

# Ranking Prediction for Indianapolis 500 Race

Yashaswini Dhatrika

School of Informatics, Computing and Engineering  
ydhatrik@iu.edu

Utkarsh Kumar

School of Informatics, Computing and Engineering  
utkumar@iu.edu

## ABSTRACT

While being one of the most challenging problems, forecasting and predicting ranking is being explored and has been gaining popularity with availability of more data and improved prediction techniques in real time. We present the ranking prediction algorithms which predicts the leading car in Indianapolis 500 race which is held every year by using 7 years of data from IndyStats website. Our approach is to extract and transform data to the desirable form, then perform exploratory analysis in order to understand the key features for the ranking and then apply different algorithms (Linear, Bagging, Boosting and Deep Learning Models (LSTM, Seq2Seq, Seq2Seq with Attention) models for predicting the rank. As the last step of process the comparative study is performed to analysis the pros /cons of each model deployed

**Keywords :** Rank Forecast, Rank Prediction, Indianapolis 500, RNN

## 1 INTRODUCTION

Advances in Machine Learning and Deep Learning today have paved way for solving more and more problems. One such challenge is predicting the rank of cars in races. This is particularly a difficult problem because of the uncertainty involved and interplay of a large number of factor involved. However, with the availability of large amount of data from sensors and other sources, we are optimistic about the possibility of finding some patterns that affect the ranking more than others. In this view, we are working on forecasting ranking and studying variables involved of Indianapolis 500 race. The race involves around 33 cars which compete for 200 laps for across 7 years (2013-2019). Numerous sensors collect data in real time such as vehicle speed, engine speed, throttle, lap times, etc. The size of this data could be between 500 MB to 2 GB. In this project, the main objective is to analyze the data given, clean it and build model that can forecast ranking of cars. We have performed the various steps like Data Extraction, Data Preprocessing, Exploratory Data Analysis to get more understanding of the data and applying a various models (Linear, Bagging and Boosting, Deep Learning Models) to accurately predict the rankings.

## 2 RELATED WORK

Unfortunately, till now there has not been any substantial work in the field of ranking prediction for car races but we have managed to find some similar work in other fields such as Harness Racing. There are other resources available too such as the article Forecasting F1 Race Outcomes by Chris Tucker which talks about predicting the outcome for the 2015 season.

The articles on Horse Racing ranking/outcome predictions involve a range of methods. Papers [1] and [2] talk about predicting rank based on probability estimation models. Machine learning methods such as Support Vector Machines have been explored in the paper [3]. More recently, neural networks and deep learning



**Figure 1:** 103rd Indianapolis 500 at Indianapolis Motor Speedway on May 26, 2019 in Indianapolis, by Clive Rose/Getty Images.

have also been utilized in predicting Horse Race outcomes and have shows very promising results. Papers [4] and [8], using DNN have been able to achieve accuracy of 74% and 77%, respectively.

All these studies and results are exciting and useful as they provide direction and basis for our research. Figure 1.

## 3 METHODOLOGY

The steps followed for the lead rank prediction are data extraction, data pre-processing, exploratory data analysis and application of various models like Linear Models (ARIMA), Random Forest, Gradient Boosting Regression, Deep Learning Models (LSTM, Encoder-Decoder with attention). And in the final step the models are compared based on various factors like accuracy, stability, time complexity etc.. The following sections discuss each of these in detail.

### 3.1 Data and Preprocessing

For our initial analysis and Base line Model we have used the Completed lap result records from the log data which contains all the Multi-Loop Protocol and Results Protocol data records. We filtered the completed lap result data records based on "\$C" identifier. As can be seen from the Figure 2, many fields in the data are in Hexadecimal form including date/time, ranks and counts. We had to convert each of these hexadecimal values into respective formats using python. Finally, we also removed duplicate rows.

## SC – Completed lap results

Fieldname	Data description	Comments
Rank	1 – FFFF	
Car number	characters	4 characters maximum.
Unique identification	0 – FFFFFFFF	This will be the transponder number.
Completed laps	0 – FFFF	Number of completed laps.
Elapsed time	0 – FFFFFFFF	Elapsed Time in ten thousandths sec (hex chars)
Last laptime	0 – FFFFFFFF	Time in ten thousandths sec (hex chars)
Lap status	T, P	Indicates where the lap was completed T = Track, P = Pit lane.
Fastest laptime	0 – FFFFFFFF	Time in ten thousandths sec (hex chars)
Fastest lap	0 – FFFF	
Time behind leader	0 – FFFFFFFF	Time in ten thousandths sec (hex chars)
Laps behind leader	0 – FFFF	
Time behind prec	0 – FFFFFFFF	Time in ten thousandths sec (hex chars) behind preceding car.
Laps behind prec	0 – FFFF	Laps behind preceding car.
Overall rank	0 – FFFF	Rank of car in all sessions of this type.
Overall best laptime	0 – FFFFFFFF	Time in ten thousandths sec (hex chars)
Current status	characters	"Active" or text from comment field from operator edit competitor option. Typically used for reason out of race.  Examples: DNQ, DNS, DNF, Contact, Mechanical, Garage, etc
Track Status	G, Y, R, W, K, U	G = Green Y = Yellow R = Red W = White K = Checkered U = Unflagged (warm up)
Pit stop count	0 – FFFF	Number of pitstops in this session
Last pitstop lap	0 – FFFF	Lap number the car last pitted.

