# Real Time Rank Prediction using Time Series on IndyCar data

By: Uma Maheswari Gollapudi and Sai Sugeeth Kamineni

B649/E599 High Performance Big Data Systems

Fall 2020

# Introduction

Real Time Rank Prediction of IndyCar Data

Why?

- Allows the crew to develop strategies on the spot for upcoming laps
- Monitor their cars more closely in case of decreased performance
- Modify pitstop timings
- Benefits people who indulge in legal race betting!

# The Data

**Size:**

4 Datasets-Cars, Laps, Pitstop, Telemetry

5,979,749 records

At this point, we are using the ‘completed laps’ records only

**Time step size:** 10

**Target:** overall_rank

**#Features:** 16

**Features:** 'overall_rank','last_laptime', 'track_status', 'pit_stop_count', 'completed_laps', 'elapsed_time','best_laptime', 'time_behind_leader','time_behind_prec', 'overall_best_laptime', 'start_position', 'laps_led', etc.

**More Data:**

Pitstop intervals

Other years

# Related Work

- Previous work is not similar to this problem
  - Not real-time
  - Do not take anomalies, pitstop timings into consideration
  - Initial findings

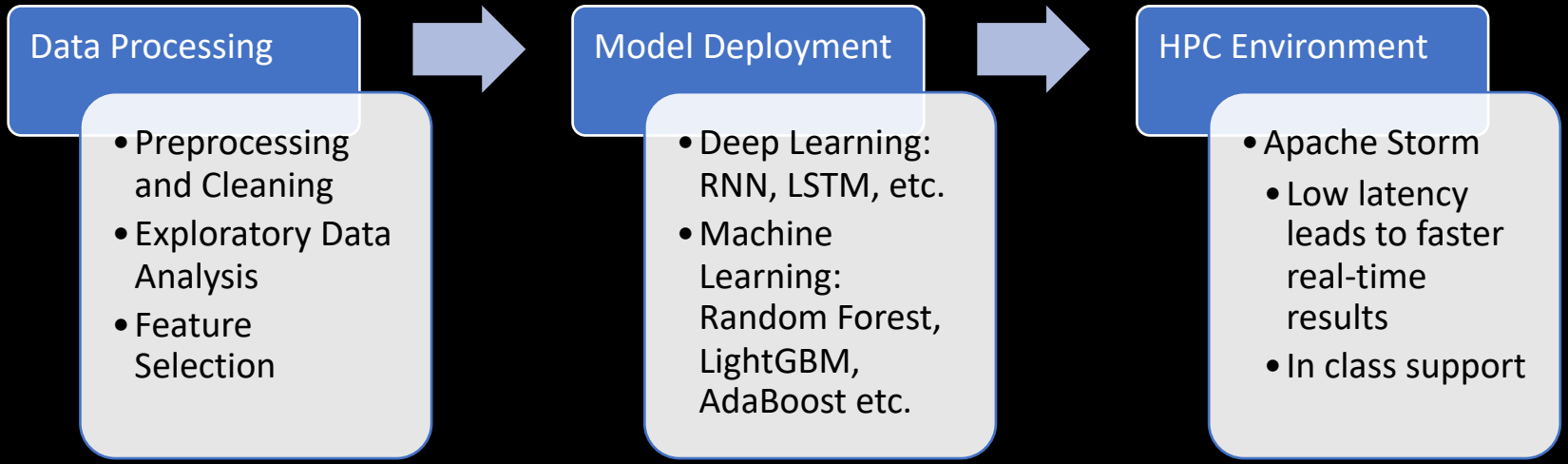
# Related Work

- Initial papers on NASCAR racing predictions:
  - *Predicting the Outcome of NASCAR Races: The Role of Driver Experience*
  - *Do Reliable Predictors Exist for the Outcomes of NASCAR Races?*
- Papers on horse racing predictions via ML:
  - *Predicting Horse Racing Result with Machine Learning*
  - *Horse Racing Predictions using Artificial Neural Networks.*
- Takeaways: Experience of driver, variables related to car speed, prior success of driver -> car ranking!

