

STAT3612 Appendix

December 7, 2021

| | count | mean | std | min | 25% | 50% | 75% | max |
|-----------------|-----------|-------------|-------------|------|--------|-----------|-------------|---------|
| assists | 4446966.0 | 0.233815 | 0.588573 | 0.0 | 0.0 | 0.0000 | 0.000000 | 22.0 |
| boosts | 4446966.0 | 1.106908 | 1.715794 | 0.0 | 0.0 | 0.0000 | 2.000000 | 33.0 |
| damageDealt | 4446966.0 | 130.717138 | 170.780621 | 0.0 | 0.0 | 84.2400 | 186.000000 | 6616.0 |
| DBNOs | 4446966.0 | 0.657876 | 1.145743 | 0.0 | 0.0 | 0.0000 | 1.000000 | 53.0 |
| headshotKills | 4446966.0 | 0.226820 | 0.602155 | 0.0 | 0.0 | 0.0000 | 0.000000 | 64.0 |
| heals | 4446966.0 | 1.370147 | 2.679982 | 0.0 | 0.0 | 0.0000 | 2.000000 | 80.0 |
| killPlace | 4446966.0 | 47.599350 | 27.462937 | 1.0 | 24.0 | 47.0000 | 71.000000 | 101.0 |
| killPoints | 4446966.0 | 505.006042 | 627.504896 | 0.0 | 0.0 | 0.0000 | 1172.000000 | 2170.0 |
| kills | 4446966.0 | 0.924783 | 1.558445 | 0.0 | 0.0 | 0.0000 | 1.000000 | 72.0 |
| killStreaks | 4446966.0 | 0.543955 | 0.710972 | 0.0 | 0.0 | 0.0000 | 1.000000 | 20.0 |
| longestKill | 4446966.0 | 22.997595 | 50.972619 | 0.0 | 0.0 | 0.0000 | 21.320000 | 1094.0 |
| matchDuration | 4446966.0 | 1579.506440 | 258.739856 | 9.0 | 1367.0 | 1438.0000 | 1851.000000 | 2237.0 |
| maxPlace | 4446966.0 | 44.504670 | 23.828105 | 1.0 | 28.0 | 30.0000 | 49.000000 | 100.0 |
| numGroups | 4446966.0 | 43.007593 | 23.289495 | 1.0 | 27.0 | 30.0000 | 47.000000 | 100.0 |
| rankPoints | 4446966.0 | 892.010457 | 736.647779 | -1.0 | -1.0 | 1443.0000 | 1500.000000 | 5910.0 |
| revives | 4446966.0 | 0.164659 | 0.472167 | 0.0 | 0.0 | 0.0000 | 0.000000 | 39.0 |
| rideDistance | 4446966.0 | 606.115669 | 1498.343513 | 0.0 | 0.0 | 0.0000 | 0.190975 | 40710.0 |
| roadKills | 4446966.0 | 0.003496 | 0.073373 | 0.0 | 0.0 | 0.0000 | 0.000000 | 18.0 |
| swimDistance | 4446966.0 | 4.509322 | 30.502199 | 0.0 | 0.0 | 0.0000 | 0.000000 | 3823.0 |
| teamKills | 4446966.0 | 0.023868 | 0.167394 | 0.0 | 0.0 | 0.0000 | 0.000000 | 12.0 |
| vehicleDestroys | 4446966.0 | 0.007918 | 0.092612 | 0.0 | 0.0 | 0.0000 | 0.000000 | 5.0 |
| walkDistance | 4446966.0 | 1154.217859 | 1183.497042 | 0.0 | 155.1 | 685.6000 | 1976.000000 | 25780.0 |
| weaponsAcquired | 4446966.0 | 3.660488 | 2.456544 | 0.0 | 2.0 | 3.0000 | 5.000000 | 236.0 |
| winPoints | 4446966.0 | 606.460131 | 739.700444 | 0.0 | 0.0 | 0.0000 | 1495.000000 | 2013.0 |
| winPlacePerc | 4446965.0 | 0.472822 | 0.307405 | 0.0 | 0.2 | 0.4583 | 0.740700 | 1.0 |

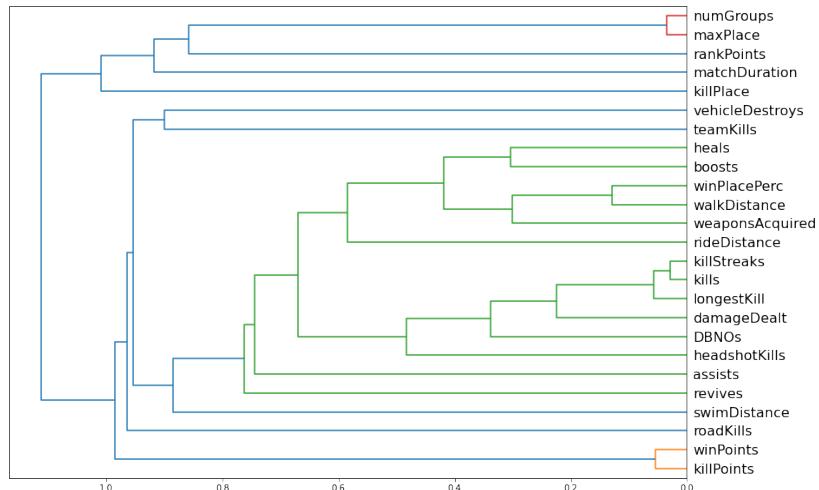


Figure 1

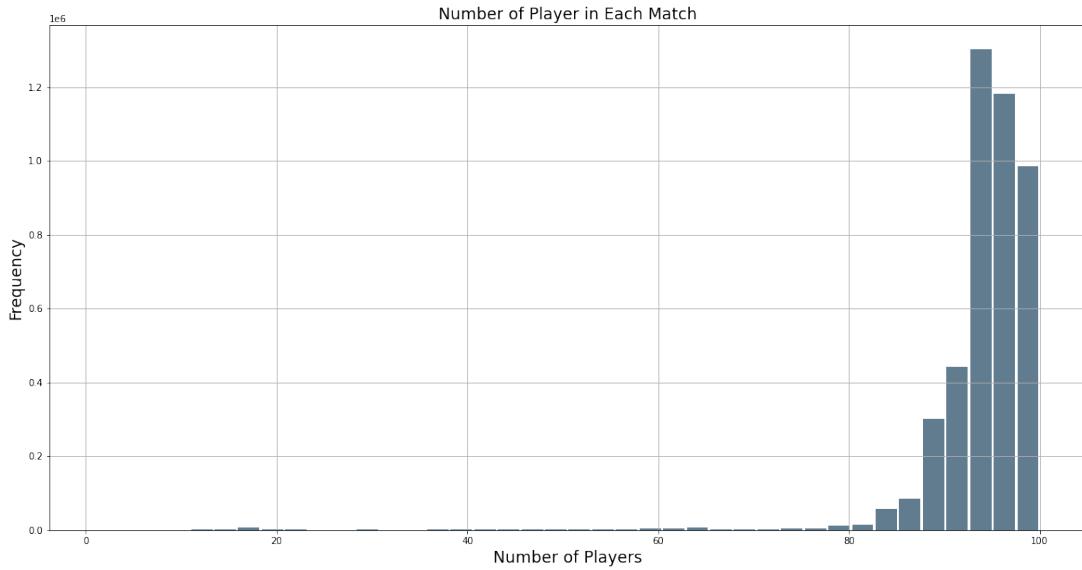


Figure 2: Histogram on the number of players in a match

| matchId | groupId | num_player | group.player | enemy.player |
|----------------|----------------|-------------------|---------------------|---------------------|
| 0000a43bce5eec | ca058240046b4c | 95 | 4 | 91 |
| 0000eb01ea6cdd | 1bedc8e1ee310b | 98 | 4 | 94 |
| 0002912fe5ed71 | 123e73e7f10a71 | 95 | 1 | 94 |
| 0003b92987589e | 0c7fbc2742917b | 100 | 2 | 98 |
| 0006eb8c17708d | ca16892a5d5285 | 93 | 2 | 91 |

Figure 3

| Id | headshotKills | kills | headshotRate |
|----------------|----------------------|--------------|---------------------|
| 58966681b2604c | 0 | 5 | 0.000000 |
| 239a776050e7f8 | 1 | 6 | 0.166667 |
| 51e37937b388fd | 1 | 2 | 0.500000 |
| 5bc301ea6d8385 | 2 | 3 | 0.666667 |
| 10ed15afaf87ec | 1 | 1 | 1.000000 |

Figure 4

| Id | walkDistance | walkSpeed | rideDistance | rideSpeed | swimDistance | swimSpeed | totalDistance |
|----------------|---------------------|------------------|---------------------|------------------|---------------------|------------------|----------------------|
| 7a5595685078c5 | 2008.0 | 1.491828 | 204.1000 | 0.151634 | 2.489 | 0.001849 | 2214.5890 |
| 02035450b9c95a | 2364.0 | 1.756315 | 0.0045 | 0.000003 | 21.030 | 0.015624 | 2385.0345 |
| 10ed15afaf87ec | 3250.0 | 2.414562 | 159.9000 | 0.118796 | 6.545 | 0.004863 | 3416.4450 |
| 9db8cb6643530a | 3335.0 | 2.477712 | 0.0060 | 0.000004 | 4.607 | 0.003423 | 3339.6130 |
| 3444098b1a9046 | 2086.0 | 1.549777 | 301.1000 | 0.223700 | 35.490 | 0.026367 | 2422.5900 |

Figure 5

| Id | heals | boosts | health_items |
|----------------|--------------|---------------|---------------------|
| 58966681b2604c | 1 | 1 | 2 |
| 239a776050e7f8 | 5 | 8 | 13 |
| 6df1e30e246e37 | 4 | 4 | 8 |
| 94d5300df5bec8 | 0 | 0 | 0 |
| f806332c66c741 | 0 | 0 | 0 |

Figure 6

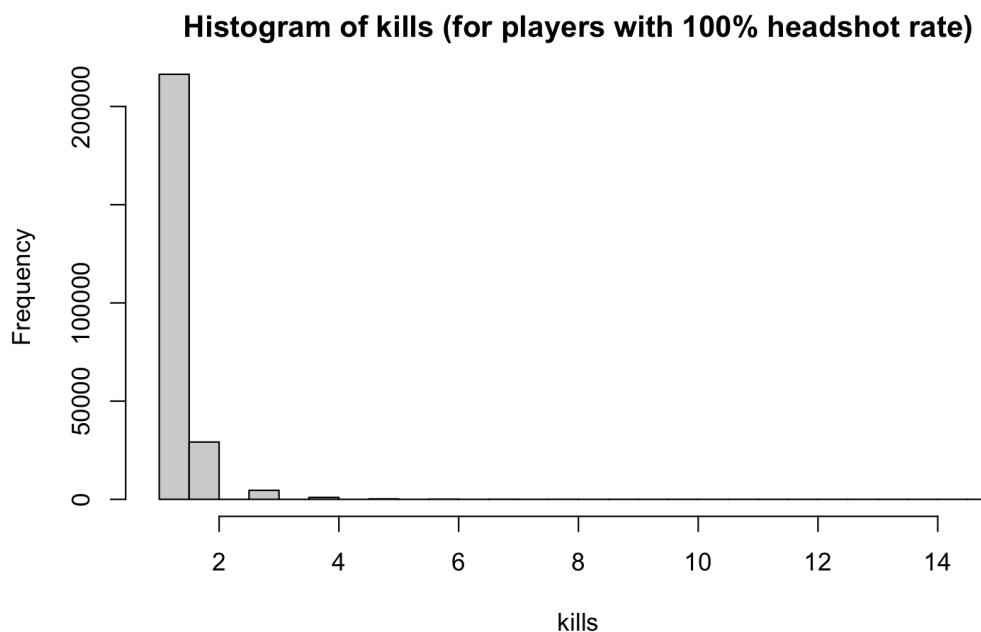


Figure 7: Histogram of kills (for players with 100% headshot rate)

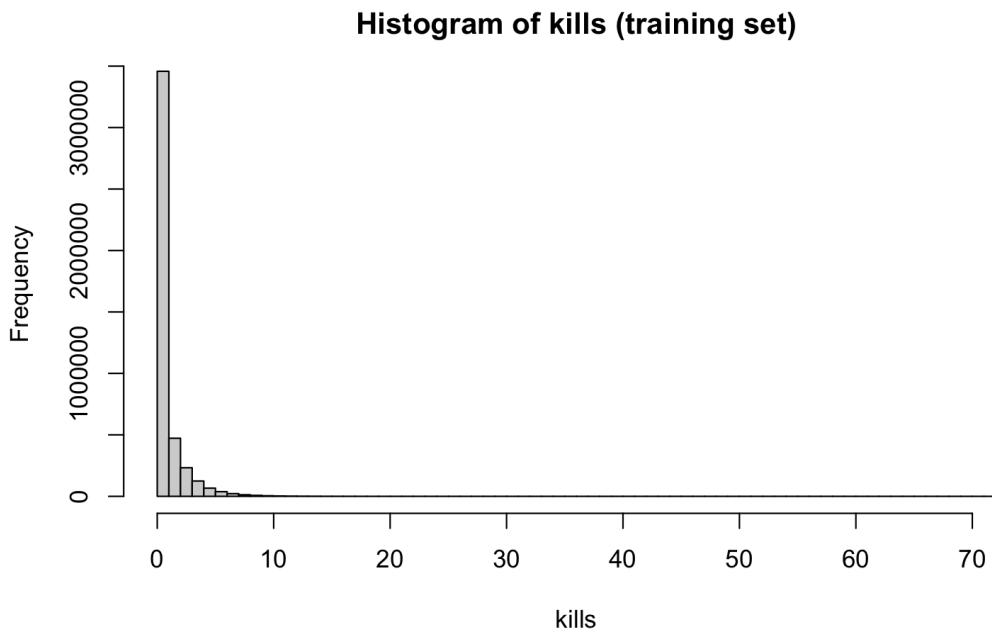


Figure 8: Histogram of kills (training set)

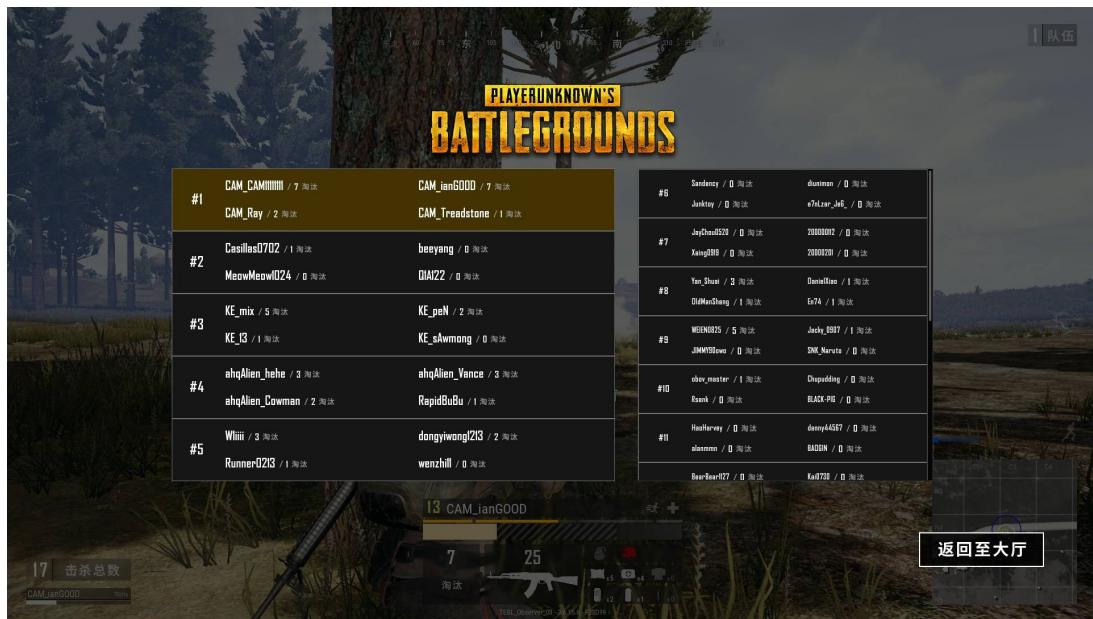


Figure 9: Players in a team will share the same rank (and WinPlacePerc)

