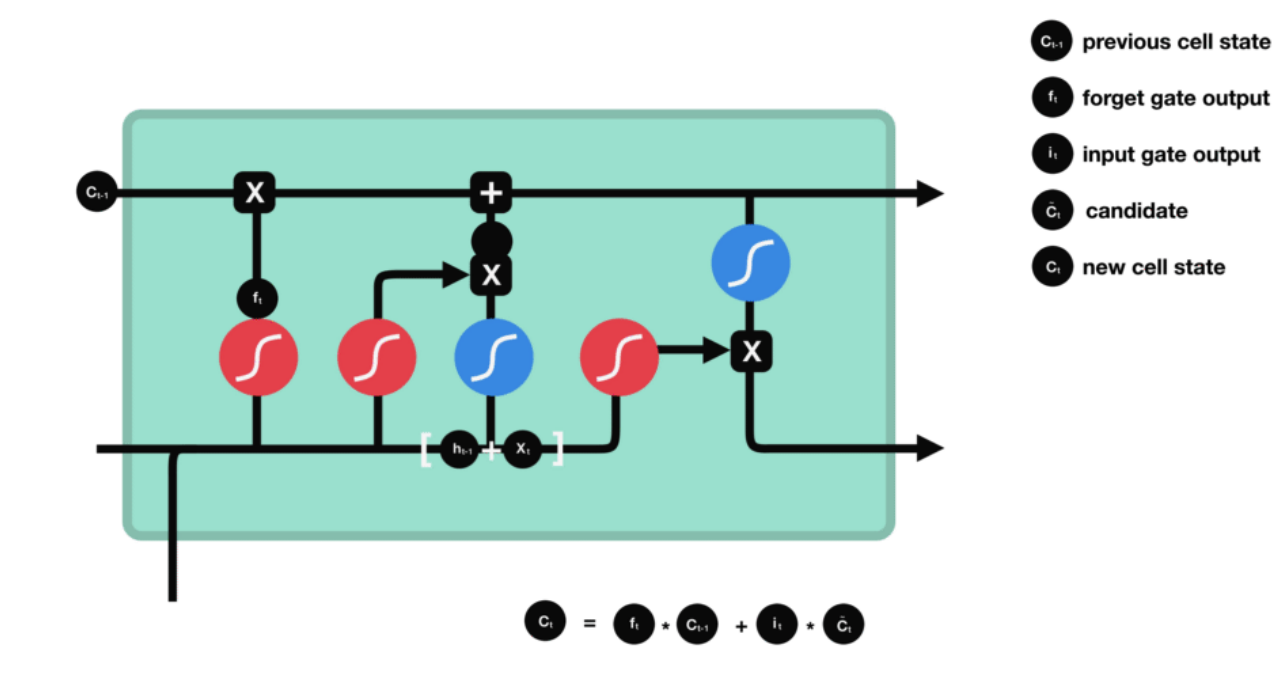
- To update the cell state, we have the input gate.
- First, we pass the previous hidden state and current input into a sigmoid function.
- That decides which values will be updated by transforming the values to be between 0 and 1. 0 means not important, and 1 means important.
- You also pass the hidden state and current input into the tanh function to squish values between -1 and 1 to help regulate the network.
- The gates can learn what information is relevant to keep or forget during training.
- Then you multiply the tanh output with the sigmoid output.
- The sigmoid output will decide which information is important to keep from the tanh output.