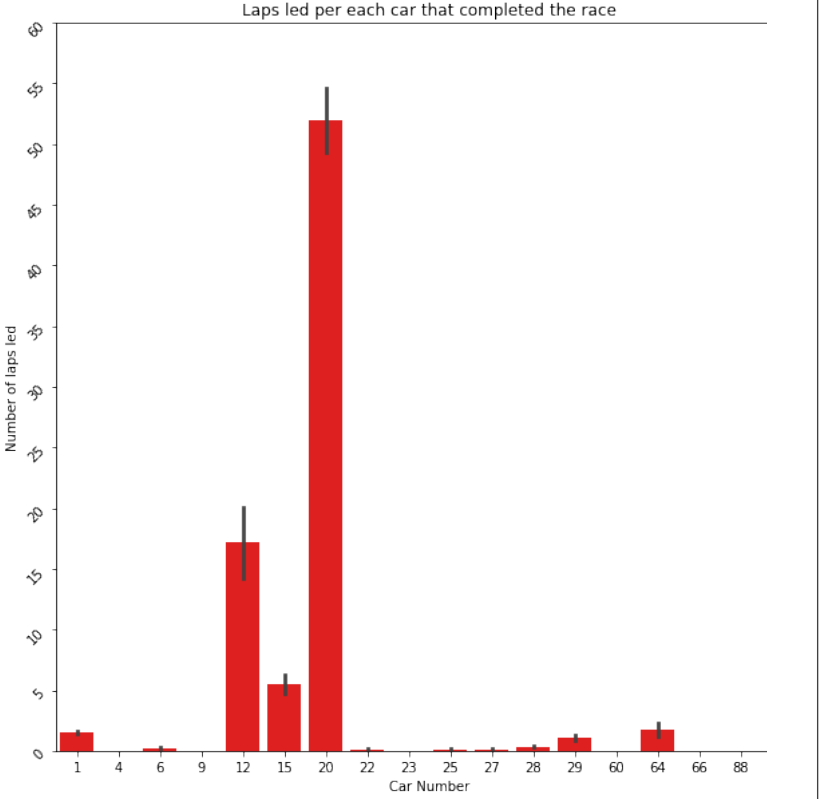
# Related Work

- Paper titled “Rank Position Forecasting in Car Racing” – IU and NEC Japan (23 Oct 2020)
- Proposes use of “RankNet” for rank forecasting
- a combination of encoder- decoder network and separate MLP network that is capable of delivering probabilistic forecasting to model the pit stop events and rank position in car racing