| | 147 | 194 | 238 | 267 | 315 | 410 | 457 |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| groupId | 024136db14272c | 009396ac573a7a | 0139cbef894d40 | 01c67e8c17a39 | 03f52b23486b7 | 00ec1e82772b58 | 06495cc57e1d28 |
| group.sum.killsNorm | 0.0 | 0.0 | 4.08 | 1.02 | 1.02 | 0.0 | 2.18 |
| group.sum.killPlace | 151 | 145 | 140 | 102 | 31 | 177 | 69 |
| group.sum.damageDealtNorm | 23.4 | 0.0 | 1022.5806 | 155.04 | 102.0 | 0.0 | 211.4927 |
| group.sum.walkDistance | 563.7 | 326.3 | 9403.0 | 1132.4 | 975.0 | 118.79 | 2781.0 |
| group.sum.health_items | 0 | 6 | 23 | 4 | 6 | 0 | 2 |
| group.sum.killStreaks | 0 | 0 | 2 | 1 | 1 | 0 | 2 |
| group.sum.weaponsAcquired | 5 | 3 | 24 | 5 | 7 | 5 | 14 |
| matchId | 0003b92987589e | 0006eb8c17708d | 00077604e50a63 | 00086c74bb4efc | 00086e740a5804 | 001125344b660c | 001125344b660c |
| | 147 | 194 | 238 | 267 | 315 | 410 | |
| group.max.assists | 1 | 0 | 2 | 0 | 0 | 0 | |
| group.max.damageDealt | 23.4 | 0.0 | 534.5 | 152.0 | 100.0 | 0.0 | |
| group.max.DBNOs | 0 | 0 | 2 | 1 | 0 | 0 | |
| group.max.headshotKills | 0 | 0 | 1 | 1 | 1 | 1 | |
| group.max.killPlace | 76 | 73 | 52 | 65 | 31 | 89 | |
| group.max.killPoints | 0 | 1176 | 0 | 0 | 0 | 0 | |
| group.max.kills | 0 | 0 | 3 | 1 | 1 | 0 | |
| group.max.killStreaks | 0 | 0 | 1 | 1 | 1 | 0 | |
| group.max.longestKill | 0.0 | 0.0 | 148.0 | 17.99 | 5.269 | 0.0 | |
| group.max.matchDuration | 1410 | 1382 | 1452 | 1465 | 1962 | 1421 | |
| group.max.maxPlace | 50 | 47 | 30 | 49 | 98 | 50 | |
| group.max.rankPoints | 1449 | -1 | 1513 | 1507 | 1528 | 1469 | |
| group.max.revives | 0 | 1 | 2 | 1 | 0 | 0 | |
| group.max.rideDistance | 0.0 | 0.0 | 890.2 | 0.0 | 726.8 | 0.0 | |
| group.max.roadKills | 0 | 0 | 0 | 0 | 0 | 0 | |
| group.max.swimDistance | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| group.max.teamKills | 0 | 0 | 0 | 0 | 0 | 0 | |
| group.max.vehicleDestroys | 0 | 0 | 0 | 0 | 0 | 0 | |
| group.max.walkDistance | 297.9 | 193.7 | 2736.0 | 628.7 | 975.0 | 79.72 | |
| group.max.weaponsAcquired | 4 | 2 | 9 | 3 | 7 | 3 | |
| group.max.winPoints | 0 | 1547 | 0 | 0 | 0 | 0 | |
| group.max.headshotRate | 0.0 | 0.0 | 0.333333 | 1.0 | 1.0 | 0.0 | |
| group.max.totalDistance | 297.9 | 193.7 | 3062.4 | 628.7 | 1701.8 | 79.72 | |
| group.max.health_items | 0 | 5 | 16 | 2 | 6 | 0 | |
| group.max.killsNorm | 0.0 | 0.0 | 3.06 | 1.02 | 1.02 | 0.0 | |
| group.max.damageDealtNorm | 23.4 | 0.0 | 545.19 | 155.04 | 102.0 | 0.0 | |
| group.max.matchDurationNorm | 1410.0 | 1478.74 | 1481.04 | 1494.3 | 2001.24 | 1477.84 | |
| group.max.killsNormRank | 0.3 | 0.301075 | 0.918367 | 0.647959 | 0.704082 | 0.286458 | |
| group.max.damageDealtNormRank | 0.4 | 0.107527 | 0.969588 | 0.683673 | 0.602041 | 0.145833 | |
| group.max.killStreaksRank | 0.3 | 0.301075 | 0.72449 | 0.714286 | 0.780612 | 0.286458 | |
| group.max.assistsRank | 0.9 | 0.419355 | 0.933673 | 0.413265 | 0.479592 | 0.421875 | |
| group.max.killPlaceRank | 0.76 | 0.784946 | 0.530612 | 0.663265 | 0.316327 | 0.927083 | |
| group.max.walkDistanceRank | 0.41 | 0.268817 | 0.887755 | 0.510204 | 0.571429 | 0.145833 | |
| group.max.healthItemsRank | 0.275 | 0.827957 | 0.949498 | 0.642857 | 0.836735 | 0.234375 | |
| group.max.totalDistanceRank | 0.41 | 0.247312 | 0.918367 | 0.510204 | 0.571429 | 0.145833 | |
| group.max.totalDistanceRank_killsNormRank | 1.366667 | 0.821429 | 3.396226 | 1.773585 | 0.811594 | 0.509091 | |
| group.max.killPlace/MaxPlaceNorm | 1.52 | 1.451581 | 1.699346 | 1.30052 | 0.310124 | 1.711538 | |
| group.max.killsNorm/totalDistance | 0.0 | 0.0 | 0.001099 | 0.001626 | 0.000599 | 0.0 | |
| matchId | 0003b92987589e | 0006eb8c17708d | 00077604e50a63 | 00086c74bb4efc | 00086e740a5804 | 001125344b660c | 001125344b660c |
| | 147 | 194 | 238 | 267 | 315 | 410 | |
| group.min.assists | 0 | 0 | 0 | 0 | 0 | 0 | |
| group.min.damageDealt | 0.0 | 0.0 | 79.33 | 0.0 | 100.0 | 0.0 | |
| group.min.DBNOs | 0 | 0 | 0 | 0 | 0 | 0 | |
| group.min.headshotKills | 0 | 0 | 0 | 0 | 1 | 0 | |
| group.min.killPlace | 75 | 72 | 8 | 37 | 31 | 88 | |
| group.min.killPoints | 0 | 1169 | 0 | 0 | 0 | 0 | |
| group.min.kills | 0 | 0 | 0 | 0 | 1 | 0 | |
| group.min.killStreaks | 0 | 0 | 0 | 0 | 1 | 0 | |
| group.min.longestKill | 0.0 | 0.0 | 0.0 | 0.0 | 5.269 | 0.0 | |
| group.min.matchDuration | 1410 | 1382 | 1452 | 1465 | 1962 | 1421 | |
| group.min.maxPlace | 50 | 47 | 30 | 49 | 98 | 50 | |
| group.min.rankPoints | 1445 | -1 | 1474 | 1506 | 1528 | 1462 | |
| group.min.revives | 0 | 0 | 0 | 0 | 0 | 0 | |
| group.min.rideDistance | 0.0 | 0.0 | 0.0 | 0.0 | 726.8 | 0.0 | |
| group.min.roadKills | 0 | 0 | 0 | 0 | 0 | 0 | |
| group.min.swimDistance | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| group.min.teamKills | 0 | 0 | 0 | 0 | 0 | 0 | |
| group.min.vehicleDestroys | 0 | 0 | 0 | 0 | 0 | 0 | |
| group.min.walkDistance | 265.8 | 132.6 | 1895.0 | 503.7 | 975.0 | 39.07 | |
| group.min.weaponsAcquired | 1 | 1 | 2 | 2 | 7 | 2 | |
| group.min.winPoints | 0 | 1534 | 0 | 0 | 0 | 0 | |
| group.min.headshotRate | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | |
| group.min.totalDistance | 265.8 | 132.6 | 2466.0 | 503.7 | 1701.8 | 39.07 | |
| group.min.health_items | 0 | 1 | 0 | 2 | 6 | 0 | |
| group.min.killsNorm | 0.0 | 0.0 | 0.0 | 0.0 | 1.02 | 0.0 | |
| group.min.damageDealtNorm | 0.0 | 0.0 | 80.9166 | 0.0 | 102.0 | 0.0 | |
| group.min.matchDurationNorm | 1410.0 | 1478.74 | 1481.04 | 1494.3 | 2001.24 | 1477.84 | |
| group.min.killsNormRank | 0.3 | 0.301075 | 0.270408 | 0.270408 | 0.0704082 | 0.286458 | |
| group.min.damageDealtNormRank | 0.18 | 0.107527 | 0.438776 | 0.153061 | 0.602041 | 0.145833 | |
| group.min.killStreaksRank | 0.3 | 0.301075 | 0.270408 | 0.270408 | 0.780612 | 0.286458 | |
| group.min.assistsRank | 0.43 | 0.419355 | 0.357143 | 0.413265 | 0.479592 | 0.421875 | |
| group.min.killPlaceRank | 0.75 | 0.774194 | 0.081633 | 0.377551 | 0.316327 | 0.916667 | |
| group.min.walkDistanceRank | 0.37 | 0.193548 | 0.683265 | 0.479592 | 0.571429 | 0.114583 | |
| group.min.health_itemsRank | 0.275 | 0.532258 | 0.265306 | 0.642857 | 0.836735 | 0.234375 | |
| group.min.totalDistanceRank | 0.37 | 0.193548 | 0.744898 | 0.479592 | 0.571429 | 0.114583 | |
| group.min.totalDistanceRank_killsNormRank | 1.233333 | 0.642857 | 0.944444 | 0.787402 | 0.811594 | 0.4 | |
| group.min.killPlace/MaxPlaceNorm | 1.5 | 1.431696 | 0.261438 | 0.740296 | 0.310124 | 1.692308 | |
| group.min.killsNorm/totalDistance | 0.0 | 0.0 | 0.0 | 0.0 | 0.000599 | 0.0 | |
| matchId | 0003b92987589e | 0006eb8c17708d | 00077604e50a63 | 00086c74bb4efc | 00086e740a5804 | 001125344b660c | 001125344b660c |

Figure 10

| | 147 | 194 | 238 | 267 | 315 | 410 | 457 |
|-----------------------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| match.sum.killsNorm | 95.0 | 90.95 | 93.84 | 96.9 | 85.68 | 91.52 | 93.74 |
| match.sum.killPlace | 5050 | 4371 | 4851 | 4851 | 4851 | 4656 | 4186 |
| match.sum.damageDealtNorm | 11236.55 | 13415.32081 | 14420.8722 | 12793.1562 | 12469.5816 | 12860.2344 | 11553.3242 |
| match.sum.walkDistance | 97311.68 | 89170.18 | 115850.6702 | 97671.879 | 101622.131 | 82992.467 | 77637.077 |
| match.sum.health_items | 161 | 243 | 223 | 247 | 250 | 246 | 89 |
| match.sum.killStreaks | 61 | 52 | 56 | 57 | 46 | 57 | 52 |
| match.sum.weaponsAcquired | 323 | 315 | 355 | 331 | 360 | 326 | 271 |
| matchId | 0003b92987589e | 0006eb8c17708d | 00077604e50a63 | 00086c74bb4efc | 00086e740a5804 | 001125344b660c | 001360264d4b5f |
| | 147 | 194 | 238 | 267 | 315 | 410 | 457 |
| match.max.killsNorm | 6.0 | 8.56 | 7.14 | 12.24 | 11.22 | 7.28 | 8.72 |
| match.max.killPlace | 100 | 93 | 98 | 98 | 98 | 96 | 91 |
| match.max.damageDealtNorm | 628.5 | 746.753 | 819.366 | 1176.06 | 1292.34 | 892.944 | 784.691 |
| match.max.walkDistance | 3473.0 | 3510.0 | 3901.0 | 5010.0 | 5357.0 | 2810.0 | 3265.0 |
| match.max.health_items | 9 | 20 | 16 | 19 | 19 | 15 | 15 |
| match.max.killStreaks | 2 | 3 | 3 | 2 | 4 | 2 | 2 |
| match.max.weaponsAcquired | 13 | 10 | 13 | 11 | 9 | 8 | 9 |
| matchId | 0003b92987589e | 0006eb8c17708d | 00077604e50a63 | 00086c74bb4efc | 00086e740a5804 | 001125344b660c | 001360264d4b5f |
| | 147 | 194 | 238 | 267 | 315 | 410 | 457 |
| match.mean.killsNorm | 0.95 | 0.977957 | 0.957551 | 0.988776 | 0.874286 | 0.953333 | 1.03011 |
| match.mean.killPlace | 50.5 | 47.0 | 49.5 | 49.5 | 49.5 | 48.5 | 46.0 |
| match.mean.damageDealtNorm | 112.3655 | 144.250761 | 147.151757 | 130.54241 | 127.240629 | 133.960775 | 126.959607 |
| match.mean.walkDistance | 973.1168 | 958.81914 | 1182.149696 | 996.651827 | 1036.96052 | 864.504865 | 853.154692 |
| match.mean.health_items | 1.61 | 2.612903 | 2.27551 | 2.520408 | 2.55102 | 2.5625 | 0.978022 |
| match.mean.killStreaks | 0.61 | 0.55914 | 0.571429 | 0.581633 | 0.469388 | 0.59375 | 0.571429 |
| match.mean.weaponsAcquired | 3.23 | 3.387097 | 3.622449 | 3.377551 | 3.673469 | 3.395833 | 2.978022 |
| matchId | 0003b92987589e | 0006eb8c17708d | 00077604e50a63 | 00086c74bb4efc | 00086e740a5804 | 001125344b660c | 001360264d4b5f |

Figure 11

| | 147 | 194 | 238 | 267 | 315 | 410 |
|--|----------------|----------------|----------------|----------------|----------------|----------------|
| perc.gsum/msum.killsNorm | 0.0 | 0.0 | 0.043478 | 0.010526 | 0.011905 | 0.0 |
| perc.gmax/mmean.killsNorm | 0.0 | 0.0 | 3.195652 | 1.031579 | 1.166667 | 0.0 |
| perc.gmax/msum.killsNorm | 0.0 | 0.0 | 0.032609 | 0.010526 | 0.011905 | 0.0 |
| perc.gmax/mmax.killsNorm | 0.0 | 0.0 | 0.428571 | 0.083333 | 0.090909 | 0.0 |
| perc.gsum/msum.killPlace | 0.029901 | 0.033173 | 0.02886 | 0.021027 | 0.00639 | 0.038015 |
| perc.gmax/mmean.killPlace | 1.50495 | 1.553191 | 1.050505 | 1.313131 | 0.626263 | 1.835052 |
| perc.gmax/msum.killPlace | 0.01505 | 0.016701 | 0.010719 | 0.013399 | 0.00639 | 0.019115 |
| perc.gmax/mmax.killPlace | 0.76 | 0.784946 | 0.530612 | 0.663265 | 0.316327 | 0.927083 |
| perc.gsum/msum.damageDealtNorm | 0.002082 | 0.0 | 0.07091 | 0.012119 | 0.00818 | 0.0 |
| perc.gmax/mmean.damageDealtNorm | 0.208249 | 0.0 | 3.704951 | 1.18766 | 0.801631 | 0.0 |
| perc.gmax/msum.damageDealtNorm | 0.002082 | 0.0 | 0.037806 | 0.012119 | 0.00818 | 0.0 |
| perc.gmax/mmax.damageDealtNorm | 0.037232 | 0.0 | 0.66538 | 0.13183 | 0.078927 | 0.0 |
| perc.gsum/msum.walkDistance | 0.005793 | 0.003659 | 0.081165 | 0.011594 | 0.009594 | 0.001431 |
| perc.gmax/mmean.walkDistance | 0.30613 | 0.202019 | 2.314428 | 0.630812 | 0.940248 | 0.092215 |
| perc.gmax/msum.walkDistance | 0.003061 | 0.002172 | 0.023617 | 0.006437 | 0.009594 | 0.000961 |
| perc.gmax/mmax.walkDistance | 0.085776 | 0.055185 | 0.701359 | 0.125489 | 0.182005 | 0.02837 |
| perc.gsum/msum.health_items | 0.0 | 0.024691 | 0.103139 | 0.016194 | 0.024 | 0.0 |
| perc.gmax/mmean.health_items | 0.0 | 1.91358 | 7.03139 | 0.793522 | 2.352 | 0.0 |
| perc.gmax/msum.health_items | 0.0 | 0.020576 | 0.071749 | 0.008097 | 0.024 | 0.0 |
| perc.gmax/mmax.health_items | 0.0 | 0.25 | 1.0 | 0.105263 | 0.315789 | 0.0 |
| perc.gsum/msum.killStreaks | 0.0 | 0.0 | 0.035714 | 0.017544 | 0.021739 | 0.0 |
| perc.gmax/mmean.killStreaks | 0.0 | 0.0 | 1.75 | 1.719298 | 2.130435 | 0.0 |
| perc.gmax/msum.killStreaks | 0.0 | 0.0 | 0.017857 | 0.017544 | 0.021739 | 0.0 |
| perc.gmax/mmax.killStreaks | 0.0 | 0.0 | 0.333333 | 0.5 | 0.25 | 0.0 |
| perc.gsum/msum.weaponsAcquired | 0.01548 | 0.009524 | 0.067606 | 0.015106 | 0.019444 | 0.015337 |
| perc.gmax/mmean.weaponsAcquired | 1.23839 | 0.590476 | 2.484507 | 0.888218 | 1.905556 | 0.883436 |
| perc.gmax/msum.weaponsAcquired | 0.012384 | 0.006349 | 0.025352 | 0.009063 | 0.019444 | 0.009202 |
| perc.gmax/mmax.weaponsAcquired | 0.307692 | 0.2 | 0.692308 | 0.272727 | 0.777778 | 0.375 |
| matchId | 0003b92987589e | 0006eb8c17708d | 00077604e50a63 | 00086c74bb4efc | 00086e740a5804 | 001125344b660c |