Core Concept of LSTM

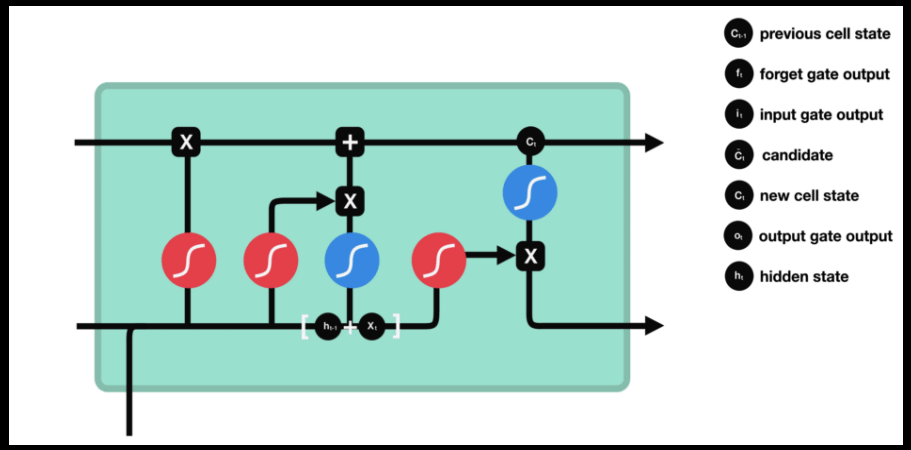
Cell State

- Now we should have enough information to calculate the cell state.
- First, the cell state gets pointwise multiplied by the forget vector.
- This has a possibility of dropping values in the cell state if it gets multiplied by values near 0.
- Then we take the output from the input gate and do a pointwise addition which updates the cell state to new values that the neural network finds relevant.
- That gives us our new cell state.



Core Concept of LSTM

Output Gate

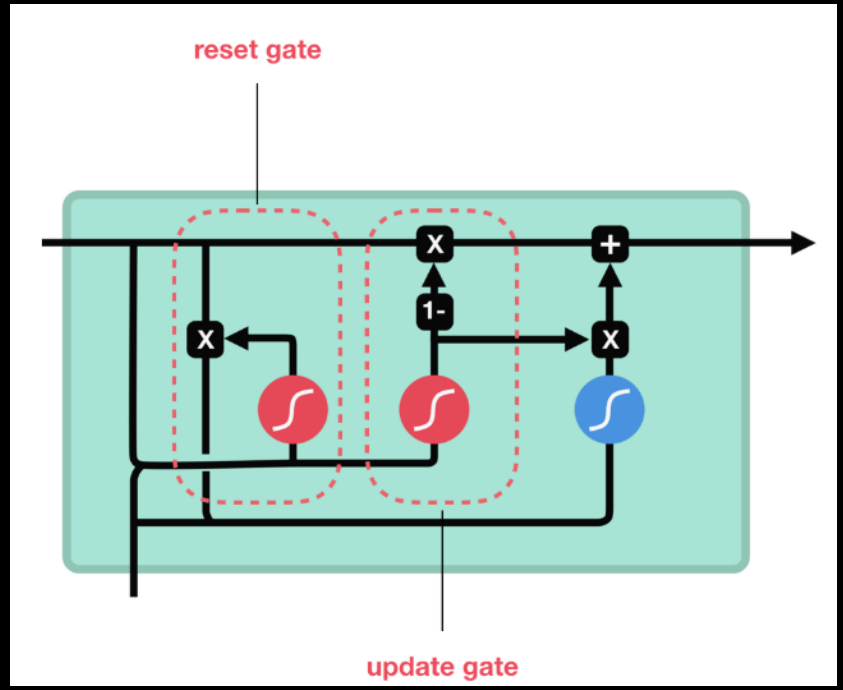


- Last we have the output gate.
- The output gate decides what the next hidden state should be.
- Remember that the hidden state contains information on previous inputs.
- The hidden state is also used for predictions.
- First, we pass the previous hidden state and the current input into a sigmoid function.
- Then we pass the newly modified cell state to the tanh function.
- We multiply the tanh output with the sigmoid output to decide what information the hidden state should carry.
- The output is the hidden state.
- The new cell state and the new hidden is then carried over to the next time step.

Core Concept of GRU

GRU

- The GRU is the newer generation of Recurrent Neural networks and is pretty similar to an LSTM.
- GRU's got rid of the cell state and used the hidden state to transfer information.
- It also only has two gates, a reset gate and update gate.
- The hidden state is also used for predictions.
- It also only has two gates, a reset gate and update gate.