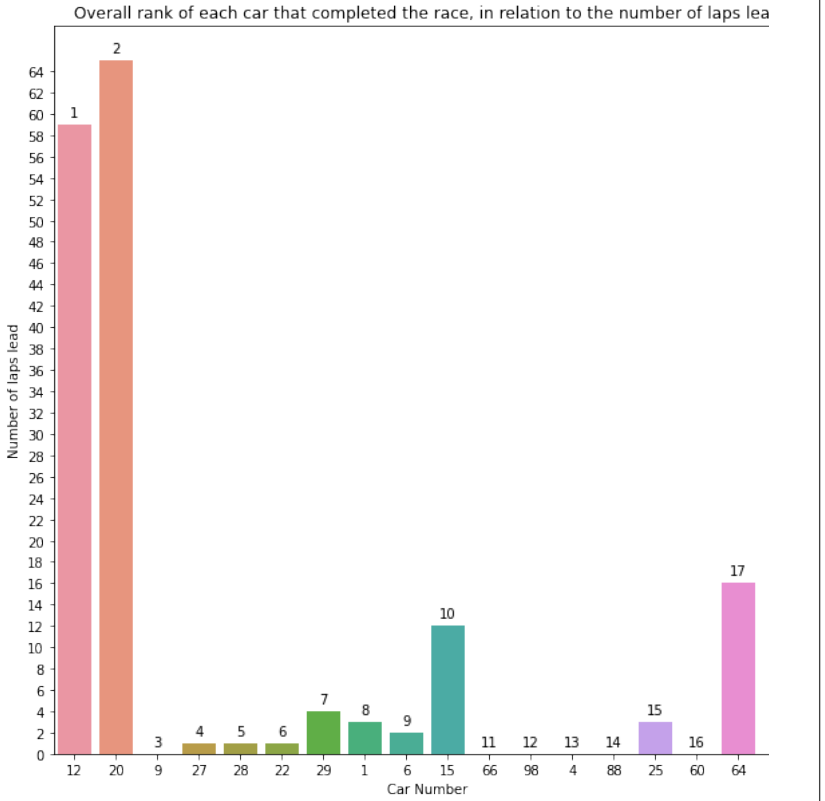
# Proposed Methodology



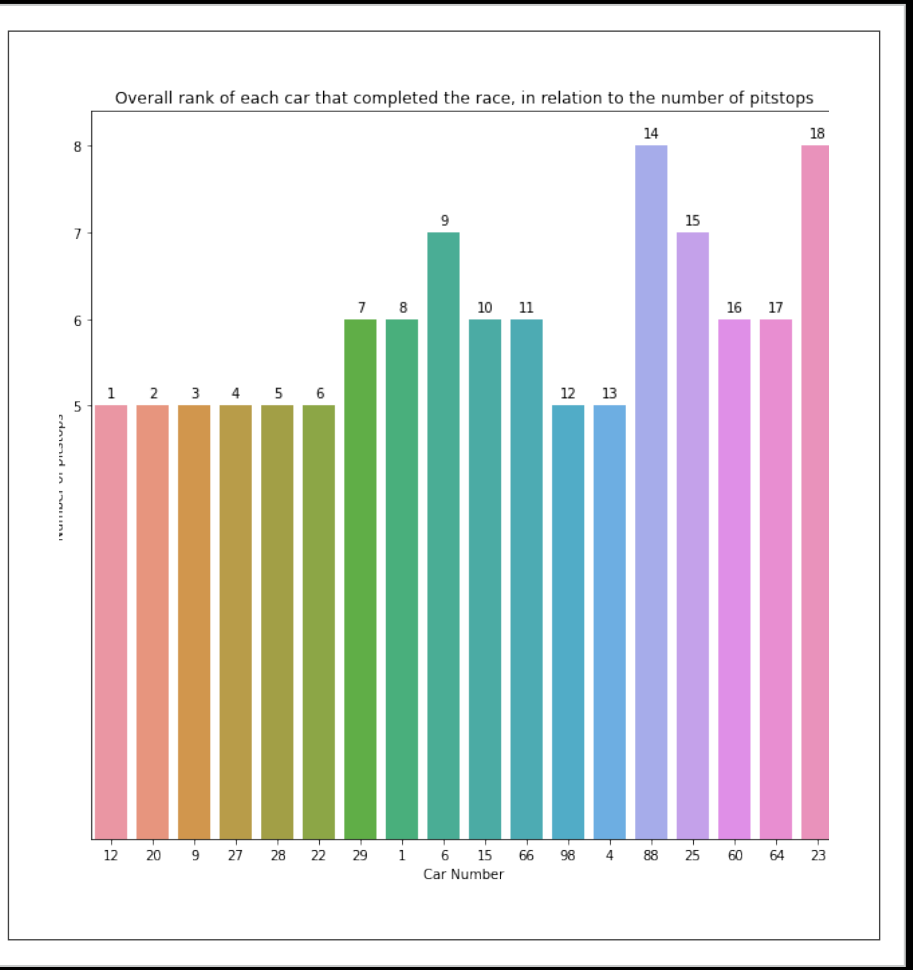
# Exploratory Data Analysis




# Laps Led per Car Number

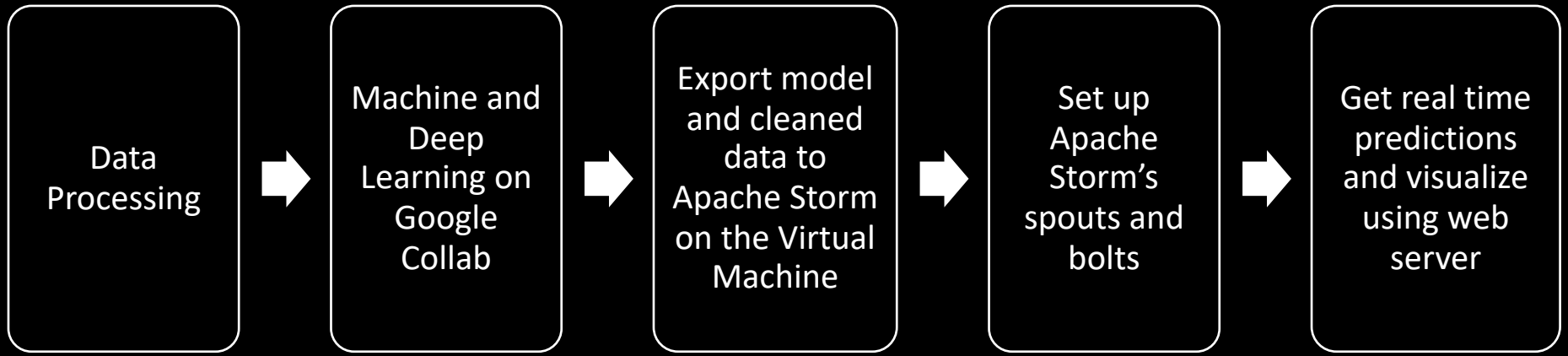



# Laps Led vs Ranking of Car

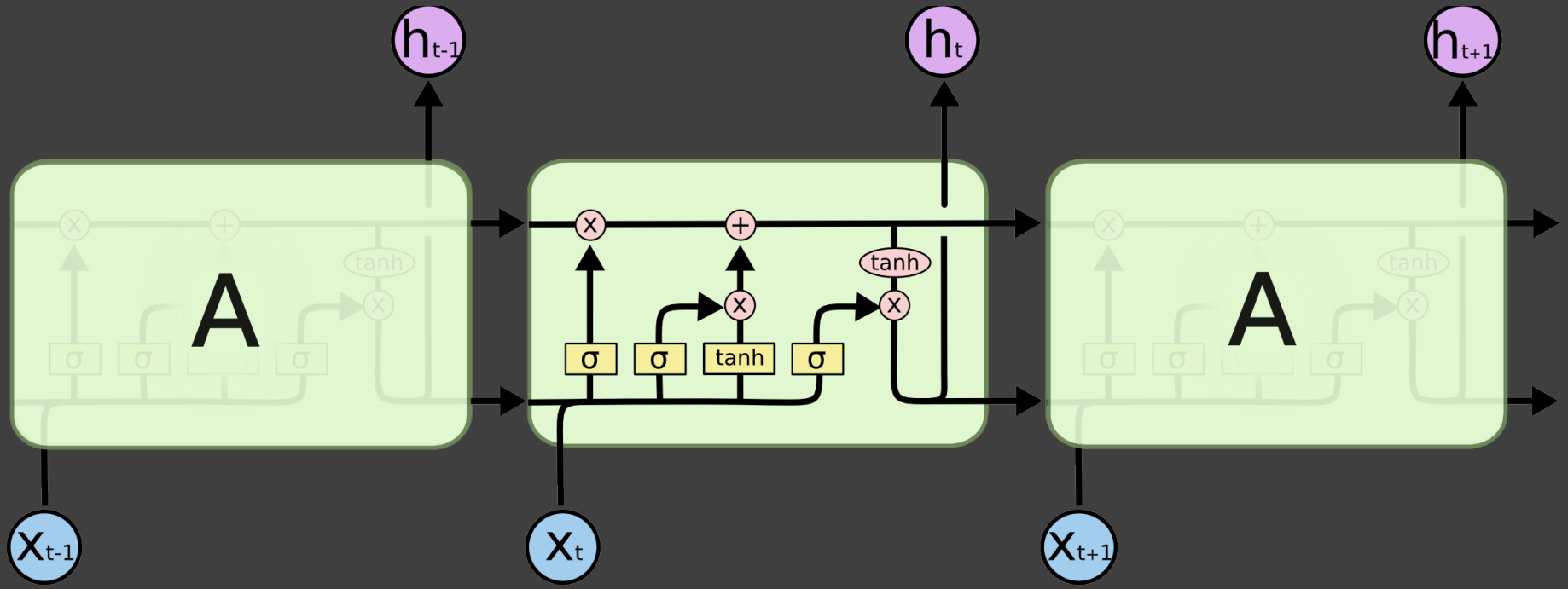