Figure 12

| killsNormRank | damageDealtNormRank | killStreaksRank | assistsRank | killPlaceRank | walkDistanceRank | health_itemsRank | totalDistanceRank | 1 |
|----------------------|----------------------------|------------------------|--------------------|----------------------|-------------------------|-------------------------|--------------------------|----------|
| 0.302083 | 0.906250 | 0.302083 | 0.385417 | 0.833333 | 0.260417 | 0.510417 | 0.260417 | |
| 0.708333 | 0.708333 | 0.755208 | 0.385417 | 0.364583 | 0.208333 | 0.510417 | 0.208333 | |
| 0.708333 | 0.885417 | 0.755208 | 0.385417 | 0.375000 | 0.322917 | 0.229167 | 0.322917 | |
| 0.652174 | 0.858696 | 0.711957 | 0.396739 | 0.434783 | 0.347826 | 0.315217 | 0.347826 | |
| 0.280220 | 0.175824 | 0.280220 | 0.467033 | 0.791209 | 0.274725 | 0.296703 | 0.274725 | |
| 0.638298 | 0.351064 | 0.707447 | 0.409574 | 0.265957 | 0.617021 | 0.808511 | 0.638298 | |
| 0.845745 | 0.734043 | 0.707447 | 0.904255 | 0.095745 | 0.776596 | 0.893617 | 0.744681 | |
| 0.941489 | 0.978723 | 0.707447 | 0.904255 | 0.053191 | 0.702128 | 0.750000 | 0.702128 | |
| 0.678161 | 0.436782 | 0.747126 | 0.408046 | 0.390805 | 0.149425 | 0.224138 | 0.149425 | |
| 0.310345 | 0.321839 | 0.310345 | 0.896552 | 0.862069 | 0.103448 | 0.224138 | 0.103448 | |

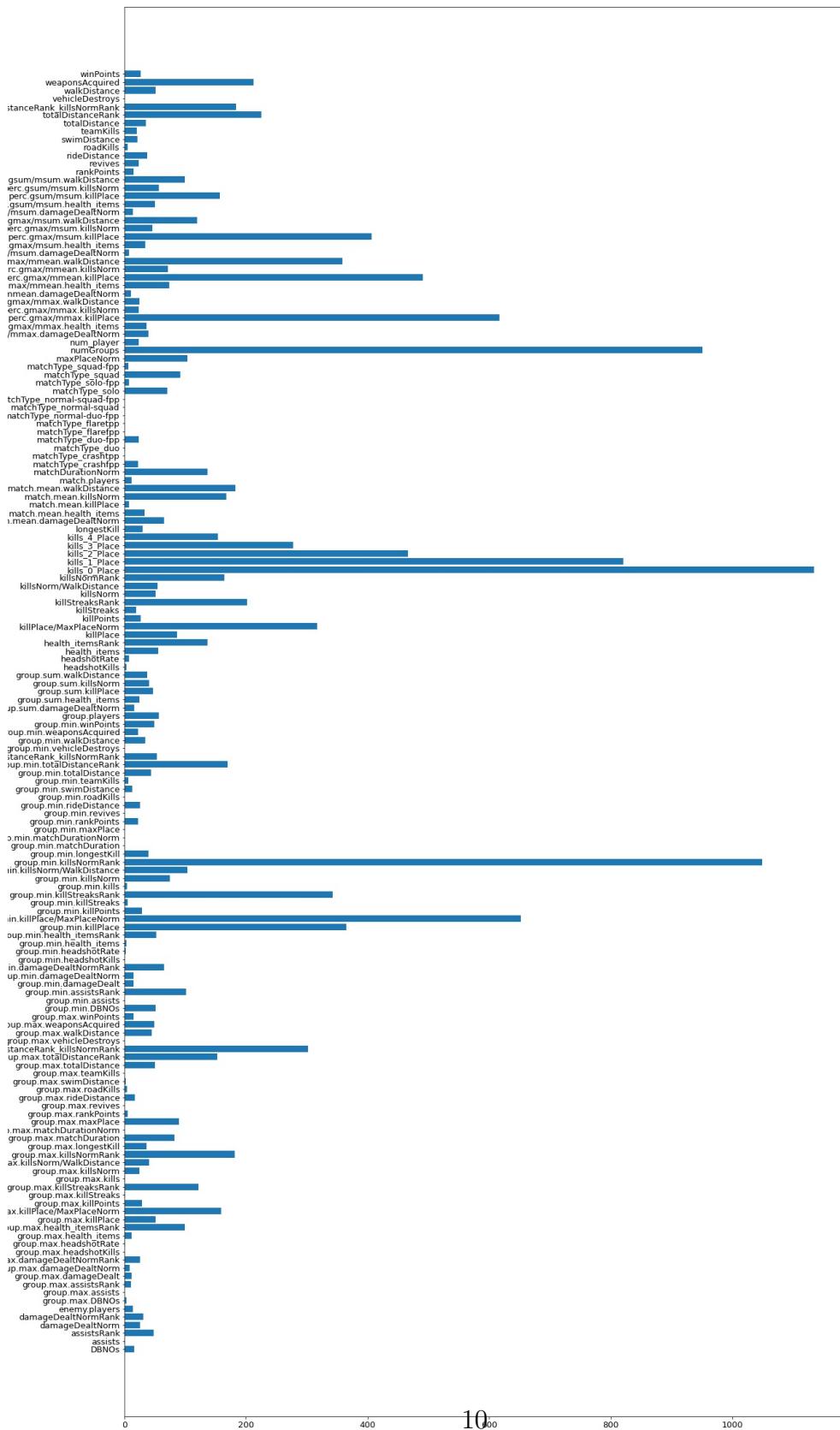
Figure 13

| | 100 | 101 | 102 | 103 | 104 | 105 |
|----------------------|----------------|----------------|----------------|----------------|----------------|----------------|
| kills_0_Place | 42.0 | 0.0 | 40.0 | 1.0 | 23.0 | 0.0 |
| kills_1_Place | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| kills_2_Place | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 | 7.0 |
| kills_3_Place | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| matchId | 0002912fe5ed71 | 0002912fe5ed71 | 0002912fe5ed71 | 0002912fe5ed71 | 0002912fe5ed71 | 0002912fe5ed71 |
| groupId | 7604543b06f72e | 7e08b49bea8b79 | 8038ec6e779bcc | 805d30e791ebba | 84d84021dbf624 | 8b95bccff863bd |

Figure 14

| | 147 | 194 | 238 | 267 | 315 | 410 |
|-----------------------------------|----------------|----------------|----------------|----------------|----------------|----------------|
| matchType_crashfpp | 0 | 0 | 0 | 0 | 0 | 0 |
| matchType_crashtp | 0 | 0 | 0 | 0 | 0 | 0 |
| matchType_duo | 1 | 0 | 0 | 0 | 0 | 1 |
| matchType_duo-fpp | 0 | 1 | 0 | 1 | 0 | 0 |
| matchType_flarefpp | 0 | 0 | 0 | 0 | 0 | 0 |
| matchType_flaretpp | 0 | 0 | 0 | 0 | 0 | 0 |
| matchType_normal-duo-fpp | 0 | 0 | 0 | 0 | 0 | 0 |
| matchType_normal-squad | 0 | 0 | 0 | 0 | 0 | 0 |
| matchType_normal-squad-fpp | 0 | 0 | 0 | 0 | 0 | 0 |
| matchType_solo | 0 | 0 | 0 | 0 | 0 | 0 |
| matchType_solo-fpp | 0 | 0 | 0 | 0 | 1 | 0 |
| matchType_squad | 0 | 0 | 0 | 0 | 0 | 0 |
| matchType_squad-fpp | 0 | 0 | 1 | 0 | 0 | 0 |
| matchId | 0003b92987589e | 0006eb8c17708d | 00077604e50a63 | 00086c74bb4efc | 00086e740a5804 | 001125344b660c |