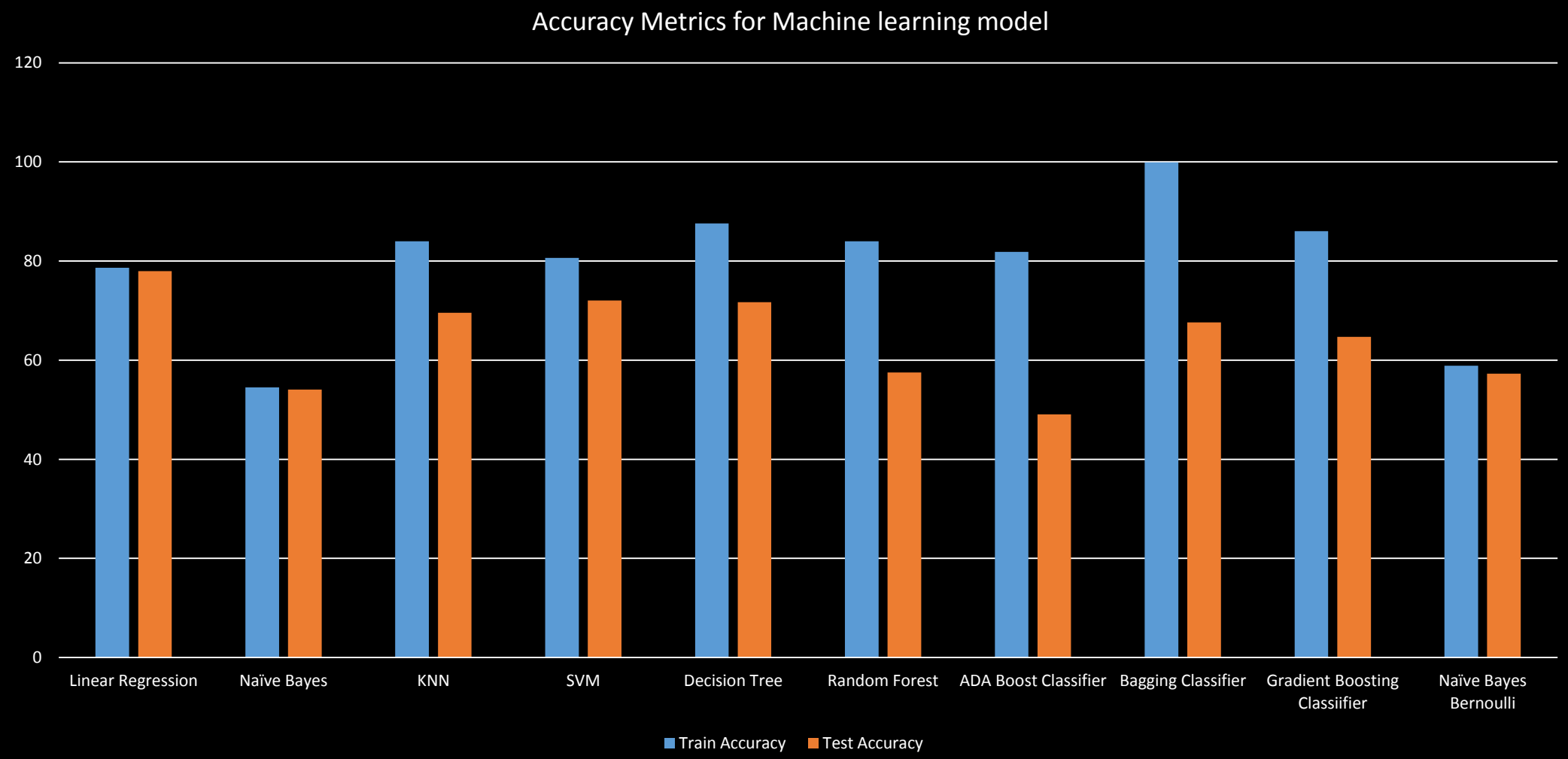
Update Gate

- The update gate acts similar to the forget and input gate of an LSTM.
- It decides what information to throw away and what new information to add.

Reset Gate

- The reset gate is another gate is used to decide how much past information to forget.
- And that's a GRU.
- GRU's has fewer tensor operations; therefore, they are a little speedier to train then LSTM's.
- There isn't a clear winner which one is better.
- Researchers and engineers usually try both to determine which one works better for their use case.