# Number of Pitstops vs Ranking of Car

# The Real-Time Prediction Framework



# Algorithm, Implementation and Results




# LSTM – The Baseline

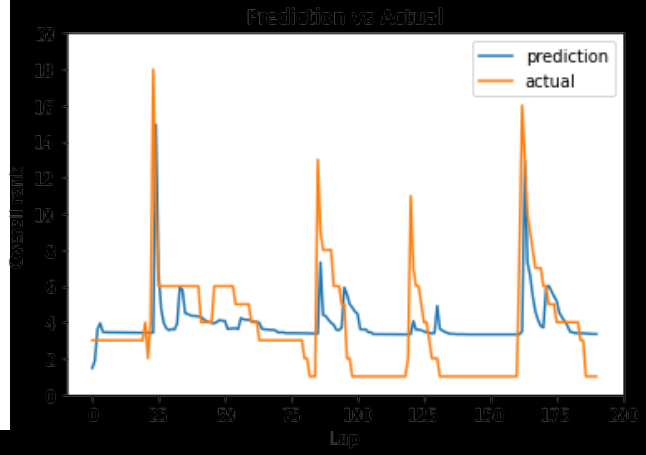
- Inputs are related
- Ability to retain memory
- Perfect match for time series data
- Retains information for a longer period in comparison to RNNs