Figure 2: Initial Data -Completed Lap Results

But Later we have focused mainly on using the Completed lap results data an pit stop data extracted from the Stats webpage on IndyCar website(<https://www.indycar.com/Stats>) for 2013-2019 years.The raw data(2013-2019) obtained is shown in the Figure 4.. Some transformation are performed to this data. The following are the steps:

- (1) the pdf file(2013-2019) is converted to excel file.
- (2) From the excel selected rows and columns are selected and combined with last pit stop data to extract time passed since last pit stop.
- (3) Then for each car starting or lap time is calculated using linear interpolation method
- (4) Then the lap time data points are converted into lap distance using the Figure 3 interpolation method

## Convert Lap Time to Time Series Lap Distance

## Interpolation

$$\text{Lap Distance}[T] = \text{Lap Length} * (T - T_0) / (\text{Lap Time})$$

Figure 3: Lap Distance Interpolation

- (5) the final data contains columns of time lap distance and last pit stop for all the 33 cars. So in totality there are 33\*2 columns.The final data set is showed in Figure 5

Section Data for Car 10 - Rosenqvist, Felix (R)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
Lap	T	T1 to S1	S1 to T2	T2 to S2	S2 to T3	T3 to S3	S3 to T4	T4 to S4	S4 to T5	T5 to S5	S5 to T6	T6 to S6	S6 to T7	T7 to S7	S7 to T8	T8 to S8	S8 to T9	T9 to S9	S9 to T10	T10 to S10	S10 to T11	T11 to S11	S11 to T12	T12 to S12	S12 to T13	T13 to S13	S13 to T14	T14 to S14	S14 to T15	T15 to S15	S15 to T16	T16 to S16	S16 to T17	T17 to S17	S17 to T18	T18 to S18	S18 to T19	T19 to S19	S19 to T20	T20 to S20	S20 to T21	T21 to S21	S21 to T22	T22 to S22	S22 to T23	T23 to S23	S23 to T24	T24 to S24	S24 to T25	T25 to S25	S25 to T26	T26 to S26	S26 to T27	T27 to S27	S27 to T28	T28 to S28	S28 to T29	T29 to S29	S29 to T30	T30 to S30	S30 to T31	T31 to S31	S31 to T32	T32 to S32	S32 to T33	T33 to S33	S33 to T34	T34 to S34	S34 to T35	T35 to S35	S35 to T36	T36 to S36	S36 to T37	T37 to S37	S37 to T38	T38 to S38	S38 to T39	T39 to S39	S39 to T40	T40 to S40	S40 to T41	T41 to S41	S41 to T42	T42 to S42	S42 to T43	T43 to S43	S43 to T44	T44 to S44	S44 to T45	T45 to S45	S45 to T46	T46 to S46	S46 to T47	T47 to S47	S47 to T48	T48 to S48	S48 to T49	T49 to S49	S49 to T50	T50 to S50	S50 to T51	T51 to S51	S51 to T52	T52 to S52	S52 to T53	T53 to S53	S53 to T54	T54 to S54	S54 to T55	T55 to S55	S55 to T56	T56 to S56	S56 to T57	T57 to S57	S57 to T58	T58 to S58	S58 to T59	T59 to S59	S59 to T60	T60 to S60	S60 to T61	T61 to S61	S61 to T62	T62 to S62	S62 to T63	T63 to S63	S63 to T64	T64 to S64	S64 to T65	T65 to S65	S65 to T66	T66 to S66	S66 to T67	T67 to S67	S67 to T68	T68 to S68	S68 to T69	T69 to S69	S69 to T70	T70 to S70	S70 to T71	T71 to S71	S71 to T72	T72 to S72	S72 to T73	T73 to S73	S73 to T74	T74 to S74	S74 to T75	T75 to S75	S75 to T76	T76 to S76	S76 to T77	T77 to S77	S77 to T78	T78 to S78	S78 to T79	T79 to S79	S79 to T80	T80 to S80	S80 to T81	T81 to S81	S81 to T82	T82 to S82	S82 to T83	T83 to S83	S83 to T84	T84 to S84	S84 to T85	T85 to S85	S85 to T86	T86 to S86	S86 to T87	T87 to S87	S87 to T88	T88 to S88	S88 to T89	T89 to S89	S89 to T90	T90 to S90	S90 to T91	T91 to S91	S91 to T92	T92 to S92	S92 to T93	T93 to S93	S93 to T94	T94 to S94	S94 to T95	T95 to S95	S95 to T96	T96 to S96	S96 to T97	T97 to S97	S97 to T98	T98 to S98	S98 to T99	T99 to S99	S99 to T100	T100 to S100	S100 to T101	T101 to S101	S101 to T102	T102 to S102	S102 to T103	T103 to S103	S103 to T104	T104 to S104	S104 to T105	T105 to S105	S105 to T106	T106 to S106	S106 to T107	T107 to S107	S107 to T108	T108 to S108	S108 to T109	T109 to S109	S109 to T110	T110 to S110	S110 to T111	T111 to S111	S111 to T112	T112 to S112	S112 to T113	T113 to S113	S113 to T114	T114 to S114	S114 to T115	T115 to S115	S115 to T116	T116 to S116	S116 to T117	T117 to S117	S117 to T118	T118 to S118	S118 to T119	T119 to S119	S119 to T120	T120 to S120	S120 to T121	T121 to S121	S121 to T122	T122 to S122	S122 to T123	T123 to S123	S123 to T124	T124 to S124	S124 to T125	T125 to S125	S125 to T126	T126 to S126	S126 to T127	T127 to S127	S127 to T128	T128 to S128	S128 to T129	T129 to S129	S129 to T130	T130 to S130	S130 to T131	T131 to S131	S131 to T132	T132 to S132	S132 to T133	T133 to S133	S133 to T134	T134 to S134	S134 to T135	T135 to S135	S135 to T136	T136 to S136	S136 to T137	T137 to S137	S137 to T138	T138 to S138	S138 to T139	T139 to S139	S139 to T140	T140 to S140	S140 to T141	T141 to S141	S141 to T142	T142 to S142	S142 to T143	T143 to S143	S143 