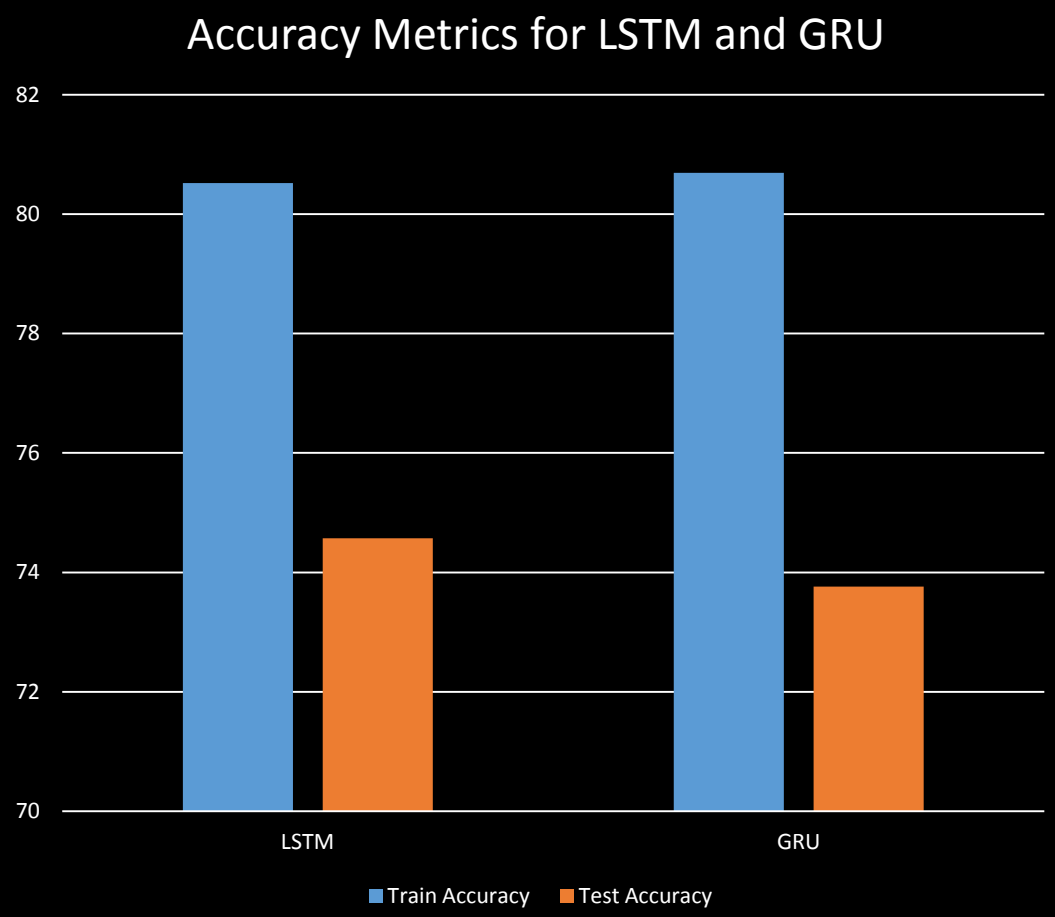
Result Analysis and Comparisons
Traditional Machine Learning Models