# Initial Results with LSTM

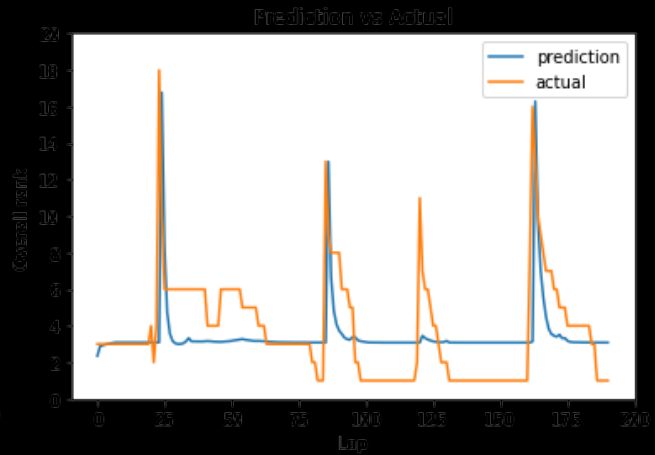
## LSTM

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 10, 32)            5504
_________________________________________________________________
batch_normalization (BatchNo (None, 10, 32)            128
_________________________________________________________________
dropout (Dropout)            (None, 10, 32)            0
_________________________________________________________________
lstm_1 (LSTM)                (None, 10, 32)            8320
_________________________________________________________________
batch_normalization_1 (Batch (None, 10, 32)            128
_________________________________________________________________
dropout_1 (Dropout)          (None, 10, 32)            0
_________________________________________________________________
lstm_2 (LSTM)                (None, 16)                3136
_________________________________________________________________
dense (Dense)                (None, 1)                 17
=================================================================
Total params: 17,233
Trainable params: 17,105
Non-trainable params: 128
_________________________________________________________________
```

## MSE = 9.1




## MAE= 2.4



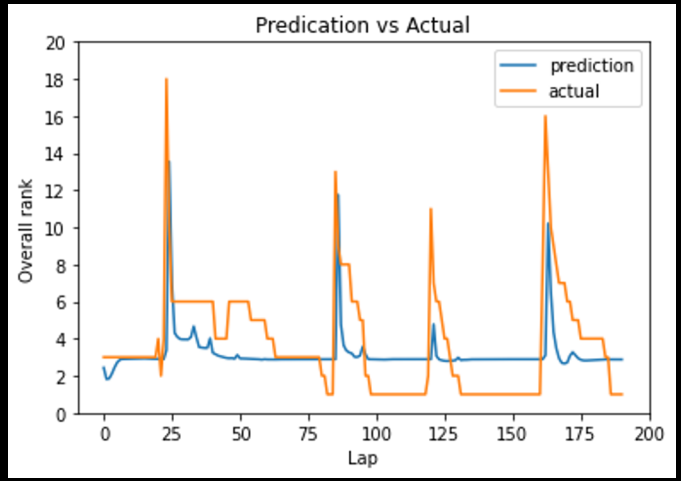

**Promising : LSTM results have similar trends in predicted vs actual values!**

# Updated Results with LSTM

## LSTM architecture

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 10, 64)            20736
_________________________________________________________________
batch_normalization (BatchNo (None, 10, 64)            256
_________________________________________________________________
dropout (Dropout)            (None, 10, 64)            0
_________________________________________________________________
lstm_1 (LSTM)                (None, 10, 64)            33024
_________________________________________________________________
batch_normalization_1 (Batch (None, 10, 64)            256
_________________________________________________________________
dropout_1 (Dropout)          (None, 10, 64)            0
_________________________________________________________________
lstm_2 (LSTM)                (None, 16)                5184
_________________________________________________________________
dense (Dense)                (None, 1)                 17
=================================================================
Total params: 59,473
Trainable params: 59,217
Non-trainable params: 256
_________________________________________________________________
```