to T144	T144 to S144	S144 to T145	T145 to S145	S145 to T146	T146 to S146	S146 to T147	T147 to S147	S147 to T148	T148 to S148	S148 to T149	T149 to S149	S149 to T150	T150 to S150	S150 to T151	T151 to S151	S151 to T152	T152 to S152	S152 to T153	T153 to S153	S153 to T154	T154 to S154	S154 to T155	T155 to S155	S155 to T156	T156 to S156	S156 to T157	T157 to S157	S157 to T158	T158 to S158	S158 to T159	T159 to S159	S159 to T160	T160 to S160	S160 to T161	T161 to S161	S161 to T162	T162 to S162	S162 to T163	T163 to S163	S163 to T164	T164 to S164	S164 to T165	T165 to S165	S165 to T166	T166 to S166	S166 to T167	T167 to S167	S167 to T168	T168 to S168	S168 to T169	T169 to S169	S169 to T170	T170 to S170	S170 to T171	T171 to S171	S171 to T172	T172 to S172	S172 to T173	T173 to S173	S173 to T174	T174 to S174	S174 to T175	T175 to S175	S175 to T176	T176 to S176	S176 to T177	T177 to S177	S177 to T178	T178 to S178	S178 to T179	T179 to S179	S179 to T180	T180 to S180	S180 to T181	T181 to S181	S181 to T182	T182 to S182	S182 to T183	T183 to S183	S183 to T184	T184 to S184	S184 to T185	T185 to S185	S185 to T186	T186 to S186	S186 to T187	T187 to S187	S187 to T188	T188 to S188	S188 to T189	T189 to S189	S189 to T190	T190 to S190	S190 to T191	T191 to S191	S191 to T192	T192 to S192	S192 to T193	T193 to S193	S193 to T194	T194 to S194	S194 to T195	T195 to S195	S195 to T196	T196 to S196	S196 to T197	T197 to S197	S197 to T198	T198 to S198	S198 to T199	T199 to S199	S199 to T200	T200 to S200	S200 to T201	T201 to S201	S201 to T202	T202 to S202	S202 to T203	T203 to S203	S203 to T204	T204 to S204	S204 to T205	T205 to S205	S205 to T206	T206 to S206	S206 to T207	T207 to S207	S207 to T208	T208 to S208	S208 to T209	T209 to S209	S209 to T210	T210 to S210	S210 to T211	T211 to S211	S211 to T212	T212 to S212	S212 to T213	T213 to S213	S213 to T214	T214 to S214	S214 to T215	T215 to S215	S215 to T216	T216 to S216	S216 to T217	T217 to S217	S217 to T218	T218 to S218	S218 to T219	T219 to S219	S219 to T220	T220 to S220	S220 to T221	T221 to S221	S221 to T222	T222 to S222	S222 to T223	T223 to S223	S223 to T224	T224 to S224	S224 to T225	T225 to S225	S225 to T226	T226 to S226	S226 to T227	T227 to S227	S227 to T228	T228 to S228	S228 to T229	T229 to S229	S229 to T230	T230 to S230	S230 to T231	T231 to S231	S231 to T232	T232 to S232	S232 to T233	T233 to S233	S233 to T234	T234 to S234	S234 to T235	T235 to S235	S235 to T236	T236 to S236	S236 to T237	T237 to S237	S237 to T238	T238 to S238	S238 to T239	T239 to S239	S239 to T240	T240 to S240	S240 to T241	T241 to S241	S241 to T242	T242 to S242	S242 to T243	T243 to S243	S243 to T244	T244 to S244	S244 to T245	T245 to S245	S245 to T246	T246 to S246	S246 to T247	T247 to S247	S247 to T248	T248 to S248	S248 to T249	T249 to S249	S249 to T250	T250 to S250	S250 to T251	T251 to S251	S251 to T252	T252 to S252	S252 to T253	T253 to S253	S253 to T254	T254 to S254	S254 to T255	T255 to S255	S255 to T256	T256 to S256	S256 to T257	T257 to S257	S257 to T258	T258 to S258	S258 to T259	T259 to S259	S259 to T260	T260 to S260	S260 to T261	T261 to S261	S261 to T262	T262 to S262	S262 to T263	T263 to S263	S263 to T264	T264 to S264	S264 to T265	T265 to S265	S265 to T266	T266 to S266	S266 to T267	T267 to S267	S267 to T268	T268 to S268	S268 to T269	T269 to S269	S269 to T270	T270 to S270	S270 to T271	T271 to S271	S271 to T272	T272 to S272	S272 to T273	T273 to S273	S273 to T274	T274 to S274	S274 to T275	T275 to S275	S275 to T276	T276 to S276	S276 to T277	T277 to S277	S277 to T278	T278 to S278	S278 to T279	T279 to S279	S279 to T280	T280 to S280	S280 to T281	T281 to S281	S281 to T282	T282 to S282	S282 to T283	T283 to S283	S283 to T284	T284 to S284	S284 to T285	T285 to S285	S285 to T286	T286 to S286	S286 to T287	T287 to S287	S287 to T288	T288 to S288	S288 to T289	T289 to S289	S289 to T290	T290 to S290	S290 to T291	T291 to S291	S291 to T292	T292 to S292	S292 to T293	T293 to S293	S293 to T294	T294 to S294	S294 to T295	T295 to S295	S295 to T296	T296 to S296	S296 to T297	T297 to S297	S297 to T298	T298 to S298	S298 to T299	T299 to S299	S299 to T300	T300 to S300	S300 to T301	T301 to S301	S301 to T302	T302 to S302	S302 to T303	T303 to S303	S303 to T304	T304 to S304	S304 to T305	T305 to S305	S305 to T306	T306 to S306	S306 to T307	T307 to S307	S307 to T308	T308 to S308	S308 to T309	T309 to S309	S309 to T310	T310 to S310	S310 to T311	T311 to S311	S311 to T312	T312 to S312	S312 to T313	T313 to S313	S313 to T314	T314 to S314	S314 to T315	T315 to S315	S315 to T316	T316 to S316	S316 to T317	T317 to S317	S317 to T318	T318 to S318	S318 to T319	T319 to S319	S319 to T320	T320 to S320	S320 to T321	T321 to S321	S321 to T322	T322 to S322	S322 to T323	T323 to S323	S323 to T324	T324 to S324	S324 to T325	T325 to S325	S325 to T326	T326 to S326	S326 to T327	T327 to S327	S327 to T328	T328 to S328	S328 to T329	T329 to S329	S329 to T330	T330 to S330	S330 to T331	T331 to S331	S331 to T332	T332 to S332	S332 to T333	T333 to S333	S333 to T334	T334 to S334	S334 to T335	T335 to S335	S335 to T336	T336 to S336	S336 to T337	T337 to S337	S337 to T338	T338 to S338	S338 to T339	T339 to S339	S339 to T340	T340 to S340	S340 to T341	T341 to S341	S341 to T342	T342 to S342	S342 to T343	T343 to S343	S343 to T344	T344 to S344	S344 to T345	T345 to S345	S345 to T346	T346 to S346	S346 to T347	T347 to S347	S347 to T348	T348 to S348	S348 to T349	T349 to S349	S349 to T350	T350 to S350	S350 to T351	T351 to S351	S351 to T352	T352 to S352	S352 to T353	T353 to S353	S353 to T354	T354 to S354	S354 to T355	T355 to S355	S355 to T356	T356 to S356	S356 to T357	T357 to S357	S357 to T358	T358 to S358	S358 to T359	T359 to S359	S359 to T360	T360 to S360	S360 to T361	T361 to S361	S361 to T362	T362 to S362	S362 to T363	T363 to S363	S363 to T364	T364 to S364	S364 to T365	T365 to S365	S365 to T366	T366 to S366	S366 to T367	T367 to S367	S367 to T368	T368 to S368	S368 to T369	T369 to S369	S369 to T370	T370 to S370	S370 to T371	T371 to S371	S371 to T372	T372 to S372	S372 to T373	T373 to S373	S373 to T374	T374 to S374	S374 to T375	T375 to S375	S375 to T376	T376 to S376	S376 to T377	T377 to S377	S377 to T378	T378 to S378	S378 to T379	T379 to S379	S379 to T380	T380 to S380	S380 to T381	T381 to S381	S381 to T382	T382 to S382	S382 to T383	T383 to S383	S383 to T384	T384 to S384	S384 to T385	T385 to S385	S385 to T386	T386 to S386	S386 to T387	T387 to S387	S387 to T388	T388 to S388	S388 to T389	T389 to S389	S389 to T390	T390 to S390	S390 to T391	T391 to S391	S391 to T392	T392 to S392	S392 to T393	T393 to S393	S393 to T394	T394 to S394	S394 to T395	T395 to S395	S395 to T396	T396 to S396	S396 to T397	T397 to S397	S397 to T398	T398 to S398	S398 to T399	T399 to S399	S399 to T400	T400 to S400	S400 to T401	T401 to S401	S401 to T402	T402 to S402	S402 to T403	T403 to S403	S403 to T404	T404 to S404	S404 to T405	T405 to S405	S405 to T406	T406 to S406	S406 to T407	T407 to S407	S407 to T408	T408 to S408	S408 to T409	T409 to S409	S409 to T410	T410 to S410	S410 to T411	T411 to S411	S411 to T412	T412 to S412	S412 to T413	T413 to S413	S413 to T414	T414 to S414	S414 to T415	T415 to S415	S415 to T416	T416 to S416	S416 to T417	T417 to S417	S417 to T418	T418 to S418	S418 to T419	T419 to S419	S419 to T420	T420 to S420	S420 to T421	T421 to S421	S421 to T422	T422 to S422	S422 to T423	T423 to S423	S423 to T424	T424 to S424	S424 to T425	T425 to S425	S425 to T426	T426 to S426	S426 to T427	T427 to S427	S427 to T428	T428 to S428	S428 to T429	T429 to S429	S429 to T430	T430 to S430	S430 to T431	T431 to S431	S431 to T432	T432 to S432	S432 to T433	T433 to S433	S433 to T434	T434 to S434	S434 to T435	T435 to S435	S435 to T436	T436 to S436	S436 to T437	T437 to S437	S437 to T438	T438 to S438	S438 to T439	T439 to S439	S439 to T440	T440 to S440	S440 to T441	T441 to S441	S441 to T442	T442 to S442	S442 to T443	T443 to S443	S443 to T444	T444 to S444	S444 to T445	T445 to S445	S445 to T446	T446 to S446	S446 to T447	T447 to S447	S447 to T448	T448 to S448	S448 to T449	T449 to S449	S449 to T450	T450 to S450	S450 to T451	T451 to S451	S451 to T452	T452 to S452	S452 to T453	T453 to S453	S453 to T454	T454 to S454	S454 to T455	T455 to S455	S455 to T456	T456 to S456	S456 to T457	T457 to S457	S457 to T458	T458 to S458	S458 to T459	T459 to S459	S459 to T460	T460 to S460	S460 to T461	T461 to S461	S461 to T462	T462 to S462	S462 to T463	T463 to S463	S463 to T464	T464 to S464	S464 to T465	T465 to S465	S465 to T466	T466 to S466	S466 to T467	T467 to S467	S467 to T468	T468 to S468	S468 to T469	T469 to S469	S469 to T470	T470 to S470	S470 to T471	T471 to S471	S471 to T472	T472 to S472	S472 to T473	T473 to S473	S473 to T474	T474 to S474	S474 to T475	T475 to S475	S475 to T476	T476 to S476	S476 to T477	T477 to S477	S477 to T478	T478 to S478	S478 to T479	T479 to S479	S479 to T480	T480 to S480	S480 to T481	T481 to S481	S481 to T482	T482 to S482	S482 to T483	T483 to S483	S483 to T484	T484 to S484	S484 to T485	T485 to S485	S485 to T486	T486 to S486	S486 to T487	T487 to S487	S487 to T488	T488 to S488	S488 to T489	T489 to S489	S489 to T490	T490 to S490	S490 to T491	T491 to S491	S491 to T492</

- (2) Number of pitstops taken by each player who completed the race and their corresponding rank to learn about their correlation.
- (3) Average time behind leader for each player over 200 laps who completed the race and their corresponding rank to learn about their correlation.
- (4) Starting position for each player who completed the race and their corresponding rank to learn about their correlation.
- (5) Distribution of lap times for each player who completed the race and those who didn't to recognize patterns and strategies.

This shed light on some interesting patterns in the data which is discussed elaborately in the results section.

### 3.3 Baseline Model

**3.3.1 Overview of ARIMA Time Series Analysis.** ARIMA processes are a class of stochastic processes used to analyze time series. The application of the ARIMA methodology for the study of time series analysis is due to Box and Jenkins Box and Jenkins in 1970 introduced the ARIMA model. It also referred to as Box-Jenkins methodology composed of set of activities for identifying, estimating and diagnosing ARIMA models with time series data. The model is most prominent methods in financial forecasting [7] [6]. ARIMA models have shown efficient capability to generate short-term forecasts. It constantly outperformed complex structural models in short-term prediction[5]. In ARIMA model, the future value of a variable is a linear combination of past values and past errors, expressed as follows:

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \dots - \theta_q \varepsilon_{t-q} \quad (1)$$

$Y_t$  is the actual value and  $\varepsilon_t$  is the random error at  $t$ ,  $\phi_i$  and  $\theta_j$  are the coefficients,  $p$  and  $q$  are integers that are often referred to as autoregressive and moving average, respectively.

#### 3.3.2 Implementation.

- (1) Lap Time is predicted using the ARIMA model for each car.
- (2) While predicting lap time for a car, each car is considered independently from another.
- (3) As there 200 laps in total, the predictions are done in windows. Windows considered are after 20,50,100,150,199.
- (4) Python's `autoarima()` function is used which optimizes the parameter based on the Akaike information criterion (AIC) and mean squared error(MSE).
- (5) After each window prediction using ARIMA for all the cars, based on the predicted lap-time, rank is obtained and compared to the actual rank output
- (6) Finally the predicted and actual rank are plotted, to see how closely they are related

### 3.4 Random Forest Regression

**3.4.1 Overview.** Random forest is a Supervised Learning algorithm which uses ensemble learning method for classification and regression. Random forest is a bagging technique and not a boosting technique. The trees in random forests are run in parallel. There is no interaction between these trees while building the trees. It

operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

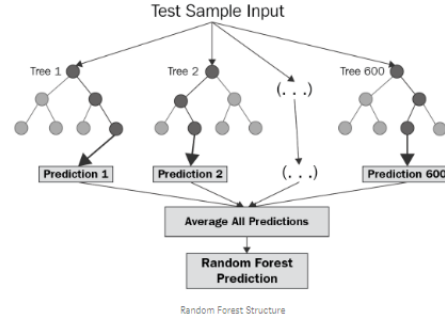


Figure 6: Random Forest

A random forest is a meta-estimator (i.e. it combines the result of multiple predictions) which aggregates many decision trees, with some helpful modifications: The number of features that can be split on at each node is limited to some percentage of the total (which is known as the hyper-parameter). This ensures that the ensemble model does not rely too heavily on any individual feature, and makes fair use of all potentially predictive features. Each tree draws a random sample from the original data set when generating its splits, adding a further element of randomness that prevents over-fitting. These decision tree classifiers can be aggregated into a random forest ensemble which combines their input.

#### 3.4.2 Implementation.

- (1) Creating of labels for each data points ( 1,2,3...33).
- (2) Creating lag variable for the lap distance (lag10, lag 20...lag 200)
- (3) Normalizing the data
- (4) Split of train and test. (70 percent of first-time stamp of each year for training and remaining data points are for validation.)
- (5) Using scikit learn library for implementation of Random Forest Regressor.K-fold valid is performed using Grid search to hyper tune the Hyper Parameters like number of decision trees, disorder calculating and depth of the tree.
- (6) Prediction is done on the test data using the optimized hyper-parameters

### 3.5 Gradient Booster Regression

**3.5.1 Overview .** Gradient boosting involves three elements:

- (1) A loss function to be optimized.
- (2) A weak learner to make predictions.
- (3) An additive model to add weak learners to minimize the loss function

1. Loss Function The loss function used depends on the type of problem being solved. It must be differentiable, but many standard loss functions are supported and you can define your own. For example, regression may use a squared error and classification may

use logarithmic loss. A benefit of the gradient boosting framework is that a new boosting algorithm does not have to be derived for each loss function that may want to be used, instead, it is a generic enough framework that any differentiable loss function can be used.

2. Weak Learner: Decision trees are used as the weak learner in gradient boosting. Specifically regression trees are used that output real values for splits and whose output can be added together, allowing subsequent models outputs to be added and “correct” the residuals in the predictions. Trees are constructed in a greedy manner, choosing the best split points based on purity scores like Gini or to minimize the loss. Initially, such as in the case of AdaBoost, very short decision trees were used that only had a single split, called a decision stump. Larger trees can be used generally with 4-to-8 levels. It is common to constrain the weak learners in specific ways, such as a maximum number of layers, nodes, splits or leaf nodes. This is to ensure that the learners remain weak, but can still be constructed in a greedy manner.

3. Additive Model: Trees are added one at a time, and existing trees in the model are not changed. A gradient descent procedure is used to minimize the loss when adding trees. Traditionally, gradient descent is used to minimize a set of parameters, such as the coefficients in a regression equation or weights in a neural network. After calculating error or loss, the weights are updated to minimize that error. Instead of parameters, we have weak learner sub-models or more specifically decision trees. After calculating the loss, to perform the gradient descent procedure, we must add a tree to the model that reduces the loss (i.e. follow the gradient). We do this by parameterizing the tree, then modify the parameters of the tree and move in the right direction by (reducing the residual loss. The output for the new tree is then added to the output of the existing sequence of trees in an effort to correct or improve the final output of the model. A fixed number of trees are added or training stops once loss reaches an acceptable level or no longer improves on an external validation dataset.

### 3.5.2 Implementation .

- (1) Creating of labels for each data points ( 1,2,3...33).
- (2) Creating lag variable for the lap distance (lag10, lag 20...lag 200)
- (3) Normalizing the data
- (4) Split of train and test. (70 percent of first-time stamp of each year for training and remaining data points are for validation.)
- (5) Using scikit learn library for implementation of Random Forest Regression. K-fold valid is performed using Grid search to hyper tune the Hyper Parameters like number of decision trees, Learning rate and depth of the tree
- (6) Prediction is done on the test data using the optimized hyper-parameters

## 3.6 LSTM and Attention based Deep Neural Network

3.6.1 Overview . We also used an LSTM and an attention based deep neural network to forecast ranking. This is a multivariate time series model with single step prediction where we have taken frames of data to predict the probability distribution of the leading

car. The LSTM network is a simple RNN with LSTM cell and a dense layer with softmax activation function to output the probability distribution of leading car. In the next experiment we used attention decoder to capture important data in our feature map that affects ranking to give a better prediction. These experiments were based on Tensorflow tutorial for multivariate time series model. Also, since Tensorflow before 2.1.0 doesn't have an inbuilt Attention layer, we used Zafarali Ahmed's "How to Visualize Your Recurrent Neural Network with Attention in Keras" in 2017 and GitHub project called "keras-attention".

3.6.2 Limitations of Encoder-Decoder Models. The main drawback of Encoder – Decoder Architecture is that the decoder receives only the encoder vector i.e., only the last encoder hidden state. This encoder vector is expected to summarize the entire input sequence and for long input sentences, we expect the decoder to create the output just based on one vector sequence. If the given input sentence is too large this will it harder for the decoder to predict the output sentence given the input sentence. For the purpose of overcoming this difficulty, we are using Attention-Based architecture.

3.6.3 Architecture of Attention Based SEQ2SEQ Model . SEQ2SEQ Model with Attention Layer consists of the following components:

- (1) Encoder: The encoder is responsible for stepping through the input time steps and encoding the entire sequence into a fixed length vector called a context vector.
- (2) Decoder: The decoder is responsible for stepping through the output time steps while reading from the context vector.
- (3) The Attention layer is considered as an interface between Encoder and the Decoder that provides each unit in the decoder with the information from the encoder hidden state.

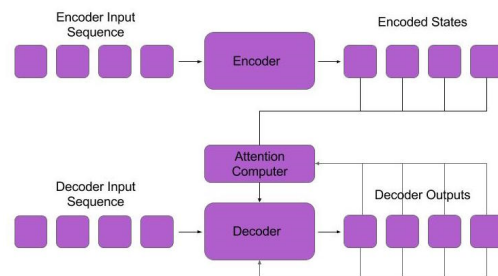


Figure 7: Overview of architecture with attention layer in Seq2Seq model



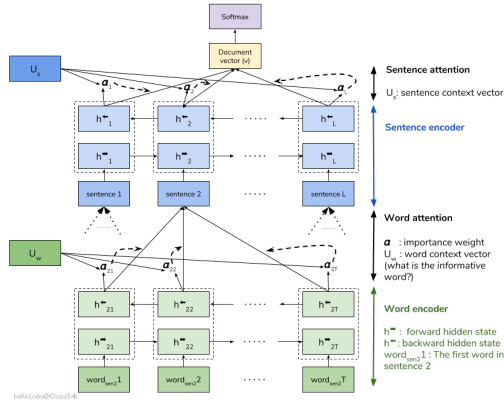


Figure 8: Detailed figure of attention mechanism

**3.6.4 Input Data and Processing.** The input data consisted of 33 cars having 2 features each: distance and time since last pit stop. So, in total we had 66 features in the input data and around 10,000 records per race representing each second. We used data from years 2013 to 2018. Data from 2013 to 2017 were used for training and 2018 was used for validation. For input, we used frames of data consisting of a certain number of steps for each experiment. The size of each input frame was 20. We also considered top 20 ranked cars as well as all 33 cars in our experiments. We achieved better results with keeping only top 20 cars in our data based on the final rank.

The data was standardized in two phases – first the pit stop time column was standardized for each year data separately. The distance column was standardized for each frame using the min max scaler function. To do this we took the maximum value in the last row of each frame and the minimum value in the first row of each frame. Then we subtracted the minimum value from all frame values for distance column and divided by the range.

**3.6.5 Models Architecture.** There were two models we used. The architecture of each model was as below:

- (1) First a simple LSTM network, with 2 LSTM layers consisting of 128 and 64 units, respectively. Another Dense layer for 64 units and finally a “SoftMax” layer for predicting the output.
- (2) The other attention based network consisted of - two layers attention encoder decoder, encoder layer with 150 units, (window size, number of features) dimension and return sequence = True. The attention decoder was of the same shape.

## 4 RESULTS

### 4.1 Exploratory Data Analysis

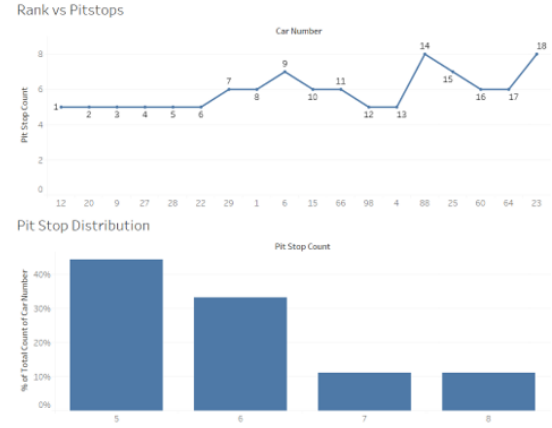


Figure 9: Analysis of pitstops in context with the final rank. Top graph shows Rank vs Pitstops and bottom graph shows the distribution of pitstops among the players who completed the race.

As we can see from Figure 9, most of the player who completed the race tried to take less number of pitstops. Top graph also indicates that players who ranked from rank 1 to 5 took 5 pitstops which was least in the race. Also, players who have ranked lower taken more pitstops. This means that on average, there is some correlation (negative) between the number of pitstops and final rank.

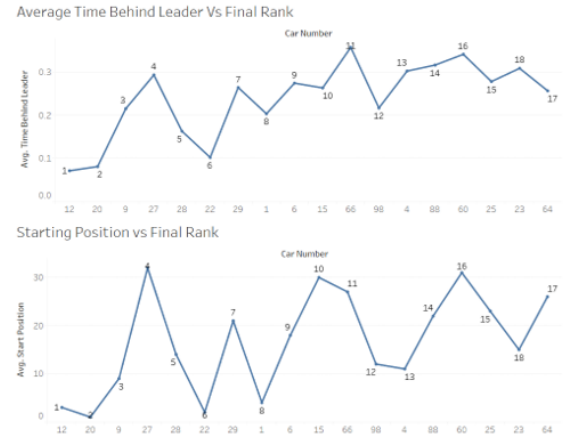
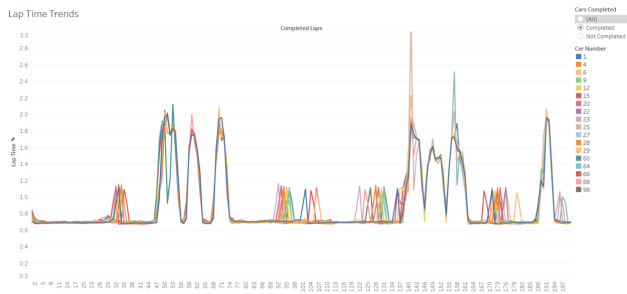


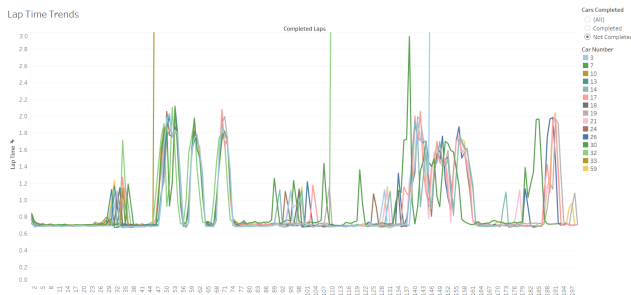
Figure 10: Analysis of average time behind leader and starting position in context with the final rank. Top graph shows Av. Time Behind Leader vs Final Rank and bottom graph shows Starting Position vs Final Rank for the players who completed the race.

From Figure 11, there seems to be a weak correlation between starting position and final rank. However, the graph Average Time Behind Leader vs Final Rank suggests that players who end up ranking high on average remain close to the leader. It indicates that time behind leader might prove important in forecasting rank in the race.

Finally from Figure 12, we can see that player who complete the race maintain similar strategies and remain consistent throughout the race. They also happen to take pitstops near the same time. The strategies only differ slightly towards the end of the race. But for players who do not complete the race, Figure 10, there is more inconsistency. These could be due to external or internal factors.