Results Analysis and Comparisons Traditional Machine Learning Models

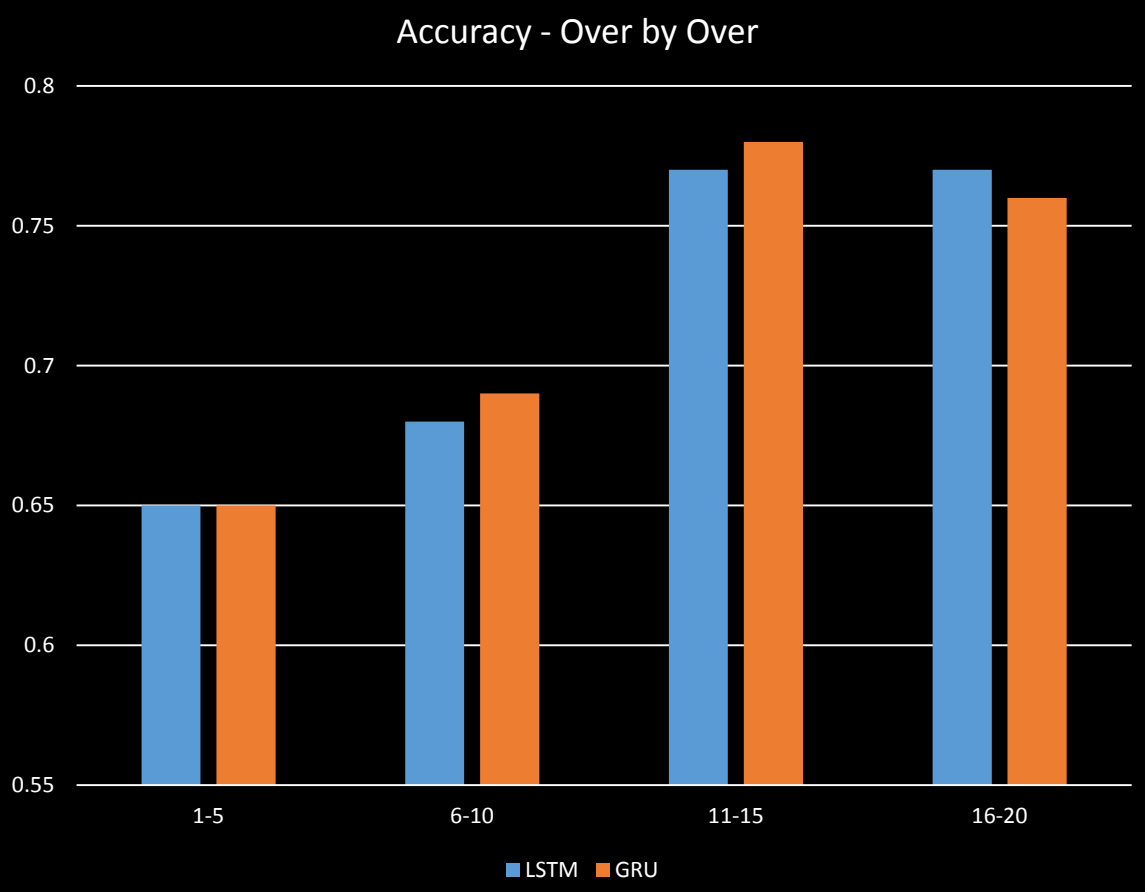
MODEL NAME	TRAIN ACCURACY	TEST ACCURACY
LINEAR REGRESSION	78.64	77.98
NAÏVE BAYES	54.52	54.08
KNN	83.98	69.57
SVM	80.62	72.06
DECISION TREE	87.61	71.7
RANDOM FOREST	83.97	57.52
ADA BOOST CLASSIFIER	81.84	49.05
BAGGING CLASSIFIER	99.95	67.64
GRADIENT BOOSTING CLASSIFIER	86.01	64.72
NAÏVE BAYES BERNOULLI	58.88	57.27

Results Analysis and Comparisons LSTM and GRU



MODEL NAME	TRAIN ACCURACY	TEST ACCURACY
LSTM	80.52	74.57
GRU	80.69	73.76

Results Analysis and Comparisons LSTM and GRU, Over by Over



OVER	LSTM	GRU
1-5	0.65	0.65
6-10	0.68	0.69
11-15	0.77	0.78
16-20	0.77	0.76

Conclusions

- First of kind model which is giving ball by ball prediction.
- First time we have used LSTM because we used sequential data.

Future directions

- The model can be further enhanced by putting more attributes for e.g., venues, off the field attributes, including player stats etc.
- Real time prediction of live matches can be integrated with the model.
- In future we also intend to build a recommendation model which can suggest during the second innings as to which player is best suited to play in a particular situation during the match.

Reference

- Sandesh Bananki Jayantha,b,∗, A team recommendation system and outcome prediction for the game of cricket, pp. 263-264
- Kumash Kapadia, Sport analytics for cricket game results using machine learning:An experimental study, Applied Computing and Informatics, (2019)
- Rabindra Lamsal, Ayesha Choudhary, Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning, arXiv:1809.09813v4 [stat.AP] 25 Jun 2019
- Analytics, C., 2017. Anaconda software distribution. Computer software Vers, 2-2.
- D. Böhning, Multinomial logistic regression algorithm, Ann. Inst. Stat. Math. 44 (1) (1992) 197–200.
- L. Breiman, Random forests, Machine Learn. 45 (1) (2001) 5–32
- R.O. Duda, P.E. Hart, D.G. Stork, Bayesian decision theory, Pattern Classification 11 (4) (2001) 99–102.
- E. Frank, Y. Wang, S. Inglis, G. Holmes, I.H. Witten, using model trees for classification, Machine Learn.32 (1) (1998) 63–76.
- Gunn, S.R., 1998. Support vector machines for classification and regression. ISIS technical report, 14(1), 5–16
- Wikipedia on the Game of Cricket, website [Online] http://en.wikipedia.org/wiki/Cricket
- CricInfo, Website for cricket data, [online] http://www.cricinfo.com
- Great Learning materials.


Thank You
JERRYL DAVIS
AMIT BHATIA
RAJESH GOEL
PULKIT MALHOTRA
HARSH BHARDWAJ
VIKAS HOODA