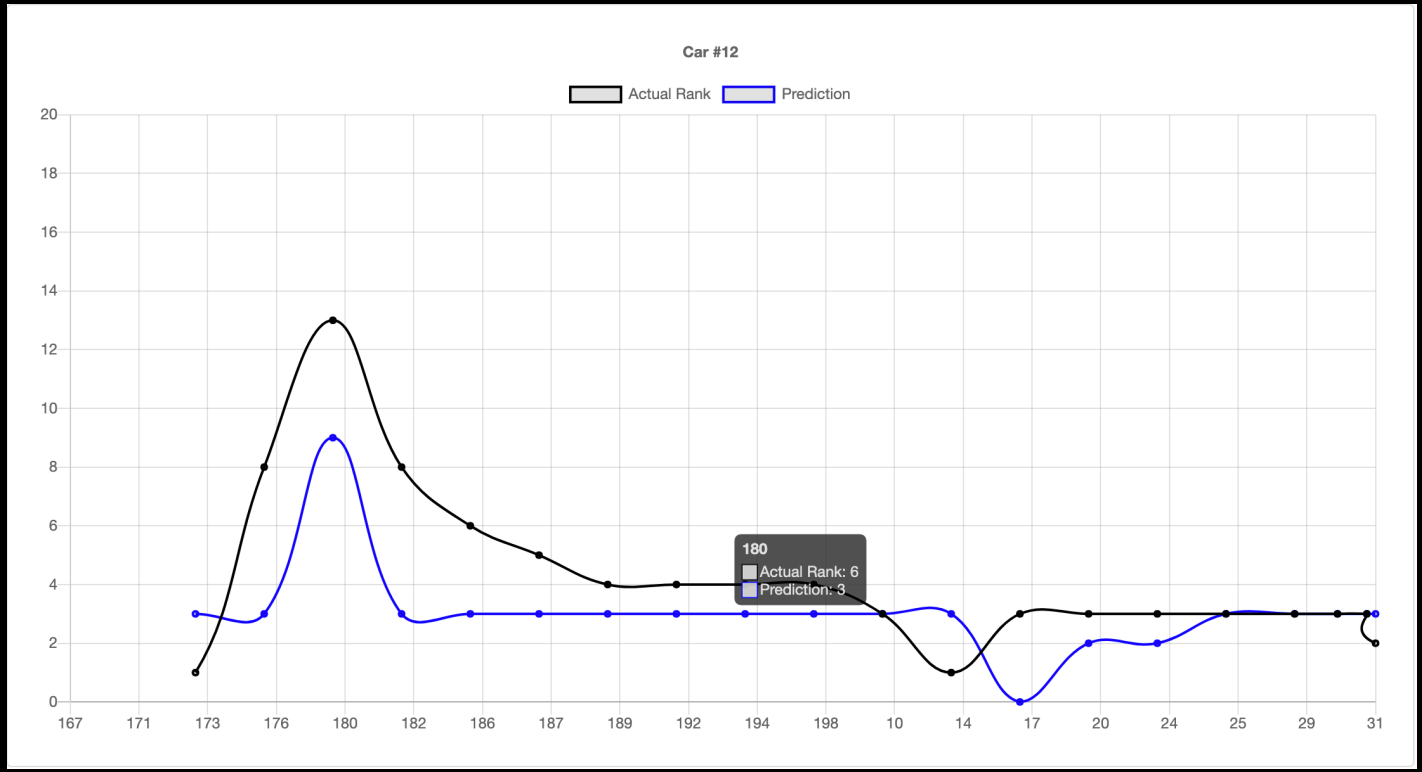
## MAE = 2.1



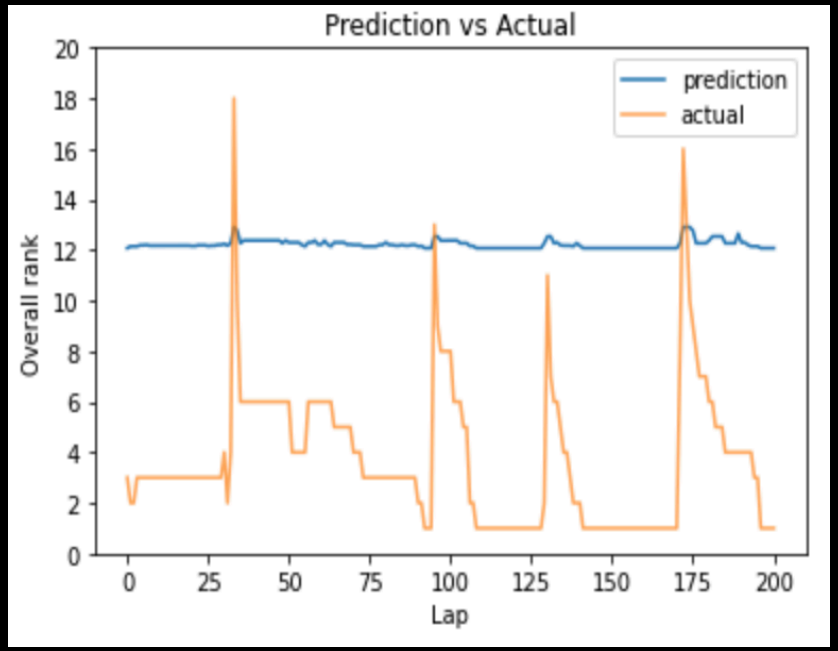

**Better performance after hyperparameter tuning!**

# Real Time Predictions in Apache Storm

For one car:

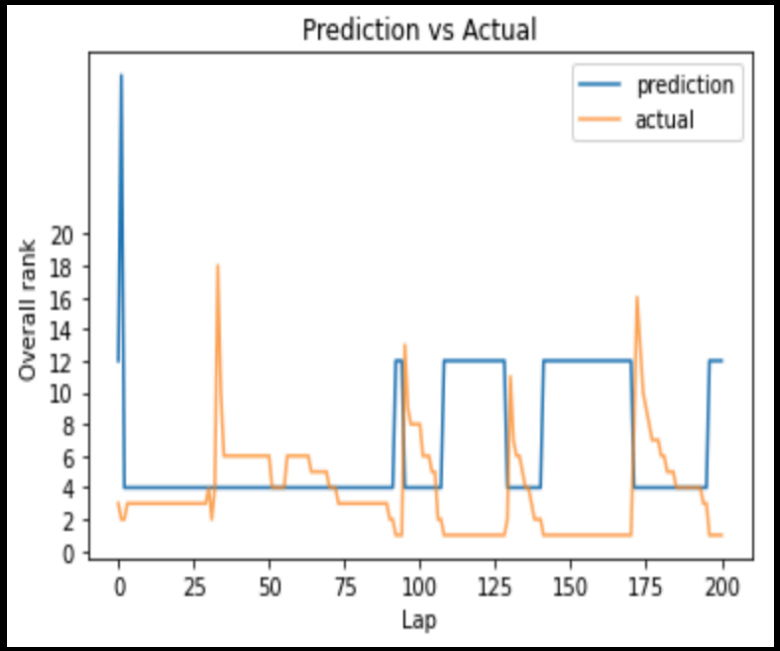


# Results with Machine Learning Models




LightGBM Regressor

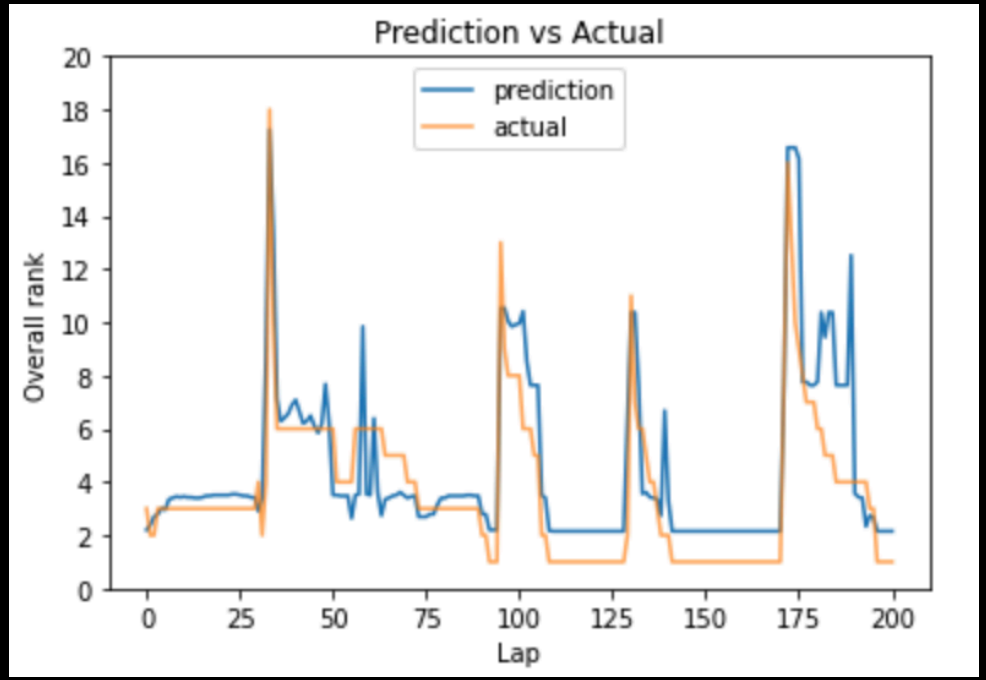
MAE = 8.63




AdaBoost Classifier

MAE = 4.66

# Note: Random Forest performs exceedingly well…




Random Forest Regressor

MAE = 1.31

The Random Forest Regressor performs better than the LSTM model, with the current parameter tuning

Oversimplification of data and overfitting, incorrect metric or genuinely good?

# Conclusion

LSTMs perform well on time series data – upto 2x better than most ML models

ML models are promising too – to an extent. More research and parameter optimization would be helpful!

Storm provides an accessible and easy to use framework.

Unsimilar previous works in this - harder to find a baseline and compare.

# Future Work

Understand performance of Random Forest

Improve LSTM model architecture

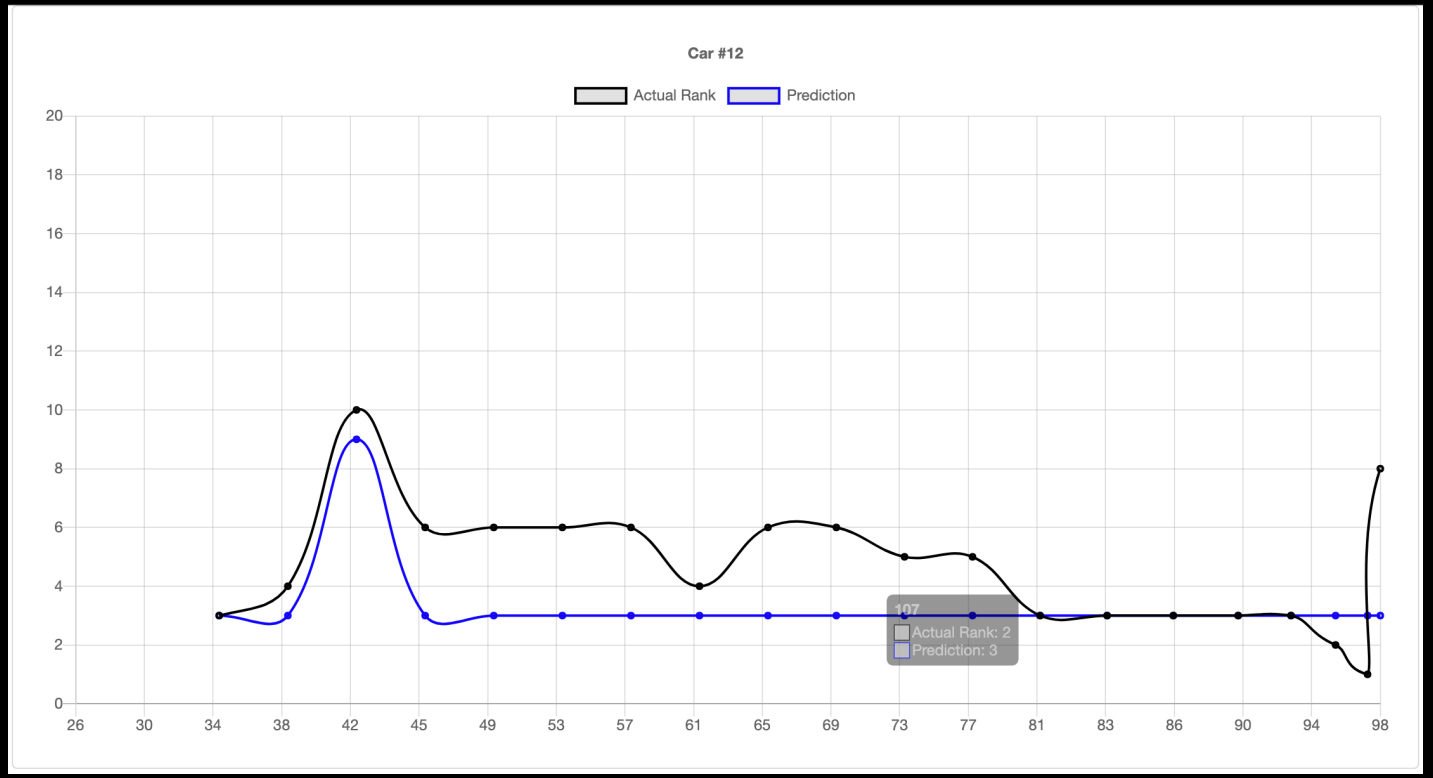
Take pitstop interval prediction into direct consideration for rank forecasting

# THANK YOU!

…..QUESTIONS,
ANYONE?

# Appendix

Single Car real time forecasting – Overlapping predictions



# Real Time Predictions in Apache Storm

For two cars:

Unable to retrieve real-time plot for both cars 12 and 20 in time for presentation : due to SSH issues with VM