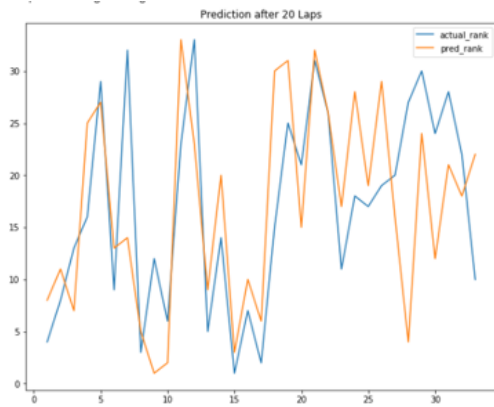


**Figure 11: Lap time trends for players who completed the race.**

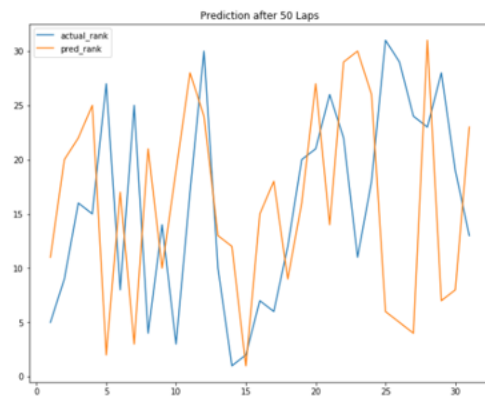


**Figure 12: Lap time trends for players who did not complete the race.**

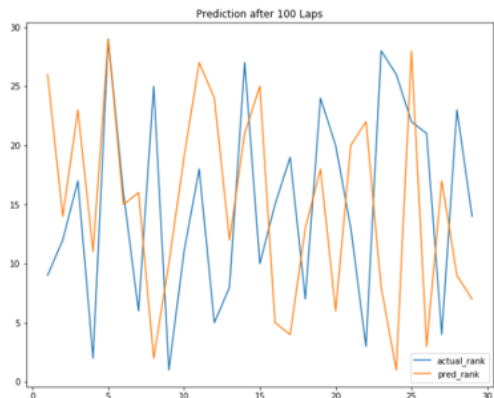
## 4.2 Baseline Model-ARIMA Model



**Figure 13: Predictions Vs Actual after 20 laps**



**Figure 14: Predictions Vs Actual after 50 laps**



**Figure 15: Predictions Vs Actual after 100 laps**

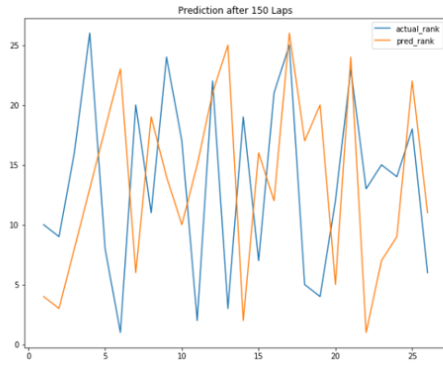


Figure 16: Predictions Vs Actual after 150 laps

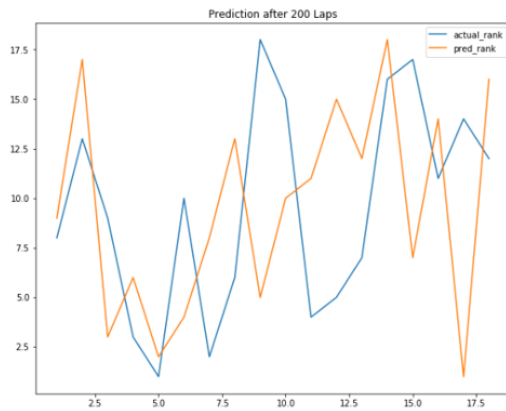


Figure 17: Predictions Vs Actual after 200 laps

From all the above figure we can notice that for same of cars the prediction is quite close but for some other there is significant difference. So this may be due to the assumption that all the cars are considered independently while predicting. So next steps would be to incorporate the dependencies.

### 4.3 Random and Gradient Forest

1. The accuracy obtained by Random forest is 42%. 2. The accuracy obtained by Gradient Boosting is 44%

### 4.4 LSTM and Attention based Deep Neural Network

The training took around 1.5 hours on full data and the validation accuracy for both models were not very stable. We were able to achieve highest accuracy of 55% with attention model and 48% without attention model. This variation in accuracy could be due to small differences in the step values within the frame for distance. For instance, one of the values in the first row was 0.998863739 and one of the values in the last row 0.999436234. As we can see the difference is very low this might make the model unstable. Even if we try increasing the frame size or make wider steps the accuracy did not improve. We also did try different configurations

for step sizes, frame sizes and the range of ranking prediction. For some models, we saw very high training accuracy > 90% but less validation accuracy and fluctuating validation accuracy. This might be caused due to weight swinging due to values in input data.

For future research, we believe adding high quality telemetry data will help achieve better accuracy as it would carry more information for the model to learn from. Information such as pattern in throttle, gears and acceleration will greatly help to achieve more accuracy. But this data needs to align properly with each other and lap result data.

### 4.5 Comparison and Conclusion

Sr.No	Model	Accuracy	Stability	Training Time	Hyperparam.	Simplicity
1.	Random Forest	42%	Yes	1.5 Hr	Few( 4)	Simple
2.	Gradient Forest	44%	Yes	1.5 Hr	Few( 4)	Simple
3.	LSTM	48%	No	30min to 1.5 Hr	Many(>4)	Complex
4.	Attention	55%	Yes	30min to 1.5 Hr	Many(>4)	Complex

- (1) From EDA, the starting position, last pit stop displayed correlation with rank
- (2) By deploying various models, we found the Seq2Seq Attention Model has performed well while compared to other models.
- (3) But when you consider the stability or variation of accuracy then the Random Forest and Gradient Boosting models are stable over other Deep Learning Models.

### 4.6 Code Repository

- (1) LSTM and Attention Notebooks
- (2) ARIMA, Bagging and Boosting Models

### REFERENCES

- [1] Fredrik Armerin, Jonas Hallgren, and Timo Koski. Forecasting ranking in harness racing using probabilities induced by expected positions. *Applied Artificial Intelligence*, pages 1–19, 10 2018.
- [2] Randall Chapman and Richard Staelin. Exploiting rank ordered choice set data within the stochastic utility model. *Journal of Marketing Research*, 19, 08 1982.
- [3] Wai-Chung Chung, Chuan-Yu Chang, and Chien-Chuan Ko. A svm-based committee machine for prediction of hong kong horse racing. pages 1–4, 08 2017.
- [4] Elnaz Davoodi and Alireza Khanteymoori. Horse racing prediction using artificial neural networks. 06 2010.
- [5] Aidan Meyler, Geoff Kenny, and Terry Quinn. Forecasting irish inflation using arima models. 1998.
- [6] Rangsan Nochai and Titida Nochai. Arima model for forecasting oil palm price. In *Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, Statistics and applications*, pages 13–15, 2006.
- [7] Ping-Feng Pai and Chih-Sheng Lin. A hybrid arima and support vector machines model in stock price forecasting. *Omega*, 33(6):497–505, 2005.
- [8] Janett Williams and Yan Li. A case study using neural networks algorithms: Horse racing predictions in jamaica. pages 16–22, 01 2008.