Figure 15



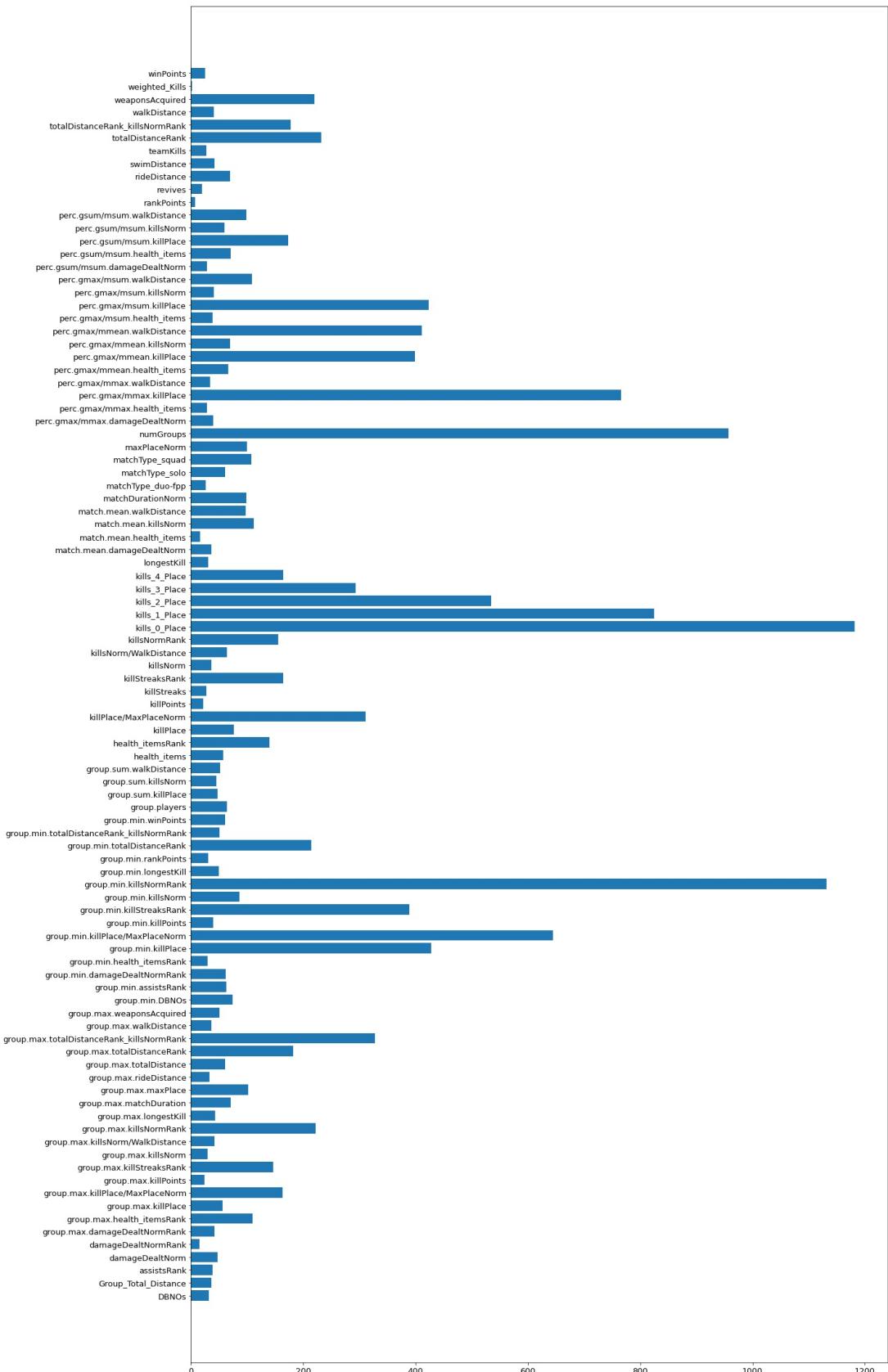


Figure 17

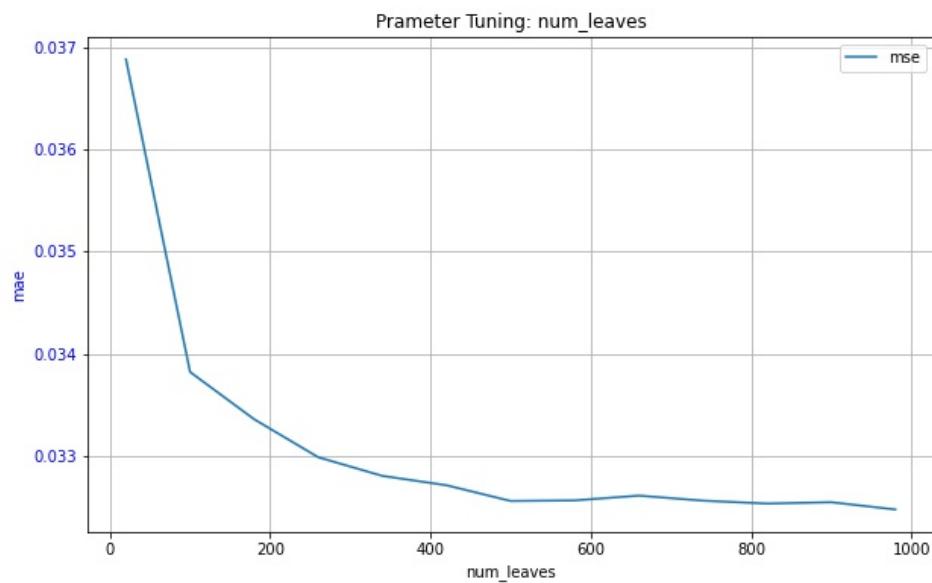


Figure 18: MSE vs. num.leaves

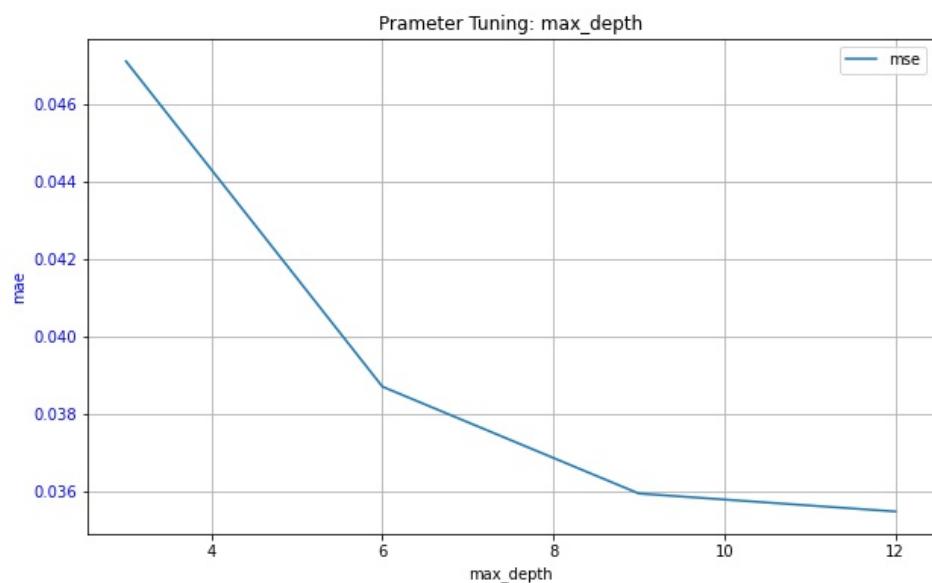


Figure 19: MSE vs. max_depth

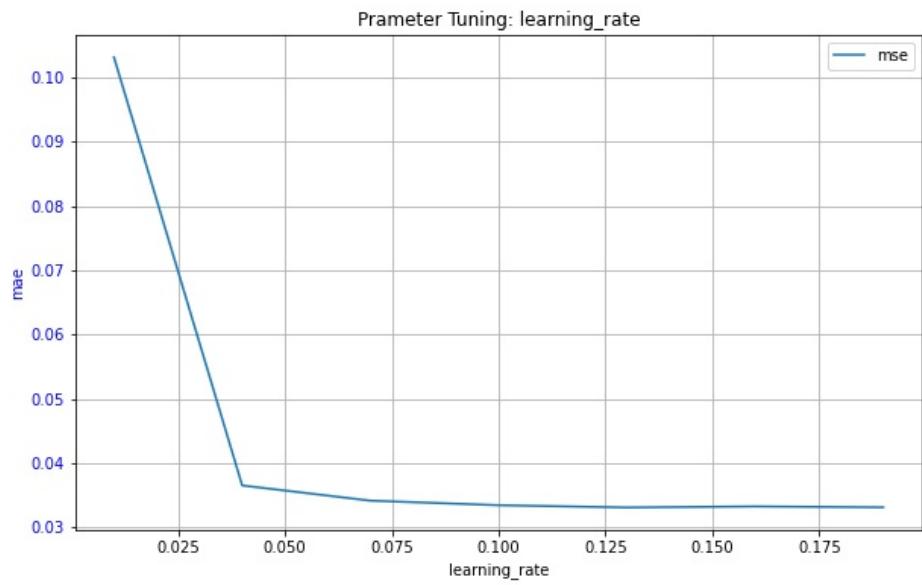


Figure 20: MSE vs. learning_rate

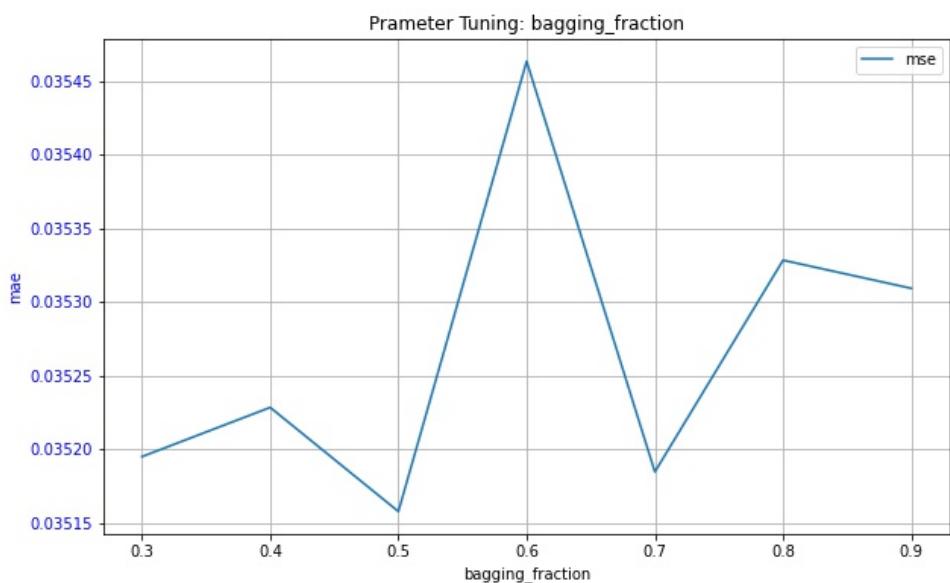


Figure 21: MSE vs. bagging_fraction

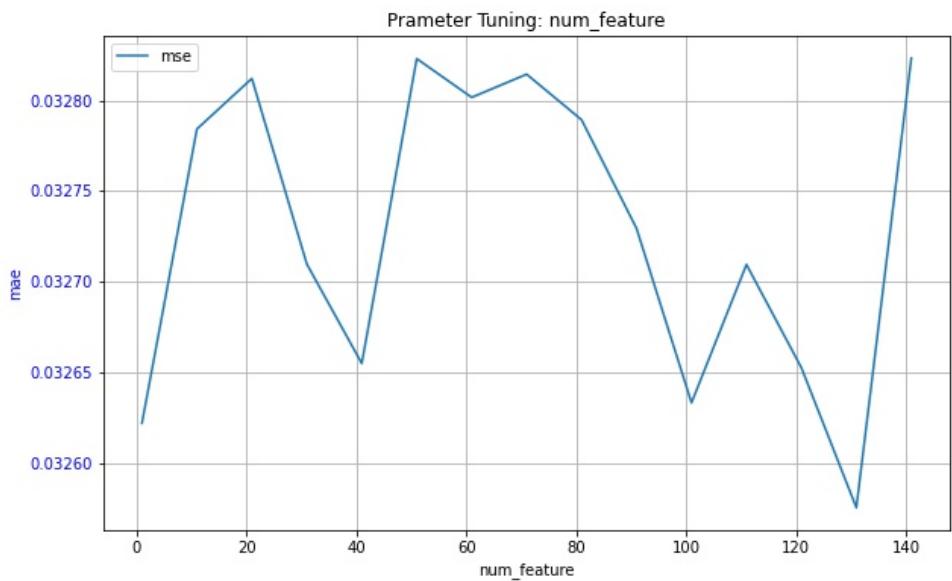


Figure 22: MSE vs. num_feature

| matchId | groupId | kills | killPlace | winPlacePerc |
|----------------|----------------|-------|-----------|--------------|
| 0002912fe5ed71 | 74a5c6ad42f065 | 0 | 50 | 0.7553 |
| 0002912fe5ed71 | 34e5e1dce2df52 | 0 | 51 | 0.7128 |
| 0002912fe5ed71 | 8f59e08bf77ed2 | 0 | 52 | 0.6702 |
| 0002912fe5ed71 | 507a615b1ea428 | 0 | 53 | 0.6489 |
| 0002912fe5ed71 | e9e67e9edaaa47 | 0 | 54 | 0.6277 |
| 0002912fe5ed71 | a3162639af4f67 | 0 | 55 | 0.5638 |
| 0002912fe5ed71 | d56624d75b9b8f | 0 | 56 | 0.5532 |
| 0002912fe5ed71 | 45420f8aa2e4e9 | 0 | 57 | 0.5426 |
| 0002912fe5ed71 | 38b2725fe16840 | 0 | 58 | 0.5213 |
| 0002912fe5ed71 | 0acd4dd6620a6c | 0 | 59 | 0.5106 |

Figure 23: Result

| | | | Overview | Data | Code | Discussion | Leaderboard | Rules | Team | My Submissions | Late Submission | ... |
|-----|---|--|----------|------|------|------------|--------------------------------|------------------------|------|---|---------------------------------|-----|
| 583 | — | | | | | | Andrey Redekop | | |  0.03093 | 13 | 3Y |
| 584 | — | | | | | | Mirodil | | |  0.03116 | 6 | 3Y |
| 585 | — | | | | | | Quyen. PT | | |  0.03147 | 1 | 3Y |
| 586 | — | | | | | | Bob3 | | |  0.03170 | 4 | 3Y |
| 587 | — | | | | | | Necromancer | | |  0.03176 | 3 | 3Y |
| 588 | — | | | | | | Ales | </> pubg_placement_... | |  0.03187 | 5 | 3Y |

Figure 24: Leaderboard on Kaggle

```
#all_data['_totalDistance'] = all_data['rideDistance'] + all_data['walkDistance'] + all_data['swimDistance']
#all_data['_headshotKillRate'] = all_data['headshotKills'] / all_data['kills']
```

In [30]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import time
from scipy.stats.mstats import winsorize
import warnings
warnings.filterwarnings("ignore")
import seaborn as sns
from IPython.display import clear_output
pd.set_option('display.max_rows', 300)
from sklearn.model_selection import train_test_split
import lightgbm as lgb
from sklearn.metrics import mean_absolute_error
from sklearn.impute import SimpleImputer
Imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
import fclib
from fclib.models.lightgbm import predict
from fclib.evaluation.evaluation_utils import MAPE
```

In [18]:

```
import numpy as np
import pandas as pd
#Import the dataset after cheater detection
train = pd.read_csv('train_v3.csv')
print(train.shape)
```

(4389111, 35)

In [19]:

```
#dataset added features: num_player headshotRate totalDistance walkSpeed rideSpe
all_data = train.copy()
all_data.drop(columns=['walkSpeed', 'rideSpeed', 'swimSpeed'], inplace=True)
print(all_data.shape)
```

(4389111, 32)

In [20]:

```
#Combine features
all_data['health_items'] = all_data['heals'] + all_data['boosts']
all_data.drop(['boosts', 'heals'], axis=1, inplace=True)
#Normalize features
#remember to change the name of variables
all_data['killsNorm'] = all_data['kills']*((100-all_data['num_player'])/100 + 1)
all_data['damageDealtNorm'] = all_data['damageDealt']*((100-all_data['num_player'])
all_data['maxPlaceNorm'] = all_data['maxPlace']*((100-all_data['num_player'])/10
all_data['matchDurationNorm'] = all_data['matchDuration']*((100-all_data['num_pl
#all_data.drop(['kills', 'damageDealt', 'maxPlace', 'matchDuration'], axis=1, inpla
```

In [21]:

```
#create match data 每个人在每场比赛里这些变量的排名percent
match = all_data.groupby('matchId')
all_data['killsNormRank'] = match['killsNorm'].rank(pct=True).values
all_data['damageDealtNormRank'] = match['damageDealtNorm'].rank(pct=True).values
all_data['killStreaksRank'] = match['killStreaks'].rank(pct=True).values
all_data['assistsRank'] = match['assists'].rank(pct=True).values
```

```
#all_data['killPlaceRank'] = match['killPlace'].rank(pct=True).values
#all_data['walkDistanceRank'] = match['walkDistance'].rank(pct=True).values
all_data['health_itemsRank'] = match['health_items'].rank(pct=True).values
all_data['totalDistanceRank'] = match['totalDistance'].rank(pct=True).values
all_data['totalDistanceRank_killsNormRank'] = all_data['totalDistanceRank'] / all_data['killsNormRank']
all_data.sort_values(['groupId']).head(10)
```

Out[21]:

| | matchId | Id | groupId | assists | damageDealt | DBNOs | he |
|---------|----------------|----------------|----------------|---------|-------------|-------|----|
| 1749413 | 660d439a723670 | 728ff47d62429d | 00000c08b5be36 | 0 | 298.00 | 2 | |
| 1749470 | 660d439a723670 | 7c0e726bddf8aa | 00000c08b5be36 | 0 | 173.50 | 1 | |
| 1749408 | 660d439a723670 | af7d162ddb751c | 00000c08b5be36 | 0 | 270.00 | 2 | |
| 937349 | 370b420efc87f4 | 6c24133622d331 | 00000d1cbbc340 | 0 | 173.70 | 0 | |
| 2134194 | 7c86ac34f9ea9c | 1312316425a52d | 000025a09dd1d7 | 0 | 0.00 | 0 | |
| 2049633 | 77a20700ee0c75 | 753f516a6e4a5e | 000038ec4dff53 | 0 | 23.57 | 0 | |
| 2049621 | 77a20700ee0c75 | a40956f7d1c4f8 | 000038ec4dff53 | 1 | 188.00 | 1 | |
| 2049614 | 77a20700ee0c75 | 045c4470d9c745 | 000038ec4dff53 | 1 | 579.10 | 4 | |
| 1642383 | 5ff11bb177a286 | aee607add206b5 | 00003a54230763 | 0 | 65.70 | 1 | |
| 1642354 | 5ff11bb177a286 | 15694464538873 | 00003a54230763 | 1 | 34.30 | 0 | |

10 rows × 42 columns

In [22]:

```
def fillInf(df, val):
    numcols = df.select_dtypes(include='number').columns
    cols = numcols[numcols != 'winPlacePerc']
    df[df == np.Inf] = np.NaN
    df[df == np.NINF] = np.NaN
    for c in cols: df[c].fillna(val, inplace=True)

all_data['killPlace/MaxPlaceNorm'] = all_data['killPlace'] / all_data['maxPlaceNorm']
all_data['killsNorm/WalkDistance'] = all_data['killsNorm'] / all_data['walkDistance']
fillInf(all_data, 0)
all_data.head(3)
```

Out[22]:

| | matchId | Id | groupId | assists | damageDealt | DBNOs | headshotK |
|---|----------------|----------------|----------------|---------|-------------|-------|-----------|
| 0 | 0000a43bce5eec | 58966681b2604c | ca058240046b4c | 1 | 375.6 | 4 | |
| 1 | 0000a43bce5eec | f806332c66c741 | 909fe185a8cdec | 0 | 49.5 | 0 | |
| 2 | 0000a43bce5eec | fd1b8609a83fc5 | f21c5900e6a0cd | 1 | 189.9 | 0 | |

3 rows × 44 columns

```
#all_data.drop(['boosts','heals','killStreaks','DBNOs'], axis=1, inplace=True)
#all_data.drop(['headshotKills','roadKills','vehicleDestroys'], axis=1, inplace=True)
#all_data.drop(['rideDistance','swimDistance','matchDuration'], axis=1, inplace=True)
#all_data.drop(['rankPoints','killPoints','winPoints'], axis=1, inplace=True)
```

In [23]:

```
match = all_data.groupby(['matchId'])
group = all_data.groupby(['matchId', 'groupId', 'matchType'])
```

```

# max, min features
max_min_col = list(all_data.columns)
exclude = ['Id', 'matchId', 'groupId', 'matchType', 'maxPlaceNorm', 'numGroups', 'winP
for i in exclude:
    max_min_col.remove(i)
print(max_min_col)

# sum features
sum_col = ['killsNorm', 'killPlace', 'damageDealtNorm', 'walkDistance', 'health_item

['assists', 'damageDealt', 'DBNOs', 'headshotKills', 'killPlace', 'killPoints',
'kills', 'killStreaks', 'longestKill', 'matchDuration', 'maxPlace', 'rankPoint
s', 'revives', 'rideDistance', 'roadKills', 'swimDistance', 'teamKills', 'vehicl
eDestroys', 'walkDistance', 'weaponsAcquired', 'winPoints', 'headshotRate', 'tot
alDistance', 'health_items', 'killsNorm', 'damageDealtNorm', 'matchDurationNor
m', 'killsNormRank', 'damageDealtNormRank', 'killStreaksRank', 'assistsRank', 'h
ealth_itemsRank', 'totalDistanceRank', 'totalDistanceRank_killsNormRank', 'killP
lace/MaxPlaceNorm', 'killsNorm/WalkDistance']

```

In [24]:

```

''' match sum, match max, match mean, group sum
'''

match_data = pd.concat([
    match.size().to_frame('match.players'),
    match[sum_col].sum().rename(columns=lambda s: 'match.sum.' + s),
    match[sum_col].max().rename(columns=lambda s: 'match.max.' + s),
    match[sum_col].mean().rename(columns=lambda s: 'match.mean.' + s)
], axis=1).reset_index()

match_data = pd.merge(match_data,
                     group[sum_col].sum().rename(columns=lambda s: 'group.sum.' + s).reset_index()

match_data.head(3)

```

Out[24]:

| | matchId | match.players | match.sum.killsNorm | match.sum.killPlace | match.sum.damage |
|---|----------------|---------------|---------------------|---------------------|------------------|
| 0 | 0000a43bce5eec | 95 | 94.5 | 4560 | 14 |
| 1 | 0000a43bce5eec | 95 | 94.5 | 4560 | 14 |
| 2 | 0000a43bce5eec | 95 | 94.5 | 4560 | 14 |

3 rows × 24 columns

In [25]:

```

''' ranking of kills and killPlace in each match
'''

minKills = all_data.sort_values(['matchId', 'groupId', 'kills', 'killPlace']).group
    ['matchId', 'groupId', 'kills']].first().reset_index().copy()

for n in np.arange(5):
    c = 'kills_' + str(n) + '_Place'
    nKills = (minKills['kills'] == n)
    minKills.loc[nKills, c] = minKills[minKills['kills'] == n].groupby(['matchId'])['killPlace']
    match_data = pd.merge(match_data, minKills[[nKills][['matchId', 'groupId', c]]],
    #match_data[c].fillna(0, inplace=True)

match_data.head(3)

```

Out[25]:

| | matchId | match.players | match.sum.killsNorm | match.sum.killPlace | match.sum.damage |
|---|----------------|---------------|---------------------|---------------------|------------------|
| 0 | 0000a43bce5eec | 95 | 94.5 | 4560 | 14 |
| 1 | 0000a43bce5eec | 95 | 94.5 | 4560 | 14 |
| 2 | 0000a43bce5eec | 95 | 94.5 | 4560 | 14 |

3 rows × 29 columns

In [26]:

```
... group mean, max, min
...
all_data2 = pd.concat([
    group.size().to_frame('group.players'),
    group.mean(),
    group[max_min_col].max().rename(columns=lambda s: 'group.max.' + s),
    group[max_min_col].min().rename(columns=lambda s: 'group.min.' + s),
], axis=1).reset_index()

print(all_data2.shape)
```

(2003765, 116)

In [27]:

all_data2.head(3)

Out[27]:

| | matchId | groupId | matchType | group.players | assists | damageDealt | DBNOs |
|---|----------------|----------------|-----------|---------------|---------|-------------|------------|
| 0 | 0000a43bce5eec | 18b16ec699d8b6 | squad-fpp | | 2 | 0.0 | 109.675000 |
| 1 | 0000a43bce5eec | 236ab9e9c081b9 | squad-fpp | | 6 | 0.0 | 47.988333 |
| 2 | 0000a43bce5eec | 3a6addfa0df938 | squad-fpp | | 2 | 0.0 | 0.000000 |

3 rows × 116 columns

In [28]:

```
numcols = all_data2.select_dtypes(include='number').columns.values
numcols = numcols[numcols != 'winPlacePerc']
all_data = pd.merge(all_data2, match_data)
all_data['enemy.players'] = all_data['match.players'] - all_data['group.players']

for c in sum_col:
    #all_data['enemy.' + c] = (all_data['m.sum.' + c] - all_data['sum.' + c]) /
    all_data['perc.gsum/msum.' + c] = all_data['group.sum.' + c] / all_data['mat
    all_data['perc.gmax/mmean.' + c] = all_data['group.max.' + c] / all_data['ma
    all_data['perc.gmax/msum.' + c] = all_data['group.max.' + c] / all_data['mat
    all_data['perc.gmax/mmax.' + c] = all_data['group.max.' + c] / all_data['mat
    all_data.drop(['match.sum.' + c, 'match.max.' + c], axis=1, inplace=True)

fillInf(all_data, 0)
all_data.head(3)
```

Out[28]:

| | matchId | groupId | matchType | group.players | assists | damageDealt | DBNOs |
|---|----------------|----------------|-----------|---------------|---------|-------------|------------|
| 0 | 0000a43bce5eec | 18b16ec699d8b6 | squad-fpp | | 2 | 0.0 | 109.675000 |
| 1 | 0000a43bce5eec | 236ab9e9c081b9 | squad-fpp | | 6 | 0.0 | 47.988333 |

| | matchId | groupId | matchType | group.players | assists | damageDealt | DBNOs |
|---|----------------|----------------|-----------|---------------|---------|-------------|-------------------|
| 2 | 0000a43bce5eec | 3a6addfa0df938 | squad-fpp | | 2 | 0.0 | 0.000000 0.000000 |

3 rows × 153 columns

```
# drop constant column
constant_column = [col for col in all_data.columns if all_data[col].nunique() == 1]
print('drop columns:', constant_column)
all_data.drop(constant_column, axis=1, inplace=True)
```

In [29]:

```
all_data = pd.get_dummies(all_data, columns=[ 'matchType' ])
list(all_data.columns)
```

Out[29]:

```
[ 'matchId',
  'groupId',
  'group.players',
  'assists',
  'damageDealt',
  'DBNOs',
  'headshotKills',
  'killPlace',
  'killPoints',
  'kills',
  'killStreaks',
  'longestKill',
  'matchDuration',
  'maxPlace',
  'numGroups',
  'rankPoints',
  'revives',
  'rideDistance',
  'roadKills',
  'swimDistance',
  'teamKills',
  'vehicleDestroys',
  'walkDistance',
  'weaponsAcquired',
  'winPoints',
  'winPlacePerc',
  'num_player',
  'headshotRate',
  'totalDistance',
  'health_items',
  'killsNorm',
  'damageDealtNorm',
  'maxPlaceNorm',
  'matchDurationNorm',
  'killsNormRank',
  'damageDealtNormRank',
  'killStreaksRank',
  'assistsRank',
  'health_itemsRank',
  'totalDistanceRank',
  'totalDistanceRank_killsNormRank',
  'killPlace/MaxPlaceNorm',
  'killsNorm/WalkDistance',
  'group.max.assists',
  'group.max.damageDealt',
  'group.max.DBNOs',
  'group.max.headshotKills',
```

```
'group.max.killPlace',
'group.max.killPoints',
'group.max.kills',
'group.max.killStreaks',
'group.max.longestKill',
'group.max.matchDuration',
'group.max.maxPlace',
'group.max.rankPoints',
'group.max.revives',
'group.max.rideDistance',
'group.max.roadKills',
'group.max.swimDistance',
'group.max.teamKills',
'group.max.vehicleDestroys',
'group.max.walkDistance',
'group.max.weaponsAcquired',
'group.max.winPoints',
'group.max.headshotRate',
'group.max.totalDistance',
'group.max.health_items',
'group.max.killsNorm',
'group.max.damageDealtNorm',
'group.max.matchDurationNorm',
'group.max.killsNormRank',
'group.max.damageDealtNormRank',
'group.max.killStreaksRank',
'group.max.assistsRank',
'group.max.health_itemsRank',
'group.max.totalDistanceRank',
'group.max.totalDistanceRank_killsNormRank',
'group.max.killPlace/MaxPlaceNorm',
'group.max.killsNorm/WalkDistance',
'group.min.assists',
'group.min.damageDealt',
'group.min.DBNOs',
'group.min.headshotKills',
'group.min.killPlace',
'group.min.killPoints',
'group.min.kills',
'group.min.killStreaks',
'group.min.longestKill',
'group.min.matchDuration',
'group.min.maxPlace',
'group.min.rankPoints',
'group.min.revives',
'group.min.rideDistance',
'group.min.roadKills',
'group.min.swimDistance',
'group.min.teamKills',
'group.min.vehicleDestroys',
'group.min.walkDistance',
'group.min.weaponsAcquired',
'group.min.winPoints',
'group.min.headshotRate',
'group.min.totalDistance',
'group.min.health_items',
'group.min.killsNorm',
'group.min.damageDealtNorm',
'group.min.matchDurationNorm',
'group.min.killsNormRank',
```

```
'group.min.damageDealtNormRank',
'group.min.killStreaksRank',
'group.min.assistsRank',
'group.min.health_itemsRank',
'group.min.totalDistanceRank',
'group.min.totalDistanceRank_killsNormRank',
'group.min.killPlace/MaxPlaceNorm',
'group.min.killsNorm/WalkDistance',
'match.players',
'match.mean.killsNorm',
'match.mean.killPlace',
'match.mean.damageDealtNorm',
'match.mean.walkDistance',
'match.mean.health_items',
'group.sum.killsNorm',
'group.sum.killPlace',
'group.sum.damageDealtNorm',
'group.sum.walkDistance',
'group.sum.health_items',
'kills_0_Place',
'kills_1_Place',
'kills_2_Place',
'kills_3_Place',
'kills_4_Place',
'enemy.players',
'perc.gsum/msum.killsNorm',
'perc.gmax/mmean.killsNorm',
'perc.gmax/msum.killsNorm',
'perc.gmax/mmax.killsNorm',
'perc.gsum/msum.killPlace',
'perc.gmax/mmean.killPlace',
'perc.gmax/msum.killPlace',
'perc.gmax/mmax.killPlace',
'perc.gsum/msum.damageDealtNorm',
'perc.gmax/mmean.damageDealtNorm',
'perc.gmax/msum.damageDealtNorm',
'perc.gmax/mmax.damageDealtNorm',
'perc.gsum/msum.walkDistance',
'perc.gmax/mmean.walkDistance',
'perc.gmax/msum.walkDistance',
'perc.gmax/mmax.walkDistance',
'perc.gsum/msum.health_items',
'perc.gmax/mmean.health_items',
'perc.gmax/msum.health_items',
'perc.gmax/mmax.health_items',
'matchType_crashfpp',
'matchType_crashtpp',
'matchType_duo',
'matchType_duo-fpp',
'matchType_flarefpp',
'matchType_flaretpp',
'matchType_normal-duo-fpp',
'matchType_normal-squad',
'matchType_normal-squad-fpp',
'matchType_solo',
'matchType_solo-fpp',
'matchType_squad',
'matchType_squad-fpp']
```

In [30]:

```
all_data.drop(['kills', 'damageDealt', 'maxPlace', 'matchDuration'], axis=1, inplace=True)
```

In [79]:

```
all_data=pd.merge(all_data.groupby('groupId')[['totalDistance']].sum().rename('Gro
```

In [91]:

```
all_data['weighted Kills']=all_data['killsNorm']*all_data['totalDistance']/all_d
```

In [255...]:

```
#data preparation finished
all_data.head(10)
```

Out[255...]:

| | groupId | Group_Total_Distance | matchId | group.players | assists | DBNOs |
|---|----------------|----------------------|----------------|---------------|----------|----------|
| 0 | 00000c08b5be36 | 271.533333 | 660d439a723670 | 3 | 0.000000 | 1.666667 |
| 1 | 00000d1cbcb340 | 135.800000 | 370b420efc87f4 | 1 | 0.000000 | 0.000000 |
| 2 | 000025a09dd1d7 | 110.800000 | 7c86ac34f9ea9c | 1 | 0.000000 | 0.000000 |
| 3 | 000038ec4dff53 | 2157.590000 | 77a20700ee0c75 | 3 | 0.666667 | 1.666667 |
| 4 | 00003a54230763 | 92.675000 | 5ff11bb177a286 | 2 | 0.500000 | 0.500000 |
| 5 | 000049feba5c83 | 2929.723333 | 789a6f3b30fb8b | 3 | 0.333333 | 1.666667 |
| 6 | 00006aa800b4a7 | 174.800000 | ea1e8b84979c8f | 1 | 0.000000 | 0.000000 |
| 7 | 00006eda19d504 | 61.080000 | d7cfa3f92b5c47 | 1 | 0.000000 | 0.000000 |
| 8 | 000073397f671b | 3295.336667 | 77b1eca53a5eb4 | 3 | 0.333333 | 0.333333 |
| 9 | 000077d5217a7c | 1357.544000 | 5b054e6d1adb85 | 2 | 0.000000 | 0.000000 |

10 rows × 163 columns

In []:

```
sample = all_data.shape[0]
df = all_data.drop(columns = ['winPlacePerc'])
y = all_data['winPlacePerc']
```

In []:

```
#we notice that kills_0_Place have outstanding predictability on winPlacePerc
regression(all_data[['kills_0_Place', 'winPlacePerc']])
```

OLS Regression Results

| | | | |
|--------------------------|------------------|----------------------------|------------|
| Dep. Variable: | winPlacePerc | R-squared: | 0.589 |
| Model: | OLS | Adj. R-squared: | 0.589 |
| Method: | Least Squares | F-statistic: | 2.874e+06 |
| Date: | Tue, 07 Dec 2021 | Prob (F-statistic): | 0.00 |
| Time: | 01:18:06 | Log-Likelihood: | 4.7606e+05 |
| No. Observations: | 2003765 | AIC: | -9.521e+05 |
| Df Residuals: | 2003763 | BIC: | -9.521e+05 |
| Df Model: | 1 | | |

Covariance Type: nonrobust

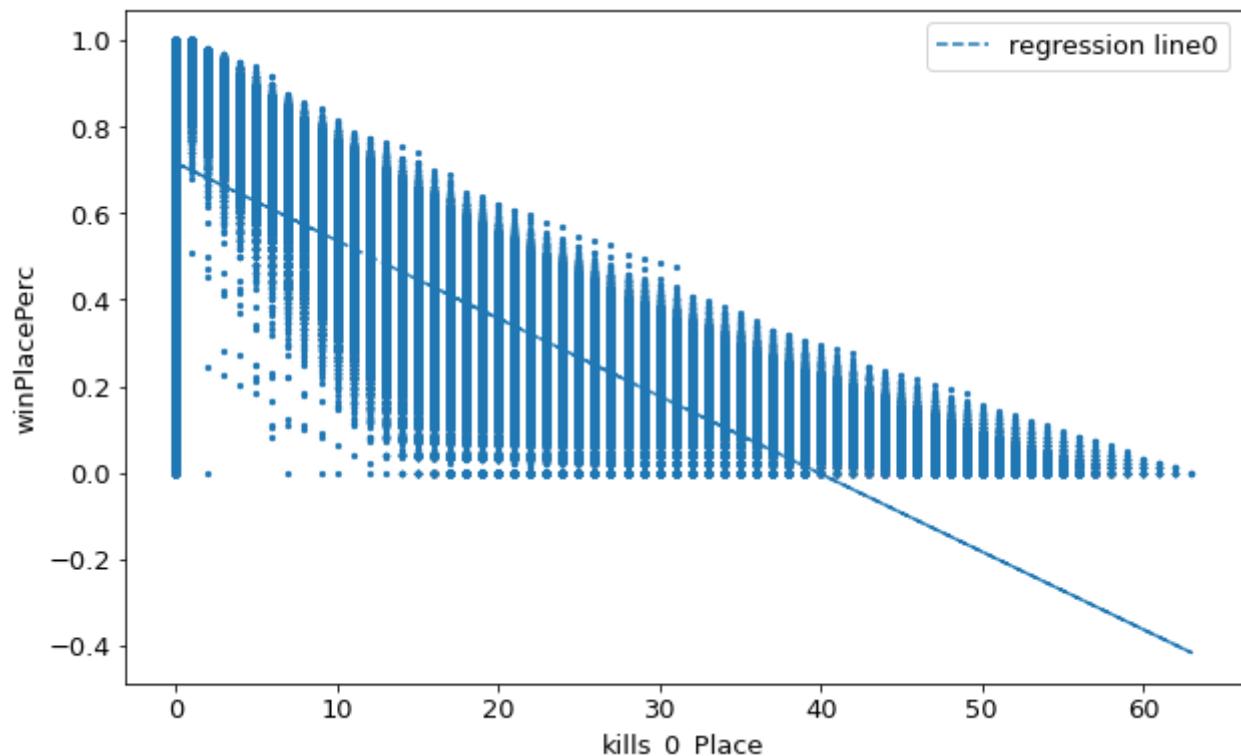
| | coef | std err | t | P> t | [0.025 | 0.975] |
|---|---------|----------|-----------------------|-------|--------|--------|
| const | 0.7169 | 0.000 | 3855.301 | 0.000 | 0.717 | 0.717 |
| kills_0_Place | -0.0180 | 1.06e-05 | -1695.150 | 0.000 | -0.018 | -0.018 |
| Omnibus: 163328.385 Durbin-Watson: 1.931 | | | | | | |
| Prob(Omnibus): 0.000 Jarque-Bera (JB): 205996.911 | | | | | | |
| Skew: -0.769 | | | Prob(JB): 0.00 | | | |
| Kurtosis: 3.319 | | | Cond. No. 24.2 | | | |

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

params: 0.7169381381253228 -0.017999584280879558

p-value: 0.0 0.0



<Figure size 720x444.96 with 0 Axes>

Out[]: <statsmodels.regression.linear_model.RegressionResultsWrapper at 0x7f8d24f4cb10>

Feature Selection

In []:

```
feature_list=list(df.columns)
feature_list.append('weighted Kills')
try:
    feature_list.remove('weighted_Kills')
except:
    pass
```

```
#remove features that are minimum distance related
for i in feature_list:
    if ('Distance' in i)&('Rank' not in i)&('min' in i):
        feature_list.remove(i)
        sample = all_data.shape[0]
df = all_data[feature_list][:7000000]
y = all_data['winPlacePerc'] [:7000000]

from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
Imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
import lightgbm as lgb
from sklearn.metrics import mean_absolute_error
from Utils import *
feature_importance=pd.DataFrame()
reg_param=pd.DataFrame()
regress_ret=pd.DataFrame()
for iteration in range(10):
    print(iteration)
    import time
    now=time.time()
    train_X, test_X, train_y, test_y = train_test_split(df.values, y.values, test_size=0.2)
    train_X=pd.DataFrame(train_X,columns=df.columns)
    # lgb_train=lgb.Dataset(train_X,train_y,feature_name=list(train.columns))
    #params={'task':'train','objective':'mape','boosting_type':'rf','verbose':1,
    #my_model=lgb.train(params, lgb_train, num_boost_round=50)
    my_model = lgb.LGBMRegressor(boosting_type='gbdt',objective='regression', n_estimators=50)
    my_model.fit(train_X, train_y, verbose=False)
    predictions = my_model.predict(test_X)
    from sklearn.metrics import mean_absolute_error
    pred = my_model.predict(test_X,num_iteration=my_model.best_iteration_)
    try:
        feature_importance=pd.concat([feature_importance,pd.DataFrame(my_model.feature_importances_)])
    except:
        feature_importance=pd.concat([feature_importance,pd.DataFrame(my_model.feature_importances_)])
    # reg_param=pd.concat([reg_param,pd.DataFrame([res.params['predictions']],res.clear_output(wait=True))]
    print(time.time()-now)
    print('mse:' +str(mean_absolute_error(pred, test_y)))
```

86.1338963508606
mse:0.034401369313695467

In [373...]: feature_importance.mean(axis=1).sort_values(ascending=False).head(10)

Out [373...]:

| | |
|----------------------------------|--------|
| kills_0_Place | 1299.0 |
| group.min.killsNormRank | 1146.7 |
| numGroups | 1075.6 |
| kills_1_Place | 878.6 |
| perc.gmax/mmax.killPlace | 814.1 |
| group.min.killPlace/MaxPlaceNorm | 647.4 |
| kills_2_Place | 562.4 |
| group.min.killPlace | 506.5 |
| perc.gmax/mmean.walkDistance | 424.9 |
| perc.gmax/msum.killPlace | 417.2 |

dtype: float64

In [375...]: feature_importance_original=feature_importance

In []:

In [166...]

```
feature_list=list(feature_importance_original.sort_values(0,ascending=False).head(100))
feature_list.append('weighted Kills')
try:
    feature_list.remove('weighted_Kills')
except:
    pass
#remove features that are minimum distance related
for i in feature_list:
    if ('Distance' in i)&('Rank' not in i)&('min' in i):
        feature_list.remove(i)
sample = all_data.shape[0]
df = all_data[feature_list][:700000]
y = all_data['winPlacePerc'] [:700000]
```

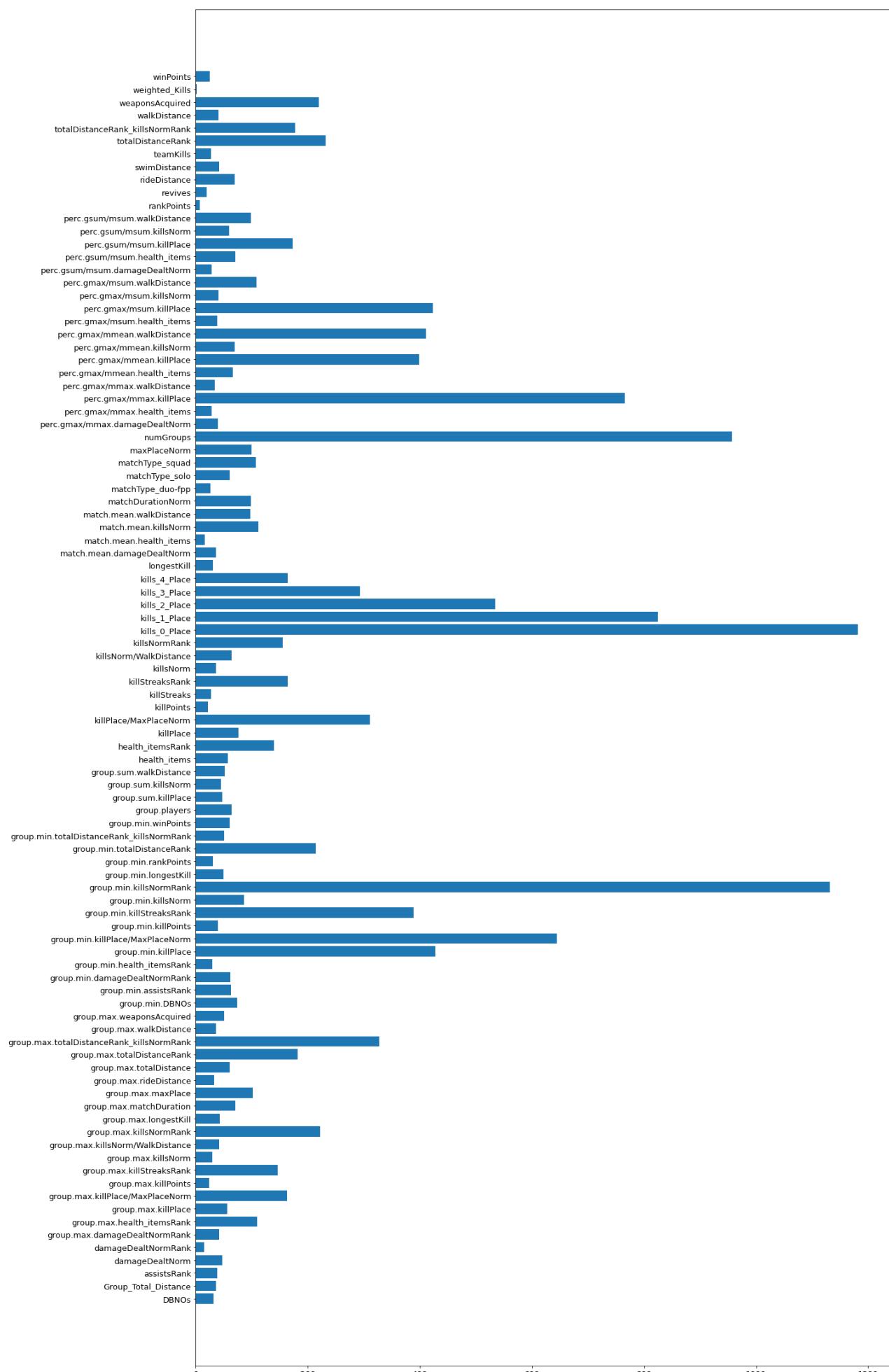
In [182...]

```
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
Imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
import lightgbm as lgb
from sklearn.metrics import mean_absolute_error
from Utils import *
feature_importance=pd.DataFrame()
reg_param=pd.DataFrame()
regress_ret=pd.DataFrame()
for iteration in range(10):
    print(iteration)
    import time
    now=time.time()
    train_X, test_X, train_y, test_y = train_test_split(df.values, y.values, test_size=0.2)
    train_X=pd.DataFrame(train_X,columns=df.columns)
    # lgb_train=lgb.Dataset(train_X,train_y,feature_name=list(train.columns))
    #params={'task':'train','objective':'mape','boosting_type':'rf','verbose':1,
    #my_model=lgb.train(params, lgb_train, num_boost_round=50)
    my_model = lgb.LGBMRegressor(boosting_type='gbdt',objective='regression', n_estimators=50)
    my_model.fit(train_X, train_y, verbose=False)
    predictions = my_model.predict(test_X)
    from sklearn.metrics import mean_absolute_error
    pred = my_model.predict(test_X)
    try:
        feature_importance=pd.concat([feature_importance,pd.DataFrame(my_model.feature_importances_,index=df.columns)])
    except:
        feature_importance=pd.concat([feature_importance,pd.DataFrame(my_model.feature_importances_,index=df.columns)])
    # reg_param=pd.concat([reg_param,pd.DataFrame([res.params['predictions']],res)])
    clear_output(wait=True)
    print(time.time()-now)
    print('mse:' +str(mean_absolute_error(pred, test_y)))
```

22.3616840839386
mse:0.034563104158023106

In [193...]

```
plt.figure(figsize=(20,40))
plt.rcParams.update({'font.size': 13})
plt.barh(feature_importance_ver2.index,feature_importance_ver2.mean(axis=1))
plt.savefig('feature_importance_10_iter_100features.jpg',bbox_inches = "tight")
```



In [186...]

```
feature_importance_ver2=feature_importance.copy()
```

In [187...]

```
feature_list=list(feature_importance_ver2.mean(axis=1).sort_values(ascending=False))
feature_list.append('weighted Kills')
try:
    feature_list.remove('weighted_Kills')
except:
    pass
#remove features that are minimum distance related
for i in feature_list:
    if ('Distance' in i)&('Rank' not in i)&('min' in i):
        feature_list.remove(i)
        sample = all_data.shape[0]
df = all_data[feature_list][:700000]
y = all_data['winPlacePerc'] [:700000]

# def split_vals(a, n : int):
#     return a[:n].copy(), a[n:].copy()

# val_perc = 0.2 # % to use for validation set
# n_valid = int(val_perc * sample)
# n_trn = len(df)-n_valid

# # Split data
# raw_train, raw_valid = split_vals(all_data, n_trn)
# X_train, X_valid = split_vals(df, n_trn)
# y_train, y_valid = split_vals(y, n_trn)
```

In [197...]

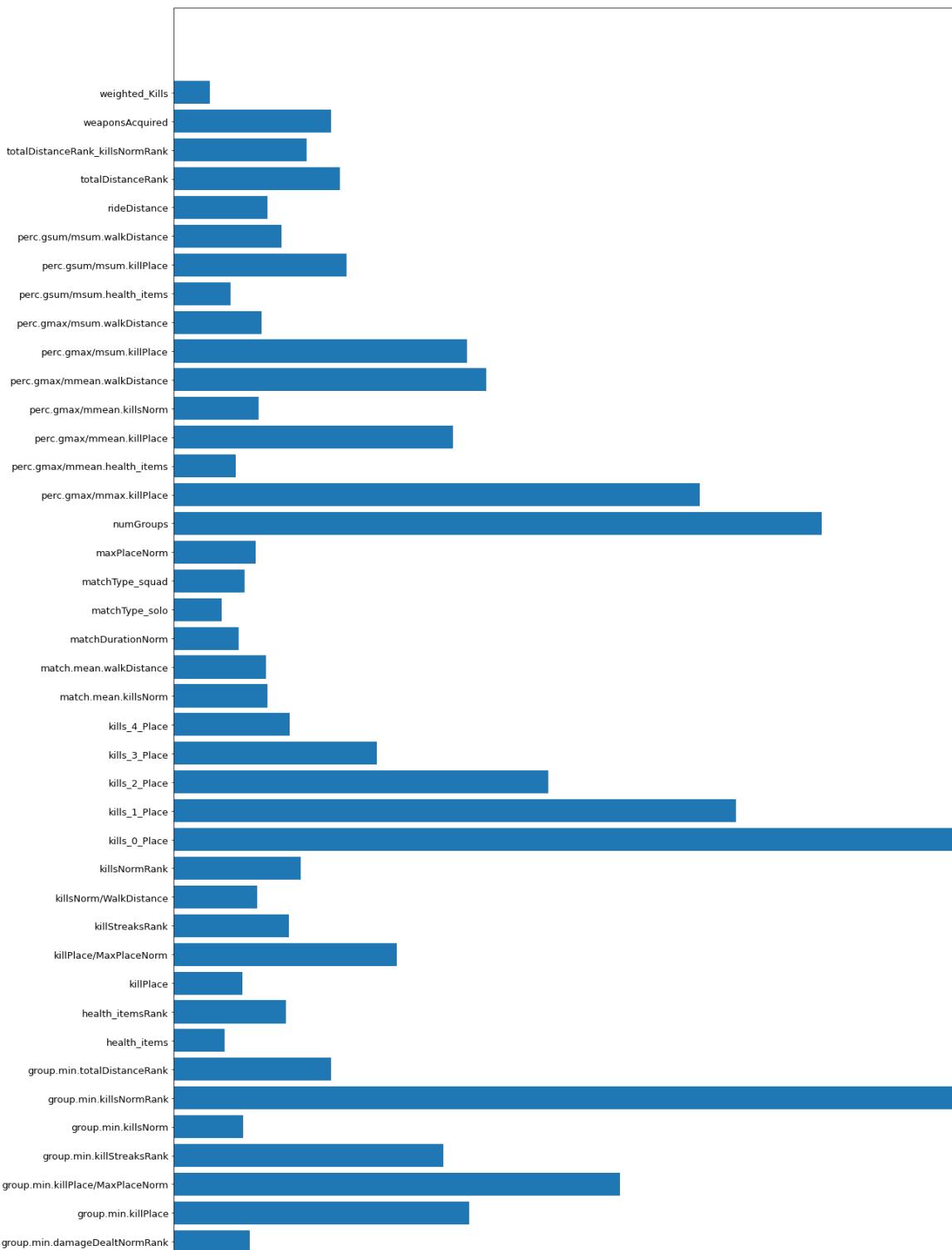
```
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
Imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
import lightgbm as lgb
from sklearn.metrics import mean_absolute_error
from Utils import *
feature_importance=pd.DataFrame()
reg_param=pd.DataFrame()
regress_ret=pd.DataFrame()
for iteration in range(10):
    print(iteration)
    import time
    now=time.time()
    train_X, test_X, train_y, test_y = train_test_split(df.values, y.values, test_size=0.2)
    train_X=pd.DataFrame(train_X,columns=df.columns)
    # lgb_train=lgb.Dataset(train_X,train_y,feature_name=list(train.columns))
    #params={'task':'train','objective':'mape','boosting_type':'rf','verbose':1,
    #my_model=lgb.train(params, lgb_train, num_boost_round=50)
    my_model = lgb.LGBMRegressor(boosting_type='gbdt',objective='regression', n_estimators=50)
    my_model.fit(train_X, train_y, verbose=False)
    predictions = my_model.predict(test_X)
    from sklearn.metrics import mean_absolute_error
    pred = my_model.predict(test_X)
    try:
        feature_importance=pd.concat([feature_importance,pd.DataFrame(my_model.feature_importances_)])
    except:
        feature_importance=pd.concat([feature_importance,pd.DataFrame(my_model.feature_importances_)]))
```

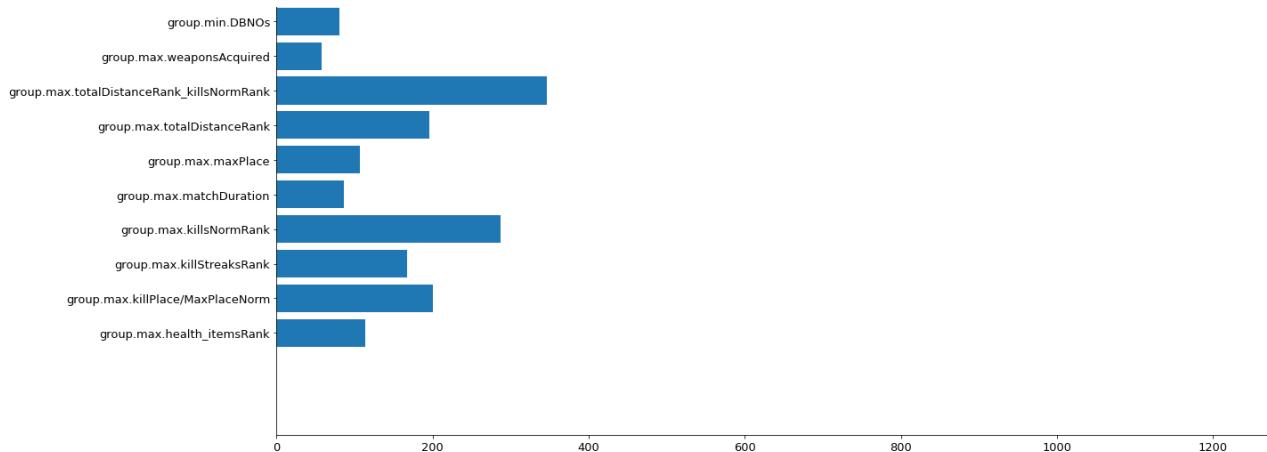
```
# reg_param=pd.concat([reg_param,pd.DataFrame([res.params['predictions']],res
clear_output(wait=True)
print(time.time()-now)
print('mse:' +str(mean_absolute_error(pred, test_y))))
```

10.111179113388062
mse:0.034531726555716666

In [198...]

```
plt.figure(figsize=(20,40))
plt.rcParams.update({'font.size': 13})
plt.barh(feature_importance.index,feature_importance.mean(axis=1))
plt.savefig('feature_importance_10_iter_50features.jpg')
```





In [200]:

```
feature_importance_ver3=feature_importance.copy()
```

In [209]:

```
feature_list=list(feature_importance_ver3.mean(axis=1).sort_values(ascending=False))
feature_list.append('weighted Kills')
try:
    feature_list.remove('weighted_Kills')
except:
    pass
#remove features that are minimum distance related
for i in feature_list:
    if ('Distance' in i)&('Rank' not in i)&('min' in i):
        feature_list.remove(i)
sample = all_data.shape[0]
df = all_data[feature_list][:7000000]
y = all_data['winPlacePerc'] [:7000000]

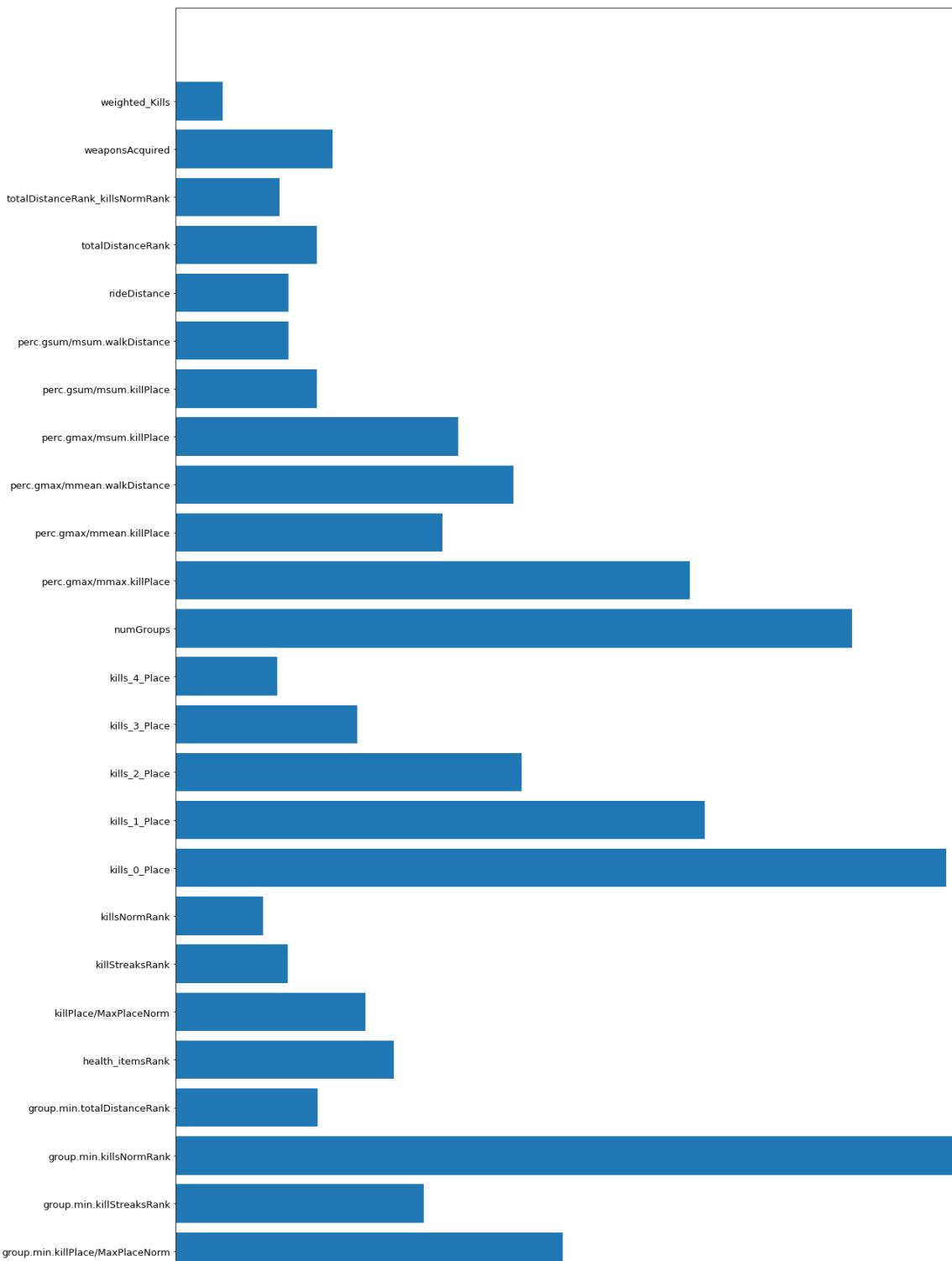
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
Imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
import lightgbm as lgb
from sklearn.metrics import mean_absolute_error
from Utils import *
feature_importance=pd.DataFrame()
reg_param=pd.DataFrame()
regress_ret=pd.DataFrame()
for iteration in range(10):
    print(iteration)
    import time
    now=time.time()
    train_X, test_X, train_y, test_y = train_test_split(df.values, y.values, test_size=0.2)
    train_X=pd.DataFrame(train_X,columns=df.columns)
    # lgb_train=lgb.Dataset(train_X,train_y,feature_name=list(train.columns))
    # params={'task':'train','objective':'mape','boosting_type':'rf','verbose':1,
    # my_model=lgb.train(params, lgb_train, num_boost_round=50)
    my_model = lgb.LGBMRegressor(boosting_type='gbdt',objective='regression', n_estimators=50)
    my_model.fit(train_X, train_y, verbose=False)
    predictions = my_model.predict(test_X)
    from sklearn.metrics import mean_absolute_error
    pred = my_model.predict(test_X,num_iteration=my_model.best_iteration_)
    try:
        feature_importance=pd.concat([feature_importance,pd.DataFrame(my_model.feature_importances_,index=test_X.columns)])
    except:
        feature_importance=pd.concat([feature_importance,pd.DataFrame(my_model.feature_importances_,index=test_X.columns)])
```

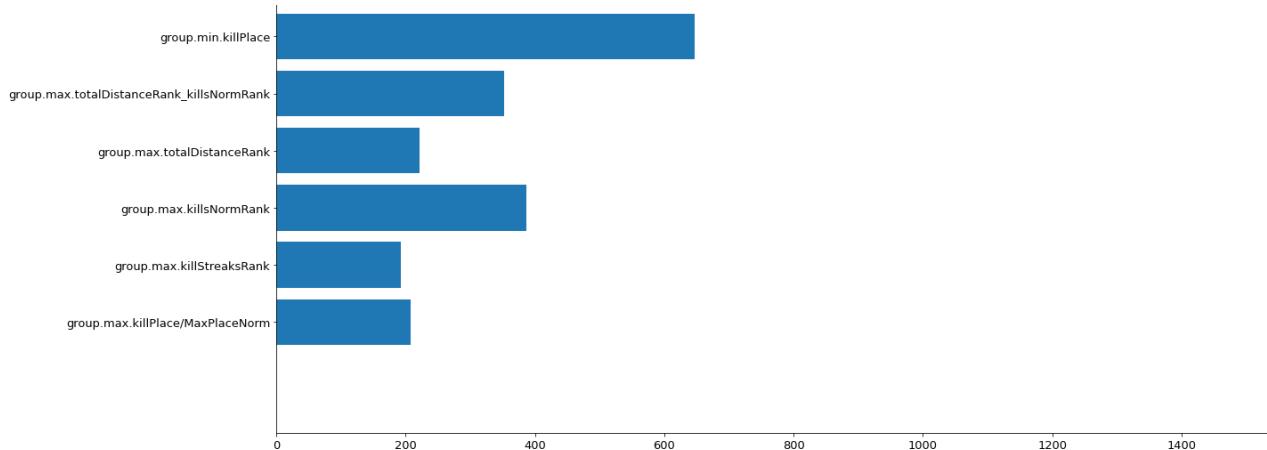
```
# reg_param=pd.concat([reg_param,pd.DataFrame([res.params['predictions']],res
clear_output(wait=True)
print(time.time()-now)
print('mse:' +str(mean_absolute_error(pred, test_y))))
```

19.653332233428955
mse: 0.034599742286705315

In [210...]

```
plt.figure(figsize=(20,40))
plt.rcParams.update({'font.size': 13})
plt.barh(feature_importance.index,feature_importance.mean(axis=1))
plt.savefig('feature_importance_10_iter_30features.jpg',bbox='tight')
```





Parameter Tuning

In [382]

```

def train(feature_list, iteration=1, new_param={}, used_samples=700000):
    feature_importance=pd.DataFrame()
    for iteration in range(iteration):
        print(iteration)
        now=time.time()
        train_X, test_X, train_y, test_y = train_test_split(df[feature_list][:us])
        train_X=pd.DataFrame(train_X,columns=df[feature_list].columns)
        # lgb_train=lgb.Dataset(train_X,train_y,feature_name=list(train.columns))
        params={'boosting_type':'gbdt','objective':'mae', 'num_leaves':150,'verb'
        for i in new_param.keys():
            params.update({i:new_param[i]})}
        my_model = lgb.LGBMRegressor(**params)
        my_model.fit(train_X, train_y, verbose=False)
        pred = my_model.predict(test_X)
        try:
            feature_importance=pd.concat([feature_importance,pd.DataFrame(my_mod
        except:
            feature_importance=pd.concat([feature_importance,pd.DataFrame(my_mod
            mse=mean_absolute_error(pred, test_y)
            print('time used:' +str(time.time()-now))
            print('mse:' +str(mse))
        return my_model, feature_importance, mse
def plot_feature_importance(feature_importance, savefig=False):
    plt.figure(figsize=(20,40))
    plt.rcParams.update({'font.size': 13})
    plt.barh(feature_importance_ver2.index,feature_importance_ver2.mean(axis=1))
    if savefig!=False:
        plt.savefig(str(savefig)+'.jpg',bbox_inches = "tight")
def plot_prediction(my_model):
    pred = my_model.predict(test_X)
    regression(pd.DataFrame({'pred':pred,'real':test_y}),summary=False)

```

In [294]

```

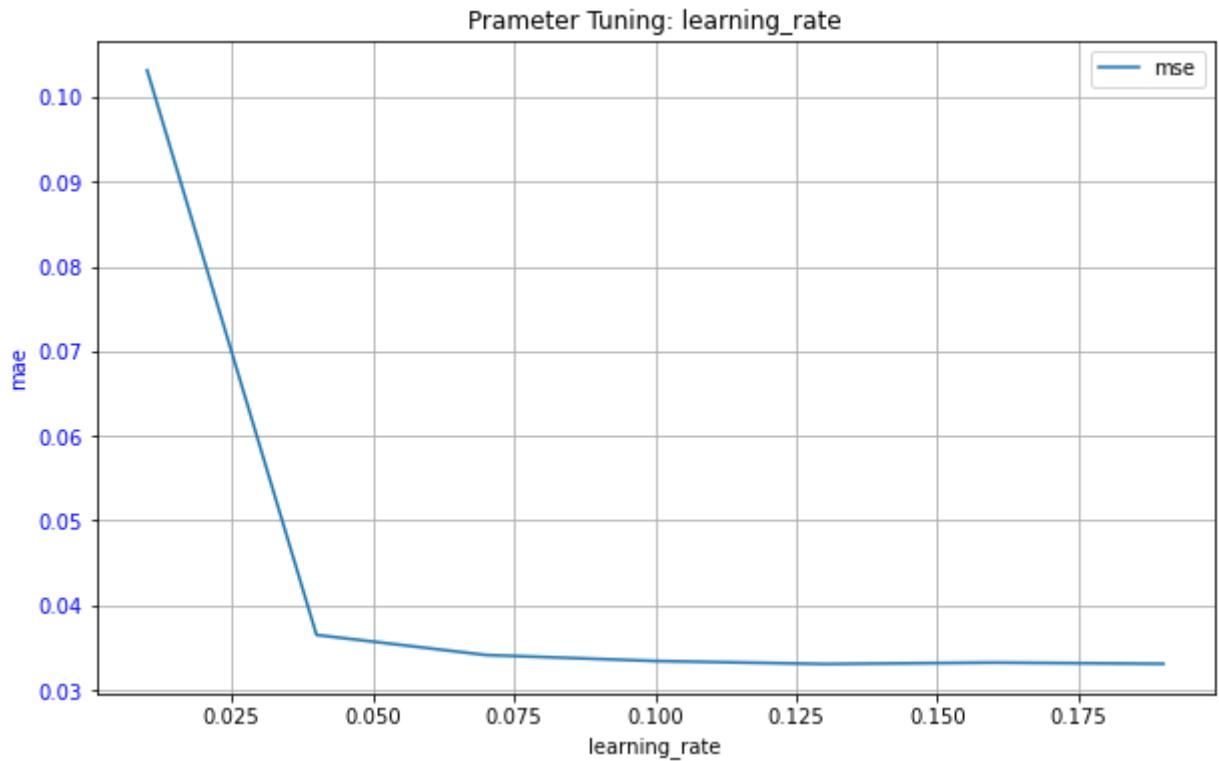
feature_list=list(feature_importance_ver2.mean(axis=1).sort_values(ascending=False)
#fine tune parameters using 50 major features
# tuning_result={}
tuning_dic={'learning_rate':[x/100 for x in list(range(1,20,3))],
            'num_leaves':list(range(20,1000,80)),
            'bagging_fraction':[x/10 for x in list(range(3,10))],
            'max_depth':list(range(3,30,3))
        }

```

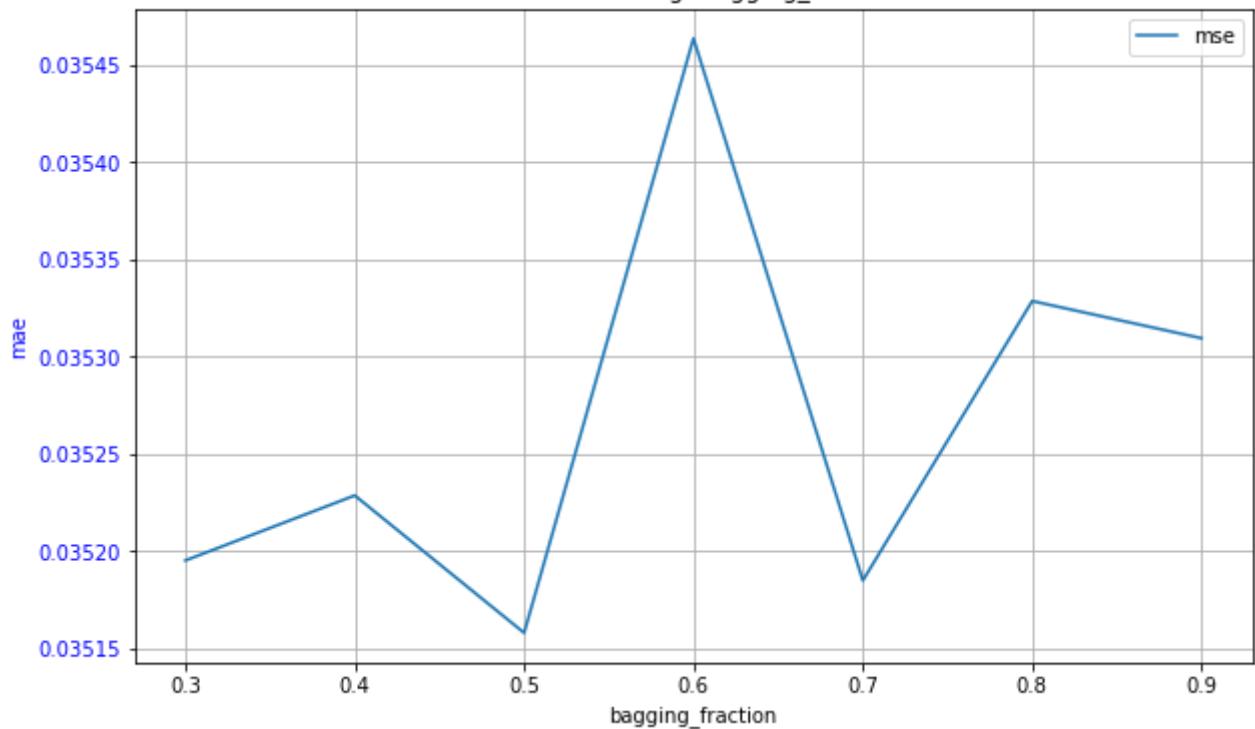
```
for param_name in tuning_dic.keys():
    if param_name not in tuning_result.keys():
        print(param_name)
        store=pd.DataFrame()
        for param_value in tuning_dic[param_name]:
            model,feature,mse=train(feature_list,new_param={param_name:param_val})
            store=store.append(pd.DataFrame({param_name:param_value,"mse":mse}))
        tuning_result.update({param_name:store})
    else:
        print(param_name+' already exists in results')
```

In [376]:

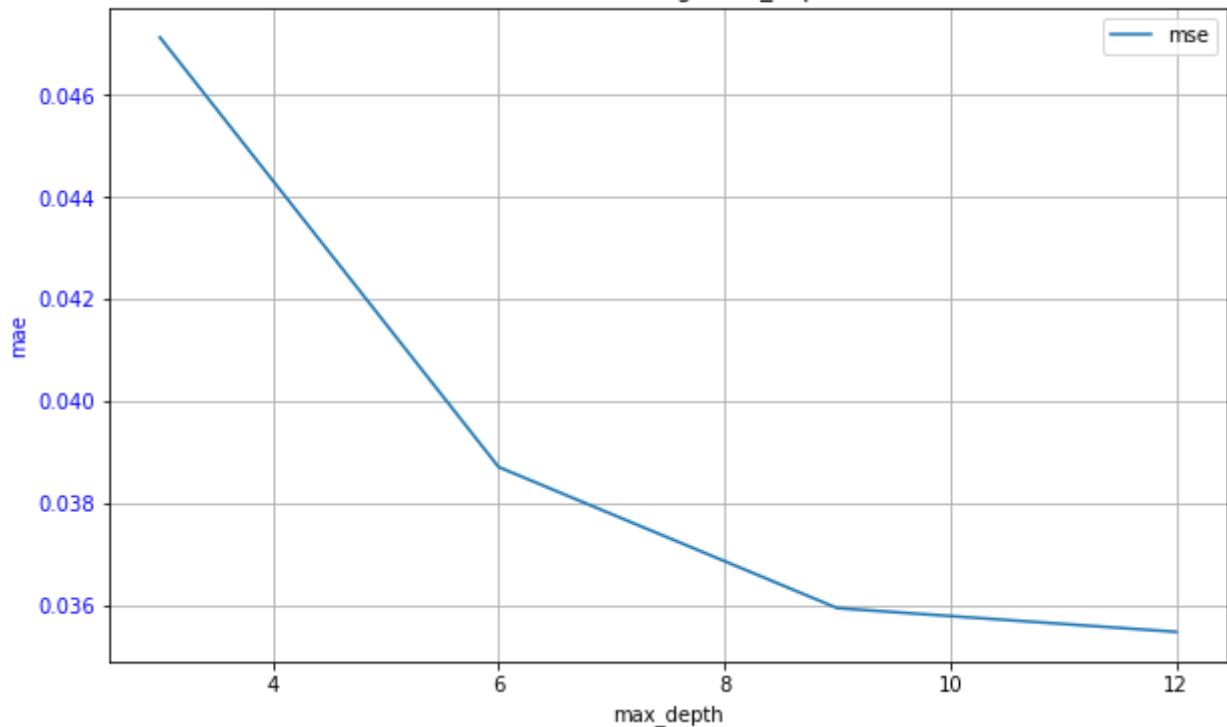
```
for param_name in tuning_result.keys():
    store=tuning_result[param_name]
    plot_line(store.set_index(param_name),ylabel='mae',xlabel=param_name,title='
```



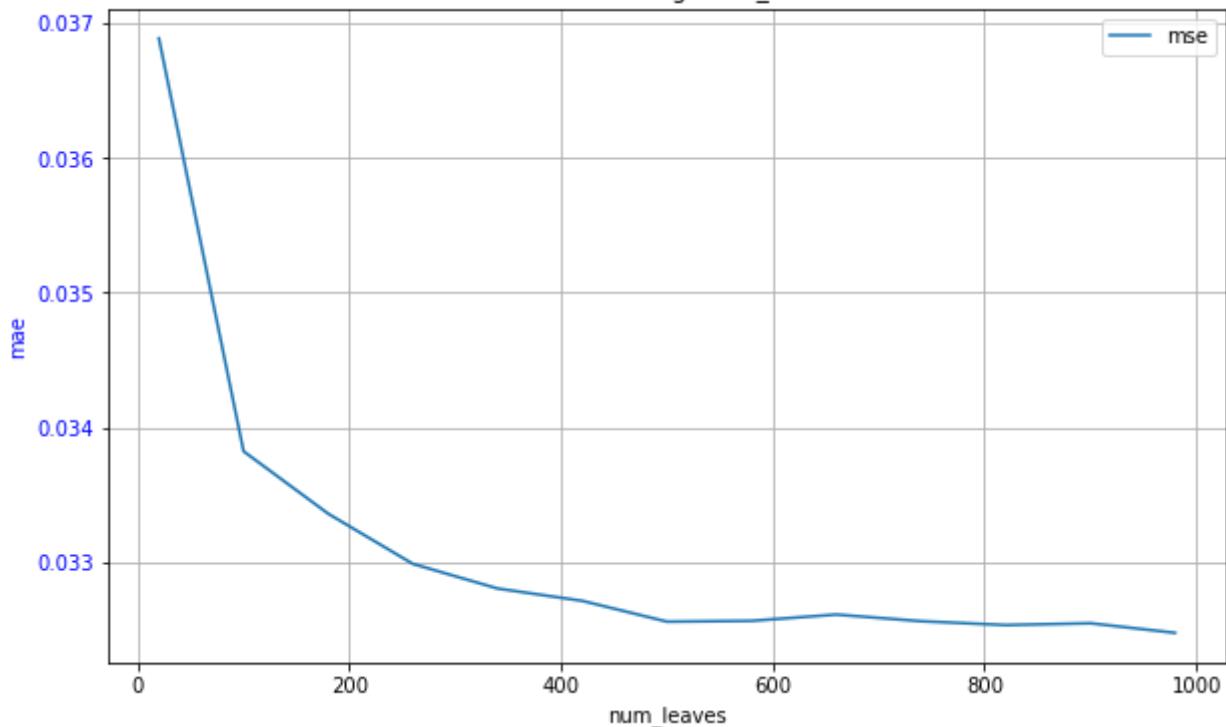
Parameter Tuning: bagging_fraction



Parameter Tuning: max_depth



Parameter Tuning: num_leaves



In [356]:

```
#fine tune parameters for feature number
feature_tuning_result={}
store=pd.DataFrame()
for num_feature in range(1,150,10):
    fureture_list=feature_importance_original.mean(axis=1).sort_values(ascending=False)
    print(num_feature)

    model,feature,mse=train(feature_list)
    store=pd.concat([store,pd.DataFrame({'num_feature':num_feature,"mse":mse}),index=False])
    feature_tuning_result.update({num_feature:store})
```

```
1
0
time used:14.359738111495972
mse:0.032622271585355515
11
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:15.470327854156494
mse:0.03278419915433369
21
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:20.974062204360962
mse:0.03281204351822167
31
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
```

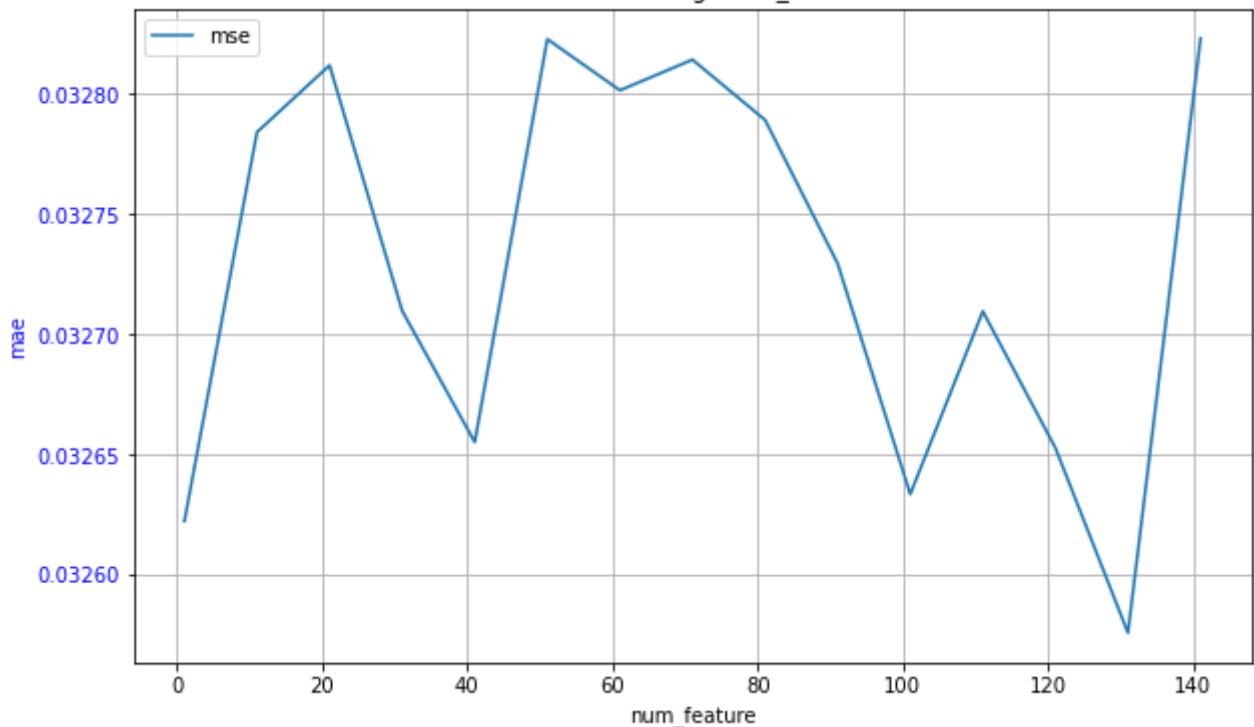
```
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:14.50774097442627
mse:0.03270986351219729
41
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:14.520225048065186
mse:0.03265514621432959
51
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:15.442805051803589
mse:0.03282294023485279
61
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:16.695987939834595
mse:0.032801621356081424
71
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:15.621287107467651
mse:0.032814380572207785
81
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:15.400895118713379
mse:0.032789268832064
91
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:15.349406003952026
mse:0.032729472750961615
101
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:20.324015140533447
```

```
mse:0.03263347004895821
111
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. C
urrent value: bagging_freq=10
time used:13.931373834609985
mse:0.0327096145302204
121
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. C
urrent value: bagging_freq=10
time used:15.633156299591064
mse:0.03265274239657525
131
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. C
urrent value: bagging_freq=10
time used:16.93601703643799
mse:0.03257574246151265
141
0
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. C
urrent value: bagging_freq=10
time used:13.581739902496338
mse:0.032823273829248004
```

In [358...]

```
param_name='num_feature'
plot_line(store.set_index(param_name),ylabel='mae',xlabel=param_name,title='Param
```

Parameter Tuning: num_feature



Model Performance Assessment

In [420...]

```
#redefine train with optimal parameters
def train(feature_list, iteration=1, new_param={}, used_samples=700000):
    feature_importance=pd.DataFrame()
    mae_list=[]
    for iteration in range(iteration):
        print(iteration)
        now=time.time()
        train_X, test_X, train_y, test_y = train_test_split(df[feature_list][:us])
        train_X=pd.DataFrame(train_X,columns=df[feature_list].columns)
        # lgb_train=lgb.Dataset(train_X,train_y,feature_name=list(train.columns))
        params={'objective':'mae','boosting_type':'gbdt','verbose':-1,'learning_rate':0.05}
        for i in new_param.keys():
            params.update({i:new_param[i]})

        my_model = lgb.LGBMRegressor(**params)
        my_model.fit(train_X, train_y, verbose=False)
        pred = my_model.predict(test_X)
        try:
            feature_importance=pd.concat([feature_importance,pd.DataFrame(my_mod
        except:
            feature_importance=pd.concat([feature_importance,pd.DataFrame(my_mod
            mae=mean_absolute_error(pred, test_y)
            print('time used:' +str(time.time()-now))
            print('mae:' +str(mae))
            mae_list.append(mae)
    return my_model, feature_importance, mse, test_y, pred
```

In [421...]

```
#using all features for training
feature_list=list(feature_importance_original.mean(axis=1).sort_values(ascending=True))
try:
    feature_list.remove('weighted_Kills')
```

```
except:
    pass
```

In [422...]

```
model,feature_importance,mae,test_y,pred=train(feature_list,used_samples=df.shape[0])
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:93.81241607666016
mae:0.031890717587039576
```

In []:

In [425...]

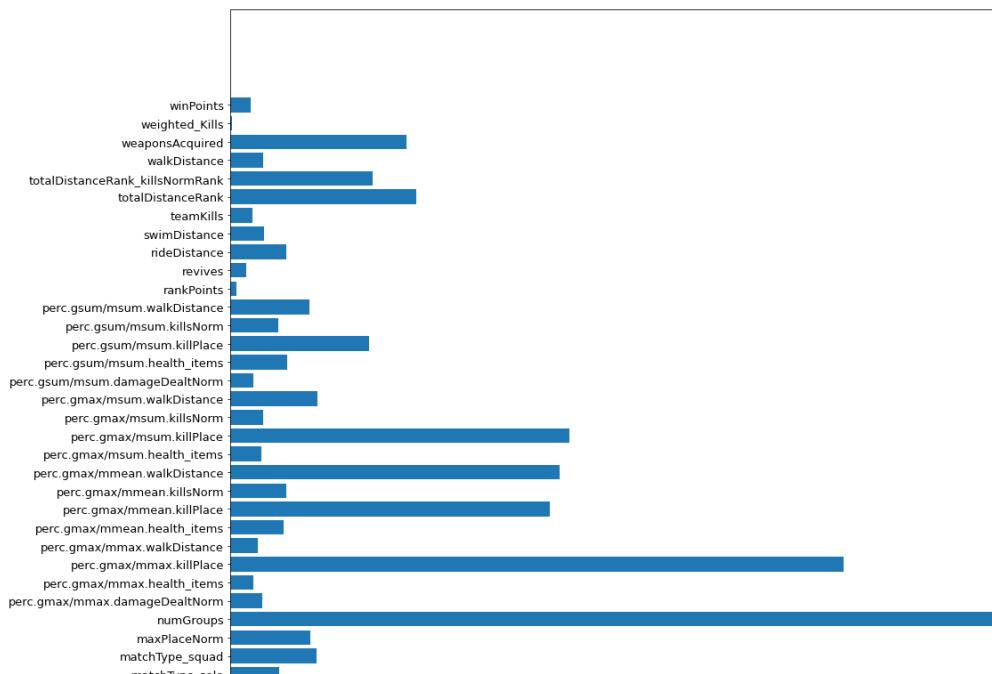
```
#using top 40 features for training in validation set
feature_list=list(feature_importance_original.mean(axis=1).sort_values(ascending=True))
try:
    feature_list.remove('weighted_Kills')
except:
    pass
```

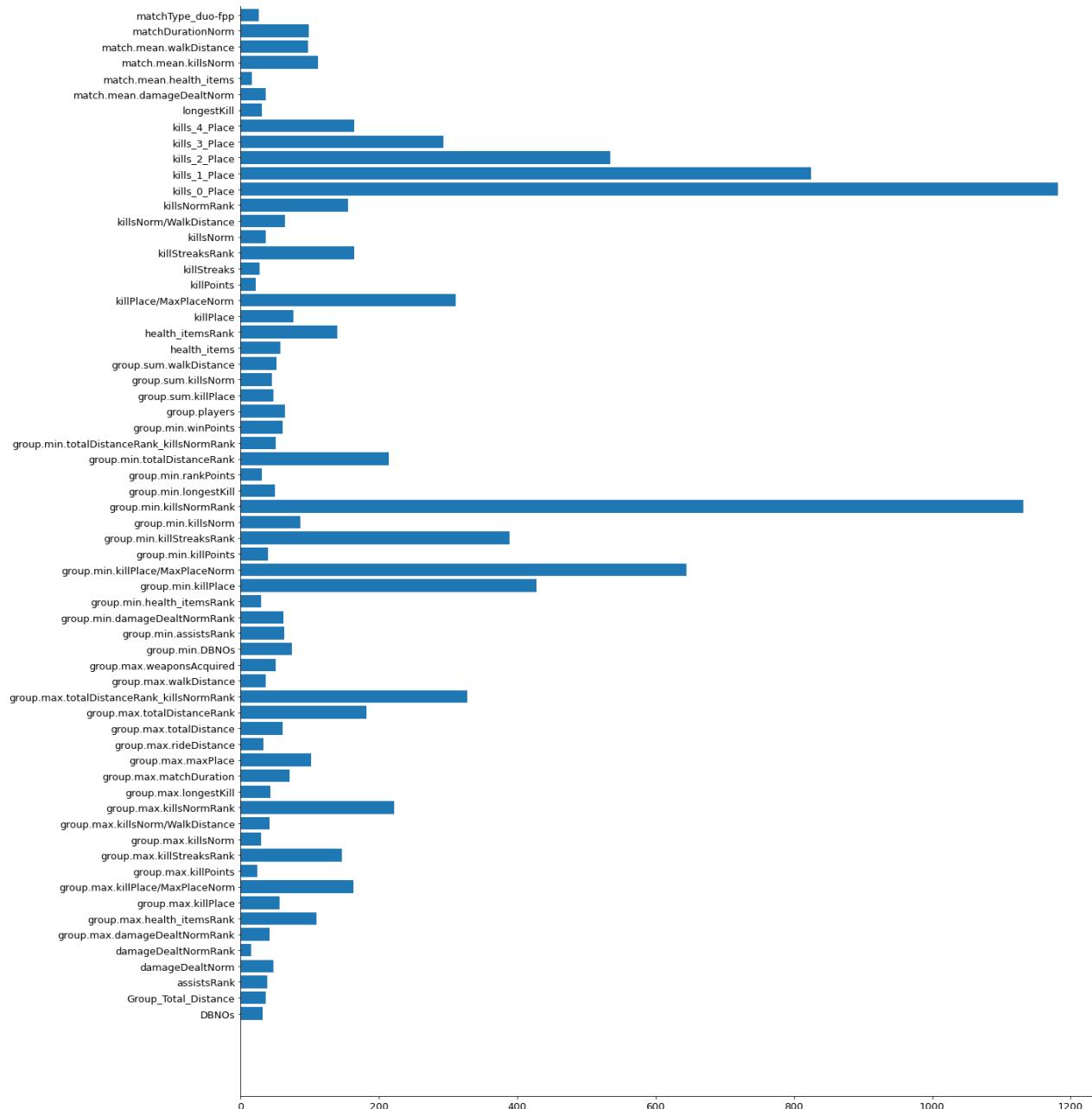
In [426...]

```
model,feature_importance,mae,test_y,pred=train(feature_list,used_samples=df.shape[0])
[LightGBM] [Warning] bagging_fraction is set=0.5, subsample=1.0 will be ignored.
Current value: bagging_fraction=0.5
[LightGBM] [Warning] bagging_freq is set=10, subsample_freq=0 will be ignored. Current value: bagging_freq=10
time used:26.874979972839355
mae:0.0321620124337536
```

In [411...]

```
plot_feature_importance(feature_importance,savefig='final validation 50 features')
```





In [413]:

```
regression(pd.DataFrame({'pred':pred, 'y':test_y}))
```

OLS Regression Results

| | | | |
|--------------------------|------------------|----------------------------|------------|
| Dep. Variable: | y | R-squared: | 0.977 |
| Model: | OLS | Adj. R-squared: | 0.977 |
| Method: | Least Squares | F-statistic: | 1.095e+07 |
| Date: | Tue, 07 Dec 2021 | Prob (F-statistic): | 0.00 |
| Time: | 17:07:10 | Log-Likelihood: | 4.3671e+05 |
| No. Observations: | 260753 | AIC: | -8.734e+05 |
| Df Residuals: | 260751 | BIC: | -8.734e+05 |
| Df Model: | 1 | | |
| Covariance Type: | nonrobust | | |

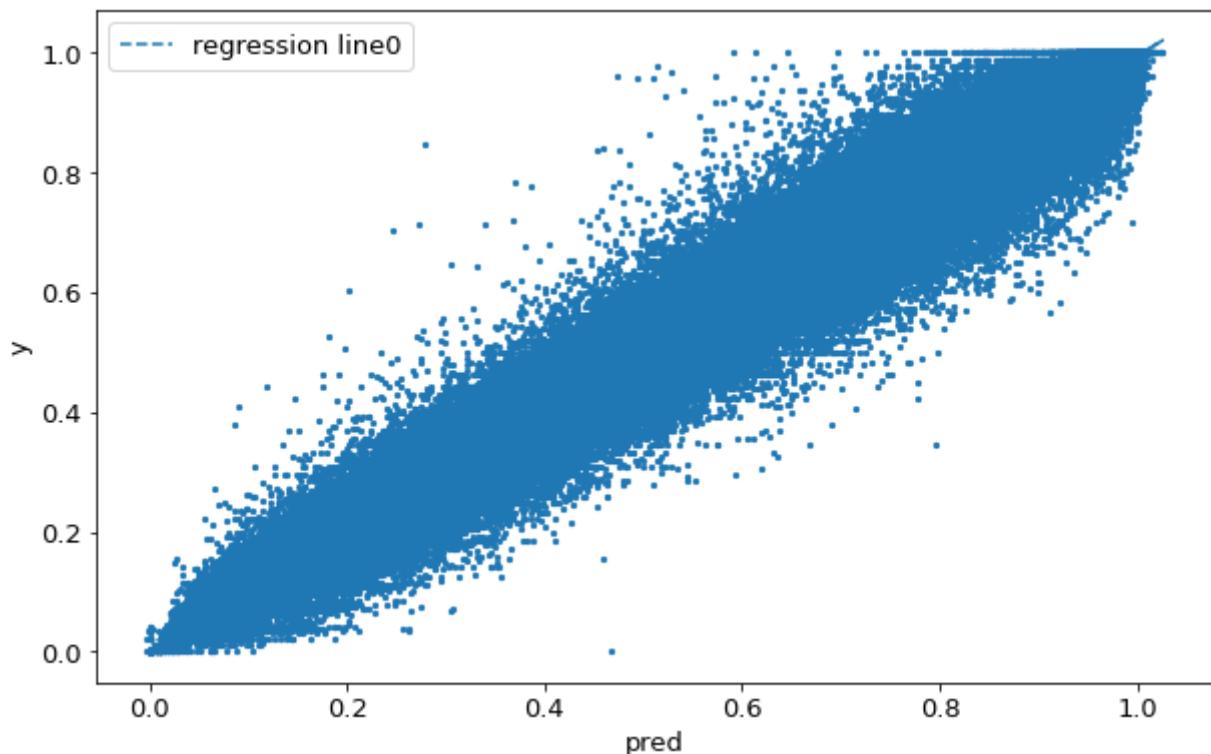
| | coef | std err | t | P> t | [0.025 | 0.975] |
|--------------|--------|---------|----------|-------|--------|--------|
| const | 0.0025 | 0.000 | 14.447 | 0.000 | 0.002 | 0.003 |
| pred | 0.9931 | 0.000 | 3308.854 | 0.000 | 0.992 | 0.994 |

| | | | |
|-----------------------|-----------|--------------------------|------------|
| Omnibus: | 21497.334 | Durbin-Watson: | 2.001 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 137225.613 |
| Skew: | -0.023 | Prob(JB): | 0.00 |
| Kurtosis: | 6.554 | Cond. No. | 4.29 |

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
params: 0.002521729369470253 0.9930642847897917
p-value: 2.7243862990920514e-47 0.0
```



<Figure size 720x444.96 with 0 Axes>

Out[413...]: <statsmodels.regression.linear_model.RegressionResultsWrapper at 0x7f896d93ec50>

In []:

In []:

In []: